# Multivariable stable generalised predictive control

B. Kouvaritakis
J.A. Rossiter

*Indexing terms: Predictive control, Multivariable control*

**Abstract:** Recent work has modified the basic generalised predictive-control strategy so as to derive a predictive-control algorithm with guaranteed closed-loop (nominal) stablity for the scalar case. The paper extends this work to the multivariable case. The proposed algorithm generates extra degrees of design freedom which can be given up systematically for the purpose of optimising closed-loop robustness properties.

## 1 Introduction

Generalised predictive control (GPC) is a well established algorithm [1, 2] for the control of scalar systems. The design philosophy behind GPC can be extended to the multivariable case and a suitable algorithm, which will be referred to as MGPC, has been presented [3]. Despite its popularity, GPC does not guarantee stability, and indeed will fail in the presence of near-coincidental unstable poles/zeros, unless one resorts to long horizons (which in turn may lead to numerical problems). Two approaches have been proposed recently to overcome this problem: one, contrained receding-horizon predictive control (CHRPC), which constrains the output to attain its target value over a horizon beyond the normal output horizon [4, 5], and the other, stable generalised predictive control (SGPC), which forms a stable closed loop before the application of the predictive-control strategy [6]. Of these the former relies on an open-loop output-prediction equation and, a weighting-sequence model for the computation of the forced response. As a result, the algorithm is computationally demanding and is susceptible to numerical inaccuracies in the case of open-loop unstable systems (especially when long output horizons have to be used). By contrast to this, SGPC performs a closed-loop prediction and uses the transfer-function-numerator polynomial to compute the forced response. Over and above its computational superiority, SGPC provides the systematic means for achieving certain robustness properties; it does this by identifying the degrees of freedom available in a pole-placement problem.

The present paper shows that the whole SGPC framework can be extended in a straightforward manner to the multivariable case. Most of the scalar results carry over to the multivariable case and hence will be stated here without proof. An exception to this concerns the prediction equation and the design for robustness; the main issue here being the noncommutativity of matrices under multiplication.



Paper 9417D (C8), first received 23rd September 1992 and in revised form 8th April 1993

The authors are with Oxford University, Department of Engineering Science, Parks Road, Oxford OX1 3PJ, United Kingdom

## 2 Conventions, notation and lemmata

In the sequel we will consider the class $F_N$ of causal matrix polynomials, $F(z) = \sum_{0,N} F_i z^{-i}$, or the class $L_N$ of bicausal matrix polynomials $L(z) = \sum_{-N,N} L_i z^{-i}$; all such matrices will be taken to be square, of dimension $m \times m$. Unless otherwise stated, $N$ will be taken to be $n$ and hence the $F(z)$ to be considered will be of degree $n$, i.e. $\delta\{F(z)\} = n$, and $F$ will denote the matrix of (matrix) coefficients of the powers of $z^{-1}$, namely $F = [F_0, F_1, \ldots, F_n]$.

### 2.1 Definitions

(i) $[\cdot]^T$ is taken to be the usual operation of transposition.

(ii) 'overbar' denotes complex conjugation.

(iii) $[\cdot]^{\#}$ reverses the order of the coefficients of a matrix polynomial:

$$\left[\sum_{i=0}^{n} F_i z^{-i}\right]^{\#} = F^{\#}(z) = \sum_{i=0}^{n} F_{n-i} z^{-i}$$

(iv) $[F(z)]^*$ denotes the adjoint of $F(z)$ given by $z^n F^{\# T}(z)$:

$$F^*(z) = F^T(z^{-1})$$

(v) $T_k\{F(z)\}$ truncates $F(z)$ after the $z^{-(k-1)}$ term; it is assumed that $k \leqslant n$:

$$T_k\left\{\sum_{i=0}^{n+p} F_i z^{-i}\right\} = \sum_{i=0}^{k-1} F_i z^{-i}$$

where $p$ is a nonnegative integer and it is assumed that $k \leqslant n$. For generality in the above, the truncation operator is applied to a matrix polynomial of degree greater than $n$.

(vi) $T_k\{F\}$ denotes the row-block matrix formed out of the first $k$ blocks of $F$.

(vii) $P\{F\}$ maps $F$ to $F(z)$; $P^{-1}\{F(z)\}$ maps $F(z)$ to $F$.

(viii) $E(z) = I$, where $I$ is the identity matrix of dimension $m$.

(ix) $E = [I, 0, 0, \ldots]$; the number of the $m \times m$ zero blocks will be problem dependent.

(x) $[L(z)]$ and $[L(z)]_+$ denote the causal and strictly anticausal part of $L(z)$:

$$\left[\sum_{i=-k_1}^{k_2} L_i z^{-i}\right] = \sum_{i=0}^{k_2} L_i z^{-i}$$

$$\left[\sum_{i=-k_1}^{k_2} L_i z^{-i}\right]_+ = \sum_{i=-k_1}^{-1} L_i z^{-i}$$

where $k_1$ and $k_2$ are two positive integers.

The authors thank the SERC for financial support.

(xi) $F_s^1(z)$, $F_u^1(z)$, $F_s^r(z)$ and $F_u^r(z)$ denote stable and unstable left and right factors of $F(z)$ defined by

$$\begin{aligned} F(z) &= F^T(z) * \bar{F}(z) + \lambda G^T(z) * \bar{G}(z) \\ &= F_s^1(z) F_u^1(z) \\ &= F_u^r(z) F_s^r(z) \end{aligned} \tag{1}$$

(xii) For $\boldsymbol{\Phi}(z) \in F_\infty$, $[\boldsymbol{\Phi}(z)]_L = z^\mu[\boldsymbol{\Phi}(z) - T_\mu\{\boldsymbol{\Phi}(z)\}]$ and $[\boldsymbol{\Phi}]_L = [\boldsymbol{\Phi}_\mu, \boldsymbol{\Phi}_{\mu+1}, \ldots]$.

In (xi) stable or unstable refers to the location of zeros (not the more usual interpretation which concerns pole locations).

*Lemma 2.1:* If the pair $\{F_s^{-1}(z), F_u^1(z)\}$ are left factors of $\mathbf{F}(z)$, then $F_u^{1*}(z) = F_s^1(z)$, $F_s^{1*}(z) = F_u^1(z)$.

*Proof:* This is a direct consequence of eqn. 1.

(xiii) $C_F$ denotes the block Toeplitz lower (block-) triangular convolution matrix of $F(z)$ whose $i$, $j$ blocks is $F_{i-j}$ for $i \geqslant j$ and zero otherwise

(xiv) $\Gamma_F$ denotes the matrix formed out of the first $\mu$ column blocks of $C_F$ whereas the remainder of $C_F$ will be denoted by $M_F$; it is assumed that $\mu \leqslant n$.

(xv) $H_F$ denotes the block hankel matrix of $F(z)$ whose $i, j$ block is $F_{i-1+j}$.

