Constrained Stable Predictive Control: An l_2-optimal Approach

B. Kouvaritakis[1], J. A. Rossiter[2] and M. Cannon[1]

[1]University of Oxford, Dept. of Eng. Sci.
Parks Rd., Oxford, OX1 3PJ, UK.

[2]Loughborough University, Dept. of Math. Sciences
Leicestershire, LE11 3TU, UK.

Abstract

Perturbations on l_2-optimal solutions are used for the derivation of efficient/feasible predictive algorithms.

1 Introduction

Model based predictive control (MBPC) has attracted a lot of research both in the unconstrained case (eg. [1, 2, 3, 4]) and the constrained case (eg.[5, 6]). The constrained case involves an infinite dimensional problem, but practicable algorithms can be derived by performing the optimization over a finite number of future control moves. This restriction has been removed in [7, 8] which is based on conditions which are sufficient and necessary for the guarantee of stability. The sequence of predicted future control moves is no longer forced to be a Finite Impulse Response (FIR), but the degrees of freedom available for the purposes of optimisation is still taken to be finite.

In the absence of constraints, a desirable solution is provided by LQR, but on account of the restricted number of degrees of freedom in the earlier MBPC algorithms, it was not possible to recover an LQR type of solution, even for the part of the trajectories that lies beyond the "fast transient" horizon, say N, for which constraint violation is not a problem. To remedy this [9] consider a state space description and propose an Algorithm which optimizes over a FIR sequence of control moves with the modification that control law switches to the LQR law beyond N.

Here we propose an alternative approach which considers the class of stable predictions and within this identifies the unconstrained l_2-optimal solution. This may not be feasible, and to cope with this problem we perturb the optimal so as to avoid constraint violations. A simple perturbation consists of a replacement of the target setpoint value by a slack variable which is then chosen to be as close to the actual setpoint as the constraints will allow. The advantage of the resulting algorithm is that it avoids the need for Quadratic Programming (QP) routines which can be computationally very demanding and that it is circumvents infeasibility problems for all achievable setpoints. A second alternative considers a FIR perturbation in the class of stable l_2-optimal prediction equations and uses QP over a finite number of degrees of freedom, n_c, invoking the constraints over a finite horizon n_{con}. This alternative shares many features in common with [9] but results in a different control law. It is conjectured that a comparison between these two approaches will be example dependent and that n_c often will be smaller than N thereby reducing the computation involved in QP. It is argued here that through the introduction of a slack variable for the setpoint it is possible to take n_c to be as small as required (and our choice will be $n_c = 0$). The paper concludes with a simple algorithm for the determination of n_{con} which improves upon earlier algorithms and with a numerical examples section.

2 Stable/optimal predictions

Consider the scalar, discrete-time, time-invariant model:

$$y_{t+i} = z^{-1}\frac{b(z)}{a(z)}u_{t+i} = z^{-1}\frac{b(z)}{A(z)}\Delta u_{t+i}, \; i = 1, 2, \dots$$
$$A(z) = (1 - z^{-1})a(z); \; \Delta u_t = u_t - u_{t-1} \tag{1}$$

where y_t is the system output at time t, and u_t is the corresponding input; z is the z-transform variable and z^{-1} is the backward shift operator. Although the objectives of the various predictive control algorithms may differ, in common to all is the requirement for a guarantee of stability, and this in turn requires the use of control laws which result in stable input and output predicted behaviour. The following section gives a brief derivation for the class of such stable predictions.

2.1 Stable input/output predictions

Let $\hat{u}_{t+i}, \; i = 0, 1, \dots$ be a predicted input trajectory and let $\hat{y}_{t+i}, \; i = 1, 2, \dots$ be the corresponding predicted output trajectory. Then from (1) we have:

$$\hat{Y}(z) = \frac{B(z)\Delta\hat{U}(z) + P(z)}{A(z)}; \; B(z) = z^{-1}b(z) \tag{2}$$

where $P(z)$ accounts for initial conditions at t. It is emphasised that $\hat{Y}(z)$ is the z-transform of the predicted output and involves future values only. Next assume for convenience (without loss of generality) that the setpoint trajectory is a constant, r_0. The resulting reference signal $r_t = r_0$ and corresponding predicted error signal $\hat{e}_t = r_t - \hat{y}_t$ have z-transforms:

$$R(z) = \frac{r_0}{1-z^{-1}}, \ \hat{E}(z) = \frac{Q(z) - B(z)\Delta\hat{U}(z)}{A(z)} \tag{3}$$

where $Q(z) = a(z)r_0 - P(z)$. Thus the predicted error will be stable if and only if:

$$Q(z) - B(z)\Delta\hat{U}(z) = A_+(z)\Phi(z) \tag{4}$$

where $\Phi(z)$ is a polynomial or stable transfer function; it is assumed that $A(z) = A_+(z)A_-(z)$ and that the roots of $A_-(z)$ are all inside the unit circle, whereas those of $A_+(z)$ are all on, or outside the unit circle. From eqn. (4) we have that the predicted input behaviour will be stable if and only if:

$$A_+(z)\Phi(z) + B_+(z)\Psi(z) = Q(z) \tag{5}$$

where, as with $A(z)$, $B(z) = B_+(z)B_-(z)$; and $\Psi(z)$ is a polynomial or stable transfer function.

