Automatica, Vol. 23, No. 2, pp. 149 160, 1987
Printed in Great Britain.

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

Generalized Predictive Control—Part II. Extensions and Interpretations*

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

By relating a novel predictive controller to LQ designs stability theorems are deduced, and extensions to Generalized Predictive Control give model following, state-dead-beat and pole-placement control objectives.

Key Words—Self-tuning control; predictive control; model-reference control; LQ control; dead-beat control; nonminimum-phase plant; variable dead-time.

Abstract—The original GMV self-tuner was later extended to provide a general framework which included feedforward compensation and user-chosen polynomials with detuned model-reference, optimal Smith predictor and load-disturbance tailoring objectives. This paper adds similar refinements to the GPC algorithm which are illustrated by a set of simulations. The relationship between GPC and LQ designs is investigated to show the computational advantage of the new approach. The roles of the output and control horizons are explored for processes with nonminimum-phase, unstable and variable dead-time models. The robustness of the GPC approach to model over- and under-parameterization and to fast sampling rates is demonstrated by further simulations. An appendix derives stability results showing that certain choices of control and output horizons in GPC lead to cheap LQ, "mean-level", state-dead-beat and pole-placement controllers.

1. INTRODUCTION

THE BASIC GPC method developed in Part I is a natural successor to the GMV algorithm of Clarke and Gawthrop (1975) in which a cost-function of the form:

$$J_{\mathrm{GMV}} = E\{(y(t+k) - w(t))^2 + \lambda u^2(t)|t\} \qquad (1)$$

was first defined and minimized. The cost of (1) is single-stage and so it is found that effective control depends on knowledge of the dead-time k of the plant and for nonminimum-phase plant stability requires a nonzero value of λ. The use of long-range prediction and a multi-stage cost in GPC overcomes the problem of stabilizing a nonminimum-phase plant with unknown or variable dead-time.

The relative importance of controller performance criteria varies with the application area. For example, in process control it is generally found that energetic control signals are undesirable, a slowly responding loop being prefered, and plant models are poorly specified in terms of dead-time and order as well as their transfer function parameters. Emphasis is therefore placed on robust and consistent performance despite variations in quantities such as dead-time and despite sustained load-disturbances. High-performance electromechanical systems tend to have well-understood models though often with lightly-damped poles, and the control requirement is for fast response, accepting the fact that the actuation might saturate. It is doubtful whether a single criterion as in (1) can deal with such a wide range of problems, so to create an effective general-purpose self-tuner it is essential to be able to adapt the basic approach by the incorporation of "tuning-knobs".

The GMV design was developed into a useful self-tuning algorithm by the addition of user-chosen transfer functions $P(q^{-1})$, $Q(q^{-1})$ and an observer polynomial $T(q^{-1})$. These time-domain performance-oriented "knobs" allow the engineer to tackle different control problems within the same overall scheme. For example, Gawthrop (1977) and Clarke and Gawthrop (1979) showed that model-following, detuned model-following, and optimal Smith prediction were interpretations which could be invoked as well as the original control weighting concept of (1). Clarke (1982) and Tuffs (1984) give further examples of the use of these polynomials in practice.

This paper introduces similar polynomials to GPC for specifying a desired closed-loop model and for tailoring the controlled responses to load disturbances, and the derivation of GPC is expanded to include the more general CARIMA model. The properties of these extensions are verified by simulations, which also show the robustness of the method to a range of practical problems such

* 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.

as model over- and under-parameterization. An appendix relates GPC with a state-space LQ control law and derives stability results for two uses of GPC: large output and control horizons N_2 and NU leading to a cheap LQ controller, and $NU = 1$ giving a "mean-level" controller for stable plants which need not be minimum-phase. Moreover, the relations between state-dead-beat control and deterministic pole-assignment with GPC are examined.

2. EXTENSIONS TO GENERALIZED PREDICTIVE CONTROL

2.1. *Model-following*: $P(q^{-1})$

Generalized Predicitive Control as described in Part I is based on minimization of a set of predicted system errors based on available input–output data, with some constraints placed on the projected control signals. It is possible to use an *auxiliary* function of the output as in the GMV development and consider the predicted system errors associated with this pseudo-output and the projected set-points.

Consider the auxiliary output:

$$\psi(t) = P(q^{-1})y(t) \text{ where}$$
$$P(q^{-1}) = Pn(q^{-1})/Pd(q^{-1}).$$

$P(q^{-1})$ is a transfer-function given by polynomials Pn and Pd in the backward shift operator q^{-1} with $P(1)$ set to unity to ensure offset-free control. The cost that the controller minimizes is the expectation subject to data available at time t of:

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

where:

$\psi(t)$	is $P(q^{-1})y(t)$;
N_1	is the minimum costing horizon;
N_2	is the maximum costing horizon, and
$\lambda(j)$	is a control-weighting sequence.

The prediction equations given in Part I must therefore be modified to forecast $\psi(t+j)$ instead of $y(t+j)$.

The locally-linearized plant model to be considered is based on an ARIMA representation of the disturbances:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1})\xi(t)/\Delta \quad \text{(2b)}$$

and again for simplicity of derivation it is assumed that $C(q^{-1}) = 1$.

Consider the Diophantine identity:

$$\frac{Pn}{Pd} = E_j A\Delta + q^{-j}\frac{F_j}{Pd} \quad j = 1, 2, \ldots \quad (3)$$

Multiplying (2b) by $q^j E_j \Delta$ and following the same route as in Part I, we obtain:

$$\hat{\psi}(t+j|t) = G_j \Delta u(t+j-1) + F_j y(t)/Pd(q^{-1}) \quad (4)$$
$$\text{where } G_j = E_j B.$$

The Diophantine recursion equations developed earlier are identical to those involved here but simply the starting point is different:

$$E_1 = \frac{Pn(0)}{Pd(0)}, \quad F_1 = q(Pn - E_1\tilde{A}), \quad \text{and } \tilde{A} = A\Delta Pd.$$

The transfer function $P(q^{-1})$ has two distinct interpretations depending on the particular application and on the control strategy envisaged. For process control, when dealing with a "simple" plant the primary design "knobs" are N_2 and NU. If the process output has a large overshoot to set-point and load changes, $P(q^{-1})$ can be used to penalize this overshoot. In high-performance applications NU is chosen larger than unity by an amount depending on the complexity of the plant. $P(q^{-1})$ can then be interpreted as the "approximate inverse closed-loop model" and when $\lambda = 0$ and $N_2 = NU \geqslant k$ the relationship is exact. For the noise-free case this implies that the closed-loop response to changes in $w(t)$ is given by:

$$y(t) \cong \frac{1}{P}w(t-k) = M(q^{-1})w(t-k)$$

where $M(q^{-1})$ is the user-chosen closed-loop model.

The model-following properties are detuned in cases where $NU < N_2$, but for large NU the change in performance is slight without the associated problem of nonminimum-phase zeros in the estimated plant which destabilize the usual model-reference controllers. It is possible to achieve model-following (as in many MRAC approaches) by using a prefilter $M(q^{-1})$, giving an intermediate set-point $w'(t) = M(q^{-1})w(t)$, and with a minimum-variance control regulating $y(t)$ about $w'(t)$. However, the disturbance rejection properties and the overall robustness are not then affected by the model as it simply operates on the set-point which is independent of variations within the loop. GPC in effect has an inverse series model whose output is affected by both $w(t)$ and the disturbances and so is a more practical approach.