*Lemma 2.2:* Let $F(z)$, $G(z) \in \boldsymbol{F}$, then from the above definitions the following hold true:

(i) $Q^T(z) = T_n\{F(z)G^T(z)\} \leftrightarrow Q^T = C_F G^T$

(ii) $Q^T(z) = T_n\{F(z)T_\mu\{G^T(z)\}\} \leftrightarrow Q^T = \Gamma_F T_\mu\{G^T\}$

(iii) $[G(z)F^{*T}(z)]_- = P\{GC_F\}$ and $T_\mu\{[G(z)F^{*T}(z)]_-\} = P\{G\Gamma_F\}$

(iv) $[G^{*T}(z)F(z)]_- = P\{GH_F\}$

(v) For $\delta\{F(z)\} < n[G^{*T}(z)F(z)] = [T_n\{G^{*T}(z)\}F(z)]_- = P\{T_n\{G\}H_F\}$

*Proof:* This is a direct extension of lemmata 2.2-3 of Reference 6 and hence the proof will be omitted. In particular conditions (i) and (iii*a*) constitute an extension of lemma 2.2 of Reference 6, conditions (ii) and (iii*b*) are extension of lemma 2.3 of Reference 6, while conditions (iv) and (v) generalise the results of lemma 2.4 of Reference 6.

*Lemma 2.3:* Let $G(z) \in F$, and let $F(z) \in F$ have no zeros on the unit circle and let $\boldsymbol{F}(z)$ be as defined in eqn. 1 with stable and unstable left factors $F_s^1(z)$, $F_u^1(z)$. Then the power series

$$\boldsymbol{\Phi}(z) = [F_{u0}^1]^{-1}[F_s^1(z)]^{-1} = \sum_0^\infty \boldsymbol{\Phi}_i z^{-i} \tag{2}$$

is convergent inside the unit circle. Furthermore

(i) $[\boldsymbol{\Phi}(z)\boldsymbol{F}(z)] = E(z)$

(ii) If $\lambda = 0$, then $P\{E(\Gamma_F^T\Gamma_F)^{-1}\} = T_\mu\{\boldsymbol{\Phi}(z)\} + \varepsilon(z)$ with $\varepsilon = T_{n-\mu}\{[\boldsymbol{\Phi}]_L\}M_F^T\Gamma_F(\Gamma_F^T\Gamma_F)^{-1}$

(iii) If $\lambda \neq 0$ then $E(\Gamma_F^T\Gamma_F + \lambda\Gamma_G^T\Gamma_G)^{-1} = T_\mu\{\boldsymbol{\Phi}\} + \eta$ with $\eta = T_{n-\mu}\{\boldsymbol{\Phi}_L\}(M_F^T\Gamma_F + \lambda M_G^T\Gamma_G)(\Gamma_F^T\Gamma_F + \lambda\Gamma_G^T\Gamma_G)^{-1}$

*Proof:* This lemma also constitutes a straightforward extension of the results presented in Reference 6 and as such does not require proof. In particular eqn. 2 is an extension of eqn. 10 of Reference 6, whereas eqn. 2 together with condition (i) forms an extension of lemma 2.5; $F_{u0}^1$ denotes the first coefficient of $F_u^1(z)$. Condition (ii) is an extension of theorem 2.1 of Reference 6, whereas condition (iii) generalises the result of theorem 4.6 of Reference 6 when the scalar functions $b(z)$ and $A(z)$ are replaced by the matrix functions $F(z)$ and $G(z)$, respectively.

## 3 Multivariable predictive algorithm

### 3.1 Stabilising feedback loop

Consider the discrete time system described by

$$y(z) = B(z)A^{-1}(z)u(z) = B(z)A^{-1}\,\Delta u(z) \tag{3}$$

$$y(z) = A^L(z)^{-1}B^L(z)u(z) = A^L(z)^{-1}B^L(z)\,\Delta u(z) \tag{4}$$

with

$$A(z) = \Delta(z)A(z) \quad A^L(z) = \Delta(z)A^L(z) \tag{5}$$

where $\Delta(z) = 1 - z^{-1}$, $A(z)$, $B(z)$, $A^L(z)$, $B^L(z)$, are $m \times m$ polynomial matrices and $y(z)$, $u(z)$ denote the $z$-transform of the output and input vectors $u$ and $y$; as usual there is an implicit dead time in the model above which has been absorbed in $B(z)$ and $B^L(z)$. In common with GPC, the insertion of the $\Delta$ operator here provides a systematic means of including integral action into the overall control scheme. It is also noted that, unlike the scalar case where the numerator polynomial and the inverse of the denominator polynomial commute under multiplication, here we shall need both a left and right factorisation for the model transfer-function matrix $G(z)$. Clearly the relationship between the coefficients of $A(z)$, $B(z)$ and those of $A^L(z)$, $B^L(z)$ are bilinear, so that, given either a left or right factorisation it is easy to compute a corresponding right or left factorisation by solving a set of linear simultaneous equations.

In a manner analogous to SGPC, prior to the application of predictive control we form a stabilising loop around the system as shown in Fig. 1. The $m \times m$

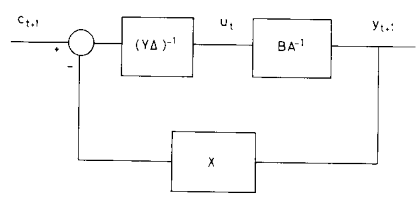


**Fig. 1** *Stabilising feedback configuration*

matrices $X^L(z)$ and $Y^L(z)$ can be chosen to be transfer functions and the whole class of these that yield stabilising controllers is well documented in the literature. For our purposes it is particularly convenient to choose these to be the minimal-order polynomial matrices which satisfy the Bezout identity

$$X^L(z)B(z) + Y^L(z)A(z) = I \tag{6}$$

It is noted that the whole class of stabilising controller can be generated if $X^L(z)$ is replaced by $X^L(z) + Q(z)A^L(z)$ and $Y^L(z)$ by $Y^L(z) - Q(z)B^L(z)$ for any $Q(z)$ stable; however it easy to show that $Q(z)$ does not affect the closed-loop relationships which, for the given choice of $X^L(z)$ and $Y^L(z)$ are given as

$$y(z) = B(z)c(z) \quad \Delta u(z) = A(z)c(z) \tag{7}$$

where $c(z)$ denotes the $z$-transform of the vector reference signal $c$. Thus if those equations were to form the basis of a predictive algorithm which computed the future values of $c$ which minimised the deviation of the predicted values of $y$ from a setpoint vector signal (when this is costed together with a measure of the future control activity), then it follows intuitively that the resulting predictive control law would imply additional feedback connections which would cancel the effect of $Q(z)$. This fact was proved rigorously for the scalar case [6] and the same proof can be extended to the multivariable case in an analogous manner. It is on account of this property that we have implicitly (see Fig. 1) taken $Q(z)$ to be zero.

