Constrained cautious stable predictive control

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

Indexing terms: Predictive control , Terminal constraints, Stable predicted trajectories

Abstract: Terminal constraints provide for stable predicted trajectories and thus form the basis of predictive control algorithms with guaranteed stability. In the presence of system input constraints, however, stability is dependent on a feasibility assumption, namely that it is possible to meet the terminal constraints with the allowable inputs. The terminal constraints used to date are only sufficient and lead to highly tuned controllers with high input activity which may cause infeasibility with a consequent loss of the guarantee of stability. Here we overcome this problem by deriving conditions which are both necessary and sufficient. Use of these leads to an increase in the degrees of freedom which can be used to either improve performance or reduce the relevant control horizon thereby affording significant computational advantages. The efficacy of the proposed algorithms are illustrated by means of numerical examples.

1 Introduction

Model-based predictive control [1, 2] owes its popularity to its simple strategy (the minimisation of the predicted tracking errors and control activity) and the fact that it can handle system constraints (e.g. [3, 4]). Early work [5–9] lacked a general stability theory, but more recent algorithms [10–13] provide the missing guarantees of stability and can be extended to handle system constraints [14–16]. The key development here is the introduction of terminal constraints, namely, that the predicted tracking errors should all be zero beyond a given output horizon and that the control moves themselves should become zero beyond a given input horizon.

These terminal constraints may come into conflict with limits on system inputs (dictated by physical constraints); such a conflict, termed infeasibility, may lead to instability. One possible remedy is to use longer horizons, but this results in a significant increase in the computation of constrained optima. An alternative is offered by setpoint conditioning [17–19], but inherent in this approach is a sacrifice of optimality in tracking in favour of retaining feasibility and hence guaranteeing stability. In this paper we use a different approach to infeasibility: earlier terminal constraints guarantee stability but are unnecessarily stringent, so here we employ constraints which are both necessary and sufficient [20] and thus, for given input horizons, maximise the control authority which is available for improving performance while respecting input constraints.

A convenient way to guarantee the stability of predictive control strategies is to force the predicted trajectories of both output error and control increments to be finite length sequences (FLS). This can be achieved by a process of cancelling the effect of all the poles appearing in the z-transforms of the input/output predictions. It has been recognised [17, 21] that for the purposes of stability one actually needs only cancel the effects of the unstable poles in the output predictions thereby turning the predicted output error trajectories into infinite length sequences (ILS). It is also the case [20] that one needs only cancel the effect of the unstable poles in the input prediction with the effect of getting predicted control increment trajectories which are ILS. However, in the presence of physical constraints ILS trajectories lead to a practical difficulty: the physical constraints have to be invoked over an infinite horizon. This problem can be overcome through the use of suitable input/output bounds. A set of such bounds with respect to output constraints have been proposed elsewhere [21, 22]. Here we are concerned with input constraints only, but we explore the use of ILS predictions for both inputs and outputs; therefore we require bounding results on inputs rather than outputs. In this paper we develop (Section 3) simple input bounding techniques which provide an efficient means of invoking the constraints over a infinite horizon by enforcing them over a finite horizon. Thus we are able to use our necessary and sufficient terminal constraints to advantage (Section 4) by allowing for: (i) the use of short command horizons; and/or (ii) the release of control authority for better transient performance. In Section 4 we also consider the use of terminal inequality constraints and thus remove restrictions on the size of setpoint changes. The efficacy/superiority of the resulting algorithms is illustrated in Section 5 by means of two numerical examples.

IEE Proceedings online no. 19971172

Paper first received 19th June and in revised form 11th December 1996

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

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

2 Review of cautious unconstrained predictive control

2.1 System model and prediction equation

Let the system model be given in terms of the transfer function equation:

$$y_t = \frac{z^{-1}b(z)}{a(z)}u_t = \frac{z^{-1}b(z)}{\alpha(z)}\Delta u_t \tag{1a}$$

$$\alpha(z) = \Delta(z)a(z) \tag{1b}$$

$$\Delta u_t = u_t - u_{t-1} \tag{1c}$$

$$\Delta(z) = 1 - z^{-1} \tag{1d}$$

where z^{-1} denotes the backward shift operator, $b(z)$, $a(z)$ are coprime polynomials, and the coefficients (in ascending powers of z^{-1}) of $b(z)$ and $\alpha(z)$ are b_i, $i = 0, \ldots, n_b$, and α_i, $i = 0, \ldots, n_\alpha$. Premultiplying eqn. 1*a* by $\alpha(z)$ and simulating forward we obtain:

$$C_\alpha \vec{y} + H_\alpha \overleftarrow{y} = C_b \Delta \vec{u} + H_b \Delta \overleftarrow{u} \tag{2a}$$

or

$$C_\alpha \vec{y} = C_b \Delta \vec{u} + p \tag{2b}$$

$$p = H_b \Delta \overleftarrow{u} - H_\alpha \overleftarrow{y} \tag{2c}$$

where

$$\Delta \vec{u} = \begin{bmatrix} \Delta u_t \\ \Delta u_{t-1} \\ \vdots \\ \Delta u_{t+n-1} \end{bmatrix} \tag{3a}$$

$$\Delta \overleftarrow{u} = \begin{bmatrix} \Delta u_{t-1} \\ \Delta u_{t-2} \\ \vdots \\ \Delta u_{t-n_b} \end{bmatrix} \tag{3b}$$

$$\vec{y} = \begin{bmatrix} y_{t+1} \\ y_{t+2} \\ \vdots \\ y_{t+n} \end{bmatrix} \tag{3c}$$

$$\overleftarrow{y} = \begin{bmatrix} y_t \\ y_{t-1} \\ \vdots \\ y_{t-n_\alpha+1} \end{bmatrix} \tag{3d}$$

$$C_\alpha = \begin{bmatrix} \alpha_0 & 0 & \cdots & \cdots & \cdots & 0 \\ \alpha_1 & \alpha_0 & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \alpha_{n_\alpha} & \alpha_{n_\alpha-1} & \cdots & \alpha_0 & 0 & \cdots \\ 0 & \alpha_{n_\alpha} & \alpha_{n_\alpha-1} & \cdots & \alpha_0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & 0 & \alpha_{n_\alpha} & \cdots & \alpha_1 & \alpha_0 \end{bmatrix} \in R^{n\times n} \tag{3e}$$

$$H_\alpha = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \cdots & \alpha_{n_\alpha} \\ \alpha_2 & \alpha_3 & \cdots & \alpha_{n_\alpha} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_{n_\alpha} & 0 & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \in R^{n\times n_\alpha} \tag{3f}$$

where n is the simulation horizon; $C_b \in R^{n\times n}$ and $H_b \in R^{n\times n_b}$ are defined in an analogous manner.

2.2 Conditions for the stability of predictions

It is normal practice in MBPC (model based predictive control) to use a finite number of nonzero future increments in the plant input, Δu, as the degrees of freedom for the purpose of optimising a cost that penalises future (predicted) errors and control activity. Our approach is somewhat different in that we shall first be concerned with the characterisation of the class of input/output predictions which are stable; this involves a set of new degrees of freedom which appear as the coefficients of a polynomial, $c(z)$, or the element of a corresponding vector $\vec{c}$. The details of this development are given in [20] and are summarised here and in Section 2.3.

Consider the z-transform of eqn. 1 which, for nonzero initial conditions, can be written as

$$\alpha(z)y(z) + y_p(z) = b(z)\Delta u(z) + \Delta u_p(z) \tag{4a}$$

$$y(z) = y_{t+1} + y_{t+2}z^{-1} + y_{t+3}z^{-2} + \cdots$$

$$y_p(z) = [1 \quad z^{-1} \quad \cdots \quad z^{-n_\alpha+1} \quad 0\cdots 0]\, H_\alpha \overleftarrow{y} \tag{4b}$$

$$\Delta u(z) = \Delta u_t + \Delta u_{t+1}z^{-1} + \Delta u_{t+2}z^{-2} + \cdots$$

$$\Delta u_p(z) = [1 \quad z^{-1} \quad \cdots \quad z^{-n_b+1} \quad 0\cdots 0]\, H_b \Delta \overleftarrow{u} \tag{4c}$$

where it is noted that $y(z)$ and $\Delta u(z)$ depend on future values of y and Δu only, whereas $y_p(z)$ and $\Delta u_p(z)$ depend on past (known) values. Eqn. 4*a* can be rewritten as

$$\alpha(z)y(z) = b(z)\Delta u(z) + p(z) \tag{5a}$$

or

$$y(z) = \frac{b(z)\Delta u(z) + p(z)}{\alpha^-(z)\alpha^+(z)} \tag{5b}$$

$$p(z) = \Delta u_p(z) - y_p(z) \tag{5c}$$

where $\alpha^+(z)$, $\alpha^-(z)$ are the factors of $\alpha(z)$ whose roots lie outside (or on), inside the unit circle and are of degree n_{α^+}, n_{α^-}, respectively. From eqn. 5*b* it is apparent that the necessary and sufficient condition for the stability of the predicted output (i.e. $y(z)$) is that

$$b(z)\Delta u(z) + p(z) = \alpha^+(z)\phi(z) \tag{6a}$$

or

$$\Delta u(z) = \frac{\alpha^+(z)\phi(z) - p(z)}{b^-(z)b^+(z)} \tag{6b}$$

where $\phi(z)$ is the z-transform of a convergent sequence $\{\phi_0, \phi_1, \phi_2, \ldots\}$ and $b^-(z)$, $b^+(z)$ are defined in a manner analogous to $\alpha^-(z)$, $\alpha^+(z)$. From eqn. 6*b* it follows that the predicted trajectories of control increments Δu will be stable if, and only if,

$$\alpha^+(z)\phi(z) - p(z) = b^+(z)\psi(z) \tag{7a}$$

or

$$\alpha^+(z)\phi(z) - b^+(z)\psi(z) = p(z) \tag{7b}$$

where $\psi(z)$ denotes the z-transform of a convergent sequence $\{\psi_0, \psi_1, \ldots\}$.