Theorem 2.1 *The class of stable input and output predictions is defined by:*

$$\hat{E}(z) = \frac{\Phi_p(z)}{A_-(z)} + \frac{B_+(z)}{A_-(z)}F(z) \tag{6}$$

$$\Delta\hat{U}(z) = \frac{\Psi_p(z)}{B_-(z)} - \frac{A_+(z)}{B_-(z)}F(z) \tag{7}$$

where $F(z)$ is an arbitrary stable transfer function and $\Phi_p(z)$, $\Psi_p(z)$ denote a pair of particular solutions to the Bezout identity of eqn. (5).

Proof: This follows from (3), (4) and the fact that

$$\begin{aligned}\Phi(z) &= \Phi_p(z) + B_+(z)F(z)\\ \Psi(z) &= \Psi_p(z) - A_+(z)F(z)\end{aligned}$$

define the entire class of solutions to eqn. (5). □

2.2 Unconstrained optimal

To define the framework for the constrained predictive control we first consider the unconstrained case. As is usual we consider as our cost, a function J that penalizes predicted error and control activity. In particular we shall take J to be:

$$\begin{aligned}J &= \sum_{i=1}^{\infty}[\hat{e}_{t+i}^2 + \lambda^2\Delta\hat{u}_{t+i-1}^2] \\ &= \frac{1}{j2\pi}\oint[|\hat{E}(z)|^2 + \lambda^2|\Delta\hat{U}(z)|^2]\frac{dz}{z} \\ &= \frac{1}{j2\pi}\oint\left\|\begin{bmatrix}\frac{\Phi_p(z)}{A_-(z)}\\ \frac{\lambda\Psi_p(z)}{B_-(z)}\end{bmatrix} + \begin{bmatrix}\frac{B(z)}{\Lambda(z)}\\ \frac{-\lambda A(z)}{\Lambda(z)}\end{bmatrix}F'(z)\right\|^2\frac{dz}{z},\end{aligned} \tag{8}$$

where $F'(z) = \Lambda(z)F(z)/A_-(z)B_-(z)$ and use has been made of Parseval's theorem and of (6,7); $\|\cdot\|$ is the Euclidean norm and all contour integrals are taken over the unit circle ($|z|=1$). Here $\Lambda(z)$ is the stable solution of the identity:

$$B^*(z)B(z) + \lambda^2A^*(z)A(z) = \Lambda^*(z)\Lambda(z) \tag{9}$$

where $(\cdot)^*$ denotes replacement of z^{-1} by z; as expected $\Lambda(z)$ will be shown below to be the l_2-optimal closed-loop pole polynomial. Then making use of the transformation (which is unitary on the unit circle):

$$\begin{bmatrix}B^*(z)/\Lambda^*(z) & -\lambda A^*(z)/\Lambda^*(z)\\ \lambda A(z)/\Lambda(z) & B(z)/\Lambda(z)\end{bmatrix}$$

we deduce that J can be written as:

$$\begin{aligned}J &= \frac{1}{j2\pi}\oint|G(z)+F'(z)|^2\frac{dz}{z}\\ &+\frac{\lambda}{j2\pi}\oint\left|\frac{A_+(z)\Phi_p(z)+B_+(z)\Psi_p(z)}{\Lambda(z)}\right|^2\frac{dz}{z};\\ G(z) &= \frac{B^*(z)\Phi_p(z)}{\Lambda^*(z)A_-(z)} - \lambda^2\frac{A^*(z)\Psi_p(z)}{\Lambda^*(z)B_-(z)}.\end{aligned}$$

The 2nd term of J is constant and can be ignored during the minimization which now becomes:

$$\min_{F(z)} J'; \ J' = \frac{1}{j2\pi}\oint|G_+(z)+F''(z)|^2\frac{dz}{z},$$

where $F''(z) = F'(z) + G_-(z)$ and $G(z) = G_+(z) + G_-(z)$, with $G_-(z) = S_1(z)/A_-(z) - \lambda^2S_2(z)/B_-(z)$, $G_+(z) = T_1(z)/\Lambda^*(z) - \lambda^2T_2(z)/\Lambda^*(z)$, and $S_1(z)$, $T_1(z)$, $S_2(z)$, $T_2(z)$ are the minimal order solutions of the Bezout identities:

$$\Lambda^*(z)S_1(z) + A_-(z)T_1(z) = B^*(z)\Phi_p(z) \tag{10}$$

$$\Lambda^*(z)S_2(z) + B_-(z)T_2(z) = A^*(z)\Psi_p(z). \tag{11}$$

With these definitions it is easy to show that:

$$J' = \frac{1}{2\pi}\int_0^{2\pi}|G_+(e^{j\theta})|^2d\theta + \frac{1}{2\pi}\int_0^{2\pi}|F''(e^{j\theta})|^2d\theta \tag{12}$$

which is clearly minimized for $F''(z) = 0$, ie. when:

$$F(z) = F_{opt}(z) = \frac{-B_-(z)S_1(z) + \lambda^2A_-(z)S_2(z)}{\Lambda(z)}. \tag{13}$$

