A Generalized Predictive Controller for a Wide Class of Industrial Processes

Carlos Bordons and Eduardo F. Camacho, Member, IEEE

Abstract—This paper presents a formulation of generalized predictive control (GPC), easy to implement and tune, that is valid for the majority of industrial processes. The method makes use of the fact that a generalized predictive controller results in a control law that can be described with few parameters. The controller has been developed for a wide class of processes in industry and a set of simple functions relating the controller parameters to the process parameters has been obtained. With this set of functions either a fixed or a selftuning GPC can be implemented in a straightforward manner. The influence of modeling errors is analyzed, and the results obtained show the stability robustness of the method, especially with respect to uncertainties in time constant and static gain.

Index Terms—Adaptive control, predictive control, process control, robustness, uncertain systems.

I. INTRODUCTION

MODEL predictive control (MPC) has developed considerably in the last few years, within the control research community and industry. This can be attributed to the fact that MPC is perhaps the most general way of posing the process control problem in the time domain. MPC formulation integrates optimal control, control of processes with dead times, multivariable control, and the use of future references when available. It can also handle constraints and nonlinear processes, which are frequently found in industry.

An MPC results in a linear control law which is easy to implement once the controller parameters are known. The derivation of the MPC parameters requires, however, some mathematical complexities. Although this is not a problem for people in the research control community where mathematical packages are normally available, it may be discouraging for those practitioners used to much simpler ways of implementing and tuning controllers.

The previously mentioned computation has to be carried out only once when dealing with processes with fixed parameters, but if the process parameters change, the controller parameters have to be derived again, perhaps in real time, at every sampling time if a selftuning control is used. This again may be a difficulty because on one hand, some distributed control equipment has only limited mathematical computation capabilities for the controllers, and on the other hand, the computation time required for the derivation of the MPC parameters may be excessive for the sampling time required by the process and the number of loops implemented.

One of the reasons for the success of the traditional PID controllers in industry is that they are very easy to implement and tune by using heuristic tuning rules such as the Ziegler–Nichols rules frequently used in practice. A fast implementation method of a GPC, a reasonable representative of the family of MPC, has been proposed in [1]; this method is valid for processes that can be modeled by the reaction curve method (most plants in the process industry), with a deadtime multiple of the sampling time. This paper describes how these results can be extended to processes having a noninteger dead time and those including integral effect. It will be shown that the controller can be implemented with a limited set of instructions, available in most distributed control systems, and that the computation time required, even for tuning, is very short. The method to implement the GPC is based on the fact that a wide range of processes in industry can be described by a few parameters and that a set of simple Ziegler–Nichols type of functions relating controller parameters to process parameters can be obtained. By using these functions the implementation and tuning of a GPC results almost as simple as the implementation and tuning of a proportional integral derivative (PID).

This paper is organized as follows: first a short review of GPC is given in Section II. Section III shows the derivation of the control law and the way to implement the method. The extension to integrating processes and ramp setpoints is dealt with in Section IV. Section V is dedicated to perform a robustness analysis of the method when modeling errors are taken into account and illustrative applications to temperature loops is presented in Section VI. The paper ends with some concluding remarks.

II. GENERALIZED PREDICTIVE CONTROL

Model (based) predictive control (MBPC or MPC), is not a specific control strategy but more of a very ample range of control methods developed around certain common ideas. The ideas appearing in all the predictive control family are the explicit use of a model to predict the process output at future time instants (horizon), the calculation of a control sequence minimizing a certain objective function and the use of a receding strategy, so that at each instant the horizon is displaced toward the future, which involves the application of the first control signal of the sequence calculated at each step. There are many applications of predictive control successfully in use at the present time, not only in the process industry.

Manuscript received August 8, 1995; revised March 11, 1997 and July 14, 1997. Recommended by Associate Editor, J. M. Maciejowski. This work was supported in part by CICYT Contract TAP 95-0370.

The authors are with the Departamento de Ingeniería de Sistemas y Automática, University of Seville, 41012 Sevilla, Spain.

Publisher Item Identifier S 1063-6536(98)03091-7.
but also applications to the control of a diversity of processes [2]–[5]. MPC is particularly attractive to staff with only a limited knowledge of control, because the concepts are very intuitive, and it can be used to control a great variety of processes, from those with relatively simple dynamics to other more complex ones.

The GPC method proposed by Clarke et al. [6] is a reasonable representative of this family and has become one of the most popular MPC methods both in industry and academia, being successfully implemented in many industrial applications [7]. The basic idea of GPC is to calculate a sequence of future control signals in such a way that it minimizes a multistage cost function defined over a prediction horizon. The index to be optimized is the expectation of a quadratic function measuring the distance between the predicted system output and some predicted reference sequence over the horizon plus a quadratic function measuring the control effort.

GPC provides an explicit solution (in the absence of constraints), it can deal with unstable and nonminimum phase plants and incorporates the concept of control horizon as well as the consideration of weighting of control increments in the cost function. The general set of choices available for GPC leads to a greater variety of control objectives compared to other approaches, some of which can be considered as subsets or limiting cases of GPC.

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

$$J = E \left\{ \sum_{j=1}^{N_2} \left[ y(t+j|t) - w(t+j) \right]^2 + \sum_{j=1}^{N_1} \lambda(j) \left| \Delta u(t+j-1) \right|^2 \right\}$$

where $E\{\}$ is the expectation operator and $y(t+j|t)$ is an optimum $j$-step ahead prediction of the system output on data up to time $t$. $N_1$ and $N_2$ are the minimum and maximum costing horizons, $N_u$ is the control horizon, $\lambda(j)$ is the control-weighing sequence, and $w(t+j)$ is the future reference trajectory.

The objective of predictive control is to compute the future control sequence $u(t), u(t+1), \ldots$ in such a way that the future plant output $y(t+j)$ is driven close to $w(t+j).$ This is accomplished by minimizing $J.$

The standard algorithm [6] involves the optimal prediction of $y(t+j)$ for $N_1 \leq j \leq N_2$, which is obtained by the recursion of a Diophantine equation and the triangularization of an $N_u \times N_u$ matrix.

III. PRECOMPUTED GPC

Most processes in industry when considering small changes around an operating point can be described by a linear model of, normally, very high order. The reason for this is that most industrial processes are composed of many dynamic elements, usually first order, so the full model is of an order equal to the number of elements. These very high-order models are not suitable for control purposes but it is possible to approximate the behavior of such high-order processes with a simplified model consisting of a first-order process combined with a dead-time element [8]. This type of system can then be described by the following transfer function:

$$G(s) = \frac{K}{1 + \tau s} e^{-\tau_d s}$$

where $K$ is the process static gain, $\tau$ is the time constant or process lag, and $\tau_d$ is the dead time or delay. This model is widely used in industry to describe the dynamics of many processes, as shown by the popularity of the reaction curve method and the open-loop Ziegler–Nichols PID tuning rules. Obviously better approximations could be obtained by using higher order models, but this would require identification packages which are not normally available in industry.

