Multiplexed Model Predictive Control ⋆

Keck Voon Ling [a,∗] Jan Maciejowski [b] Bing Fang Wu [a]

[a] *School of Electrical and Electronics Engineering*
Nanyang Technological University
Singapore, 639798

[b] *Department of Engineering, University of Cambridge, United Kingdom*

Abstract

Most academic control schemes for MIMO systems assume all the control variables are updated simultaneously. MPC outperforms other control strategies through its ability to deal with constraints. This requires on-line optimization, hence computational complexity can become an issue when applying MPC to complex systems with fast response times. The multiplexed MPC scheme described in this paper solves the MPC problem for each subsystem sequentially, and updates subsystem controls as soon as the solution is available, thus distributing the control moves over a complete update cycle. The resulting computational speed-up allows faster response to disturbances, which may result in improved performance, despite finding sub-optimal solutions to the original problem. The multiplexed MPC scheme is also closer to industrial practice in many cases. This paper presents initial stability results for multiplexed MPC.

Key words: predictive control, decentralised control, multivariable, control, periodic systems, constrained optimization.

⋆ This work is supported by A*STAR project "Model Predictive Control on a Chip" (Ref: 022-106-0044). An earlier version of the paper was presented at the IFAC World Congress 2005. KV Ling would also like to acknowledge the support of Tan Chin Tuan Fellowship and of Pembroke College, Cambridge.

∗ Corresponding author
Email address: ekvling@ntu.edu.sg (Keck Voon Ling).

1 Introduction

1.1 The basic idea

Model Predictive Control (MPC) has become an established control technology in the petrochemical industry, and its use is currently being pioneered in an increasingly wide range of process industries. It is also being proposed for a range of higher bandwidth applications, such as ships (Perez *et al.*, 2000), aerospace (Murray *et al.*, 2003) (Richards and How., 2003), and road vehicles (Morari *et al.*, 2003).

This paper is concerned with facilitating applications of MPC in which computational complexity, in particular computation time, is likely to be an issue. One can foresee that applications to embedded systems, with the MPC algorithm implemented in a chip or an FPGA(Bleris *et al.*, 2006) (Johanson *et al.*, 2006) (Ling *et al.*, 2006), are likely to run up against this problem.

MPC operates by solving an optimisation problem on-line, in real time, in order to decide how to update the control inputs (manipulated variables) at the next update instant. All MPC theory to date, and as far as we know all implementations, assumes that all the control inputs are updated at the same instant (Maciejowski, 2002). Suppose that a given MPC control problem can be solved in not less than T seconds, so that the smallest possible update interval is T. The computational complexity of typical MPC problems, including time requirements, tends to vary as $O(m^3)$, where m is the number of control inputs. We propose to use MPC to update only one control variable at a time, but to exploit the reduced complexity to update successive inputs at intervals smaller than T, typically T/m. After m updates a fresh cycle of updates begins, so that each whole cycle of updates repeats with cycle time T. We call this scheme *multiplexed MPC*, or MMPC. We assume that fresh measurements of the plant state are available at these reduced update intervals T/m. The main motivation for this scheme is the belief that in many cases the approximation involved in updating only one input at a time will be outweighed — as regards performance benefits — by the more rapid response to disturbances, which this scheme makes possible. It is often the case that "do something sooner" leads to better control than "do the optimal thing later". Figure 1 shows the pattern of input moves in the multiplexed MPC scheme with $m = 3$, compared with the conventional scheme in which the three input moves are synchronised. (We will refer to conventional MPC as *synchronised MPC*, or SMPC in the rest of this report.)

A specific example of an application in which our scheme could be effective is the roll stabilisation of ships. Typically rotatable fins are installed for the

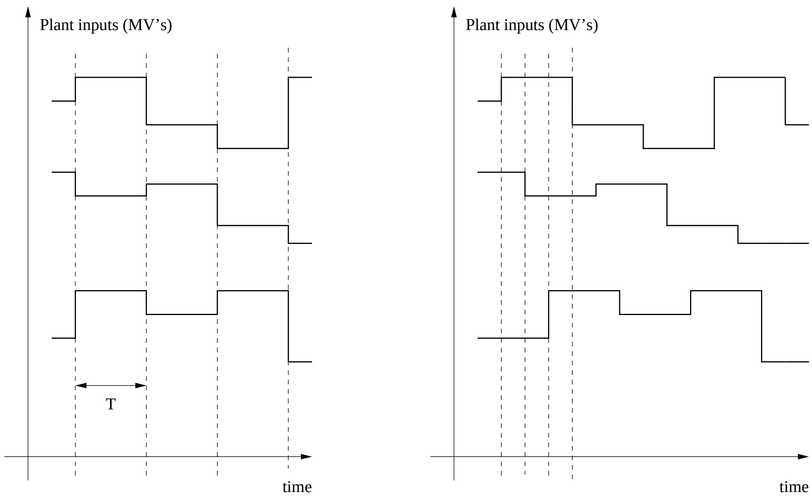


Fig. 1. Patterns of input moves for conventional 'synchronised' MPC (left), and for the Multiplexed MPC (right) introduced in this paper.