2.2. *Coloured noise* $C(q^{-1})$ *and the design polynomial* $T(q^{-1})$

Most practical processes have more than one disturbance or noise source acting on them to give an effective plant model:

$$y(t) = \frac{B}{A}u(t-k) + \frac{C_1}{A\Delta}\xi_1(t) + \dots + \frac{C_n}{A\Delta}\xi_n(t).$$

The noise components can be combined into a single random sequence $\frac{C}{A\Delta}\xi(t)$ where the noise-colouring polynomial $C(q^{-1})$ has all of its roots within the unit circle provided that at least one noise element has nonzero-mean and is persistently exciting. Note that $C(q^{-1})$ is a time-invariant polynomial only if the individual noise variances σ_i^2 remain constant. However, with a typical industrial process this will rarely hold in practice, so successful identification of $C(q^{-1})$ is unlikely. If the structure of the variations cannot be estimated on-line, a design polynomial $T(q^{-1})$ can be used to represent prior knowledge about the process noise.

One interpretation of $T(q^{-1})$ is as a fixed observer for the prediction of future (pseudo-) outputs (Åström and Wittenmark, 1984). In particular, if $T(q^{-1}) = C(q^{-1})$ and the stochastic model of (2b) is valid, then the predictions are asymptotically optimal (minimum-variance) and the controller will minimize the variance of the output subject to the prespecified constraints on the input and output sequences. The following demonstrates the inclusion of $T(q^{-1})$ or $C(q^{-1})$ in the GPC control scheme with the full CARIMA model.

As before, defining a Diophantine identity:

$$T(q^{-1}) = E_jA\Delta + q^{-j}F_j \tag{5}$$

and proceeding in the usual manner we obtain:

$$T(q^{-1})\hat{y}(t+j|t) = G_j\Delta u(t+j-1) + F_jy(t)$$

or:

$$\hat{y}(t+j|t) = G_j\Delta u^f(t+j-1) + F_jy^f(t) \tag{6}$$

where "^{f}" denotes a quantity filtered by $1/T(q^{-1})$.

The constraints and the cost are in terms of $\Delta u(t+j)$ for $j = 0, 1, \dots$ rather than Δu^f and therefore the prediction equation must be modified. Consider the following:

$$G_j(q^{-1}) = G_j'(q^{-1})T(q^{-1}) + q^{-1}\Gamma_j(q^{-1}). \tag{7}$$

The coefficients of G' are those of G where the initial identity of (5) has $T = 1$. These coefficients together with those of Γ_j can be found by the recursion equation outlined in Appendix A. Combining (6) and (7) gives:

$$\hat{y}(t+j|t) = G_j'\Delta u(t+j-1) + \Gamma_j\Delta u^f(t-1) + F_jy^f(t). \tag{8}$$

The minimization procedure to provide the optimal control sequence is then as given in Part I.

The choice of $T(q^{-1})$ follows the procedure adopted in GMV designs (Clarke, 1982; Tuffs, 1984). If a constrained minimum-variance solution is required $C(q^{-1})$ must be estimated and T put equal to $\hat{C}$, but for most practical applications T can be taken as a fixed first-order polynomial where $1/T$ is a low-pass filter.

3. RELATION OF GPC WITH STATE-SPACE LQ DESIGN

Any linear controller may be implemented in a state-space framework by the appropriate state transformations and if the control scheme minimizes a quadratic cost Riccati iteration may be employed. Details of such a controller based on a CARIMA model representation are given elsewhere (Clarke *et al.*, 1985). In order to consider stability and numerical properties, however, inclusion of disturbances is not necessary, but the insights based on well-known state-space controller design can be applied to the GPC method. This is explored in detail in Appendix B, where in particular it is shown how various choices of control and output horizons lead to cheap L Q, "mean-level", state-dead-beat and pole-placement controllers.

Consider the plant:

$$A\Delta y(t) = B\Delta u(t-1).$$

Its state-space representation in observable canonical form may be written as:

$$\begin{aligned}\mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}\Delta u(t) + \tilde{\mathbf{w}}(t)\\ y(t) &= \mathbf{c}^T\mathbf{x}(t) + \mathbf{w}(t).\end{aligned} \tag{9}$$

The matrix $\mathbf{A}$ and the vectors $\mathbf{b}$, $\mathbf{c}$ are defined in Appendix B. The vector $\tilde{\mathbf{w}}$ is associated with the set-point (see Clarke *et al.*, 1985). For this model the multi-stage cost of GPC becomes:

$$J = \mathbf{x}(t+N_2)^T\mathbf{x}(t+N_2) + \sum_{i=t}^{N_2+t-1}[\mathbf{x}(i)^T\mathbf{Q}\mathbf{x}(i) + \lambda(i)\Delta u(i)^2]. \tag{10}$$

The control giving minimum cost is therefore:

$$\begin{aligned}\Delta u(t) &= \mathbf{k}^T\hat{\mathbf{x}}(t|t)\\ \mathbf{k}^T &= (\lambda(t) + \mathbf{b}^T\mathbf{P}(t)\mathbf{b})^{-1}\mathbf{b}^T\mathbf{P}(t)\mathbf{A}\end{aligned}$$

where $\mathbf{P}(t)$ is given by the backwards iteration of the following Riccati equation starting from the terminal covariance $\mathbf{Q}$:

$$\mathbf{P}^*(i) = \mathbf{P}(i+1) - \mathbf{P}(i+1)\mathbf{b}(\lambda(i) + \mathbf{b}^T\mathbf{P}(i+1)\mathbf{b})^{-1}\mathbf{b}^T\mathbf{P}(i+1) \tag{11}$$

$$\mathbf{P}(i) = \mathbf{A}^T\mathbf{P}^*(i)\mathbf{A} + \mathbf{Q}$$

$$\mathbf{Q} = [1, 0, 0, \ldots, 0]^T[1, 0, 0, \ldots, 0]. \tag{12}$$

For the cases where $NU < N_2$ (i.e. when some of the control increments in the future are assumed to be zero), the value of $\lambda(i)$ is time-varying, as fixing the control signal is equivalent to employing a very large penalty on the particular control increment. In this way all combinations of GPC may be implemented in a state-space framework. Recall, however, that because the states are not accessible a state-observer or state-reconstruction scheme must be employed. Lam (1980) uses a transmittance matrix and his approach was employed by Clarke *et al.* (1985) in their LQ self-tuner. Three points are of relevance.

(i) The stability properties of GPC and the deterministic LQ method with a finite horizon of predictions are identical. Appendix B examines the stability properties for special cases of the GPC settings.

(ii) It is known that numerical properties of LQ in the state-space formulation are good though its drawback is in execution time. The GPC approach requires the inversion of $(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})$ which can be done using UDU factorization. Vectors $\mathbf{f}$ and $\mathbf{w}$ need never be formed since the multiplication of $\mathbf{G}^T(\mathbf{w} - \mathbf{f})$ can also be implemented recursively at the same time as that of the G and F parameters. In the LQ case the measurement-update equation (11) can be done using UDU and (12) can be done via a modified weighted Gram–Schmidt algorithm. For large NU both methods required NU iterations of a UDU algorithm. Assuming the time-updates of (12) are equivalent to calculating Gs and Fs in the GPC approach, the burden of computing the feedback gains will be approximately equal. However, with GPC the top row of $(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}$ is simply multiplied by the vector $\mathbf{G}^T(\mathbf{w} - \mathbf{f})$. In the LQ case typically three calls to the transmittance matrix routines are required to obtain state estimates which are avoided by GPC. For $NU = 1$ inversion is a scalar calculation whilst for LQ matrix operations in addition to state estimation are required. Hence the "one shot" algorithm of GPC is computationally less demanding.

(iii) The Riccati equation implementation implicitly assumes that there are no rate or amplitude limits on the control signal. With GPC it is possible to perform a constrained optimization which includes these limits.

4. SIMULATION EXAMPLES

Simulations were performed to demonstrate the effect of the design features of GPC using a self-tuning control package FAUST (Tuffs and Clarke, 1985). Two principal types were undertaken: in one case the plant was constant and the exercise was intended to show the effect of changing one of the design "knobs" on the transient response, whilst the second set involved a time-varying plant and the objective was to show the robustness of the adaptive use of GPC. Some simulations used continuous-time models to illustrate the effect of sampling, though in all cases the estimated model used in the self-tuned version of GPC was in the equivalent discrete-time form.