When the dead time $\tau_d$ is an integer multiple of the sampling time $T$ ($\tau_d = dT$), the corresponding discrete transfer function of (2) has the form

$$G(z^{-1}) = \frac{b z^{-1}}{1 - az^{-1}} z^{-d}$$

where discrete parameters $a$, $b$, and $d$ can easily be derived from the continuous parameters by discretization of the continuous transfer function, resulting in the following expressions:

$$a = e^{-(T/\tau)}, \quad b = K(1-a), \quad d = \frac{\tau_d}{T}.$$
function of \( \hat{y}(t + d|t), \hat{y}(t + d - 1|t), \Delta u(t + N_2 - d - 1), \Delta u(t + N_2 - d - 2) \cdots \Delta u(t) \) and the reference sequence. Minimizing \( J(N) \) with respect to \( \Delta u(t), \Delta u(t + 1) \cdots \Delta u(t + N - 1) \) leads to

\[
\text{Minimize } J(N) \quad \text{with respect to } \Delta u(t), \Delta u(t + 1) \cdots \Delta u(t + N - 1)
\]

\[
\begin{align*}
\text{subject to } & 
\begin{align*}
\Delta u(t) &= [\Delta u(t) \Delta u(t + 1) \cdots \Delta u(t + N - 1)]^T \\
y &= [\hat{y}(t + d|t) \hat{y}(t + d - 1|t)]^T \\
w &= [u(t + d + 1) u(t + d + 2) \cdots u(t + d + N)]^T.
\end{align*}
\end{align*}
\]

\[M u = Py + Rw\]  

where

\[
u = [\Delta u(t) \Delta u(t + 1) \cdots \Delta u(t + N - 1)]^T
\]

\[
y = [\hat{y}(t + d|t) \hat{y}(t + d - 1|t)]^T
\]

\[
w = [u(t + d + 1) u(t + d + 2) \cdots u(t + d + N)]^T.
\]

\(M\) and \(R\) are matrices of dimension \(N \times N\) and \(P\) of dimension \(N \times 2\). Let us call \(q\) the first row of matrix \(M^{-1}\). Then \(\Delta u(t)\) is given by

\[
\Delta u(t) = q Py + q R w.
\]  

When the future setpoints are unknown, \(u(t + d + i)\) is supposed to be equal to the current reference \(r(t)\). Then the reference sequence can be written as

\[
w = [1 \cdots 1] r(t)
\]

and the control increment \(\Delta u(t)\)

\[
\Delta u(t) = l_{y1} \hat{y}(t + d|t) + l_{y2} \hat{y}(t + d - 1|t) + l_{r1} r(t)
\]  

where \(\alpha = l_{y1} l_{y2} \sum_{i=1}^{N} u_i \sum_{j=1}^{N} r_{ij}\) and \(l_{r1} = \lambda(i)\). The coefficients \(l_{y1}, l_{y2}, l_{r1}\) are functions of \(a, b,\) and \(\lambda(i)\). If the GPC is designed considering the plant to have a unit static gain, the coefficients in (9) will only depend on \(\lambda(i)\) (which is supposed to be fixed) and on the pole of the plant which will change for the adaptive control case. Notice that by doing this, a normalized weighting factor \(\lambda\) is used and that it should be corrected accordingly for systems with different static gains.

The resulting control scheme is shown in Fig. 1. The estimated plant parameters are used to compute the controller coefficients \((l_{y1}, l_{y2}, l_{r1})\). The values \(\hat{y}(t + d|t), \hat{y}(t + d - 1|t)\) are obtained by the use of the predictor given by (5) which basically consists of a model of the plant which is projected toward the future with the values of past inputs and outputs and only requires straightforward computation. The control signal is divided by the process static gain in order to get a system with a unitary static gain.

Notice that the controller coefficients do not depend on the dead time \(d\) and for fixed values of \(\lambda(i)\), they will be a function of the estimated pole \((\hat{a})\). The standard way of computing the controller coefficients would be by computing the matrices \(M, P,\) and \(R\) and solving (7) followed by the generation of the control law of (9). This involves the triangularization of a \(N \times N\) matrix which could be prohibitive for some real-time applications.

The controller coefficients can be obtained by interpolating in a set of previously computed values as shown in Fig. 2.

A. Computation of the Controller Parameters

The algorithm described above can be used to compute controller parameters of GPC for plants which can be described by (3) over a set covering the region of interest.

The curves shown in Fig. 2 correspond to the controller parameters \((l_{y1}, l_{y2}, l_{r1})\) obtained for \(\lambda(i) = 0.8\) and \(N = 15\) as functions of the pole (notice that the pole of the discrete form of the plant transfer function is going to vary between 0.5 and 0.95 for most industrial processes when sampled at appropriate rates [9]). Due to the fact that the closed-loop static gain must be equal to one, the sum of the three parameters equals zero. This result implies that only two of the three parameters need to be known.

By looking at Fig. 2 it can be seen that the functions relating the controller parameters to the process pole can be approximated by functions of the form

\[l_{yi} = k_{i1} + k_{i2} \frac{\alpha}{k_{i3} - \alpha} \quad i = 1, 2\]  

The coefficients \(k_{ij}\) can be calculated by a least squares adjustment using the set of known values of \(l_{yi}\) for different
values of $a$. In this case, the controller coefficients are given by

\[
\begin{align*}
l_{y1} &= -0.845 - 0.564a/(1.05 - a) \\
l_{y2} &= 0.128 + 0.459a/(1.045 - a) \\
l_{r1} &= -l_{y1} - l_{y2}.
\end{align*}
\]

These expressions give a very good approximation to the true controller parameters and fit the set of computed data with a maximum error of less than one percent of the nominal value for the range of interest of the open-loop pole.

The control-weighting factor $\lambda$ affects the control signal in (9). The bigger this value is, the smaller the control effort is allowed to be. If it is given a small value, the system response will be fast since the controller tends to minimize the error between the output and the reference, forgetting about control effort. The controller parameters $l_{y1}, l_{y2},$ and $l_{r1}$ and therefore the closed-loop poles depend on the values of $\lambda$.

A set of functions was obtained by making $\lambda$ change from 0.3 to 1.3 by increments of 0.1. It was found that the values of the parameters $k_{ji}(\lambda)$ of (10) obtained could be approximated by the following functions:

\[
\begin{align*}
k_{11} &= -\exp(0.3598 - 0.9127\lambda + 0.3165\lambda^2) \\
k_{21} &= -\exp(0.0875 - 1.2309\lambda + 0.5086\lambda^2) \\
k_{31} &= 1.05 \\
k_{12} &= \exp(-1.7383 - 0.40403\lambda) \\
k_{22} &= \exp(-0.32157 - 0.81926\lambda + 0.3109\lambda^2) \\
k_{32} &= 1.045.
\end{align*}
\] (11)

The maximum relative error for $0.55 < a < 0.95$ and $0.3 \leq \lambda \leq 1.3$ is less than 3%.