### *3.2 Theoretical approach to the algorithm*

Eqn. 7 forms a simple theoretical basis for predictive control; this treatment is not suitable for the practical implementation of the algorithm (which will be given later) but it facilitates the stability analysis. In particular, simulating the equation forward in time we obtain the vectors of predicted (future) output-vector and control-increment vector values as

$$\begin{aligned} \boldsymbol{y}_{t+1} &= \Gamma_B \boldsymbol{c}_{t+1} + M_B c_\infty + H_B \boldsymbol{c}_t \\ \Delta \boldsymbol{u}_t &= C_A \boldsymbol{c}_{t+1} + M_A c_\infty + H_A \boldsymbol{c}_t \end{aligned} \tag{8}$$

where $\boldsymbol{y}_{t+1} = [y_{t+1}^T, y_{t+2}^T, \ldots, y_{t+n}^T]^T$, $\Delta u_t = [\Delta u_t^T, \Delta u_{t+1}^T, \ldots, \Delta u_{t+n-1}^T]^T$, $c_{t+1} = [c_{t+1}^T, c_{t+2}^T, \ldots, c_{t+\mu}^T]^T$, $c_t = [c_t^T, c_{t-1}^T, \ldots, c_{t-n}^T]^T$, and $c_\infty$ consists of the vector values of $c_{t+i}$, $i = 1, 2, \ldots, n - \mu$ that eliminate steady-state offsets between the output $y$ and the setpoint vector signal $r$. The particular choice of $c_\infty$ is indicated by eqn. 7 as

$$c_\infty = ((B(1))^{-1}[r_{t+\mu+1}^T, \ldots, r_{t+n}^T])^T = E_r \boldsymbol{r}_{t+1} \tag{9}$$

where $E_r$ is a block diagonal matrix (of conformal dimensions) with $B(1)^{-1}$ on all diagonal positions and zero elsewhere. Note that for convenience we have used $n$ and $\mu$ to denote the output and (reference) input horizon instead of $n_y$ and $n_c$ (which would constitute a more conventional notation). Then given the vector of future vector set points, $r_{t+1}$ of conformal dimension to $y_{t+1}$, the obvious predictive control cost with the corresponding optimal solution are

$$\begin{aligned} J_{GPC} &= \|\boldsymbol{r}_{t+1} - \Gamma_B \boldsymbol{c}_{t+1} - M_B c_\infty - H_B \boldsymbol{c}_t\|^2 \\ &\quad + \lambda \|\Gamma_A \boldsymbol{c}_{t+1} + M_A c_\infty + H_A \boldsymbol{c}_t\|^2 \end{aligned} \tag{10}$$

$$\begin{aligned} c_{t+1} &= E(\Gamma_B^T \Gamma_B + \lambda \Gamma_A^T \Gamma_A)^{-1} \\ &\quad \times (\Gamma_B^T(\boldsymbol{r}_{t+1} - M_B c\infty - H_B \boldsymbol{c}_t) \\ &\quad - \lambda \Gamma_A^T(M_A c_\infty + H_A \boldsymbol{c}_t)) \end{aligned} \tag{11}$$

The role of the $E$ matrix in the left-hand side of the above is to isolate the component of the optimal vector which is to be used next in the implementation of the predictive-control algorithm. Substituting for $c_\infty$ in eqn. 11 and rearranging we derive

$$A_{MSGPC} c_{t+1} = P_r \boldsymbol{r}_{t+1} \tag{12}$$

with

$$\begin{aligned} A_{MSGPC} &= [\{E(\Gamma_B^T \Gamma_B + \lambda \Gamma_A^T \Gamma_A)^{-1}\}^{-1}, \Gamma_B^T H_B \\ &\quad + \lambda \Gamma_A^T H_A] \end{aligned} \tag{13}$$

$$\begin{aligned} P_r &= E(\Gamma_B^T \Gamma_B + \lambda \Gamma_A^T \Gamma_A)^{-1} \\ &\quad \times (\Gamma_B^T - \Gamma_B^T M_B E_r - \lambda \Gamma_A^T M_A E_r) \end{aligned} \tag{14}$$

### *3.3 Nominal stability*

Eqns. 12–14 in conjunction with eqn. 7 imply that

$$y(z) = B(z) A_{MSGPC}(z)^{-1} P_r(z) r(z) \tag{15}$$

which identifies $P_r(z)$ as a prefilter, and $B(z)$, $A_{MSGPC}(z)$ as the closed-loop numerator and denominator polynomials; $r(z)$ denotes the $z$-transform of the setpoint vector signal $r$. Hence the nominal stability properties of the predictive algorithm can be determined by the stability properties of the zeros of $A_{MSGPC}$. We consider first some special cases for which this analysis is straightforward.

*Theorem 3.1:* For a control horizon of one ($\mu = 1$) and $(n = n_y) > \max\{\delta(B), \delta(A)\}$ MSGPC is stable.

*Proof:* For simplicity consider first the case of no control weighting ($\lambda = 0$) for which eqn. 11 reduces (after some rearrangement) to

$$\Gamma_B^T[\Gamma_B, H_B]\boldsymbol{c}_{t+1} = \Gamma_B^T \Gamma_B \gamma \tag{16}$$

where $\gamma$ is a vector function of the setpoint signal, and does depend on the reference signal $c$. Now for $\mu = 1$ it is easy to show that $\Gamma_B^T[\Gamma_B, H_B] = \Gamma_B^T H_{z-1B}$ so that from eqn. 16 we have

$$A_{MSGPC}(z) = P\{\Gamma_B^T H_{z-1B}\} = [B^*(z)B(z)]_- \tag{17}$$

where use has been made of lemma 2.2. Now let $U_0 \Lambda_0^2 U$OTDR$^T$ denote the eigenvalue/vector decomposition of the coefficient of $z^\circ$ in $B^*(z)B(z)$, and let $B^\sim(z) = B(z)U_0\Lambda_0^{-1}$ so that

$$B^{\sim *}(z)B^\sim(z) = \{R_-(z) + I + R_-(z)\}^* \tag{18}$$

Then, for $z = e^{j\theta}$ the eigenvalues of the left-hand side of the above are positive and hence (by symmetry and the eigenvalue shift property) the real part of the eigenvalues of $R_-(z)$ must be greater than $-1/2$. Thus $[B^{\sim *}(z)B^\sim(z)] = R_-(z) + I$ is positive for all $z$ on the unit circle, and as such (by the principle of the argument) will have no zeros outside the unit circle (since it has no poles outside the unit circle). However $[B^{\sim *}(z)B^\sim(z)]_-$ and $A_{MSGPC}(z)$ share common zeros, and this completes the proof for case $\lambda = 0$. Similar arguments applied to

$$A_{MSGPC} = [B^*(z)B(z) + \lambda A^*(z)A(z)]_- \tag{19}$$

rather than the closed-loop $A$ matrix of eqn. 17 readily yield the proof for the case of $\lambda \neq 0$.

*Theorem 3.2:* For $n \geqslant \mu + \max\{\delta(B), \delta(A)\}$ and $\mu \to \infty$ MSGPC is stable.

*Proof:* First consider the case for $\lambda = 0$; the identity $[I, \ E(\Gamma_B^T \Gamma_B)^{-1}\Gamma_B^T H_B] = E(\Gamma_B^T \Gamma_B)^{-1}\Gamma_B H_{z-1B}$ implies $A_{MSGPC} = E(\Gamma_B^T \Gamma_B)^{-1}\Gamma_B^T H_{z-1B}$, and this in conjunction with lemma 2.3 gives $A_{MSGPC} = (T_n\{\Phi\})\Gamma_B^T H_{z-1B} + ([T_n\{\Phi\} - T_\mu\{\Phi\}] + \varepsilon)\Gamma_B^T H_{z-1B}$. In the limit, as $\mu \to \infty$ the second term in the right-hand side will vanish, so by lemma 2.2 we have $\lim_{\mu \to \infty} A_{MSGPC} = P^{-1}\{[\Phi^{*T}(z)B^T(z)B^-(z)]_-\} = F_s^1(z)$, where the last equality follows from the definition of $\Phi(z)$ and the fact that the quantity inside the square brackets itself equals the stable factor $F_s^1(z)$.

