PERGAMON

Control Engineering Practice 7 (1999) 729–740

A Smith-predictor-based generalised predictive controller for mobile robot path-tracking

Julio E. Normey-Rico[a], Juan Gómez-Ortega[b,*], Eduardo F. Camacho[b]

[a]Departamento de Automação e Sistemas. Federal Univ. of Santa Catarina. Cx. P. 476, CEP 88040-900 Florianópolis-SC, Brazil
[b]Departamento de Ingeniería de Sistemas y Automática, Univ. de Sevilla, Escuela Superior de Ingenieros, Camino de los Descubrimientos s/n, 41092-Sevilla, Spain

Received 16 June 1998; accepted 15 February 1999

Abstract

This paper shows how to improve the robustness of mobile robot path tracking when predictive control algorithms are used. When uncertainties in a mobile robot are considered, it is shown how the classical generalised predictive controller (GPC) may lead to an unstable behaviour of the mobile robot. The equivalence between the GPC and a structure composed by an optimal predictor and a classical controller is given, and it is shown that use of a Smith predictor instead of an optimal predictor increases the robustness of the system. A new strategy, the Smith-predictor-based GPC, is proposed. Experimental tests carried out on a LABMATE mobile robot validate the performance of the proposed controller. © *1999 Elsevier Science Ltd. All rights reserved.*

Keywords: Mobile robots; Path tracking; Predictive controllers; Dead time

1. Introduction

The field of mobile robots has been the focus of a great deal of research effort in the recent years. One of the main lines of research is concerned with driving the vehicle as close as possible to a desired explicit path, with the least control effort. This is known as the path-tracking problem (PTP). "Explicit paths" must be understood as those paths that are geometrically defined by means of a set of consecutive points or a set of analytical functions (lines and circular arcs, splines) that describe geometric segments joined together. In this work, the PTP will not be considered as following any environmental feature such as a road edge (Pomerlau, 1990) or virtual paths defined by ceiling lights (Martínez et al., 1992), or even wall-following (Delaplace et al., 1992).

The consideration of the control effort as a magnitude to be reduced is a desired feature for a mobile robot control system, because this will increase its autonomy in terms of energy, which is one of the main constraints in real applications of this kind of robot.

* Corresponding author. Tel.: + 34-95-4487346; fax: + 34-95-4487340.
E-mail address: juango@cartuja.us.es (J. Gómez-Ortega)

The use of a path-tracking module as part of a real navigation system is justified not only when the work environment is well known and no unexpected obstacles are considered in the problem, but also when the environment is partially structured, and unexpected obstacles can be found during the navigation. In the first case, the reference path can be the result of an off-line path-planning problem, where all the obstacles are known and then taken into account in this path-planning stage. In the second case, many obstacle-avoidance strategies found in the literature use a decoupled scheme (Gómez-Ortega, 1994), where the reference path is replanned on-line when an unexpected obstacle is detected, and then either the original path or the replanned path is used as reference for the path-tracking module.

Most path-tracking methods are based finding the error between the current vehicle's current position and heading and the desired position and heading of a reference point located on the reference path to be followed. There are several approaches for the selection of this reference point. One approach chooses a reference point located on the desired path, at a predefined distance ahead of the actual robot position (Amidi, 1990). This *lookahead* (Ollero and Amidi, 1991) distance can be fixed, but can also be changed on-line, depending on several dynamic parameters such as robot velocity or reference

PII: S0967-0661(99)00025-8

path curvature (Ollero et al., 1994). In Egerstedt et al. (1998) the time evolution of the reference point is governed by a differential equation that contains the position error. Both the reference point and the associated differential equation are referred there as a "virtual car". Another approach is used with some optimal control techniques, where a set of consecutive reference points, all of them located on the reference path, are chosen instead of an isolated point (Gómez-Ortega, 1994; Gómez-Ortega and Camacho, 1996).

Different approaches can be found for the mobile robot control law. In Nelson and Cox (1990) a linear proportional feedback of the lateral and offset path-tracking errors is implemented. In Kanayama et al. (1990), a classical PID controller is used. Geometric approaches are also encountered in the literature. In this case, the control strategy consists of computing an analytical curve (for example a spline (Nelson, 1989) or a quintic polynomial (Shin and Singh, 1990)) that complies with several constraints (continuity in curvature, initial and final points, etc.). The structure of the curve is fixed, and its parameters are calculated at every sampling interval. In Amidi (1990), a strategy called *pure pursuit* is presented. The robot path curvature is calculated at each sampling time as the curvature of a circumference that passes through the robot's position and a *lookahead* point located on the reference path.

Optimal control techniques can also be used for path-tracking. In Ollero and Amidi (1991), generalised predictive control (GPC) is chosen as control strategy for the outdoor NAVLAB vehicle. A linear model of the robot's kinematics is considered for the prediction of its future positions and headings. A quadratic cost function, where the tracking error and the control effort are penalised, is minimised. Also, a predictive technique is encountered in Gómez-Ortega and Camacho (1994, 1996), but a nonlinear model of the robot, an indoor LABMATE differential-drive cart, is considered, and a multilayer perceptron is used to implement the predictive controller in real time. In Hemami et al. (1992), based on a dynamic model for the plane motion of a three-wheeled cart with a front steering wheel, an optimal controller is proposed where a performance criterion consisting of the squared errors in position and orientation is minimised. In (Freund and Mayr, 1997) and (Thuilot et al., 1996), a nonlinear controller is proposed, based on the feedback linearisation principle.

The problem that is raised in this paper is that of driving a differential-drive (Nelson, 1989) mobile robot to follow a previously calculated reference path, defined as a set of consecutive points. In this context, a predictive control technique seems very promising, for the following reasons:

- the reference path is previously known;
- smooth driving and an improvement of the autonomy in terms of energy can be achieved because the control effort can be penalised;
- extension to MIMO systems (which is the case of a mobile robot) is easy;
- dead-time compensation is intrinsic;
- constraints in the variables can be considered;
- predictive control fundamentals are used by people for car driving.

The timing of control commands is another issue of concern in path-tracking. Using an operating system like Unix, the procedure used is to poll a clock so as to ensure that the commands are sent at regular intervals. However, such operating systems do not guarantee a real-time response, and the interval between commands cannot be considered to be constant. Other sources for this uncertainty in the timing could include changes in the mobile robot's dynamic time constant, which can be influenced by payload changes, level of batteries, etc.

A common issue in predictive control techniques is their high dependence on a prediction model. The uncertainties in this model can lead to poor performance, or may make the system unstable. In this paper, a new GPC is proposed for mobile robot path-tracking. This controller uses a Smith predictor instead of an optimal predictor, increasing the robustness of the controller against uncertainties in the estimated time delay.

The paper is organised as follows. In Section 2, a brief description of the GPC is presented. The model considered for the mobile robot's kinematics, and the way in which the GPC is applied to the PTP, are shown in the same section. In Section 3 a classical representation of the GPC is shown. Section 4 is dedicated to presenting a robustness analysis, and Section 5 presents the proposed controller. The method has been tested on a LABMATE mobile robot, and some experimental results are presented in Section 6.

