Infinite Horizon Stable Predictive Control

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

Abstract—**Terminal constraints guarantee the stability of predicted trajectories and form the basis of predictive control algorithms with guaranteed stability. Earlier work in the literature uses terminal constraints which define sufficient but not necessary conditions for the stability of predicted trajectories. In this paper we deploy conditions which are both necessary and sufficient and hence release more degrees of freedom for optimizing performance and/or meeting constraints. Also an alternative means of computing the implied infinite horizon GPC cost, avoiding the need for solving a Lyapunov equation, is presented.**

I. Introduction

Model-based predictive control (MBPC) (e.g., [1]–[6]), and in particular generalized predictive control (GPC) [7], has become an increasingly popular control strategy because of the apparent common sense of the control objective, that is the minimization of predicted tracking errors and control activity, and because it is a natural framework within which to handle system constraints. It is primarily this feature of MBPC that makes it an attractive alternative to LQ control; because of an implicit restriction in the degrees of freedom, MBPC can be thought as a suboptimal LQ strategy, but this restriction affords significant computational advantages and renders the problem of handling physical limits tractable. MBPC is guaranteed stable [8]–[11] if a terminal constraint is employed (e.g., that the predicted inputs/outputs become fixed beyond the input/output horizons, respectively); however, this constraint is sufficient but not necessary, and thus the derived control laws can be highly tuned, have poor robustness, and lead to constraint violations (with consequent loss of the stability guarantee) [12], [13]. The terminal condition can be relaxed [12]–[15] by only requiring the output to converge *asymptotically* to its desired value, but this is still only a sufficient condition as the input is assumed to become fixed beyond the input horizon. Necessary and sufficient conditions for stable predictions would invoke an *asymptotic* convergence requirement only on both output and input. An algorithm [14] using such conditions has been developed, but the proposed cost penalizes filtered rather than actual error predictions and hence may lead to suboptimal performance. This paper remedies that problem.

In Section II, we derive the necessary and sufficient conditions for stable input/output prediction pairs and deploy these in an infinite horizon GPC control law. Our approach gives an *explicit* characterization of the class of input/output predictions which meet asymptotic terminal constraints; this is in contrast to earlier work which satisfied terminal constraints only implicitly (through the use of equality constraints) thus providing no clear handle on the degrees of freedom. Infinite horizons do not lead to practically implementable control laws, so in Section III two methods are considered for overcoming this problem. The first is an extension of ideas in [12], and the second is an alternative that avoids the need for solving a Lyapunov equation and which, for high order models, can be more efficient. The results of the paper are illustrated in Section IV using numerical examples which highlight the advantages of the proposed algorithm.

It is noted that all the results in this paper are applicable to the case with input constraints, and in fact because necessary and sufficient conditions for stable predictions are employed, the approach is less likely than earlier approaches to run into feasibility difficulties. However, for reasons of clarity and brevity the relevant results are not given here. The implementation is as standard for predictive control with the exception that the infinite horizons require the use of methods (e.g., [15], [16]) to reduce an infinite constraint horizon to a finite one.

Manuscript received September 22, 1995; revised January 28, 1996 and February 21, 1996.

J. A. Rossiter is with the Department of Mathematical Sciences, Loughborough University, Leicestershire LE11 3TU UK. (e-mail: J.A.Rossiter@lut.ac.uk).

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

Publisher Item Identifier S 0018-9286(96)05784-4.

II. Background

A. Notation

In this paper we shall make use of the following matrix notation: O is a matrix (or vector) of zeros whose size is conformal to its use, the operator $\hat{(.)}$ reverses the column order, and $\mathbf{m}$ is the vector of coefficients of the n_mth order polynomial $m(z^{-1})$ e.g.,

$$m(z^{-1}) = m_0 + \cdots + m_{n_m} z^{-n_m}; \quad \mathbf{m} = [m_0, m_1, \cdots, m_{n_m}]^T. \tag{1}$$

To give a formal definition of stable input/output predictions we introduce the factorization

$$m(z^{-1}) = m^+(z^{-1}) m^-(z^{-1}) \tag{2}$$

where the roots of $m^+(z^{-1})$ lie outside or on the unit circle, and the roots of $m^-(z^{-1})$ lie strictly inside the unit circle. Furthermore, define n_{m^+}, n_{m^-} to be the orders of $m^+(z^{-1}), m^-(z^{-1})$, respectively. For convenience, the constant term of $m^-(z^{-1})$ will be taken to be one.

The lower triangular Toeplitz matrix C_m is the $n_y \times n_y$ matrix whose i, j element is m_{i-j}, H_m is the $n_y \times n_m$ Hankel matrix whose i, j element is m_{i+j-1}. For example their structure is

$$H_m = \begin{bmatrix} m_1 & \cdots & m_{n_m-1} & m_{n_m} \\ m_2 & \cdots & m_{n_m} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ m_{n_m} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix}$$
$$C_m = \begin{bmatrix} m_0 & 0 & \cdots & 0 \\ m_1 & m_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & m_0 \end{bmatrix}. \tag{3}$$

Let the system model be given in terms of the z-transform transfer function equation

$$y(z^{-1}) = z^{-1}\frac{b(z^{-1})}{a(z^{-1})}u(z^{-1}) = z^{-1}\frac{b(z^{-1})}{A(z^{-1})}\Delta u(z^{-1})$$
$$A(z^{-1}) = a(z^{-1})\Delta(z^{-1}) \tag{4}$$

where $a(z^{-1}) = 1 + \cdots + a_{n_a} z^{-n_a}$, $b(z^{-1}) = b_0 + \cdots + b_{n_b} z^{-n_b}$ ($a(z^{-1}), b(z^{-1})$ assumed coprime), $\Delta(z^{-1}) = 1 - z^{-1}$, and $\Delta u(z^{-1})$ is the z-transform of Δu_t ($\Delta u_t = u_t - u_{t-1}$). For clarity here we present the disturbance-free case. The development incorporating disturbances is straightforward, but more complicated, and is given in Appendix A. Converting (4) into a difference equation and simulating forward over n_y samples, we derive the prediction equation