A particular minimal order solution to the Diophantine eqn. 7*b* is defined by the vector of coefficients, $[\phi_p^T, \psi_p^T]^T = [\phi_0, \phi_1, \ldots, \phi_{n_\phi}, \psi_0, \psi_1, \ldots, \psi_{n_\psi}]^T$ of the z-transforms, $\phi_p(z)$, $\psi_p(z)$:

$$[\Gamma_{\alpha^+} \quad -\Gamma_{b^+}] \begin{bmatrix} \phi_p \\ \psi_p \end{bmatrix} = \begin{bmatrix} p \\ p_a \end{bmatrix}$$

$$
\text{with}\begin{cases} n_\phi = n_{b^+} - 1, \\ n_\psi = n_{\alpha^+} - 1, \\ p_a = [.] = empty & \text{for } n_{\alpha^+} + n_{b^+} = n_\alpha \\ n_\phi = n_{b^+} - 1, \\ n_\psi = n_{\alpha^+} - 1, \\ p_a = \mathbf{0}_{(1,v)} & \text{for } n_{\alpha^+} + n_{b^+} > n_\alpha \\ n_\phi = n_{\alpha^-} - 1, \\ n_\psi = n_{\alpha^+} - 1, \\ p_a = [.] & \text{for } n_{\alpha^+} + n_{b^+} < n_\alpha \end{cases} \tag{8}
$$

where Γ_{α^+}, Γ_{b^+} are the matrices defined by the first $n_\phi + 1$, $n_\psi + 1$ columns of the $\{n_\phi + n_\psi + 2\} \times \{n_\phi + n_\psi + 2\}$ convolution matrices C_{α^+}, C_{b^+}; use has been made of the fact that the degree of $p(z)$ is $n_\alpha - 1$; $v = n_{\alpha^+} + n_{b^+} - n_\alpha$. Since $\alpha^+(z)$ and $b^+(z)$ are coprime, it is easy to show that the matrix on the LHS of eqn. 8*a* is invertible and thus leads to the particular solution:

$$
\begin{aligned}
\begin{bmatrix} \phi_p \\ \psi_p \end{bmatrix} &= [\Gamma_{\alpha^+} \quad -\Gamma_{b^+}]^{-1} \begin{bmatrix} p \\ p_a \end{bmatrix} \\
&= \begin{bmatrix} P_1 & P_3 \\ P_2 & P_4 \end{bmatrix} \begin{bmatrix} p \\ p_a \end{bmatrix} \\
&= \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} p
\end{aligned} \tag{9}
$$

where P_{1-4} denote partitions of the inverse in eqn. 9 of dimensions conformal to ϕ_p, ψ_p, P, p_a. Noting that the pair $\phi(z) = b^+(z)d(z)$ and $\psi(z) = \alpha^+(z)d(z)$, where $d(z)$ denotes the z-transform of a convergent sequence $\{d_0, d_1, ...\}$, solves eqn. 7*b* for $p(z) = 0$, we may give the general solution pair as

$$
\phi(z) = b^+(z)d(z) + \phi_p(z) \tag{10a}
$$

$$
\psi(z) = \alpha^+(z)d(z) + \psi_p(z) \tag{10b}
$$

This, together with eqns. 5–7 defme the necessary and sufficient condition for the stability of the predicted y and Δu and form the basis of CaSC (cautious stable control). The implied input/output predictions are then

$$
y(z) = \frac{\phi(z)}{\alpha^-(z)} \tag{11a}
$$

$$
\Delta u(z) = \frac{\psi(z)}{b^-(z)} \tag{11b}
$$

Remark 2.1: Eqn. 11 shows that our development cancels the effects of unstable poles and nonminimum phase zeros on the predicted inputs/outputs. However, this cancellation: (i) concerns prediction equations only; (ii) involves past values of inputs/outputs through the presence of $p(z)$ in the numerator of eqns. 5*b* and 6*b*. As such, it is not to be confused with pole-zero cancellation between controller and plant within a feedback loop, which could lead to internal instability difficulties.

Remark 2.2: The characterisation of the class of stable predictions bears some superficial similarity to the Youla characterisation of internally stabilising controllers. However, here (as explained in Remark 2.1) we allow for unstable pole/zero cancellations. Furthermore, the relevant Diophantine equations involves the 'unstable' part of $\alpha(z)$ and $b(z)$ and finally the right-hand side of the Diophantine equation depends on initial conditions.

2.3 Cautious stable control (CaSC)

In this Section we show how the characterisation of stable input/output predictions derived above can be used in the design of a predictive control law with guaranteed stability. The design of predictive controllers normally involves the minimisation over the set of predicted input increments of a performance index J which penalises predicted tracking errors and control activity. Here, instead of minimising over the future controls increments Δu, we minimise over the degrees of freedom in predictions eqns. 10 and 11.

A convenient way to guarantee stability is to define the cost J in such a way that: (i) its value can be shown to be monotonically decreasing (a Lyapunov function) and (ii) convergence of J implies convergence of y and Δu. One means of achieving this with GPC is to use a suitable terminal constraint to force the input/output predictions, which are used in J to be FLS [11–13]. The alternative approach adopted here is to use the conditions of Section 2.2 that yield predictions which are ILSs rather than FLSs. Then, to get FLSs for use in J, it remains to: (i) weight $y(z)$ of eqn. 11*a* by $\alpha^-(z)$ and $\Delta u(z)$ of eqn. 11*b* by $b^-(z)$; (ii) invoke an appropriate terminal constraint. Now, instead of using weighted variables in (i) it is possible to use new variables:

$$
\tilde{y}_t = \alpha^-(z)y_t \tag{12a}
$$

$$
\Delta\hat{u}_t = b^-(z)\Delta u_t \tag{12b}
$$

$$
\tilde{r}_t = \alpha^-(z)r_t \tag{12c}
$$

in J; and for (ii), we require that the future values of $\tilde{y}$ be equal to the future values of $\tilde{r}$ after some output horizon n_y, and that the future values of $\Delta\hat{u}$ be zero after some input horizon n_u. Collecting these observation together, we therefore write:

$$
\tilde{y}(z) = \sum_{i=1}^{n_y} \tilde{y}_{t+i} z^{-i+1} + \frac{z^{-n_y}}{1 - z^{-1}} \tilde{r}_\infty \tag{13a}
$$

$$
\Delta\hat{u}(z) = \sum_{i=0}^{n_u - 1} \Delta\hat{u}_{t+i} z^{-i} \tag{13b}
$$

$$
\tilde{r}(z) = \sum_{i=1}^{n_y} \tilde{r}_{t+i} z^{-i+1} + \frac{z^{-n_y}}{1 - z^{-1}} \tilde{r}_\infty \tag{13c}
$$

$$
\tilde{r}_\infty = \alpha^-(1)r \tag{13d}
$$

$$
\begin{aligned}
J &= \sum_{i=1}^{n_y} (\tilde{r}_{t+i} - \tilde{y}_{t+i})^2 + \lambda \sum_{i=0}^{n_u - 1} \Delta\hat{u}_{t+i}^2 \\
&= \| \vec{\tilde{r}} - \vec{\tilde{y}} \|_2^2 + \lambda \| \Delta \vec{\hat{u}} \|_2^2
\end{aligned} \tag{13e}
$$

where $r(z)$ is the z-transform of the sequence $\{r_{t+1}, r_{t+2}, ..., r_{t+n_r}, r_{t+n_r}, ...\}$, n_r is the reference horizon and r, by itself, represents the value of setpoint at the reference horizon, r_{t+n_r}; $\vec{\tilde{r}}$, $\vec{\tilde{y}}$, and $\Delta\vec{\hat{u}}$ are vectors of the coefficients of $\tilde{r}(z)$, $\tilde{y}(z)$, $\Delta\hat{u}(z)$. The definitions of eqns. 13*a* and *b* implicitly incorporate the terminal constraints:

$$
\tilde{y}_{t+i} = \tilde{r}_{t+i} \qquad i > n_y \tag{14a}
$$

$$
\Delta\hat{u}_{t+i} = 0 \qquad i \geq n_u \tag{14b}
$$

These terminal constraints together with eqns. 10 and 11 imply that $d(z)$ must be of the form:

$$
d(z) = c(z) + \frac{z^{-n_c}}{\Delta(z)} c_\infty \tag{15a}
$$

$$
c(z) = c_0 + c_1 z^{-1} + \cdots + c_{n_c - 1} z^{-n_c + 1} \tag{15b}
$$

$$
c_\infty = \frac{\alpha^-(1)}{b^+(1)} r \tag{15c}
$$

where the value of c_∞ is dictated by the final value theorem and where the output horizon, n_y, and the input horizon, n_u, are related to the command horizon, n_c, by $n_y = n_c + n_{b^+}$, and $n_u = n_c + n_{\alpha^+} + 1$. Combining eqn. 15a with eqns. 10 and 11, and defining $a^+(z)$ to be the polynomial $\alpha^+(z)/\Delta(z)$ we get

$$y(z) = \frac{b^+(z)}{\alpha^-(z)}c(z) + \frac{z^{-n_c}b^+(z)}{\Delta(z)\alpha^-(z)}c_\infty + \frac{1}{\alpha^-(z)}\phi_p(z) \quad (16a)$$

$$\Delta u(z) = \frac{\alpha^+(z)}{b^-(z)}c(z) + \frac{z^{-n_c}a^+(z)}{b^-(z)}c_\infty + \frac{1}{b^-(z)}\psi_p(z) \quad (16b)$$

$$u(z) = \frac{a^+(z)}{b^-(z)}c(z) + \frac{z^{-n_c}a^+(z)}{\Delta(z)b^-(z)}c_\infty + \frac{1}{\Delta(z)b^-(z)}\psi_p(z) + \frac{1}{\Delta(z)}u_{t-1} \quad (16c)$$

where use has been made of the fact that $u(z) = [\Delta u(z) + u_{t-1}]/\Delta(z)$. Premultiplying $y(z)$ by $\alpha^-(z)$ and $\Delta u(z)$ by $b^-(z)$, and accounting for the past, enables us to write vector forms of the above prediction equations as

$$\vec{y} = \Gamma_{b^+}\vec{c} + m_{b^+}c_\infty + P_1 p + H_{\alpha^-}\overleftarrow{y} \quad (17a)$$

$$\Delta\vec{u} = \Gamma_{\alpha^+/b^-}\vec{c} + m_{a^+/b^-}c_\infty + \Gamma_{1/b^-}P_2 p \quad (17b)$$

$$\Delta\vec{\hat{u}} = \Gamma_{\alpha^+}\vec{c} + m_{\alpha^+}c_\infty + P_2 p + H_{b^-}\Delta\overleftarrow{u} \quad (17c)$$

$$\vec{u} = \Gamma_{a^+/b^-}\vec{c} + m_{a^+/b^-}c_\infty + \Gamma_{1/\Delta b^-}P_2 p + u_{t-1}\mathbf{1} \quad (17d)$$

$$\vec{\tilde{r}} = c_{\alpha^-}\vec{r} + H_{\alpha^-}\overleftarrow{r} \quad (17e)$$

where the Γ matrices have n_c columns with the ith column containing $i - 1$ zero elements followed by the impulse response of the transfer function indicated by the corresponding subscript, the m vectors contain n_c zero elements, followed by the step response of the corresponding transfer function, and **1** denotes a vector of 1s of conformal dimension.

Prediction eqns. 17a and c imply that the cost J of eqn. 13e is quadratic in the vector of future cs and hence can be minimised explicitly and this forms the basis of the CaSC algorithm:

Algorithm 2.1: The predictive control law, CaSC, is defined through: (i) the minimisation over c of the cost J of eqn. 13e, with $\vec{y}^\sim$, $\Delta\vec{u}^\wedge$, $\vec{r}^\sim$ as defined in (eqns. 17a, c, e and 15c) and (ii) of the elements of the minimising $\vec{c}$, use c_0 to compute/implement Δu_0 and repeat the calculation at the next sampling instant with new plant data.

Theorem 2.1 [20]: CaSC has guaranteed stability and asymptotic tracking.

Proof: Let J_t be the optimum J at time t, and $J_{t+1|t}$ be the value of J at $t + 1$ for the optimal control increments computed at t. Then,

$$J_{t+1|t} = J_t - [(\tilde{r}_{t+1} - \tilde{y}_{t+1})^2 + \lambda\Delta\hat{u}_t^2] + [(\tilde{r}_{t+n_y+1} - \tilde{y}_{t+n_y+1})^2 + \lambda\Delta\hat{u}^2_{t+n_u}]$$

However, on account of the terminal constraint of eqns. 14a and b, we have that $\tilde{r}_{t+n_y+1} - \tilde{y}_{t+n_y+1} = 0$ and $\Delta\hat{u}_{t+n_u} = 0$, so that $J_{t+1|t} \leq J_t$. $J_{t+1|t}$ is based on the optimisation performed at t; further optimisation at $t + 1$ will give an optimal cost $J_{t+1} \leq J_{t+1|t} \leq J_t$; the equality $J_{t+1} = J_t$ can only persist if $\tilde{r}_{t+i} = \tilde{y}_{t+i}$, $\Delta\hat{u}_{t+i} = 0$ for all i in which case $J_{t+i} = 0$ for all i. Hence J_t is a monotonically decreasing function of t, thereby ensuring that $\tilde{e}_t = \tilde{r}_i - \tilde{y}_t$, $\Delta\hat{u}_t$ both converge to zero. But, by definition, $e = (1/a^-)\tilde{e}$, $\Delta u = (1/b^-)\Delta\hat{u}$ and so e and Δu also will converge to zero.

Remark 2.3: CaSC resembles the algorithm proposed in [21] in that they both deploy stable ILS output predictions, but differs in that in [21] the input predictions are forced to be FLSs (i.e. for n_u degrees of freedom it is assumed that $\Delta u_{t+n_u+i} = 0$ for $i \geq 0$). By contrast, in CaSC the input predictions are allowed to be ILS. In particular, CaSC uses condition (eqn. 7) which is both necessary and sufficient for the stability of input/output predictions, whereas [21] in essence uses condition (eqn. 7) with $b^+(z)$ replaced by $b(z)$. The use of ILSs in place of FLSs results in 'cautious' control: it avoids dead-beat predictions which lead to highly tuned controllers with the attendant feasibility difficulties. This observation, together with the fact that we preserve the guarantee of stability, explains the use of the terminology 'cautious stable control'.

3 Feasibility of cautious predictive control with input constraints

3.1 Definition of problem

Most practical control systems are subject to (hard) input constraints such as

$$\underline{u} \leq u_{t+i} < \overline{u} \quad (18a)$$

$$\underline{\Delta u} \leq \Delta u_{t+i} \leq \overline{\Delta u} \quad i \geq 0 \quad (18b)$$

where it is assumed (as is reasonable) that the steady-state input/control increment required to reach the set-point are at least a distance ϵ (for ϵ arbitrarily small) inside the above limits:

$$\underline{u} + \epsilon \leq u_\infty \leq \overline{u} - \epsilon \quad (19a)$$

$$\underline{\Delta u} + \epsilon \leq 0 \leq \overline{\Delta u} - \epsilon \quad (19b)$$

and for convenience (only) we have assumed that the input limits are time invariant. The input constraints above may come into conflict with the use of terminal constraints and this condition we term infeasibility. By contrast henceforth, the compatibility of terminal constraints with input constraints shall be referred to as feasibility.

CaSC does not take constraints (eqns. 18a and b) into consideration, and hence the optimal predicted values for u and/or Δu may lie outside the limits of eqns. 18a and b; this will lead to suboptimality and may even result in instability. Thus it is important to incorporate constraints (eqns. 18a and b) into the optimisation problem; earlier work (e.g. [2, 3, 14]) achieves this through the use of quadratic programming. Work, to date, has restricted attention to finite length sequences for the future Δus; this is a consequence of the fact that terminal constraint (eqn. 14b) has been invoked on Δu rather than on $\Delta\hat{u}$. Terminal constraints of this sort are convenient in establishing stability, which depends on a guarantee of feasibility (i.e. a guarantee that the terminal constraints can be met within the input constraints). However, such terminal constraints are only sufficient for the stability of predicted trajectories and thus may result in an unnecessary restriction of the degrees of design freedom. This could have a significant effect on performance; furthermore, on account of the feasibility requirement it may necessitate the use of longer control horizons with a concomitant significant increase in the quadratic programming computational load.

