Automatica, Vol. 23, No. 2, pp. 137–148, 1987
Printed in Great Britain.

0005-1098/87 $3.00+0.00
Pergamon Journals Ltd.

Generalized Predictive Control—Part I. The Basic Algorithm*

D. W. CLARKE†, C. MOHTADI† and P. S. TUFFS‡

A new member of the family of long-range predictive controllers is shown to be suitable for the adaptive control of processes with varying parameters, dead-time and model-order.

Key Words—Adaptive control; predictive control; LQ control; self-tuning control; nonminimum-phase plant.

Abstract—Current self-tuning algorithms lack robustness to prior choices of either dead-time or model order. A novel method—generalized predictive control or GPC—is developed which is shown by simulation studies to be superior to accepted techniques such as generalized minimum-variance and pole-placement. This receding-horizon method depends on predicting the plant's output over several steps based on assumptions about future control actions. One assumption—that there is a "control horizon" beyond which all control increments become zero—is shown to be beneficial both in terms of robustness and for providing simplified calculations. Choosing particular values of the output and control horizons produces as subsets of the method various useful algorithms such as GMV, EPSAC, Peterka's predictive controller (1984, *Automatica*, **20**, 39–50) and Ydstie's extended-horizon design (1984, IFAC 9th World Congress, Budapest, Hungary). Hence GPC can be used either to control a "simple" plant (e.g. open-loop stable) with little prior knowledge or a more complex plant such as nonminimum-phase, open-loop unstable and having variable dead-time. In particular GPC seems to be unaffected (unlike pole-placement strategies) if the plant model is overparameterized. Furthermore, as offsets are eliminated by the consequence of assuming a CARIMA plant model, GPC is a contender for general self-tuning applications. This is verified by a comparative simulation study.

1. INTRODUCTION

ALTHOUGH SELF-TUNING and adaptive control has made much progress over the previous decade, both in terms of theoretical understanding and practical applications, no one method proposed so far is suitable as a "general purpose" algorithm for the stable control of the majority of real processes. To be considered for this role a method must be applicable to:

(1) a nonminimum-phase plant: most continuous-time transfer functions tend to exhibit discrete-time zeros outside the unit circle when sampled at a fast enough rate (see Clarke, 1984);

(2) an open-loop unstable plant or plant with badly-damped poles such as a flexible spacecraft or robots;

(3) a plant with variable or unknown dead-time: some methods (e.g. minimum-variance self-tuners, Åström and Wittenmark, 1973) are highly sensitive to the assumptions made about the dead-time and approaches (e.g. Kurz and Goedecke, 1981) which attempt to estimate the dead-time using operating data tend to be complex and lack robustness;

(4) a plant with unknown order: pole-placement and LQG self-tuners perform badly if the order of the plant is overestimated because of pole/zero cancellations in the identified model, unless special precautions are taken.

The method described in this paper—Generalized Predictive Control or GPC—appears to overcome these problems in one algorithm. It is capable of stable control of processes with variable parameters, with variable dead-time, and with a model order which changes instantaneously provided that the input/output data are sufficiently rich to allow reasonable plant identification. It is effective with a plant which is simultaneously nonminimum-phase and open-loop unstable and whose model is overparameterized by the estimation scheme without special precautions being taken. Hence it is suited to high-performance applications such as the control of flexible systems.

Hitherto the principal applied self-tuning methods have been based on the "Generalized Minimum-Variance" approach (Clarke and Gawthrop, 1975, 1979) and the pole-placement algorithm (Wellstead *et al.*, 1979; Åström and

* Received 2 March 1985; revised 7 July 1985; revised 3 March 1986; revised 22 September 1986. The original version of this paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor M. Gevers under the direction of Editor P. C. Parks.

† Department of Engineering Science, Parks Road, Oxford OX1 3PJ, U.K.

‡ Aluminum Company of America, Alcoa Technical Center, Pittsburgh, PA 15069, U.S.A.

Wittenmark, 1980). The implicit GMV self-tuner (so-called because the controller parameters are directly estimated without an intermediate calculation) is robust against model order assumptions but can perform badly if the plant dead-time varies. The explicit pole-placement method, in which a Diophantine equation is numerically solved as a bridging step between plant identification and the control calculation, can cope with variable dead-time but not with a model with overspecified order. Its good behaviour with variable dead-time is due to overparameterization of the numerator dynamics $B(q^{-1})$; this means that the order of the denominator dynamics has to be chosen with great care to avoid singularities in the resolution of the Diophantine identity. The GPC approach being based on an explicit plant formulation can deal with variable dead-time, but as it is a predictive method it can also cope with overparameterization.

The type of plant that a self-tuner is expected to control varies widely. On the one hand, many industrial processes have "simple" models: low order with real poles and probably with dead-time. For some critical loops, however, the model might be more complex, such as open-loop unstable, underdamped poles, multiple integrators. It will be shown that GPC has a readily-understandable *default* operation which can be used for a simple plant without needing the detailed prior design of many adaptive methods. Moreover, at a slight increase of computational time more complex processes can be accommodated by GPC within the basic framework.

All industrial plants are subjected to load-disturbances which tend to be in the form of random-steps at random times in the deterministic case or of Brownian motion in stochastic systems. To achieve offset-free closed-loop behaviour given these disturbances the controller must possess inherent integral action. It is seen that GPC adopts an integrator as a natural consequence of its assumption about the basic plant model, unlike the majority of designs where integrators are added in an *ad hoc* way.

2. THE CARIMA PLANT MODEL AND OUTPUT PREDICTION

When considering regulation about a particular operating point, even a non-linear plant generally admits a locally-linearized model:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + x(t) \tag{1}$$

where A and B are polynomials in the backward shift operator q^{-1}:

$$A(q^{-1}) = 1 + a_1 q^{-1} + \cdots + a_{na} q^{-na}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \cdots + b_{nb} q^{-nb}.$$

If the plant has a non-zero dead-time the leading elements of the polynomial $B(q^{-1})$ are zero. In (1), $u(t)$ is the control input, $y(t)$ is the measured variable or output, and $x(t)$ is a disturbance term.

In the literature $x(t)$ has been considered to be of moving average form:

$$x(t) = C(q^{-1})\xi(t) \tag{2}$$

where $C(q^{-1}) = 1 + c_1 q^{-1} + \cdots + c_{nc} q^{-nc}$.

In this equation, $\xi(t)$ is an uncorrelated random sequence, and combining with (1) we obtain the CARMA (Controlled Auto-Regressive and Moving-Average) model:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1})\xi(t). \tag{3}$$

Though much self-tuning theory is based on this model it seems to be inappropriate for many industrial applications in which disturbances are non-stationary. In practice, two principal disturbances are encountered: random steps occuring at random times (for example, changes in material quality) and Brownian motion (found in plants relying on energy balance). In both these cases an appropriate model is:

$$x(t) = C(q^{-1})\xi(t)/\Delta \tag{4}$$

where Δ is the differencing operator $1 - q^{-1}$. Coupled with (1) this gives the CARIMA model (integrated moving-average):

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1})\xi(t)/\Delta.$$

This model has been used by Tuffs and Clarke (1985) to derive GMV and pole-placement self-tuners with inherent integral action. For simplicity in the development here $C(q^{-1})$ is chosen to be 1 (alternatively C^{-1} is truncated and absorbed into the A and B polynomials; see Part II for the case of general C) to give the model:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \xi(t)/\Delta. \tag{5}$$

To derive a j-step ahead predictor of $y(t+j)$ based on (5) consider the identity:

$$1 = E_j(q^{-1})A\Delta + q^{-j}F_j(q^{-1}) \tag{6}$$

where E_j and F_j are polynomials uniquely defined given $A(q^{-1})$ and the prediction interval j. If (5) is multiplied by $E_j \Delta q^j$ we have:

$$E_j A \Delta y(t+j) = E_j B \Delta u(t+j-1) + E_j \xi(t+j)$$

and substituting for $E_j A \Delta$ from (6) gives:

$$y(t+j) = E_j B \Delta u(t+j-1) + F_j y(t) + E_j \xi(t+j). \qquad (7)$$

As $E_j(q^{-1})$ is of degree $j-1$ the noise components are all in the future so that the optimal predictor, given measured output data up to time t and any given $u(t+i)$ for $i > 1$, is clearly:

$$\hat{y}(t+j\,|\,t) = G_j \Delta u(t+j-1) + F_j y(t) \qquad (8)$$

where $G_j(q^{-1}) = E_j B$.