2. Generalised predictive control path-tracking

2.1. GPC strategy

The objective of GPC (Clarke et al., 1987; Camacho and Bordons, 1995) is to obtain a future control action sequence $(\mathbf{u}(t), \mathbf{u}(t+1|t), \ldots, \mathbf{u}(t+N_u-1|t))$ in such a way that the future predicted system outputs $\hat{\mathbf{y}}(t+j|t)$ will be as close as possible to the future desired references $\mathbf{y}_r(t+j)$ over the prediction horizon. This is accomplished by the minimisation of a cost function J with respect to the control variables, that has the following form:

$$J = \sum_{j=N_1}^{N_2} \|\hat{\mathbf{y}}(t+j|t) - \mathbf{y}_r(t+j)\|^2_{\mathbf{W}_1} + \sum_{j=1}^{N_u} \|\Delta\mathbf{u}(t+j-1)\|^2_{\mathbf{W}_2}, \tag{1}$$

where N_1 and N_2 are the minimum and maximum costing horizons, $\mathbf{W}_1$ and $\mathbf{W}_2$ are positive definite weighting

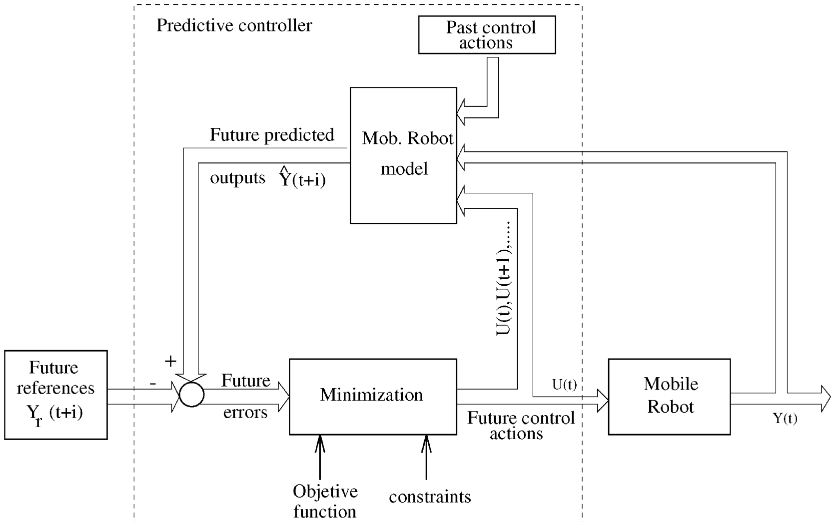

Fig. 1. The predictive controller scheme.

matrices, N_u is the control horizon, $\Delta\mathbf{u}(t)$ are the control increments and $\hat{\mathbf{y}}(t+j|t)$ is the optimum j-step-ahead prediction of the system output on data up to time t.

The future system outputs, $\hat{\mathbf{y}}(t+j|t)$ for $i = N_1, \ldots, N_2$, are predicted from a model of the process, from past inputs and outputs, and from the control actions foreseen for the future, $\mathbf{u}(t+j|t)$ for $j = 0, \ldots, N_u - 1$, 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.

After this sequence has been obtained, a *receding horizon* approach is considered. This consists of applying only the first calculated control action $\mathbf{u}(t)$. 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. A block diagram of the system is shown in Fig. 1.

2.2. Prediction model

For a GPC formulation, a linear model of the mobile platform is needed, to predict the future positions and headings of the robot. As a testbed, a differential-drive LABMATE mobile robot (Gómez-Ortega and Camacho, 1994) has been used. To obtain a linear model of the mobile robot the increments in the heading $\Delta\theta(t)$ are assumed to be small at each sample period. Also, to obtain a simple linearised model it is interesting to use a local reference frame (see Fig. 2). With these two assumptions, the linear model of the mobile robot is given by (see the appendix for more details):

$$(1 - z^{-1})\mathbf{I}\begin{bmatrix}\theta(t)\\ y(t)\end{bmatrix} = VT\begin{bmatrix}1\\ VT/2\end{bmatrix}\gamma\,(t-1-d), \qquad (2)$$

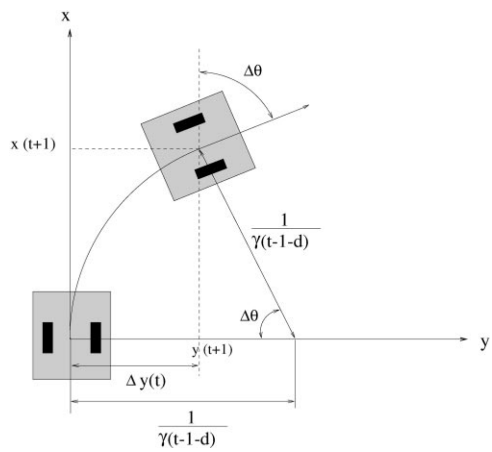

Fig. 2. The local reference model.

where $\mathbf{y}(t) = (\theta(t)\ y(t))^T$ and $\mathbf{u}(t) = \gamma(t)$. $\theta(t)$ is the robot heading in a global reference frame, $y(t)$ is the robot position in a reference frame attached to the vehicle, $\gamma(t)$ is the curvature, d is the dead time, V is the linear velocity, T is the sample period and $\mathbf{I}$ is a identy 2×2 matrix. In this work V is considered to be constant.

As has been mentioned, these linearised equations are valid only for small values of $\Delta\theta$. Thus, if the robot is not positioned on the reference path, the values for $\Delta\theta$ needed to reach the desired path will be too large, and the linearised model will not be valid (see Fig. 3). This fact implies that, in order to avoid this problem, approach routes have to be considered. In this way, the mobile robot is always located on the reference path, and the values of $\Delta\theta$ are small.

In this paper, the approaching paths have been computed using the *pure pursuit* strategy (although other strategies could be used). That is, the next N reference

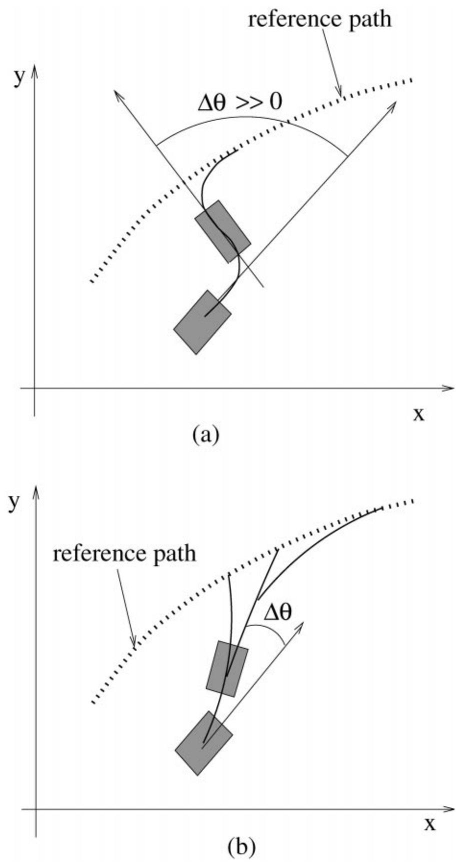

Fig. 3. Approach paths.

points are predicted as if the robot were being controlled with a pure pursuit scheme. At each sampling instant, a circumference is traced between the present robot position and a lookahead point, located at a predefined distance on the reference path. This circumference has to be tangent to the robot heading, and its curvature is chosen as a reference for the predictive controller.

