Low Time-Consuming Implementation of Predictive Path-Tracking Control for a "Synchro-drive" Mobile Robot

J.E. Normey-Rico*, J. Gómez-Ortega**, I. Alcalá-Torrego** and E.F. Camacho**

*Dpto. de Automação e Sistemas. Federal Univ. of Santa Catarina, Brazil. E-mail:julio@cartuja.us.es
**Dpto. Ingeniería de Sistemas y Automática, Univ. de Sevilla.
Escuela Superior de Ingenieros, Camino de los Descubrimientos s/n, 41092 Sevilla, Spain.
Fax: +34-954463153, email: juango@cartuja.us.es

Abstract

This paper presents the application of a Smith predictor based generalised predictive controller (SPGPC) to the path-traking problem of a synchro drive mobile robot. The higher robustness of the SPGPC algorithm, if compared to a generalised predictive controller (GPC) based on an optimal predictor, allows the use of a simple model for the mobile robot. Thus, the final control law is simpler and less time consuming than the GPC, mantaining the same nominal performance and better robustness. Experimental tests carried out on a Nomad 200 mobile robot validate the performance of the proposed strategy.

1 Introduction

One of the most important issues in the field of mobile robots is the path-tracking (PT) problem. The objective of PT is concerned with driving the vehicle as close as possible to a desired path, with least control effort. This is an important point because allows the operation with reduced energy consumision.

Several applications of predictive control techniques to the PT problem can be found in the literature. In [1] a Generalised Predictive Control is chosen as the control strategy with a linear model of the robot kinematics for the prediction of the future robot positions. Also a predictive technique is proposed in [2], [3], but a non linear model of the robot is considered and a multilayer perceptron is used to implement the predictive controller in real time.

The problem that we raise in this paper is that of driving a mobile robot to follow a previously calculated reference path, defined as a set of consecutive points. In this work a sinchro drive mobile robot is used and a Smith Predictor Based Generalised Predictive Controller is proposed to solve the path tracking problem. This controller has the same performance as the normal GPC but is more robust and allows the use of simple models.

The paper is organised as follows: In section 2 a description of the proposed SPGPC is presented. In section 3 the model considered for the mobile robot kinematics is presented and in secion 4 the way in which the SPGPC is applied to the PT problem is shown. In section 5 the advantages in the implementation of the control algorithm are presented and in section 6 experimental results on a Nomad 200 mobile robot are shown. The paper ends with the conclusions.

2 The Smith Predictor Based Generalised Predictive Controller

The SPGPC was proposed in [4] and was applied to the mobile robot path-tracking problem in [5]. As in the normal Generalised Predictive Control (GPC) [6], the objective of the algorithm is to compute a future control action sequence $(U(t), U(t+1|t), ..., U(t+N_u-1|t))$ in order to minimize (with respect to the control variables) a cost function J of the form:

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

where $\mu(i)$ and $\lambda(i)$ are respectively the error and control weighting factors; $\hat{Y}(t+i|t)$ are the future predicted system outputs; $Y_d(t+i)$ are the desired references; N_u is the control horizon and $N = N_2 - N_1$ is the output prediction horizon.

After the control sequence is obtained, a *receding horizon* approach is considered. This consists of only applying the first calculated control action $U(t)$ and to repeat this process at every sampling interval. A block diagram of the control scheme is shown in figure 1.

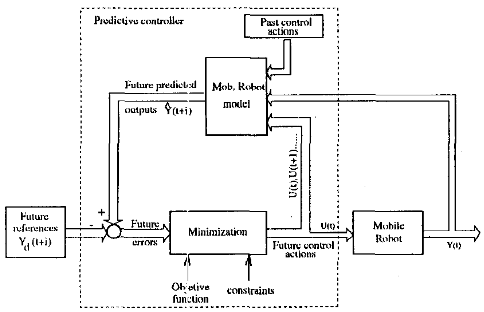

Figure 1: The predictive controller scheme.

