Improved algorithm for multivariable stable generalised predictive control

B. Kouvaritakis
J.A. Rossiter
J.R. Gossner

Indexing terms: Algorithm, Generalised predictive control, Stabilising loop

Abstract: The use of an inner stabilising loop can be deployed to derive a generalised predictive control law with guaranteed stability, and can also be employed in the extension of this control law to the multivariable case. This approach affords useful insight into the mechanism that guarantees stability, but the development of this work is unnecessarily complicated. Recent work has overcome this problem in the scalar case, and in this note, the results are extended to derive a simple and efficient implementation of the stable generalised predictive control law for multivariable systems.

1 Introduction

Recent algorithms [1–3] bestow onto generalised predictive control (GPC) [4] the attribute of guaranteed stability and have been shown to produce equivalent control laws [5]. However, they differ in implementation, and here, we shall concentrate on stable generalised predictive control (SGPC) [2], in particular, because of its advantageous numerical properties [5]. As with GPC [6], the extension of SGPC to the multivariable case is straightforward [12], but the deployment of an inner stabilising loop (though useful in the derivation of finite impulse responses, and therefore expeditious in the proof of stability), results in prediction equations which are more involved than is necessary. Recent work has illustrated that the unifying factor in all GPC algorithms with guaranteed stability is the use of endpoint constraints and has shown [7–9] that this insight can be used for the systematic development of a numerically efficient SGPC algorithm in the scalar case. Here, we extend these ideas to the case of systems with multiple inputs/outputs and propose an appropriate multivariable stable generalised predictive control (MSGPC) algorithm.

IEE Proceedings online no. 19971171

Paper first received 29th May and in revised form 13th December 1996

B. Kouvaritakis and J.R. Gossner are with the Department of Engineering Science, Parks Road, Oxford OX1 3PJ, UK

J.A. Rossiter is with Loughborough University, Department of Mathematical Sciences, Leicestershire LE11 3TU, UK

2 Multivariable algorithm

Let the left and right coprime polynomial factorisation of a multivariable transfer function model with inputs $u^{(i)}$ and outputs $y^{(i)}$ for $i = 1, 2, ..., m$ be

$$y_t = z^{-1}[A^L(z)]^{-1}B^L(z)u_t = z^{-1}B^R(z)[A^R(z)]^{-1}u_t$$
$$y_t = \begin{bmatrix} y_t^{(1)} \\ \vdots \\ y_t^{(m)} \end{bmatrix}, \quad u_t = \begin{bmatrix} u_t^{(1)} \\ \vdots \\ u_t^{(m)} \end{bmatrix} \tag{1}$$

where $A^L(z)$, $B^L(z)$, $A^R(z)$, $B^R(z)$ are nth, (n–1)th, nth, (n–1)th order matrix polynomials in the delay operator, z^{-1}, such that for a general $M(z)$:

$$M(z) = M_0 + M_1 z^{-1} + \cdots + M_{n_M} z^{-n_M} \tag{2}$$

The degree of a polynomial will be indicated as $\delta\{.\}$ (e.g. $\delta\{M(z)\} = n_M$) and is the highest negative power z with nonzero coefficient.

The aim of stable predictive control strategies is to minimise a performance index, J, subject to input/state constraints and subject to a set of endpoint constraints which ensure stability by causing J to behave like a stable Lyapunov function [10]. At time $t = 0$, the cost J is to be minimised over the future control increments Δu_0, Δu_1, ... and is given by

$$J = \sum_{i=1}^{n_y} \|e_i\|_2^2 + \lambda \sum_{i=0}^{n_u-1} \|\Delta u_i\|_2^2 \tag{3}$$

where $\mathbf{e}_i = \mathbf{r}_i - \mathbf{y}_i$ denotes the predictive output error vectors ($\mathbf{r}_i$ denotes the future setpoint vectors), n_y is the output horizon, n_u is the input horizon, λ is the control weighting and $\Delta\mathbf{u}_i = \mathbf{u}_i - \mathbf{u}_{i-1}$. For simplicity and without loss of generality, it will be assumed that $\mathbf{r}_i = \mathbf{r}_0$ where $\mathbf{r}_0$ is a constant vector. Then the endpoint constraints for SGPC can be written as

$$y_i = r_0, \quad i > n_y; \quad \Delta u_i = 0, \quad i \geq n_u \tag{4}$$

and typical input constraints are

$$\left.\begin{matrix} \underline{u}^{(j)} \leq u_i^{(j)} \leq \overline{u}^{(j)} \\ \Delta\underline{u}^{(j)} \leq \Delta u_i^{(j)} \leq \Delta\overline{u}^{(j)} \end{matrix}\right\} \forall i \geq 0, \quad j = 1, \ldots, m \tag{5}$$