specific purpose of opposing the rolling torque that arises from wave actions. But the ship's rudder can also produce a rolling torque, and so is available as a second actuator for roll stabilisation (Perez *et al.*, 2000). This is useful for combatting the effects of unusually large waves, or for reducing the mean-square roll angle in heavy seas, when the fins may be saturated in rate or angle, and thus leads to an interest in designing a roll controller which uses both actuators. If the interval between actuator actions is T, then the ship's roll is essentially uncontrolled during intervals of duration T. Now if the roll angle is considered to be a band-limited stochastic process, such that its spectrum $S_{\phi\phi}(\omega) = 0$ for $\omega > \Omega$, then the mean-square variation of the roll angle between control actions is limited by T^2 (Papoulis, 1984):

$$E\{[\phi(t+T) - \phi(t)]^2\} \leq \Omega T^2 E\{\phi(t)^2\} \tag{1}$$

Thus reducing T offers the possibility of reducing the controlled mean-square roll angle. In order to reduce T the possibility of controlling the fins and the rudder by means of our multiplexed MPC scheme could be considered.

The scheme which we investigate here is close to common industrial practice in complex plants, where it is often impossible to update all the control inputs simultaneously, because of their sheer number, and the limitations of the communications channels between the controller and the actuators.

Various generalizations of our scheme are possible. For example, subsets of control inputs might be updated simultaneously, perhaps all the inputs in each subset being associated with one subunit. Alternatively, sensor outputs might become available one at a time (or one subset at a time), as in the common practice of "polling" sensors. A further generalization, albeit involving a significantly harder problem, would be not to update each control input in a fixed sequence, but to decide in real time which input (if any) needs updating most urgently — one could call this *just-in-time MPC*; note that this would then resemble *statistical process control (SPC)*, which is used widely in manufacturing processes (Box and Luceno, 1997).

1.2 Related Work

Several works have been published which propose 'decentralized MPC' in the sense that subsets of control inputs are updated by means of an MPC algorithm. But these usually assume that several sets of such computations are performed in parallel, on the basis of local measurements only, and that all the control inputs are then updated simultaneously. In some applications, such as formation flying of unmanned vehicles, it is assumed that the state vectors of subunits (vehicles) are distinct, and that coupling between subunits occurs only through constraints and performance measures. In (Venkat *et al.*, 2004) five different MPC-based schemes are proposed, of which four are decentralized MPC schemes of some kind. Their schemes 4 and 5 are the closest to our multiplexed scheme. In these schemes an MPC solution is solved iteratively for each control input, but it is assumed that no new sensor information arrives during the iteration, and that all the control inputs are updated simultaneously when the iterations have been completed.

As far as we are aware, the original feature of the scheme proposed in this paper is that the inputs are updated sequentially, and that each control update takes account of all the information available at that time, namely knowledge of all updates already performed, and of the latest sensor outputs. The distinction between previous proposals for decentralised MPC and our proposal for *multiplexed MPC* is analogous to the distinction between the Jacobi and the Gauss-Seidel iterative algorithms for solving a system of linear equations (Barrett *et al.*, 1994). The Jacobi algorithm updates every variable using values only from the previous iteration. The Gauss-Seidel algorithm, on the other hand, immediately uses new values of those variables that have already been updated within the current iteration. Multiplexed MPC shares this idea with the Gauss-Seidel method; the move for the current actuator takes into account moves already made by other actuators in the same cycle of iterations.

An alternative strategy for speeding up the computations involved in MPC is off-line precomputation of the 'pieces' of the piecewise-affine controller which is the optimal solution (Morari *et al.*, 2003). But that is not feasible if the number of 'pieces' required is excessively large, or if the constraints or the plant model change relatively frequently.

The rest of this paper is organized as follows.

In Section 2, two possible schemes for Multiplexed MPC are formulated. Section 3 establishes the stability of these and other schemes. Section 4 gives numerical simulation examples which compares the performance of MMPC with SMPC. Finally, concluding remarks are given in Section 5. For completeness, we include, in the appendix, derivation of the equivalent LQ problem for SMPC.

2 Problem formulation

2.1 Preliminary

We consider the following discrete-time linear plant model in state-space form, with state vector $x_k \in \mathbb{R}^n$ and m (scalar) inputs $u_{1,k}, \ldots, u_{m,k}$:

$$x_{k+1} = Ax_k + \sum_{j=1}^{m} B_j \Delta u_{j,k} \tag{2}$$

where each B_j is a column vector and $\Delta u_{j,k} = u_{j,k} - u_{j,k-1}$. (This could be generalised to the case where $B_j \in \mathbb{R}^{n \times p_j}$ and $\Delta u_{j,k} \in \mathbb{R}^{p_j}$, with $\sum_j p_j$ inputs.) We assume that $(A, [B_1, \ldots, B_m])$ is stabilizable. For ease of notation, when we drop the index j, we mean the complete B matrix and the input vector so that the system (2) may be written as

$$x_{k+1} = Ax_k + B\Delta u_k$$

The unique advantage of MPC, compared with other control strategies, is its capacity to take account of constraints in a systematic manner. As usual in MPC, we will suppose that constraints may exist on the input amplitudes, $\|u_k\|_\infty \leq U$, on the input moves, $\|\Delta u_k\|_\infty \leq D$, and on states, $Mx_k \leq v$. (These can all be generalised substantially, so long as linear inequalities are retained.)