The parameters of the A and B polynomials were estimated by a standard UDU version of RLS (Bierman, 1977) using the incremental model:

$$y^f(t) = y^f(t-1) + q(1 - A(q^{-1}))\Delta y^f(t) + B(q^{-1})\Delta u^f(t) + D(q^{-1})\Delta v^f(t) + \varepsilon(t)$$

where "f" denotes signals filtered by $1/T(q^{-1})$, if used. For simplicity of exposition a fixed forgetting-factor was adopted whose value was normally one (no forgetting), unless otherwise stated. The signal $v(t)$ is a measured disturbance signal (feedforward) which, as with GMV, can readily be added to GPC. The parameter estimates were initialized in the simulations with $\hat{b}_0$ equal to one and the rest equal to zero and with the covariance matrix set to diag$\{10\}$.

The figures consist of two sets of graphs covering the behaviour over 400 or 800 samples, one showing the set-point $w(t)$ together with the plant output $y(t)$, and the other showing the control signal $u(t)$ and possibly a feedforward signal $v(t)$. The scales for each graph are shown on the axes; in all examples the control was limited to lie in the range $[-100, 100]$. Load-disturbances were of two principal types, one (called "*dcu*") consisted of steps in $d(t)$ for the model:

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

and the other ("*dcy*") of steps in $d(t)$ in the model:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + A(q^{-1})d(t).$$

The *dcy* disturbance, equivalent to an additive step on the plant output, is a particularly severe test of an adaptive algorithm.

4.1. *The effect of* $P(q^{-1})$

These simulations were designed to demonstrate the uses of P: penalizing overshoot (process control) and model-following (high-performance role).

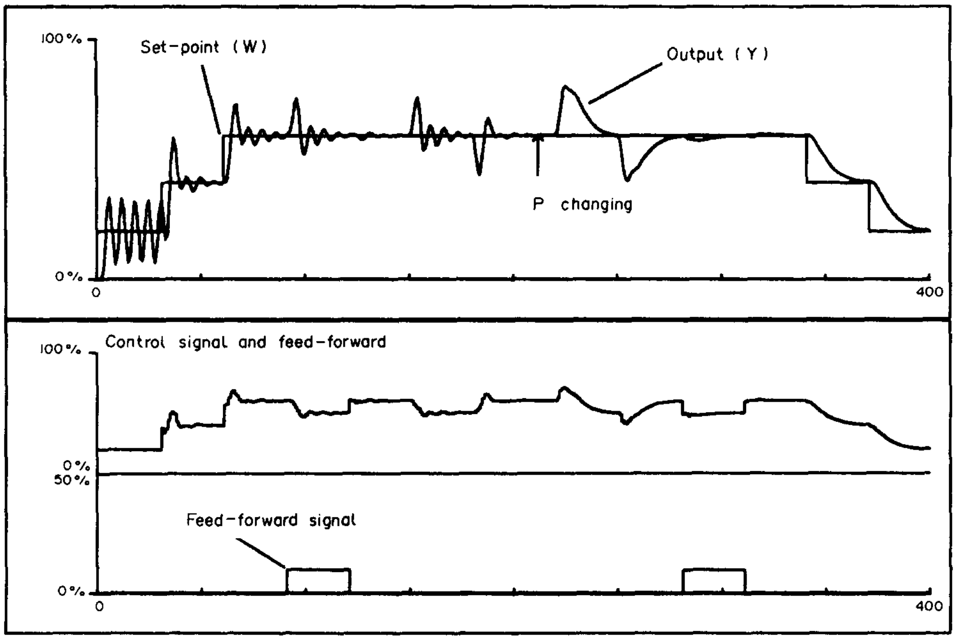

FIG. 1. The effect of the P transfer-function on closed-loop performance.

The first plant simulated was the third-order oscillator:

$$(1 + s)(1 + s^2)y(t) = u(t) + d(t) + v(t),$$

where s is the differential operator $\frac{d}{dt}$, sampled at 1 s intervals. Also, $3\,A(q^{-1})$, $3\,B(q^{-1})$ and $3\,D(q^{-1})$ (feed-forward) parameters were estimated, with an assumed delay of unity. The horizon N_2 was chosen as 10 samples and $NU = 1$. The control signal was fixed at 20 units for the first 30 samples while the estimator was enabled. Two set-point changes of 20 units each were applied at intervals of 30 samples. Unmeasurable step load-disturbances dcu of ± 10 units were added at $t = 150$, 180 and $t = 210$, 240. Measurable step load-disturbances $v(t)$ of the same size were added at $t = 90$, 120 and $t = 280$, 310. $P(q^{-1})$ was initially set to unity. As seen in Fig. 1, the set-point response and the load-disturbance rejection although stable, had excessive overshoot, but once the feed-forward parameters were tuned the feed-forward disturbance rejection was almost exact. Note that in order to include feed-forward in the prediction equations a model of the form:

$$\begin{aligned} A(q^{-1})\Delta y(t) = {} & B(q^{-1})\Delta u(t-1) \\ & + D(q^{-1})\Delta v(t-1) + \xi(t) \end{aligned}$$

is assumed where $\Delta v(t + j) = 0$ for $N_2 > j > 0$.

For the second half of the simulation $P(q^{-1})$ was set to $(1 - 0.8q^{-1})/0.2$ at $t = 190$. Both the disturbance rejection and the set-point responses were thereby detuned and the overshoot removed altogether. Feed-forward rejection, on the other hand, was unaffected by the change made in P.

For the second example shown in Fig. 2, consider the double-oscillator plant whose transfer function is given by:

$$(1 + s^2)(1 + 1.5s^2)y(t) = u(t).$$

The model chosen had $Pn(q^{-1}) = (1 - 0.5q^{-1})^2$, and $Pd(q^{-1})$ set to $0.2778(1 - 0.1q^{-1})$. A constant control signal of one unit was applied for the first 10 samples after which it was set to zero for 70 samples. The estimator was enabled from the begining of the run, whereas the controller was switched in the loop at the 80th sample and the set-point was also changed to 20 units at that instant. NU was set to two at the start of the simulation and as seen this choice did not give good model-following characteristics although the closed-loop was stable. NU was set to four at the downward set-point change to 20 units and at each subsequent downward set-point change to 20 units NU was incremented by two. Clearly the control ringing increased with the increase in NU but the designed model-following closed-loop performance remained almost the same for $NU > 4$ (the number of plant poles). If desired, the control modes can be damped using a nonzero value of λ to give "fine-tuning" as in GMV.