The final model is a multivariable one-input/two-output system, where the mobile robot path curvature has been chosen as the control variable, and the robot's heading and local position are the controlled variables. Although the system should have a single input and a single output (note that $y(t) = (VT/2)\theta(t)$), it has been noticed that the system performance is much better when two outputs ($\theta(t)$ and $y(t)$) are considered. This is because the approach path depends on the robot heading. Thus, an error in $\theta(t)$ leads to badly fitted approach trajectories. On the other hand, if the error in $y(t)$ were not penalised, the robot would not be able to approximate the desired global path.

2.3. Disturbances model

In the GPC algorithm, a stochastic model of the disturbances is usually considered. An ARIMA (auto-regressive and integrated moving average) model of the disturbances is normally chosen (Clarke et al., 1987). In this case the model of the plant and disturbances is given by:

$$(1 - z^{-1})\mathbf{I}\begin{bmatrix}\theta(t)\\ y(t)\end{bmatrix} = \begin{bmatrix}VT\\ (VT)^2/2\end{bmatrix}\gamma(t-1-d) + \begin{bmatrix}1\\ 1\end{bmatrix}n(t), \quad (3)$$

where $n(t)$ can be computed as

$$n(t) = \frac{T(z^{-1})e(t)}{D(z^{-1})}, \quad (4)$$

where the polynomial $D(z^{-1})$ explicitly includes an integrator $(1 - z^{-1})$ and $e(t)$ is a zero mean white noise. The use of an integrator in D guarantees an offset-free system. The polynomials $T(z^{-1})$ and $D(z^{-1})$ could be used to adapt the model to the real characteristics of the disturbances, although in practice $T(z^{-1})$ is not related to the disturbances model, and the GPC is computed in two steps: first using $T(z^{-1}) = 1$ and then tuning $T(z^{-1})$ in order to increase the robustness of the GPC (Robinson and Clarke, 1991; Yoon and Clarke, 1995). The following analysis will be done using: $T(z^{-1}) = 1$ and $D(z^{-1}) = 1 - z^{-1}$. In this case the previous model can be represented by the matrix transfer function:

$$\begin{bmatrix}\theta(t)\\ y(t)\end{bmatrix} = \frac{VTz^{-\mathrm{d}}}{(1 - z^{-1})}\begin{bmatrix}1\\ \dfrac{(VT)}{2}\end{bmatrix}\gamma(t-1) + \frac{1}{(1 - z^{-1})^2}\begin{bmatrix}1\\ 1\end{bmatrix}e(t). \quad (5)$$

As can be seen, the SISO models relating the input γ to the heading θ and the position y have the same dynamics and a different static gain. In what follows $P = Gz^{-\mathrm{d}}$ will represent the transfer function between input γ and θ, where $G = (VTz^{-1})/(1 - z^{-1})$ and d is the dead time.

2.4. The GPC structure for the mobile robot

In this work the GPC parameters are chosen in the following manner: $N_u = N$, $N_1 = d + 1$ and $N_2 = d + N$ (this is an appropriate choice when the model of the plant has a dead time (Camacho and Bordons, 1995)). In order to compute the future control sequence $\mathbf{u}(t), \mathbf{u}(t+1), \ldots,$ it is necessary to compute the predictions $\hat{\mathbf{y}}(t+j|t)$, $j = d+1, \ldots, d+N$. For this case the system output $\mathbf{y}(t)$ is composed of the robot heading $\theta(t)$ and the local position $y(t)$. Using the prediction model of the mobile robot with disturbances, a Diophantine equation can be solved in order to obtain an optimal prediction of the output (Clarke et al., 1987). In this case, because of the decoupled structure of the model, two Diophantine equations are solved, one for each controlled variable $\theta(t)$ and $y(t)$, for j varying between $d+1$ and $d+N$:

$$\hat{\theta}(t+j|t) = 2\hat{\theta}(t+j-1|t) - \hat{\theta}(t+j-2|t) + VT\Delta\gamma(t+j-d-1), \quad (6)$$

$$\hat{y}(t+j|t) = 2\hat{y}(t+j-1|t) - \hat{y}(t+j-2|t) + \frac{(VT)^2}{2}\Delta\gamma(t+j-d-1) \qquad (7)$$

or in a vectorial form:

$$\begin{bmatrix} \hat{\theta}(t+d+1|t) \\ \vdots \\ \hat{\theta}(t+d+N|t) \\ \hat{y}(t+d+1|t) \\ \vdots \\ \hat{y}(t+d+N|t) \end{bmatrix} = \mathbf{H}\begin{bmatrix} \Delta\gamma(t) \\ \Delta\gamma(t+1) \\ \vdots \\ \Delta\gamma(t+N-1) \end{bmatrix} + \mathbf{S}\begin{bmatrix} \hat{\theta}(t+d|t) \\ \hat{\theta}(t+d-1|t) \\ \hat{y}(t+d|t) \\ \hat{y}(t+d-1|t) \end{bmatrix}, \qquad (8)$$

where $\mathbf{H}$ and $\mathbf{S}$ are constant matrices given by:

$$\mathbf{H} = \begin{bmatrix} VT\mathbf{H}_1 \\ \frac{(VT)^2}{2}\mathbf{H}_1 \end{bmatrix} \mathbf{H}_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 2 & 1 & \cdots & 0 \\ & \vdots & & \\ N & N-1 & \cdots & 1 \end{bmatrix}, \qquad (9)$$

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_1 \end{bmatrix} \mathbf{S}_1 = \begin{bmatrix} 2 & -1 \\ 3 & -2 \\ \vdots & \vdots \\ N+1 & -N \end{bmatrix}. \qquad (10)$$

Note that $\mathbf{S}$ and $\mathbf{H}$ are constant for a given N, V and T. Using this relation in the expression of J, the minimisation of the cost function gives a linear relation between the future control action and the predictions and references:

$$\mathbf{M}\begin{bmatrix} \Delta\gamma(t) \\ \Delta\gamma(t+1) \\ \vdots \\ \Delta\gamma(t+N-1) \end{bmatrix} = \mathbf{P}_1\begin{bmatrix} \hat{\theta}(t+d|t) \\ \hat{\theta}(t+d-1|t) \\ \hat{y}(t+d|t) \\ \hat{y}(t+d-1|t) \end{bmatrix} + \mathbf{P}_2\begin{bmatrix} \theta_r(t+d+1) \\ \theta_r(t+d+2) \\ \vdots \\ \theta_r(t+d+N) \\ y_r(t+d+1) \\ y_r(t+d+2) \\ \vdots \\ y_r(t+d+N) \end{bmatrix}, \qquad (11)$$

where $\theta_r(t)$ and $y_r(t)$ represent, respectively, the future references in the heading $\theta(t)$ and position error $y(t)$, $\mathbf{M} = \mathbf{H}^T\mathbf{W}_1\mathbf{H} + \mathbf{W}_2$, $\mathbf{P}_1 = -\mathbf{H}^T\mathbf{W}_2\mathbf{S}$ and $\mathbf{P}_2 = \mathbf{H}^T\mathbf{W}_2$. The weighting matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ are chosen in this case as

$$\mathbf{W}_1 = \begin{bmatrix} \mu_\theta\mathbf{I}_\mathbf{N} & 0 \\ 0 & \mu_y\mathbf{I}_\mathbf{N} \end{bmatrix}, \quad \mathbf{W}_2 = \lambda\mathbf{I}_{2\mathbf{N}}, \qquad (12)$$

where $\mathbf{I}_\mathbf{N}$ is an identity matrix of dimension N. This is the most common choice in practice, although other expressions for $\mathbf{W}_1$ and $\mathbf{W}_2$ could be used. Using a receding-horizon algorithm, only the value of $\Delta\gamma(t)$ is computed, so using the first row ($\mathbf{m}_1$) of the matrix $\mathbf{M}^{-1}$, the final control law is given by:

$$\Delta\gamma(t) = l_{11}\hat{\theta}(t+d|t) + l_{12}\hat{\theta}(t+d-1|t) + l_{21}\hat{y}(t+d|t) + l_{22}\hat{y}(t+d-1|t) + \sum_{i=1}^{N} f_i\theta_r(t+d+i) + \sum_{i=N+1}^{2N} f_i y_r(t+d+i), \qquad (13)$$

where

$$[l_{11}\, l_{12}\, l_{21}\, l_{22}] = \mathbf{m}_1\mathbf{P}_1 \qquad f_i = \sum_{j=1}^{N} m_{1j}\, p2_{ij}. \qquad (14)$$

(m_{1j} and $p2_{ij}$ are, respectively, the elements of $\mathbf{m}_1$ and $\mathbf{P}_2$).