Here, our concern is to use the maximum degrees of design freedom possible and this implies the need for conditions such as those developed in the previous Section which are both necessary and sufficient. The difficulty with these, however, is that they result in future control trajectories which form infinite length sequences; at first sight, this requires that feasibility be tested over an infinite horizon. To overcome this problem here, we develop some efficient (albeit loose) bounding results which enable the definition of a finite horizon, n_{con}, referred to as the constraints horizon, which has the property that feasibility over n_{con} implies actual feasibility over an infinite horizon. It is noted that the reconciliation of finite with infinite constraints horizons has been addressed elsewhere [21, 22], but in a different context (related to output constraints) using different (state-space) bounding techniques. The techniques developed here address a new problem, namely, the maximisation of available control freedom with the view to satisfying the feasibility requirement as well as improving performance and/or reducing the computational burden.

3.2 Bounding conditions

We wish to invoke terminal constraints (eqns. 14*a* and *b*) without violating physical limits (eqns. 18*a* and *b*), and, as remarked above, the difficulty here is that future control trajectories are ILSs. So now we seek to determine a finite (and preferably small) value for n_{con} such that feasibility for all future times is guaranteed by feasibility up to n_{con}. An obvious way to achieve this is to: (i) consider a particular future time instant $t + i$; (ii) derive bounds on the maximum and minimum values of the predicted us and Δus beyond this time instant $t + i$, namely, bounds for future values at all times $t + j$, $j \geq i$; and (iii) increase i until the bounds of (ii) are within the physical limits of eqns. 18*a* and *b*.

Remark 3.1: The bounds used in the determination of n_{con} need not be 'tight', in that 'loose' bounds would merely result in a conservative choice for n_{con}. The result of this is that eqns. 18*a* and *b* will be checked at more time instants than necessary. This will not have a significant effect on computations because the extra checks will correspond to constraint inequalities which by definition will be inactive: if future us and Δus are within 'loose' bounds which themselves are within the limits of eqns. 18*a* and *b*, such us and Δus will satisfy eqns. 18*a* and *b a fortiori*. For this reason, the emphasis in what follows will be on ease of presentation/computation rather than obtaining the tightest bounds possible. Along the same lines we shall refrain from defining different values for n_{con} for each of conditions eqns. 18*a* and *b*.

From prediction eqns. 17*b* and *d* it is apparent that the future predicted values of u and Δu depend on future values of c which as yet are unknown. In determining bounds on u and Δu therefore, we must first stipulate bounds of c.

Lemma 3.1: Input constraints (eqn. 18*a* and *b*) will be violated if any of the future cs lie outside the respective interval defined below:

$$\underline{c}_i \leq c_i \leq \overline{c}_i \quad i = 0, \ldots, n_c - 1 \tag{20a}$$

$$\overline{c}_i = \min\left[\sum_{j=1}^{n_c}(\max[S_{ij}\overline{\Delta u}, S_{ij}\underline{\Delta u}] - S_{ij}s_j), \sum_{j=1}^{n_c}(\max[V_{ij}\overline{u}, V_{ij}\underline{u}] - V_{ij}v_j)\right] \tag{20b}$$

$$\underline{c}_i = \max\left[\sum_{j=1}^{n_c}(\min[S_{ij}\overline{\Delta u}, S_{ij}\underline{\Delta u}] - S_{ij}s_j), \sum_{j=1}^{n_c}(\min[V_{ij}\overline{u}, V_{ij}\underline{u}] - V_{ij}v_j)\right] \tag{20c}$$

where S_{ij} and V_{ij} denote the ijth elements of matrices S and V, whereas s_j, v_j denote the jth element of vectors s, v; S and V are defined as the inverses of block matrices formed out of the first n_c rows of Γ_{α^+/b^-}, and Γ_{a^+/b^-}, respectively, and s, v are the vectors formed out of the first n_c elements of $\Gamma_{1/b^-}P_2p$, and $\Gamma_{1/\Delta b^-}P_2p + u_{t-1}\mathbf{1}$.

Proof: With the definitions of S, V, s and v, the first n_c scalar equations implied by eqns. 17*b* and *d* can be rearranged to give

$$\vec{c} = V\left(\begin{bmatrix} u_t \\ \vdots \\ u_{t+n_c-1} \end{bmatrix} - v\right) \tag{21a}$$

$$\vec{c} = S\left(\begin{bmatrix} \Delta u_t \\ \vdots \\ \Delta u_{t+n_c-1} \end{bmatrix} - s\right) \tag{21b}$$

where use has been made of the fact that, due to the presence of z^{-n_c} in the second term of eqns. 16*b* and *c*, the first n_c elements of $m_{\alpha^+/b^-}c_\infty$ are zero. Thus c_i is given as (i) the sum (over j) of terms $S_{ij}\Delta u_{t+j} - S_{ij}s_j$ and (ii) the sum (over j) of terms $V_{ij}u_{t+j} - V_{ij}v_j$. The proof is completed by invoking the limits of eqns. 18*a* and *b* on Δu_{t+j} and u_{t+j}; (i) and (ii) imply different intervals for c_i, and since both must hold true, the intersection of the two is used to define the bounds of eqn. 20.

To derive bounds on the predicted Δu_{t+j} for $j \geq i$, consider the first term of the RHS of eqn. 16*b* which is of the form $h(z)f(z)$, with $h(z) = \alpha^+/b^-(z)$ and $f(z) = c(z)$. The other two terms have the same form and the overall result can be assembled by a process of linear superposition; the only difference is that $f(z)$ for the second and third term are known, whereas $f(z)$ for the first term is unknown, but bounded. The lemmata below deal with these two different cases separately.

Lemma 3.2: Let $g(z) = h(z)f(z)$, where $h(z)$ is an asymptotically stable transfer function with impulse response $\{h_0, h_1, \ldots\}$ and $f(z)$ is a polynomial in z^{-1} of degree n_f whose coefficients f_i are known. Then bounds on each (and all subsequent) element of the impulse response of $g(z)$ are given by

$$\underline{G}_i \leq g_j \leq \overline{G}_i \quad (j \geq i)$$

$$\begin{cases} \underline{G}_i = \sum_{k=0}^{n_f}\min[\underline{H}_{i-k}f_k, \overline{H}_{i-k}f_k] \\ \overline{G}_i = \sum_{k=0}^{n_f}\max[\underline{H}_{i-k}f_k, \overline{H}_{i-k}f_k] \end{cases}$$

$$\begin{cases} \underline{H}_i = \min_l h_l \quad (l \geq i) \\ \overline{H}_i = \max_l h_l \quad (l \geq i) \end{cases} \tag{22}$$

Proof: This follows from the definition of $g(z)$ according to which g_i is given as the appropriate sum of products $h_{i-k}f_k$.

Lemma 3.3: Let $g(z) = h(z)f(z)$ where $h(z)$ is an asymptotically stable transfer function with impulse response $\{h_0, h_1, ...\}$, and $f(z)$ is a polynomial in z^{-1} of degree n_f whose coefficients f_i are unknown, but bounded by $\underline{f}_i \le f_i \le \overline{f}_i$, and let the bounds, $\underline{H}_i$, and $\overline{H}_i$ be as defined in eqn. 22. Then bounds on each (and all subsequent) element of the impulse response of $g(z)$ are given by

$$\underline{G}_i \le g_j \le \overline{G}_i \quad (j \ge i)$$

$$\begin{cases} \underline{G}_i = \sum_{k=0}^{n_f} \min[\underline{H}_{i-k}\underline{f}_k, \overline{H}_{i-k}\underline{f}_k, \underline{H}_{i-k}\overline{f}_k, \overline{H}_{i-k}\overline{f}_k] \\ \overline{G}_i = \sum_{k=0}^{n_f} \max[\underline{H}_{i-k}\underline{f}_k, \overline{H}_{i-k}\underline{f}_k, \underline{H}_{i-k}\overline{f}_k, \overline{H}_{i-k}\overline{f}_k] \end{cases} \tag{23}$$

Proof: This is the same as for Lemma 3.2, except that, when dealing with the products $h_{i-k}f_k$, we now have to consider intervals for both the coefficients of $h(z)$ and $f(z)$.

Remark 3.2: The bounding results above afford a significant online computational advantage because bounds, $\underline{H}_i$, and $\overline{H}_i$ are time invariant and thus can be calculated off line and saved in a look-up table. Therefore the bounds which apply over an infinite horizon are calculated by summing a finite (and indeed small) number of maxima/minima over two or four scalars.