The proof for the case when $\lambda \neq 0$ runs along similar lines. Thus from eqn. 11 we have:

$$\begin{aligned} A_{MSGPC} &= E(\Gamma_B^T \Gamma_B + \lambda \Gamma_A^T \Gamma_A)^{-1} \\ &\quad \times (\Gamma B^T H_{z-1B} + \lambda \Gamma_A^T H_{z-1A}) \end{aligned} \tag{20}$$

so that by lemma 2.3 for $F(z) = B(z)$, $G(z) = A(z)$ we have

$$A_{MSGPC} = (T_\mu\{\Phi\} + \eta)(\Gamma_B^T H_{z-1B} + \lambda \Gamma_A^T H_{z-1A}) \tag{21}$$

Then by lemma 2.2 we obtain

$$\lim_{\mu \to \infty} A_{MSGPC}(z) = [\Phi^{*T} * (B^*(z)B(z) + \lambda A^*(z)A(z))]_- = F_s^l(z) \quad (22)$$

which completes the proof.

Although theorems 3.1 and 3.2 do not establish stability for the general case, they define the closed-loop $A$ matrix (and hence the closed-loop pole positions) exactly for the case of $\mu = 1$, and to within a specified error for $\mu$ large. The result below, on the other hand, establishes the stability property of MSGPC without recourse to $A_{MSGPC}$.

*Theorem 3.3:* For $n - \mu \geq \max\{\delta(B), \delta(A)\}$ the nominal MSGPC is stable.

*Proof:* This is similar to that given in the scalar case and will not be elaborated here. It merely considers the case of step setpoint vector signals and establishes that, for these, the choice of $c_\infty$ ensures that the cost of eqn. 10 is monotonically decreasing until it becomes zero.

## 4 Practical implementation of MSGPC

Eqn. 7, though convenient from a theoretical viewpoint, does not provide for any regression on the values of the output vector $y_t$; as such it cannot form the basis of a practical predictive algorithm which must have a feedback mechanism for correcting the theoretical output predictions in the face of model mismatches and disturbances. The approach below (Sections 4.1 and 4.2) extends the scalar SGPC from the scalar to the multivariable case and thus provides the required answer. However, in the scalar case polynomials (and their inverses) commute under multiplication and this is exploited in conjunction with the corresponding Bezout identity to avoid matrix inversion altogether in the prediction equations. In the multivariable case we are dealing with matrix polynomials and these (together with their inverses) do not commute under multiplication. The following results, through suitable deployment of right and left-polynomial-matrix factorisation, provide the means of avoiding matrix inversions during the prediction stage of the multivariable algorithm.

*Lemma 4.1:* Given the Bezout identity (eqn. 6) and the above definitions, the following identity holds true:

$$[C_{YL} C_{BL}^{-1} C_{AL} + C_{XL}]^{-1} = C_B \quad (23a)$$

$$[C_{YL} + C_{XL} C_{AL}^{-1} C_{BL}]^{-1} = C_A \quad (23b)$$

*Proof:* The Bezout identity implies that

$$C_{YL} C_A + C_{XL} C_B = I \quad (24)$$

but given that $A^L(z)B(z) = B^L(z)A(z)$ we have

$$C_{AL} C_B = C_{BL} C_A \quad \text{or} \quad C_A [C_B]^{-1} = [C_{BL}]^{-1} C_{AL}$$

so that after some rearrangement eqn. 24 can be rewritten as

$$C_{YL} C_{BL}^{-1} C_{AL} + C_{XL} = C_B^{-1} \quad (25)$$

which establishes eqn. 23*a*; eqn. 23*b* can be derived in an analogous manner.

*Lemma 4.2:* Given $G(z) = B(z)A^{-1}(z)$ with $B(z)$ and $A(z)$ right coprime and $A(z)$ nonsingular, there exist polynomial matrices $A^L(z)$, $B^L(z)$, $X(z)$, $Y(z)$, $X^L(z)$ and $Y^L(z)$ such that

$$X^L(z)B(z) + Y^L(z)A(z) = I$$

$$-X^L(z)Y(z) + Y^L(z)X(z) = 0$$

$$-A^L(z)B(z) + B^L(z)A(z) = 0$$

$$A^L(z)Y(z) + B^L(z)X(z) = I \quad (26)$$

*Proof:* See Reference 7, p. 382.

*Lemma 4.3:* Under the dual coprime factorisation of lemma 4.2 and the definitions the following identity holds true:

$$C_{YL} C_{BL}^{-1} = C_B^{-1} C_Y \quad (27a)$$

$$C_{XL} C_{AL}^{-1} = C_A^{-1} C_X \quad (27b)$$

*Proof:* Eqns. 26 imply

$$A(z) = [B^L(z)]^{-1} A^L(z) B(z) \quad (28a)$$

$$X(z) = [Y^L(z)]^{-1} X^L(z) Y(z) \quad (28b)$$

$$A^L(z) = Y^{-1}(z) - B^L(z) X(z) Y^{-1}(z) \quad (28c)$$

$$Y^L(z) A(z) + X^L(z) B(z) = I \quad (28d)$$

Substitute eqn. 28*b* into eqn. 28*c*, and substitute the new expression for $A^L(z)$ into eqn. 28*a* to obtain

$$A(z) = [[B^L(z)]^{-1} Y^{-1}(z) - [Y^L(z)]^{-1} X^L(z)] B(z) \quad (29)$$

which, when introduced into eqn. 28*d*, after some rearrangement, gives

$$B(z) Y^L(z) = Y(z) B^L(z) \quad (30a)$$

or

$$C_B C_{YL} = C_Y C_{BL} \quad (30b)$$

Premultiplication of eqn. 30*b* by $C_B^{-1}$ and postmultiplication by $C_{BL}^{-1}$ establishes eqn. 27*a*; eqn. 27*b* can be derived in an analogous manner.

### 4.1 Prediction equations

Rather than simulate forward in time the closed-loop equations (eqns. 7), here we deploy the open-loop-model and stabilising-controller equations

$$A^L y_{t+1} = B^L \Delta u_t \quad Y^L \Delta u_t = c_t - X^L z^{-1} y_{t+1} \quad (31)$$

from which the vector of predicted outputs can be related to past/future values of $\Delta u$ and $c$:

$$H_{AL} y + C_{AL} y = H_{BL} \Delta u + C_{BL} \Delta u \quad (32)$$

$$H_{YL} \Delta u + C_{YL} \Delta u = c - H_{z^{-1}XL} y - C_{z^{-1}XL} y \quad (33)$$

where $y = [y_t, y_{t-1}, \ldots, y_{t-n}]$; $\Delta u = [\Delta u_{t-1}, \Delta u_{t-2}, \ldots, \Delta u_{t-n}]$, and $\Delta u = [\Delta u_t, \Delta u_{t+1}, \ldots, \Delta u_{t+n-1}]$. Then using lemmata 4.1 and 4.3 we have the following simple result.