A block diagram of this structure is shown in Fig. 4.

3. Classical representation of the GPC

In this section an equivalent classical representation of the GPC control law of Eq. (13) is presented. The equivalent control structure is based on an optimal predictor plus two PI controllers and two reference filters, as shown in Fig. 5. Also, the optimal predictor is represented as a dead-time compensator as shown in Fig. 6. This result will be used later to analyse the robustness of the controller, and to justify the use of the Smith predictor structure instead of the optimal one in the GPC.

To obtain the expressions of the PI controllers C_1 and C_2 and the filters F_1 and F_2, Eq. (13) is written as

$$(1-z^{-1})\gamma(t) = (l_{11} + l_{12}z^{-1})\hat{\theta}(t+d|t) + (l_{21} + l_{22}z^{-1})\hat{y}(t+d|t) + (f_1 z^{d+1} + f_2 z^{d+2} + \cdots + f_N z^{d+N})\theta_r(t) + (f_{N+1}z^{d+1} + f_{N+2}z^{d+2} + \cdots + f_{2N}z^{d+N})y_r(t). \qquad (15)$$

Thus,

$$\gamma(t) = \frac{(l_{11} + l_{12}z^{-1})}{(1-z^{-1})}\left[\hat{\theta}(t+d|t) + \frac{(f_1 z^{d+1} + f_2 z^{d+2} + \cdots + f_N z^{d+N})}{(l_{11} + l_{12}z^{-1})}\theta_r(t)\right]$$

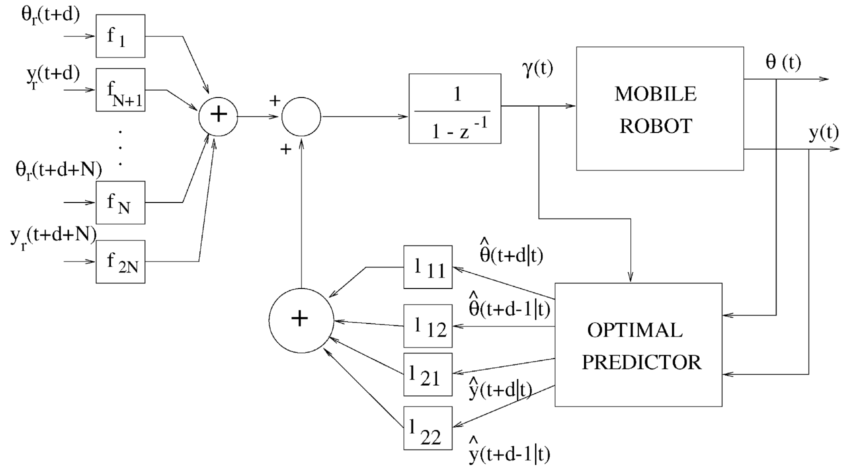

Fig. 4. Control structure of the GPC controller for the mobile root.

$$+\frac{(l_{21}+l_{22}z^{-1})}{(1-z^{-1})}\left[\hat{y}(t+d|t) + \frac{(f_{N+1}z^{d+1}+f_{N+2}z^{d+2}+\cdots+f_{2N}z^{d+N})}{(l_{21}+l_{22}z^{-1})}y_r(t)\right], \tag{16}$$

giving:

$$C_i(z)=\frac{(l_{i1}+l_{i2}z^{-1})}{(1-z^{-1})},\quad i=1,2, \tag{17}$$

$$F_1(z)=\frac{f_1z^{d+1}+f_2z^{d+2}+\cdots+f_Nz^{d+N}}{l_{11}+l_{12}z^{-1}}, \tag{18}$$

$$F_2(z)=\frac{f_{N+1}z^{d+1}+f_{N+2}z^{d+2}+\cdots+f_{2N}z^{d+N}}{l_{21}+l_{22}z^{-1}}. \tag{19}$$

Note that:

- The predictions of θ and y are computed independently (see Eqs. (7) and (6)) and so the final structure is a simple multivariable scheme where the control action is composed by the addition of a pair of monovariable classical controllers.
- The tuning of the parameters of the PI controllers and the reference filters could be done using other control methodologies (for example, the PI controllers can be tuned using a root-locus diagram), although the GPC strategy guarantees an optimal choice according to the defined cost function that takes account of both the error in the heading and in the local position.

To obtain the final control diagram of Fig. 6, the optimal prediction for the robot heading and position at time $t+j$ will be computed directly from the actual values of $\gamma(t)$, $\theta(t)$ and $y(t)$ (Normey-Rico et al., 1998), instead of the recursive expressions of the predictions normally used in the GPC (Camacho and Bordons, 1995). Thus, the relation between the prediction and the input and output of the plant will be computed directly from $G(z)$ and the dead time d of the plant.

Using Eq. (7), the prediction of θ in $t+1$ is

$$\hat{\theta}(t+1|t)=(2-z^{-1})\,\theta(t)+VT(1-z^{-1})\gamma(t-d). \tag{20}$$

Using this relation recursively to compute $\hat{\theta}(t+i|t)$ for $i=2\ldots d$ it follows that

$$\hat{\theta}(t+d|t)=(2-z^{-1})^d\theta(t) + \sum_{i=1}^{d}(2z^{-1}-z^{-2})^{i-1}VT(1-z^{-1})\gamma(t-1). \tag{21}$$

Now, note that:

$$\begin{aligned}&\sum_{i=1}^{d}(2z^{-1}-z^{-2})^{i-1}VT(1-z^{-1})\gamma(t-1)\\ &\quad=VT(1-z^{-1})z^{-1}\gamma(t)\sum_{i=1}^{d}(2z^{-1}-z^{-2})^{i-1}\\ &\quad=VT(1-z^{-1})z^{-1}\gamma(t)\frac{1-(2z^{-1}-z^{-2})^d}{(1-z^{-1})^2}\\ &\quad=G_n(z^{-1})(1-(2z^{-1}-z^{-2})^d)\gamma(t).\end{aligned} \tag{22}$$

Using the last expression and defining:

$$R(z^{-1})=((2z^{-1}-z^{-2})z)^d=(2-z^{-1})^d. \tag{23}$$

It follows that

$$\hat{\theta}(t+d|t)=R(z)[\theta(t)-P_n(z)\gamma(t)]+G_n\gamma(t). \tag{24}$$

Using the same procedure for the optimal prediction of y, it is possible to obtain:

$$\hat{y}(t+d|t)=R(z)\left[y(t)-\frac{VT}{2}P_n(z)\gamma(t)\right]+\frac{VT}{2}G_n\gamma(t), \tag{25}$$