We wish to devise a control strategy based on MPC which, at discrete-time index k, changes only plant input $(k \bmod m) + 1$. In this paper, we consider two alternative schemes for determining the appropriate plant inputs. In both schemes, an increase of k by 1 corresponds to a time duration of T/m, where T is the complete update cycle duration — see section 1.

We assume in both schemes that at time step k the complete state vector x_k is known exactly from measurements. We will consider only the regulation problem in detail, but tracking problems, especially those with non-zero constant references, can be easily transformed into equivalent regulation problems.

As we will be referring to the expression $(k \bmod m) + 1$ often in this paper, it is convenient to introduce the indexing function

$$\sigma(k) = (k \bmod m) + 1 \tag{3}$$

The constraint

$$\Delta u_{j,k+i} = 0 \text{ if } j \neq \sigma(k+i) \tag{4}$$

then expresses our desired control updating pattern as shown in Fig. 1.1.

An alternative representation of the system described by (2), together with the desired control updating pattern (4) is as a periodic linear system with one input:

$$x_{k+1} = Ax_k + B_{\sigma(k)}\Delta\tilde{u}_k \tag{5}$$

where $\Delta\tilde{u}_k = \Delta u_{\sigma(k),k}$.

From this point onwards, we use this description of the plant and we exploit known results on periodic systems.

The N-step prediction model for the system described by eq(5) is

$$\vec{X}_{k+1} = \Phi x_k + G_{\sigma(k)}\Delta\vec{U}_k$$

where

$$\vec{X}_{k+1} = \begin{bmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{k+N} \end{bmatrix}, \qquad \Delta\vec{U}_k = \begin{bmatrix} \Delta\tilde{u}_k \\ \Delta\tilde{u}_{k+1} \\ \vdots \\ \Delta\tilde{u}_{k+N-1} \end{bmatrix}$$

$$\Phi = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A_N \end{bmatrix}, \qquad G_{\sigma(k)} = \begin{bmatrix} B_{\sigma(k)} & 0 & \dots & 0 \\ AB_{\sigma(k)} & B_{\sigma(k+1)} & \dots & 0 \\ \vdots & & \ddots & \\ A^{N-1}B_{\sigma(k)} & \dots & AB_{\sigma(k+N-2)} & B_{\sigma(k+N-1)} \end{bmatrix} \tag{6}$$

Let $N = (N_u - 1)m + 1$ where N_u is the control horizon, a design parameter which will later be used to denote the number of control moves to be optimized per input channel of the original system (2). Then we can re-write the prediction model by grouping the control signals into m vectors as follows:

$$\vec{X}_{k+1} = \Phi x_k + g_1^{\sigma(k)}\Delta\vec{u}_{k,0} + g_2^{\sigma(k)}\Delta\vec{u}_{k,1} + \ldots + g_m^{\sigma(k)}\Delta\vec{u}_{k,m-1}$$

where

$$\Delta\vec{u}_{k,0} = \begin{bmatrix} \Delta\tilde{u}_k \\ \Delta\tilde{u}_{k+m} \\ \vdots \\ \Delta\tilde{u}_{k+(Nu-1)m} \end{bmatrix}$$

and for $i = 1, 2, \ldots, m-1$,

$$\Delta\vec{u}_{k,i} = \begin{bmatrix} \Delta\tilde{u}_{k+i} \\ \Delta\tilde{u}_{k+m+i} \\ \vdots \\ \Delta\tilde{u}_{k+(Nu-2)m+i} \end{bmatrix}$$

and $g_i^{\sigma(k)},\ i = 1, \ldots, m$ is the i'th column of $G_{\sigma(k)}$.

In this paper, we consider two MMPC schemes. In scheme 1, all the $\Delta\vec{u}_{k,i}, i = 0, \ldots, m-1$ are decision variables at time step k. In scheme 2, the future trajectory of only one input of system (2) is optimised, and we make assumptions about the future behaviour of the other inputs. More precisely, in scheme 2, only $\Delta\vec{u}_{k,0}$ is taken as the decision variable at time k, and appropriate assumptions are made about $\Delta\vec{u}_{k,i},\ i = 1, \ldots, m-1$.

Scheme 2 has the advantage of reducing computational complexity — the other inputs are treated like measured disturbances, in effect. Note that the length of $\Delta\vec{u}_{k,0}$ is N_u while the length of $\Delta\vec{u}_{k,i}$ for $i = 1, \ldots, m-1$ is $N_u - 1$. When $N_u = 1$, $\Delta\vec{u}_{\sigma(k),i},\ i = 1, \ldots, m-1$, become zero length vectors and MMPC scheme 1 and scheme 2 are equivalent.