*Theorem 4.1:* If $A(z)$, $B(z)$, $A^L(z)$, $B^L(z)$, $X(z)$, $Y(z)$, $X^L(z)$, $Y^L(z)$ form a dual coprime factorisation as per lemma 4.2, then the predicted output and control-increment vectors are given as

$$y = \Gamma_B c + M_B E_r r - (C_B H_{XL} + C_Y H_{AL}) y - (C_B H_{YL} + C_Y H_{BL}) \Delta u \quad (34)$$

$$\Delta u = \Gamma_A c + M_A E_r r - (C_A H_{XL} + C_X H_{AL}) y - (C_A H_{YL} + C_X H_{BL}) \Delta u \quad (35)$$

*Proof:* Eqn. 34 can be proven by solving eqn. 32 for $\Delta u$, substituting into eqn. 33 and solving for $y$, invoking lemma 4.1 to avoid having to invert the coefficient of $y$; lemma 4.3 has also been used to avoid having to invert $C_B$. Eqn. 35 can be established in a dual manner by solving eqn. 32 for $y$ and then substituting into eqn. 33.

*Theorem 4.2:* Given the dual-coprime factorisation of lemma 4.2, the optimal MSGPC control law can be written as

$$c_{t+1} = P_r r_{t+1} - P_y y_t - z^{-1} P_u \Delta u_{t+1} \tag{36}$$

where

$$P_r = E(\Gamma_B^T \Gamma_B + \lambda \Gamma_A^T \Gamma_A)^{-1} \times [\Gamma_B^T(I - M_B E_r) - \lambda \Gamma_A^T M_A E_r] \tag{37}$$

$$P_y = E(\Gamma_B^T \Gamma_B + \lambda \Gamma_A^T \Gamma_A)^{-1} [\Gamma_B^T(C_B H_{XL} + C_Y H_{AL}) + \lambda \Gamma_A^T(C_A H_{XL} + C_X H_{AL})] \tag{38}$$

$$P_u = E(\Gamma_B^T \Gamma_B + \lambda \Gamma_A^T \Gamma_A)^{-1} [\Gamma_B^T(C_B H_{YL} + C_Y H_{BL}) + \lambda \Gamma_A^T(C_A H_{YL} + C_X H_{BL})] \tag{39}$$

*Proof:* The result follows directly from substituting eqns. 34 and 35 into the cost

$$J_{MSGPC} = \|r - y\|^2 + \lambda \|\Delta u\|^2 \tag{40}$$

and setting the derivative of the result, with respect to $c$, equal to zero; the matrix $E$ premultiplies all three expressions in eqns. 37–39, because, out of the entire optimal future vector $c$, we implement only the component which relates to the next sampling instant.

*Corollary 4.1:* The MSGPC controller and overall closed-loop numerator and denominator polynomial matrices are given by

$$N_k(z) = P_y(z) + X^L(z) \tag{41a}$$

$$D_k(z) = [z^{-1} P_u(z) + Y^L(z)]\, \Delta(z) \tag{41b}$$

$$B_{MSGPC} = B(z) \tag{42a}$$

$$A_{MSGPC} = [P_y(z) + X^L(z)]B(z) + [z^{-1} P_u(z) + Y^L(z)]A(z) \tag{42b}$$

*Proof:* This follows directly from eqn. 36 in conjunction with eqns. 31.

Clearly in the absence of any mismatch and/or disturbances the control system of theorem 4.2 and corollary 4.1 (depicted in Fig. 2) is identical to that given in the

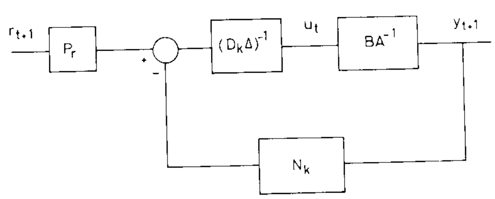


**Fig. 2** *Feedback configuration after optimisation*

theoretical development of Section 3. As a consequence, the prefilter $P_r(z)$ of eqn. 36 as well as the closed-loop polynomial $A_{MSGPC}$ of eqns. 42 are identical to those given in eqns. 12–14. However the implementation of Fig. 2, by expressing the optimal reference signal $c$ as a function of the future $r$s as well as the past $y$s and $\Delta u$s, provides a feedback mechanism which can respond to both model mismatch and disturbances.

A consequence of the above remarks is that any controller comprising a numerator and denominator polynomial $N_k''(z)$, $D_k''(z)$ which satisfy the identity

$$N_k''(z)B(z) + D_k''(z)A(z) = A_{MSGPC}(z) \tag{43}$$

where $A_{MSGPC}(z)$ can be computed from theorem 4.2 and corollary 4.1, or more simply from eqns. 12–14, together with the prefilter $P_r(z)$ of eqns. 36–39 or more simply eqns. 12–14 will result in an identical feedback configuration; such a configuration will therefore provide optimal preditive control. Thus one can combine the convenience of the theoretical treatment of Section 3, together with the implementation of Fig. 2 to obtain a simple but also practical implementation of MSGPC.

### 4.2 MSGPC algorithm

*Step 1:* Obtain the minimal-order solutions $X^L(z)$, $Y^L(z)$ of the Bezout identity

$$X^L(z)B(z) + Y^L(z)A(z) = I$$

*Step 2:* Select suitable output and control horizons $n$ and $\mu$; to guarantee stability $n$, $\mu$ should be subject to the mild constraint that $n - \mu \geqslant \max\{\delta[A(z)], \delta[B(z)]\}$.

*Step 3:* Compute $P_r(z)$ and $A_{MSGPC}(z)$ from eqns. 12–14.

*Step 4:* Compute any pair $N_k(z)$, $D_k(z)$ that satisfy the Bezout identity (eqn. 43), say compute the minimal order $N_k(z)$, $D_k(z)$.

*Step 5:* Use these together with $P_r(z)$ to implement MSGPC as per Fig. 2.

The algorithm in the form given above obviates the need for the dual-coprime factorisation of lemma 4.2, and is computationally undemanding; the most taxing part of the algorithm being the inversion of an $\mu m \times \mu m$ matrix ($m$ denoting the number of system inputs and outputs).