As mentioned previously, the predicted Δus and us of eqns. 16*b* and *c* can be viewed as the sum of products, $h(z)f(z)$, all of which can be bounded (as per Lemmata 3.2, 3.3) and then combined by linear superposition. To that end, we now defme bounds for each part of eqns. 16*b* and *c*, except for the last part of eqn. 16*c* whose upper and lower bounds are equal to u_{t-1} (Table 1): It is then a simple matter to give bounds on the future elements of Δu and u as

$$\underline{G}_i^1 + \underline{G}_i^2 + \underline{G}_i^3 \le \Delta u_{t+j} \le \overline{G}_i^1 + \overline{G}_i^2 + \overline{G}_i^3 \tag{24a}$$

$$\underline{G}_i^4 + \underline{G}_i^5 + \underline{G}_i^6 \le u_{t+j} \le \overline{G}_i^4 + \overline{G}_i^5 + \overline{G}_i^6 + u_{t-1} \quad j \ge i \tag{24b}$$

Theorem 3.1: Let n_{con} be the smallest value of i for which the derived bounds are 'inside' the limits of input constraints (eqns. 18*a* and *b*), i.e.

$$\underline{G}_{n_{con}}^1 + \underline{G}_{n_{con}}^2 + \underline{G}_{n_{con}}^3 \ge \underline{\Delta u} \tag{25a}$$

$$\underline{G}_{n_{con}}^4 + \underline{G}_{n_{con}}^5 + \underline{G}_{n_{con}}^6 + u_{t-1} \ge \underline{u} \tag{25b}$$

$$\overline{G}_{n_{con}}^1 + \overline{G}_{n_{con}}^2 + \overline{G}_{n_{con}}^3 \le \overline{\Delta u} \tag{25c}$$

$$\overline{G}_{n_{con}}^4 + \overline{G}_{n_{con}}^5 + \overline{G}_{n_{con}}^6 + u_{t-1} \le \overline{u} \tag{25d}$$

Then, if prediction eqns. 17*b* and *d* are made to satisfy constraints (eqns. 18*a* and *b*) at $t + i$, for $i \le n_{con}$, they will satisfy these constraints for all future times.

Proof: This follows from Lemmata 3.1–3.3 and the bound definitions of Table 1.

Corollary 3.1: A finite n_{con} can always be found which satisfies Theorem 3.1.

Proof: Consider the bounds $\underline{G}_i^k$, $\overline{G}_i^k$ for k = 1–3 associated with Δu_j, and apply the final value theorem to the corresponding $h(z)$ of Table 1 to deduce that, as $i \to \infty$, the elements h_i of the impulse response of $h(z)$ go to zero. As a consequence, the bounds $\underline{H}_i$, $\overline{H}_i$ (defined in eqn. 22) will also converge to zero, and hence so will the bounds on Δu_j of eqn. 24*a* because of their definition as given in eqns. 22 or 23. Now from eqn. 19*b* we know that $\underline{\Delta u}$, $\overline{\Delta u}$ are at least a distance ε away from 0; hence for any ε (no matter how small), n_{con} can be chosen large enough so that the moduli of the bounds on Δu_j of conditions (eqn. 24*a*) are less than ε: therefore conditions eqns. 25*a* and *c* will hold true.

To complete the proof, we need to show that eqns. 25*b* and *d* hold true, and this can be established by similar arguments, except that now, by eqn. 19*a*, we need to show that the bounds on u of eqn. 24*b* converge to u_∞. This follows from:

(i) application of the final value theorem to eqn. 16*c* according to which:

$$\begin{aligned} u_\infty &= \lim_{z\to 1}(1 - z^{-1})u(z) \\ &= 0 + \frac{a^+(1)}{b^-(1)}c_\infty + \frac{\psi_p(1)}{b^-(1)} + u_{t-1} \end{aligned} \tag{26}$$

(ii) application of the final value theorem to the bounds on u of eqn. 24*b* which are the sum of the products $h(z)f(z)$ of the last three columns of Table 1 and u_{t-1}, according to which

$$\begin{aligned} &\lim_{n_{con}\to\infty} \left(\underline{G}_{n_{con}}^4 + \underline{G}_{n_{con}}^5 + \underline{G}_{n_{con}}^6\right) + u_{t-1} \\ &= \lim_{n_{con}\to\infty} \left(\overline{G}_{n_{con}}^4 + \overline{G}_{n_{con}}^5 + \overline{G}_{n_{con}}^6\right) + u_{t-1} \\ &= \lim_{z\to 1}(1 - z^{-1}) \left[\frac{\alpha^+(z)}{b^-(z)}c(z) + \frac{z^{-n_c}a^+(z)}{\Delta(z)b^-(z)}c_\infty \right. \\ &\qquad \left. + \frac{\psi_p(z)}{\Delta(z)b^-(z)}\right] + u_{t-1} \\ &= 0 + \frac{a^+(1)}{b^-(1)}c_\infty + \frac{\psi_p(1)}{b^-(1)} + u_{t-1} \end{aligned} \tag{27}$$

Algorithm 3.1 (Computation of n_{con}):

Step 1: Set $i = n_u$.

Step 2: Compute $\underline{G}_i^k$, $\overline{G}_i^k$, k = 1–6.

Step 3: Check the conditions of Theorem 3.1. If true, set $n_{con} = i$ and stop; otherwise, set $i = i + 1$ and goto step 2.

4 Constrained CaSC algorithm

Here, we present the constrained cautious stable control (CCaSC) algorithm and prove its stability prop-

Table 1: Definitions of bounds

$h(z)$:	$\frac{\alpha^+(z)}{b^-(z)}$	$\frac{z^{-n_c}a^+(z)}{b^-(z)}$	$\frac{1}{b^-(z)}$	$\frac{\alpha^+(z)}{b^-(z)}$	$\frac{z^{-n_c}a^+(z)}{\Delta(z)b^-(z)}$	$\frac{1}{\Delta(z)b^-(z)}$
f(z):	$c(z)$	c_∞	$\psi_p(z)$	$c(z)$	c_∞	$\psi_p(z)$
lemma:	3.3	3.2	3.2	3.3	3.2	3.2
bounds:	$\underline{G}_i^1$, $\overline{G}_i^1$	$\underline{G}_i^2$, $\overline{G}_i^2$	$\underline{G}_i^3$, $\overline{G}_i^3$	$\underline{G}_i^4$, $\overline{G}_i^4$	$\underline{G}_i^5$, $\overline{G}_i^5$	$\underline{G}_i^6$, $\overline{G}_i^6$

erty. For ease of presentation we consider first the simpler case where setpoint changes are assumed not to lead to infeasibility. We deal with the general case in Section 4.2 where the feasibility assumption is removed by a suitable change of terminal constraints; this change is so devised that the resulting algorithm preserves the guarantee of stability.

4.1 Case of feasible setpoint changes

Earlier work [14–16, 21, 22] has combined sufficient (but not necessary) terminal constraints with input constraints to give stable predictive control algorithms. Here, we do the same with respect to terminal constraints which are both sufficient and necessary.

Algorithm 4.1:

Step 1: Invoke Algorithm 3.1 to compute n_{con}.

Step 2: Minimise over $\vec{c}$ the cost J of eqn. 13*e* with $\vec{y}^{\sim}$, $\Delta\vec{u}^{\wedge}$, $\vec{r}^{\sim}$ as pe eqns. 17*a*, *c* and *e* and subject to input constraints (eqns. 18*a* and *b*) for $i \le n_{con}$.

Step 3: Implement the first element of the corresponding vector of future *u*s given in eqn. 17*d*; increment t and goto step 1.

Theorem 4.1: Assume that: (i) there are no disturbances nor model uncertainty, and (ii) at startup and at times of setpoint changes the terminal and input constraints are consistent. Then the CCaSC algorithm is stabilising and gives asymptotic tracking.

Proof: Assumption (ii) gives feasibility at startup, and Theorem 3.1 and Corollary 3.1 ensure that feasibility is also guaranteed at the next instant, because at least one feasible future control trajectory exists, namely that employed at the previous time instant. This argument can be propagated until t_s the time of the next setpoint change, but assumption (ii) ensures feasibility at t_s and thus the argument above can be extended for all times.

The rest of the proof relies on showing that the cost function J is a stable Lyapunov function. This is so because, at any time t, the terminal constraints ensure that both $\tilde{r}_{t+n_y+i} - \tilde{y}_{t+n_y+i}$, $\Delta\hat{u}_{t+n_u+i}$ are zero for $i > 0$; thus the optimal J at t gives an upper bound on the value of J at $t + 1$, and the cost cannot stay at this upper bound for more than a finite number of steps (namely, the maximum of n_y and n_u), unless the cost is zero. Thus, $\tilde{y} \to \tilde{r}$, and $\Delta\hat{u} \to 0$; since $y(z) = \tilde{y}(z)/\alpha^{-}(z)$, and $\Delta u(z) = \Delta\hat{u}(z)/b^{-}(z)$, where both $\alpha^{-}(z)$ and $b^{-}(z)$ are stable, the same asymptotic property will apply with respect to y, r, and Δu.