Note that $G_j(q^{-1}) = B(q^{-1})[1 - q^{-j}F_j(q^{-1})]/A(q^{-1})\Delta$ so that one way of computing G_j is simply to consider the Z-transform of the plant's step-response and to take the first j terms (Clarke and Zhang, 1985).

In the development of the GMV self-tuning controller only one prediction $\hat{y}(t+k\,|\,t)$ is used where k is the assumed value of the plant's dead-time. Here we consider a whole set of predictions for which j runs from a minimum up to a large value: these are termed the minimum and maximum "prediction horizons". For $j < k$ the prediction process $\hat{y}(t+j\,|\,t)$ depends entirely on available data, but for $j \geqslant k$ assumptions need to be made about future control actions. These assumptions are the cornerstone of the GPC approach.

2.1. *Recursion of the Diophantine equation*

One way to implement long-range prediction is to have a bank of self-tuning predictors for each horizon j; this is the approach of De Keyser and Van Cauwenberghe (1982, 1983) and of the MUSMAR method (Mosca *et al.*, 1984). Alternatively, (6) can be resolved numerically for E_j and F_j for the whole range of js being considered. Both these methods are computationally expensive. Instead a simpler and more effective scheme is to use recursion of the Diophantine equation so that the polynomials E_{j+1} and F_{j+1} are obtained given the values of E_j and F_j.

Suppose for clarity of notation $E = E_j$, $R = E_{j+1}$, $F = F_j$, $S = F_{j+1}$ and consider the two Diophantine equations with $\tilde{A}$ defined as $A\Delta$:

$$1 = E\tilde{A} + q^{-j}F \qquad (9)$$

$$1 = R\tilde{A} + q^{-(j+1)}S. \qquad (10)$$

Subtracting (9) from (10) gives:

$$0 = \tilde{A}(R - E) + q^{-j}(q^{-1}S - F).$$

The polynomial $R - E$ is of degree j and may be split into two parts:

$$R - E = \tilde{R} + r_j q^{-j}$$

so that:

$$\tilde{A}\tilde{R} + q^{-j}(q^{-1}S - F + \tilde{A}r_j) = 0.$$

Clearly then $\tilde{R} = 0$ and also S is given by $Sq(F - \tilde{A}r_j)$.

As $\tilde{A}$ has a unit leading element we have:

$$r_j = f_0 \qquad (11a)$$

$$s_i = f_{i+1} - \tilde{a}_{i+1} r_j \qquad (11b)$$

for $i = 0$ to the degree of $S(q^{-1})$;

and:
$$R(q^{-1}) = E(q^{-1}) + q^{-j} r_j \qquad (12)$$

$$G_{j+1} = B(q^{-1})R(q^{-1}). \qquad (13)$$

Hence given the plant polynomials $A(q^{-1})$ and $B(q^{-1})$ and one solution $E_j(q^{-1})$ and $F_j(q^{-1})$ then (11) can be used to obtain $F_{j+1}(q^{-1})$ and (12) to give $E_{j+1}(q^{-1})$ and so on, with little computational effort. To initialize the iterations note that for $j = 1$:

$$1 = E_1\tilde{A} + q^{-1}F_1$$

and as the leading element of $\tilde{A}$ is 1 then:

$$E_1 = 1, \qquad F_1 = q(1 - \tilde{A}).$$

The calculations involved, therefore, are straightforward and simpler than those required when using a separate predictor for each output horizon.

3. THE PREDICTIVE CONTROL LAW

Suppose a future set-point or reference sequence $[w(t+j);\ j = 1, 2, \ldots]$ is available. In most cases $w(t+j)$ will be a constant w equal to the current set-point $w(t)$, though sometimes (as in batch process control or robotics) future variations in $w(t+j)$ would be known. As in the IDCOM algorithm (Richalet *et al.*, 1978) it might be considered that a smoothed approach from the current output $y(t)$ to w is required which is obtainable from the simple

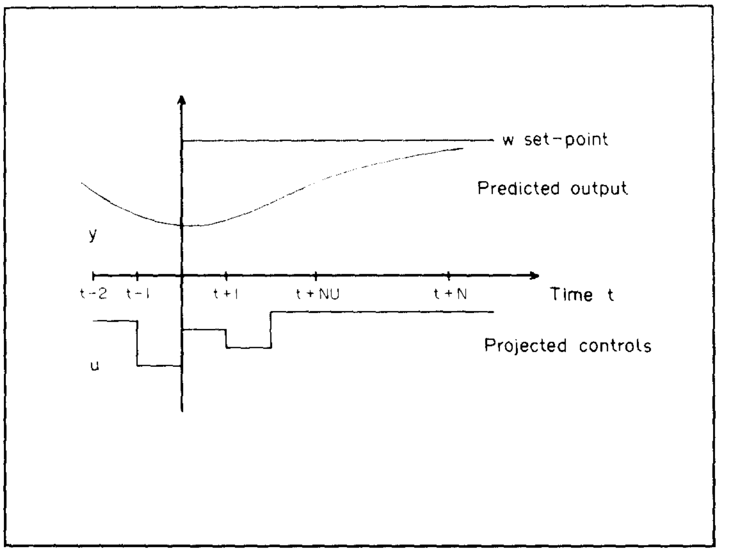

FIG. 1. Set-point, control and outputs in GPC.

first-order lag model:

$$w(t) = y(t)$$
$$w(t+j) = \alpha w(t+j-1) + (1-\alpha)w \qquad j = 1, 2, \ldots$$

where $\alpha \approx 1$ for a slow transition from the current measured-variable to the real set-point w. GPC is capable of considering both constant and varying future set-points.

The objective then of the predictive control law is to drive future plant outputs $y(t+j)$ "close" to $w(t+j)$ in some sense, as shown in Fig. 1, bearing in mind the control activity required to do so. This is done using a receding-horizon approach for which at each sample-instant t:

(1) the future set-point sequence $w(t+j)$ is calculated;

(2) the prediction model of (8) is used to generate a set of predicted outputs $\hat{y}(t+j\,|\,t)$ with corresponding predicted system errors $e(t+j) = w(t+j) - \hat{y}(t+j\,|\,t)$ noting that $\hat{y}(t+j\,|\,t)$ for $j > k$ depends in part on future control signals $u(t+i)$ which are to be determined;

(3) some appropriate quadratic function of the future errors and controls is minimized, assuming that after some "control horizon" further increments in control are zero, to provide a suggested sequence of future controls $u(t+j)$;

(4) the first element $u(t)$ of the sequence is asserted and the appropriate data vectors shifted so that the calculations can be repeated at the next sample instant.

Note that the effective control law is stationary, unlike a fixed-horizon LQ policy. However, in the self-tuned case new estimates of the plant model parameters requires new values for the parameter polynomials, which means that the fast Diophantine recursion is useful for adaptive control applications of the GPC method.