The proof of (12) relies on an application of Cauchy's Theorem for the case when $F''(z)$ is strictly proper (when viewed as a rational function in z^{-1}), or in all other cases one can resort to the application of the Residue Theorem (applied to all the simple poles).

The following optimal predicted control increment and output error equations can be derived by substitution of eqn. (13) into eqns. (6,7):

$$\hat{E}_{opt}(z) = \frac{\Lambda(z)\Phi_p(z) - B(z)S_1(z)}{A_-(z)\Lambda(z)} + \lambda^2\frac{B_+(z)S_2(z)}{\Lambda(z)}$$
$$\Delta\hat{U}_{opt}(z) = \frac{\Lambda(z)\Psi_p(z) - \lambda^2 A(z)S_2(z)}{B_-(z)\Lambda(z)} + \frac{A_+(z)S_1(z)}{\Lambda(z)} \qquad (14)$$

Although it appears that the stable poles and minimum phase zeros appear as poles in the optimal prediction above. However using the Bezout identities of (10) and (5) it is easy to show that the numerator of the first term in the expression for $\hat{E}_{opt}(z)$ vanishes at the roots of $A_-(z)$; similarly using (11) and (5) it is easy to show that the numerator of the first term of $\Delta\hat{U}_{opt}(z)$ vanishes at the roots of $B_-(z)$.

3 Predictive algorithm

In most practical applications the input values u_t and/or the control increments, Δu_t, are subject to (hard) constraints of the form:

$$\underline{u} \leq u_{t+i} \leq \overline{u},\ \underline{\Delta u} \leq \Delta u_{t+i} \leq \overline{\Delta u};\ i \geq 0 \qquad (15)$$

where $\underline{u}$, $\overline{u}$, $\underline{\Delta u}$, $\overline{\Delta u}$ are assumed to be constants satisfying the obvious steady state requirements:

$$\underline{u}+\epsilon \leq \frac{a(1)}{b(1)}r_0 \leq \overline{u}-\epsilon;\ \underline{\Delta u}+\epsilon' \leq 0 \leq \overline{\Delta u}-\epsilon';\ \epsilon, \epsilon' > 0 \qquad (16)$$

Remark 3.1 On account of the stable nature of (6,7) it is not necessary to invoke condition (15) for all positive i; indeed there will always exist [8, 10] a finite constraint horizon, n_{con}, with the property that if (15) holds true for all $0 \leq i \leq n_{con}$, it will hold for all positive i. For convenience n_{con} should be chosen to be as small as possible, and suitable algorithms for the selection of n_{con} have already been proposed [8, 10]; an alternative algorithm which enables the determination of non-conservative n_{con} will be described in section 5.

The existence of hard input constraints implies that the optimal solution of (13) may not be feasible. In the next section we deal with the general constrained predictive control strategy, but here we begin with the case where the constraints of (15) are wide enough as not to cause infeasibility. In this case the optimal predicted control law of (14) is feasible and suggests as the current optimal control increment the value:

$$\Delta u_t = \qquad (17)$$
$$\lim_{z\to\infty}\left[\frac{\Psi_p(z)}{B_-(z)} - \lambda^2\frac{A(z)S_2(z) + A_+(z)B_-(z)S_1(z)}{B_-(z)\Lambda_-(z)}\right].$$

Thus given the previous control u_{t-1}, it is possible to can be implemented and the procedure repeated at the next sampling instant, thus yielding the following l_2-optimal predictive control algorithm:

Algorithm 3.1 (i). Set t=0.

(ii). Given the past outputs and control increments, y_{t-i}, Δu_{t-i-1}, $i = 0, 1, \ldots, n$ (where n is the degree of $A(z)$, and for convenience it has been assumed that the degree of $B(z)$ is one less that of $A(z)$) form the polynomial $P(z)$.

(iii). For the $P(z)$ above and the given setpoint r_0, compute $Q(z)$ in eqn. (3) and solve the Bezout identity of eqn. (5) for $\Phi_p(z)$, $\Psi_p(z)$. For simplicity these two polynomials could be taken to be the minimal order solutions of (5).

(iv). For the $\Phi_p(z)$, $\Psi_p(z)$ of step (iii) obtain the minimal order solution to the Bezout identities (10,11) and introduce these in (17) to compute the current optimal control input u_t.