$$\underset{\rightarrow}{\mathbf{y}} = C_A^{-1}[C_b \Delta \underset{\rightarrow}{\mathbf{u}} + \mathbf{p}]; \quad \mathbf{p} = H_b \Delta \underset{\leftarrow}{\mathbf{u}} - H_A \underset{\leftarrow}{\mathbf{y}}$$

$$\underset{\rightarrow}{\mathbf{y}} = \begin{bmatrix} y_{t+1} \\ \vdots \\ y_{t+n_y} \end{bmatrix}; \quad \Delta \underset{\rightarrow}{\mathbf{u}} = \begin{bmatrix} \Delta u_t \\ \vdots \\ \Delta u_{t+n_u-1} \end{bmatrix} \tag{5}$$

$$\underset{\leftarrow}{\mathbf{y}} = \begin{bmatrix} y_t \\ \vdots \\ y_{t-n_a} \end{bmatrix}; \quad \Delta \underset{\leftarrow}{\mathbf{u}} = \begin{bmatrix} \Delta u_{t-1} \\ \vdots \\ \Delta u_{t-n_b} \end{bmatrix}.$$

Let $y(z^{-1})$, $\Delta u(z^{-1})$ be the z-transforms of the predicted values only of $y, \Delta u$, $p(z^{-1})$ is the polynomial corresponding to $\mathbf{p}$ as per (1), then the z-transform equivalent of (5) is

$$A(z^{-1})y(z^{-1}) = b(z^{-1})\Delta u(z^{-1}) + p(z^{-1})$$
$$\Rightarrow \quad y(z^{-1}) = \frac{b(z^{-1})\Delta u(z^{-1}) + p(z^{-1})}{A(z^{-1})}. \tag{6}$$

Equation (6) could be derived directly from (4), but the notation used here will prove useful in the sequel. It now follows that the z-transform of the error, $e(z^{-1}) = r(z^{-1}) - y(z^{-1})$, where $r(z^{-1})$ is the z-transform of the setpoint trajectory and is given by

$$e(z^{-1}) = \frac{r}{1 - z^{-1}} - y(z^{-1}) = \frac{q(z^{-1}) - b(z^{-1})\Delta u(z^{-1})}{A(z^{-1})}$$
$$q(z^{-1}) = a(z^{-1})r - p(z^{-1}). \tag{7}$$

Note that for clarity of presentation, it has been assumed in (7) that all future values of the set-point trajectory are the same and equal to r. The treatment for general set-point trajectories is more involved, and for completeness a brief account of it is given in Appendix B.

B. Necessary and Sufficient Conditions for the Stability of Input/Output Predictions

We now derive necessary and sufficient conditions for the stability of the prediction pair $e(z), \Delta u(z)$. Given that $a(z^{-1}) = a^+(z^{-1})a^-(z^{-1})$, $b(z^{-1}) = b^+(z^{-1})b^-(z^{-1})$, $A^+(z^{-1}) = a^+(z^{-1})\Delta(z^{-1})$, then from (7a), a necessary and sufficient condition for the stability of $e(z^{-1})$ is

$$q(z^{-1}) - b(z^{-1})\Delta u(z^{-1}) = A^+(z^{-1})\phi(z^{-1}); \phi(z^{-1}) \in \Pi \tag{8}$$

where Π is the set of polynomials of either finite order or with coefficients which converge to zero. Solving (8) for $\Delta u(z^{-1})$ gives

$$\Delta u(z^{-1}) = \frac{q(z^{-1}) - A^+(z^{-1})\phi(z^{-1})}{b^+(z^{-1})b^-(z^{-1})} \tag{9}$$

which implies that a necessary and sufficient condition for $\Delta u(z^{-1})$ to be stable is that

$$q(z^{-1}) - A^+(z^{-1})\phi(z^{-1}) = b^+(z^{-1})\psi(z^{-1}); \quad \psi(z^{-1}) \in \Pi. \tag{10}$$

Hence, all the stable input/output prediction pairs can be derived by solving (10) for $\phi(z^{-1}), \psi(z^{-1})$ and substituting the solutions back into (8) and (9) to find $e(z^{-1}), \Delta u(z^{-1})$. It is clear that $\phi(z^{-1}) = -b^+(z^{-1})c(z^{-1})$ and $\psi(z^{-1}) = A^+(z^{-1})c(z^{-1})$ for $c(z^{-1}) = c_0 + \cdots + c_{n_c}z^{-n_c}$, an arbitrary polynomial in Π satisfies (10) for $q(z^{-1}) = 0$, and hence the class of solutions to (10) is

$$\phi(z^{-1}) = \phi_p(z^{-1}) - b^+(z^{-1})c(z^{-1})$$
and
$$\psi(z^{-1}) = \psi_p(z^{-1}) + A^+(z)c(z^{-1}); \quad c(z^{-1}) \in \Pi \tag{11}$$

where $\phi_p(z^{-1}), \psi_p(z^{-1})$ denote a particular solution and $c(z^{-1})$ characterizes all the available degrees of freedom. For convenience we take $n_{\phi_p} = n_{b+} - 1$ and $n_{\psi_p} = \max[n_{A+} - 1, n_A - n_{b+}]$, the coefficients can then be easily computed from the vectors ϕ_p, ψ_p defined via

$$[\Gamma'_{A+}, \Gamma'_{b+}]\begin{bmatrix} \phi_p \\ \psi_p \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ O \end{bmatrix} \tag{12}$$

where $\Gamma'_{A+}, \Gamma'_{b+}$ are defined from the first $n_{\phi_p}+1, n_{\psi_p}+1$ columns of the $(n_{\phi_p}+n_{\psi_p}+2)\times(n_{\phi_p}+n_{\psi_p}+2)$ Toeplitz matrices C_{A+}, C_{b+}; $\mathbf{q}$ is the vector of coefficients of the polynomial $q(z^{-1})$ given in (7). Equation (12) implies

$$\begin{bmatrix} \phi_p \\ \psi_p \end{bmatrix} = [\Gamma'_{A+}, \Gamma'_{b+}]^{-1}\begin{bmatrix} \mathbf{q} \\ O \end{bmatrix} = \begin{bmatrix} P_1 & P_3 \\ P_2 & P_4 \end{bmatrix}\begin{bmatrix} \mathbf{q} \\ O \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}\mathbf{q} \tag{13}$$

where $P_i, i = 1, 2, 3, 4$ are matrices conformal to the dimensions of ϕ_p, ψ_p.