4.2 General case

The CaSC terminal constraint (eqn. 14*a*) can be rewritten as

$$\tilde{y}_{t+i} = b^{+}(1)c_{\infty} \tag{28a}$$

with

$$c_{\infty} = \frac{\alpha^{-}(1)}{b^{+}(1)} r_{t+i} \quad i \geq n_y \tag{28b}$$

Viewed this way, the defmition of c_∞, is such that the predicted $\tilde{y}$ reaches, within n_y steps, its demanded target $\tilde{r} = \alpha^{-}(1)r$. On account of this implicit requirement, large setpoint changes may necessitate the use of large control moves which exceed the limits defined by the input constraints and thus may lead to infeasibility. An obvious way to avoid this difficulty is to allow $c_\infty$to become a degree of freedom, but constrain it to converge to the CaSC value given above. We implement this by replacing eqn. 28*b* with

$$|c_{\infty} - c_{\infty}^{0}| \leq w|c_{\infty}^{old} - c_{\infty}^{0}| \quad 0 \leq w < 1 \tag{29}$$

where $c_\infty{}^{old}$ is the optimal value of c_∞ computed at the previous time instant, and $c_\infty{}^0$ is the value of c_∞ closest to $\alpha^{-}(1)r/b^{+}(1)$ which does not violate the input constraints; as seen later the computation of $c_\infty{}^0$ simply involves a linear optimisation program.

This modification can be easily introduced into Algorithm 4.1 by including c_∞ of prediction eqns. 17*a*–*d* as an extra degree of freedom along with those contained in $\vec{c}$. However, the calculation of n_{con} has to be modified accordingly to accommodate the fact that c_∞ is unknown. This is easy to do providing that c_∞ is bounded, say by $\underline{c}_\infty$, $\bar{c}_\infty$, because then Algorithm 3.1 can be invoked exactly as given in Section 3.1 but with the bounds of $\underline{G}_i^k$, $\bar{G}_i^k$ for $k = 2$ and $k = 4$ computed as dictated by Lemma 3.3 rather than Lemma 3.2. By eqn. 29, c_∞ must lie in the interval defined by $c_\infty{}^{old}$ and $c_\infty{}^0$ and therefore will also lie (*a fortiori*) in the interval defined by $c_\infty{}^{old}$ and $\alpha^{-}(1)r/b^{+}(1)$. The latter is more conservative than the former, but is preferred because, strictly speaking, the computation of $c_\infty{}^0$ requires prior knowledge of n_{con}. This simplification may result in larger than necessary values for n_{con}, but, as per Remark 3.1, this is a matter of no computational significance. Thus the bounds on c_∞ to be used in Lemma 3.3 will be taken to be

$$\underline{c}_{\infty} = \min[c_{\infty}^{old}, \alpha^{-}(1)r/b^{+}(1)] \tag{30a}$$

$$\bar{c}_{\infty} = \max[c_{\infty}^{old}, \alpha^{-}(1)r/b^{+}(1)] \tag{30b}$$

Remark 4.1: At first sight it may appear that an unknown c_∞ would affect Lemma 3.1 (and hence 3.3 also) since the vectors v and s of eqns. 21*a* and *b* themselves would be unknown. This is not true, because as pointed out in the proof of Lemma 3.1, c_∞ makes no contribution to the first n_c elements of the prediction vectors in eqns. 17*b* and *d*. Equally, the use of a value for c_∞, which is not equal $\alpha^{-}(1)r/b^{+}(1)$ does not affect adversely the proof of Corollary 3.1: conditions (eqns. 25*a* and *c*) are unaffected because the steady-state value of the bounds on Δu of condition (eqn. 24*a*) do not depend on c_∞, whereas, following the procedure given in the proof of the corollary and by eqn. 30, it is easy to show that the steady-state value of the bounds on u of condition (eqn. 24*b*) will be

$$\frac{a^{+}(1)}{b^{-}(1)} c_{\infty}^{old} + \frac{\psi_P(1)}{b^{-}(1)} + u_{t-1} \tag{31a}$$

and

$$\frac{a(1)}{b(1)} r + \frac{\psi_P(1)}{b^{-}(1)} + u_{t-1} \tag{31b}$$

Both these values are within the physical limits of eqn. 18*a* by an amount at least as large as the ε of eqn. 19*a*.

Algorithm 4.2 (CCaSC):

Step 1: Compute n_{con} as per Algorithm 3.1 except that the bounds $\underline{G}_i^k$, $\bar{G}_i^k$ for $k = 2, 4$ now should be evaluated using Lemma 3.3 (invoking the limits on c_∞ of eqn. 30) instead of Lemma 3.2.

Step 2: $\min_{\vec{c},c_\infty} |\alpha^{-}(1)r/b^{+}(1)c_\infty|$, subject to constraints (eqns. 18*a* and *b*) and let the minimising solution to this linear program for c_∞ be $c_\infty{}^0$.

Step 3: Minimise over $\vec{c}$ and c_∞ the cost J of eqn. 13 with $\vec{y}^{\sim}$, $\Delta\vec{u}^{\wedge}$, $\vec{r}^{\sim}$ as per eqns. 17*a*, *c* and *e* and subject to input constraints (eqn. 18*a* and *b*) for $i \le n_{con}$ and to constraint eqn. 29.

Step 4: Implement the first element of the corresponding vector of future us given in eqn. 17d; increment t and goto step 1.

Theorem 4.2: Given feasibility at startup, CCaSC has guaranteed stability and asymptotic tracking.

Proof: Assume feasibility at a time $t-1$, so that, at t, c_∞^{old} defines a feasible choice for c_∞. On the other hand, c_∞^0 is feasible since it is defined to be the value of c_∞, which will take the steady state value of $\tilde{y}$ as close to the desired value of $\alpha^-(1)r$ as is possible without violating the input constraints. Therefore, constraint (eqn. 29) can be met and so replacing equality terminal constraint (eqn. 14a) with inequality terminal constraint (eqn. 29) guarantees that feasibility at $t-1$ implies feasibility at t for any setpoint change. This arguments propagates all the way back to startup which by the assumption of the theorem respects feasibility; thus feasibility is guaranteed for all time.

The consequence of inequality constraint (eqn. 29) is that if $c_\infty^0 \neq c_\infty^{old}$ for all future times, then c_∞ will converge asymptotically to the terminal equality constraint value of $\alpha^-(1)r/b^+(1)$; therefore, by Theorem 4.1 we have stability and asymptotic tracking. This arguments holds even if $c_\infty^0 = c_\infty^{old}$ so long as this is not true for all future times; a case that can only arise if c_∞^0, c_∞^{old} and c_∞^0 all 'stagnate' at some value other than $\alpha^-(1)r/b^+(1)$. This cannot happen, because, after n_{con} steps, the input constraints would hold with strict inequality and so the linear program of step 2 of the algorithm would use the available control authority to move c_∞^0 closer to $\alpha^-(1)r/b^+(1)$.

5 Illustrative examples and comparisons

Earlier constrained predictive algorithms with guaranteed stability are based on conditions which are only sufficient for the stability of the prediction equations. The algorithms presented in this paper are based on conditions which are both necessary and sufficient and therefore release as much control authority as is possible. The extra control authority thus generated can be deployed (i) in the further reduction of the cost J and hence improving performance, and/or (ii) in the reduction of the command horizon n_c (without violating feasibility) with a significant concomitant reduction in the computational burden involved in the application of constrained optimisation routines such as quadratic programming. In this Section we illustrate these points by means of two numerical examples.

5.1 Example 1

The model to be considered is given as

$$a(z) = 1 - 2.5z^{-1} + z^{-2} \quad (32a)$$

$$b(z) = 1 - 0.7z^{-1} \quad (32b)$$

and has an unstable pole at $z = 2$. The control parameters are chosen to be $n_c = 3$, $\lambda = 1$, $-\underline{\Delta u} = \overline{\Delta u} = 0.35$, $-\underline{u} = \bar{u} = 5$ and $w = 0.1$. The simulated response to a unit step demand in r are plotted in Figs. 1–4: Fig. 1 depicts the input response (tending to its steady-state value of −1.667), the output response (tending to its steady-state value of 1) and the setpoint trajectory which is 0 up to $t = 10$, and becomes 1 thereafter), Fig. 2 shows the trajectory of control increments, Fig. 3 gives a plot of n_{con}, whereas Fig. 4 plots minus the shortest distance of the predicted u and Δu from their respective limits (so that negative values imply that none of the limits are reached, a zero value corresponds to when one or more limits are reached, and a positive value would have corresponded to constraint violations). The responses are good in that they are fast and nonoscillatory, and n_{con}, although based on conservative bounding results, was no bigger than 14 throughout the entire simulation. From Fig. 4 it can be seen that, in all but four time instants ($t = 11$–14), none of the constraints was active and thereby obviating the need for quadratic programming (for t from 0 to 10, and 15 to 29).