(v). Implement u_t, increment t by 1; GOTO (ii).

Remark 3.2 The implementation of the algorithm is not in its most efficient form; it is clear from (14) that the z-transform of the optimal predicted error and control increment trajectories is of the form:

$$\hat{E}_{opt}(z) = \frac{\Phi_p(z)}{A_-(z)} + \frac{B_+(z)}{A_-(z)\Lambda(z)}N(z)$$
$$\Delta\hat{U}_{opt}(z) = \frac{\Psi_p(z)}{B_-(z)} - \frac{A_+(z)}{B_-(z)\Lambda(z)}N(z)$$

where $N(z)$ is a polynomial of given order (dictated by (14)). A more efficient way of computing the optimal $N(z)$ is to: (i) form a vector $\mathbf{N}$ out of the coefficients of $N(z)$ taken in ascending powers of z^{-1}, (ii) use the Theorem of the Residues to express the cost of (8) as a quadratic function of $\mathbf{N}$. The optimal choice can then be computed by setting the gradient of J with respect to $\mathbf{N}$ equal to zero. Another efficient alternative can be derived by writing :

$$\hat{E}_{opt}(z) = \frac{N_E(z)}{\Lambda(z)};\ \Delta\hat{U}_{opt}(z) = \frac{N_{\Delta U}(z)}{\Lambda(z)} \qquad (18)$$

and recognising that $N_E(z)$ and $N_{\Delta U}(z)$ satisfy the Bezout identity:

$$A(z)N_E(z) - B(z)N_{\Delta U}(z) = \Lambda(z)Q(z)$$

Solving this for polynomials $N_E(z)$, $N_{\Delta U}(z)$ of the appropriate degrees (as indicated by eqn. (14)) will generate the unique optimal solution.

Theorem 3.1 *In the absence of model mismatch and disturbances, Algorithm 3.1 has guaranteed stability and will cause the actual error signal $e_t = r_0 - y_t$ to*

Proof: This is easy to establish by showing that the optimal cost function is a monotonically decreasing function of time. This in turn follows from the feasibility assumption and the stability of control increment and error predictions of eqns. (6,7). □

4 Constrained algorithm

In general the optimal solution of eqn. (14) may not be assumed to be feasible due to the presence of constraints. At times of infeasibility therefore it becomes necessary to perturb $F(z)$ away from its optimal choice given in eqn. (13). We will do this subject to the obvious requirement that the new algorithm will converge in finite time to a $F_{opt}(z)$ which is feasible. In this section we consider two alternative algorithms that satisfy both these requirements.

4.1 Using a setpoint slack variable

Depending on the size of the setpoint change, r_0, it may not be possible to implement the optimal solution of (14) without violating the constraints of (15) and r_0 will be termed respectively feasible (for sufficiently small r_0) and infeasible (for all other r_0). A way of avoiding infeasibility is to adopt, in place of r_0, the feasible setpoint value r_0^* which is nearest to r_0. At the next time instant, r_0^* will still be feasible, but a new r_0^* which is nearer to r_0 can be defined and adopted as the new target. Because of the stable nature of the predicted input trajectories the difference in r_0^* at successive times cannot be zero over an infinite time interval. It follows that after a finite (but unknown) time the input and control increments will fall well within their respective constraints so that r_0 itself will become feasible. From then on the procedure outlined will become identical to that described in section 3 and will therefore lead to stability and asymptotic l_2-optimal tracking.

Theorem 4.1 *The conmputation of r_0^*, the feasible setpoint is a linear program.*

Proof: It was already remarked in section 2 that the numerator and denominator of the first term in the expression for $\Delta\hat{U}_{opt}$ of eqn. (14) have $B_-(z)$ as a common factor, as a result of which we may write:

$$\Lambda(z)\Psi_p(z) - \lambda^2 A(z)S_2(z) - B_-(z)A_+(z)S_1(z) = B_-(z)N_{\Delta U}(z)$$

which is linear in the coefficients of $\Psi_p(z)$, $S_1(z)$, $S_2(z)$, $N_{\Delta U}(z)$. Hence the vector $\mathbf{N}_{\Delta U}$ of the coefficients of $N_{\Delta U}(z)$ can be written as a linear combination of the vectors $\boldsymbol{\Psi}_p$, $\mathbf{S}_1$, $\mathbf{S}_2$ of the coefficients of $\Psi_p(z)$, $S_1(z)$, $S_2(z)$. By (10,11) it is obvious that $\mathbf{S}_1$, $\mathbf{S}_2$ are linear functionals of $\boldsymbol{\Phi}_p$, $\boldsymbol{\Psi}_p$, respectively, and these in turn from (5) can be seen to be linear in $\mathbf{Q}$, the vector of coefficients of $Q(z)$. Combining these observations with the definition of $Q(z)$ (as given in (3)) we deduce that:

$$\mathbf{N}_{\Delta U} = \mathbf{V}_1 x + \mathbf{V}_2 \tag{19}$$

where $\mathbf{V}_1$, $\mathbf{V}_2$ are known vectors and where x has been used in place of r_0. Then, denoting by h_i, $i = 0, 1, \ldots$ the elements of the impulse response of $1/\Lambda(z)$, we have from eqn. (18) that:

$$\Delta\hat{u}_{t+i} = \mathbf{H}_i^T(\mathbf{V}_1 x + \mathbf{V}_2),\ i = 0, 1, \ldots n_{con}, \tag{20}$$

where $\mathbf{H}_i^T = [h_i\ h_{i-1}\ \ldots\ h_0\ 0]$. We also have that:

$$\hat{u}_{t+i} = \sum_{j=0}^{j=i} \Delta\hat{u}_{t+j} + u_{t-1},\ i = 0, 1, \ldots n_{con}.$$

Thus both the predicted absolute and incremental values for the optimal inputs are affine in r_0^* and therefore the minimization problem

$$r_0^* = \arg\{\min_x |r_0 - x|;\ \text{subject to constraints (15)}\} \tag{21}$$

constitutes a linear program. □

Algorithm 4.1 This is identical to Algorithm 3.1, except that infeasible r_0 are to be replaced by the optimal solution r_0^* to the minimization problem of (21).

Theorem 4.2 *Algorithm 4.1 has guaranteed stability and will track asymptotically any setpoint r_0. Feasible setpoint trajectories will be tracked optimally (in the l_2 sense), whereas for r_0 infeasible the algorithm will steer the system in such a way that r_0 will become feasible at some finite moment in time, say $t = t_0$ and thereafter the algorithm will give trajectories with are optimal with respect to the initial conditions at t_0.*

Proof: For r_0 feasible Algorithm 4.1 is identical to Algorithm 3.1 and the proof proceeds as per that given for Theorem 3.1. On the other hand at all t for which r_0 is infeasible, the algorithm will adopt a sequence of different setpoints r_0^* which by their definition (eqn. (21)) will tend towards r_0. This together with the stable nature of the predictions of eqn. (14) (computed for r_0^*) imply that within finite time, say $t = t_0$, r_0 will become feasible. From then on (in the absence of model mismatch and disturbances) the algorithm will follow the trajectories of eqn. (14) which are optimal with respect to the l_2 cost of eqn. (8) for the initial conditions at $t = t_0$. □

Remark 4.1 An important advantage of Algorithm 4.1 is that though it clearly addresses the problem of optimising performance it does so in a manner that avoids computationally expensive techniques such as QP which is used widely in the constrained predictive control literature. Instead, all that is required here is the solution of a linear program.

4.2 Perturbations on the optimal $F(z)$

An alternative way of regaining the feasibility of r_0 is to detune the optimal control law of (14) via the introduction of extra degrees of freedom. In particular, with $F(z) = F_{opt}(z) + C(z)$ in (6,7) one could stipulate the detuned predicted trajectories:

$$\begin{aligned}\hat{E}(z) &= \hat{E}_{opt}(z) + \frac{B_+(z)}{A_-(z)}C(z) \\ \Delta\hat{U}(z) &= \Delta\hat{U}_{opt}(z) - \frac{A_+(z)}{B_-(z)}C(z)\end{aligned} \qquad (22)$$

where $C(z)$ is a polynomial of finite degree whose coefficients are to be used in the minimization of J subject to constraints (15). Introducing these expressions into the cost of eqn. (8) and using Cauchy's Integral Theorem we deduce that:

$$\begin{aligned}J &= J_{opt} + \frac{1}{j2\pi}\oint\left[\left|\frac{B_+(z)}{A_-(z)}\right|^2 + \lambda^2\left|\frac{A_+(z)}{B_-(z)}\right|^2\right]|C(z)|^2\frac{dz}{z} \\ &= J_{opt} + \mathbf{C}^T S\mathbf{C}\end{aligned} \qquad (23)$$

where the Theorem of the Residues has been used to express the perturbation on the cost as a quadratic function of the vector $\mathbf{C}$ of the coefficients of $C(z)$; S is a known symmetric positive definite matrix and J_{opt} denotes the value of the cost J for $F(z) = F_{opt}(z)$.