The differences between the traditional GPC and the SPGPC used here are: (i) the GPC uses the same model of the plant and disturbances (a CARIMA model) both for computing the predictions and the control law while in the SPGPC an incremental model is used for the computation of the control law and an open loop model of the plant is used to compute the predictions; (ii) in the SPGPC the prediction of the output of the plant at time $t + d - i$ is corrected adding the error between the output and the prediction at time $t - i$, where d is the delay of the plant:

$$\hat{Y}_{SP}(t+d-i|t) = \hat{Y}(t+d-i|t) + Y(t-i) - \hat{Y}(t-i|t)$$

where $\hat{Y}(t + d - i|t)$ and $\hat{Y}(t - i)$ are respectively the prediction of the output of the plant at time $t + d - i$ and $t - i$ computed using the open loop model; $Y(t-i)$ is the output of the plant at time $t - i$ and $\hat{Y}_{SP}(t + d - i|t)$ is the obtained corrected prediction used by the SPGPC.

When the model of the plant is linear and no constraints are considered the final control law is linear. It is easy to show ([7]) that the incremental control action is proportional to the error between the free response f and the reference w, where the matrix gain (K) is a function of the parameters of the controller and the model of the plant:

$$U(t) = U(t-1) + K(f - w)$$

As the GPC and SPGPC use the same procedure to compute the control law the gain matrix K will be the same for both algorithms and only the free response f will be computed in a different maner. Then, it is easy to show (see [4] and [5]) that the control scheeme for both algorithms can be drawn as in the block diagram of figure 2.

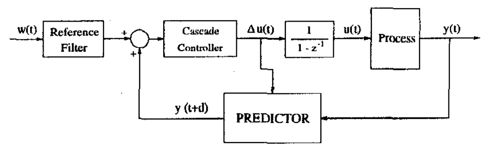

Figure 2: GPC and SPGPC control scheme.

In this diagram, the prediction and the rest of the control law are computed in two different steps. In the GPC, the block "predictor" is an optimal predictor [6] while in the SPGPC the same block is a Smith predictor [8]. Using this approach [5] showed that the SPGPC has the same nominal performance and better robustness than the GPC. In fact, the robust stability boundary of the SPGPC has higger values than the robust stability boundary of the SPGPC, principally at high frequencies. Usually, in real applications, low order tranfer functions are used to model the behaviour of the plant and there are errors in the estimation of the delay of the system; thus the most common uncertainties are high frequency unmodelled dynamics. The nominal performance of the SPGPC and GPC is the same because when the model used in the controller is perfect the predictor has no influence in the behaviour of the closed-loop. This feature of the proposed controller allows the use of a simpler model to predict the future behaviour of the mobile robot. This simpler model allows also a simpler final control law as it is analysed in the next section.

3 Mobile robot kinematic model

The SPGPC proposed in this work has been inplemented in a Nomad 200 mobile robot from Nomadic Technologies (see figure 3). This robot has a *synchro drive* type locomotion system which consist of three drive wheels which orientations vary simultaneously.

The robot also provides a position estimation system based on odometry; that is, the relative robot position is estimated from the number of wheels turns and the wheels steering angle, both computed from the information provided by incremental encoders attached to the drive and steering motors. It is well known that odometry is a technique which has an accumulative error which implies the need for update

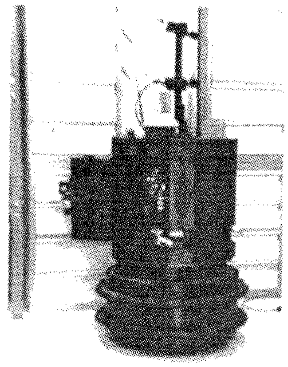

Figure 3: The Nomad 200 mobile robot

the estimation from enviromental data provided from sensor systems with a predetermined frequency. This problem is considered to be uncoupled with the path tracking problem and has not been accomplished in this work. Thus reference paths no longer than 15 m of lenght has been used.

The complete kinematic model of the robot is given by the following non-linear equations, where x_g, y_g are the coordinates of the robot in a global reference frame and δ is the wheels steering angle:

If $\dot{\delta}(t) \neq 0$, then:

$$\delta(t+1) = \delta(t) + \dot{\delta}(t)T$$

$$x_g(t+1) = x_g(t) + \frac{V}{\dot{\delta}(t)}(\sin(\delta(t) + \dot{\delta}(t)T) - \sin(\delta(t)))$$

$$y_g(t+1) = y_g(t) - \frac{V}{\dot{\delta}(t)}(\cos(\delta(t) + \dot{\delta}(t)T) - \cos(\delta(t)))$$

If $\dot{\delta}(t) = 0$, then:

$$\begin{aligned} \delta(t+1) &= \delta(t) \\ x_g(t+1) &= x_g(t) + VT\ cos\ \delta(t) \\ y_g(t+1) &= y_g(t) + VT\ sen\ \delta(t) \end{aligned}$$