In scheme 1, although only one input is updated at any one time, the computational complexity is the same as that of standard centralised MPC, since the number of decision variables is the same. Thus scheme 1 is not likely to be of much practical interest. We present it here for clarity and completeness.

2.2 Scheme 1

Scheme 1 involves solving the following finite-time constrained linear periodic control problem:

$$
\begin{aligned}
\mathcal{P}(x_k) : \text{Minimise} \quad & J_k = F(x_{k+N}) + \textstyle\sum_{i=0}^{N-1} \left(\|x_{k+i+1}\|_q^2 + \|\Delta\tilde{u}_{k+i}\|_r^2 \right) \\
\text{wrt} \quad & \Delta\vec{u}_{k,i}, \quad (i = 0, 1, \ldots, m-1) \\
\text{s.t.} \quad & \Delta\tilde{u}_{k+i} \in \mathbb{U}, \quad (i = 0, 1, \ldots, N-1) \\
& x_{k+i} \in \mathbb{X}, \quad (i = 1, \ldots, N) \\
& x_{k+N+1} \in \mathcal{X}_I(K_{\sigma(k)}) \\
& x_{k+1} = Ax_k + B_{\sigma(k)}\Delta\tilde{u}_k
\end{aligned}
\tag{7}
$$

where $F(x_{k+N})$ is a suitably chosen terminal cost, and $\mathbb{X}$ and $\mathbb{U}$ are compact polyhedral sets containing the origin in their interior. $\mathcal{X}_I(K_{\sigma(k)})$ denotes the set in which none of the constraints is active, and which is the maximum positively invariant set (Blanchini, 1999) for the linear periodic system (5), when a stabilizing linear periodic feedback controller $K_{\sigma(k)}$ is applied, namely

$$x_k \in \mathcal{X}_I(K_{\sigma(k)}) \Rightarrow K_{\sigma(k)}x_k \in \mathbb{U} \text{ and } (A - B_{\sigma(k)}K_{\sigma(k)})x_k \in \mathcal{X}_I(K_{\sigma(k)})$$

where $\mathcal{X}_I(K_{\sigma(k)}) \subset \mathbb{X}$.

We denote the resulting optimizing control sequence as $\Delta\mathbf{u}^o(x_k)$. Only the first control $\Delta\tilde{u}_k^o$ in $\Delta\mathbf{u}^o(x_k)$ is applied to the system at time k, so that we apply the predictive control in the usual receding-horizon manner.

2.3 Scheme 2

Scheme 2 involves solving the following finite-time constrained linear periodic control problem:

$$
\begin{array}{lll}
\mathcal{P}_{\sigma(k)}(x_k): \text{Minimise} & J_k = F(x_{k+N}) + \sum_{i=0}^{N-1}\left(\|x_{k+i+1}\|_q^2 + \|\Delta\tilde{u}_{k+i}\|_r^2\right) \\
\text{wrt} & \Delta\vec{u}_{k,0} \\
\text{s.t.} & \Delta\tilde{u}_{k+i} \in \mathbb{U}, \quad (i = 0, 1, \ldots, N-1) \\
& x_{k+i} \in \mathbb{X}, \quad (i = 1, \ldots, N) \\
& x_{k+N+1} \in \mathcal{X}_I(K_{\sigma(k)}) \\
& x_{k+1} = Ax_k + B_{\sigma(k)}\Delta\tilde{u}_k \\
& \text{assumptions about } \Delta\vec{u}_{k,i}, (i = 1, \ldots, m-1) \text{ are satisfied.}
\end{array}
\tag{8}
$$

Some assumptions must be made about those inputs $\Delta\vec{u}_{k,i}, (i = 1, \ldots, m-1)$ which have already been planned but which have not yet been executed. We will assume that all such planned decisions are known to the controller, and that it assumes that they will be executed as planned. (This assumption will usually be false, because new decisions will be made in the light of new measurements.)

Thus, in Scheme 2, there are essentially m MPC controllers, operating in sequence, in a cyclic manner. They share information, however, in the sense that the complete plant state is available to each controller — although not at the same times — and the currently planned future moves of each controller are also available to all the others.