Consider a cost function of the form:

$$J(N_1, N_2) = E\left\{\sum_{j=N_1}^{N_2} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_2} \lambda(j)[\Delta u(t+j-1)]^2\right\} \tag{14}$$

where:

N_1 is the minimum costing horizon;
N_2 is the maximum costing horizon, and
$\lambda(j)$ is a control-weighting sequence.

The expectation in (14) is conditioned on data up to time t assuming no future measurements are available (i.e. the set of control signals are applied in open-loop in the sequel). As mentioned earlier, the first control is applied and the minimization is repeated at the next sample. The resulting control law belongs to the class known as Open-Loop-Feedback-Optimal control (Bertsekas, 1976). Appendix A examines the relation between GPC and Closed-Loop-Feedback-Optimal when the disturbance process is autoregressive. It is seen that costing on the control is over all future inputs which affect the outputs included in J. In general N_2 is chosen to encompass all the response which is significantly affected by the current control; it is reasonable that it should at least be greater than the degree of $B(q^{-1})$ as then all states contribute to the cost (Kailath, 1980), but more typically N_2 is set to approximate the rise-time of the plant. N_1 can often be taken as 1; if it is known *a priori* that the dead-time of the plant is at least k sample-intervals then

N_1 can be chosen as k or more to minimize computations. It is found, however, that a large class of plant models can be stabilized by GPC with default values of 1 and 10 for N_1 and N_2. Part II provides a theoretical justification for these choices of horizon. For simplicity in the derivation, below $\lambda(j)$ is set to the constant λ, N_1 to 1 and N_2 to N: the "output horizon".

Recall that (7) models the future outputs:

$$\begin{aligned}
y(t+1) &= G_1\Delta u(t) + F_1 y(t) + E_1\xi(t+1)\\
y(t+2) &= G_2\Delta u(t+1) + F_2 y(t) + E_2\xi(t+2)\\
&\cdot\\
&\cdot\\
y(t+N) &= G_N\Delta u(t+N-1)\\
&\quad + F_N y(t) + E_N\xi(t+N).
\end{aligned}$$

Consider $y(t+j)$. It consists of three terms: one depending on future control actions yet to be determined, one depending on past known controls together with filtered measured variables and one depending on future noise signals. The assumption that the controls are to be performed in open-loop is tantamount to ignoring the future noise sequence $\{\xi(t+j)\}$ in calculating the predictions. Let $f(t+j)$ be that component of $y(t+j)$ composed of signals which are known at time t, so that for example:

$$\begin{aligned}
f(t+1) &= [G_1(q^{-1}) - g_{10}]\Delta u(t)\\
&\quad + F_1 y(t), \quad \text{and}\\
f(t+2) &= q[G_2(q^{-1}) - q^{-1}g_{21}\\
&\quad - g_{20}]\Delta u(t) + F_2 y(t),\\
&\quad \text{etc.}
\end{aligned}$$

where $G_i(q^{-1}) = g_{i0} + g_{i1}q^{-1} + \dots$.

Then the equations above can be written in the key vector form:

$$\hat{\mathbf{y}} = \mathbf{G}\tilde{\mathbf{u}} + \mathbf{f} \tag{15}$$

where the vectors are all $N \times 1$:

$$\begin{aligned}
\hat{\mathbf{y}} &= [\hat{y}(t+1), \hat{y}(t+2), \dots, \hat{y}(t+N)]^{\mathrm{T}}\\
\tilde{\mathbf{u}} &= [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N-1)]^{\mathrm{T}}\\
\mathbf{f} &= [f(t+1), f(t+2), \dots, f(t+N)]^{\mathrm{T}}.
\end{aligned}$$

As indicated earlier, the first j terms in $G_j(q^{-1})$ are the parameters of the step-response and therefore $g_{ij} = g_j$ for $j = 0, 1, 2, \dots < i$ independent of the particular G polynomial.

The matrix $\mathbf{G}$ is then lower-triangular of dimension $N \times N$:

$$\mathbf{G} = \begin{bmatrix} g_0 & 0 & \cdots & 0\\ g_1 & g_0 & \cdots & 0\\ \cdot & \cdot\cdot & \cdots & \cdot\\ \vdots & & & \cdot\\ g_{N-1} & g_{N-2} & \cdots & g_0 \end{bmatrix}.$$

Note that if the plant dead-time is $k > 1$ the first $k-1$ rows of $\mathbf{G}$ will be null, but if instead N_1 is assumed to be equal to k the leading element is non-zero. However, as k will not in general be known in the self-tuning case one key feature of the GPC approach is that a stable solution is possible *even if the leading rows of* G *are zero.*

From the definitions of the vectors above and with:

$$\mathbf{w} = [w(t+1), w(t+2), \dots, w(t+N)]^{\mathrm{T}}$$

the expectation of the cost-function of (14) can be written:

$$\begin{aligned}
J_1 &= E\{J(1,N)\}\\
&= E\{(\mathbf{y}-\mathbf{w})^{\mathrm{T}}(\mathbf{y}-\mathbf{w}) + \lambda\tilde{\mathbf{u}}^{\mathrm{T}}\tilde{\mathbf{u}}\}
\end{aligned} \tag{16}$$

i.e.

$$J_1 = \{(\mathbf{G}\tilde{\mathbf{u}} + \mathbf{f} - \mathbf{w})^{\mathrm{T}}(\mathbf{G}\tilde{\mathbf{u}} + \mathbf{f} - \mathbf{w}) + \lambda\tilde{\mathbf{u}}^{\mathrm{T}}\tilde{\mathbf{u}}\}.$$

The minimization of J_1 assuming no constraints on future controls results in the projected control-increment vector:

$$\tilde{\mathbf{u}} = (\mathbf{G}^{\mathrm{T}}\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^{\mathrm{T}}(\mathbf{w} - \mathbf{f}). \tag{17}$$

Note that the first element of $\tilde{\mathbf{u}}$ is $\Delta u(t)$ so that the current control $u(t)$ is given by:

$$u(t) = u(t-1) + \bar{\mathbf{g}}^{\mathrm{T}}(\mathbf{w} - \mathbf{f}) \tag{18}$$

where $\bar{\mathbf{g}}^{\mathrm{T}}$ is the first row of $(\mathbf{G}^{\mathrm{T}}\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^{\mathrm{T}}$. Hence the control includes integral action which provides zero offset provided that for a constant set-point $w(t+i) = w$, say, the vector $\mathbf{f}$ involves a unit steady-state gain in the feedback path.

Now the Diophantine equation (6) for $q = 1$ gives

$$1 = E_j(1)A(1)\Delta(1) + F_j(1)$$

and as $\Delta(1) = 0$ then $F_j(1) = 1$ so that $f(t+j) = F_j y(t)$ is a signal whose mean value equals that of $y(t)$. Furthermore, defining $F_j'(q^{-1})$ to be $E_j(q^{-1})\tilde{A}(q^{-1})$ gives:

$$\begin{aligned}
F_j(q^{-1})y(t) &= (1 - F_j'(q^{-1})\Delta)y(t)\\
&= y(t) - F_j'\Delta y(t)
\end{aligned}$$

which shows that if $y(t)$ is the constant $\bar{y}$ so that $\Delta y(t) = 0$ then the component $F_j(q^{-1})y(t)$ reduces to $\bar{y}$. This, together with the control given by (18), ensures offset-free behaviour by integral action.