$\dot{\delta}(t)$ follows its reference $\dot{\delta}_r(t)$ with a non linear dynamics determined by:

$$\dot{\delta}(t+1) = \ddot{\delta}T sign(\dot{\delta}_r(t+1) - \dot{\delta}(t)) + \dot{\delta}(t)$$

If $(\dot{\delta}_r(t+1) < \dot{\delta}(t))$ and $((\dot{\delta}(t+1) < \dot{\delta}_r(t+1))$

$$\dot{\delta}(t+1) = \dot{\delta}_r(t+1)$$

If $((\dot{\delta}_r(t+1) > \dot{\delta}(t))$ and $((\dot{\delta}(t+1) > \dot{\delta}_r(t+1))$

$$\dot{\delta}(t+1) = \dot{\delta}_r(t+1)$$

where $\ddot{\delta}$ is the steering acceleration and it is constant. The real performance of $\dot{\delta}(t)$ when step changes in the reference are considered is shown in figure 4.

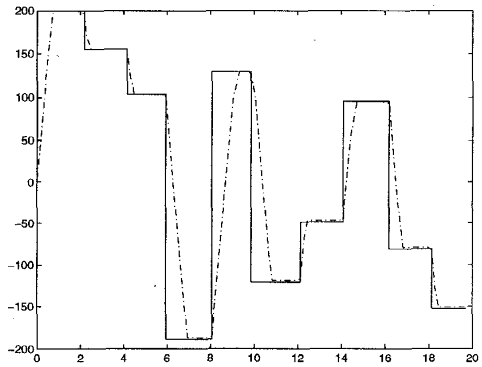

Figure 4: Behaviour of $\dot{\delta}(t)$ of the Nomad 200 mobile robot for step changes in the reference $\dot{\delta}_r(t)$

For the predictive algorithm it is necessary to use a linear model to describe the relation between $\dot{\delta}(t)$ and $\dot{\delta}_r(t)$:

$$\dot{\delta}(t) = \frac{B}{D}\dot{\delta}_r(t)$$

where B and D are polynomials in the back shift operator z^{-1}.

To complete the model needed by the SPGPC to predict the robot future positions and steering angles, a linearised model for the robot kinematic must be used. The linear model used has been obtained from the hypothesis that the increments in the wheels steering between two sampling times, $\Delta\delta$, are small. Thus, a local reference frame, attached to the mobile robot, has been used (see figure 5). Also a simplified model for the dinamics of $\dot{\delta}$ has been used in order to reduce the time needed for the computation of the control law. This simplified model consist of a step plus a delay d. The good robustness properties of the SPGPC allows the use of this reduced model without changes in the controller performance.

The linear model obtained is as follows:

$$\begin{bmatrix} \delta(t) \\ y_r(t) \end{bmatrix} = \frac{1}{(1-z^{-1})} \begin{bmatrix} T \\ \frac{VT^2}{2} \end{bmatrix} \dot{\delta}(t-1-d)$$

where $\delta(t)$ is the robot wheels steering angle, $y_r(t)$ is the robot position in the local reference frame (see figure 5), V is the robot linear velocity, which has been considered constant and T is the sampling time.

This linearised model is valid only for small $\Delta\delta$ values. Thus, if the robot is not positioned on the reference path (for example at the initial position) approaching paths will be needed. In this paper the approaching paths has been computed using a strategy called *pure pursuit* [9]. Thus, the N future reference points needed for the SPGPC strategy are computed

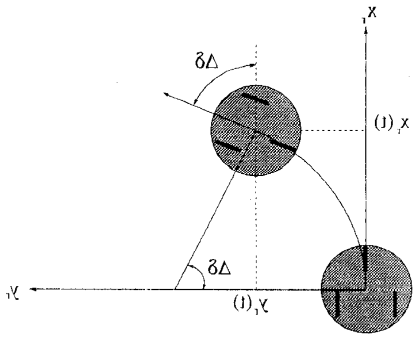

Figure 5: Local reference model for the mobile robot Nomad 200

at each sampling time as if the mobile robot would be controlled by a proportional controller calculated with the pure pursuit strategy.

