Feasibility and stability for constrained stable predictive control

J.A. Rossiter,
Dept. Mathematical Sciences,
Loughborough University,
Leicestershire. LE11 3TU

B. Kouvaritakis, and J.R. Gossner
Dept. Engineering Science,
Oxford University,
Oxford. OX1 3PJ

"3rd IEEE-CCA"
Keywords: *Predictive, constraints, stability, feasibility*

Abstract

Predictive control strategies which handle input/output constraints optimize output tracking over a horizon, and thus tend to drive the controls to the constraint limits; this can lead to infeasibility and/or instability. Here we develop necessary and sufficient conditions for feasibility and stability, and propose an algorithm which overcomes finite horizon infeasibility and gives stability and asymptotic tracking.

1. Introduction

The aim in predictive control is to predict, over a horizon n_y, the vector of future tracking errors and minimize its norm over a given number n_u of future control moves. This is effective but can only guarantee stability for special cases. To remedy this, Constrained Receding Horizon Predictive Control (CRHPC) [1] adopts the basic Generalized Predictive Control (GPC) [2] strategy but introduces some terminal constraints; a similar algorithm was also proposed in [3]. An alternative approach was taken by Stable Generalized Predictive Control (SGPC) [4] which forms a stabilizing loop around the system first and then applies GPC to a closed loop configuration with finite impulse responses (FIRs); the use of FIRs implies that minimization of the predicted error norm yields a monotonically decreasing cost, and this guarantees stability and asymptotic tracking.

These properties carry over to the case of predictive control with constraints so long as the implied optimization problem (a quadratic programming problem in Quadratic Programming Generalized Predictive Control, QPGPC, see Ref. [5], or a mixed weights least squares, MWLS, problem in Constrained Stable Generalized Predictive Control, CSGPC, see Ref. [6]) is feasible. This is a strong assumption: it requires "short term" feasibility (feasibility over finite horizons); short term infeasibility does not imply overall infeasibility. Thus the requirement that the output reach a target value may be sensible, in that a feasible solution exists, but this does not imply that the target can be reached within n_y steps without constraint violations. What is worse, in some cases both QPGPC and CSGPC, can convert a feasible problem into an infeasible one: in order to cause the predicted output to reach its target within n_y steps it may be necessary to drive the controls to their limits, but this is done without taking future stability into account. For unstable and/or non-minimum phase systems, future stability may require even harder future control moves which are not possible. The inevitable result of all this is instability.

In this paper we develop necessary and sufficient conditions under which this situation can be avoided; these conditions are a posteriori conditions in that they are based on past data. Violation of a posteriori conditions will cause instability. These results provide a test for when things have gone wrong, but do not provide the mechanism for avoiding instability. To do this one can advance the a posteriori conditions one step ahead in time and derive a priori conditions which can be used to limit the choice of future control moves. This work falls beyond the scope of this paper; here we propose a simple modification to CSGPC which, though not necessarily optimal in the sense that it does not make use of a priori feasibility conditions, nevertheless guarantees stability and asymptotic tracking. The idea is as follows. Under the assumption of short term feasibility CSGPC has guaranteed stability and asymptotic tracking; however no such guarantees can be given in the case of short term infeasibility. Short term infeasibility is caused by the insistence that the output should reach its target value within n_y steps, so one could relax this requirement. However the stability proof of SGPC and CSGPC depends on the property implied by the use of FIR's that the output settles after n_y steps. Therefore one must retain this last property but allow the value to which the output settles at to become a degree of freedom. Then, modifying MWLS appropriately one can place a penalty on the deviation of this value from the desired value as well as handle constraints. The net result is that the new algorithm is activated only when CSGPC becomes infeasible, and guarantees recovery of short term feasibility. This, together with the stability property of CSGPC guarantee overall stability and asymptotic tracking. The paper does not consider the effects of model mismatch and disturbances.

2. Brief review of CSGPC

2.1 The SGPC strategy

Consider a system with transfer function $G(z)=b(z)/a(z)$, with $b(z)$, $a(z)$ polynomials in z^{-1} of degree n, and the leading coefficients $b_0=0$, $a_0=1$. Form a stabilising loop such that the system input u, output y and commanding input c are related as [4]:

$$y(z)=b(z)c(z);\quad \Delta u(z)=A(z)c(z);\quad A(z)=a(z)\Delta(z) \tag{1}$$

Simulating eqn. 1 forward in time we may write:

$$\underset{\rightarrow}{y}=\Gamma_b\underset{\rightarrow}{c}+y_f;\quad \Delta\underset{\rightarrow}{u}=\Gamma_A\underset{\rightarrow}{c}+\Delta u_f;\quad y_f=H_b\underset{\leftarrow}{c}+M_bc^{\leftarrow};\quad \Delta u_f=H_A\underset{\leftarrow}{c}+M_Ac^{\leftarrow} \tag{2}$$

where y_f and Δu_f are known, and the "under-arrows" signify

$$\underset{\rightarrow}{v}=[v(t_o),v(t_o+1),\ldots,v(t_o+n_-)]^T;\ \underset{\leftarrow}{v}=[v(t_o),v(t_o-1),\ldots,v(t_o-n_+)]^T \tag{3}$$

with t_o and n_- being t+1 and n_y for y, t+1 and n_c for c, and t and n_y-1 for Δu, while t_o and n_+ are t and n for c; n_y and n_c are the output and reference horizons; Γ, H and M are as defined in [4]. The vector $c^{\leftarrow}$ is chosen to remove steady-state offsets. Eqns. 2a,b lead to the cost:

$$J=\|\underset{\rightarrow}{r}-\underset{\rightarrow}{y}\|_2^2+\lambda\|\Delta\underset{\rightarrow}{u}\|_2^2=[\underset{\rightarrow}{c}-c_o]^TS^2[\underset{\rightarrow}{c}-c_o]+\gamma;\quad S^2=\Gamma_b^T\Gamma_b+\lambda\Gamma_A^T\Gamma_A \tag{4-5}$$

$$c_o=S^{-2}[\Gamma_b^T(\underset{\rightarrow}{r}-y_f)-\lambda\Gamma_A^T\Delta u_f];\ \gamma=\|\underset{\rightarrow}{r}-y_f\|^2+\lambda\|\Delta u_f\|^2-\|Sc_o\|^2$$

The 1st value of c_o is used only; the implied law is $c_t=[p_r(z)/p_c(z)]r_{t+ny}$. $p_r(z)$, $p_c(z)$ are a pre-filter and the closed-loop pole-polynomial [4]. This strategy mirrors that of GPC with one important difference: in GPC the free variable Δu is related to y through the infinite impulse response of G(z), whereas here the free variable c is related to y (and Δu) through FIRs (eqn. 1a,b). On account of this, for $n_y>n+n_c$ the cost of 4a is monotonically decreasing thus guaranteeing stability [4].