3.1. *The control horizon*

The dimension of the matrix involved in (17) is $N \times N$. Although in the non-adaptive case the inversion need be performed once only, in a self-tuning version the computational load of inverting at each sample would be excessive. Moreover, if the wrong value for dead-time is assumed, $\mathbf{G}^T\mathbf{G}$ is singular and hence a finite non-zero value of weighting λ would be required for a realizable control law, which is inconvenient because the "correct" value for λ would not be known *a priori*.

The real power of the GPC approach lies in the assumptions made about future control actions. Instead of allowing them to be "free" as for the above development, GPC borrows an idea from the Dynamic Matrix Control method of Cutler and Ramaker (1980). This is that after an interval $NU < N_2$ *projected control increments are assumed to be zero*,

$$\text{i.e.} \qquad \Delta u(t + j - 1) = 0 \qquad j > NU. \tag{19}$$

The value NU is called the "control horizon". In cost-function terms this is equivalent to placing effectively infinite weights on control changes after some future time. For example, if $NU = 1$ only one control change (i.e. $\Delta u(t)$) is considered, after which the controls $u(t + j)$ are all taken to be equal to $u(t)$. Suppose for this case that at time t there is a step change in $w(t)$ and that N is large. The choice of $u(t)$ made by GPC is the optimal "mean-level" controller which, if sustained, would place the settled plant output to w with the same dynamics as the open-loop plant. This control law (at least for a simple stable plant) gives actuations which are generally smooth and sluggish. Larger values of NU, on the other hand, provide more active controls.

One useful intuitive interpretation of the use of a control horizon is in the stabilization of nonminimum-phase plant. If the control weighting λ is set to zero the optimal control which minimizes J_1 is a cancellation law which attempts to remove the process dynamics using an inverse plant model in the controller. As is well known, such (minimum-variance) laws are in practice unstable because they involve growing modes in the control signal corresponding to the plant's nonminimum-phase zeros. Constraining these modes by placing infinite costing on future control *increments* stabilizes the resulting closed-loop even if the weighting λ is zero. The use of $NU < N$ moreover significantly reduces the computational burden, for the vector $\tilde{\mathbf{u}}$ is then of dimension NU and the prediction equations reduce to:

$$\hat{\mathbf{y}} = \mathbf{G}_1\tilde{\mathbf{u}} + \mathbf{f}$$

where:

$$\mathbf{G}_1 = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \cdot & & g_0 \\ \vdots & \cdot & & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_{N-NU} \end{bmatrix} \text{ is } N \times NU.$$

The corresponding control law is given by:

$$\tilde{\mathbf{u}} = [\mathbf{G}_1^T\mathbf{G}_1 + \lambda\mathbf{I}]^{-1}\mathbf{G}_1^T(\mathbf{w} - \mathbf{f}) \tag{20}$$

and the matrix involved in the inversion is of the much reduced dimension $NU \times NU$. In particular, if $NU = 1$ (as is usefully chosen for a "simple" plant), this reduces to a scalar computation. An example of the computations involved is given in Appendix B.

3.2. *Choice of the output and control horizons*

Simulation exercizes on a variety of plant models, including stable, unstable and nonminimum-phase processes with variable dead-time, have shown how N_1, N_2 and NU should best be selected. These studies also suggest that the method is robust against these choices, giving the user a wide latitude in his design.

3.2.1. N_1: *The minimum output horizon.* If the dead-time k is exactly known there is no point in setting N_1 to be less than k since there would then be superfluous calculations in that the corresponding outputs cannot be affected by the first action $u(t)$. If k is not known or is variable, then N_1 can be set to 1 with no loss of stability and the degree of $B(q^{-1})$ increased to encompass all possible values of k.

3.2.2. N_2: *The maximum output horizon.* If the plant has an initially negative-going nonminimum-phase response, N_2 should be chosen so that the later positive-going output samples are included in the cost: in discrete-time this implies that N_2 exceeds the degree of $B(q^{-1})$ as demonstrated in Appendix B. In practice, however, a rather larger value of N_2 is suggested, corresponding more closely to the rise-time of the plant.

3.2.3. NU: *The control horizon.* This is an important design parameter. For a simple plant (e.g. open-loop stable though with possible dead-time and nonminimum-phasedness) a value of NU of 1 gives generally acceptable control. Increasing NU makes the control and the corresponding output response more active until a stage is reached where any further increase in NU makes little difference. An increased value of NU is more appropriate for complex systems where it is found that good control is achieved when NU is at least equal to the number of unstable or badly-damped poles.

One interpretation of these rules for chosing NU is as follows. Recall that GPC is a receding-horizon LQ control law for which the future control sequence is recalculated at each sample. For a simple, stable plant a control sequence following a step in the set-point is generally well-behaved (for example, it would not change sign). Hence there would not be significant corrections to the control at the next sample even if $NU = 1$. However, it is known from general state-space considerations that a plant of order n needs n different control values for, say, a dead-beat response. With a complex system these values might well change sign frequently so that a short control horizon would not allow for enough degrees of freedom in the derivation of the current action.

Generally it is found that a value of NU of 1 is adequate for typical industrial plant models, whereas if, say, a modal model is to be stabilized NU should be set equal to the number of poles near the stability boundary. If further damping of the control action is then required λ can be increased from zero. Note in particular that, unlike with GMV, GPC can be used with a nonminimum-phase plant even if λ is zero.

The above discussion implies that GPC can be considered in two ways. For a process control *default* setting of $N_1 = 1$, N_2 equal to the plant rise-time and $NU = 1$ can be used to give reasonable performance. For high-performance applications such as the control of coupled oscillators a larger value of NU is desirable.

4. RELATIONSHIP WITH OTHER APPROACHES

GPC depends on the integration of five key ideas: the assumption of a CARIMA rather than a CARMA plant model, the use of long-range prediction over a finite horizon greater than the dead-time of the plant and at least equal to the model order, recursion of the Diophantine equation, the consideration of weighting of control *increments* in the cost-function, and the choice of a control horizon after which projected control increments are taken to be zero. Many of these ideas have arisen in the literature in one form or another but not in the particular way described here, and it is their judicious combination which gives GPC its power. Nevertheless, it is useful to see how previous successful methods can be considered as subsets of the GPC approach so that accepted theoretical (e.g. convergence and stability) and practical results can be extended to this new method.

The concept of using long-range prediction as a potentially robust control tool is due to Richalet *et al.* (1978) in the IDCOM algorithm. This method, though reportedly having some industrial success, is restricted by its assumption of a weighting-sequence model (all-zeros), with an *ad hoc* way of solving the offset problem and with no weighting on control action. Hence it is unsuitable for unstable or nonminimum-phase open-loop plants. The DMC algorithm of Cutler and Ramaker (1980) is based on step-response models but does include the key idea of a control horizon. Hence it is effective for nonminimum-phase plants but not for open-loop unstable processes. Again the offset problem is dealt with heuristically and moreover the use of a step-response model means that the savings in parameterization using a $A(q^{-1})$ polynomial are not available. Clarke and Zhang (1985) compare the IDCOM and DMC designs.

The GMV approach of Clarke and Gawthrop (1975) for a plant with known dead-time k can be seen to be a special case of GPC in which both the minimum and maximum horizons N_1 and N_2 are set to k and only one control signal (the current control $u(t)$ or $\Delta u(t)$) is weighted. This method is known to be robust against overspecification of model-order but it can only stabilize a certain class of nonminimum-phase plant for which the control weighting λ has to be chosen with reasonable care. Moreover, GMV is sensitive to varying dead-time unless λ is large, with correspondingly poor control. GPC shares the robustness properties of GMV without its drawbacks.