4.2. *The effect of* $T(q^{-1})$

A second-order plant with time-delay was simulated where:

$$\begin{aligned} (1 + 15s + 50s^2)y(t) = {} & e^{-2s}u(t) + 10d(t) \\ & + (1 + 15s + 50s^2)dcy(t) + (1 + 15s + 50s^2)\xi(t) \end{aligned}$$

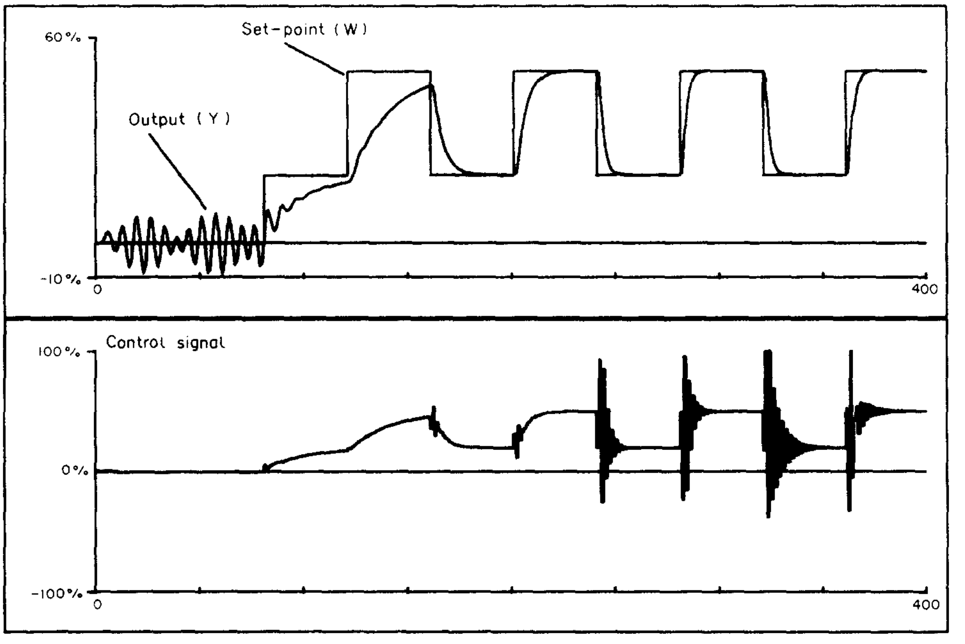

FIG. 2. (Detuned) model-following using $P(q^{-1})$.

in which $\xi(t)$ was an uncorrelated random sequence with zero mean and RMS value of two units for $210 < t < 240$. A step-disturbance dcy of three units was added to the output at $150 < t < 180$. Another step-disturbance dcu, exciting all of the modes of the plant, of ± 10 units was added at $120 < t < 150$ and $270 < t < 300$. Two step changes in set-point at the end of a period of regulation were employed to see the effect of disturbances on the servo performance of the controller, and three A and four B parameters were estimated with an assumed delay of unity.

The initial set-point response and subsequent load-disturbance rejection were good, as seen in Fig. 3. The rejection of dcy, on the other hand, was very active initially and inconsistent in the second change of load. This was due to dynamic parameter changes caused by not estimating parameters associated with the noise structure. Subsequent behaviour based on the poor model was not very good as the control was far too active.

In the second simulation shown in Fig. 4, $T(q^{-1})$ was chosen to be $(1 - 0.8q^{-1})$. Note that although T improved the disturbance rejection of the closed-loop it had no effect on the set-point response. In addition, since the parameter estimator was better conditioned in the second case, the final set-point responses were almost identical to the initial ones.

4.3. *The effect of* N_2 *and the sampling period*

One of the major criticisms of digital controllers is that most designs only work well if the sampling period is chosen carefully (approximately 1/4 to 1/10 of the settling-time of the plant). A slow plant with three real poles was chosen to investigate whether GPC suffers from this problem:

$$(1 + 10s)^3 y(t) = u(t).$$

Five A and five B parameters were estimated; N_1 was chosen to be 1 and N_2 was initially set to 10 but doubled at every upward-going step in $w(t)$ to 50. A sampling time of 1 s was chosen; note that the settling time of the plant is about 160 s. Figure 5 shows that the initial control was stable but had a small ringing mode and attained the imposed saturation limits. When an output horizon of 20 samples was chosen this mode was removed. At $N_2 = 40$ (the rise-time of the plant) the control was much smoother. Increasing the horizon of output prediction to the settling time of the plant ($N_2 = 160$) caused the speed of the closed-loop under this condition to be almost the same as that of the open-loop, verifying the "mean-level" theory of Appendix B. In all cases, then, the responses were smooth despite the rapid sampling.

4.4. *Over-parameterization*

One of the problems with many adaptive control schemes is that an exact knowledge of the model order is required; of particular interest is the ability to over-parameterize the plant parameter estimator in order to model the plant well in case of dynamic changes.

A first-order plant was simulated in discrete time:

$$(1 - 0.9q^{-1})y(t) = u(t - 1).$$

Estimation was disabled and the parameters were fixed *a priori* to the required values given below.

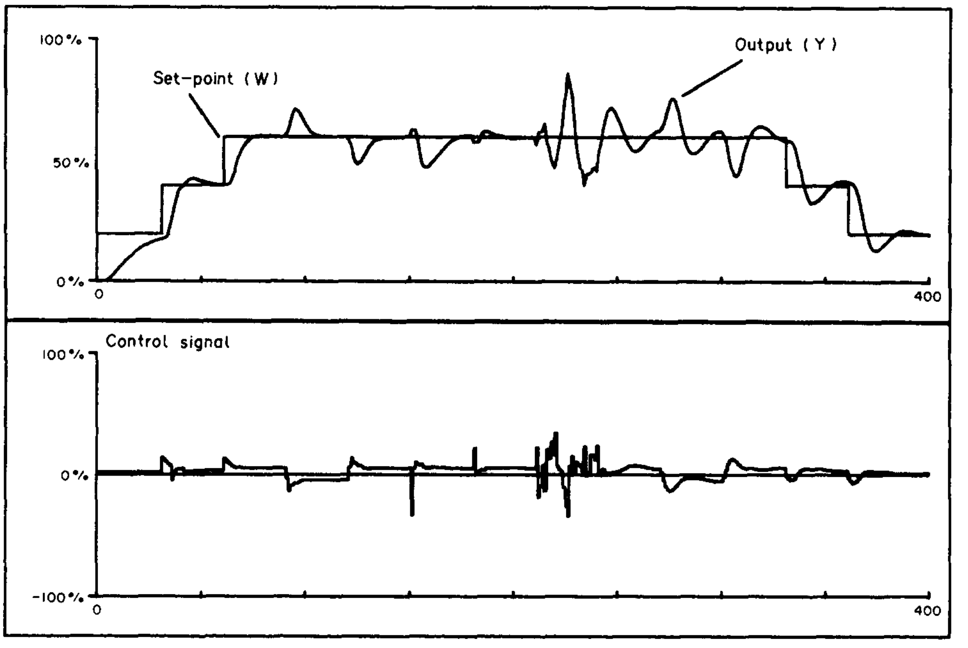

FIG. 3. The control of a plant with additive disturbances (without the T polynomial).

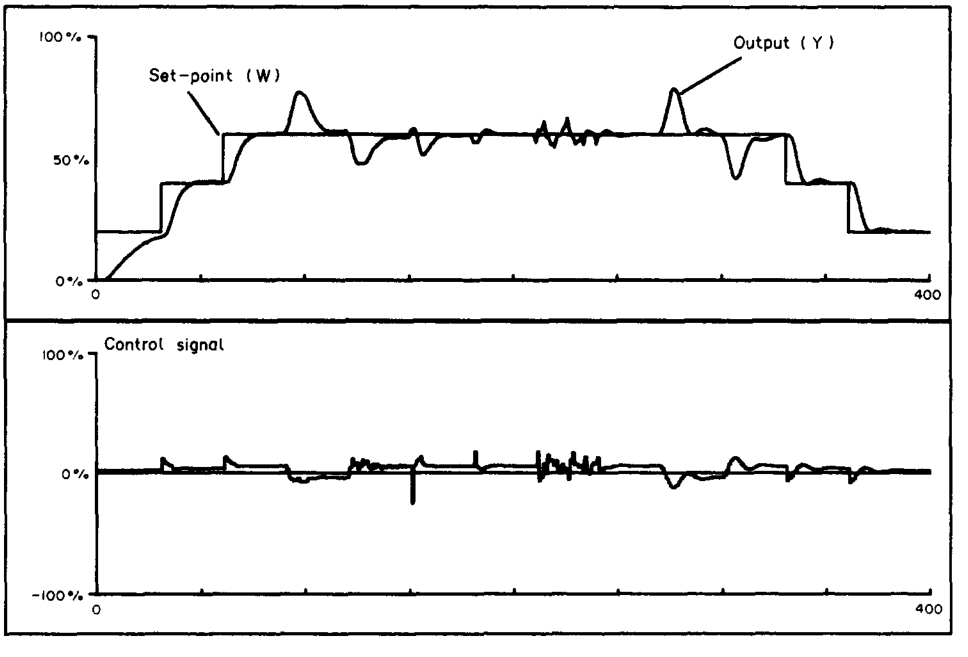

FIG. 4. The control of a plant with additive disturbances (with the T polynomial).