2.2 Introducing constraints into SGPC

Often there are constraints on the values that system i/ps, o/ps can take. For brevity here we consider only i/p absolute constraints, i/p rate constraints and o/p absolute constraints of the form

$$|(u_i-U_o)/U|\le 1;\ |(u_i-u_{i-1})/R|\le 1;\ |(y_{i+1}-Y_o)/Y|\le 1 \tag{6}$$

for $i=0,1,\ldots,n_y-1$; time-varying limits can be handled in a similar fashion. Considering absolute i/p, rate i/p and o/p constraints in this order it is possible, through eqns. 2a,b, to rewrite conditions 6 as [6]:

$$\|M\underset{\rightarrow}{c}-v(t)\|_\infty\le 1 \tag{7}$$

Next define the "feasible region", $F_{nc}^{ny}[v(t)]$ as the subspace of all vectors c which, for a given n_y, satisfy (7). Infeasibility occurs when $F_{nc}^{ny}[v(t)]$ is empty for any n_c and n_y; "short term" infeasibility arises when $F_{\infty}^{ny}[v(t)]$ is empty for a particular n_c and n_y.

In the case of short term infeasibility one may wish to choose c so as to minimize the worst case constraint violation ie minimize $\|Mc-v(t)\|_{\infty}$; this can be done via Lawson's weighted least squares algorithm [7]. But, when $\min(\|Mc-v(t)\|_{\infty})<1$, then one would wish the strategy for choosing c to be dominated by the minimization of the cost of (4). This is done by the mixed weight least squares (MWLS) iteration defined by the minimization over $c^{(i+1)}$ of:

$$J_{MWLS}^{(i+1)}=\left\|\begin{bmatrix}[w^{(i+1)}]^{1/2}\epsilon^{(i+1)}\\ [W^{(i+1)}]^{1/2}e^{(i+1)}\end{bmatrix}\right\|^2; \quad \epsilon^{(i+1)}=S[c^{(i+1)}-c_o]; \quad e^{(i+1)}=Mc^{(i+1)}-v(t)$$

$$w^{(i+1)}=w^{(i)} \Big/ \sum_{k=1}^{m} W^{(i)}_{kk}|e^{(i)}_k|; \qquad W^{(i+1)}_{jj}=W^{(i)}_{jj}|e^{(i)}_j| \Big/ \sum_{k=1}^{m} W^{(i)}_{kk}|e^{(i)}_k| \tag{8-9}$$

MWLS has some desirable properties [6]: under short term feasibility it can only converge to the constrained optimum, c*; in the case of short term infeasibility it converges to the solution that minimizes the max. constraint violation. This is important because it gives a useful way of handling "short term" infeasibility. By minimizing the max. constraint violation the algorithm allows for the possibility of avoiding infeasibility altogether. Thus let the constraint violation concern some predicted future value of the i/p-o/p. If all the "optimal" future control increments computed at t were to be implemented, then a constraint violation would occur some time in the future. However, at t+1 MWLS will have another go at minimizing the worst case violation and the value for $\|Mc-v(t+1)\|_{\infty}$ will be less than or at worst equal to the value at t. Hence MWLS offers the potential for avoiding constraint violation altogether. This is in sharp contrast to other methods, such as QPGPC, which revert to the unconstrained optimum. Naturally the unconstrained minimum has to be "trimmed" so as to yield a feasible solution. This however would have the effect of increasing the infinity norm of Mc-v(t) resulting in sub-optimality and possibly instability.

A further bonus of MWLS is that the weights converge to zero if the corresponding elements of Mc*-v(t) are less than 1, and to non-zero values otherwise. This provides the means of identifying the active constraint set which can be used to computational advantage.

CSGPC like SGPC has some attendant stability results, which can be established by proving that the relevant cost forms a monotonically decreasing function of time [6]. However for CSGPC this requires a feasibility assumption as stated in the theorem below.

Theorem 2.1 Let a linear system with transfer function $G(z)=z^{-1}b(z)/a(z)$ be subject to i/p, o/p constraints which, at t and for an o/p horizon n_y and control horizon n_c, are given as $\|Mc-v(t)\|_{\infty}\leq 1$, and let F_{∞}^{ny} be the feasibility region for c. Then if $F_{\infty}^{ny}[v(t)]$ is non-empty, CSGPC will cause y to follow asymptotically any setpoint change.

Theorem 2.1 assumes feasibility, a condition that must be testsed. It is the main purpose of this paper is to derive such tests.

3. Setting up the conditions for feasibility with stability

3.1 Mathematical preliminaries

Under zero initial conditions simulate $u_t=a(z)c_t$ over N steps, to get $u=C_a c$, where C_a is the lower triangular $N\times N$ toeplitz convolution matrix formed of the coefficients of a(z). Then we have:

Lemma 3.1 Let σ_i be the singular values of C_a and let $\sigma_1\geq\sigma_2\geq\ldots\geq\sigma_N$. Then if $a(z)=1-pz^{-1}$ with $|p|>1$ and $q=1/p$:

$$(|p|-1)\leq\sigma_{N-1}\leq\sigma_{N-2}\leq\ldots\leq\sigma_1\leq(|p|+1); \quad \text{and} \quad \sigma_N<|q|^{N-1} \tag{10}$$

Proof: Condition 10a follows from an application of Gershgoreen's theorem to the matrix $C_a^T C_a$ which has a tridiagonal form with N-1 diagonal elements equal to p^2+1 and one diagonal element equal to 1, whereas all the non-zero off diagonal elements are equal to p. To complete the proof post-multiply C_a by the vector $[q^{N-1},q^{N-2},\ldots,1]^T$ to get the first standard vector multiplied by q^{N-1}. Then invoking a norm inequality we get $\sigma_N(1-q^{2N})/(1-q^2)\leq q^{N-1}$, which leads to condition 10b.