Theorem 2.1: The predicted error and corresponding predicted control increment trajectories, defined in (7), are stable if and only if the z-transform of the future Δu's is given by

$$\Delta u(z^{-1}) = \frac{A^+(z^{-1})c(z^{-1}) + \psi_p(z^{-1})}{b^-(z^{-1})}. \tag{14}$$

The corresponding z-transform of the predicted error trajectory is then given by

$$e(z^{-1}) = \frac{-b^+(z^{-1})c(z^{-1}) + \phi_p(z^{-1})}{a^-(z^{-1})} \tag{15}$$

where $c(z^{-1})$ is any arbitrary polynomial in Π and $\psi_p(z^{-1}), \phi_p(z^{-1})$ are the polynomials whose coefficients are given by the elements of the vectors ψ_p, ϕ_p of (13).

Proof: Equations (14) and (15) follow directly from the substitution of (11) into (7)–(9). Sufficiency and necessity follow from the arguments given in the derivation of (8) and (10). □

Upon inverse z-transformation of (14) and (15), we derive the prediction equations

$$\underset{\rightarrow}{\mathbf{e}} = \Gamma_{\frac{1}{a^-}}[-\Gamma_{b+}\mathbf{c} + P_1\mathbf{q}]; \quad \Delta \underset{\rightarrow}{\mathbf{u}} = \Gamma_{\frac{1}{b^-}}[\Gamma_{A+}\mathbf{c} + P_2\mathbf{q}]. \tag{16}$$

$\Gamma_{\frac{1}{a^-}}, \Gamma_{\frac{1}{b^-}}$ comprise the first $n_{b+} + n_c + 1$ and $n_{A+} + n_c + 1$ columns of the infinite Toeplitz matrices $C_{\frac{1}{a^-}}, C_{\frac{1}{b^-}}$, respectively, and $\mathbf{c} = [c_0, \cdots, c_{n_c}]^T$ is the vector of the degrees of freedom. Γ_{b+}, Γ_{A+} are defined in a manner analagous to $\Gamma'_{b+}, \Gamma'_{A+}$ but have different dimensions and comprise the first $n_c + 1$ columns of $n_{b+}+n_c+1, n_{A+}+n_c+1$ Toeplitz matrices C_{b+}, C_{A+}, respectively. It is assumed that P_1, P_2 are augmented with rows of zeros to make the dimensions conformal.

C. Nominal Stable Control Law

The infinite horizon stable predictive control (IHSPC) cost is given as

$$J = \sum_{i=1}^{\infty} [e_{t+i}^2 + \lambda \Delta u_{t+i-1}^2]. \tag{17}$$

Stability is guaranteed by virtue of the infinite horizon deployed in (17) which allows standard Lyapunov arguments to be used. Introducing prediction equations (16a), (16b) into J gives

$$J = [-\Gamma_{b+}\mathbf{c} + P_1\mathbf{q}]^T \Gamma_{\frac{1}{a^-}}^T \Gamma_{\frac{1}{a^-}}[-\Gamma_{b+}\mathbf{c} + P_1\mathbf{q}]$$
$$+ \lambda[\Gamma_{A+}\mathbf{c} + P_2\mathbf{q}]^T \Gamma_{\frac{1}{b^-}}^T \Gamma_{\frac{1}{b^-}}[\Gamma_{A+}\mathbf{c} + P_2\mathbf{q}] \tag{18}$$

which is minimized with respect to $\mathbf{c}$ by

$$\mathbf{c} = -\left(\Gamma_{b+}^T \Gamma_{\frac{1}{a^-}}^T \Gamma_{\frac{1}{a^-}} \Gamma_{b+} + \lambda \Gamma_{A+}^T \Gamma_{\frac{1}{b^-}}^T \Gamma_{\frac{1}{b^-}} \Gamma_{A+}\right)^{-1}$$
$$\times \left[-\Gamma_{b+}^T \Gamma_{\frac{1}{a^-}}^T \Gamma_{\frac{1}{a^-}} P_1\mathbf{q} + \lambda \Gamma_{A+}^T \Gamma_{\frac{1}{b^-}}^T \Gamma_{\frac{1}{b^-}} P_2\mathbf{q}\right]. \tag{19}$$

Only the first element of $\Delta \underset{\rightarrow}{\mathbf{u}}(\Delta u_t)$ is implemented, and the computation is repeated at the next sampling instant. Δu_t can be defined through (16b) and (7b) and (5b) as

$$\Delta u_t = c_0 + [1, 0, \cdots, 0]P_2(\mathbf{a}r + H_A \underline{\mathbf{y}} - H_b \Delta \underline{\mathbf{u}}). \tag{20}$$

It is clear that there are implementation difficulties associated with the control law of (19) as some of the matrices involved are infinite dimensional. Fortunately, these infinite dimensional matrices appear in product forms which are finite dimensional, and in the sequel we develop convenient and computationally efficient means of computing and minimizing J.

III. Infinite Horizon Stable Predictive Control

It is well known that the sum of squares of infinite stable sequences can be computed through the use of Lyapunov equations. This idea was deployed in [12] to minimize a GPC cost where *only* the output horizon was infinite; the input horizon was finite. Here we removed this limitation by deriving necessary and sufficient conditions (rather than sufficient conditions only) for stable input and output prediction pairs while keeping the number of degrees of freedom finite. Earlier work identified the future control increments as the degrees of freedom and therefore precluded the use of infinite input horizons. Here we show that Lyapunov techniques can be used to compute and minimize the cost J of (18), despite the use of infinite output and input horizons, and also propose an alternative implementation to the use of Lyapunov equations which can be more efficient.