B. Implementation Algorithm

Once the $\lambda$ factor has been decided, the values $k_{ji}$ can very easily be computed by (11) and the approximate adaptation laws given by (10) can easily be employed. The proposed algorithm in the adaptive case can be seen below:

0) Compute $k_{ji}$ with (11),

1) Perform an identification step.

2) Make $l_i = k_{3i} + k_{2i} \frac{\hat{a}}{k_{3i} - \hat{a}}$ for $i = 1, 2$ and $l_{r1} = -l_{y1} - l_{y2}$.

3) Compute $\hat{y}(t + d|t)$ and $\hat{y}(t + d - 1|t)$, using (5) recursively.

4) Compute control signal $u(t)$ with

\[
\Delta u(t) = l_{y1}\hat{y}(t + d|t) + l_{y2}\hat{y}(t + d - 1|t) + l_{r1}r(t)
\]

5) Divide the control signal by the static gain.

6) Go to Step 1).

Notice that in a fixed-parameter case the algorithm is simplified since the controller parameters need to be computed only once (unless the control weighting factor $\lambda$ is changed) and only Steps 3) and 4) have to be carried out at every sampling time.

C. Fractional Dead Time

When the dead time $\tau_d$ of the process is not an integer multiple of the sampling time $T$, $dT \leq \tau_d \leq (d+1)T$, (3) cannot be employed. In this case the fractional delay time can be approximated [10] by the first two terms of the Padé
expansion, which gives a good approximation in the gain although the approximation in the phase usually deteriorates for high frequencies. The plant discrete transfer function can be written as

\[ G(z^{-1}) = \frac{b_0 z^{-1} + b_1 z^{-2}}{1 - a z^{-1}} = \frac{z^{-d}}{1 - a z^{-1}}. \quad (12) \]

The transfer function presents an additional zero; a new parameter appears in the numerator. Using the same procedure as in the previous case, a similar implementation of GPC can be obtained for this family of processes. To obtain the discrete parameters \( a, b_0, \) and \( b_1, \) the following relations can be used [10]: first, the dead time is decomposed as \( \tau_d = dT + \epsilon T \) with \( 0 < \epsilon < 1. \) Then the parameters are

\[ a = e^{-T/\tau}, \quad b_0 = K(1 - a)(1 - \alpha), \quad b_1 = K(1 - \alpha) \alpha \]

with \( \alpha = a(a^{-1} - 1)/(1 - a). \)

Since the derivation of the control law is very similar in this case to the previous section, some steps will be omitted. The system can be written as

\[ y(t + 1) = (1 + a) y(t) - a y(t - 1) + b_0 \Delta u(t - d) + b_1 \Delta u(t - d - 1) + \epsilon(t + 1). \quad (13) \]

If \( \hat{y}(t + d + i - 1|t) \) and \( \hat{y}(t + d + i - 2|t) \) are known, the best expected value for \( \hat{y}(t + d + i|t) \) is given by

\[ \hat{y}(t + d + i|t) = (1 + a) \hat{y}(t + d + i - 1|t) - a \hat{y}(t + d + i - 2|t) + b_0 \Delta u(t + i - 2). \]

Now the control increment \( \Delta u(t) \) results as

\[ \Delta u(t) = l_{y_1} \hat{y}(t + d|t) + l_{y_2} \hat{y}(t + d - 1|t) + l_{u_1} \tau(t) + l_{u_1} \Delta u(t - 1). \quad (14) \]

The values \( \hat{y}(t + d|t) \), \( \hat{y}(t + d - 1|t) \) are obtained by the use of the predictor, that basically consists of a model of the plant which is projected toward the future with the values of past inputs and outputs.

The plant estimated parameters can be used to compute the controller coefficients \( l_{y_1}, l_{y_2}, l_{u_1}, \) and \( l_{u_1}. \) These coefficients are functions of the plant parameters \( a, b_0, b_1, \) and \( \lambda(t). \) If the GPC is designed considering the plant to have a unit static gain, there exists a relationship between the plant parameters \( (1 - a = b_0 + b_1), \) so only two of the three parameters will be needed to calculate the coefficients in (14). One parameter will be the system pole \( a \) and the other will be \( m = b_0/(b_0 + b_1), \) which indicates how close the true dead time is to parameter \( d \) in the model used in (12). That is, if \( m = 1 \) the plant dead time is \( d \) and if \( m = 0 \) it is \( d + 1 \); so a range of \( m \) between one and zero covers the fractional dead times between \( d \) and \( d + 1. \)

Once the values of \( \lambda(t) \) have been chosen and the plant parameters are known, the controller coefficients can easily be derived.

In order to avoid the heavy computational requirements needed to calculate the control signal, the coefficients can be obtained by interpolating in a set of previously computed values as shown in Fig. 3. Notice that this can be accomplished in this case because the controller coefficients only depend on two parameters. As they have been obtained considering a unitary static gain, they must be corrected dividing the coefficients \( l_{y_1}, l_{y_2}, \) and \( l_{u_1} \) by this value.

The curves shown in Fig. 3 correspond to the controller parameters \( l_{y_1}, l_{y_2}, \) and \( l_{u_1} \) for \( \lambda(t) = \lambda \) with \( \lambda = 0, 0.8 \) and \( N = 15. \) Notice that due to the fact that the closed-loop static gain must equal the value one, the sum of parameters \( l_{y_1}, l_{y_2}, \) and \( l_{u_1} \) equals zero. This result implies that only three of the four parameters need to be known.

The expressions relating the controller parameters to the process parameters have been approximated by functions of the form

\[ k_{ji}(m) + k_{j2}(m) \frac{\lambda}{k_{j2}(m) - a}. \quad (15) \]

The coefficients \( k_{ji}(m) \) depend on the value of \( m \) and can be calculated by a least squares fitting using the set of known values of \( l_{y_i} \) for different values of \( a \) and \( m. \) Low-order polynomials that give a good approximation for \( k_{ji}(m) \) have been obtained. In the case of \( m = 0.5, \lambda = 0.8 \) and for a
control horizon of 15, the controller coefficients are given by

\[ l_{g1} = -0.9427 - 0.5486a/(1.055 - a) \]
\[ l_{g2} = 0.1846 + 0.5082a/(1.0513 - a) \]
\[ l_{u1} = -0.3385 + 0.0602a/(1.2318 - a) \]
\[ l_{u1} = -l_{g1} + l_{g2}. \]

These expressions provide a very good approximation to the true controller parameters, fitting the set of computed data with a maximum error of less than 2%.