By choosing $N_1 = N_2 = d > k$ and $NU = 1$ with $\lambda = 0$, GPC becomes Ydstie's (1984) extended-horizon approach. This has been shown theoretically to be a stabilizing controller for a stable nonminimum-phase plant. Ydstie, however, uses a CARMA model and the method has not been shown to stabilize open-loop unstable processes. Indeed, for a plant with poorly damped poles simulation experience shows that the extended-horizon method is unstable, unlike the GPC design. This is because in this case more than one future output needs to be accounted for in the cost-function for stability (i.e. $N_2 > N_1$).

With $N_1 = 1$, $N_2 = d > k$ and $NU = 1$ with $\lambda = 0$ and a CARIMA model, GPC reduces to the independently-derived EPSAC algorithm (De Keyser and Van Cauwenberghe, 1985). This was shown by the authors to be a particularly useful method with several practical applications; this is verified by the simulations described below where the "default" settings of GPC that were adopted with $N_2 = 10$ are similar to those of EPSAC. Part II of this paper gives a stability result for these settings of the GPC horizons.

Peterka's elegant predictive controller (1984) is nearest in philosophy to GPC: it uses a CARIMA-like model and a similar cost-function, though concentrating on the case where $N_2 \to \infty$. The algorithmic development is then rather different from GPC, relying on matrix factorization and decomposition for the finite-stage case, instead of the rather more direct formulation described here. Though Peterka considers two values of control weighting (λ_1 for the first set of controls and λ_2 for the final δB steps), he does not consider the useful GPC case where $\lambda_2 = \infty$. GPC is essentially a finite-stage approach with a restricted control horizon. This means (though not considered here) that constraints on both future controls and control rates can be taken into account by suitable modifications of the GPC calculations—a generalization which is impossible to achieve in infinite-stage designs. Nevertheless, the Peterka procedure is an important and applicable approach to adaptive control.

5. A SIMULATION STUDY

The objective of this study is to show how an adaptive GPC implementation can cope with a plant which changes in dead-time, in order and in parameters compared with a fixed PID regulator, with a GMV self-tuner and with a pole-placement self-tuner. The PID regulator was chosen to give good control of the initial plant model which for this case was assumed to be known. For simplicity all the adaptive methods used a standard recursive-least-squares parameter estimator with a fixed forgetting-factor of 0.9 and with no noise included in the simulation.

The graphs display results over 400 samples of simulation for each method, showing the control signal in the range -100 to 100, and the set-point $w(t)$ with the output $y(t)$ in the range -10 to 80. The control signal to the simulated process was clipped to lie in $[-100, 100]$, but no constraint was placed on the plant output so that the limitations on $y(t)$ seen in Fig. 4 are graphical with the actual output exceeding the displayed values.

Each simulation was organized as follows. During the first 10 samples the control signal was fixed at 10 and the estimator (initialized with parameters $[1, 0, 0, \ldots]$) was enabled for the adaptive controllers. A sequence of set-point changes between three distinct levels was provided with switching every 20 samples. After every 80 samples the simulated continuous-time plant was changed, following the models given in Table 1. It is seen that these changes in dynamics are large, and though it is difficult to imagine a real plant varying in such a drastic way,

TABLE 1. TRANSFER-FUNCTIONS OF THE SIMULATED MODELS

Number	Samples	Model
1	1–79	$\dfrac{1}{1 + 10s + 40s^2}$
2	80–159	$\dfrac{e^{-2.7s}}{1 + 10s + 40s^2}$
3	160–239	$\dfrac{e^{-2.7s}}{1 + 10s}$
4	240–319	$\dfrac{1}{1 + 10s}$
5	320–400	$\dfrac{1}{10s(1 + 2.5s)}$

the simulations were chosen to illustrate the relative robustness and adaptivity of the methods. The models are given as Laplace transforms and the sampling interval was chosen in all cases to be 1 s.

Figure 2 shows the behaviour of a fixed digital PID regulator which was chosen to give reasonable if rather sluggish control of the initial plant. It was implemented in interacting PID form with numerator dynamics of $(1 + 10s + 25s^2)$ and with a gain of 12. For models 1 and 5 this controller gave acceptable results but its performance was poor for the other models with evident excessive gain. Despite integral action (with desaturation) offset is seen with models 3 and 4 due to persistent control saturation.

The first adaptive controller to be considered was the GMV approach of Clarke and Gawthrop (1975, 1979) using design transfer-functions:

$$P(q^{-1}) = (1 - 0.5q^{-1})/0.5 \quad \text{and}$$
$$Q(q^{-1}) = (1 - q^{-1})/(1 - 0.5q^{-1}),$$

with the "detuned model-reference" interpretation. This implicit self-tuner used a k-step-ahead predictor model with $2F(q^{-1})$ and $5\,G(q^{-1})$ parameters and with an adopted fixed value for k of 2. The detuned version of GMV was chosen as the dead-time was known to be varying and the use of $Q(q^{-1})$ makes the design less sensitive to the value of k. Note in particular that for models 2 and 3 the real dead-time is greater than in the two samples assumed.

The simulation results using GMV are shown in Fig. 3; reasonable if not particularly "tight" control was achieved for models 1, 3, 4 and 5. The weighting of control increments (as found in other cases) contributes to the overshoots in the step responses. The behaviour with model 2 was, however, less acceptable with poorly damped transients. Nevertheless, the adaptation mechanism worked well and the responses are certainly better than with the nonadaptive PID controller.

An increasingly popular method in adaptive

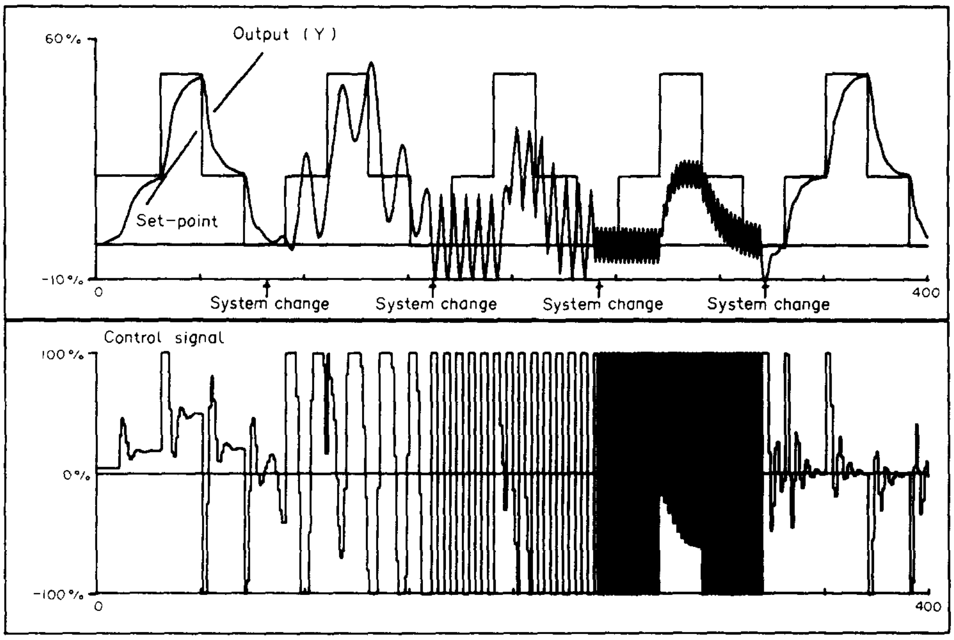

FIG. 2. The fixed PID controller with the variable plant.