A. Using a Lyapunov Equation to Sum an Infinite Sequence

The key to this approach is to find a means of summing the squares of a stable sequence derived from a transfer function $g(z^{-1}) = n(z^{-1})/d(z^{-1})$. This involves expressing the infinite sum in terms of the first $n_n + 1$ coefficients only. Thus, assume that $g(z^{-1}) = \sum_{i=0}^{\infty} g_i z^{-i}$, then using $d(z^{-1})g(z^{-1}) = n(z^{-1})$ for all $g_i, i > n_n$, the following recursive relationship holds:

$$\begin{bmatrix} g_i \\ g_{i+1} \\ \vdots \\ g_{i+n_d-1} \end{bmatrix} = M \begin{bmatrix} g_{i-n_d} \\ g_{i-n_d+1} \\ \vdots \\ g_{i-1} \end{bmatrix}; \quad M = C_d^{-1}\hat{H}_d \tag{21}$$

where C_d, H_d are the $n_d \times n_d$ Toeplitz and Hankel matrices of $d(z^{-1})$. Hence it can be seen that

$$\sum_{i=0}^{\infty} g_i^2 = \left(\sum_{i=0}^{n_n - n_d} g_i^2\right) + [g_{n_n-n_d+1} \cdots g_{n_n}] S \begin{bmatrix} g_{n_n-n_d+1} \\ g_{n_n-n_d+2} \\ \vdots \\ g_{n_n} \end{bmatrix}$$

$$S = \sum_{i=0}^{\infty} (M^T)^i M^i. \tag{22}$$

Hence the cost of (22) is easily evaluated as S can be computed from the Lyapunov equation

$$S = I + M^T S M. \tag{23}$$

B. IHSPC Using a Lyapunov Equation

The results of Section III-A can be used to evaluate the cost of (17) simply by taking the polynomials $n(z^{-1}), d(z^{-1})$ to be the numerator and denominator polynomials of (15) first and (14) second with orders $n_{b+} + n_c + 1$ and n_{a-}, and $\max[n_{A+} + n_c + 1, n_{v_p}]$ and n_{b-} respectively. Defining $\Gamma^{(1)}_{\frac{1}{a^-}}, \Gamma^{(2)}_{\frac{1}{a^-}}, \Gamma^{(1)}_{\frac{1}{b^-}}, \Gamma^{(2)}_{\frac{1}{b^-}}$ as appropriate partitions of the first $n_c + n_{b+} + 1, n_c + n_{A+} + 1$ rows of the matrices $\Gamma_{\frac{1}{a^-}}, \Gamma_{\frac{1}{b^-}}$ of (16), respectively,[1] gives

$$\begin{bmatrix} c_0 \\ \vdots \\ c_{n_{b+}+n_c-n_{a-}+1} \\ \hline c_{n_{b+}+n_c-n_{a-}+2} \\ \vdots \\ c_{n_{b+}+n_c+1} \end{bmatrix} = \begin{bmatrix} \Gamma^{(1)}_{\frac{1}{a^-}} \\ \hline \Gamma^{(2)}_{\frac{1}{a^-}} \end{bmatrix} (-\Gamma_{b+}\mathbf{c} + P_1\mathbf{q})$$

$$\begin{bmatrix} \Delta u_0 \\ \vdots \\ \Delta u_{n_{A+}+n_c-n_{b-}+1} \\ \hline \Delta u_{n_{A+}+n_c-n_{b}+2} \\ \vdots \\ \Delta u_{n_{A+}+n_c+1} \end{bmatrix} = \begin{bmatrix} \Gamma^{(1)}_{\frac{1}{b^-}} \\ \hline \Gamma^{(2)}_{\frac{1}{b^-}} \end{bmatrix} (\Gamma_{A+}\mathbf{c} + P_2\mathbf{q}). \tag{24}$$

Further define $M_a = C_{a^-}^{-1} H_{a^-}, M_b = C_{b^-}^{-1} H_{b^-}$ and solve for S_a, S_b from $S_a = I + M_a^T S_a M_a$ and $S_b = I + M_b^T S_b M_b$. Then, using result (22), the performance index (17) can be rewritten as

$$\begin{aligned} J &= (-\Gamma_{b+}\mathbf{c} + P_1\mathbf{q})^T B_1(-\Gamma_{b+}\mathbf{c} + P_1\mathbf{q}) \\ &\quad + \lambda(\Gamma_{A+}\mathbf{c} + P_2\mathbf{q})^T B_2(\Gamma_{A+}\mathbf{c} + P_2\mathbf{q}) \\ B_1 &= \left[\Gamma^{(1)T}_{\frac{1}{a^-}}\Gamma^{(1)}_{\frac{1}{a^-}} + \Gamma^{(2)T}_{\frac{1}{a^-}} S_a \Gamma^{(2)}_{\frac{1}{a^-}}\right] \\ B_2 &= \left[\Gamma^{(1)T}_{\frac{1}{b^-}}\Gamma^{(1)}_{\frac{1}{b^-}} + \Gamma^{(2)T}_{\frac{1}{b^-}} S_b \Gamma^{(2)}_{\frac{1}{b^-}}\right]. \end{aligned} \tag{25}$$

This is minimized for

$$\begin{aligned} \mathbf{c} &= -P^{-1}R\mathbf{q} \\ P &= \left[\Gamma_{b+}^T B_1 \Gamma_{b+} + \lambda \Gamma_{A+}^T B_2 \Gamma_{A+}\right] \\ R &= -\Gamma_{b+}^T B_1 P_1 + \lambda \Gamma_{A+}^T B_2 P_2. \end{aligned} \tag{26}$$

The first element of the optimum $\mathbf{c}$ can be substituted into (20) to define the control law.

C. Alternative Solution for IHSPC

The infinite dimensional matrices in (19) always appear in quadratic forms, i.e., $\Gamma^T_{\frac{1}{a^-}}\Gamma_{\frac{1}{a^-}}, \Gamma^T_{\frac{1}{b^-}}\Gamma_{\frac{1}{b^-}}$. However, it is easy to identify the coefficients of the of these matrices directly.