The influence of the control weighting factor \( \lambda \) on the controller parameters can also be taken into account. For small values of \( \lambda \), the parameters are bigger so as to produce a bigger control effort, thus this factor has to be considered in the approximative functions. With a procedure similar to that of the previous section, the values of \( k_{ji}(m) \) in (15) can be approximated as functions of \( \lambda \), obtaining a maximum error of around 3%.

D. Comparison with Standard GPC

The approximations made in the method can affect the quality of the controlled performance. Some simulation results are presented that compare the results obtained with the proposed method with those when the standard GPC algorithm as originally proposed by Clarke et al. [6] is used.

Two indexes are used to measure the performance: ISE (sum of the square errors during the transient) and ITAE (sum of the absolute error multiplied by discrete time). Also the number of floating point operations and the computing time needed to calculate the control law are analyzed.

First, the performance of the proposed algorithm is compared with that of the standard GPC with no modeling errors. In this situation the error is only caused by the approximative functions of the controller parameters. For the system \( G(s) = \frac{1.5}{(1 + 10s)e^{-1s}} \) with a sampling time of one second, the values for the proposed algorithm when the process is perturbed by a white noise uniformly distributed in the interval \([-0.015, 0.015]\) are: ISE = 7.132, ITAE = 101.106 and for the standard controller: ISE = 7.122, ITAE = 100.536. The plot comparing both responses is not shown because practically there is no difference.

The plant model is supposed to be first-order plus dead-time. If the process behavior can be reasonably described by this model, there will not be a substantial loss of performance. Consider for instance the process modeled by

\[ G_p(z) = \frac{0.1125z^{-1} - 0.225z^{-2}}{1 - z^{-1}} + 0.009z^{-2} z^{-3}. \]

For control purposes it is approximated by the following first-order model, obtained from data generated by the process \( G_p \)

\[ G_m(z) = \frac{0.1125z^{-1}}{1 - 0.8855z^{-1}} z^{-3}. \]

That is, the precomputed GPC is working in the presence of unmodeled dynamics. From the previous studies of robust stability, it can be deduced that the closed-loop system is going to be stable. The performance in this situation is shown in Fig. 4 where the system response for both controllers is shown; notice that for \( t = 100 \) a disturbance is added to the output. The figure also shows the behavior of the precomputed GPC when an additional dead-time mismatch is included, that is, the controller uses a model with \( d = 2 \) instead of the true value \( d = 3 \).

Logically, there is a slight loss of performance due to the uncertainties, that must be considered in conjunction with the benefits in the calculation. Besides, consider that in a real case the uncertainties (such as dead-time mismatch) can also affect the standard GPC since high-frequency effects are usually very difficult to model.

The computational requirements of the method are compared with the standard in Table I for this example, working with a control horizon of \( N = 15 \) and \( \lambda = 0.8 \). The table shows the computation needed for the calculation of the control law both in floating point operations and computation time on a personal computer.

As can be seen, these examples show that although a little performance is lost, there is a great improve in real-time implementability, reaching computing times around 275 times smaller. This advantage can represent a crucial factor for the implementation of this strategy in small size controllers with low computational facilities, considering that the impact on the performance is negligible. The simulations also show the robustness of the controller in the presence of structured uncertainties.
IV. EXTENSION TO INTEGRATING PROCESSES AND RAMP SETPOINTS

In industrial practice it is easy to find some processes including an integral effect. The output of one of these processes grows infinitely when excited by a step input. This is the case of a tank, where the level increases provided there is an input flow and a constant output, or the angle of an electrical motor shaft which grows while being powered until the torque equals the load. The behavior of these processes differs drastically from that of the ones considered up to now in this paper.

These processes need the addition of a 1/s term in order to model the integrating effect. Hence, the transfer function will be

$$G(s) = \frac{K}{s(1 + ts)} e^{-\tau d s}, \quad (16)$$

In the general case of dead time being non multiple of the sampling time the equivalent discrete transfer function when a zero-order hold is employed is given by

$$G(z) = \frac{b_0 z^{-1} + b_1 z^{-2} + b_2 z^{-3}}{(1 - z^{-1})(1 - az^{-1})} z^{-d} \cdot \quad (17)$$

In the simpler case of the dead time being an integer multiple of the sampling time the term $b_2$ disappears.

The GPC control law for processes described by (16) is presented in this section. Notice that some formulations of MPC are unable to deal with these processes since they use the truncated impulse or step response, which is not valid for unstable processes. As GPC makes use of the transfer function, there is no problem about unstable processes.

A. Derivation of the Control Law

The procedure for obtaining the control law is analogous to the one used in previous sections, although, logically, the predictor will be different and the final expression will change slightly.

Now the system can be written as

$$y(t) = (2 + a) y(t) - (1 + 2a) y(t - 1) + a_0 y(t - 2) + b_0 \Delta u(t - d) + b_1 \Delta u(t - d - 1) + b_2 \Delta u(t - d - 2) + \varepsilon(t + 1).$$

If the values of $\dot{y}(t + d + i - 1) \Delta t$, $\dot{y}(t + d + i - 2) \Delta t$, and $\dot{y}(t + d + i - 3) \Delta t$ are known, then the best predicted output at instant $t + d + i$ will be

$$\hat{y}(t + d + i) = (2 + a) \hat{y}(t + d + i - 1) - (1 + 2a) \hat{y}(t + d + i - 2) + a_0 \hat{y}(t + d + i - 3) + b_0 \Delta u(t + i - 1) + b_1 \Delta u(t + i - 2) + b_2 \Delta u(t + i - 3).$$

With these expressions of the predicted outputs, the cost function to be minimized will be a function of $\hat{y}(t + d) \Delta t$, $\dot{y}(t + d - 1 \Delta t)$, and $\dot{y}(t + d - 2 \Delta t)$, as well as the future control signals $\Delta u(t + N - 1), \Delta u(t + N - 2) \ldots \Delta u(t)$, and past inputs $\Delta u(t - 1)$ and $\Delta u(t - 2)$ and, of course, of the reference trajectory.

Minimizing $J(N)$ when the reference is considered to be constant over the prediction horizon and equal to the current setpoint leads to the control law

$$\Delta u(t) = l_{y1} \dot{y}(t + d) \Delta t + l_{y2} \dot{y}(t + d - 1) \Delta t + l_{y3} \dot{y}(t + d - 2) \Delta t + l_{u1} \Delta u(t - 1) + l_{u2} \Delta u(t - 2), \quad (18)$$

Therefore the control law results in a linear expression depending on six coefficients which depend on the process parameters (except on the dead time) and on the control weighting factor $\lambda$. Furthermore, one of these coefficients is a linear combination of the others, since the following relation must hold so as to get a closed loop with unitary static gain

$$l_{y1} + l_{y2} + l_{y3} + l_{r1} = 0. \quad (B)$$

B. Controller Parameters

The control law (18) is very easy to implement provided the controller parameters $l_{y1}, l_{y2}, l_{y3}, l_{r1}, l_{u1},$ and $l_{u2}$ are known. The existence of available relationships of these parameters with process parameters is of crucial importance for a straightforward implementation of the controller. In a similar way to the previous sections, simple expressions for these relationships have been obtained.