Earlier work met the endpoint constraints of eqn. 5 in an indirect manner, namely through the use of a stabilising (dead-beat) inner loop. Here, we adopt a direct polynomial approach which uses information on past values of inputs/outputs and the future setpoint to define the class of future (predicted) inputs/outputs which satisfy eqn. 5; a byproduct of this approach is the explicit characterisation of all the available degrees of freedom in the minimisation of the cost J.

It is convenient (and common practice in model based predictive control) to replace the model of eqn. 1 with the incremental model:

$$D^L(z)y_t = z^{-1}B^L(z)\Delta u_t; \quad D^L(z) = A^L(z)\Delta(z)$$
$$D^R(z) = A^R(z)\Delta(z); \quad \Delta(z) = 1 - z^{-1} \qquad (6)$$

Next, define the Toeplitz/Hankel matrices, C_M, H_M, for the polynomial matrix, $M(z)$, as

$$C_M = \begin{bmatrix} M_0 & 0 & \cdots & \cdots & 0 \\ M_1 & M_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ M_{n_M} & \vdots & \vdots & \vdots & 0 \\ \cdots & \cdots & \cdots & \cdots & M_0 \end{bmatrix}$$
$$H_M = \begin{bmatrix} M_1 & M_2 & \cdots & M_{n_M} \\ M_2 & \cdots & M_{n_M} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ M_{n_M} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \qquad (7)$$

where the dimensions of C_M, H_M are as implied by the context. Simulating the incremental model of eqn. 7 forward in time (starting from $t = 0$), we get the input/output prediction equation:

$$C_{D^L}Y = C_{B^L}\Delta U + P$$
$$P = H_{B^L}\Delta U_{past} - H_{D^L}Y_{past} \qquad (8)$$

where the output/input prediction and past vectors are defined as

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \end{bmatrix} \qquad \Delta U = \begin{bmatrix} \Delta u_0 \\ \Delta u_1 \\ \vdots \end{bmatrix}$$
$$Y_{past} = \begin{bmatrix} y_0 \\ \vdots \\ y_{-n} \end{bmatrix} \qquad \Delta U_{past} = \begin{bmatrix} \Delta u_{-1} \\ \vdots \\ \Delta u_{-n+1} \end{bmatrix} \qquad (9)$$

Eqn. 8 can be rewritten in z-transform form as

$$D^L(z)y(z) = B^L(z)\Delta u(z) + p(z)$$
$$y(z) = y_1 + y_2z^{-1} + \cdots$$
$$\Delta u(z) = \Delta u_0 + \Delta u_1 z^{-1} + \cdots$$
$$p(z) = [I, Iz^{-1}, \ldots, Iz^{-n+1}]P \qquad (10)$$

Theorem 2.1: The endpoint constraints of eqn. 4 are met if, and only if

$$\Delta u(z) = D^R(z)c(z) + \psi(z)$$
$$c(z) = c_0 + c_1z^{-1} + \cdots + c_{n_c-1}z^{-n_c+1}$$
$$n_c \leq \min\{n_u - n - 1, n_y - n + 1\} \qquad (11)$$

The corresponding prediction equation for the error is

$$e(z) = -B^R(z)c(z) + \phi(z) \qquad (12)$$

Furthermore, $\psi(z)$, $\phi(z)$ are the minimal order vector polynomial solutions to the diophantine equation:

$$D^L(z)\phi(z) + B^L(z)\psi(z) = q(z)$$
$$q(z) = A^L(z)r_0 - p(z) \qquad (13)$$

Proof: Noting that $\mathbf{r}(z) = \mathbf{r}_0/\Delta(z)$ and subtracting $D^L(z)\mathbf{r}(z)$ from both sides of eqn. 10a gives:

$$-D^L(z)e(z) = B^L(z)\Delta u(z) + p(z) - \frac{D^L(z)r_0}{\Delta(z)}$$
$$\Leftrightarrow D^L(z)e(z) + B^L(z)\Delta u(z) = q(z) \qquad (14)$$

where use has been made of eqn. 13b and the fact that $D^L(z) = \Delta(z)A^L(z)$.