Lemma 3.1: Let $d(z^{-1})$ be a polynomial with all its roots inside the unit circle. Then the i, j coefficient, $i \geq j$ (the argument for $j > i$ is identical) of the matrix $\Gamma^T_{\frac{1}{d}}\Gamma_{\frac{1}{d}}$, is given by the coefficient of z^{i-j} in the Laurent series expansion about $z = 0$ of $f(z)$, where

$$f(z) = \frac{1}{d(z)d(z^{-1})}. \tag{27}$$

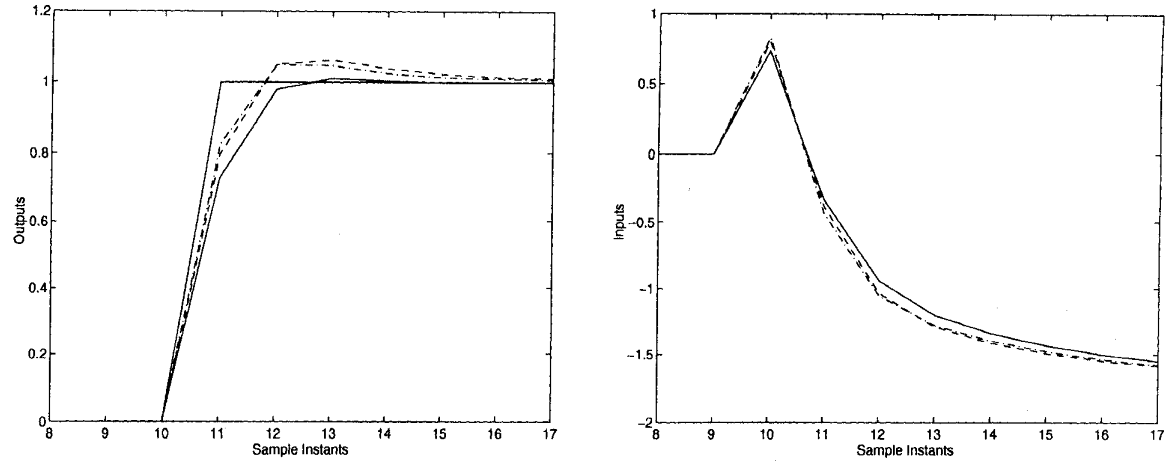
Proof: The jth column of $\Gamma_{\frac{1}{d}}$ contains the coefficients of a Taylor series expansion of $\frac{z^{-j+1}}{d(z^{-1})}$ in terms of z^{-1} (which is also the expansion of $\frac{z^{j-1}}{d(z)}$ in terms of z). Let these expansions with the corresponding $f(z)$ be

$$\frac{1}{d(z^{-1})} = \sum_{i=0}^{\infty} g_i z^{-i}; \qquad \frac{1}{d(z)} = \sum_{i=0}^{\infty} g_i z^i$$

$$f(z) = \left[\sum_{i=0}^{\infty} g_i z^{-i}\right]\left[\sum_{i=0}^{\infty} g_i z^i\right] = \sum_{i=-\infty}^{\infty} f_i z^i$$

$$f_i = f_{-i} = \sum_{j=i}^{\infty} g_j g_{j-i}. \tag{28}$$

[1] Note that if $n_d > n_n$, then $\Gamma_d^{(1)} = \emptyset$, and $\Gamma_d^{(2)}$ can simply be augmented on top by $n_d - n_n - 1$ rows of zeros.



(a) (b)

Fig. 1. (a) Output responses. (b) Input responses.

The proof is completed by noting that the ith row of $\Gamma^T_{\frac{1}{d}}$ is given by $[0,\cdots,0,g_0,g_1,\cdots]$, and the jth column of $\Gamma_{\frac{1}{d}}$ is given by $[0,\cdots,0,g_0,g_1,\cdots]^T$ so that the i,j element of $\Gamma^T_{\frac{1}{d}}\Gamma_{\frac{1}{d}}$ is given by $[g_0g_{i-j}+g_1g_{i-j+1}+\cdots]$ which is identical to the definition of f_{i-j}. □

Remark 3.1: $f_i = f_{-i}$, so if $\Gamma_{\frac{1}{d}}$ has $m+1$ columns, then there are only $m+1$ distinct values of f_i in the matrix $\Gamma^T_{\frac{1}{d}}\Gamma_{\frac{1}{d}}$. Therefore, the computation of this matrix reduces to the computation of the first $m+1$ coefficients of the Laurent series expansion of $f(z)$.

We now propose a numerically efficient means of computing the first $m+1$ coefficients of the Laurent expansion of $f(z)$ defined in (27). Equating the coefficients in $z^i, (0 \le i \le m)$ of $1/d(z^{-1})$ to those of $f(z)d(z)$ gives the following independent set of $m+1$ linear equations:

$$\left[\begin{array}{ccccc|c|cc} d_0 & d_1 & \cdots & d_{n_d} & \cdots & 0 & 0 & \cdots \\ 0 & d_0 & \cdots & d_{n_d-1} & \cdots & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & d_0 & d_1 & \cdots \end{array}\right] \left[\begin{array}{c} f_m \\ f_{m-1} \\ \cdots \\ \hline f_0 \\ \hline f_1 \\ \vdots \\ f_m \end{array}\right]$$

$$= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{d_0} \end{bmatrix} = [A_1 \; |A_2| \; A_3] \begin{bmatrix} \mathbf{f} \\ f_0 \\ \hat{\mathbf{f}} \end{bmatrix} \qquad (29)$$

where $\mathbf{f} = [f_m,\cdots,f_1]^T$ and $\hat{\mathbf{f}} = [f_1,\cdots,f_m]^T$. The coefficients of $f(z)$ can now be computed as

$$[A_1 + \hat{A}_3 \quad A_2] \begin{bmatrix} \mathbf{f} \\ f_0 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{d_0} \end{bmatrix}. \qquad (30)$$

Equation (30) is clearly trivial to solve. It is noted that the minimum number of coefficients that can be computed is $n_d + 1$.

Theorem 3.1: The elements of the matrices B_1, B_2 of (25) can be computed by substituting $a^-(z^{-1})$ and $b^-(z^{-1})$, respectively, in place of $d(z^{-1})$ into (30).