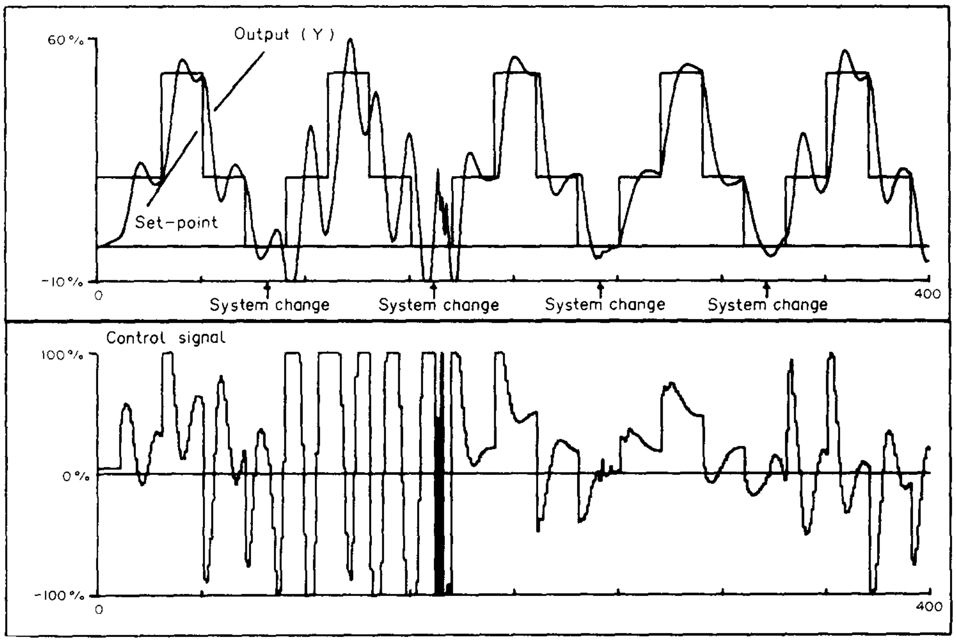

FIG. 3. The behaviour of the adaptive GMV controller.

control is pole-placement (Wellstead *et al.*, 1979) as variations in plant dead-time can be catered for using an augmented $B(q^{-1})$ polynomial. To cover the range of possibilities of dead-time here $2A(q^{-1})$ and $6B(q^{-1})$ parameters were estimated and the desired pole-position was specified by a polynomial $P(q^{-1}) = 1 - 0.5q^{-1}$ (i.e. like the GMV case without detuning). Figure 4 shows how an adaptive pole-placer which solves the corresponding Diophantine equation at each sample coped with the set of simulated models. For cases 1, 2 and 5 the output response is good with less overshoot than the GMV approach. For the first-order models 3 and 4, however, the algorithm is entirely ineffective due to singularities in the Diophantine solution. This verifies the observations that pole-placement is sensitive to model-order changes. In other cases, even though the response is good, the control is rather "jittery".

The GPC controller involved the same numbers of A and B (2 and 6) parameters as in the pole-placement simulation. Default settings of the output and control horizons were chosen with $N_1 = 1$, $N_2 = 10$ and $NU = 1$ throughout, as it has been found that this gives robust performance. Figure 5 shows the excellent behaviour achieved in all cases

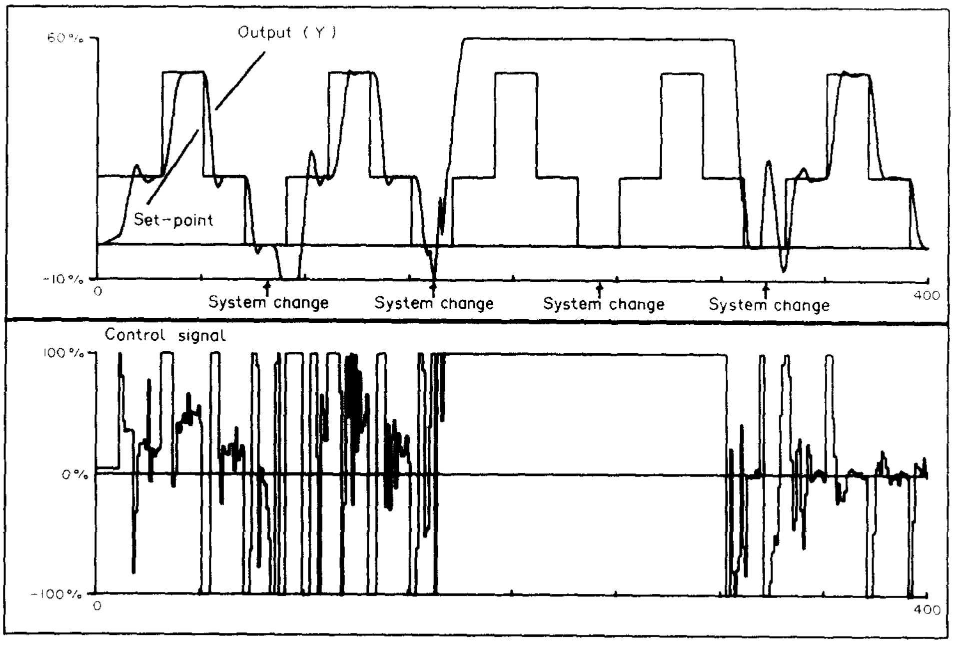

FIG. 4. The behaviour of the adaptive pole-placer.

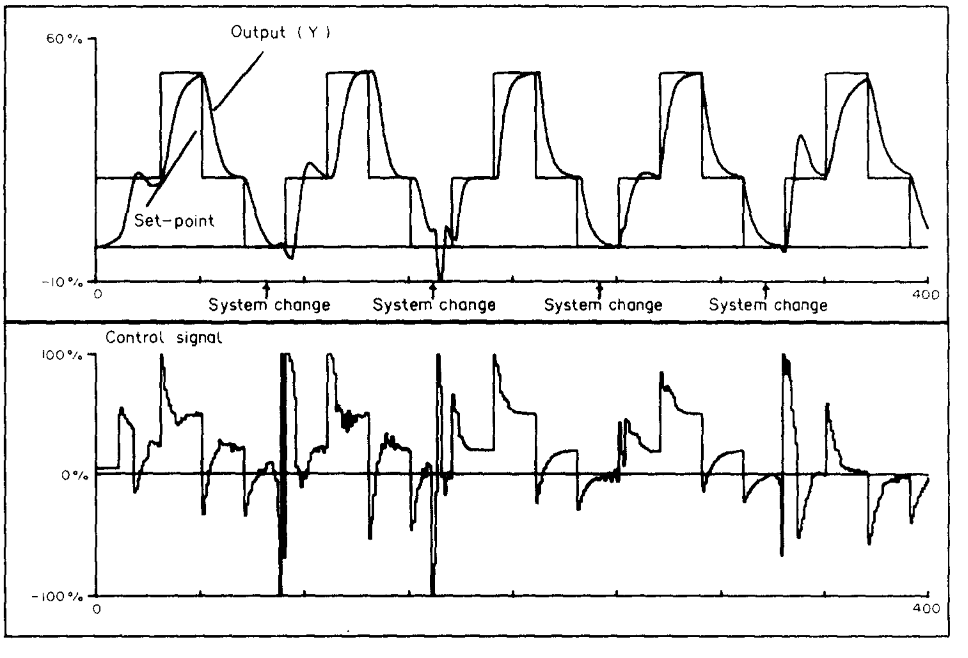

FIG. 5. The behaviour of the adaptive GPC algorithm.

by the GPC algorithm. For each new model only at most two steps in the set-point were required for full adaptation, but more importantly there is no sign of instability, unlike all the other controllers.

6. CONCLUSIONS

This paper has described a new robust algorithm which is suitable for challenging adaptive control applications. The method is simple to derive and to implement in a computer; indeed for short control horizons GPC can be mounted in a micro-computer. A simulation study shows that GPC is superior to currently accepted adaptive controllers when used on a plant which has large dynamic variations.