Theorem 3.1 With the definitions of Lemma 3.1 let k_i be the principal input directions of $A^T=[1,q,q^2,\ldots,q^{N-1}]$, and k_N be the direction associated with the non-zero singular value of A. Then writing any w in R^N as $w=K\alpha$, for $K=[k_1,k_2,\ldots,k_N]$, $\alpha=[\alpha_1,\alpha_2,\ldots\alpha_N]^T$, we have

$$\|C_a^{-1}w\|_2\leq\frac{1}{|p|-1}\|\alpha\|_2 \quad \text{for} \quad \alpha_N=0; \quad \|C_a^{-1}w\|_2\geq|\alpha_N||p|^{N-1} \quad \text{for} \quad \alpha_N\neq 0 \tag{11}$$

Proof: Let $[x_i,\sigma_i,y_i]$ denote the output principal directions, singular values and input principal directions of C_a respectively. Then by Lemma 3.1 it is easy to show that $A/\|A\|_2=k_N=x_N$, so that k_i can always be chosen to be $k_i=x_i$. Then given that $[y_i,1/\sigma_i,x_i]$ denotes the singular value/vector triple for C_a^{-1} it follows that

$$\|C_a^{-1}w\|^2 = \Big\|\sum_{i=1}^{N}(\alpha_i/\sigma_i)y_i\Big\|^2 = \sum_{i=1}^{N}(\alpha_i/\sigma_i)^2 \tag{12}$$

The result follows by invoking Lemma 3.1 for $\alpha_N=0$ and $\alpha_N\neq 0$.

Lemma 3.2 Let $a_i(z)=1-p_i z^{-1}$ with a corresponding convolution matrix C_{ai}, for $i=1,2,\ldots,n$ and let $a(z)=a_1(z)\times\ldots\times a_n(z)$. Then C_a, the convolution matrix for a(z) is the product of the C_{ai} taken in any order.

Proof: This is straightforward and for brevity will be omitted.

Theorem 3.2 Let the roots of a(z) be p_i, let $|p_i|>1$ for $i=1,2,\ldots m$ and let the matrix A have as its i^{th} row the vector $[1,q_i,q_i^2,\ldots,q_i^{N-1}]^T$, where $q_i=1/p_i$. Furthermore denote by k_i the input principal directions of A arranged so that the k_i for $i=N-m+1,N-m+2,\ldots,N$ correspond to the non-zero singular values of A. Then expressing any vector w in R^N as $w=K\alpha+K'\alpha'$ where $K=[k_1,k_2,\ldots,k_{N-m}]$ and $K'=[k_{N-m+1},k_{N-m+2},\ldots k_N]$, for $j=1,\ldots,m$ we have

$$\|C_a^{-1}w\|_2\leq\prod_{i=1}^{N-m}\left(\frac{1}{|p_i|-1}\right)\|\alpha\|_2 \quad \alpha'=0; \quad \|C_a^{-1}w\|_2 \geq O(p_j^{N-1}) \quad \alpha'\neq 0 \tag{13}$$

Proof: Let m=2, so that $C_a=C_{a1}C_{a2}$, let K_1, K_2 be matrix representations of the kernel of the rows of A, A_1^T, A_2^T, defined in a manner analogous to K, and let $C_{a1}^{-1}w=v$. Then $\alpha'=0$ together with the fact that the last row of C_{a1}^{-1} is $p_1^{N-1}[1,q_1,q_1^2,\ldots,q_1^{N-1}]$ imply that the last element of v is zero, so that $w=C_{a1}v$ gives a matrix representation of the convolutions implied by $w(z)=v(z)(1-p_1z^{-1})$. But, by definition $A\times w=0$, so $w(z)=(1-p_1z^{-1})(1-p_2z^{-1})\phi(z)$, where $\phi(z)$ is of order N-m-2 with stable roots, and so $v(z)=(1-p_2z^{-1})\phi(z)$; hence $A_2^Tv=0$, or $v=K_2\beta$ for some non-zero vector β. Then by Theorem 3.1 we have

$$\|C_{a2}^{-1}v\| \leq \|\beta\|/(|p_2|-1); \qquad \|v\|=\|C_{a1}^{-1}w\| \leq \|\gamma\|/(|p_1|-1) \tag{14}$$

γ is the projection of w onto the kernel of A_1^T. However $w=K\alpha=K_1\gamma$ and $v=K_2\beta$, so that $\|\gamma\|=\|\alpha\|$ and $\|\beta\|=\|v\|$ which combine with conditions 14a,b to give condition 13a for m=2; these arguments apply for a general m. Conversely, if at least one of the elements of α', say the j^{th}, is non-zero then by Theorem 3.1 $\|C_{aj}^{-1}w\|\geq O(p_j^{N-1})$.

3.2 Necessary and sufficient conditions for feasibility and stability

In the absence of constraints SGPC has guaranteed stability, because u is chosen so that the predicted y reaches its setpoint within a finite number of steps; this involves implicitly the cancellation of all the poles of a(z) [8]. When u is constrained however and we hit short term infeasibility, this property is lost, and u does not cancel the unstable poles of a(z); then not only will the predicted y not reach its steady state value, but the control system will go unstable. Below we state the necessary and sufficient conditions which avoid this situation.

Eqn. 1b gives the dependence of u on c. CSGPC ensures that this holds even when u violates constraints by ensuring that $c_t=u_t^*-a_1c_{t-1}-a_2c_{t-2}-\ldots-a_nc_{t-n}$, where u_t^* is the value nearest to u_t which satisfies the constraints. Thus c can be generated by simulating eqn. 1b to get

$$C_a\underline{c} = \underline{u}-H_a\underline{c}; \quad \text{or} \quad \underline{c} = C_a^{-1}\underline{w}; \quad \text{where} \quad \underline{w} = \underline{u}-H_a\underline{c} \tag{15}$$

Note now that due to the FIRs used in CSGPC, short term feasibility implies feasibility, namely if $\|Mc-v(t)\|_\infty \leq 1$ for some finite n_c, then this inequality will also hold true as n_c tends to infinity. Thus, determining necessary and sufficient conditions for feasibility (with stability) is equivalent to looking for the necessary and sufficient conditions under which, for $N \to \infty$, there exists a w such that the vector c of eqn. 15b is bounded and such that the elements of the vector of future u's defined by eqn.15b satisfy the input constraints.

Theorem 3.3 Let u be subject to constraints 6a and 6b, let the poles p_i of a(z) be such that $|p_j| > 1$ for $j=1,2,\ldots,m$ and let the matrix A be defined as per Theorem 3.2. Then at time t the CSGPC is feasible (and stable) if, and only if for $N \to \infty$ there exists a vector u such that

$$A\underline{u}=b \quad \text{and} \quad \|C^{*}\underline{u}-u^{o}\|_\infty \leq 1; \quad \text{where}$$

$$b=AH_a\underline{c}; \quad [C^{*}]^T=[\frac{1}{U}I_N,\ldots,\frac{1}{R}E]^{T^T}; \quad u^{oT}=[\frac{U_o}{U}I_N[1,\ldots,1]^T,\frac{u_t}{R}e^T]^T \tag{16}$$

e is the 1st column of I_N and E is I_N with -1's in positions i+1,i.