Remark 1 *Different assumptions are possible here. We will assume that each input is computed so as to optimise "its" cost over the prediction horizon, and that after the end of the horizon each input is determined according to an optimal linear state-feedback law.*

3 Stability of MMPC

3.1 Unconstrained LQ optimal control of periodic systems

The following results on unconstrained infinite-time linear quadratic control of periodic systems are known (Bittanti *et al.*, 1988). Consider the plant (5) and the quadratic cost function

$$J_k = \sum_{i=0}^{\infty} \left(\|x_{k+i+1}\|_q^2 + \|\Delta \tilde{u}_{k+i}\|_r^2 \right) \tag{9}$$

Then this cost is minimised by finding $\bar{P}_i, i = 1, \ldots, m$, the Symmetric, Periodic and Positive Semidefinite (SPPS) solution of the following discrete-time periodic Riccati equation (DPRE)

$$P_k = A^T P_{k+1} A - A^T P_{k+1} B_{\sigma(k)} (B_{\sigma(k)}^T P_{k+1} B_{\sigma(k)} + r)^{-1} B_{\sigma(k)}^T P_{k+1} A + q \tag{10}$$

and setting

$$\Delta \tilde{u}_k = -K_{\sigma(k)} x_k \tag{11}$$

where

$$K_{\sigma(k)} = (B_{\sigma(k)}^T \bar{P}_{\sigma(k+1)} B_{\sigma(k)} + r)^{-1} B_{\sigma(k)}^T \bar{P}_{\sigma(k+1)} A \tag{12}$$

Remark 2 *The set of periodic gains $\{K_j : j = 1, \ldots, m\}$ is stabilising, namely, the monodromy matrix*

$$\Psi_1 = \Phi_m \Phi_{m-1} \ldots \Phi_2 \Phi_1 \tag{13}$$

has all its eigenvalues inside the unit circle, where

$$\Phi_j = A - B_j K_j \tag{14}$$

It is a standard fact that this is a necessary and sufficient condition for closed-loop stability of a linear periodic system, and that the eigenvalues of the monodromy matrix are invariant under cyclic permutations of the Φ_j matrices.

Remark 3 *In (Bittanti* et al.*, 1988) conditions are established for the existence of the optimal solution, which generalise the familiar conditions for the existence of solutions to LQ problems for LTI systems. In this paper we shall assume that such conditions are satisfied.*

3.2 Stability of MMPC schemes

Proposition 1 *MMPC schemes 1 and 2, obtained by solving the finite-time constrained linear periodic optimal control problems* (7) *and* (8)*, respectively, give closed-loop stability if the problems are well-posed, and if $F(x_{k+N})$ is the value function of the unconstrained infinite horizon periodic optimal control problem, namely if*

$$\begin{aligned} F(x_{k+N}) &= \min_{\Delta \tilde{u}} \{ \textstyle\sum_{i=N}^{\infty} \left(\|x_{k+i+1}\|_q^2 + \|\Delta \tilde{u}_{k+i}\|_r^2 \right) \mid x_{k+1} = A x_k + B_{\sigma(k)} \Delta \tilde{u}_k \} \\ &= x_{k+N}^T \bar{P}_{\sigma(k+N)} x_{k+N} \end{aligned} \tag{15}$$

where $\bar{P}_{\sigma(k+N)}$ is defined in section 3.1.

Proof:
According to (Mayne *et al.*, 2000, page 797), if the following four conditions hold:

(1) state constraints satisfied in terminal constraint set
(2) control constraints satisfied in terminal constraint set
(3) the terminal constraint set is positively invariant under a local controller
(4) the terminal cost is a local Lyapunov function

then closed loop stability is obtained.

In our set up, conditions (1)–(3) are satisfied by assumption since $x_{k+N+1} \in \mathcal{X}_I(K_{\sigma(k)})$. We have only to check that condition (4) is satisfied, namely that $F(x_{k+N+1}) - F(x_{k+N}) \leq 0$.

Now according to (Bittanti *et al.*, 1988) $F(x_{k+N})$ as defined in (15) is the value function of the unconstrained infinite horizon (periodic) optimal control problem. As a consequence

$$F(x_{k+N}) = F(x_{k+N+1}) + \|x_{k+N+1}\|_q^2 + \|\Delta\tilde{u}_{k+N+1}\|_r^2 \tag{16}$$

Hence

$$F(x_{k+N+1}) - F(x_{k+N}) = -\|x_{k+N}\|_q^2 - \|\Delta\tilde{u}_{k+N}\|_r^2 \leq 0 \tag{17}$$

with equality satisfied if and only if $x_{k+N} = \Delta\tilde{u}_{k+N} = 0$. This shows that $F(\cdot)$ is a Lyapunov function in some neighbourhood of 0, in particular for $x \in \mathcal{X}_I(K_{\sigma(k)})$. ■

Note that the terminal cost used in scheme 2 is the same as that used in scheme 1. However, the value functions of the two problems $\mathcal{P}$ and $\mathcal{P}_{\sigma(k)}$ are different, because they have different sets of decision variables.

It can be seen that this result, and its proof, depend only on using an appropriate terminal cost $F(\cdot)$, and not at all on the details of the constrained optimization over the horizon of length N. Consequently the result and its proof hold for any other MMPC scheme which involves constrained optimization over a finite horizon, providing that the terminal cost is given by (15). For example, for a particular 3-input system, it may be beneficial to update the inputs in the sequence $(1, 2, 1, 3, 1, 2, 1, 3, \ldots)$, thus updating one of the inputs twice as often as the others. The stability of such a scheme is proved by the argument used above, providing that the sequencing function $\sigma(\cdot)$ is redefined appropriately.