## 5 Robustness

It was pointed out above that the implementation of MSGPC is not unique, and that any controller $[D_k''(z)]^{-1} N_k''(z)$ satisfying eqn. 43 will result in precisely the same optical transfer-function matrices from $r$ to $y$ and $r$ to $\Delta u$. Thus if $N_k(z)$, $D_k(z)$ denote the minimal-order solutions to eqn. 43, the whole family of MSGPC controllers will be given as

$$K_{MSGPC} = [\Delta(z)\{D_k(z) - Q(z)B^L(z)\}]^{-1} \times [N_k(z) + Q(z)A^L(z)] \tag{44}$$

where $Q(z)$ is any stable transfer function matrix. Thus $Q(z)$ represents the degrees of freedom available within MSGPC that can be given up to maximise the robustness properties of the overall feedback configuration. For example, suppose that the transfer-function matrix $G(z) = B(z)A^{-1}(z) = A^{L-1}B^L(z)$ were subject to an additive unstructured uncertainty $\Delta G(z)$, which is unknown but is subject to the constraint $\sigma_{max}[\Delta G(z)] \leqslant W(z)$, for all $z = \exp(j\theta)$, $0 \leqslant \theta < \pi$, $\sigma_{max}$ denoting the maximum singular value of a matrix. Then the necessary and sufficient condition for robust stability is

$$\sigma_{max}[W(z)K_{MSGPC}(z)[I + G(z)K_{MSGPC}(z)]^{-1}] < 1 \quad \text{for all } z \text{ on the unit circle} \tag{45}$$

thus, the optimal choice $K_{MSGPC}$ is that which minimises the supremal value of the maximum singular value above over all $z = \exp(j\theta)$, $0 \leqslant \theta \leqslant \pi$; such a controller would maximise the system tolerance to uncertainty in that it would maximise the value of a positive scaling factor $\alpha$ for which the system will be robustly stable when $G(z)$ is subject to an unstructured additive uncertainty subject to the constraint $\sigma_{max}[\Delta G(z) \leqslant W(z)$. Hence the $Q(z)$ which maximises tolerance to uncertainty is the $Q(z)$ which minimises the norm

$$\begin{aligned} \| W(z)K_{MSGPC}(z)[I + G(z)K_{MSGPC}(z)]^{-1} \|_\infty \\ = \| W(z)[D_k(z)A(z) + N_k(z)B(z)]^{-1} \\ \times [N_k(z) + Q(z)A^L(z)] \|_\infty \\ = \| T_1(z) - T_2(z)Q(z)T_3(z) \|_\infty \end{aligned} \tag{46}$$

where the definition of $T_i(z)$, $i = 1, 2, 3$ is implicit in the above.

This minimisation is a standard $H_\infty$ problem and can be solved using well known state–space techniques. However, given the polynomial nature of $A(z)$, $B(z)$, $X(z)$, $Y(z)$, $N_k(z)$, $D_k(z)$ it appears desirable that one should look for a solution for $Q(z)$ which is itself a polynomial. One possible avenue for this is to transform eqn. 45 (through an inner–outer factorisation) to $\|R(z) - Q''(z)\|_\infty$ and then consider the Laurent expansion of $R(z)$ with the view to obtaining a matrix representation of the appropriate Hankel operator. The norm of this matrix can be minimised using Parrott's theorem [8] to select the matrix coefficients of $Q''(z)$, sequentially. Unfortunately, this rather elegant manner of solution would result in a high-order $Q''(z)$ and thus an even higher-order $Q(z)$.

Here we appeal to Trefethen's work [9] (based on the Caratheodory–Fejer theorem) to derive a suitable multivariable algorithm for the computation of an optimal finite-length $Q(z)$. To simplify notation, and in particular to get rid of the subscripts in eqn. 45, let the matrix on the right-hand side of eqn. 45 be redefined as

$$E(z) = K(z) - L(z)Q(z)M(z) \tag{47}$$

where here $E(z)$ denotes the error matrix, and is not to be confused with the identity-matrix polynomial used earlier. Then, letting a change from capitals to lower case indicate the stacking of the column vectors to form a block vector, we can write

$$e(z) = k(z) - \Xi_{L,M}(z)q(z) \tag{48a}$$

with

$$\Xi_{L,M}(z) = L^T(z) \otimes M^T(z) \tag{48b}$$

where $\otimes$ denotes the Kronecker product of two matrices. Next without loss of generality suppose that all the elements of $Q(z)$ have the same order, say $s$, and let $q_{ij}^T$ denote the column vector of coefficients of $q_{ij}(z)$; then we may write

$$q_{ij}(z) = f(z)q_{ij}^T \quad f(z) = [1, z^{-1}, \ldots, z^{-s}] \tag{49}$$

from which it follows that

$$q(z) = F(z)\theta \tag{50}$$

where

$$F(z) = \text{diag}\{[f(z), f(z), \ldots, f(z)]\}$$

and

$$\theta = [q_{11}, q_{21}, \ldots, q_{m1}, q_{12}, \ldots, q_{mm}]$$

With this notation eqns. 48 can be rewritten as

$$e(z) = k(z) - \Xi_{L,M}(z)F(z)\theta \tag{51}$$

and therefore we can state the following result.

*Lemma 5.1:* Given the definitions of eqns. 47–51, and for $z = e^{j\phi}$ the Euclidean norm of $e(z)$ is related to the singular values $\sigma_i(z)$ of $E(z)$ as

$$\| k(z) - \Xi_{L,M}(z)F(z)\theta \|^2 = \sum \sigma_i^2(z) \tag{52}$$

*Proof:* The result follows from the observation that $e^*(z)e(z) = \text{trace}\{E^*(z)E(z)\}$ so that $\|e(z)\|^2$ is equal to the sum of the eigenvalues of $E^*(z)E(z)$ which, for $z = e^{i\phi}$ coincide with the squares of the singular values of $E(z)$.

By their definition, $K(z)$, $L(z)$ and $M(z)$ do not have unstable poles, and thus have causal and convergent weighting sequences. As a consequence, by considering the number of significant terms in these sequences, it is possible to apply the inverse-sampling theorem and determine the minimum number at which $E(z)$ can be sampled without loss of information. In accordance with the usual discrete-Fourier-transform techniques, therefore, we will sample $E(z)$ at $z' = \exp\{j(i2\pi/v)\}$ where $v$ is a large enough integer dictated by the inverse-sampling theorem. Then an obvious first step toward the minimisation of the infinity norm is to apply Lawson's algorithm [10] to the cost

$$J = \sum \|e(z_i)\|^2 \tag{53}$$

However, such a procedure would yield the solution $\theta$ for which $\sum \sigma_i^2(z)$ is made as near allpass as possible and would minimise the infinity norm of the this sum. Our interest is in the maximum singular value rather than the sum of the squares of all the singular values. With this in mind we propose the following modification of Lawson's algorithm.

*Robustness algorithm:* Let $E^{(j)}$ denote the value of $E(z)$ for $\theta = \theta^{(j)}$ and let $\sigma_1(z)$ denote the maximum singular value of $E^{(j)}(z)$; furthermore let $j = 0$ and $w_i^{(\theta)} = 1$ for all $i$.

*Step 1:* Consider the cost

$$J^{(j)} = \sum w_i^{(j)} \|e(z_i)\|^2 \tag{54}$$

which is quadratic in $\theta$. Set the derivative of $J^{(j)}$ with respect to $\theta$ equal to zero and solve the resulting linear equation for $\theta$ to obtain the value $\theta^{(j)}$ which minimises $J^{(j)}$.

*Step 2:* If $\sum [w_i^{(j)} - w_i^{(j-1)}]^2$ is above a preset threshold, update the weights in accordance with the recurrence relationship

$$w_i^{(j+1)} = \frac{w_i^{(j)}\sigma_i^{(j)}(z_i)}{\sum w_k^{(j)}\sigma_1^{(j)}(z_k)}$$

and repeat step 1 for the new value of $j$.

*Theorem 5.1:* If the dimension of $\theta$ is large enough then, to within truncation errors, the robustness algorithm has a fixed point (for a zero threshold). Furthermore, for a $\theta$ of any dimension (and for a zero threshold) the algorithm can only converge to a solution for which $\sigma_1(z_i) = \sigma_1(z_k)$ for all $i$, $k$ which are associated with nonzero weights (and $i \neq k$).