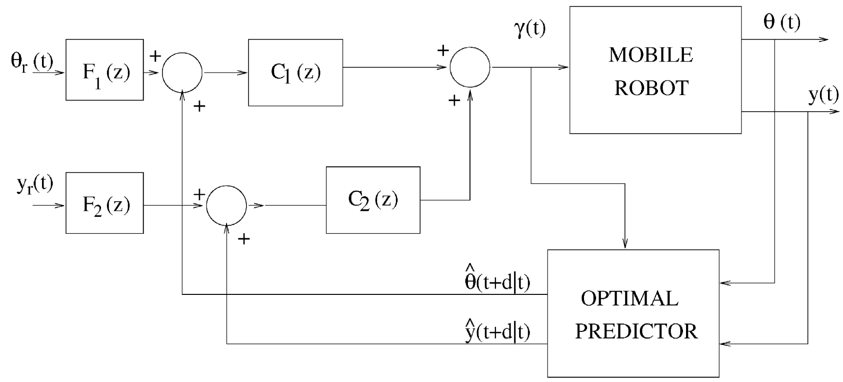

Fig. 5. Equivalent control structure of the GPC controller for the mobile root.

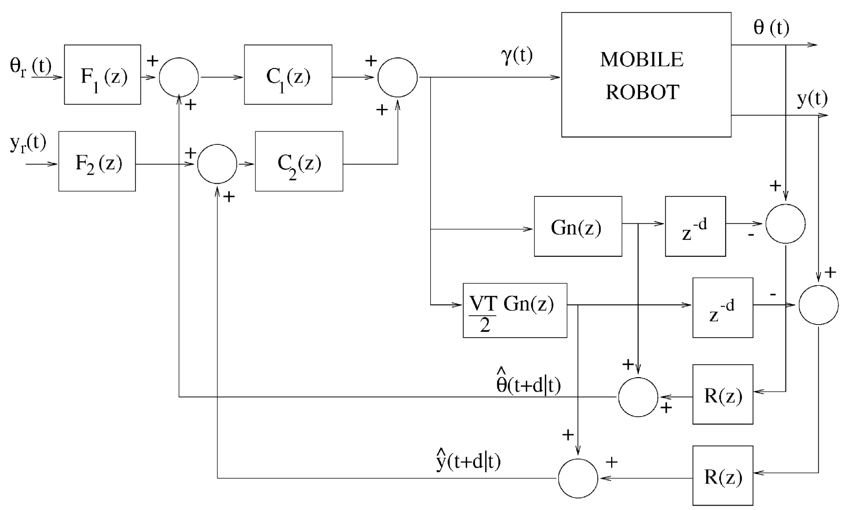

Fig. 6. Equivalent structure for robustness analysis.

showing that the final block diagram representation of the optimal predictor and the controller can be drawn as in Fig. 6.

From the block diagram of Fig. 6 it is easy to see that the GPC can be considered as a dead-time compensator, similar to the Smith predictor (Smith, 1957), in which the prediction at time $t + d$ is computed by a model of the plant without the dead time (G_n), and in order to correct any mismatch between the plant and model, the error between the output of the process and the model including the dead time (P_n) is fed back. In the GPC case this error is filtered with the block R while in the original work of Smith no filter is used in the feedback; this is equivalent to setting $R = 1$.

However, it must be noted that in some modified versions of the Smith predictor, filters are used in the feedback in order to improve the robustness or the disturbance rejection of the controller (Palmor, 1996; Normey-Rico et al., 1997). In the structure of Fig. 6, the blocks C_1, C_2, F_1 and F_2 represent a "two-degree-of-freedom primary controller" that is not affected by the value of R. Thus, the same linear control law given by the GPC algorithm could be used with another predictor structure like the one proposed by Smith (1957).

The results of this analysis will be used to study the influence of the predictor on the robustness of the control structure.

4. Robustness analysis

In this section, a robust analysis of the GPC is carried out using the block diagram of Fig. 6. The analysis is made independently for each controlled variable, because of the decoupled structured plant model.

The matrix transfer function representation will be used, and unstructured uncertainties will be considered. It will be assumed from now on that the behaviour of the process is described by a family of linear models. Note that in this case the two elements of the matrix have the same poles and zeros and only a different gain, so the analysis is made only for P defined in Section 2. The real plant $P = Gz^{-d}$ will be in the vicinity of the nominal plant $P_n = G_n z^{-d_n}$, that is: $P = P_n + \delta P$.

To analyse the effect of the predictor structure on the robustness of GPC, the characteristic Eq. ($Q(z)$) of the closed loop is computed, giving:

$$Q(z) = 1 + C_1(G_n + R\delta P) + (VT/2)C_2(G_n + R\delta P) = 0 \tag{26}$$

and so the uncertainty norm-bound is defined by the following expression (Morari and Zafiriou, 1989), valid for $\omega \in [0,\pi]$;

$$|\delta P(e^{j\omega})| < DP_{\text{gpc}}(\omega) = \frac{|1 + C(e^{j\omega})G_n(e^{j\omega})|}{|C(e^{j\omega})R(e^{j\omega})|}, \tag{27}$$

where $C = C_1 + (VT/2)C_2$. Note that the expression of the uncertainty norm-bound is also valid for the Smith predictor control structure using $R = 1$. That is, R can be used as a measurement index of robustness when considering the same nominal performance (when the reference filters and PI controllers are the same in both structures). Note that if the model is perfect and no disturbances are considered, R has no effect on the closed loop.

On the other hand, it is easy to show that $|R(e^{j\omega})| \geqslant 1$ for all frequencies in the GPC. In fact, the modulus of the vector $2 - z^{-1}$ varies between 1 and 3 when ω varies between 0 and π (see Fig. 7). Note that $|R(e^{j\omega})|$ will increase with the dead time, and this implies that the robustness of the GPC will be worst for plants with high dead time. In Fig. 7, $|R(e^{j\omega})|$ is plotted for $d = 1$ (using a solid line) and $d = 2$ (dashed line) in order to show the shapes of the curves and the increase in $|R(e^{j\omega})|$ with d.

As R appears in the denominator of the expression of DP_{gpc}, the uncertainty norm-bound of the GPC will be

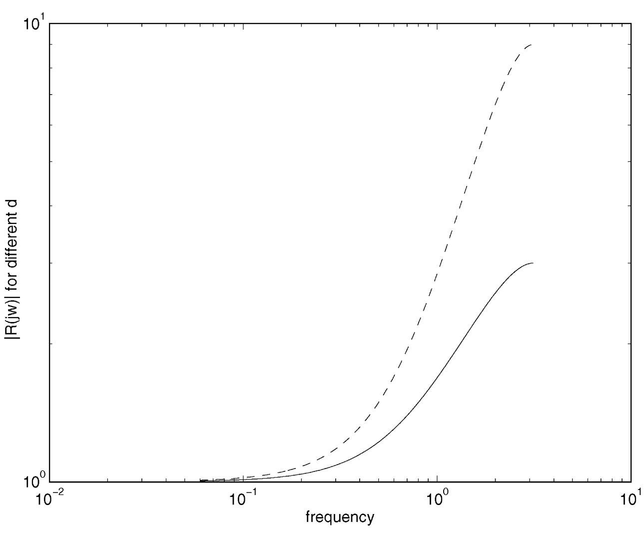

Fig. 7. $|R(jw)|$ for $d = 1$ (solid) and $d = 2$ (dashed).

lower than that of the Smith predictor, especially at high frequencies. Because of this characteristic, only small plant uncertainties will be acceptable at high frequencies in the GPC.