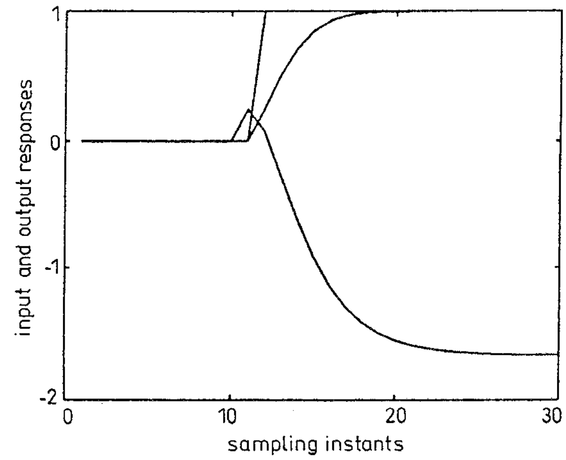

Fig. 1 *CCaSC time responses for Example 1; command horizon $n_c = 3$ input response (tending to its steady-state value of 1); Setpoint trajectory, which is 0 up to $t = 10$, becomes 1 thereafter*

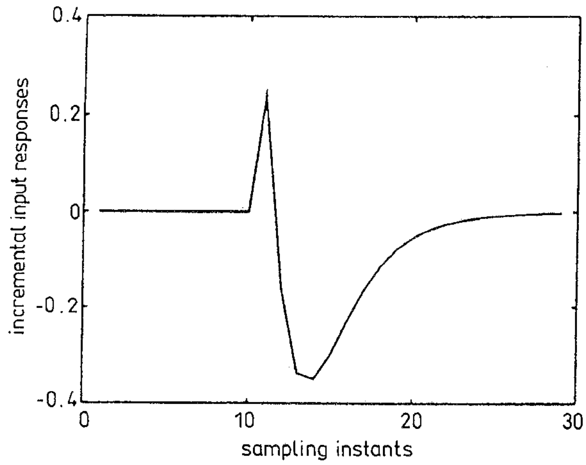

Fig. 2 *CCaSC time responses for Example 1; command horizon $n_c = 3$*
Trajectory of control increments

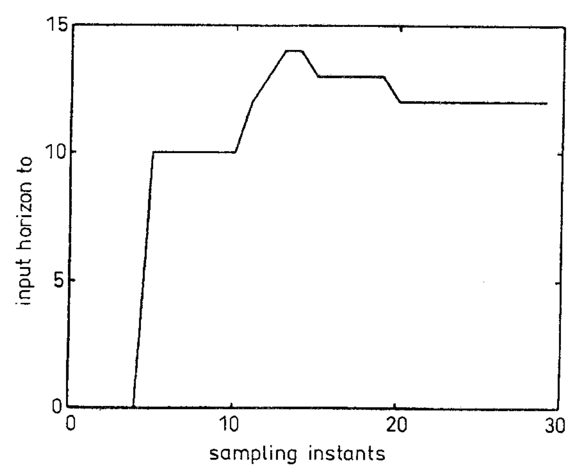

Fig. 3 *CCaSC time responses for Example 1; command horizon $n_c = 3$*
Plot of n_{con}

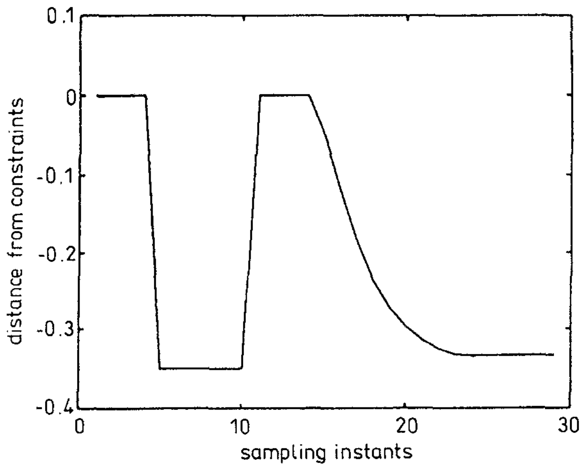

Fig. 4 *CCaSC time responses for Example 1; command horizon $n_c = 3$*
Minus shortest distance of predicted u and Δu from respective inputs

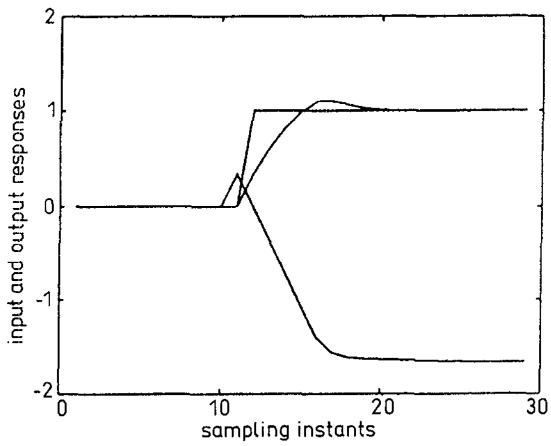

Fig. 5 *CSGPC time responses for Example 1; command horizon $n_c = 7$*
Input/Output

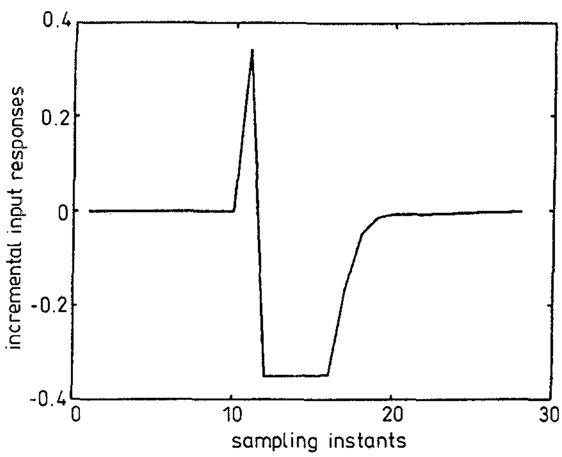

Fig. 6 *CSGPC time responses for Example 1; command horizon $n_c = 7$*
Control increment responses

In contrast to this, to retain feasibility in CSGPC (constrained stable generalised predictive control), it was necessary to raise n_c to 7 (or more) thereby increasing significantly the computational burden on the quadratic programming optimisation which for $n_c = 7$ is called upon six times (as opposed to the four calls for CCaSC with $n_c = 3$); the CSGPC input/output and control increment responses for $n_c = 7$ are shown in Figs. 5 and 6 respectively; as before the different traces are easy to identify by their steady-state values (i.e. 1 for the output response and -1.667 for the inputs response). Despite the larger value of n_c, the plots of Figs. 5 and 6 are clearly inferior to those of Figs. 1–4. To complete the comparison, in Figs. 7 and 8 we give the results for CCaSC with $n_c = 7$, which can also be seen to be superior to those of Figs. 5 and 6. It is interesting to note that the traces of Fig. 7 are not very different to those of Fig. 1, thus identifying a further benefit afforded by the use of CCaSC, namely, that it is possible to retain feasibility and therefore stability and good performance for much smaller command horizon: n_c for the plots of Fig. 7 is $n_c = 7$, whereas for the plots of Fig. 1 it is $n_c = 3$.

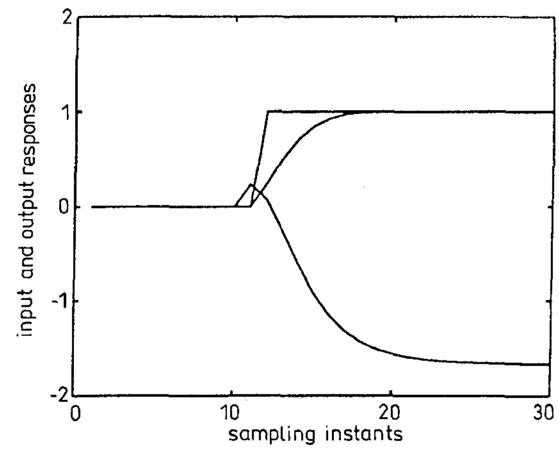

Fig. 7 *CCaSC time responses for Example 1; command horizon $n_c = 7$*
Input/Output

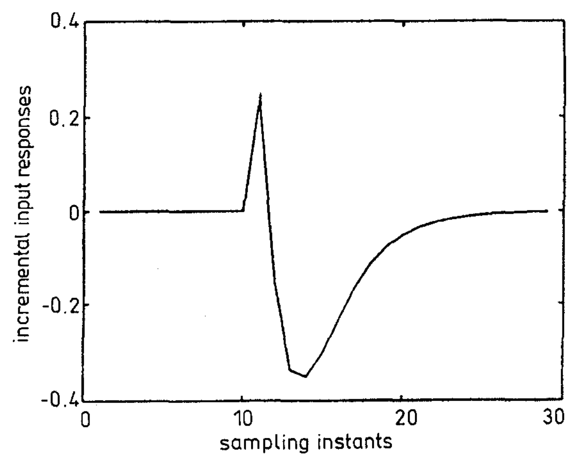

Fig. 8 *CCaSC time responses for Example 1; command horizon $n_c = 7$*
Control increment responses

5.2 Example 2

This example is included to illustrate that the definition of $\alpha^+(z)$ and $\alpha^-(z)$, which is based on the division of open loop poles into unstable and stable sets, can be modified to divide the poles into unstable/undesirable and desirable sets. Such a definition would prevent stable but slow and/or oscillatory poles affecting the response of y which is related to $\tilde{y}$ through the relationship $y(z) = \tilde{y}(z)/\alpha^-(z)$. The model to be considered is given as

$$
\begin{aligned}
a(z) &= 1 - z^{-1} + 0.01z^{-2} + 0.12z^{-3} \\
b(z) &= 1 - 2.4z^{-1} + 0.8z^{-2}
\end{aligned} \qquad (33)
$$

and has no unstable poles, but has a slow pole at $z = 0.8$ which will be included as a root of $\alpha^+(z)$ rather than $\alpha^-(z)$. The other parameters of the problem are given as $n_c = 3$, $\lambda = 0.1$, $-\underline{\Delta u} = \overline{\Delta u} = 0.2$, $-\underline{u} = \bar{u} = 0.35$, $w = 0.1$, and $n_r = 4$; the simulation results for a unit setpoint change are given in Figs. 9–12. For this example, only three calls to a quadratic program were necessary and the maximum value of n_{con} was only 6. Again, the responses are good in that they are fast and nonoscillatory.