Proof: 16b is a consequence of eqn. 6 in conjunction with eqn. 16d,e.

Corollary 3.1 Let u be subject to 6a,b, let y be subject to 6c, let the roots ζ_i of b(z) be such that $|\zeta_j| > 1$ for $j=1,2,\ldots,m_b$, and let the matrix B be defined in an analogous manner to A, using ζ_j instead of p_j. The necessary and sufficient conditions for CSGPC to be feasible (and stable) at t are: (i) a vector u satisfying the conditions of Theorem 3.3 exists; (ii) for $N \to \infty$, a vector y exists satisfying the conditions:

$$B\underline{y}=b; \quad \text{and} \quad \|\frac{1}{Y}\underline{y} - \frac{y_o}{Y}[1,\ldots,1]^T\|_\infty \leq 1 \quad \text{where} \quad b=BH_b\underline{c} \tag{17}$$

Proof: This is similar to the proof of Theorem 3.3.

The results above appear to be of limited practical use because of the requirement that $N \to \infty$. However, given stability, the future values of u will settle at some constant value u_∞ after say N-1 time instants. In the light of this remark Theorem 3.3 may be restated as follows.

Theorem 3.4 Let u be subject to the constraints of Theorem 3.3. Then the necessary and sufficient condition for feasibility (and stability) is that for some $N > n$ there exists a vector u satisfying conditions 16 providing that the elements $A_{j,N}$ are replaced by $q_j^{N-1}/(1-q_j)$.

Proof: Let all the future values of u from the $(N-1)^{th}$ time instant onwards be u_∞ then

$$\sum_{i=1}^{\infty} q_j^{i-1}u_{t+i-1} = \sum_{i=1}^{N-1} q_j^{i-1}u_{t+i-1} + \frac{q_j^{N-1}}{1-q_j}u_\infty \tag{18}$$

and thus condition 16 may be restated for N finite so long as the last column of A is replaced by the vector having as its j^{th} element the ratio $q_j^{N-1}/(1-q_j)$. Conditions 16 will then be a sufficient condition for feasibility since it will guarantee the existence of a particular u which both satisfies the constraints and results in a stable c. However since N is allowed to be as large as needs be, the condition is also necessary.

Corollary 3.2 Let y be subject to the constraints of Corollary 3.1. Then the necessary and sufficient condition for feasibility is that for some $N > n$ there exists a vector y satisfying conditions 17 providing that the elements $B_{i,N}$ are replaced by $(1/\zeta_i)^{N-1}/[1-(1/\zeta_i)]$.

All conditions above involve an infinity norm constraint and thus are not easy to use: (i) to test feasibility given past data (a posteriori feasibility); and (ii) given feasibility at t to derive conditions on c_t that preserve feasibility (a priori feasibilty). A posteriori feasibility is dealt with below; a priori feasibility lies beyond the scope of this paper.

4. Necessary and sufficient conditions for a posteriori feasibility

4.1 The feasibility conditions and the algorithm for testing feasibility

The fact that $a(z)y_t=b(z)u_t$ simulated forward in time implies:

$$C_a\underline{y}+H_a\underline{y}=C_b\underline{u}+H_b\underline{\mu} \tag{19}$$

which can be used to write the conditions of Theorem 3.4 and Corollary 3.2 in the compact form below; x is given by $x^T=[u^T,y^T]$:

$$Fx=b; \qquad \|Qx-v\|_\infty \leq 1$$

$$b=\begin{bmatrix} AH_a\underline{c} \\ BH_b\underline{c} \\ H_a\underline{y}-H_b\underline{\mu} \end{bmatrix}; \quad F=\begin{bmatrix} A & 0 \\ 0 & B \\ C_b & -C_a \end{bmatrix}; \quad Q=\begin{bmatrix} C^{*} & 0 \\ 0 & \frac{1}{Y}I_N \end{bmatrix}; \quad v=\begin{bmatrix} u^{o} \\ \frac{Y_o}{Y}\begin{bmatrix} 1 \\ \cdot \\ \cdot \\ 1 \end{bmatrix} \end{bmatrix} \tag{20}$$

Thus the problem of investigating feasibility boils down to that of finding whether an N exists for which conditions (20) have a solution x. This can be tested easily as stated below.

Theorem 4.1 Let K be a matrix representation of the kernel of F, then there exists a vector x which satisfies conditions 20 if, and only if

$$\inf_{x'} \|M'x'-v'\|_\infty \leq 1; \quad \text{where} \quad M'=QK; \quad v'=v-QF^T(FF^T)^{-1}b \tag{21}$$

Proof: Eqn. 20a gives $x=Kx'+F^T(FF^T)^{-1}b$, with x' any vector of conformal dimension, which (by 20b) implies that $\|QKx'-v'\|_\infty \leq 1$; hence a necessary and sufficient condition for feasibility is that the infimal value of this norm over all x' be less than or equal to 1.

Remark 4.1 Infimization can be performed using Lawson's algorithm, thus for any N it is easy to check CSGPC feasibility at any particular t. Also N need not be taken too large because the i^{th} column of F decays to zero with increasing i; also the N^{th} element of x (for $N > n_c$) is by definition given by u_{t+N+1} which, due the input constraints, will be finite. Thus providing that N is taken to be large enough so as to make the N^{th} column of A sufficiently small (say 10^{-6}), the introduction of further degrees of freedom through an increased N will not affect (significantly) the solution of Fx=b and thus cannot affect feasibility.

Remark 4.2 Theorem 4.1 involves the past history of c, but this can be written as a linear function of the vectors of past inputs/outputs [4].

4.2 Explicit conditions for a posteriori feasibility.

Theorem 4.1 provides an algorithm for investigating the a posteriori feasibility of CSGPC. One can use this algorithm to determine a priori feasibility: advance the algorithm one step in t still using past data, in order to determine the range of values of u_{t+1} for which feasibility is preserved. However the computational burden would be prohibitive. This is so because the necessary and sufficient conditions of Theorem 4.1 are not explicit. It is the purpose of this section to show that in some cases explicit conditions are easy to derive.