Theorem 4.3 *Let $n_c - 1$ denote the degree of the polynomial $C(z)$ and assume that $\Delta\hat{U}_{opt}(z)$ for a setpoint r_0 is feasible at time t. Then n_c can always be chosen to be large enough (but finite) so that at the next time instant the $\Delta\hat{U}(z)$ of eqn. (22) computed for a new setpoint $r_0 + \delta r_0$ will be feasible, irrespective of how large r_0; it is assumed that r_0 does not violate the steady state constraints (16).*

Proof: This is straightforward, but involves a fair amount of algebra and is given only in outline. Here is a sketch of the key steps: Let $\Delta\hat{U}_{t+1|t}(z)$ denote the z-transform of the predicted values for the control increments at $t+1$ given the optimal control law at t. Then it is possible to show that:

$$\Delta\hat{U}_{t+1}(z) = \Delta\hat{U}_{t+1|t}(z) + \frac{\alpha A(z)}{\Lambda(z)}\delta r_0 - \frac{A_+(z)}{B_-(z)}C(z) \qquad (24)$$

where α is a constant. By assumption $\Delta\hat{U}_{t+1|t}(z)$ is feasible, therefore all that is required is a choice of $C(z)$ that will counteract the effect of r_0. Clearly if it were possible to chose:

$$C(z) = \alpha\frac{B_-(z)A_-(z)}{\Lambda(z)}\delta r_0 \qquad (25)$$

then the effect of r_0 would be cancelled altogether. However this would require an $C(z)$ whose sequence of coefficients is infinitely long. Yet given the stability of the RHS of eqn. (24) it follows that a finite time t' exists such that the sum of the impulse responses of the first two terms (of the RHS (24) is less than the ϵ of condition (16); therefore $C(z)$ need only match the MacLaurin expansion (in terms of powers of z^{-1}) of the RHS of eqn. (25) up to the t'-th power. □

From eqn. (22) it is possible to write the input constraints of (15) as linear inequalities in $\mathbf{C}$:

$$\begin{aligned}\underline{\Delta u} &\leq [\Delta\hat{u}_{opt}]_{t+i} + \mathbf{G}_i^T\mathbf{C} \leq \overline{\Delta u} \\ \underline{u} &\leq \sum_{j=0}^{j=i}\{[\Delta\hat{u}_{opt}]_{t+j} + \mathbf{G}_i^T\mathbf{C}\} + u_{t-1} \leq \overline{u}\end{aligned} \qquad (26)$$

for $i = 0, 1, \ldots n_{con}$, where $\mathbf{G}_i^T = [g_i \; g_{i-1} \; \ldots \; g_0 \; 0]$, and g_i, $i = 0, 1, \ldots$ are the elements of the impulse response of $-A_+(z)/B_-(z)$.

Algorithm 4.2 As per Algorithm 3.1, with $\Delta\hat{U}_{opt}(z)$ of (14) replaced by $\Delta\hat{U}(z)$ of (29) for $C(z)$ corresponding to the solution of:

$$\mathbf{C}_{opt} = \arg\{\min_{\mathbf{C}} J;\ \text{subject to constraints (26)}\} \qquad (27)$$

Note that $\mathbf{C}_{opt}$ can be obtained using standard QP.

Theorem 4.4 *Algorithm 4.2 has guaranteed stability and asymptotic tracking. Furthermore, for r_0 feasible it will be identical to Algorithm 3.1, whereas for r_0 infeasible it will steer the system in such a way that after a finite time t_0 the optimal control law of eqn. (14) will be feasible and thereafter will give input/output trajectories which are l_2-optimal with respect to the initial conditions at t_0.*

Proof: From the form of the cost J of eqn. (23) it is obvious that for feasible r_0 the optimal choice for $\mathbf{C}$ is $\mathbf{C} = \mathbf{0}$ which shows that Algorithm 4.2 will be identical to Algorithm 3.1 for r_0 feasible.

For r_0 infeasible and n_c large enough, the QP problem of eqn. (27) is feasible and hence it is easy to show that J will be a monotonically decreasing function of time. This relies on the fact that the $\mathbf{C}_{opt}$ computed at t provides a predicted control trajectory which is feasible and will remain so at the next instant if the control law is not changed but will (on account of the infinite horizon used in the cost) result in a smaller cost which may be reduced further upon further optimization (over $\mathbf{C}$) at $t+1$. The monotonic decrease of cost combined with earlier arguments can be used to show that within finite time the control law of (14) will become feasible at which point the optimal choice for $\mathbf{C}_{opt}$ is $\mathbf{C}_{opt} = \mathbf{0}$, and this completes the proof. □

Remark 4.2 For certain initial conditions and large r_0 it may be necessary to use a large n_c in order to ensure feasibility. However n_c is the number of elements in the vector $\mathbf{C}$ and as such represents the number of degrees of freedom available in the QP problem of eqn. (27). This may increase the computational load unacceptably. At such times it is perfectly possible to use a smaller n_c and allow the working setpoint to become a slack variable along the lines indicated in section 4.1. The earlier arguments can be applied once again to prove that the resulting algorithm will steer the system in such a way as to make r_0 feasible within finite time, upon which finite time the algorithm will automatically revert to Algorithm 3.1.