Proof: This is an obvious consequence of Lemma (3.1) and the development earlier in this section and by noting from (18) and (25) that $B_1 = \Gamma^T_{\frac{1}{a^-}}\Gamma_{\frac{1}{a^-}}$ and $B_2 = \Gamma^T_{\frac{1}{b^-}}\Gamma_{\frac{1}{b^-}}$. □

Corollary 3.1: The optimizing $\mathbf{c}$ can be defined in the same form used in (26) as

$$\begin{aligned} \mathbf{c} &= -P^{-1}R\mathbf{q} \\ P &= \left[\Gamma^T_{b+}B_1\Gamma_{b+} + \lambda\Gamma^T_{A+}B_2\Gamma_{A+}\right] \\ R &= -\Gamma^T_{b+}B_1P_1 + \lambda\Gamma^T_{A+}B_2P_2. \end{aligned} \qquad (31)$$

Remark 3.2: The only difference between the approach of Sections III-A and III-B is in the computation of the matrices B_1, B_2, hence a comparison of their relative efficiencies is of interest.

The Lyapunov approach defines B_1, B_2 from (25) and requires: i) computation of S_a, S_b involving the solution of $(n_{a^-}^2 + n_{a^-})/2, (n_{b^-}^2 + n_{b^-})/2$ linear equations, respectively; ii) computation of M_a, M_b (approximately $n_{a^-}^3 + n_{b^-}^3$ multiplications); iii) computation of $\Gamma^{(1)}_{\frac{1}{a^-}}, \Gamma^{(1)}_{\frac{1}{b^-}}, \Gamma^{(2)}_{\frac{1}{a^-}}, \Gamma^{(2)}_{\frac{1}{b^-}}$ (approximately $n_{a^-}(n_{b+} + n_c) + n_{b^-}(n_{A+} + n_c)$ multiplications; and iv) computation of $B_1 = (\Gamma^{(1)T}_{\frac{1}{a^-}}\Gamma^{(1)}_{\frac{1}{a^-}} + \Gamma^{(2)T}_{\frac{1}{a^-}}S_a\Gamma^{(2)}_{\frac{1}{a^-}})$ and $B_2 = (\Gamma^{(1)T}_{\frac{1}{b^-}}\Gamma^{(1)}_{\frac{1}{b^-}} + \Gamma^{(2)T}_{\frac{1}{b^-}}S_b\Gamma^{(2)}_{\frac{1}{b^-}})$ (approximately $2n_{a^-}^3 + 2n_{b^-}^3$ multiplications).

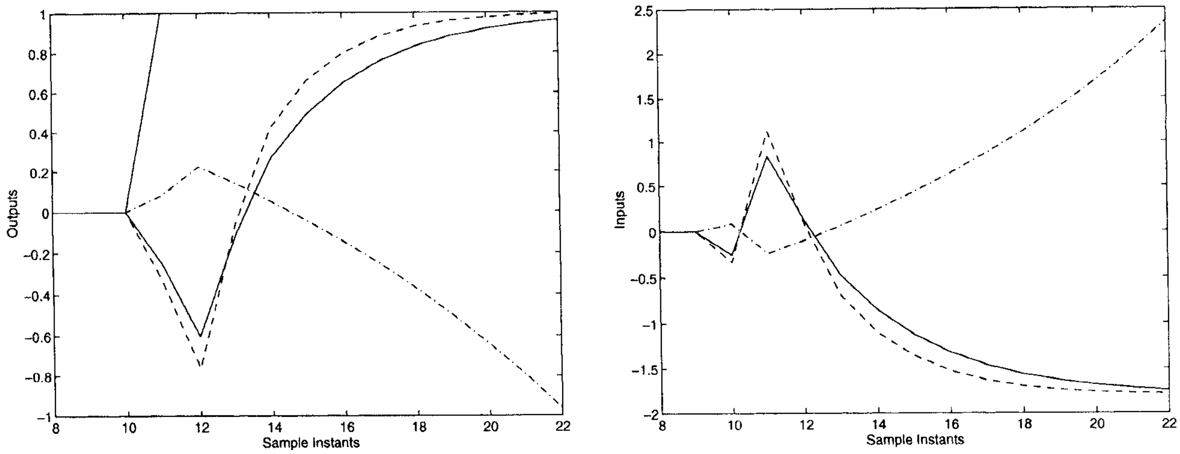
The approach of Section III-C requires the solution of m_b, m_A linear equations [of the form (30)], respectively, where $m_b = \max[n_{b+} + n_c + 1, n_{a^-} + 1]$ and $d(z^{-1}) = a^-(z^{-1})$ and $m_A = \max[n_{A+} + n_c + 1, n_{\psi_p}, n_{b^-} + 1]$ and $d(z^{-1}) = b^-(z^{-1})$.

Depending on the values of $n_{a^-}, n_{b^-}, n_{A+}, n_{b+}, n_c$, either of the above two algorithms may be better. However, if n_c is small but n_{a^-}, n_{b^-} are large, then less linear equations are implied in the solution of (30) than in the computation of S_a, S_b, favoring the algorithm of Section III-C.

IV. Simulation Example

In this section we demonstrate the efficacy of the IHSPC algorithm by comparing it with: i) the original GPC algorithm as presented in [7] with the same number of degrees of freedom and ii) the algorithm of [12] with the same number of degrees of freedom. The algorithms will be compared by way of simulation plots (the plots for IHSPC in solid line, for [12] in dashed line, and for GPC in dash-dot lines) and the measure of performance J_{run} defined as

$$J_{\text{run}} = \sum_{i=0}^{\text{runtime}} \left[e_i^2 + \lambda\Delta u_i^2\right]. \qquad (32)$$



(a) (b)

Fig. 2. (a) Output responses. (b) Input responses.

TABLE I
RUNTIME COSTS FOR EXAMPLE 1

GPC	[12]	IHSPC
2.7716	2.5975	2.2156

TABLE II
RUNTIME COSTS FOR EXAMPLE 2

GPC	[12]	IHSPC
∞	9.5993	7.8363

A. Example 1

The model was taken to be

$$a(z) = 1 - 2.5z^{-1} + z^{-2} \qquad b(z) = 1 - 0.7z^{-1} \tag{33}$$

which has an unstable pole at $z = 2$. The control parameters were $n_c = 1, \lambda = 0.1$. The output plots and input plots are given in Figs. 1(a) and (b), respectively, and the measure of runtime performances given in Tables I and II illustrate the improvements gained by the new algorithm.