Initially a common factor of $(1 + 2q^{-1})$ was set between the estimated A and B polynomials, giving:

$$\hat{A}(q^{-1}) = 1 + 1.1q^{-1} - 1.8q^{-2}$$

$$\hat{B}(q^{-1}) = 1 + 2q^{-1}.$$

NU was set to one at the set-point change from 0 to 20, to two for the change 20–40, four for the change 40–20 and finally $NU = 10$ for the change 20–0. The common root was then moved to -0.5, 0.5 and 2 in succession and the transient test repeated. Figure 6 shows that in all cases the control performance of GPC was unaffected by the common factor.

4.5. *Under-parameterization*

Most industrial processes are nonlinear and therefore may only be approximated by high-order linear models. A good choice of sample-rate, on the other hand, enables the designer to use low-order models for control: slow sampling masks the high-order fast dynamics.

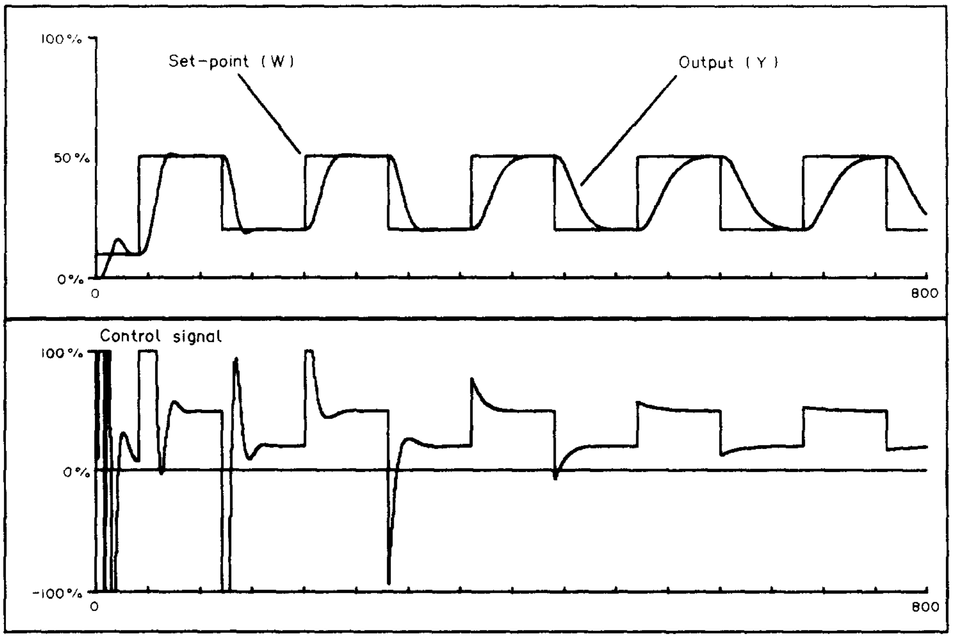

FIG. 5. The effect of fast sampling and the prediction horizon.

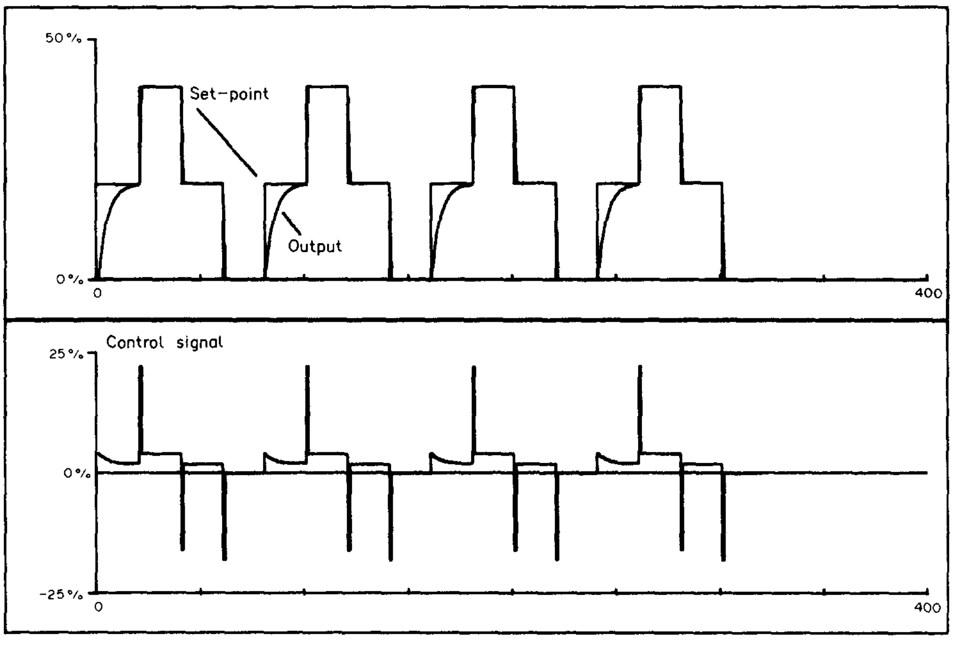

FIG. 6. The effect of common factors in the estimated parameters.

Consider the fourth-order plant:

$$(1 + s)^2(1 + 3s)^2 y(t) = u(t) + d(t).$$

A second-order model was assumed and the plant was sampled at 1 s intervals; note that the sampling process was *not* masking the slightly faster poles. In this case two A and three B parameters were estimated and the assumed time-delay was unity. N_2 was set to 10 and NU was set to one. The set-point sequence was a square wave with a period of 40 samples. Load-disturbances of 10 and 20 units were added at the marked times and as shown in Fig. 7, offset-free control was achieved. The overall performance was good despite the wrong parameterization; the overshoot could have been reduced using $P(q^{-1})$ as shown in the previous sections.

4.6. *Unknown or variable time-delay*

The GMV design is sensitive to choice of dead-time; GPC is however robust provided that δB is chosen to absorb any change in the time-delay. Consider the plant:

$$(1 - 1.1q^{-1})y(t) = -(0.1 + 0.2q^{-1})u(t - k)$$

where $k = 1, 2, 3, 4, 5$ at different stages in the trial.

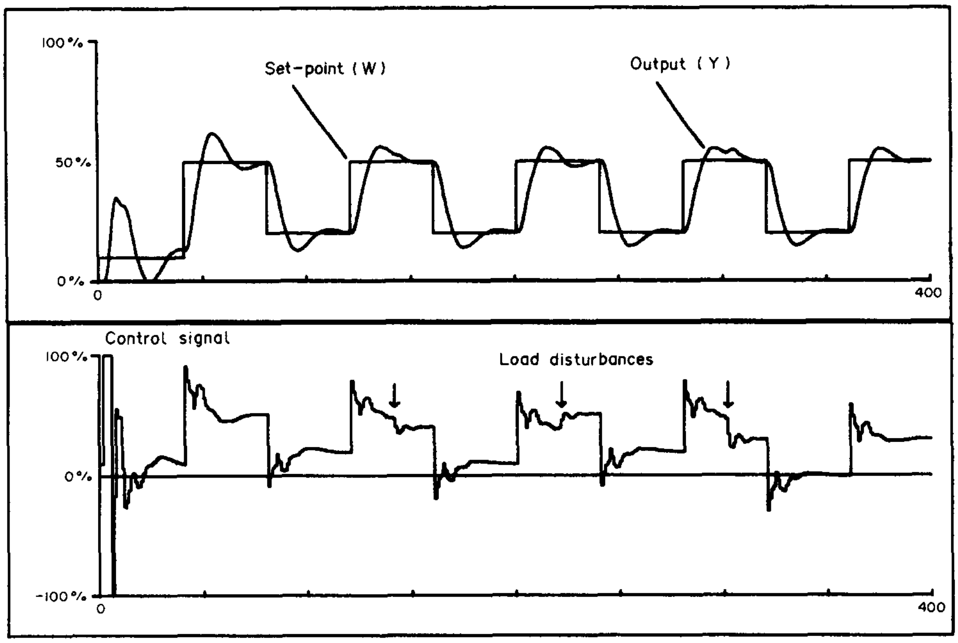

FIG. 7. The effect of under-parameterization.