However, the proof depends on the constrained optimization being feasible at each step, and the feasibility at any particular time step depends on the details of the constrained optimization problem that is being solved. (But note that with a perfect model and in the absence of disturbances, if feasible solutions are obtained over an initial period, then feasibility is assured thereafter.)

4 An Example

In this section, we give a numerical simulation example to illustrate the proposed MMPC schemes. In particular, we focus on the disturbance rejection performance and compare it with the conventional 'synchronised MPC' (SMPC) scheme. The simulations were carried out with the plant modelled in continuous-time and the controller modelled in discrete-time.

The plant has a continuous-time model

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{1}{7s+1} & \frac{1}{3s+1} \\ \frac{2}{8s+1} & \frac{1}{4s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix} \tag{18}$$

Although simple, such dynamics are representative of what is commonly encountered in, e.g. multiple chamber or multi-zone temperature control problems, where the temperatures of the individual chamber/zone need to be controlled independently in the presence of thermal interactions between the chambers/zone. A possible approach is to perform multivariable control design, but this may lead to a large centralised controller if the number of chambers/zones is large. An alternative is to employ decentralised control where each chamber is assigned a SISO controller suitably de-tuned to account for the inter-loop interactions. Here we propose the multiplexed MPC approach as a compromise between these two extremes.

The plant has two inputs and hence $m = 2$ for this example. We chose the sampling time to be $T = 1s$. More specifically, for SMPC, the states were measured at time interval of 1s and *both* u_1 and u_2 were changed at the same time and held constant over a period of 1s. For MMPC schemes, the states were measured at $T/m = 0.5s$ with u_1 and u_2 alternatively applied at 0.5s intervals, but each held constant over a period of $T = 1s$ (see Fig 1.1), i.e. u_1 is updated at times $(0, 1s, \ldots)$ and u_2 is updated at times $(0.5s, 1.5s, \ldots)$.

We designed the SMPC for this plant with the usual set-up: the control moves $\{\Delta u_{1,k}, \ \ldots, \ \Delta u_{m,k}\}$ all applied at the same time instant but held constant over period T. To make the comparison with MMPC meaningful, the terminal cost for SMPC was determined using the *same* cost function as that of MMPC, i.e. by summing up the state and control moves at every T/m period, which corresponds to increment of k by 1. More precisely, the terminal cost for SMPC is determined by the following LQ problem:

$$\min_{\Delta u} \sum_{i=0}^{\infty} \|x_{k+i+1}\|_q^2 + \|\Delta u_{k+i}\|_{\bar{R}}^2$$

subject to system (2), with $\Delta u_{k+i} = 0, i \neq 0, m, \ldots$ and $\bar{R} = \mathrm{diag}(r, \ldots, r)$.

The terminal cost for SMPC can be determined from the equivalent LQ problem with the following cost function (see appendix for a derivation and the definition of the symbols used):

$$J_{\text{SMPC}} = \sum_{j=0}^{\infty} \|x_{k+jm}\|_Q^2 + \|\Delta u_{k+jm}\|_R^2 + 2x_{k+jm}^T S \Delta u_{k+jm}$$

where $x_{k+(j+1)m} = A^m x_{k+jm} + A^{m-1} B \Delta u_{k+jm}$, $\bar{Q} = \mathrm{diag}(q, \ldots, q)$, $Q = \bar{A}^T \bar{Q} \bar{A}$, $R = \bar{B}^T \bar{Q} \bar{B} + \bar{R}$ and $S = \bar{A}^T \bar{Q} \bar{B}$.

Fig. 2 shows the simulation result when a unit step input disturbance was introduced at time 5.1s, i.e. just after input 1 has been updated. The upper half of the figure compares the closed-loop responses between SMPC and MMPC schemes 1 and 2, with the design parameter $N_u = 5$. It can be seen that due to a faster reaction time, the MMPC schemes were more effective in reducing the peak overshoot at the plant outputs, although there were larger undershoots for the MMPC schemes. The lower half of the figure shows MMPC scheme 2 for $N_u = 1, \ldots, 5$. With increasing N_u the peak on the output after a disturbance is reduced but the undershoot increases.

Fig. 3 shows the same simulation scenario when input constraints of $|u_1| \leq 1.1$ and $|u_2| \leq 1.1$ were added. It can be seen that the undershoot in MMPC scheme 2 has been significantly reduced. Again, increasing N_u in MMPC scheme 2 has the effect of reducing the peak overshoot at the plant outputs.

MMPC are periodic control schemes, thus their performance depends on the time at which the disturbance occurs. Fig 4 and 5 show a similar set of simulation results but when the input disturbance occurs at time 5.6s, i.e. just after input 2 has been updated. From Fig. 4, it can be seen that MMPC has little advantage in the y_1 channel, while still offering some advantage in the y_2 channel. Again, we notice that increasing N_u in MMPC scheme 2 has the effect of reducing the peak overshoot at the plant outputs, and adding constraints on the inputs helps to reduce the undershoots.