B. Example 2

This model has unstable poles at $z = 2, 3$, an unstable zero at $z = -2$, and

$$\begin{aligned} a(z^{-1}) &= 1 + -5.5z^{-1} + 8.54z^{-2} - 3.2z^{-3} + 0.24z^{-4} \\ b(z^{-1}) &= 1 + 0.1z^{-1} - 3.1z^{-2} + 1.4z^{-3}. \end{aligned} \tag{34}$$

We chose $n_c = 1, \lambda = 1$. The output/input plots are given in Figs. 2(a) and (b), respectively, and J_{run} is presented in Table II. Again the improvement is clear and in fact GPC is unstable.

It is noted that although better results for GPC can be obtained, with this example (for $\lambda = 0.1$): i) for a control horizon n_u of 1, 2 GPC cannot stabilize the model and ii) for $n_u = 3$, too small an output horizon n_y gave instability and too large and n_y gave numerical problems (the model is open-loop unstable); reasonable results could only be obtained for $n_y \approx 15$. In contrast, both the algorithm of [12] and the algorithm of this paper have guaranteed stability and moreover gave good performance for all three control horizons of 1, 2, and 3.

V. CONCLUSION

This paper has two main contributions to make. First, it presents an efficient means of classifying pairs of stable input/output predictions in a manner which makes the resulting degrees of freedom transparent and second proposes a framework for the use of these predictions in an IHSPC law. This is unlike earlier work, where the stability was enforced by an equality terminal constraint which is numerically more difficult to handle. Moreover, the conditions used to classify the predictions are both necessary and sufficient for stability, whereas earlier work used sufficient conditions only. Here we allow the input trajectories to be infinite sequences, whereas other work in this area forces the input trajectory to have a finite number of changes. The use of necessary conditions will have two obvious benefits. It releases as many degrees of freedom as possible for meeting performance criteria, and it is more likely to be short-term feasible in the presence of system constraints. This latter point can be used to significant computational advantage and makes feasibility assumptions easier to meet.

Finally, it was noted that most authors use a Lyapunov equation to calculate infinite horizon performance indexes. Here we propose an alternative which is particularly suitable for infinite horizon GPC and in many cases will be computationally more efficient.

APPENDIX A

Equation (4) does not allow for disturbances and can be replaced by

$$\begin{aligned} y(z^{-1}) &= z^{-1}\frac{b(z^{-1})}{a(z^{-1})}u(z^{-1}) + \frac{T(z^{-1})}{\Delta(z^{-1})}\xi(z^{-1}) \\ T(z^{-1}) &= 1 + T_1 z^{-1} + \cdots + T_{n_T} z^{-n_T} \end{aligned} \tag{35}$$

where ξ is assumed to be a zero mean uncorrelated random variable. This implies that

$$\Delta(z^{-1})a(z^{-1})\tilde{y}(z^{-1}) = z^{-1}b(z^{-1})\Delta\tilde{u}(z^{-1}) + \xi(z^{-1}) \tag{36}$$

where $T(z^{-1})\tilde{y}(z^{-1}) = y(z^{-1})$, $T(z^{-1})\Delta\tilde{u}(z^{-1}) = \Delta u(z^{-1})$. The prediction equations for $\tilde{y}, \Delta\tilde{u}$ are

$$\begin{aligned} C_A \underset{\rightarrow}{\tilde{\mathbf{y}}} &= C_b \Delta \underset{\rightarrow}{\tilde{\mathbf{u}}} + \tilde{\mathbf{p}}; &\quad \tilde{\mathbf{p}} &= H_b \Delta \underset{\leftarrow}{\tilde{\mathbf{u}}} - H_A \underset{\leftarrow}{\tilde{\mathbf{y}}} \\ C_T \underset{\rightarrow}{\tilde{\mathbf{y}}} &= \underset{\rightarrow}{\mathbf{y}} - H_T \underset{\leftarrow}{\tilde{\mathbf{y}}}; &\quad C_T \Delta \underset{\rightarrow}{\tilde{\mathbf{u}}} &= \Delta \underset{\rightarrow}{\mathbf{u}} - H_T \Delta \underset{\leftarrow}{\tilde{\mathbf{u}}} \end{aligned} \tag{37}$$

where all future values of ξ are taken to be zero. C_T, H_T have dimensions $n_y \times n_y$, $n_y \times n_T$, respectively. Multiplying (37a) by

C_T and substituting for $\underset{\rightarrow}{\tilde{\mathbf{y}}}, \Delta \underset{\rightarrow}{\tilde{\mathbf{u}}}$ from (37b) and (37c) gives

$$\begin{aligned} C_A \underset{\rightarrow}{\mathbf{y}} - C_A H_T \underset{\leftarrow}{\tilde{\mathbf{y}}} &= C_b \Delta \underset{\rightarrow}{\mathbf{u}} - C_b H_T \Delta \underset{\leftarrow}{\tilde{\mathbf{u}}} + C_T \check{\mathbf{p}} \\ \Rightarrow \quad \underset{\rightarrow}{\mathbf{y}} &= C_A^{-1}[C_b \Delta \underset{\rightarrow}{\mathbf{u}} + \mathbf{p}'] \\ \mathbf{p}' &= -C_b H_T \Delta \underset{\leftarrow}{\tilde{\mathbf{u}}} + C_T \check{\mathbf{p}} + C_A H_T \underset{\leftarrow}{\tilde{\mathbf{y}}}. \end{aligned} \tag{38}$$

This is exactly the same as (5) except that $\mathbf{p}$ has been replaced by $\mathbf{p}'$. From this point on the treatment is identical to that presented in the main text of the paper.