The value of k was changed at the downward-going steps in set-point increasing initially from one to five and then decreasing from five back to one again. The adaptive controller estimated two A and six B parameters and a scalar forgetting-factor of 0.9 was employed to enable tracking of variations in the dead-time. N_2 was set to 10 and NU to one. The performance of GPC shown in Fig. 8 is good; note that the plant was both nonminimum-phase and open-loop unstable with variable dead-time, yet stable control was achieved with the default settings of this algorithm.

5. CONCLUSIONS

This paper has shown that GPC can be equipped with the design features of the well-known GMV approach and given a wide range of possible control objectives, which can be interpreted by its relationship with LQ algorithms based on state-space models. These results are summarized in Table 1. It might seem that there are many possible choices of design parameters in GPC, but the table shows that many combinations lead to well-understood control laws. In practice not all this flexibility would be required and many processes can be effectively controlled using default settings. Closer inspection of Table 1 shows that a "large" value of N_2 is generally recommended and that NU and P can then be chosen according to the control philosophy appropriate for the plant and the computing power available. Hence the "knobs" can be used to tailor an adaptive controller to precise specifications, which is of great value in the high-performance role. In particular, the method no longer needs to employ control weighting when applied to a varying dead-time plant which is a requirement of GMV designs.

The simulations show that GPC can cope with the control of complex processes under realistic conditions. As it is relatively insensitive to basic assumptions (model order, etc.) about the process, GPC can be easily applied in practice without a prolonged design phase. These features ensure that the method provides an effective approach to the adaptive control of an industrial plant.

REFERENCES

Åström, K. J. and B. Wittenmark, (1984). *Computer Controlled Systems—Theory and Design*. Prentice-Hall, Englewood Cliffs, NJ.

Bierman, G. J. (1977). *Factorization Methods for Discrete System Estimation*. Academic Press, New York.

Clarke, D. W. (1982). The application of self-tuning control. *Trans. Inst. M.C.*, **5**, 59–69.

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., P. P. Kanjilal and C. Mohtadi, (1985). A generalised LQG approach to self-tuning control. *Int. J. Control*, **41**, 1509–1544.

Gawthrop, P. J. (1977). Some interpretations of the self-tuning controller. *Proc. IEE*, **124**, 889–894.

Kwakernaak, H. and R. Sivan, (1972). *Linear Optimal Control Systems*. Wiley, New York.

Lam, K. P. (1980). Implicit and explicit self-tuning controllers. D. Phil Thesis, Oxford University.

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

Tuffs, P. S. (1984). Self-tuning control: algorithms and applications. D. Phil. Thesis, Oxford University.

Tuffs, P. S. and D. W. Clarke, (1985). FAUST: a software package for self-tuning control. *IEE Conf. "Control 85"*, Cambridge.

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

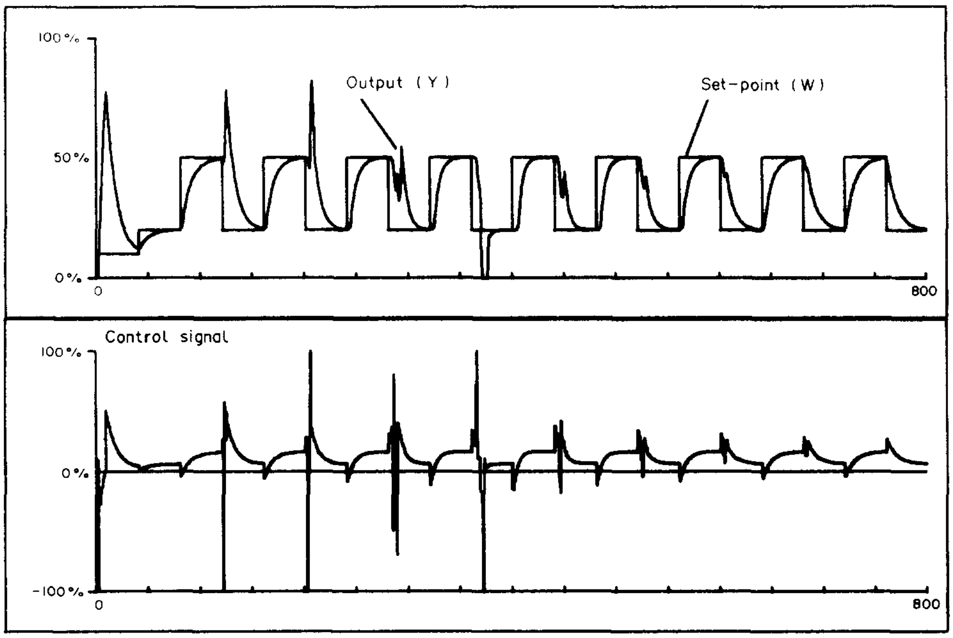

FIG. 8. The control of a variable dead-time plant.

TABLE 1. SPECIAL CASES OF GPC SETTINGS

NU	N_1	N_2	P	λ	Plant	Controller
1	1	10	1	0	s,d	"Default"
1	1	$\to\infty$	1	0	s,d	"Mean-level"
N_2	1	$\geqslant k$	P	0	mp	Exact model-following $P = 1/M$
$<N_2$	1	$\geqslant k$	P	$0,\lambda$		"Detuned" model-following
N_2	1	$\to\infty$	1	>0	s,d	LQ infinite-stage
$N_2 - n + 1$	1	$\to\infty$	1	0	s,d	Cheap LQ
n	n	$\geqslant 2n - 1$	1	0	o,c	State-dead-beat
n	n	$\geqslant 2n - 1$	P	0	o,c	Pole-assignment
				λ	s,d	"Detuned" pole-assignment

s: stabilizable; d: detectable; o: observable; c: controllable; mp: minimum-phase.

APPENDIX A. RECURSION OF THE POLYNOMIAL $G'(q^{-1})$

Consider the successive Diophantine identities:

$$G_j = G_j'T + q^{-j}\Gamma_j \tag{A.1}$$

$$G_{j+1} = G_{j+1}'T + q^{-j-1}\Gamma_{j+1}. \tag{A.2}$$

Note that $G_j = E_jB$, where:

$$E_j(q^{-1}) = e_0 + e_1q^{-1} + \ldots + e_{j-1}q^{-j+1}.$$

Subtracting equations (A.1) from (A.2) we obtain:

$$q^{-j}e_jB = q^{-j}g_{j+1}'T + q^{-j}(q^{-1}\Gamma_{j+1} - \Gamma_j). \tag{A.3}$$

Hence the update equations become:

$$g_{j+1}' = 1/t_0(e_jb_0 + \gamma_{(j)0}) \tag{A.4}$$

and, for $i = 1$ to $\max(\delta B, \delta T)$:

$$\gamma_{(j+1)i} = \gamma_{(j)i} + e_jb_i + g_{j+1}'t_i \tag{A.5}$$

where $\gamma_{(j)i}$ denotes the ith coefficient of the polynomial Γ_j associated with q^{-1}. Note that the coefficients of B or T with indices greater than their respective degrees are zero.