As the process can be modeled by (17) four parameters ($a, b_0, b_1, \text{and } b_2$) are needed to describe the plant. Expressions relating the controller coefficients with these parameters can be obtained as previously, although the resulting functions are not as simple, due to the number of plant parameters involved. As the dead time can often be considered as a multiple of the sampling time, simple functions will be obtained for this case from now on. Then $b_2$ will be considered equal to zero.

In a similar way to the process without integrator case, the process can be considered to have $\alpha$ in order to work with normalized plants. Then the computed parameters must be divided by this value.

The controller coefficients will be obtained as a function of the pole $a$ and a parameter: $n = b_0 / (b_1 + b_2)$. This parameter has a short range of variability for any process. As $b_0$ and $b_1$ are related to the continuous parameters by (see [11])

$$b_0 = K(T + \tau [[] - 1 + e^{-c(T/\tau)}]], \quad b_1 = K[\tau + e^{-c(T/\tau)}(T + \tau)]$$

then $n = (a - 1 - \log a) / ((a - 1) \log a)$. That for the usual values of the system pole is going to vary between $n = 0.5$ and $n = 0.36$. Therefore the controller parameters can be expressed as functions of the system pole, and $n$ for a fixed value of $\lambda$.

The shape of the parameters is displayed in Fig. 5 for a fixed value of $\lambda = 1$. It can be seen that the coefficients depend mainly on the pole $a$, being almost independent of $n$ except in the case of $l_{u1}$. Functions of the form

$$f(a, n, \lambda) = k_2(n, \lambda) + k_3(n, \lambda) \frac{\alpha}{k_3(n, \lambda) - a}$$
Fig. 5. Controller coefficients $I_{g1}$, $I_{g2}$, $I_{g3}$, and $I_{u1}$.

where $k_i$ can be approximated by

\begin{align*}
  k_{g1,1} &= -\exp(0.955 - 0.559\lambda + 0.135\lambda^2) \\
  k_{g1,2} &= -\exp(0.5703 - 0.513\lambda + 0.138\lambda^2) \\
  k_{g1,3} &= 1.0343 \\
  k_{g2,1} &= \exp(0.597 - 0.420\lambda + 0.0953\lambda^2) \\
  k_{g2,2} &= \exp(1.016 - 0.4251\lambda + 0.109\lambda^2) \\
  k_{g2,3} &= 1.0289 \\
  k_{g3,1} &= -\exp(-1.761 - 0.422\lambda + 0.071\lambda^2) \\
  k_{g3,2} &= -\exp(0.103 - 0.353\lambda + 0.089\lambda^2) \\
  k_{g3,3} &= 1.0258 \\
  k_{u1,1} &= 1.631n - 1.468 + 0.215\lambda - 0.056\lambda^2 \\
  k_{u1,2} &= -0.124n + 0.158 - 0.026\lambda + 0.006\lambda^2 \\
  k_{u1,3} &= 1.173 - 0.019\lambda \\
\end{align*}

provide good approximations for $I_{g1}$, $I_{g2}$, $I_{g3}$, and $I_{u1}$ in the usual range of the plant parameter variations. The functions fit the set of computed data with a maximum error of less than 1.5% of the nominal values.

C. Consideration of Ramp Setpoints

It is usual for a process reference signal to keep a certain constant value for a time and to move to other constant values by step changes during normal plant operation. This is what has been considered up to now, that is, $w(t + d + 1) = w(t + d + 2) = \cdots = r(t)$, $r(t)$ being the setpoint at instant $t$ which is going to maintain a fixed value. But the reference evolution will not behave like this in all circumstances. On many occasions it can evolve as a ramp, which changes smoothly to another constant setpoint. In general it would be desirable for the process output to follow a mixed trajectory composed of steps and ramps. This situation frequently appears in different industrial processes. In the food and pharmaceutical industries some thermal processes require the temperature to follow a profile given by ramps and steps. It is also of interest that in the control of motors and in robotics applications the position or velocity follow evolutions of this type.

GPC will be reformulated when the reference is a ramp, defined by a parameter $\alpha$ indicating the increment at each sampling time. The reference trajectory is therefore

\begin{align*}
  w(t + d + 1) &= r(t + d) + \alpha \\
  w(t + d + 2) &= r(t + d) + 2\alpha \\
  \cdots &= \cdots \\
  w(t + d + N) &= r(t + d) + N\alpha.
\end{align*}

Employing the procedure used throughout this chapter, and for first-order systems with dead time, the term of the control law including the reference takes the form: $I_{r1} \ r(t + d) + \alpha \ I_{r2}$. 
TABLE II

COEFFICIENTS THAT MAY APPEAR IN THE CONTROL LAW.
THE × INDICATES THAT THE COEFFICIENT EXISTS. \( \omega \) IS THE REFERENCE TRAJECTORY (C = CONSTANT, R = RAMP)

<table>
<thead>
<tr>
<th>Process</th>
<th>( \frac{1}{\tau_2} ) integer</th>
<th>( \frac{1}{\tau_2} ) fractional</th>
<th>( \frac{1}{(1+\tau d)} ) integer</th>
<th>( \frac{1}{(1+\tau d)} ) fractional</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>×</td>
<td>0</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>R</td>
<td>×</td>
<td>0</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

The control law can now be written as

\[
\Delta u(t) = -l_{g1}\dot{y}(t) + l_{g2}\dot{y}(t + d - 1) \, \text{ (20)} + l_{r1}\bar{r}(t + d) + \alpha l_{r2} + l_{u1}\Delta u(t - 1). \, \text{ (21)}
\]

The new coefficient \( l_{r2} \) is due to the ramp. It can be noticed that when the ramp becomes a constant reference, the control law coincides with the one developed for the constant reference case. The only modification that needs to be made because of the ramps is the term \( l_{r2} \). The predictor is the same and the resolution algorithm does not differ from the one used for the constant reference case. The new parameter is a function of the process parameters (\( \alpha, m \)) and of the control weighting factor (\( \lambda \)). As in the previous cases an approximating function can easily be obtained. Notice that the other parameters are exactly the same as in the constant reference case, meaning that the previously obtained expressions can be used.