Now endpoint constraints (eqn. 4) will be satisfied if, and only if, $\mathbf{e}(z)$ and $\Delta\mathbf{u}(z)$ are vector polynomials with the further restriction that

$$\delta\{e(z)\} \leq n_y - 1; \qquad \delta\{\Delta u(z)\} \leq n_u - 1 \qquad (15)$$

Thus, eqn. 14b must be a polynomial diophantine equation and, given that $A^L(z)$ and $B^L(z)$ (and hence $D^L(z)$ and $B^L(z)$) are left coprime, this diophantine equation will have a solution which satisfies eqn. 15 if, and only if, [11]:

$$n_y \geq n - 1; \qquad n_u \geq n + 1 \qquad (16)$$

The general form of the solution is as given in eqn. 11a and eqn. 12. This follows from the fact that eqn. 14b is a linear equation and therefore its solution will comprise a particular solution, which can be chosen to be the minimal order solution, $\mathbf{e}(z) = \phi(z)$ and $\Delta\mathbf{u}(z) = \psi(z)$, plus the solution to eqn. 14b for $\mathbf{q}(z) = \mathbf{0}$, which is given by $\mathbf{e}(z) = -B^R(z)\mathbf{c}(z)$ and $\Delta\mathbf{u}(z) = D^R(z)\mathbf{c}(z)$ for $\mathbf{c}(z)$ any arbitrary polynomial.

However, since $\phi(z)$ and $\psi(z)$ have been chosen to be minimal order solutions to eqn. 13, it follows from eqns. 11a and 12 that

$$\delta\{e(z)\} = (n-1) + (n_c - 1)$$
$$\delta\{\Delta u(z)\} = (n+1) + (n_c - 1) \qquad (17)$$

and this, together with eqn. 15, imply the condition of the theorem given in eqn. 11c. It is noted that n_c must be non-negative and this, in combination with eqn. 11c, ensures that condition (eqn. 16) will be satisfied automatically. □

Transforming the class of vector polynomials which satisfy endpoint constraints (eqn. 4) into prediction equation form yields

$$E = -\Gamma_{B^R}C + \Phi; \qquad \Delta U = \Gamma_{D^R}C + \Psi \qquad (18)$$

where $\mathbf{E}$ and $\mathbf{C}$ are the vectors of future errors and degrees of freedom, respectively, and are defined in a manner analoguous to $\mathbf{Y}$; Γ_M is the matrix formed out of the first n_c block columns of C_M; and Φ, Ψ are defined in a manner analogous to eqn. 9. Then the cost J of eqn. 3 can be written as

$$\begin{aligned} J =& E^TE + \lambda\Delta U^T\Delta U \\ =& C^T(\Gamma_{B^R}^T\Gamma_{B^R} + \lambda\Gamma_{D^R}^T\Gamma_{D^R})C \\ &- C^T(\Gamma_{B^R}^T\Phi - \lambda\Gamma_{D^R}^T\Psi) \\ &+ \Phi^T\Phi + \lambda\Psi^T\Psi \end{aligned} \qquad (19)$$

which is quadratic in $\mathbf{C}$ and can be minimised subject to input constraints (eqn. 5).

The receding horizon MSGPC algorithm is then defined as follows:

2.1 Agorithm 2.1 (MSGPC)

Minimise performance index (eqn. 19) with respect to $\mathbf{C}$ and subject to input constraints (eqn. 5). Of the optimal input increments $\Delta\mathbf{U}$, implement $\Delta\mathbf{u}_0$ and then redo the optimisation with new plant data at the next time instant.