APPENDIX B. SOME STABILITY RESULTS FOR LIMITING CASES OF GPC

Consider the plant given in shift-operator form by:

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

A state-space model of this plant can be written as:

$$\mathbf{x}(t + 1) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}\Delta u(t) \tag{B.2}$$

$$y(t) = \mathbf{c}^{\mathrm{T}}\mathbf{x}(t) \tag{B.3}$$

where $\mathbf{A}$ is the state transition matrix which we take to be in observable canonical form, $\mathbf{b}$ is the vector of B parameters and $A\Delta$ polynomials (Clarke et al., 1985). Since disturbances do not affect the stability properties of the controller, only deterministic elements are considered in the following section. Defining the augmented polynomial $\tilde{A}$ to be:

$$\tilde{A}(q^{-1}) = A\Delta = 1 + \tilde{a}_1q^{-1} + \tilde{a}_2q^{-2} + \ldots + \tilde{a}_nq^{-n}$$

then the matrices and vectors in the state-space form are:

$$\mathbf{A} = \begin{bmatrix} -\tilde{a}_1 & 1 & 0 & \ldots & 0 \\ -\tilde{a}_2 & 0 & 1 & \ldots & 0 \\ : & : & : & : & 0 \\ . & . & . & . & . \\ . & . & . & . & 1 \\ -\tilde{a}_n & . & . & . & 0 \end{bmatrix}$$

$$\mathbf{b} = [b_0, b_1, \ldots, b_{n-1}]^{\mathrm{T}}$$

where $\tilde{a}_i = 0$ for $i > \deg(\tilde{A})$ and $b_i = 0$ for $i > \deg(B)$.

The cost-function in the state-space formulation can be written as:

$$J = \sum_{i=1}^{N_2} [\mathbf{x}(t+i-1)^T \mathbf{Q} \mathbf{x}(t+i-1) + \lambda(t+i-1)\Delta u(t+i-1)^2] \quad (B.4)$$

where $\mathbf{Q} = [1,0,0,\ldots,0]^T[1,0,0,\ldots,0]$. The set-point has been omitted for simplicity because the stability properties are independent of inputs.

The solution is obtained by iterating the equations below.

Measurement update:

$$\mathbf{P}^*(i) = \mathbf{P}(i+1) - \mathbf{P}(i+1)\mathbf{b}(\lambda(i) + \mathbf{b}^T\mathbf{P}(i+1)\mathbf{b})^{-1}\mathbf{b}^T\mathbf{P}(i+1). \quad (B.5)$$

Time update:

$$\mathbf{P}(i) = \mathbf{Q} + \mathbf{A}^T\mathbf{P}^*(i)\mathbf{A} \quad (B.6)$$

$$\Delta u(t) = -(\lambda(t) + \mathbf{b}^T\mathbf{P}(t)\mathbf{b})^{-1}\mathbf{b}^T\mathbf{P}(t)\mathbf{A}\mathbf{x}(t). \quad (B.7)$$

$\mathbf{P}$ is called the "covariance matrix" (from duality with the estimation equations). N_2 iterations are performed backwards starting from $\mathbf{Q}$, the terminal covariance matrix. Note that since both the "one-shot" (GPC) method of cost minimization and the dynamic programming approach of state-space (LQ) succeed in minimizing the same cost under certain conditions (i.e. linear plant and no constraints on the control signal), the resulting control law must be the same because there is only one minimum and so their stability characteristics must be identical.

Note that fixing the projected control signal in the future is equivalent to employing a large penalty on the appropriate increment (i.e. $\lambda(i) \to \infty$). This means that the measurement update need not be performed for the particular value of i. Three special cases of GPC are considered below.

Theorem 1. The closed-loop system is stable if the system $(\mathbf{A}, \mathbf{b}, \mathbf{c})$ is stabilizable and detectable and if:

(i) $N_2 \to \infty$, $NU = N_2$ and $\lambda(i) > 0$ or

(ii) $N_2 \to \infty$, $NU \to \infty$ where $NU \leqslant N_2 - n + 1$ and $\lambda(i) = 0$.

Proof. Part (i) is easily proven from the stability conditions of the state-space LQ controller. The cost of (B.4) tends to the infinite stage cost and for convergence to the algebraic Riccati equation (ARE) solution. $\mathbf{Q}$ can be positive semi-definite if λ is positive definite for all terminal covariances (Kwakernaak and Sivan, 1972). For part (ii), note that for the first $n-1$ iterations of the Riccati equation only the time-updates (B.6) are necessary ($\lambda(i) \to \infty$). Note, moreover, that the terminal covariance $\mathbf{Q}$ is of rank one. Each of the time-updates increases the rank of the matrix $\mathbf{P}$ by one and after $n-1$ iterations the matrix $\mathbf{P}(N_2 - n + 1)$ is of full rank. It is seen that $\mathbf{P}(N_2 - n + 1)$ is $\Sigma \mathbf{A}^{iT}\mathbf{c}\mathbf{c}^T\mathbf{A}^i$—the observability Grammian which is guaranteed positive definite for the structure assumed. Then as $NU \to \infty$ the iterations (B.5, B.6) converge to the ARE solution for all values $\lambda(i) = \lambda \geqslant 0$.

Remark 1. Part (ii) is a special case of the stabilizing controller of Peterka (1984), using $\lambda(i) = \omega_s$ where $0 < \omega_s < \infty$.

Remark 2. For $\lambda \to 0$ the GPC laws above are equivalent to the constrained minimum-variance regulator derived by spectral factorization for noise models of regression type.

Theorem 2. For open-loop stable processes the closed-loop is stable and the control tends to a mean-level law for $NU = 1$ and $\lambda(i) = 0$ as $N_2 \to \infty$.

Proof. For simplicity assume that the matrix $\mathbf{A}$ has distinct eigenvalues and can therefore be written as:

$$\mathbf{A} = \Sigma \lambda_i \mathbf{q}_i \mathbf{r}_i^T \quad (B.8)$$

where λ_i are the eigenvalues and $|\lambda_i| < 1$ for all $i \neq 1$ and $\lambda_1 = 1$ and $\mathbf{q}_i$ and $\mathbf{r}_i$ are right and left eigenvectors associated with the particular eigenvalue of $\mathbf{A}$.

The right and left eigenvectors associated with the eigenvalue at 1 are given by:

$$\mathbf{q}_1^T = [1, 1 + \tilde{a}_1, 1 + \tilde{a}_1 + \tilde{a}_2, \ldots]$$
$$\mathbf{r}_1^T = [1, 1, 1, 1, \ldots, 1].$$

Hence, the matrix

$$\mathbf{A}^m \to \mathbf{q}_1\mathbf{r}_1^T \text{ as } m \to \infty.$$

Note, however, that the choice $NU = 1$ implies that there will be $N_2 - 1$ time-updates followed by a single full update at the last iteration, giving:

$$\mathbf{P}(t+1) = \mathbf{Q} + \mathbf{A}^T\mathbf{Q}\mathbf{A} + \mathbf{A}^{2T}\mathbf{Q}\mathbf{A}^2 + \mathbf{A}^{3T}\mathbf{Q}\mathbf{A}^3 + \ldots.$$