In what has been seen up to now (nonintegrating processes, integrating processes, constant reference, ramp reference), a new coefficient appeared in the control law with each new situation. All these situations can be described by the following control law:

\[
\Delta u(t) = -l_{g1}\dot{y}(t + d) + l_{g2}\dot{y}(t + d - 1) \, \text{ (20)} + l_{r1}\bar{r}(t + d) + \alpha l_{r2} + l_{u1}\Delta u(t - 1). \, \text{ (21)}
\]

The modeling errors, or uncertainties, can be represented in different forms, reflecting in certain ways the knowledge of the physical mechanisms which cause the discrepancy between the model and the process as well as the capacity to formalize these mechanisms so that they can be handled. Uncertainties can, in many cases, be expressed in a structured way, as expressions in function of determined parameters which can be considered in the transfer function [14]. However, there are usually residual errors particularly dominant at high frequencies which cannot be modeled in this way, which constitute unstructured uncertainties [15]. In this section a study of the precalculated GPC stability in the presence of both types of uncertainties is made; that is, the stability robustness of the method will be studied.

This section aims to study the influence of uncertainties on the behavior of the process working with a controller which has been developed for the nominal model. That is, both the predictor and the controller parameters are calculated for a model which does not exactly coincide with the real process to be controlled. The following question is asked: what discrepancies are permissible between the process and the model in order for the controlled system to be stable?

The controller parameters \( l_{g1}, l_{g2}, l_{r1}, \) and \( l_{u1} \) that appear in the control law

\[
\Delta u(t) = -l_{g1}\dot{y}(t + d) + l_{g2}\dot{y}(t + d - 1) + l_{r1}\bar{r}(t) + l_{u1}\Delta u(t - 1)
\]

have been precalculated for the model (not for the process as this is logically unknown). Likewise the predictor works with the parameters of the model, although it keeps up to date with the values taken from the output produced by the real process.

A. Structured Uncertainties

A first-order model with pure delay, in spite of its simplicity, describes the dynamics of most plants in the process industry. However it is fundamental to consider the case where the model is unable to completely describe all the dynamics of the real process. Two types of structured uncertainties are considered: parametric uncertainties and unmodeled dynamic uncertainties. In the first case, the order of the control model is supposed to be identical to the order of the plant but the parameters are considered to be within an uncertainty region around the nominal parameters (these parameters will be the pole, the gain, and the coefficient \( m = \frac{b_0}{(b_0 + b_2)} \) that measures the fractional delay between \( d \) and \( d + 1 \)). The other type of uncertainty will take into account the existence of process dynamics not included in the control model as an additional unmodeled pole and delay estimation error. This will be reflected in differences between the plant and model orders.

The uncertainty limits have been obtained numerically for the range of variation of the process parameters \( 0.5 < a < 0.98, 0 \leq m \leq 1 \) with a delay \( 0 \leq d \leq 10 \) obtaining the following results (for more details, see [12]).

- **Uncertainty at the Pole:** For a wide working zone \( a < 0.75 \) and for normal values of the delay an uncertainty of more than ±20% is allowed. For higher poles the...
upper limit decreases due almost exclusively to the fact that the open loop would now be unstable. The stable area only becomes narrower for very slow systems with large delays. Notice that this uncertainty refers to the time constant (τ) uncertainty of the continuous process \( [a = \exp(-T/\tau)] \) and thus the time constant can vary around 500% of the nominal one in many cases.

- **Gain Uncertainty:** When the gain of the model is \( G_a \) and that of the process is \( \gamma \times G_a \), \( \gamma \) will be allowed to move between 0.5 and 1.5, that is, uncertainties in the value of the gain of about 50% are permitted. For small delays (1, 2) the upper limit is always above the value \( \gamma = 2 \) and only comes close to the value 1.5 for delays of about ten. It can thus be concluded that the controller is very robust when faced with this type of error.

- **Uncertainty in \( m \):** The effect of this parameter can be ignored since a variation of 300% is allowed without reaching instability.

- **Unmodeled Pole:** The real process has another less dominant pole \((k \times a)\) apart from the one appearing in the model \((a)\), and the results show that the system is stable even for values of \( k \) close to one; stability is only lost for systems with very large delays.

- **Delay Estimation Error:** From the results obtained in a numerical study, it is deduced that for small delays stability is guaranteed for errors of up to two units through all the range of the pole, but when bigger poles are dealt with this only happens for small values of \( \alpha \), and even for delay ten only a delay mismatch of one unit is permitted. It can be concluded, therefore, that a good delay estimation is fundamental to GPC, because for errors of more than one unit the system can become unstable if the process delay is high.

### B. Unstructured Uncertainties

In order to consider unstructured uncertainties, it will be assumed from now on that the dynamic behavior of a determined process is described not by an invariant time linear model but by a family of linear models. Thus the real possible processes \((G)\) will be in a vicinity of the nominal process \((\hat{G})\), which will be modeled by a first-order plus delay system.

A family \( \mathcal{F} \) of processes in the frequency domain will therefore be defined which in the Nyquist plane will be represented by a region about the nominal plant for each \( \omega \) frequency. If this family is defined as

\[
\mathcal{F} = \{ G; |G(i\omega) - \hat{G}(i\omega)| \leq l_a(\omega) \}
\]

the region consists of a disc with its center at \( G(\omega) \) and radius \( l_a(\omega) \). Therefore any member of the family fulfills the condition

\[
G(i\omega) = \hat{G}(i\omega) + l_a(i\omega) \quad |l_a(i\omega)| \leq l_a(\omega).
\]

This region will change with \( \omega \) because \( l_a \) does and, therefore, in order to describe family \( \mathcal{F} \) we will have a zone formed by the discs at different frequencies.

If one wishes to work with multiplicative uncertainties the family of processes can be described by

\[
\mathcal{F} = \left\{ G; \left| \frac{G(i\omega) - \hat{G}(i\omega)}{\hat{G}(i\omega)} \right| \leq l_m(\omega) \right\}
\]

simply considering

\[
l_m(i\omega) = l_a(i\omega)/\hat{G}(i\omega) \quad l_m(i\omega) = l_a(\omega)/|\hat{G}(i\omega)|.
\]

Therefore any member of family \( \mathcal{F} \) satisfies that

\[
G(i\omega) = \hat{G}(i\omega)\left[1 + l_m(i\omega)\right] \quad |l_m(i\omega)| \leq l_m(\omega).
\]

This representation of uncertainties in the Nyquist plane as a disc around the nominal process can encircle any set of structured uncertainties, although sometimes it can result in a rather conservative attitude [16].