It is known that the robustness of the GPC can be improved by the use of filters. If the polynomial $T(z^{-1})$ of the model of the noise given in Section 2 is included in the predictions, then $R(z^{-1})$ and $C(z^{-1})$ are $T(z^{-1})$-dependent. Thus, if $T(z^{-1})$ is used as a design parameter, it is possible to change the high-pass characteristics of $R(z^{-1})$. However, the design of these filters is not straightforward, especially for MIMO plants with a dead time, as is the case here (Yoon and Clarke, 1995; Ansay and Wertz, 1997). In this paper a different method for the robustification of the controller is presented. This new approach is based in a modification of the predictor structure. It is simple, and the controller has the same tuning parameters as the original GPC. This is not the case when filters are included in the design of the GPC.

5. The proposed SPGPC for the mobile robot

In order to improve robustness, a new algorithm, the Smith-predictor-based generalised predictive controller (SPGPC) is proposed. This new control strategy is based on the Smith predictor structure (Smith, 1957), and computes the primary controller using the GPC approach.

For the computation of the controller parameters, the following procedure must be executed:

- Compute the open-loop prediction $\hat{\theta}_0(t + d - i|t)$, $\hat{y}_0(t + d - i|t)$ for $i = 0, 1$ using the local open-loop model of the mobile robot without considering disturbances.
- Compute the mismatch between the output and the open-loop prediction for $i = 0$, 1:

  $$e_\theta(t - i) = \theta(t - i) - \hat{\theta}_0(t - i),$$

  $$e_y(t - i) = y(t - i) - \hat{y}_0(t - i).$$

- Add the error to the open-loop prediction for $i = 0$, 1 to compute the final value of the predictions:

  $$\hat{\theta}(t + d - i|t) = \hat{\theta}_0(t + d - i|t) + e_\theta(t - i),$$

  $$\hat{y}(t + d - i|t) = \hat{y}_0(t + d - i|t) + e_y(t - i).$$

- Compute the coefficients l_{ij} and f_i using the same procedure as in the normal GPC, as explained in Section 2.
- Compute the control law ($\Delta\gamma$) using the same control equation as in the GPC (Eq. (13)).

Remarks: (i) The blocks C_1 and C_2 are the same in both the GPC and the SPGPC because both controllers use the same tuning parameters. Notice that, in the absence of model uncertainties or perturbations, both the

GPC and the SPGPC give the same nominal performance, independently of the selection of the controller parameters (look-ahead, N, λ, μ_θ and μ_y).

(ii) As no filter is used in the error between the prediction and the plant output ($R = 1$), the uncertainty norm-bound of this new controller is defined by:

$$DP_{spgpc}(\omega) = \frac{|1 + C(e^{j\omega})G_n(e^{j\omega})|}{|C(e^{j\omega})|}, \tag{28}$$

(valid for $\forall\omega \in [0, \pi]$) showing that the robust stability bound is the same as if the system had no dead time in the transfer function (Morari and Zafiriou, 1989) (note that d does not appear in the expression of DP_{spgpc}) and that the SPGPC has better robustness than the GPC:

$$DP_{spgpc}(\omega) \geqslant DP_{gpc}(\omega) \quad \forall\omega \in [0, \pi]. \tag{29}$$

In order to illustrate the improvement in robustness of the SPGPC, a simulation example is presented. The reference path is a straight-line trajectory (dashed line in Fig. 8), and the kinematic behaviour of the mobile robot is computed using the complete nonlinear model of the differential-drive mobile robot, given in Gómez-Ortega (1994); Gómez-Ortega and Camacho (1996).

The control is computed using a normal GPC, and also a SPGPC with the same controller parameters: $\lambda = 0.5$, $\mu_\theta = 0.5$, $\mu_y = 1$ and $N = 15$. In both cases the sampling time is $T = 0.2$ s, the look-ahead parameter is fixed at 0.6 m, and the linear velocity $V = 0.1$ m/s. As was stated in a previous section, dead-time uncertainties must be considered in the model, due to the different values of the acceleration and the different values of the robot payload. Also, in this example an error in the real linear velocity V is considered.

First the modulus of the multiplicative modelling error ($|\delta P/P_n|$) and the multiplicative uncertainty norm-bound ($DP/|P_n|$) of the GPC and SPGPC are shown in Fig. 9. For this example the modelling error (dotted line) is generated, using a 10% uncertainty in the gain of the plant and an error in the delay of two samples. Note that for this case the GPC is not robust (dashed line) and the SPGPC is robust (solid line). To analyse the dynamic behaviour the nominal ($d_n = 2$) performance of both controllers is compared in Fig. 8. As can be seen, as expected, the SPGPC (solid line) and the normal GPC (dashed line) have similar responses. In Fig. 10 the performance of the two controllers is compared for the real case. Note that, confirming the results of Fig. 9, the GPC (dashed line) cannot follow the reference (dotted line) and that the SPGPC (solid line) has the same performance as in the nominal case. Note the instability in the GPC case. As

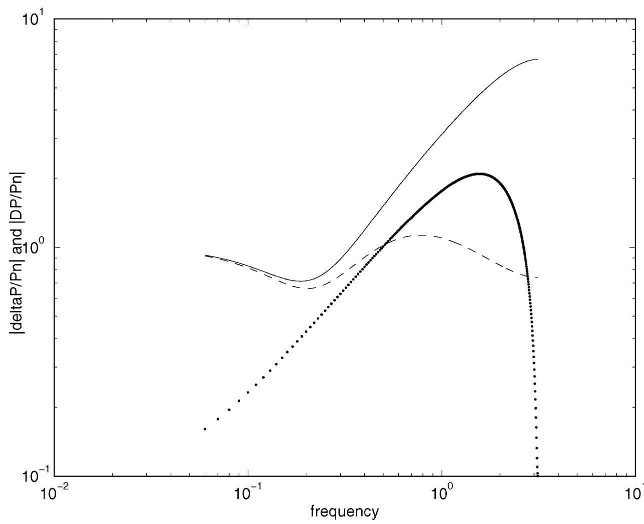

Fig. 9. $|\delta P/P_n|$ (dotted) and $DP/|P_n|$ for the GPC (dashed) and SPGPC (solid).

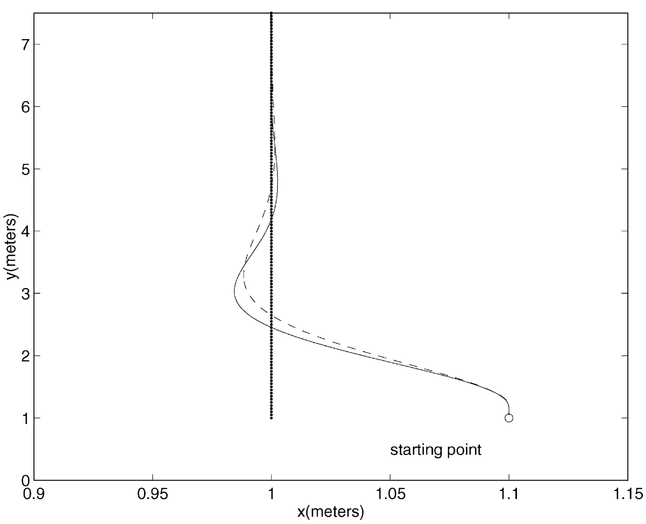

Fig. 8. Mobile robot path-tracking for the SPGPC (solid line) and GPC (dashed line); nominal case. Path reference in dotted line.