Lemma 4.1 Let a(z) have only one unstable pole, let that pole be at p and let q=1/p. Then eqns. 16a,c can be rewritten as

$$A[\Delta\underline{u}]=b; \quad b=(1-q)AH_a\underline{c}-u_t; \quad \text{where} \quad \Delta\underline{u}=E\underline{u}-u_te \tag{22}$$

Proof: From eqns. 16a,c we have

$$AE^{-1}(E\underline{u}-u_te)=AH_a\underline{c}-AE^{-1}eu_t=AH_a\underline{c}-u_t/(1-q) \tag{23}$$

and by definition we also have that $AE=(1-q)A$, or $AE^{-1}=[1/(1-q)]A$. Combining this result with eqn. 23 yields the result.

Lemma 4.2 Let the unstable pole of Lemma 4.1 be positive and let CSGPC be subject to the input constraints 6a,b. Furthermore let m_U and m_L denote the largest integer such that $u_t+m_UR \leq U_o+U$ and $u_t-m_LR \geq U_o-U$. Then the max/min values that the left hand side of 22a can assume over all feasible vectors of future values of u are given as:

$$[b^{*}]_{max}=(1-q^{m_U})\frac{R}{1-q}+q^{m_U}(U_o+U-u_t-m_UR);$$
$$[b^{*}]_{min}=-(1-q^{m_L})\frac{R}{1-q}+q^{m_L}(U_o-U-u_t+m_LR) \tag{24}$$

Proof: All the elements of A are positive, hence the larger the Δu's the larger the LHS of 22a. But the Δu's are limited by R and by

$$u_{t+k}=u_t+\Delta u_{t+1}+\Delta u_{t+2}+\ldots+\Delta u_{t+k}\le U_o+U \tag{25}$$

But the elements of A decay geometrically hence the value of the LHS of 22a is maximized when Δu_{t+i} causes u_{t+k} to reach its max value in minimum time (m_U+1 steps). Thus the maximizing vector of Δu's is

$$\Delta u=[R,R,\ldots,R,U_o+U-(u_t+m_U R),0,0,\ldots]^T \tag{26}$$

where the number of repeated values R is m_U. Substitution of this into 22a yields 24a. Eqn. 24b can be got using similar arguments; the only difference is that the Δu's must be such that u reaches is lowest permissible value of U_o-U in minimum time, namely in m_L+1 time steps. The corresponding vector of control increments will have the form

$$\Delta u=[-R,-R,\ldots,-R,U_o-U-(u_t-m_L R),0,0,\ldots]^T \tag{27}$$

Lemma 4.3 Let the unstable pole of Lemma 4.1 be negative and let CSGPC be subject to i/p constraints 6a,b. Then max/min values that the LHS of 22a assumes over all feasible vectors of future u's are:

$$\begin{aligned}
&[b^-]_{max}=\alpha-u_t+\frac{q(\beta-\alpha)}{1-|q|};\quad \alpha=\min[u_t+R,U_o+U];\quad \beta=\max[\alpha-R,U_o-U]\\
&[b^-]_{min}=\alpha-u_t+\frac{q(\beta-\alpha)}{1-|q|};\quad \alpha=\max[u_t-R,U_o-U];\quad \beta=\min[\alpha+R,U_o+U]
\end{aligned} \tag{28}$$

Proof: This is similar to the proof above, except that the signs of the elements of A alternate, so the elements maximizing/minimizing vectors must alternate between their max and min allowable values.

Theorem 4.4 Let a(z) have only one unstable pole, say at p, and let CSGPC be subject to i/p constraints 6a,b. Then the necessary and sufficient feasibility conditions at t in terms of the b of eqn. 24b are:

$$b_{min}\le b\le b_{max}\qquad \text{where} \tag{29}$$

$$\text{for } p>0\ \ b_{min}=[b^+]_{min};b_{max}=[b^+]_{max};\ \text{for } p<0\ \ b_{min}=[b^-]_{min};b_{max}=[b^-]_{max}$$

Proof: For each of the two cases considered, b_{min}, b_{max} give the max/min values that the LHS of 22a can assume over all feasible future control moves. Thus if b lies in the interval $[b_{min},b_{max}]$ there will exist at least one solution (or a whole family of solutions if 29a holds with strict inequality on either side) which is both feasible and does not violate the stability condition of Theorem 3.3. Conversely, by Lemmata 4.2 and 4.3 there will not exist feasible solutions which satisfy eqn. 22a if b lies outside the interval $[b_{min},b_{max}]$.

The same idea carries over to the case where a(z) has two unstable real poles, for which the matrix A of eqn. 22 will have two row vectors, A_1, A_2, and the right hand side of (22) will be a two dimensional vector, say $b=[b_1,b_2]^T$. However applying Theorem 4.4 to $A_1\Delta u=b_1$ and $A_2\Delta u=b_2$ independently will only generate necessary conditions; below we state conditions which are both necessary and sufficient.

Theorem 4.5 Let a(z) have 2 unstable real poles at p_1, p_2. Then the matrix A of 22a will have 2 rows, A_1, A_2, for $q_1=1/p_1$ and $q_2=1/p_2$. Let $[b_1]_{max}$ and $[b_1]_{min}$ be as per Theorem 4.4, and let Δu_1 be the vector of future control increments for which $A_1\Delta u_1$ attains its max value of $[b_1]_{max}$. Then the necessary and sufficient feasibility conditions are

$$[b_1]_{min}\le b_1\le[b_1]_{max};\quad A_2\Delta u_1-\max_x[A_2x]\le b_2\le A_2\Delta u_1-\min_x[A_2x] \tag{30}$$

where the vector x is constrained to satisfy the condition

$$A_1x=[b_1]_{max}-b_1 \tag{31}$$

Proof: By Theorem 4.4, condition 30a guarantees that there exist vectors Δu which satisfy the equation $A_1\Delta u=b_1$. The totality of such vectors can be written as $\Delta u=\Delta u_1$-x, where x must satisfy eqn. 31. Such vectors Δu will make $A_2\Delta u$ equal to $A_2\Delta u_1$-A_2x, and so this right hand side will lie in the interval defined by condition 30b. Clearly then the equation $A\Delta u=b$ will admit a feasible solution if and only if both conditions 30a and 30b are satisfied.