The measurement of the robustness of the method can be tackled using the robust stability theorem [16], that for discrete systems states:

**Suppose that all processes \( G \) of family \( \mathcal{F} \) have the same number of unstable poles, which do not become unobservable for the sampling, and that a controller \( C(\zeta) \) stabilizes the nominal process \( \hat{G}(\zeta) \). Then the system has robust stability with controller \( C \) if and only if the complementary sensitivity function for the nominal process satisfies the following relation:**

\[
|I_e(i\omega)\hat{C}(i\omega)| l_m(\omega) < 1 \quad 0 \leq \omega \leq \pi/T.
\]

Using this condition the robustness limits will be obtained for systems that can be described by (12), and for all the values of the parameters that describe the system \((a, m, \alpha, \text{ and } d)\). For each value of the frequency \( \omega \) the limits can be calculated as

\[
l_m = \left| 1 + \hat{G}C \right| \quad l_a = l_m|\hat{G}|.
\]

In Fig. 6 the form taken by the limits in function of \( \omega T \) can be seen for some values of \( \alpha \) and fixed values of \( m \) and \( d \). Both limits are practically constant and equal to unity at low frequencies and change (the additive limit \( l_a \) decreases and the multiplicative \( l_m \) increases) at a certain point. Notice that these curves show the great degree of robustness that the GPC possesses since \( l_m \) is relatively big at high frequencies, where multiplicative uncertainties are normally smaller than unity, and increases with frequency as uncertainties do. The small value of \( l_a \) at high frequencies is due to the fact that the process itself has a small gain at those frequencies; remember that both limits are dependent and related by \( l_a = l_m|\hat{G}| \).

Fig. 7 shows the frequency response of the nominal process alone with the controller, as well as the discs of radius \( l_a \) and \( l_m|\hat{G}| \) for a certain frequency. All the \( G \) processes belonging to the \( \mathcal{F} \) family maintaining the stability of the closed loop can be found inside the disc of radius \( l_a \). The shape of the frequency response leads to limits \( l_a \) and \( l_m \).

Thus, \( \hat{G}C(\omega) \) has a big modulus (due to the integral term) at low frequencies, leading to a value of \( l_m \) close to unity. When \( \omega \) increases, \( \hat{G}C(\omega) \) separates from \(-1\) (without decreasing in modulus) and therefore the limit can safely grow.
Fig. 6. General shape of $T_a$ and $T_m$.

Fig. 7. Polar diagram of the process $\hat{G}$ and $\hat{GC}$ showing the limits for a given frequency.

It can be seen that the most influential parameters are pole $a$ and delay $d$. The evolution of limit $T_a$ with frequency $(\omega T)$ for parameter $a$ changing between 0.5 and 0.98 is presented in Fig. 8 for a concrete value of delay, $d = 1$, and for an average value of $m$, $m = 0.5$. As was to be expected, the limit decreases for greater poles because with open-loop poles near to the limit of the unit circle the uncertainty allowed is less, as it would be easier to enter the open-loop unstable zone.

C. General Comments

The results obtained for both types of uncertainties are qualitatively the same. It can be concluded that the factor that mainly affects robustness is delay uncertainty, because of its effect at high frequencies. The robustness zone decreases when the open-loop pole increases while the parameter $m$ hardly has any influence. As the analysis has been performed based on a particular choice of parameters in the GPC formulation the conclusions depend on these values. The influence of the choice of these parameters on the closed-loop stability is studied in [17].

In any case, the GPC algorithm presented has shown itself to be very robust against the types of uncertainties considered. For small delays the closed loop is stable for static gain mismatch of more than 100% and time constant mismatch of more than 200%.

The stability robustness of GPC can be improved with the use of an observer polynomial, the so-called $T(z^{-1})$ polynomial. In [18] a reformulation of the standard GPC algorithm including this polynomial can be found. In order to do this, the CARIMA model is expressed in the form

\[
A(z^{-1})y(t) = B(z^{-1})u(t-1) + \frac{T(z^{-1})}{\Delta} \xi(t).
\]

Up to now the $T(z^{-1})$ has been considered equal to one, describing the most common disturbances or as the coloring polynomial $C(z^{-1})$. But it can also be considered as a design parameter. In consequence the predictions will not be optimal but on the other hand robustness in the face of uncertainties can be achieved, in a similar interpretation as that used by Ljung [19]. Then this polynomial can be considered as a prefilter as well as an observer. The effective use of observers is known to play an essential role in the robust realization of predictive controllers (see [18] for the effect of prefiltering on robustness and [20] for guidelines for the selection of $T$).

This polynomial can be easily added to the proposed formulation, computing the prediction with the values of inputs and outputs filtered by $T(z^{-1})$. Then, the predictor works with $y^f(t) = y(t)/T(z^{-1})$ and $u^f(t) = u(t)/T(z^{-1})$. The actual prediction for the control law is computed as $\hat{y}(t+d) = T(z^{-1})\hat{y}^f(t+d)$. 
VI. APPLICATIONS

This section is dedicated to presenting some applications of the proposed GPC scheme to the control of different processes. A self-tuning version of this method has been successfully applied to a distributed collector field of a solar power plant [21], [22]. The application of this control strategy to an evaporator can be found in [12]. In this paper, and in order to illustrate how easily the control scheme can be used in any commercial distributed control system, some applications concerning the control of temperatures of different processes of a pilot plant are presented.

The tests are carried out on a pilot plant existing in the Departamento de Ingeniería de Sistemas y Automática of the University of Seville, Spain, that is used as a test bed for new control strategies which can be implemented on an industrial SCADA connected to it. This plant is basically a system using water as the working fluid in which various thermodynamic processes with interchange of mass and energy can take place. It essentially consists of a tank with internal heating with a series of input–output pipes and recirculation circuit with a heat exchanger.

The design of the plant allows for various control strategies to be tested in a large number of loops. Depending on the configuration chosen, it is possible to control the types of magnitudes most frequently found in the process industry such as temperature, flow, pressure, and level. For this, four actuators are available: three automatic valves and one electric heater that heats the interior of the tank. Later two of the possible loops are chosen (considered as being independent) for implanting the GPC controllers.

A. Plant Description

A diagram of the plant which shows its main elements is given in Fig. 9. It has a feed circuit with two input pipes, a cold water one and a hot water one, with motorized valves for regulating the input flows, and a thermally insulated tank with a 15-kW electric resistance for heating. The hot water in the tank can be cooled by entering cold water through the cooling circuit, composed of a centrifugal pump that circulates the hot water from the bottom of the tank through a tube bundle heat exchanger returning at a lower temperature at its top.

B. Plant Control

To control the installation there is an ORSI Automazione Integral Cube distributed control system, composed of a con-
controller and a supervisor connected by a local data highway. The former is in charge of carrying out the digital control and analogous routines while the latter acts as a programming and communications platform with the operator. On this distributed control system the GPC algorithms seen before will be implemented. This control system constitutes a typical example of an industrial controller, having the most normal characteristics of medium size systems to be found in the market today. As in most control computers the calculation facilities are limited and there is little time available for carrying out the control algorithm because of the attention called for by other operations. It is thus an excellent platform for implanting precomputed GPC in industrial fields.