3 Numerical example

To clarify our notation and to illustrate the form of the vector, matrices, polynomial vectors and matrices used in this paper, we include a simple numerical example. Thus consider a multivariable system whose transfer function matrix is given in terms of the right factorisation:

$$A^R(z) = A_0^R + A_1^R z^{-1} + A_2^R z^{-2}$$

$$B^R(z) = B_0^R + B_1^R z^{-1}$$

$$A_0^R = B_0^R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = A_2^R$$

$$A_1^R = \begin{bmatrix} -2.5 & 0 \\ 0 & -3.3333 \end{bmatrix}$$

$$B_1^R = \begin{bmatrix} -1.5 & 0.8 \\ 0.6 & 2.5 \end{bmatrix}$$

Then, solving the identity $A^L(z)B^R(z) = B^L(z)A^R(z)$, we get the left transfer function matrix factorisation as

$$A^L(z) = A_0^L + A_1^L z^{-1} + A_2^L z^{-2}$$

$$B^L(z) = B_0^L + B_1^L z^{-1}$$

$$A_0^L = B_0^L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$A_1^L = \begin{bmatrix} -2.2248 & -0.1517 \\ 1.7218 & -3.6086 \end{bmatrix}$$

$$A_2^L = \begin{bmatrix} 0.8158 & -0.0018 \\ -1.0566 & 1.2280 \end{bmatrix}$$

$$B_1^L = \begin{bmatrix} -1.2248 & 0.6483 \\ 2.3218 & 2.2248 \end{bmatrix}$$

and multiplying $A^L(z)$ by $(1 - z{-}1)$ we derive:

$$D^L(z) = D_0^L + D_1^L z^{-1} + D_2^L z^{-2} + D_3^L z^{-3}$$

$$D_0^L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$D_1^L = \begin{bmatrix} -3.2248 & -0.1517 \\ 1.7218 & -4.6085 \end{bmatrix}$$

$$D_2^L = \begin{bmatrix} 3.0406 & 0.1500 \\ -2.7784 & 4.8366 \end{bmatrix}$$

$$D_3^L = \begin{bmatrix} -0.8158 & 0.0018 \\ 1.0566 & -1.2280 \end{bmatrix}$$

For simplicity, we shall consider the calculation of the optimal $\mathbf{C}$ at $\mathbf{t} = 0$ only when the initial conditions are assumed to be zero. As a result, the vector $\mathbf{p}(z)$ is zero, and for a setpoint vector $\mathbf{r}_0 = [0\ 1]^T$ the right-hand side and the solutions of the diophantine equation of Theorem 2.1 are given as

$$q(z) = A^L(z)r_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -0.1517 \\ -3.6086 \end{bmatrix} z^{-1} + \begin{bmatrix} -0.0018 \\ 1.228 \end{bmatrix} z^{-2};$$

$$\phi(z) = \begin{bmatrix} -0.3587 \\ 0.7758 \end{bmatrix}$$

$$\psi(z) = \begin{bmatrix} 0.3587 \\ 0.2242 \end{bmatrix} + \begin{bmatrix} -0.8969 \\ -0.7474 \end{bmatrix} z^{-1} + \begin{bmatrix} 0.3587 \\ 0.2242 \end{bmatrix} z^{-2}$$

Stacking the vector coefficients of $\phi(z)$, $\psi(z)$ we can form the vectors Φ, Ψ of eqn. 18 as

$$\Phi = [-0.3587 \quad 0.7758]^T$$

$$\Psi = [0.3587\ 0.2242\ -0.8969\ -0.7474\ 0.3587\ 0.2242]^T$$

Taking, for simplicity, n_c to be 1 would make $\mathbf{C}$ a two-dimensional vector, and for $n_y = 2$, $n_u = 4$ the matrices $\Gamma_B R$, $\Gamma_D R$ assume the dimensions 4 × 2 and 8 × 2, respectively, and are given by

$$\Gamma_{B^R} = \begin{bmatrix} B_0^R \\ B_1^R \end{bmatrix}; \qquad \Gamma_{D^R} = \begin{bmatrix} D_0^R \\ D_1^R \\ D_2^R \\ D_3^R \end{bmatrix}$$