The control variable used has been the steering velocity $\dot{\delta}$ which is related to the robot curvature γ by:

$$\dot{\delta} = \gamma V$$

Althought the system should has single input/single output (note that $y_r(t) = (VT/2)\delta(t)$), it has been noticed experimentally that the system performance is much better when two outputs ($\delta(t)$ and $y_r(t)$) are considered. This is because the approaching path depends on the robot heading. Thus an error in $\delta(t)$ leads to not well fitted approaching trajectories. On the other hand, if the error $y_r(t)$ is not penalised, the robot could not approximate the desired global path.

4 The SPGPC Structure for the NOMAD Mobile Robot

In this work, the SPGPC parameters are chosen in the following manner: $N_u = N$, $N_1 = d+1$ and $N_2 = d+1+N$. In order to compute the future control sequence $U(t)$, $U(t+1)$,...it is necessary to compute $\hat{Y}(t+j \mid t)$. To compute the coefficients of the control law an incremental model is used to obtain $\hat{Y}(t+d+j \mid t)$:

$$\begin{aligned} \hat{\delta}(t+j \mid t) &= 2\hat{\delta}(t+j-1 \mid t) - \\ -\hat{\delta}(t+j-2 \mid t) &+ T\,\Delta\,\dot{\delta}(t+j-d-1) \end{aligned} \qquad (1)$$

$$\begin{aligned} \hat{y}(t+j \mid t) &= 2\hat{y}(t+j-1 \mid t) - \\ -\hat{y}(t+j-2 \mid t) &+ \frac{VT^2}{2}\,\Delta\,\dot{\delta}(t+j-d-1) \end{aligned} \qquad (2)$$

or in a vectorial form:

$$\begin{bmatrix} \hat{\delta}(t+d+1 \mid t) \\ \hat{\delta}(t+d+2 \mid t) \\ \vdots \\ \hat{y}(t+d+N \mid t) \end{bmatrix} = G \begin{bmatrix} \Delta\dot{\delta}(t) \\ \Delta\dot{\delta}(t+1) \\ \vdots \\ \Delta\dot{\delta}(t+N-1) \end{bmatrix} +$$

$$S \begin{bmatrix} \hat{\delta}(t+d \mid t) \\ \hat{\delta}(t+d-1 \mid t) \\ \hat{y}(t+d \mid t) \\ \hat{y}(t+d-1 \mid t) \end{bmatrix} \qquad (3)$$

where G and S are constant matrices. Using this relation in the expresion of J, the minimization of the cost function gives:

$$M \begin{bmatrix} \Delta\dot{\delta}(t) \\ \Delta\dot{\delta}(t+1) \\ \vdots \\ \Delta\dot{\delta}(t+N-1) \end{bmatrix} = P_1 \begin{bmatrix} \hat{\delta}(t+d \mid t) \\ \hat{\delta}(t+d-1 \mid t) \\ \hat{y}(t+d \mid t) \\ \hat{y}(t+d-1 \mid t) \end{bmatrix} +$$

$$+P_2 \begin{bmatrix} w(t+d+1) \\ w(t+d+2) \\ \vdots \\ w(t+d+2N) \end{bmatrix} \qquad (4)$$

where $w(t)$ represents the future references in the steering and position error $y(t)$ and M, P_1 and P_2 are constant matrices.

Thus the final control law is given by:

$$\Delta\dot{\delta}(t) = l_{11}\hat{\delta}(t+d \mid t) + l_{12}\hat{\delta}(t+d-1 \mid t) + l_{21}\hat{y}(t+d \mid t) +$$

$$l_{22}\hat{y}(t+d-1 \mid t) + \sum_{i=1}^{2N} f_i w(t+d+i) \qquad (5)$$

where the coefficients l_{ij} and f_i are functions of V, T, $\mu(i)$ and $\lambda(i)$ and the predictions are now obtained using the open loop model:

$$\hat{\delta}(t+j \mid t) = \hat{\delta}(t+j-1 \mid t) + T\dot{\delta}(t+j-d-1) \qquad (6)$$

$$\hat{y}(t+j \mid t) = \hat{y}(t+j-1 \mid t) + VT^2/2\dot{\delta}(t+j-d-1) \qquad (7)$$

and the correction ($i = 0, 1$):

$$\hat{\delta}(t+d-i \mid t) = \hat{\delta}(t+d-i \mid t) + \delta(t-i) - \hat{\delta}(t-i) \qquad (8)$$

$$\hat{y}(t+d-i \mid t) = \hat{y}(t+d-i \mid t) + y(t-i) - \hat{y}(t-i) \qquad (9)$$

A block diagram of this stucture is shown in figure 4.