From all the possible loops that could be controlled the results obtained in two situations will be shown. These are: control of the output temperature of the heat exchanger $TT_4$ with valve $V_6$ and control of the tank temperature $TT_5$ with the resistance.

C. Temperature Control at the Exchanger Output

The heat exchanger can be considered to be an independent process within the plant. The exchanger reduces the temperature of the recirculation water, driven by the pump, using a constant flow of cold water for this. The way of controlling the output temperature is by varying the flow of the recirculation water with the motorized valve $V_6$; thus the desired temperature is obtained by variations in the flow. In brief, the heat exchanger is nothing more than a tube bundle with hot water inside that exchanges heat with the exterior cold water. It can thus be considered as being formed of a large number of first-order elements that together act as a first-order system with pure dead time (2). Thus the $TT_4-V_6$ system will be approached by a transfer function of this type. Firstly the parameters identifying the process are obtained using the reaction curve, and then the coefficients of the GPC are found using the proposed method. From the data obtained when a
step is produced, it can be calculated that

\[ K = 0.119, \quad \tau = 18\; \text{s}, \quad \tau_d = 20\; \text{s} \]

when a sampling time \( T = 4\; \text{s} \) is used, the parameters for the corresponding discrete model are

\[ a = 0.80073, \quad b = 0.0237, \quad d = 5. \]

The control signal can easily be computed using the expression

\[ u(t) = u(t-1) + \left[ y_1 g(t+5) + l_2 g(t+4) + l_{r1} r(t) \right] / K \]

where \( u(t) \) is the position of the valve \( V_0 \) and \( y(t) \) is the value of the temperature \( T_{T1} \). Using the approximation (10) with \( \lambda = 0.8 \), the controller gains result as

\[ l_{g1} = -2.656, \quad l_{g2} = 1.632, \quad l_{r1} = 1.023. \]

Some of the results obtained are shown in Fig. 10. The set-point was changed from 38 to 34°C. As can be seen in Fig. 10, the heat exchanger outlet temperature evolved to the new setpoint quite smoothly without exhibiting oscillations. Two different types of external disturbances were introduced. First the manual valve of the refrigerating cold water was closed for a few seconds. As was to be expected, the outlet temperature of the heat exchanger increased very rapidly because of this strong external perturbation and then it was taken back to the desired value by the GPC. The second perturbation is caused by decreasing the duty cycle of the resistor in the tank, thus decreasing the inlet hot water temperature and changing the heat exchanger operating point. As can be seen, the GPC rejects almost completely this perturbation, caused by a change in its dynamics.

**D. Temperature Control in the Tank**

The next example chosen is also that of a very typical case in the process industry: the temperature of the liquid in a tank.
Fig. 12. Results obtained with a PID controller.

The manipulated variable in this case is the duty cycle of the heating resistor.

The process has integral effect and was identified around the nominal operating conditions (50°C). The following model was obtained:

\[ G(s) = \frac{0.41}{s(1 + 50s)} e^{-50s}. \]

The GPC was applied with a sampling time \( T = 10 \) s, \( \lambda = 1.2 \) and \( N = 15 \). As in the previous case, the controller parameters were computed by the formulas given for integrating processes. The results obtained are shown in Fig. 11. A perturbation (simulating a major failure of the actuator) was introduced. As can be seen, after the initial drop in the temperature of the tank, caused by the lack of actuation, the control system is able to take the tank temperature to the desired value with a very smooth transient. A change in the set point from 50 to 60°C was then introduced. The temperature of the tank evolves between both set points without big oscillations.

E. Comparison with a PID Controller

The main objective of the control examples presented in this section was to show how easily GPC can be implemented on a commercial distributed control system by using this implementation technique. The GPC’s were implemented without difficulties using the programming language (ITER) of the Integral Cube distributed control system.

Although comparing the results obtained by GPC with those obtained by using other control techniques was not one of the objectives, GPC has been shown to produce better results than the traditional PID on the examples treated. In the case of the heat exchanger, a PID was tuned by the Ziegler–Nichols open-loop tuning rules and was tested under the same conditions as the proposed GPC (set point change and two perturbations).
Although the controller reacts to the setpoint change, the output shows oscillatory behavior (see upper graph in Fig. 12) after the first perturbation. Better results were obtained after a long commissioning period where “optimal” PID parameters were found (see lower graph), although always worse than the ones obtained with the proposed method (Fig. 10). The commissioning of the GPC controllers was done virtually in no time, since the computation of the controller parameters is straightforward.

VII. CONCLUSIONS

A method for approximating GPC parameters for a wide class of industrial processes has been presented. Very simple formulas have been obtained to approximate the GPC parameters, allowing an adaptive policy even when the available time is limited by the computer performance or the number of loops to be controlled. This straightforward formulation not only reduces calculations, but poses GPC implementation and tuning in an intuitive form similar to that usually employed by practitioners used to PID controllers and Ziegler–Nichols tuning rules.

The stability robustness analysis shows that the method is rather insensitive to uncertainties, especially in time constant and static gain. It can be applied to the majority of industrial processes, provided they can be modeled by a first-order system plus dead time (possibly including an integrator). The application to a pilot plant controlled by a commercial control system shows how easily the method can be implemented as well as the goods results than can be obtained.

ACKNOWLEDGMENT

The authors would like to acknowledge the anonymous reviewers for their helpful comments.

REFERENCES


Carlos Bordons received the Ph.D. degree in electrical engineering from the University of Sevilla in 1981.

He joined the Escuela Superior de Ingenieros de Sevilla as an Assistant Professor in 1989 and is currently an Associate Professor there. He has worked on different projects in collaboration with industry in fields such as control of steam generation in sugar factories or simulation and optimization of oil pipeline networks. His current research interests include advanced process control, especially model-based predictive control, as well as robust and adaptive control. He is coauthor of the book Model Predictive Control in the Process Industry (London, U.K.: Springer-Verlag, 1995).

Eduardo F. Camacho (M’85) received the Ph.D. degree in electrical engineering from the University of Sevilla, Spain, in 1977.

He is the Chairman of the Departamento de Ingeniería de Sistemas y Automática of the University of Sevilla. He has more than 20 years of experience in computer simulation and control where he has worked and coordinated projects in cooperation with industry. He has written two books: Model Predictive Control in the Process Industry and Advanced Control of Solar Power Plants (Berlin, Germany: Springer-Verlag). He has authored and coauthored more than 100 technical papers in international journals and conferences.

He is a member of the European Control Association governing body and has served as a member of different technical committees such as the Application Committee and the System Engineering Committee of the IFAC. He has been serving as reviewer for different technical journals such as the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, International Journal of Adaptive Control and Signal Processing, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, and Control Engineering Practice, where he is at present one of the editors.