In order to invoke Theorem 4.5, one needs to determine the maximizing/minimizing vectors x, but this is straightforward when p_1, and p_2 are real and share the same sign. For example let $p_1, p_2>0$, and $q_1>q_2$, and let u_1, u_2 be the vectors of future u's that correspond to Δu_1 and Δu_2. Then it can be shown that the x that maximizes A_2x is the vector causing the vector of future u's corresponding to $\Delta u=\Delta u_1$-x, to follow u_2 for as long as possible (as dictated by eqn. 31) and then at the maximum possible rate reach the maximum limit on u, namely U_o+U. The form of such a vector x will be

$$\begin{aligned}
x=[&2R,\ldots,2R,2R-\beta,0,\ldots,0,-(R-U_o-U+u_t+m_U R),-R,\ldots,-R,\\
&-(U_o+U-u_t-m_U R-\beta),0,0,\ldots]^T
\end{aligned} \tag{32}$$

where there are respectively μ_1 and $2\mu_1+1$ terms before and after the first string of zeros; this string of zeros is m_U-μ_1 long. The integer μ_1 is less than or equal to m_U and the rounded down integer value that μ_1 would assume if it were allowed to be the real number for which A_1x would become equal to $[b_1]_{max}$-b_1 for $\beta=0$; this value can be obtained from the solution of a quadratic equation. The value of β is then obtained by substituting eqn. 32 into eqn. 31. The detailed proof for the structure of x is simple but long and will be omitted. Here we simply make two obvious remarks: (a) since the starting point, u_t, is common to both u_1 and u, and since the ultimate value of U_o+U is also common to both, the sum of the elements of the vector x must be zero; (b) because the elements of A_2 decay as a geometric series, the vector which maximizes A_2x must have as large a front end as eqn. 31 and the absolute and rate constraints will permit. A similar procedure can be used to determine the minimizing x and complete the proof.

The case of two real unstable poles of different sign, or the case of two complex conjugate unstable poles is considerably more complicated and will not be given here. In cases like this as well as for the general case of any number of unstable poles one must revert to Theorem 4.1 for a necessary and sufficient test of feasibility.

4.3 Illustrative examples

Example 4.1 Let the system with $a(z)=1-2.2z^{-1}+0.09z^{-2}+0.252z^{-3}$, $b(z)=2z^{-1}+0.45z^{-2}+z^{-3}$ be subject to the i/p constraints with $U_o=0$, $U=25$, $R=0.04$; a(z) has only one unstable pole, so feasibility can be tested with the explicit conditions of Theorem 4.4. Assume the system is at rest, r=0 for $t\le22$ and r=1 for $t>23$. Clearly $u_t=0$ for $t\le23$ and at t=23 $m_U=m_L=U/R=625$; $m_U=m_L$ because the absolute limits are symmetric about $u_{23}=0$. Since R is small, m_U and m_L will be very large for all values of t from 23 to 30 and so q^{m_U} and q^{m_L} will be insignificantly small, (because q=1/2.1). Then from eqn. 36 we have:

$$b_{max}=[b^+]_{max}=\frac{R}{1-q}=0.0764;\ \text{and}\ b_{min}=[^+]_{min}=-\frac{R}{1-q}=-0.0764 \tag{33}$$

The necessary and sufficient condition for feasibility therefore is that for t=23,24,...,30 the b of eqn. 22b must lie between -0.0764 and 0.0764. Upon application of CSGPC the optimal value of Δu_{23}, as seen from Fig. 1b is 0.04, but for this the corresponding value of b, as shown in Fig. 1c, is -0.084. Thus the first control move recommended by CSGPC results in infeasibility, so that at t=24 there will not exist a Δu for which conditions 6b and 30a,b can be satisfied simultaneously. However the i/p constraints are hard so eqn. 6b will be satisfied; as a consequence condition 30a will be violated and the feedback system will go unstable. This is illustrated in Fig. 1a which shows the response of the output. Infeasibility means lack of feasibility over an infinite horizon which also implies short term infeasibility; as a result the infimal value of $\|Mc-v(t)\|_\infty$ will be greater than 1. Furthermore, due to instability this infimal value diverges (Fig. 1d). If the unstable pole were at 2 instead of 2.1 the feasibility interval would become [-0.084,0.084] and so CSGPC would operate at the limit of stability as shown in Fig. 2.

Example 4.2 In the example above, short term infeasibility caused CSGPC to choose control moves which render the problem infeasible. This need not always be the case, especially if the short term infeasibility concerns future constraint violations. Through the use of the MWLS cost, CSGPC will choose c so as to minimize the worst case constraint violations, and if these violations are in the future then CSGPC will be able to reduce this violation further at the next step. By the time we come to implement the offending value of c, infeasibility may have disappeared altogether.

The system to be considered has a transfer function with $a(z)=1-1.3z^{-1}+0.144z^{-3}$, $b(z)=2z^{-1}+0.45z^{-2}+z^{-3}$ and has only one unstable pole (at 1.2). The system input is subject to constraints defined by $U_o=0$, $U=0.05$ and $R=0.2$. The values of b_{min} and b_{max} are calculated as per Remark 4.3 to be -0.3 and 0.3 respectively, and the corresponding value of b is plotted in Fig. 3b; clearly CSGPC satisfies the necessary and sufficient feasibility condition, despite the fact that it runs up against short term infeasibility as demonstrated by Fig. 3d. The implied constraint violations however are in the future and CSGPC manages to recover short term feasibility a few time steps later; as a result the algorithm gives a stable and satisfactory output response (Fig. 3a) without violating any constraints (Fig. 3c).

5. Modified CSGPC algorithm with guaranteed convergence

Theorem 2.1 states that if at some t CSGPC is feasible for a finite n_y, then the algorithm will lead to asymptotic stability. However nothing can be said about the stability of CSGPC in the case of "short term infeasibility". As was seen in Example 4.3, by minimizing the worst case future constraint violation, CSGPC can overcome short term infeasibility and give stability and asymptotic tracking. This property is extremely useful but cannot be guaranteed in general. Here we propose a modification to CSGPC which always leads to stability.