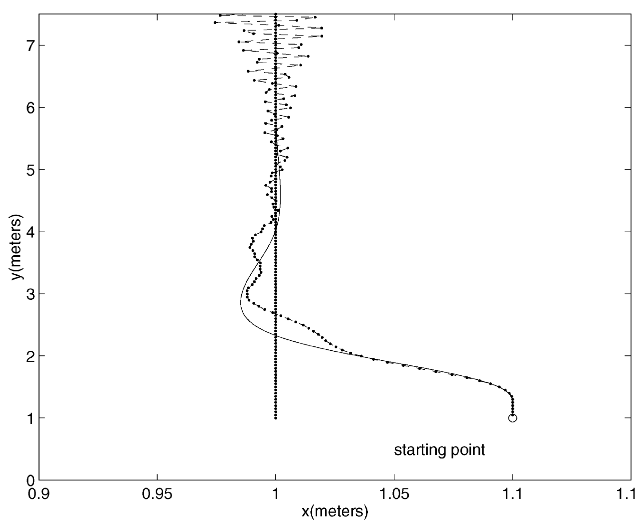

Fig. 10. Mobile robot path-tracking for the SPGPC (solid line), GPC (dashed line); dead-time uncertainties case. Path reference in dotted line.

was mentioned above, the GPC time response can be stabilised by the use of filters in the controller, although the tuning of this filter is not simple and the resulting controller is more complex. The SPGPC can cope with the model uncertainties, maintaining the original simple structure and tuning of the controller.

6. Experimental results

The proposed SPGPC was implemented in a LABMATE mobile root, shown in Fig. 11. The mobile robot configuration (position and heading) has been computed at each sampling time using an odometry algorithm, based on internal sensors (incremental encoders) of the LABMATE mobile robot. This is not an error-free approach, and the robot's configuration estimation errors are accumulative. Thus, for long paths, the odometry system should be updated with data from an external sensor system (video camera, range laser, etc.), at a predefined frequency. This issue is separate from the control problem, and has not been adressed in the paper.

Two experimental tests were performed. Fig. 12 shows the first experimental test, where the reference path is drawn by a dashed line. The controller parameters have been chosen as follows: $\lambda = 0.5$, $\mu_\theta = 0.5$, $\mu_y = 1$, $N = 10$, $T = 0.2$ s, look-ahead $= 0.6$ m and $V = 0.1$ m/s. In this case, it is assumed that the error in the dead time is 0.4 s (two sampling periods). Fig. 12 shows the reasonably good performance of the proposed controller, in spite of the error in the dead-time estimation. Note also that the reference path chosen has a small radius of curvature, which makes the path reference following more difficult.

In the second experiment, the control parameters are the same as in test 1, but a higher error in the delay is considered (three samples $= 0.6$ s). Fig. 13 shows the good performance of the controller, in spite of the error in the dead time being greater than 100%.

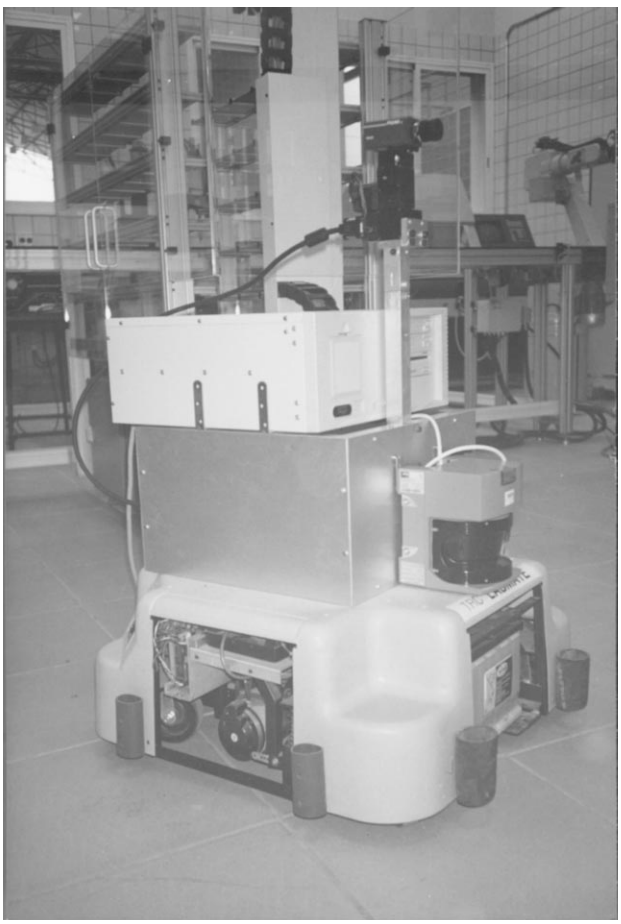

Fig. 11. LABMATE mobile root.

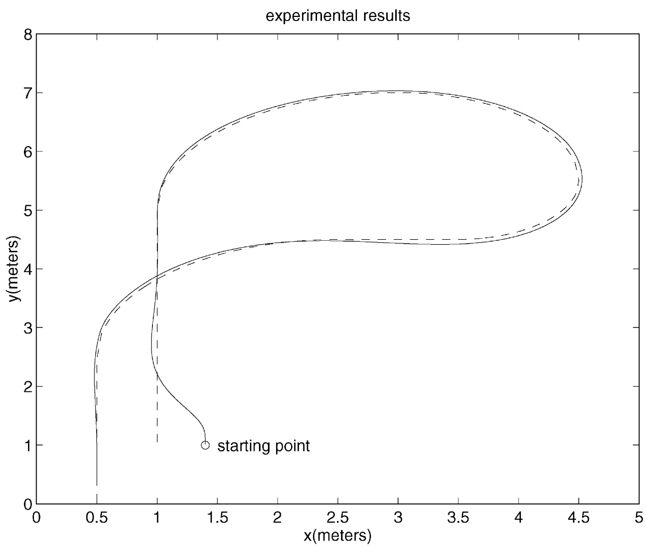

Fig. 12. Mobile robot path-tracking for the SPGPC. Path reference in dashed line.

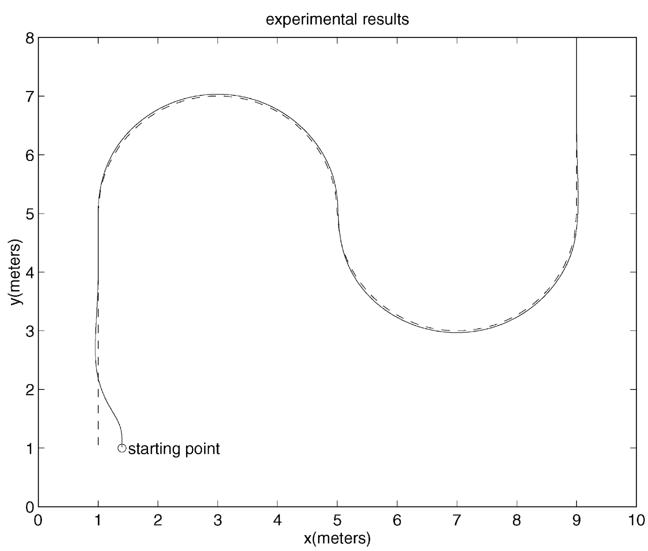

Fig. 13. Mobile robot path-tracking for the SPGPC. Path reference in dashed line.