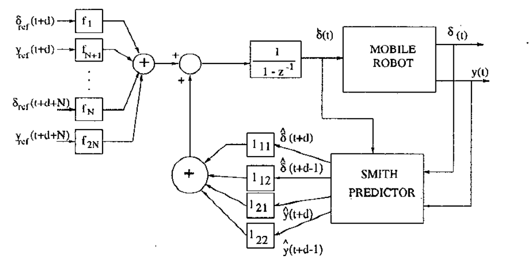

Figure 6: Control structure of the SPGPC controller for the mobile robot

5 Computation of the control law

As was mentioned in previous sections, the SPGPC allows the use of a low order linear model for the mobile robot maintaining both a good performance and robustness. As will be shown here the simpler model used for the prediction and the simpler structure of the Smith predictor allows a considerable reduction in the time consumed by the algorithm if compared to a normal GPC.

It is a known result that the complexity of the control algorithm increases with the order of the model used in the predictor. After the computation of the gains of the controller, the number of products and additions that have to be done is directly related to the order of polinomials B and D. Suposing that the coeficients of the linear model of the robot do not vary with time, the constant matrix K of the SPGPC (or GPC) can be computed once and only the predicted outputs must be computed at each sample time in order to obtain the free response f. Thus the complexity of the predictor will define the amount of time necessary for the computations. If a less robust algorithm is used (like the GPC) the order of the transfer function B/D must be higher than in the SPGPC case in order to reduce the uncertainties at high frequencies. In the SPGPC case it is possible to use a simpler pure delay model for the transfer function B/D maintaining a good performance and robustness.

The other important difference between the two predictive controllers is that in the GPC it is necessary to update all the past values of the predicted outputs at each period in order to compute the future predictions. As the values of $\hat{y}_r$ are local coordenates (they are refered to the local axis at each t) it is necessary to use two coordenate transformations to compute each value of $\hat{y}_r(t-i)$ at each sampling period (to change from the local axis in time $t-1$ to the global axis and then from the global axis to the local axis at time t). This leads to a high number of computations for the prediction of y_r. Note that δ is a global variable and does not need any coordenate transformations (this is valid also for the SPGPC).

For the SPGPC it is not necessary to update the previous predicted values of $\hat{y}_r$ to compute the future predictions, because the Smith predictor uses an open loop model of the plant. On the other hand, errors $y_r - \hat{y}_r$ and $\delta - \hat{\delta}$ must be computed in order to correct the predictions (see equations 8 and 9). Note that as the local axis are fixed to the mobile robot $\hat{y}_r(t) = 0$. Also the value of $\hat{y}_r(t-1)$ is computed directly. To show this denote $\hat{y}_r^t(t-1)$ the local predicted value of $y_r(t-1)$ refered to the coordenate axis of instant t. If small changes in δ are considered (as is the case) the diagram of figure 7 can be used to compute $\hat{y}_r^t(t-1)$.

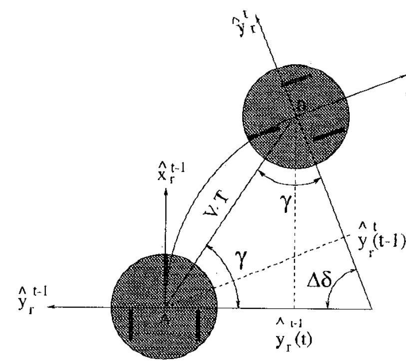

Figure 7: Diagram for the computation of the predictions

As $\Delta\delta$ is small and a *pure-pursuit* strategy is used, AB can be considered as a small arc of a circle. Then, the ordenate of A in the new reference frame (located in B) is $\hat{y}_r^t(t-1) = VTcos\gamma$ and also $\hat{y}_r^{t-1}(t) = VTcos\gamma$, that is, the ordenate of B in the reference frame located in A. Thus $\hat{y}_r^t(t-1) = \hat{y}_r^{t-1}(t)$ that has been computed in the previous step. So, for this algorithm no changes of coordenates are necessary for the computation of the predictions.

Now it is clear that using the SPGPC the order of the predictor is lower and the computation of the predicted values is simpler. As this procedure is done at each sampling time, the implementation of the proposed SPGPC is simpler and less time consuming than the GPC.