For $\lambda = 1$, these matrices together with Φ, Ψ give the optimal solution for $\mathbf{C}$ and the corresponding optimal cost as

$$C_{opt} = S^{-1}v = \begin{bmatrix} -0.169 \\ -0.076 \end{bmatrix}$$

$$J_{opt} = C^T SC - 2C^T v + \Phi^T \Phi + \lambda \Psi^T \Psi = 1.3091$$

$$S = \Gamma_{B^R}^T \Gamma_{B^R} + \lambda \Gamma_{D^R}^T \Gamma_{D^R} = \begin{bmatrix} 30.11 & 0.3 \\ 0.3 & 47.4456 \end{bmatrix}$$

$$v = \Gamma_{B^R}^T \Phi - \lambda \Gamma_{D^R}^T \Psi = \begin{bmatrix} -5.112 \\ -3.6587 \end{bmatrix}$$

The corresponding optimal predictions for e and δu can be derived from eqn. 18 as

$$E_{opt} = -\Gamma_{B^R} C_{opt}$$

$$= [-0.1897 \quad 0.8518 \quad -0.1927 \quad 0.2915]^T$$

$$\Delta U_{opt} = \Gamma_{D^R} C_{opt} + \Psi$$

$$= [0.1897\ 0.1482\ -0.3053\ -0.4179\ -0.2328\ -0.1053\ 0.169\ 0.076]^T$$

The first element of $\Delta \mathbf{U}_{opt}$ is used to calculate the current optimal control input, $\mathbf{u}_0 = 0.1897 + u{-}1 = 0.1897$, and the whole procedure is repeated at the next sampling instant.

4 Conclusion

This note gives an algorithm for computing the MSGPC control law based on the simple derivation of input/output predictions which meet endpoint constraints known to be the unifying key to predictive control algorithms with guaranteed stability. The development improves in clarity and efficiency upon the algorithm of [12].

5 References

1 CLARKE, D.W., and SCATTOLINI, R.: 'Constrained receding horizon predictive control', *IEE Proc. D, Control Theory Appl. (UK)*, 1991, **138**, pp. 347–354

2 KOUVARITAKIS, B., ROSSITER, J.A., and CHANG, A.O.T.: 'Stable generalized predictive control', *IEE Proc. D, Control Theory Appl. (UK)*, 1992, **139**, pp. 349–362

3 MOSCA, E., and ZHANG, J.: 'Stable redesign of predictive control', *Automatica*, 1992, **28**, pp. 1229–1233

4 CLARKE, D.W., MOHTADI, C., and TUFFS, P.S.: 'Generalized predictive control', *Automatica*, 1987, **23**, (1, 2), pp. 137–160

5 ROSSITER, J.A., and KOUVARITAKIS, B.: 'Numerical robustness and efficiency of generalized predictive control algorithms with guaranteed stability', *IEE Proc. D, Control Theory Appl. (UK)*, 1994, **141**, pp. 154–162

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

7 GOSSNER, J.R., KOUVARITAKIS, B., and ROSSITER, J.A.: 'Cautious stable predictive control: a guaranteed stable predictive control algorithm with low input activity and good robustness'. Proceedings of the 3rd IEEE Mediterranean symposium on *New directions in control and automation*, Cyprus, 1995, Vol. 2, pp. 243–250

8 ROSSITER, J.A.: 'GPC controllers with guaranteed stability and mean-level control of unstable plant'. Proceedings of the 33rd CDC, Florida, 1994, pp. 3579–3580

9 KEERTHI, S.S., and GILBERT, E.G.: 'Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: stability and moving-horizon approximations', *J. Optimization Theory Appl.*, 1988, **57**, pp. 265–293

10 ROSSITER, J.A., and KOUVARITAKIS, B.: 'Constrained stable generalized predictive control', *IEE Proc. D, Control Theory Appl. (UK)*, 1993, **140**, pp. 243–254

11 KUCERA, V.: 'Discrete linear control' (John Wiley, 1979)

12 KOUVARITAKIS, B., and ROSSITER, J.A.: 'Multivariable stable generalized predictive control', *IEE Proc. D, Control Theory Appl. (UK)*, 1993, **140**, pp. 364–372