Short term infeasibility arises due to the implicit requirement that y should reach its target value in n_y steps. This is done through the choice of far future values of c, c^{∞} as $c^{\infty}=c_{\infty}[1,1,\ldots,1]^T$ with $c_{\infty}=r/b(1)$; for convenience we assume that r(t) is a step of size r. Thus the only way to overcome short term infeasibility is to relax this requirement. A simple way to achieve this, while preserving the stability attributes of CSGPC, is to include c_{∞} as a degree of freedom; clearly one needs to penalize the deviation of c_{∞} from its desired value of r/b(1). Accordingly we substitute c^{∞} into eqn. 2 to get

$$\begin{gathered}\underline{y}=[\Gamma_b,m_b]\underline{c}^*+y_f^*; \quad \Delta\underline{u}=[\Gamma_A,m_A]\underline{c}^*+\Delta u_f^*; \\ [m_A,m_b]=[M_A,M_b][1,\ldots,1]^T;\ c^*=[\underline{c}^T,\ c_{\infty}^T]^T;\ y_f^*=H_b\underline{c};\ \Delta u_f^*=H_A\underline{c}\end{gathered} \tag{34}$$

Now by its definition (Ref. 6) v(t) of (7) is a linear function of c_{∞} and v(t) and the corresponding constraints inequality can be written as

$$v(t)=v^*(t)+mc_{\infty};\quad \|M^*\underline{c}^*-v^*(t)\|_{\infty}\leq 1;\quad \text{where}\quad M^*=[M,m] \tag{35}$$

On the basis of (34) and (35) the CSGPC problem can be restated with respect to the augmented vector c* of future c's, and this modified procedure can be deployed at all time instants when the original CSGPC algorithm runs into short term infeasibility.

Algorithm 5.1 <u>Step 1:</u> Test feasibility (Th 4.). If due to a setpoint change the problem is infeasible issue warning and proceed to Step 2
<u>Step 2:</u> Apply CSGPC (Section 2.2). If $\|Mc-v(t)\|_{\infty}\leq 1$ increment t by one and return to Step 1; otherwise proceed to Step 3.
<u>Step 3:</u> Use MWLS (Section 2.2) with the following modifications

$$\epsilon^{(i+1)} \rightarrow c^{*(i+1)}-r/b(1);\quad e^{(i+1)} \rightarrow M^*c^{*(i+1)}-v^*(t)$$

let MWLS converge the optimal vector of future c* and implement the implied first future value of u. Increment t, go to Step 1.

Theorem 5.1 In the absence of hard output constraints and under the assumption that the setpoint changes do not cause the control problem to become infeasible, Algorithm 5.1 has guaranteed stability and will cause the output y to reach asymptotically its target value.

Proof: By the feasibility assumption of the theorem, Step 1 of the Algorithm 5.1 will always lead to Step 2. Furthermore if the problem is short term feasible for all t, Step 3 will never be entered, so that Algorithm 5.1 will operate exactly as the CSGPC algorithm of Section 2.2, and so by Theorem 2.1 we have stability and asymptotic tracking.

Now let us assume that at some t, CSGPC is short term infeasible and Algorithm 5.1 enters Step 3. Clearly the implied optimization problem is always feasible because c_{∞} is now a degree of freedom and can be chosen so as to require as little movement in the u's, and Δu's as is necessary; hence MWLS will converge to the solution which minimizes the distance of c_{∞} from r/b(1), and satisfies inequality 35. The implied cost of Step 3 has exactly the same form as that of eqn. 8, and so like the original cost it will be monotonically decreasing; the arguments which prove this assertion are identical to those used for the original CSGPC algorithm [6]. Hence Step 3 will cause c_{∞} to assume its target value of r/b(1) and will do so in the minimum number of time instants. Now at all times we have:

$$\|M^*c^*-v^*(t)\|_{\infty}=\|Mc-v(t)\|_{\infty} \tag{35}$$

and so one time instant before c_{∞} is made equal to r/b(1) the CSGPC problem will become short term feasible. This is so because, if Step 3 were applied one more time it would give a vector c* whose last element would be r/b(1) and for which the quantity of eqn. 36 would be less than or equal to 1; hence at that time instant it would be known that for $c_{\infty}=r/b(1)$ there exists a vector c which satisfied inequality 7.

In conclusion Step 3 will always recover short term feasibility and hence by Theorem 2.1, the overall algorithm will be stable and will cause y to track its target asymptotically.

Example 5.1 In Example 4.1 we saw that at t=23 the CSGPC algorithm of Section 2.2 leads to infeasibility and therefore short term infeasibility. Repeated application of the algorithm, as expected leads to instability. This problem is overcome entirely by the application of Algorithm 5.1 which at t=23 invokes Step 3 and therefore results in a Δu_{23} which is less than the maximum allowable size of 0.04 (see Fig. 4b). This reduction of Δu_{23} in turn results in a b of smaller absolute value (see Fig. 4c) which lies within the interval of -0.0764 and 0.0764. From Fig. 4b it can be seen that all the u's used by the algorithm stay small and hence m_U and m_L will be large and b_{min}/b_{max} will stay at -0.0764/0.0764 throughout the application of the algorithm. Then from Fig. 4c it is seen that the algorithm maintains feasibility throughout, and recovers short term feasibility at t=25 (see Fig. 4d). The o/p response (Fig. 4a) and can be seen to be good. Fig. 5 shows the results for the unstable pole of the example at 2 instead of 2.1.

For completeness we include here the simulation results for CSGPC (Fig. 6) and Algorithm 5.1 (Fig. 7) obtained from a industrial application based on the model of a dynamometer with $b(z)=0.0466z^{-1}+0.44992z^{-2}+0.3921z^{-3}+0.031z^{-4}$, $a(z)=1-3.4122z^{-1}+4.355z^{-2}-2.43z^{-3}+0.5066z^{-4}$.

6. References

[1] D.W. Clarke, and R. Scattolini. Constrained receding horizon predictive control, *Proc. IEE, Pt. D*, 1992, Vol. 138, No.4,pp.347-354

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

[3] E. Mosca, and J. Zhang. Stable redesign of predictive control, *Automatica*, 1992, Vol. 28, No. 6, pp. 1229-1233

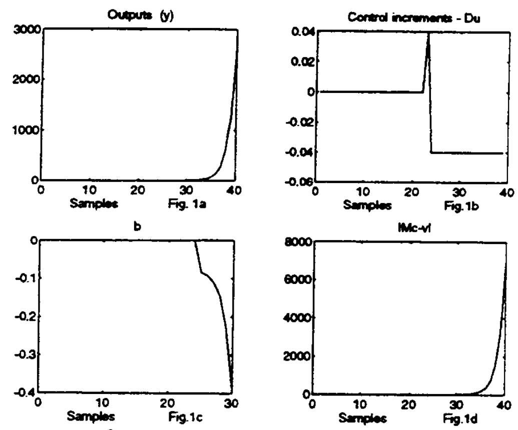
[4] B. Kouvaritakis, J.A. Rossiter, and A.O.T. Chang. Stable generalized predictive control, Proc. IEE, Pt.D, 1992, 13 pp. 349-362