Montague and Morris (1985) report a comparative study of GPC, LQG, GMV and pole-placement algorithms for control of a heat-exchanger with variable dead-time (12–85 s, sampled at 6 s intervals) and for controlling the biomass of a penicillin fermentation process. The controllers were programmed on an IBM PC in pro-Fortran/proPascal. Their paper concludes that the GPC approach behaved consistently in practice and that "the LQG and GPC algorithms gave the best all-round performance", the GPC method being preferred for long time-delay processes, and it confirms that GPC is simple to implement and to use.

The GPC method is a further generalization of the well-known GMV approach, and so can be equipped with design polynomials and transfer-functions which have interpretations as in the GMV case. The companion paper Part II, which follows, explores these ideas and presents simulations which show how GPC can be used for more demanding control tasks.

REFERENCES

Åström, K. J. and B. Wittenmark (1973). On self-tuning regulators. *Automatica*, **9**, 185–199.

Åström, K. J. and B. Wittenmark (1980). Self-tuning controllers based on pole-zero placement. *Proc. IEE*, **127D**, 120–130.

Bertsekas, D. P. (1976). *Dynamic Programming and Stochatic Control*. Academic Press, New York.

Clarke, D. W. (1984). Self-tuning control of nonminimum-phase systems. *Automatica*, **20**, 501–517.

Clarke, D. W. and P. J. Gawthrop (1975). Self-tuning controller. *Proc. IEE*, **122**, 929–934.

Clarke, D. W. and P. J. Gawthrop (1979). Self-tuning control. *Proc. IEE*, **126**, 633–640.

Clarke, D. W. and L. Zhang (1985). Does long-range predictive control work? IEE Conference "Control 85", Cambridge.

Cutler, C. R. and B. L. Ramaker (1980). *Dynamic Matrix Control—A Computer Control Algorithm*. JACC, San Francisco.

De Keyser, R. M. C. and A. R. Van Cauwenberghe (1982). Typical application possibilities for self-tuning predictive control. *IFAC Symp. Ident. Syst. Param. Est.*, Washington.

De Keyser, R. M. C. and A. R. Van Cauwenberghe (1983). Microcomputer-controlled servo system based on self-adaptive long-range prediction. Advances in Measurement and Control—MECO 1983.

De Keyser, R. M. C. and A. R. Van Cauwenberghe (1985). Extended prediction self-adaptive control. *IFAC Symp. Ident. Syst. Param. Est.*, York.

Kailath, T. (1980). *Linear Systems*. Prentice-Hall, Englewood Cliffs, NJ.

Kurz, H. and W. Goedecke (1981). Digital parameter-adaptive control of processes with unknown dead time. *Automatica*, **17**, 245–252.

Montague, G. A. and A. J. Morris (1985). Application of adaptive control: a heat exchanger system and a penicillin fermentation process. Third workshop on the theory and application of self-tuning and adaptive control, Oxford, U.K.

Mosca, E., G. Zappa and C. Manfredi (1984). Multistep horizon self-tuning controllers: the MUSMAR approach. IFAC 9th World Congress, Budapest, Hungary.

Peterka, V. (1984). Predictor-based self-tuning control. *Automatica*, **20**, 39–50.

Richalet, J., A. Rault, J. L. Testud and J. Papon (1978). Model predictive heuristic control: applications to industrial processes. *Automatica*, **14**, 413–428.

Schweppe, F. C. (1973). *Uncertain Dynamic Systems*. Prentice Hall, Englewood Cliffs, NJ.

Tuffs, P. S. and D. W. Clarke (1985). Self-tuning control of offset: a unified approach. *Proc. IEE*, **132D**, 100–110.

Wellstead, P. E., D. Prager and P. Zanker (1979). Pole assignment self-tuning regulator. *Proc. IEE*, **126**, 781–787.

Ydstie, B. E. (1984). Extended horizon adaptive control. IFAC 9th World Congress, Budapest, Hungary.

APPENDIX A. MINIMIZATION PROPERTIES OF GPC

Consider the performance index

$$J = (\mathbf{y} - \mathbf{w})^T(\mathbf{y} - \mathbf{w}) + \lambda \tilde{\mathbf{u}}^T\tilde{\mathbf{u}} \quad \text{where } \mathbf{y} = \mathbf{G}\tilde{\mathbf{u}} + \mathbf{f} + \mathbf{e}$$

where $\mathbf{y}$, $\mathbf{w}$, $\tilde{\mathbf{u}}$ are as defined in the main text and

$$\mathbf{e} = [E_1\xi(t+1), E_2\xi(t+2), \ldots, E_N\xi(t+N)]^T.$$

Clearly as $\mathbf{e}$ is a stochastic process the expected value of the cost subject to appropriate conditions must be minimized. The cost to be minimized therefore becomes

$$J_1 = E\{(\mathbf{y} - \mathbf{w})^T(\mathbf{y} - \mathbf{w}) + \lambda \tilde{\mathbf{u}}^T\tilde{\mathbf{u}}\}.$$

Two different assumptions will be made in the next section about the class of admissible controllers for the minimization of the cost above. Only models of autoregressive type are considered:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \xi(t)/\Delta.$$

In the general case of CARIMA models where the disturbances are non-white the predictions $f(t+i)$ are only optimal asymptotically as they have poles at the zeros of the $C(q^{-1})$ and the effect of initial conditions is only eliminated as time tends to infinity. The closed-loop policy requires an optimal Kalman filter with time-varying filter gains, see Schweppe (1973); the parameters of the C-polynomial are only the gains of the steady-state Kalman filter once convergence is attained. Thus the closed-loop and the open-loop-feedback policies examined below are different in the case of coloured noise.

A.1. RELATION OF GPC AND OPEN-LOOP-FEEDBACK-OPTIMAL CONTROL

In this strategy it is assumed that the future control signals are independent of future measurements (i.e. all will be performed in open-loop) and we calculate the set $\tilde{\mathbf{u}}$ such that it minimizes the cost J_1. The first control in the sequence is applied and at the next sample the calculations are repeated. The cost becomes:

$$J_1 = E\{\tilde{\mathbf{u}}^T(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})\tilde{\mathbf{u}} + 2\tilde{\mathbf{u}}^T(\mathbf{G}^T(\mathbf{f} - \mathbf{w}) + \mathbf{G}^T\mathbf{e}) + \mathbf{f}^T\mathbf{f} + \mathbf{e}^T\mathbf{e} + \mathbf{w}^T\mathbf{w} + 2\mathbf{f}^T(\mathbf{e} - \mathbf{w}) - 2\mathbf{e}^T\mathbf{w}\}.$$

Because of the assumption above, $E\{\tilde{\mathbf{u}}^T\mathbf{G}^T\mathbf{e}\} = 0$; note also that the cost is only affected by the first two terms. By putting the first derivative of the cost equal to zero one obtains:

$$\tilde{\mathbf{u}} = (\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T(\mathbf{w} - \mathbf{f}).$$

A.2. RELATION OF GPC AND CLOSED-LOOP-FEEDBACK-OPTIMAL CONTROL

Choose the set $\tilde{\mathbf{u}}$ such that the cost J_1 is minimized subject to the condition that $\Delta u(i)$ is only dependent on data up to time i. This is the statement of causality. In order to find the set $\tilde{\mathbf{u}}$ assume that by some means $\{\Delta u(t), \Delta u(t+1), \ldots, \Delta u(t+N-2)\}$ is available.