Appendix B

Here we discuss the general set-point trajectory $r_{t+1}, \cdots, r_{t+n_y-1}, r, r, \cdots$ whose z-transform is

$$r(z^{-1}) = r_{t+1} + r_{t+2}z^{-1} + \cdots + r_{t+n_y-1}z^{-n_y+1} + \frac{rz^{-n_y}}{1-z^{-1}}. \tag{39}$$

In this case (7) must be replaced by

$$\begin{aligned} e(z^{-1}) &= \frac{r}{1-z^{-1}} - y(z^{-1}) + \rho(z^{-1}) \\ &= \frac{q(z^{-1}) - b(z^{-1})\Delta u(z^{-1})}{A(z^{-1})} + \rho(z^{-1}) \\ q(z^{-1}) &= a(z^{-1})r - p(z^{-1}) \end{aligned} \tag{40}$$

where $\rho(z^{-1}) = (r_{t+1} - r) + (r_{t+2} - r)z^{-1} + \cdots + (r_{t+n_y-1} - r)z^{-n_y+1}$. The definitions of $\phi(z^{-1})$ and $\psi(z^{-1})$ are exactly the same, but now (16) becomes

$$\begin{aligned} \underset{\rightarrow}{\mathbf{e}} &= \Gamma_{\frac{1}{a^-}}[-\Gamma_{b^+}\mathbf{c} + P_1\mathbf{q}] + \rho \\ \Delta \underset{\rightarrow}{\mathbf{u}} &= \Gamma_{\frac{1}{b^-}}[\Gamma_{A^+}\mathbf{c} + P_2\mathbf{q}] \\ \rho &= [(r_{t+1} - r), \cdots, (r_{t+n_y-1} - r), 0, \cdots]^T. \end{aligned} \tag{41}$$

The optimal solution for the future values of c given in (26) must be replaced by

$$\mathbf{c} = -P^{-1}[R\mathbf{q} - 2\Gamma_{b^+}^T \Gamma_{\frac{1}{a_-}}^T \rho]. \tag{42}$$

References

[1] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice, A survey," *Automatica*, vol. 25, pp. 335–348, 1989.

[2] M. Morari, "Model predictive control: Mulitvariable control technique choice of the 1990's," in *Advances in Model Based Predictive Control*, D. W. Clarke, Ed. Oxford: Oxford Sci., 1994.

[3] C. R. Cutler and B. L. Ramaker, "Dynamic matrix control—A computer control algorithm," in *Proc. JACC*, San Francisco, 1980.

[4] D. Q. Mayne and E. Polak, "Optimization based design and control," *IFAC World Congr.*, vol. 3, pp. 129–138, 1993.

[5] J. A. Richalet, A. Rault, J. L. Testud, and J. Papon, "Model predictive heuristic control: Applications to a industrial process," *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.

[6] D. W. Clarke and P. Gawthrop, "Self-tuning control," *Proc. IEE*, Pt. D, vol. 126, pp. 633–640, 1979.

[7] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control, Parts 1 and 2," *Automatica*, vol. 23, pp. 137–160, 1987.

[8] S. S. Keerthi and E. G. Gilbert, "Optimal Infinite horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving horizon approximations," *J. Optimization Theory Appl.*, vol. 57, no. 2, pp. 265–293, 1988.

[9] E. Mosca and J. Zhang, "Stable redesign of predictive control," *Automatica*, vol. 28, no. 6, pp. 1229–1233, 1992.

[10] D. W. Clarke and R. Scattolini, "Constrained receding horizon predictive control," *Proc. IEE*, Pt. D, vol. 138, no. 4, pp. 347–354, 1991.

[11] B. Kouvaritakis, J. A. Rossiter, and A. O. T. Chang, "Stable generalized predictive control: An algorithm with guaranteed stability," *Proc. IEE*, Pt. D, vol. 139, no. 4, pp. 349–362, 1992.

[12] J. B. Rawlings and K. R. Muske, "The stability of constrained receding horizon control," *IEEE Trans. Automat. Contr.*, vol. 38, no. 10, pp. 1512–1516, 1993.

[13] J. A. Rossiter, "GPC controllers with guaranteed stability and mean-level control of unstable plant," in *Proc. 33rd Conf. Decision Contr.*, Orlando, pp. 3579–3580, 1994.

[14] J. R. Gossner, B. Kouvaritakis, and J. A. Rossiter, "Cautious stable predictive control: A guaranteed stable predictive control algorithm with low input activity and good robustness," *IJC*, submitted.

[15] E. G. Gilbert, I. Kolmanovsky, and K. T. Tan, "Non-linear control of discrete-time linear systems with state and control constraints: A reference governor with global convergence properties" in *Proc. 33rd Conf. Decision Contr.*, Orlando, FL, pp. 144–149, 1994.

[16] J. R. Gossner, B. Kouvaritakis, and J. A. Rossiter, "Constrained cautious stable predictive control," *Proc. IEE*, Pt. D, submitted.

A Revisit to the Gain and Phase Margins of Linear Quadratic Regulators

Cishen Zhang and Minyue Fu

***Abstract*—In this paper, we revisit the well-known robustness properties of the linear quadratic regulator (LQR), namely, the guaranteed gain margin of -6 to $+\infty$ dB and phase margin of $-60°$ to $+60°$ for single-input systems. We caution that these guaranteed margins need to be carefully interpreted. More specifically, we show via examples that an LQR may have a very small margin with respect to the variations of the gain and/or phase of the open-loop plant. Such a situation occurs in most practical systems, where the set of measurable state variables cannot be arbitrarily selected. Therefore the lack of robustness of the LQR can be very popular and deserves attention.**

I. Introduction

The robustness properties of the linear quadratic regulators (LQR) have been known for many years. That is, an LQR for a single-input plant possesses a guaranteed gain margin of -6 to $+\infty$ dB and phase margin of $-60°$ to $60°$; see [6], [1], and [2]. This result is extended in [9] and [7] to the multivariable case, where the weighting matrix for the control is diagonal.

In this paper, we point out that the aforementioned robustness properties of LQR's should be carefully interpreted.

Consider the following single-input/single-output (SISO) plant

$$G(s) = KG_0(s) \tag{1}$$

where $G_0(s)$ is a fixed transfer function, and K, having a nominal value of one, is a complex parameter representing gain and phase variations of the plant. Suppose a set of state variables is measurable and an LQR is designed.

The basic robustness question is: do the guaranteed gain and phase margins apply to the gain and phase variations of the open-loop plant? We show via examples that the answer is negative, in general. It turns

Manuscript received November 9, 1994. This work was supported by the Australian Research Council.

C. Zhang is with the Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, VIC 3052 Australia.

M. Fu is with the Department of Electrical and Computer Engineering, University of Newcastle, Newcastle, N.S.W. 2308, Australia (e-mail: eemf@ee.newcastle.edu.au).

Publisher Item Identifier S 0018-9286(96)06770-0.