In the limit as $N_2 \to \infty$ the matrix $\mathbf{P}(t+1)$ will satisfy:

$$\mathbf{P}(t+1)/m \to \mathbf{q}_1\mathbf{r}_1^T\mathbf{Q}\mathbf{q}_1\mathbf{r}_1^T \text{ as } m \to \infty$$

i.e. $\mathbf{P}(t+1)/m \to \mathbf{r}_1\mathbf{q}_1^T[1,0,\ldots,0]^T[1,0,\ldots,0]\mathbf{q}_1\mathbf{r}_1^T$

or:

$$\mathbf{P}(t+1)/m \to \mathbf{r}_1\mathbf{r}_1^T. \quad (B.9)$$

This implies that:

$$\mathbf{b}^T\mathbf{P}(t+1)\mathbf{b}/m \to (\Sigma b_i)^2$$

and finally substituting the asymptotic value of $\mathbf{P}(t+1)$ from (B.9) into (B.7) gives for $\lambda = 0$:

$$\Delta u(t) = -(\Sigma b_i)^{-1}[1, 1, 1, \ldots, 1]\mathbf{x}(t). \quad (B.10)$$

Recall the formulation is that of the observable canonical form so that in a deterministic environment the states may be written as follows:

$$\begin{aligned} x_n(t) &= -\tilde{a}_n y(t-1) + b_{n-1}\Delta u(t-1) \\ x_{n-1} &= -\tilde{a}_{n-1}y(t-1) - \tilde{a}_n y(t-2) \\ &\quad + b_{n-1}\Delta u(t-2) + b_{n-2}\Delta u(t-1) \\ &\vdots \\ x_1(t) &= -\tilde{a}_1 y(t-1) - \ldots + b_0 \Delta u(t-1) + \ldots \end{aligned} \quad (B.11)$$

and

$$y(t) = x_1(t).$$

Combining (B.11) with the expression for the control signal (B.10) yields the parametric representation of the controller:

$$(\Sigma b_i)\Delta u(t) = -y(t) - (1 + \tilde{a}_1)y(t-1) - \ldots - (\Sigma b_i - b_0)$$
$$\Delta u(t-1) - (\Sigma b_i - b_0 - b_1)\Delta u(t-2) - \ldots \quad (B.12)$$

or:

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

where G is the solution of the Diophantine identity

$$G(q^{-1})A(q^{-1})\Delta + q^{-1}A(q^{-1})B(q^{-1})/B(1) = A(q^{-1}). \quad (B.13)$$

This can be verified simply by comparing the coefficients of G with those derived in (B.12). These are the equations of pole-assignment placing the closed-loop poles at the open-loop positions. Hence for open-loop stable processes the closed-loop is stable as $N_2 \to \infty$, and because the closed-loop poles are placed in exactly the same locations as those of the open-loop poles, this controller is a mean-level controller.

Remark 1. A mean-level controller provides a step in control following a step in the set-point which will drive the plant output exactly to the set-point and hence provide the same closed-loop dynamics as of the open-loop. Note that steps in load-disturbance are, however, rejected since the controller includes an integrator.

Remark 2. The spectral decomposition of the matrix $\mathbf{A}$ shows that a value of N_2 equal to the settling time of the plant is equivalent for control purposes to $N_2 \to \infty$.

Remark 3. For cases where there are multiple eigenvalues the spectral factorization (B.8) is not valid and the Jordan canonical form must be employed. However, because all plant eigenvalues are assumed to lie inside the unit circle the value of $\mathbf{A}^m$ as $m \to \infty$ remains the same as for the distinct eigenvalue case and the rest of the argument follows.

Theorem 3. The closed-loop system is equivalent to a stable state-dead-beat controller if

(1) the system $(\mathbf{A}, \mathbf{b}, \mathbf{c})$ is observable and controllable and

(2) $N_1 = n$, $N_2 \geqslant 2n - 1$, $NU = n$ and $\lambda = 0$, where n is the number of states of the plant.

Proof. Initially the case of $N_2 = 2n - 1$ is considered. The cost minimized by the controller proposed above is

$$J = \sum_{i=t}^{N_2+t} \mathbf{x}(i)^{\mathrm{T}}\mathbf{Q}(i)\mathbf{x}(i) \tag{B.14}$$

where $\mathbf{Q}(i) = 0$ for $i < t + N_1$ and $\mathbf{Q}(i) = \mathbf{cc}^{\mathrm{T}}$ for $i \geqslant t + N_1$. The Kalman control gains can therefore be calculated using the iterations (B.5, B.6, B.7). For the first $n - 1$ iterations of the Riccati equation only (B.6) is used with $\mathbf{Q} = \mathbf{cc}^{\mathrm{T}}$, hence obtaining the observability Grammian in Theorem (1). For the next n iterations (B.5, B.6) are used employing $\mathbf{Q} = 0$. Note that these iterations yield a unique Kalman control gain with $\mathbf{P}(t + N_2) = \mathbf{cc}^{\mathrm{T}}$.

From the predictive point of view, calculation of the control signal $\Delta u(t)$ is tantamount to solving the set of simultaneous equations:

$$\begin{bmatrix} g_{n-1} & g_{n-2} & \cdots & g_0 \\ g_n & g_{n-1} & \cdots & g_1 \\ \vdots & \vdots & \vdots & \vdots \\ g_{2n-2} & g_{2n-1} & \cdots & g_{n-1} \end{bmatrix} \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+n-1) \end{bmatrix} = \begin{bmatrix} \mathbf{w} - f(t+n) \\ \mathbf{w} - f(t+n+1) \\ \vdots \\ \mathbf{w} - f(t+2n-1) \end{bmatrix}.$$

Clearly if the matrix $\mathbf{G}$ is of full rank then $\Delta u(t)$ is unique.

$$\text{But } \mathbf{G} = \begin{bmatrix} \mathbf{c}^{\mathrm{T}} \\ \mathbf{c}^{\mathrm{T}}\mathbf{A} \\ \vdots \\ \mathbf{c}^{\mathrm{T}}\mathbf{A}^{n-1} \end{bmatrix} [\mathbf{A}^{n-1}B, \mathbf{A}^{n-2}B, \ldots, B].$$

By assumption 1 the controller minimizing the cost above is unique because $\mathbf{G}$ is full rank.

As the system is assumed controllable there exists a unique feedback gain $\mathbf{k}$ such that $(\mathbf{A} - \mathbf{bk}^{\mathrm{T}})^n\mathbf{x}(t) = 0$, independent of $\mathbf{x}(t)$. Such a controller is the state-dead-beat controller and is stable, independent of the pole-zero locations of the system. Note that the cost incurred by choosing $\mathbf{k}$ as the dead-beat controller is exactly zero (in the deterministic case)—the minimum achievable cost. Recall however, that the controller minimizing the cost was shown to be unique. Therefore, the controller derived from iterations of the Riccati equation or solving the set of simultaneous equations is the dead-beat controller and hence stable. Note that since the cost incurred is zero, increasing the horizon N_2 does not affect the minimum nor the solution of the optimization problem.

Remark 1. Note that the state-dead-beat control is equivalent to placing all of the closed-loop poles at the origin.

Remark 2. As seen in the assumption (1), the condition of minimal realization is necessary for the control calculation and therefore the choice of horizons above must in practice be used with caution. This is the same condition required when solving the Diophantine identity for a pole-assignment controller. Note that λ may be used to improve the condition of the $\mathbf{G}$ matrix.

Remark 3. If instead of (B.1) the augmented plant:

$$A(q^{-1})\Delta P(q^{-1})y(t) = B(q^{-1})\Delta P(q^{-1})u(t-1)$$

is considered and dead-beat control is performed on $\psi(t) = Py(t)$ instead of $y(t)$, the closed-loop poles will be located at the zeros of $P(q^{-1})$. This is because the closed-loop poles of the augmented plant $(w \to \psi(t))$ are all at the origin and therefore the closed-loop poles of the actual plant are at the zeros of $P(q^{-1})$. The predictions $f(t+j)$ can be obtained from (6) replacing $\psi(t)$ for $y(t)$ and the G-parameters from the algorithm given in Appendix A. The signal $P\Delta u(t)$ and subsequently $u(t)$ are calculated from the equations given above. This is then equivalent to the standard pole-placement controller.