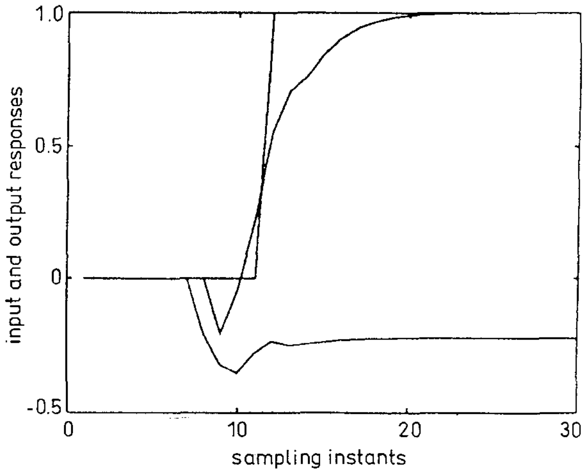

Fig. 9 *CCaSC time responses for Example 2: z = 0.8 root included in $\alpha_+(z)$; and $n_c = 3$*
Input/Output

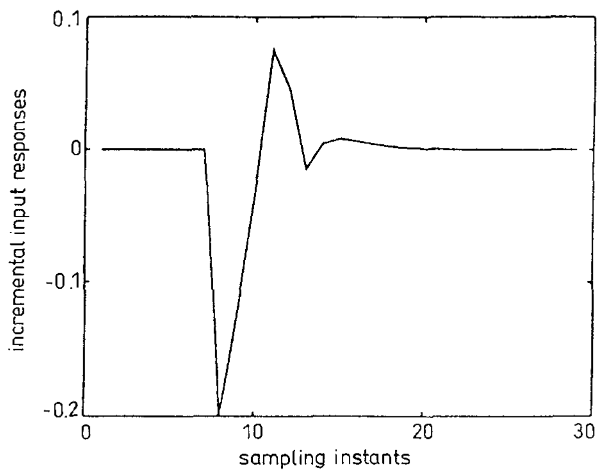

Fig. 10 *CCaSC time responses for Example 2: z = 0.8 root included in $\alpha_+(z)$; and $n_c = 3$*
Control increment responses

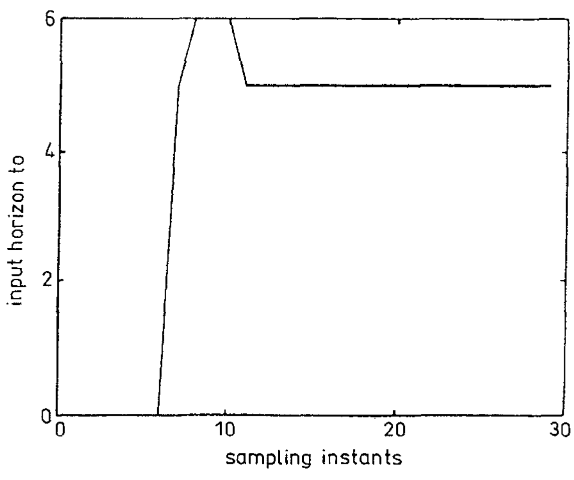

Fig. 11 *CCaSC time responses for Example 2: z = 0.8 root included in $\alpha_+(z)$; and $n_c = 3$*
Input horizon

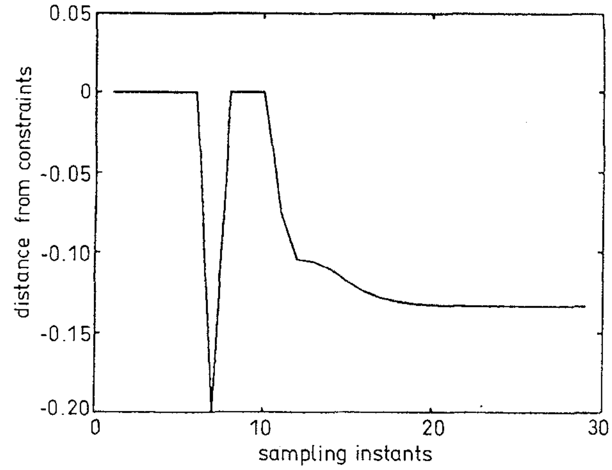

Fig. 12 *CCaSC time responses for Example 2: z = 0.8 root included in $\alpha_+(z)$; and $n_c = 3$*
Distance from constraints

6 Conclusions

Terminal constraints, which allowed the predicted output error to be an infinite length sequence, have been explored before, but infinite length sequences for the control increments have not previously been considered. This is impractical in the case where control increments are defined to be the degrees of freedom, and poses difficulties with respect to the enforcement of input constraints over an infinite horizon. In this paper we have overcome both of these problems by an appropriate characterisation of conditions, which are both necessary and sufficient for the stability of predictions and by the introduction of appropriate bounds (which are easy to compute) on the predicted input and control increments. We have thus reserved the maximum possible control authority for the purpose of reducing computational complexity (through the use of smaller command horizons) and/or improving performance.

7 References

1 GARCIA, C.E., PRETT, D.M., and MORARI, M.: 'Model predictive control: theory and practice, a survey', *Automatica*, 1989, **25**, pp. 335–348

2 MORARI, M.: 'Model predictive control: multivariable control technique choice of the 1990s' *in* CLARKE, D.W. (Ed.): 'Advances in model based predictive control' (Oxford Science Publ., Oxford, 1994)

3 TSANG, T.T.C., and CLARKE, D.W.: 'Generalized predictive control with constraints', *Proc. IEE D, Control Theory Appl.*, 1988, **135**, pp. 451–460

4 ZAFIRIOU, E.: 'Robust model predictive control of processes with hard constraints', *Comp. Chem. Eng.*, 1990, **14**, pp. 359–371

5 RICHALET, J.A., RAULT, A., TESTUD, J.L., and PAPON, J.: 'Model predictive heuristic control: applications to a industrial process', *Automatica*, 1978, **14**, pp. 413–428

6 CUTLER, C.R., and RAMAKER, B.L.: 'Dynamic matrix control: a computer control algorithm'. Proceedings of JACC, San Francisco, 1980

7 PRETT, D.M., and GILLETTE, R.D.: 'Optimization and multivariable control of a catalytic cracking unit'. Proceedings of JACC, San Francisco, 1979

8 CLARKE, D.W., and GAWTHROP, P.: 'Self-tuning controller', *Proc. IEE D, Control Theory Appl. (UK)*, 1979, **126**, pp. 633–640

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

10 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

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

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

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

14 ROSSITER, J.A., and KOUVARITAKIS, B.: 'Constrained stable generalized predictive control', *Proc. IEE D, Control Theory Appl.*, 1993, **140**, pp. 243–254
15 ZHENG, Z.Q., and MORARI, M.: 'Robust stability of constrained model predictive control'. Proceedings of ACC, San Francisco, 1993, pp. 379–383
16 MAYNE, D.Q., and POLAK, E.: 'Optimization based design and control'. Proceedings of the IFAC World Congress, 1993, Vol. 3, pp. 129–138
17 ROSSITER, J.A., KOUVARITAKIS, B., and GOSSNER, J.R.: 'Feasibility and stability results for constrained stable generalized predictive control', *Automatica*, 1995, **31**, pp. 863–877
18 BEMPORAD, A., and MOSCA, E.: 'Constraint fulfilment in feedback control via predictive reference management'. Proceedings of the 3rd IEEE CCA, Glasgow, 1994, pp. 1909–1914
19 GILBERT, E.G., KOLMANOVSKY, I., and TAN, K.T.: 'Nonlinear control of discretetime linear systems with state and control constraints: a reference governor with global convergence properties'. Proceedings of the 33rd CDC, Florida, 1994, pp. 144–149
20 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
21 RAWLINGS, J.B., and MUSKE, K.R.: 'The stability of constrained receding horizon control', *IEEE Trans.*, 1993, **AC-38**, pp. 1512–1516
22 ZHENG, A., and MORARI, M.: 'Stability of model predictive control with soft constraints'. Proceedings 33rd CDC, Florida, 1994, pp. 1018–1023