5 The selection of n_{con}

For simplicity we consider here rate constraints only; the extension to the general case is obvious. Let ν denote the dimension of $\mathbf{N}_{\Delta U}$, and invoke the rate constraints of (15) to (19,20) to get:

$$\left[\frac{2\mathbf{H}_i^T}{\overline{\Delta u}-\underline{\Delta u}}\mathbf{N}_{\Delta U} - \frac{\overline{\Delta u}+\underline{\Delta u}}{\overline{\Delta u}-\underline{\Delta u}}\right]^2 \leq 1;\ i=0,1,\ldots\nu \tag{28}$$

which has the form of the "bounded noise" identification problem [11]. Following this work it is possible to use a Kalman iteration type of algorithm to define an ellipsoid within which $\mathbf{N}_{\Delta U}$ must lie:

$$(\mathbf{N}_{\Delta U} - \boldsymbol{\theta}_0)^T V (\mathbf{N}_{\Delta U} - \boldsymbol{\theta}_0) \leq 1 \tag{29}$$

where $\boldsymbol{\theta}_0$ is a constant vector and V a symmetric positive definite matrix. Combining the above with eqns. (19,20) it is easy to deduce that:

$$L_i \leq \Delta\hat{u}_{t+i} \leq U_i;\ i > \nu \tag{30}$$

where $L_i = \mathbf{H}_i^T\boldsymbol{\theta}_0 - \beta_i$, $U_i = \mathbf{H}_i^T\boldsymbol{\theta}_0 + \beta_i$ and $\beta_i = (\mathbf{H}_i^T V^{-1}\mathbf{H}_i)^{1/2}$.

Theorem 5.1 *Let i_{con} be the smallest integer i for which L_i and U_i in (30) lie in the interval $[\underline{\Delta u}, \overline{\Delta u}]$ for all $i > i_{con}$ and define $n_{con} = \max\{\nu, i_{con}\}$. Then if the rate constraints of (15) are satisfied for all $i \leq n_{con}$, they will also be satisfied for all i.*

Proof: If the rate constraints hold true for up to $i = n_{con}$, then $\mathbf{N}_{\Delta U}$ will lie in the ellipsoid of (29) so that (30) will hold true. Thus if the predicted control law is feasible for all $i \leq n_{con}$, it will also be feasible for all $i > i_{con}$ and also for all $i > n_{con}$. □

Remark 5.1 None of (28,29,30) depend on time t, hence the computation of n_{con} can be performed off line; furthermore, the computational burden is clearly small. Despite this, simulation has shown that the values of n_{con} suggested by Theorem 5.1 are smaller (in cases by a considerable amount) to those given by the algorithms proposed in [8, 10].

6 Illustrative Example

An example is included to illustrate the advantages of Algorithms 4.1 and 4.2 over existing algorithms. Of the existing algorithms we select Infinite Horizon Predictive Control (IHPC) [7] as representative, because it considers the whole class of stable predictions and does not constrain the sequence of the future control increments to be FIR's. The means of comparison will be plots of simulated responses and an evaluation $\tilde{J}$ of the cost J over the actual error and control increment values. We emphasise here that because of the nonlinear nature of the control and the use of $\tilde{J}$, comparisons are example dependent; the example discussed below has been chosen to: (i) be simple; (ii) demonstrate an instance when the algorithms developed in this paper afford significant advantages.

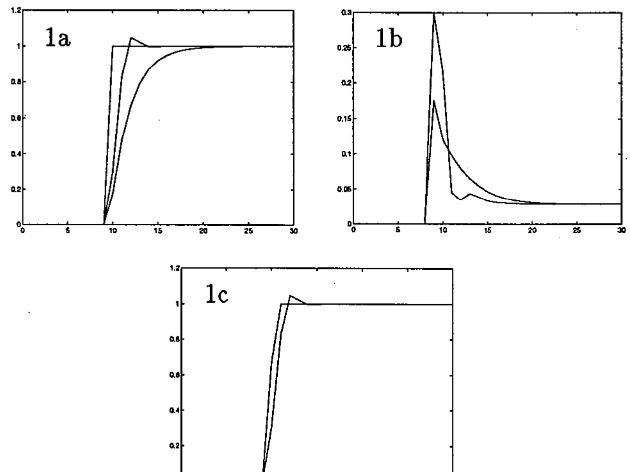

Figure 1: Responses for IHPC, Algorithm 4.2 and Algorithm 4.1; (a) outputs and setpoint, (b) inputs, (c) output and slack variable for Algorithm 4.1

The parameters for this numerical example are:

$$\begin{gathered} a(z) = 1 - 1.55z^{-1} + 0.65z^{-2} - 0.076z^{-3}, \\ b(z) = 1 + 0.5z^{-1} - 0.66z^{-2}, \\ \overline{\Delta u} = -\underline{\Delta u} = 0.3,\ \overline{u} = -\underline{u} = 0.6, \\ n_c = 1,\ \lambda = 0.5,\ r_0 = 1 \end{gathered} \tag{31}$$