7. Conclusions

A Smith-predictor-based GPC has been proposed for a mobile robot path-tracking problem. A robustness analysis has been carried out, showing the more-robust performance of this new controller. Some simulation and experimental results have shown that classical GPC cannot be used for this problem in the presence of dead-time uncertainties, because the mobile robot became unstable. The new control strategy gives similar nominal performance to that of the normal GPC, and also robust performance for this case.

Acknowledgements

The authors would like to acknowledge the helpful comments of the anonymous reviewers; also, the *CICYT* under TAP96-884 and TAP98-0541, and the *CAPES* under Contract BEXO448/95-6 for funding this work.

Appendix A

A.1. Linearised kinematic model for a differential-drive cart

In this appendix, the linearised model for the differential-drive cart used in the experiments is obtained. This model is based on Fig. 2, where a local reference frame is attached to the mobile robot.

When the maximum attainable acceleration for each wheel angular velocity is selected (1 m/seg^2 for the LABMATE), a simplified model for the actuator dynamics can be assumed (Gómez-Ortega, 1994), where the mobile robot angular velocity $\dot{\theta}$ (and thus also the robot curvature γ) can be considered constant during the sampling period T. Then, from the curvature definition, and considering that the robot linear velocity V is constant over time and there is a time delay of d seconds in the control:

$$\Delta\theta(t) = VT_{\gamma}(t-1-d).$$

Also from Fig. 2 it can be seen that:

$$\Delta y(t) = \frac{(1-\cos\Delta\theta)}{\gamma(t-1-d)}.$$

Using now the approximate expansion of the $\cos\Delta\theta$ into a Taylor series

$$\cos\Delta\theta = 1 - \frac{\Delta^2}{2!}$$

the following expression for $\Delta y(t)$ is obtained:

$$\Delta y(t) = \frac{(VT)^2}{2}\gamma(t-1-d).$$

Finally, the discrete version of this model can be written in the following matrix form:

$$\begin{bmatrix}\theta(t)\\ y(t)\end{bmatrix} = \frac{VT}{(1-z^{-1})}\begin{bmatrix}1\\ VT/2\end{bmatrix}\gamma\,(t-1-d).$$

References

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

Ansay, P., & Wertz, V. (1997). Model uncertainties in gpc: a systematic two-step design. Proceedings of the ECC 97, Brussels.

Camacho, E. F., & Bordons, C. (1995). *Model predictive control in process industry*. London: Springer.

Clarke, D. W., Mothadi, C., & Tuffs, P. S. (1987). Generalized predictive control. Part I the basic algorithm and Part II extensions and interpretations. *Automatica*, *23*(2), 137–160.

Delaplace, S., Blazevic, P., Fontaine, J. G., Pons, N., & Rabit, J. (1992). *Trajectory tracking for mobile robot*. Paris: Laboratoire de Robotique de Paris. Internal report.

Egerstedt, M., Hu, X., & Stotsky, A. (1998). Control of car-like robot using a dynamic model. *Proceedings of the 1998 IEEE-Int. Conference on Robotics and Automation* (pp. 3273–3278). Belgium: Leuven.

Freund, E., & Mayr, R. (1997). Nonlinear path control in automated vehicle guidance. *IEEE Transactions on Robotics and Automation*, *13*(1), 49–60.

Gómez-Ortega, J. (1994). Navegación en robots móviles basada en técnicas de control predicitvo neuronal. PhD thesis. Escuela Sup. de Ingenieros, Univ. de Sevilla

Gómez-Ortega, J., & Camacho, E. F. (1994). Neural network MBPC for mobile robots path tracking. *Robotics and Computer Integrated Manufacturing Journal*, *11*(4), 271–278.

Gómez-Ortega, J., & Camacho, E. F. (1996). Mobile robot navigation in a partially structured static environment, using neural predictive control. *Control Engineering Practice*, *4*(12), 1669–1679.

Hemami, A., Mehrabi, M. G., & Cheng, R. M. H. (1992). Synthesis of an optimal control law for path tracking in mobile robots. *Automatica*, *28*(2), 383–387.

Kanayama, Y., Kimura, Y., Noguchi, T., & Miyakazi, F. (1990). A stable tracking control method for an autonomous mobile robot. Proceedings of the IEEE International Conference on Robotics and Automation (pp. 384–389).

Martínez, A. B., Climent, J., Asensio, J. M., & Batlle, J. (1992). Absolute positioning for indoor mobile robots guidance. Proceedings of the International Symposium on Industrial Robots. (pp. 529–532) Barcelona.

Morari, M., & Zafiriou, E. (1989). *Robust process control*. Englewood Cliffs, NJ: Prentice-Hall.

Nelson, W. L. (1989). Continous steering-function control of robot carts. *IEEE Transactions on Industrial Electronics*, *36*(3), 330–337.

Nelson, W., & Cox, I., (1990). Autonomous robot vehicles (pp. 38–44). Chap. local path control for an autonomous vehicle, London: Springer.

Normey-Rico, J. E., Camacho, E. F., & Gómez-Ortega, J. (1998). Robustness and predictions in generalised predictive controller. Proceedings of the XII Brazilian Control Conference (CBA 98) (pp. 157–162) Uberlandia, Brazil.

Normey-Rico, J. E., Bordons, C., & Camacho, E. F. (1997). Improving the robustness of dead-time compensating PI controllers. *Control Engineering Practice*, *5*(6), 801–810.

Ollero, A., & Amidi, O. (1991). Predictive path tracking of mobile robots. Applications to the CMU Navlab. Proceedings of the IEEE Fifth International Conference on Advanced Robotics. (pp. 1081–1086) Pisa.

Ollero, A., García-Cerezo, A., & Martínez, J. L. (1994). Fuzzy supervisory path tracking of autonomous vehicles. *Control Engineering Practice*, *2*(2), 313–319.

Palmor, Z. J. (1996). *The Control Handbook Chap. No. 10.8–10.10. Smith Predictors and Modifications*. Florida: CRC Press and IEEE Press.

Pomerlau, D.A. (1990). Vision and Navigation. The Carnegie Mellon Navlab (Chap. 5, pp. 83–93). Neural Network Based Autonomous Navigation, Boston: Kluwer Academic Publishers.

Robinson, T., & Clarke, D. (1991). Robustness effects of a prefilter in receding-horizon predictive control. *IEE-D*, *138*, 2–8.

Shin, D.H., & Singh, S. (1990). *Vision and Navigation. The Carnegie Mellon Navlab*. (Chap. 13, pp. 283–307) Vehicle and Path Models for Autonomous Navigation. Boston: Kluwer Academic Publishers.

Smith, O. J. M. (1957). Closed control of loops with dead time. *Chemical Engineering Progress*, *53*, 217.

Thuilot, B., d'Andrea Novel, B., & Micaelli, A. (1996). Modeling and feedback control mobile robots equipped with several steering wheels. *IEEE Transactions on Robotics and Automation*, *12*(3), 375–390.

Yoon, T., & Clarke, D. (1995). Observer design in receding-horizon control. *International Journal of Control*, *2*, 171–191.