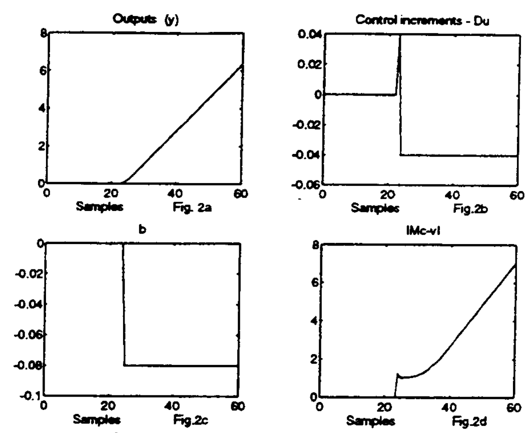
[5] T.T.C. Tsang, and D.W. Clarke. Generalized predictive control with constraints, *Proc. IEE , Pt. D*, 1988, 135, Vol. 6, pp. 451-460

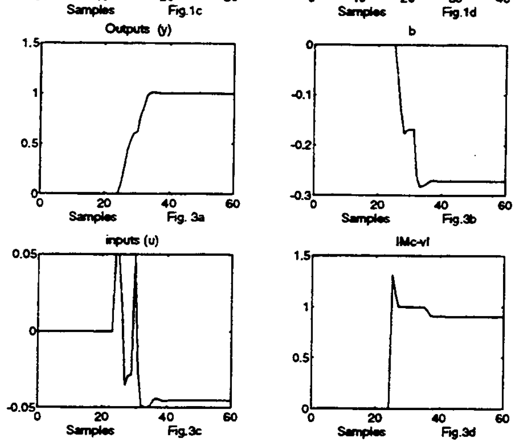
[6] J.A. Rossiter, and B. Kouvaritakis. Constrained stable generalized predictive control, *Proc. IEE, Pt. D*, 1993, 140, Vol. 4, pp. 243-254

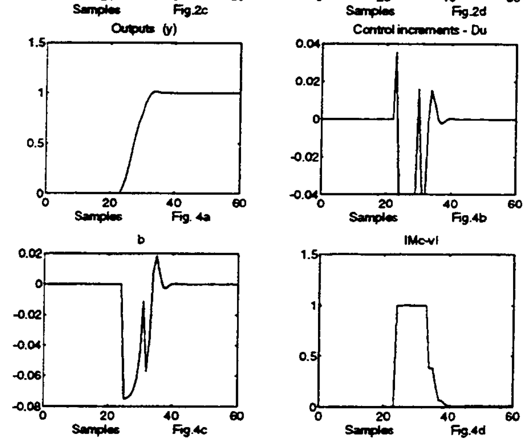
[7] C.L. Lawson, and R.J. Hanson. Solving least squares problems, Prentice Hall, 1974

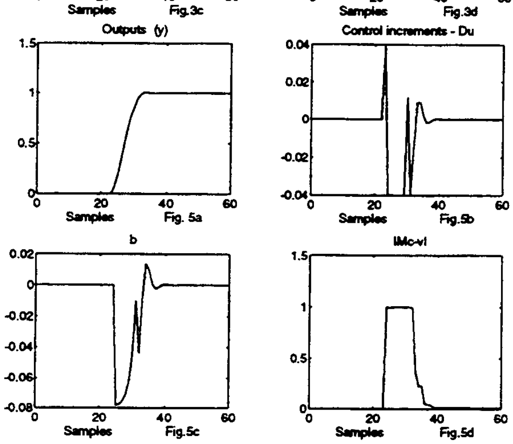
[8] B. Kouvaritakis, and J.A. Rossiter. Robustness and efficiency of GPC algorithms with guaranteed stability, to appear *Proc. IEE, Pt. D.*

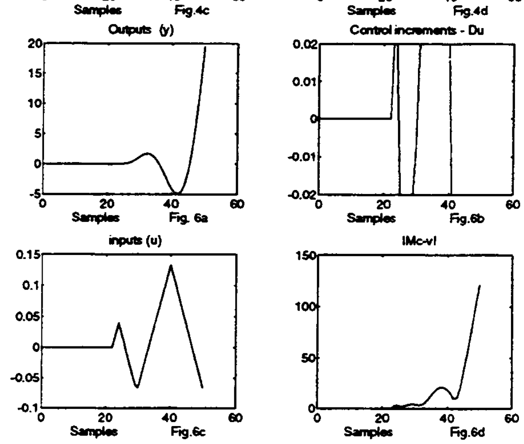

Outputs (y)
Samples
Fig. 1a
Control increments - Du
Samples
Fig.1b
b
Samples
Fig.1c
IMc-vl
Samples
Fig.1d

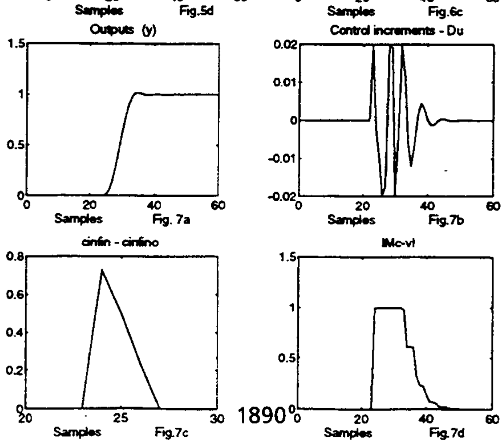

Outputs (y)
Samples
Fig. 2a
Control increments - Du
Samples
Fig.2b
b
Samples
Fig.2c
IMc-vl
Samples
Fig.2d


Outputs (y)
Samples
Fig. 3a
b
Samples
Fig.3b
inputs (u)
Samples
Fig.3c
IMc-vl
Samples
Fig.3d


Outputs (y)
Samples
Fig. 4a
Control increments - Du
Samples
Fig.4b
b
Samples
Fig.4c
IMc-vl
Samples
Fig.4d


Outputs (y)
Samples
Fig. 5a
Control increments - Du
Samples
Fig.5b
b
Samples
Fig.5c
IMc-vl
Samples
Fig.5d


Outputs (y)
Samples
Fig. 6a
Control increments - Du
Samples
Fig.6b
inputs (u)
Samples
Fig.6c
IMc-vl
Samples
Fig.6d


Outputs (y)
Samples
Fig. 7a
Control increments - Du
Samples
Fig.7b
cinfin - cinfino
Samples
Fig.7c
IMc-vl
Samples
Fig.7d