*Proof:* The first part of the theorem follows from the discussion at the very beginning of the Section which asserts that there exists a unique infinite but convergent series

$[Q_c, Q_1, \ldots]$ which defines the $Q(z)$ which minimises the infinity norm of $E(z)$.

To prove the second part of the theorem assume that the weights have converged, in which case by eqn. 54, for all $i$ such that $w_i^{(j)} \neq 0$ we have

$$\sigma_1^{(j)}(z_i) = \left[\sum w_k^{(j)}\sigma_1^{(j)}(z_k)\right] \tag{55}$$

and hence $\sigma_1^{(j)}(z_i)$ is equal to a constant for all $i$.

This theorem together with considerable numerical evidence suggest the following:

*Conjecture 5.1:* The robustness algorithm will always converge and will yield the polynomial $Q(z)$ of a given order that minimises the infinity norm of $E(z)$.

Finally, note that $A^L(z)$ in eqn. 44 has $\Delta(z)$ as a factor which implies that the $\theta$ of the robustness algorithm will lead to a $\sigma_1(z)$ which is allpass at all $z_i$ other than $z_0 = 1$ (for which $\sigma_1$ is independent of $Q$).

## 6 Examples

Here we give two numerical examples which illustrate the efficacy of MSGPC (and its superiority over MGPC, the multivariable extension of GPC), demonstrates the usefulness of the theoretical analysis of Section 3 and illustrate the optimality of the robustness algorithm (as well as confirm conjecture 5.1).

### *6.1 Example 1*

In the single-input/single-output case it is known that near coincidence of unstable poles/zeros causes considerable stability problems for GPC. This difficulty clearly carries over to the multivariable case as is demonstrated here in relation to the model

$$A(z) = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} + \begin{bmatrix}-2.4 & 0\\0 & -1.9\end{bmatrix}z^{-1} + \begin{bmatrix}0.68 & 0\\0 & 0.28\end{bmatrix}z^{-2} + \begin{bmatrix}0.24 & 0\\0 & 0.48\end{bmatrix}z^{-3} \tag{56}$$

$$B(z) = \begin{bmatrix}0.219 & 0.6789\\0.047 & 0.6793\end{bmatrix}z^{-1} + \begin{bmatrix}0.0766 & -0.05702\\0.00165 & -0.5706\end{bmatrix}z^{-2} + \begin{bmatrix}-0.9820 & -0.643\\-0.211 & -0.6434\end{bmatrix}z^{-3} \tag{57}$$

which has two near coincident unstable poles at 2 and 1.5 with corresponding zeros at 1.95 and 1.48. For small control horizons, MGPC gives poor results and for a control horizon of 1 it fails to produce stable results altogether. In contrast to this, MSGPC gives perfectly satisfactory results for any value of $\mu$. However, to permit a direct comparison here we choose $\mu = 3$. For this value of control horizon Fig. 3 illustrates the stability properties of MGPC by plotting (in broken lines) the variation with values of output horizon, of the distance of the closed-loop poles from the origin, when the control weight is taken to be $\lambda = 0$ (Fig. 3*a*) and $\lambda = 1$ (Fig. 3*b*); superimposed on these plots are the corresponding results for MSGPC plotted with a minus sign and in chain-dotted lines to make the distinction between the two sets of results clearer. From Section 3 it is known that MSGPC achieves stability for $n - \mu \geqslant \max\{\delta(B), \delta(A)\}$ — and in fact for $\lambda = 0 n - \mu \geqslant \delta(B)$ — which for this example imply that MSGPC stability is guaranteed for $n > 6$ for $\lambda \neq 0$ and $n > 4$ for $\lambda = 0$; this is clearly borne out by Fig. 3 from which it is seen that MSGPC

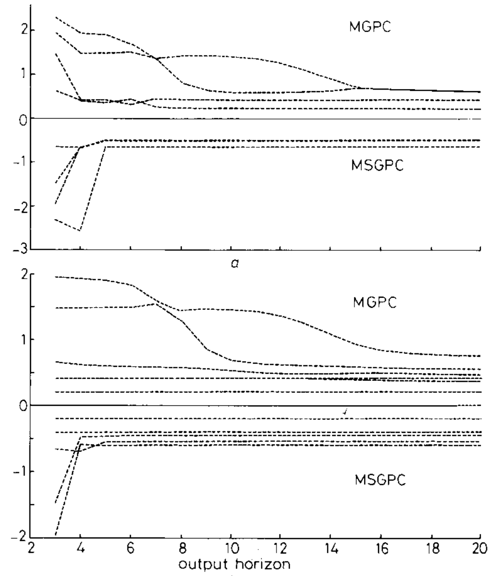


**Fig. 3** *Moduli of closed-loop poles as n varies*
$a$ $\mu = 3$ $\lambda = 0$
$b$ $\mu = 3$ $\lambda = 1$

achieves stability for $n > 4$, $\lambda = 0$ and for $n > 3$, $\lambda = 1$. In contrast to this, MSGPC cannot stabilise this model unless the output horizon is taken to be large ($n > 14$ for $\lambda = 0$ and $n > 15$ for $\lambda = 1$). Fig. 4 provides alternative evidence of the good stability properties of MSGPC by illustrating the variation of the distance of closed loop from the origin with varying $\mu$, for $n = 20$, $\lambda = 0$ (Fig. 4*a*) and $n = 20$, $\lambda = 1$ (Fig. 4*b*); from these it can be seen that the closed-loop poles reach their asymptotic positions (as per theorem 3.2) for small values of $\mu$ ($\mu$ about 6 for $\lambda = 0$ and $\mu$ about 8 for $\lambda = 1$). For $\lambda = 0$ and $n - \mu \leqslant 1$ we know that there is no guarantee of stability and this helps to explain the divergence of the closed-loop poles shown in Fig. 4*a* for $\mu > 18$.

Further evidence of the superiority of MSGPC over MGPC is provided by Figs. 5 and 6 which present the

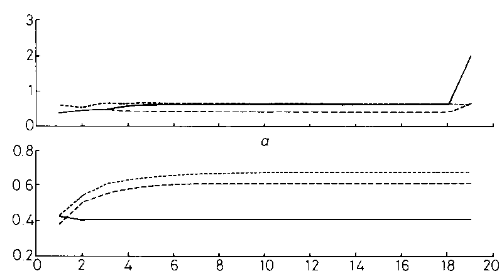


**Fig. 4** *Modulis of closed-loop poles as μ varies*
$a$ $n = 20$ $\lambda = 0$
$b$ $n = 20$ $\lambda = 1$

simulation results for a unit step demand on the first and second output, respectively; Figs. 5*a* and 6*a* depict the output responses, whereas Figs. 5*b* and 6*b* show the response of the inputs. The horizons for this run were set