6 Experimental Results

As a test bed a Nomad 200 synchro-drive mobile robot has been used. For the experimantal tests the controller parameters have been chosen as follows: $\lambda = 0.5$, $\mu_\delta = 1$, $\mu_y = 1$, $N = 20$, look ahead $= 0.6$ meters, $V = 0.4\ m/s$, $d = 1$ and the steering aceleration $\ddot{\delta} = 30$ degrees$/s^2$. The sampling time T was chosen equal to 0.2 seconds because of the time constraints of the computational system used by the mobile robot, and it is not due to the control algorithm complexity.

Figures 8 and 9 show the quite good performance of the proposed controller for two different paths. Note that, in both cases, the reference path chosen has small curvatures, which makes more dificult to follow the reference.

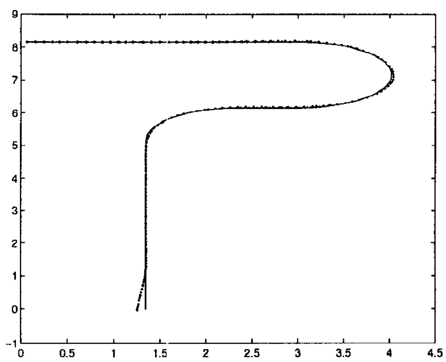

Figure 8: Mobile robot path-tracking for the SPGPC. Path reference in solid line.

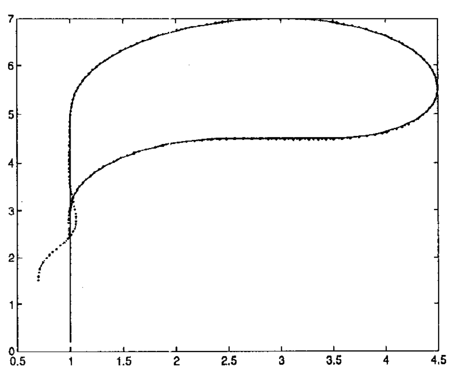

Figure 9: Mobile robot path-tracking for the SPGPC. Path reference in solid line.

7 Summary and Conclusions

A Smith Predictor based Generalised Predictive Controller has been proposed for a mobile robot Path Tracking problem. The robustness of the controller allows the use of a simpler model in the predictor given a good performance and a low time consuming algorithm. Some experimental results have shown the good performance of the proposed SPGPC.

Acknowledgments

The authors would like to acknowledge the *CICYT* (contract TAP95-0307) and *CAPES* (contract BEXO 448-95/6) for funding this work.

References

[1] A. Ollero, A. García-Cerezo, and J. L. Martínez, "Fuzzy Supervisory Path Tracking of Mobile Robots", in *Proc. First IFAC Int. Workshop*, Hampshire, 1993.

[2] J. Gómez Ortega and E. F. Camacho, "Neural Network MBPC for Mobile Robots Path Tracking", *Robotics and Computer Integrated Manufacturing Journal*, vol. 11, pp. 271–278, December 1994.

[3] J. Gómez Ortega and E. F. Camacho, "Mobile Robot Navegation in a Partially Structured Static Enviroment, Using Neural Predictive Control", *Control Engineering Practice*, vol. 4, pp. 1669–1679, December 1996.

[4] J.E.Normey-Rico and E.F.Camacho, *A Smith predictor based generalised predictive controller*, University of Seville. Technical report GAR 1996/07, 1996.

[5] J.E. Normey-Rico, J. Gomez-Ortega, and E.F.Camacho, "A smith predictor based generalised predictive controller for mobile robot path-tracking", in *3rd Ifac Symposium on Intelligent Autonomous Vehicles*, pp. 471–476, Madrid, March 1998.

[6] C. Mothadi D.W. Clarke and P.S. Tuffs, "Generalized Predictive Control. Part I The Basic Algorithm and Part II Extensions and Interpretations", *Automatica*, vol. 23, pp. 137–160, 1987.

[7] E.F. Camacho and C. Bordons, *Model Predictive Control in Process Industry*, Springer Verlag, 1995.

[8] O.J.M Smith, *Feedback Control Systems*, Mc Graw Hill, 1958.

[9] O. Amidi, *Integrated Mobile Robot Control*, Carnegie Mellon University Robotic Institute (Pittsburgh). Technical Report CMU-RI-TR-90-17, 1990.