$$\begin{aligned} J_1 = E\{&(\Delta u(t), \ldots, \\ &\Delta u(t+N-1))(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})(\Delta u(t), \ldots, \\ &\Delta u(t+N-1))^T \\ &+ 2\tilde{\mathbf{u}}^T(\mathbf{G}^T(\mathbf{f} - \mathbf{w})) \\ &+ 2(\Delta u(t), \ldots, \Delta u(t+N-1))\mathbf{G}^T\mathbf{e}\} \\ &+ \text{extra terms independent of minimization.} \end{aligned}$$

Note that in evaluating $\Delta u(t+N-1)$ the last row $\mathbf{G}^T\mathbf{e}$ is $g_0\sum e_i\xi(t+N-i)$ and $E\{\Delta u(t+N-1)\xi(t+N)\} = 0$ by the assumption of partial information. Therefore $\Delta u(t+N-1)$ can be calculated assuming $\xi(t+N) = 0$. From general state-space considerations it is evident that when minimizing a quadratic cost, assuming a linear plant model with additive noise, the resulting controller is a linear function of the available measurements (i.e. $u(t) = -\mathbf{k}\hat{\mathbf{x}}(t)$ where $\mathbf{k}$ is the feedback gain and $\hat{\mathbf{x}}(t)$ is the state estimate from the appropriate Kalman filter).

$$\begin{aligned} \Delta u(t+N-1) = \text{linear function}\,(&\Delta u(t+N-2), \\ &\Delta u(t+N-3), \ldots, \xi(t+N-1), \\ &\xi(t+N-2), \ldots, y(t), y(t-1), \ldots). \end{aligned}$$

In calculating $\Delta u(t + N - 2)$ the following expectations are zero:

$$E\{\Delta u(t + N - 2)\xi(t + N)\} = 0,$$
$$E\{\Delta u(t + N - 2)\xi(t + N - 1)\} = 0.$$

In order to work out $E\{\Delta u(t + N - 1)\Delta u(t + N - 2)\}$, the part of the expectation associated with $\xi(t + N - 1)$ is set to zero as the control signal $\Delta u(t + N - 2)$ cannot be a function of $\xi(t + N - 1)$. As $\Delta u(t + N - 1)$ is a linear function of $\xi(t + N - 1)$ this implies that as far as calculation of $\Delta u(t + N - 2)$ is concerned, $\Delta u(t + N - 1)$ could be calculated by assuming that $\xi(t + N - 1)$ and $\xi(t + N)$ were zero. Similarly, the same argument applies for the rest of control signals at each step i assuming $E\{\Delta u(t + i)\xi(t + j)\} = 0$ for $j > i$. This implies that each control signal $\Delta u(t + i)$ is the ith control in the sequence assuming that $\xi(t + j) = 0$ for $j > i$. Hence for $\Delta u(t)$ the solution amounts to calculating the first control signal setting $\xi(t + j)$ for $j > 0$ to zero—the same as that considered in part I for an Open-Loop-Feedback-Optimal controller. Note that in the case of $NU < N_2$ the minimization is performed subject to the constraint that the last $N_2 - NU + 1$ control signals are all equal. Consideration of causality of the controller implies that $\Delta u(t + NU - 1)$ is the first free signal that can be calculated as a function of data available up to time $t + NU - 1$ in the sequence and the argument above follows accordingly. Therefore, the GPC minimization (OLFO) is equivalent to the closed-loop policy for models of regression type.

APPENDIX B. AN EXAMPLE FOR A FIRST ORDER PROCESS

This appendix examines the relation of N_2 and the closed-loop pole position for a simple example. Consider a nonminimum-phase first-order process (first-order + fractional dead-time):

$$(1 + a_1q^{-1})y(t) = (b_0 + b_1q^{-1})u(t - 1)$$
$$(1 - 0.9q^{-1})y(t) = (1 + 2q^{-1})u(t - 1).$$

B.1. E AND F PARAMETERS

From the recurrence relationships in the main text it follows that:

$$e_i = 1 - a_1e_{i-1}; f_{i0} = 1 - f_{i1}; f_{i1} = e_ia_1$$
$$e_0 = 1{:}[1.0]; f_{10} = (1 - a_1){:}[1.9]; f_{11} = a_1{:}[-0.9]$$
$$e_1 = 1 - a_1{:}[1.9]; f_{20} = 1 - a_1(1 - a_1){:}[2.71];$$
$$f_{21} = a_1(1 - a_1){:}[-1.71]$$
$$e_2 = 1 - a_1(1 - a_1){:}[2.71];$$
$$f_{30} = 1 - a_1(1 - a_1(1 - a_1)){:}[3.439]$$
$$f_{31} = a_1(1 - a_1(1 - a_1)){:}[-2.439].$$

B.2. CONTROLLER PARAMETERS

$$g_0 = b_0{:}[1.0]; g_{11} = b_1{:}[2.0]$$
$$g_1 = b_0(1 - a_1) + b_1{:}[3.9]; g_{22} = b_1(1 - a_1){:}[3.8]$$
$$g_2 = b_0(1 - a_1(1 - a_1)) + b_1(1 - a_1){:}[6.51];$$
$$g_{33} = b_1(1 - a_1(1 - a_1)){:}[5.42].$$

Assuming $NU = 1$, the effect of N_2 on the control calculation and the closed-loop pole position is now examined:

$$\Delta u(t) = \left(\sum g_i(w - f_{i0}y(t - 1) - f_{i1}y(t - 2)\right)$$
$$- g_{ii}\Delta u(t - 1)\sum (g_i)^2.$$

For a controller of the form $R(q^{-1})\Delta u(t) = w(t) - S(q^{-1})y(t)$

$$s_0 = \sum g_if_{i0}/\sum g_i$$
$$s_1 = \sum g_if_{i1}/\sum g_i$$
$$r_0 = \sum (g_i)^2/\sum g_i$$
$$r_1 = \sum g_ig_{ii}/\sum g_i$$

and the closed-loop pole-positions are at the roots of $RA\Delta + q^{-1}SB$.

$N_2 = 1$

$$\Delta u(t) = [w - 1.9y(t) + 0.9y(t - 1) - 2\Delta u(t - 1)]$$
$$RA\Delta + q^{-1}SB \Rightarrow (1 + 2q^{-1}).$$

This is the minimum-prototype controller which, because of the cancellation of the nonminimum-phase zero, is unstable.

$N_2 = 2$ (*i.e. greater than deg*(B))

$$\Delta u(t) = [w - 1.9y(t) + 0.9y(t - 1) - 2\Delta u(t - 1)$$
$$+ 3.9(w - 2.71y(t) + 1.71y(t - 1)$$
$$- 3.8\Delta u(t - 1))]/16.21$$

or:

$$\Delta u(t) = [4.9w - 12.469y(t) + 7.569y(t - 1)$$
$$- 16.82\Delta u(t - 1)]/16.21$$

and:

$$AR\Delta + q^{-1}SB \Rightarrow (1 - 0.09q^{-1}).$$

The closed-loop pole is within the unit circle and therefore the closed-loop is stable. In fact it can easily be demonstrated that for all first order processes $N_2 > \delta B(q^{-1}) + k - 1$ stabilizes an open-loop stable plant, for $\text{sign}\left(\sum g_i\right) = \text{sign(D.C. gain)}$.

$N_2 = 3$

$$\Delta u(t) = [11.41w - 34.857y(t) + 23.447y(t - 1)$$
$$- 52.104\Delta u(t - 1)]/58.591$$

and:

$$AR\Delta + q^{-1}SB \Rightarrow (1 - 0.416q^{-1}).$$

The pole is again within the unit circle. Note that the closed-loop pole is tending towards the open-loop pole as N_2 increases (see Part II).