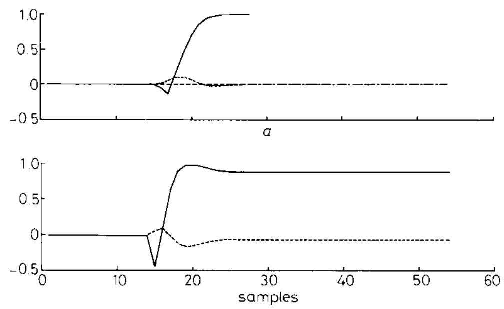


**Fig. 5** *Output and input responses (unit step in loop 1)*
*a* Output response
*b* Input response

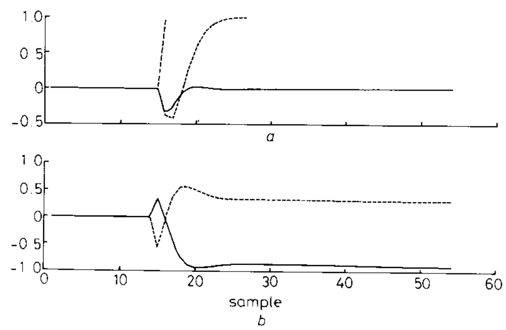


**Fig. 6** *Output and input responses (unit step in loop 2)*
*a* Output response
*b* Input response

at $\mu = 3$, $n = 10$, while the control weight was chosen to be $\lambda = 1$. The responses are fast, largely nonoscillatory and noninteractive. In contrast to this, for the same choice of horizons and control weights MGPC gives unstable simulation results (not shown here); this of course is consistent with the information presented in Fig. 3.

*Example 2:* This example is chosen so that both MGPC and MSGPC give a comparable (and satisfactory) simulation results (which therefore will not be displayed). The purpose of the example is to demonstrate that the $Q$ parameter, rather than the $T$ filter provides the systematic means of achieving robustness with respect to additive unstructured uncertainty. There is a host of other desirable robustness properties; the one chosen here is representative.

Consider the model with a transfer-function matrix defined by

$$A(z) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} -1.4345 & 0.3573 \\ 0.4279 & -0.5138 \end{bmatrix} z^{-1} + \begin{bmatrix} -0.0867 & 0.2140 \\ -0.6687 & 0.1000 \end{bmatrix} z^{-2} \quad (58)$$

$$B(z) = \begin{bmatrix} 0.1446 & -0.6171 \\ 0.8107 & -1.1918 \end{bmatrix} z^{-1} + \begin{bmatrix} -2.0820 & 1.1506 \\ -0.9729 & 0.0429 \end{bmatrix} z^{-2} \quad (59)$$

whose poles are at 1.4817, 0.7519, $-0.1427 \pm j0.3167$, and whose zeros are at $1.4547 \pm j1.0122$. Clearly, this model does not have near-pole/zero cancellations outside the unit disc. Using eqns. 58 and 59 and selecting the same values for $n$, $\mu$, $\lambda$ as for example 1 (i.e. 10, 3, 1), the MSGPC and MGPC controllers $K(z)$ were computed for $Q(z) = 0$ and $T(z) = I$. As mentioned above the closed-loop simulations for these two controllers are nearly identical and will not be presented as they provide no means of comparison. Instead in Fig. 7*a* we plot the frequency of $\sigma_{max}[K(z)\{I + G(z)K(z)\}^{-1}]$, the supremal

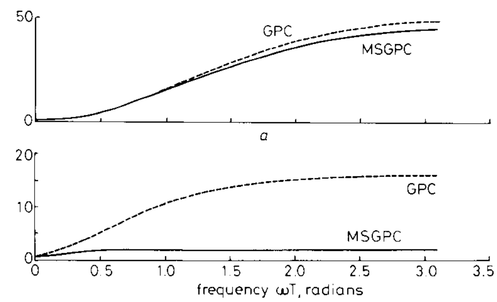


**Fig. 7** *Robustness indicators*
*a* Before compensation
*b* After compensation

value of which serves as an indicator of robustness. From the Figure it is obvious that, without the $Q$ parameter or the $T$ filter, both MSGPC and MGPC produced plots which fluctuate a great deal with frequency and resulted in large supremal values (about 48 for MGPC and 44 for MSGPC), thereby indicating poor robustness.

Next we introduce the $Q$ parameter designed using the procedure outlined in Section 5 and a $T$ filter taken to be $T(z) = (1 - 0.8z^{-1})I$; this value is normally recommended in the literature. Neither $Q$ nor $T$ affect the nominal closed-loop responses but they both have a significant effect on the robustness indicator $\sigma_{max}[K(z)\{I + G(z)K(z)\}^{-1}]$, as shown in Fig. 7*b*. Although the $T$ filter reduced the supremal value of the maximim singular value by a factor of about 3, the plot is still far from flat indicating that there is still scope for improvement. This improvement is afforded by the $Q$ parameter which (with the exception of the very low frequencies) gave a flat frequency response with a supremal value which is a factor of about 10 better than that achieved by the $T$ filter. The lack of flatness at very low frequencies is a direct consequence of the introduction of integral action through the use of $\Delta(z) = 1 - z^{-1}$; to avoid this difficulty one must use a nonunit weight $W(z)$ in the cost of eqn. 45.

## 7 Conclusions

In this paper we have shown how it is possible to extend SGPC together with its advantages from the scalar to the multivariable case. The resulting algorithm, MSGPC, has guaranteed stability (under the mild constraint

$n - \mu \geqslant \max\ \{\delta(B), \delta(A)\}$ and is computationally simple. Furthermore it provides a systematic way of using all the available degrees of freedom for the optimisation of robustness properties.

## 8 References


1 CLARKE, D.W., MOHTAHDI, C., and TUFFS, P.S.: 'Generalized predictive control: Part 1: The basic algorithm', *Automatica*, 1987, **23**, (2), pp. 137–148

2 CLARKE, D.W., MOHTAHDI, C., and TUFFS, P.S.: 'Generalized predictive control. Part 2: Extensions and interpretations', *Automatica*, 1987, **23**, (2), pp. 149–160

3 SHAH, S.L., MOHTAHDI, C., and CLARKE, D.W.: 'Multivariable adaptive control without prior knowledge of the delay matrix', *Syst. Control Lett.*, 1987, **9**, pp. 295–306

4 CLARKE, D.W., and SCATTOLINI, R.: 'Constrained receding-horizon predictive control', *IEE Proc. D*, 1991, **138**, (4), pp. 347–354

5 MOSCA, E., LEMOS, J.M., and ZHANG, J.: 'Stabilising I/O receding-horizon control'. Proceedings of 29th IEEE CDC

6 KOUVARITAKIS, B., ROSSITER, J.A., and CHANG, A.O.T.: 'Stable generalized predictive control: an algorithm with guaranteed stability', *IEE Proc. D*, 1992, **139**, (4), pp. 349–362

7 KAILATH, T.: 'Linear systems' (Prentice-Hall, 1980)

8 DOYLE, J., CHU, C.C., FRANCIS, B., KHARGONEKAR, P., and STEIN, G.: 'Advances in multivariable control — lecture notes' (ONR/Honeywell Workshop, 1984)

9 TREFETHEN, L.N.: 'Near circularity of the error curve in complex Chebyshev approximation', *J. Approximation Theory*, 1981, **31**, pp. 344–367

10 LAWSON, C.L.: 'Contributions to the theory of linear least maximum approximations'. MS thesis, University of California, Los Angeles, USA, 1961