With zero initial conditions the chosen setpoint is feasible, and so all three algorithms produce stable responses as shown in Fig. 1a and Fig. 1b. From these it is seen that IHPC, due to its "cautious" aspect (it replaces only unstable and undesirable/slow open loop poles) produces a slow output response, and does not use the whole range available for the control increments ($|\Delta u| < 0.155$ for IHPC in Fig 1b). In contrast Algorithms 4.1 and 4.2 hit the limit of 0.3 and produce much faster responses (these are indistinguishable from eachother in Fig.s 1a,b). This is due to the wide input constraints which result in a sequence $\{0.68,\ 1,\ 1,\ \ldots\}$ of slack setpoint variables used by Algorithm 4.1 (as shown in Fig. 1c), deviating from the desired target of $r_0 = 1$ only at the first sampling instant. The respective costs are:

$$\tilde{J}_{\text{IHPC}} = 1.146;\ \tilde{J}_{4.1} \simeq \tilde{J}_{4.2} = 0.585;\ \tilde{J}_{\text{U}} = 0.470$$

where the cost $\tilde{J}_{\text{U}}$ for the unconstrained case is included to show that the constraints, though loose, have a significant effect.

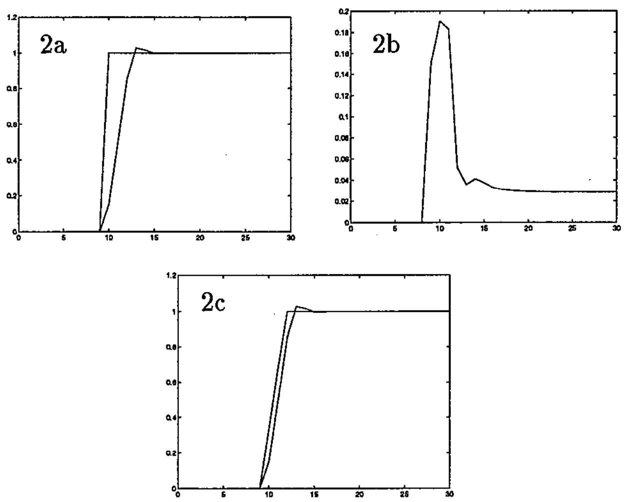

Figure 2: Response for Algorithm 4.1; (a) output and setpoint, (b) input, (c) output and slack variable

The effect of constraints is very pronounced with a rate limit of 0.15 in (31). The costs are then:

$$\tilde{J}_{\text{IHPC}} = 1.263;\ \tilde{J}_{4.1} = 1.018;\ \tilde{J}_{4.2} = 170.2.$$

Clearly, with only one degree of freedom Algorithm 4.2 is unstable due to infeasibilty problems, whereas IHPC, though not unstable, gives a larger cost. The sequence for the slack setpoint variable in this case is $\{0.33,\ 0.68,\ 1,\ 1,\ \ldots\}$ (as shown in Fig. 2c) and causes a slower output response (Fig. 2a). Nevertheless Algorithm 4.1 uses the full range of control increments (as deduced from the input response shown in Fig. 2b) and gives stable and satisfactory responses at a computational cost which is considerably smaller than that required by either IHPC or Algorithm 4.2.

References

[1] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized predictive control, parts 1 and 2. *Automatica*, 23:137–160, 1987.

[2] B. K. Kouvaritakis, J. A. Rossiter, and A. O. T. Chang. Stable generalized predictive control: an algorithm with guaranteed stability. *Proc. IEE Part D*, 139(4):349–363, 1992.

[3] D. W. Clarke and R. Scattolini. Constrained receding horizon predictive control. *Proc. IEE Part D*, 138(4):347–354, 1991.

[4] E. Mosca and J. Zhang. Stable redesign of predictive control. *Automatica*, 28(6):1229–1223, 1992.

[5] J. A. Rossiter and B. Kouvaritakis. Constrained stable generalized predictive control. *Proc. IEE Part D*, 140(4):243–254, 1993.

[6] J. B. Rawlings and K. R. Muske. The stability of constrained receeding horizon control. *IEEE Trans. AC*, 38(10):1512–1516, 1993.

[7] J. A. Rossiter, J. R. Gossner, and B. K. Kouvaritakis. Infinite horizon stable predictive control. *IEEE Trans. AC*, 41(10):1522–1527, 1996.

[8] J. A. Rossiter, B. Kouvaritakis, and J. R. Gossner. Constrained cautious stable predictive control. *Proc. IEE part D (to appear)*, 1996.

[9] P. O. M. Scokaert and J. B. Rawlings. Infinite horizon linear quadratic control with constraints. *IFAC 13th World Congress, San Francisco*, 1996.

[10] C. E. Gilbert and K. T. Tan. Linear systems with state and control constraints: the theory and practice of maximal admissible sets. *IEEE Trans. AC*, 36(9):1008–1020, 1991.

[11] E. Fogel and Y. F. Huang. On the value of information in system identification – bounded noise case. *Automatica*, 18(2):229–238, 1982.