In this example MMPC gives better performance than SMPC, as judged by recovery from step disturbances. This cannot be expected to be true always. But even if the performance is worse than that of SMPC, MMPC scheme 2 offers considerably lower computational complexity, which might make it the more desirable control algorithm.

5 Conclusion

In this work, two versions of a novel control scheme known as *Multiplexed* MPC were proposed. The second of these is expected to be of practical benefit because it offers reduced computational complexity. Both of our proposed MMPC schemes have been proved to be nominally stable. The nominal stability of a large class of other multiplexed MPC schemes follows by the same argument as we used in this paper.

It is interesting, and potentially important, to observe that the assumption of equal intervals between the updates of plant inputs is not essential to our proposal. Any pattern of update intervals can be supported, providing that it repeats in subsequent update cycles.

Some performance benefit over conventional MPC can be obtained as a result of faster reactions to disturbances, despite suboptimal solutions being obtained. This has been demonstrated by an example. However, the closed loop disturbance rejection performance under MMPC is time varying because of the periodic nature of the control scheme.

References

Barrett, R., M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine and H. Van der Vorst (1994). *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. 2nd ed.. SIAM. Philadelphia PA.

Bittanti, S., P. Colaneri and G. De Nicolao (1988). The difference periodic Riccati equation for the periodic prediction problem. *IEEE Transactions on Automatic Control,* **33**(8), 706–712.

Blanchini, F. (1999). Set invariance in control. *Automatica* **35**, 1747–1767.

Bleris, L.G., P.D. Vouzis, M.G. Arnold and M.V. Kothare (2006). A co-processor fpga platform for the implementation of real-time model predictive control. In: *Proc. American Control Conference*. pp. 1912–1917. Minneapolis.

Box, G. and A. Luceno (1997). *Statistical Control by Monitoring and Feedback Adjustment*. Wiley. New York.

Johanson, T.A., W. Jackson, R. Schreibar and P. Tondel (2006). Hardware architecture design for explict model predictive control. In: *Proc. American Control Conference*. pp. 1924–1929. Minneapolis.

Ling, K.V., S.P. Yue and J.M. Maciejowski (2006). An FPGA implementation of model predictive control. In: *Proc. American Control Conference*. pp. 1930–1935. Minneapolis.

Maciejowski, J.M. (2002). *Predictive Control with Constraints.* Prentice-Hall. Harlow UK.

Mayne, D.Q., J.B. Rawlings, C.V. Rao and P.O.M. Scokaert (2000). Constrained model predictive control: stability and optimality. *Automatica* **36**, 789–814.

Morari, M., M. Baotić and F. Borrelli (2003). Hybrid systems modeling and control. *European Journal of Control* **9**(2–3), 177–189.

Murray, R.M., J. Hauser, A. Jadbabie, M.B. Milam, N. Petit, W.B. Dunbar and R. Franz (2003). Online control customization via optimization-based control. In: *Software-Enabled Control* (T. Samad and G. Balas, Eds.). IEEE Press and Wiley.

Papoulis, A. (1984). *Probability, Random Variables, and Stochastic Processes.* 2nd ed.. Mc-Graw Hill.

Perez, T., G.C. Goodwin and C.W. Tzeng (2000). Model predictive rudder roll stabilization control for ships. In: *Proc. 5th IFAC Conf. on Manoeuvring and Control of Marine Craft.* Aalborg, Denmark.

Richards, A. and J.P. How. (2003). Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility. In: *Proc. American Control Conference.* Denver.

Venkat, A.N., J.B. Rawlings and S.J. Wright (2004). Plant-wide optimal control with decentralized MPC. In: *Proc. 7th International Symposium on Dynamics and Control of Process Systems (DYCOPS).* Cambridge, MA.

A Derivation of the Equivalent LQ Problem for SMPC

First, denote $\bar{Q} = \mathrm{diag}(q, \ldots, q)$ and

$$\bar{x}_{k+jm} = \begin{bmatrix} x_{k+jm+1} \\ x_{k+jm+2} \\ \vdots \\ x_{k+jm+m} \end{bmatrix} = \bar{A}x_{k+jm} + \bar{B}\Delta u_{k+jm}$$

where

$$\bar{A} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^m \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} B \\ AB \\ \vdots \\ A^{m-1}B \end{bmatrix}$$

Next, perform the following manipulations to the SMPC cost function:

$$\begin{aligned}
J_{\text{SMPC}} &= \sum_{i=0}^{\infty} \|x_{k+i+1}\|_q^2 + \|\Delta u_{k+i}\|_{\bar{R}}^2 \\
&= \sum_{j=0}^{\infty} \sum_{i=0}^{m-1} (\|x_{k+jm+i+1}\|_q^2 + \|\Delta u_{k+jm+i}\|_{\bar{R}}^2) \\
&= \sum_{j=0}^{\infty} \|\bar{x}_{k+jm}\|_{\bar{Q}}^2 + \|\Delta u_{k+jm}\|_{\bar{R}}^2 \\
&= \sum_{j=0}^{\infty} \|\bar{A}x_{k+jm} + \bar{B}\Delta u_{k+jm}\|_{\bar{Q}}^2 + \|\Delta u_{k+jm}\|_{\bar{R}}^2 \\
&= \sum_{j=0}^{\infty} \|x_{k+jm}\|_{\bar{A}^T\bar{Q}\bar{A}}^2 + \|\Delta u_{k+jm}\|_{\bar{B}^T\bar{Q}\bar{B}+\bar{R}}^2 + 2x_{k+jm}^T\bar{A}^T\bar{Q}\bar{B}\Delta u_{k+jm} \\
&= \sum_{j=0}^{\infty} \|x_{k+jm}\|_Q^2 + \|\Delta u_{k+jm}\|_R^2 + 2x_{k+jm}^T S\Delta u_{k+jm}
\end{aligned}$$

where

$$Q = \bar{A}^T\bar{Q}\bar{A}, \quad R = \bar{B}^T\bar{Q}\bar{B} + \bar{R}, \quad \text{and } S = \bar{A}^T\bar{Q}\bar{B}$$

and note that

$$x_{k+(j+1)m} = A^m x_{k+jm} + A^{m-1}B\Delta u_{k+jm}.$$

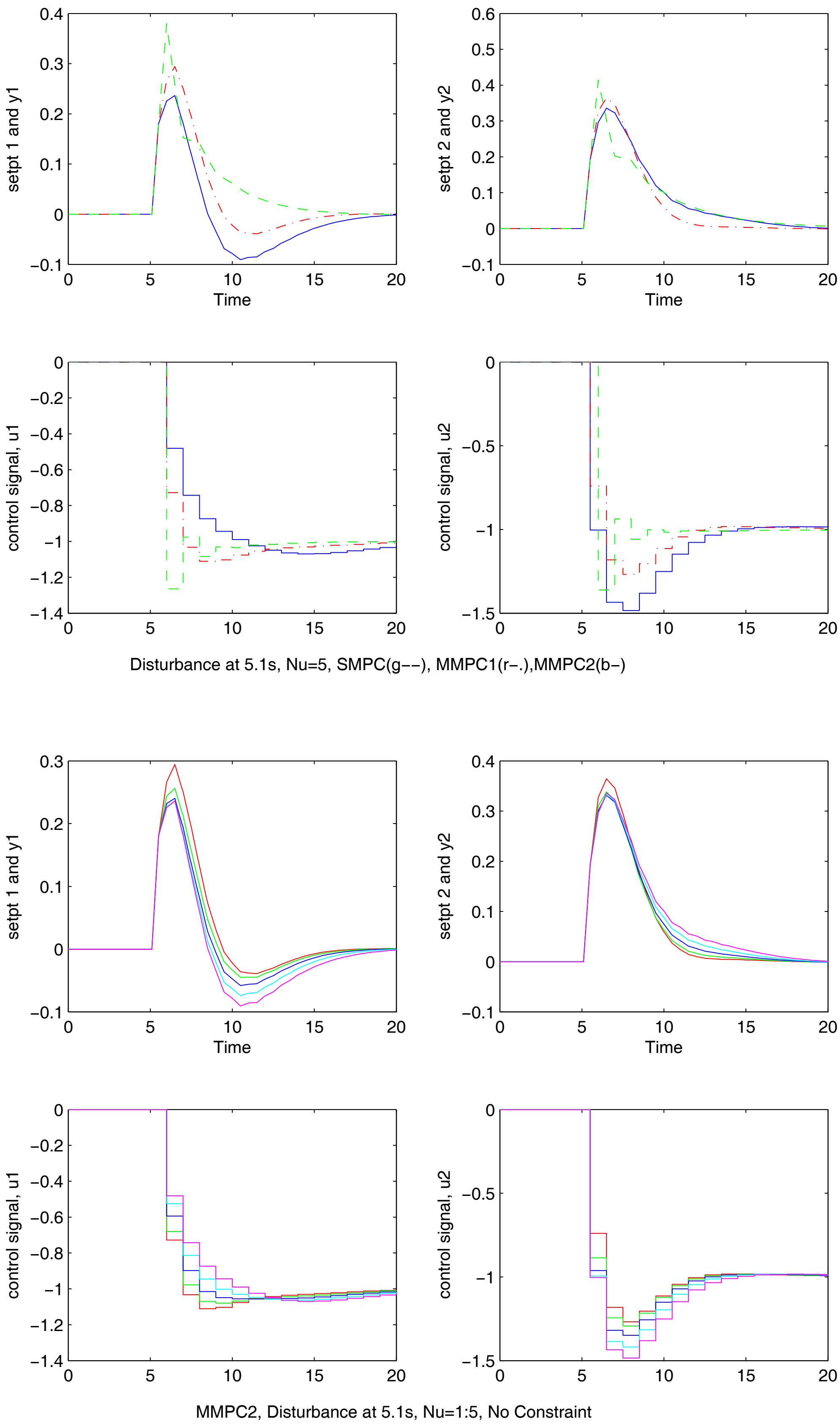
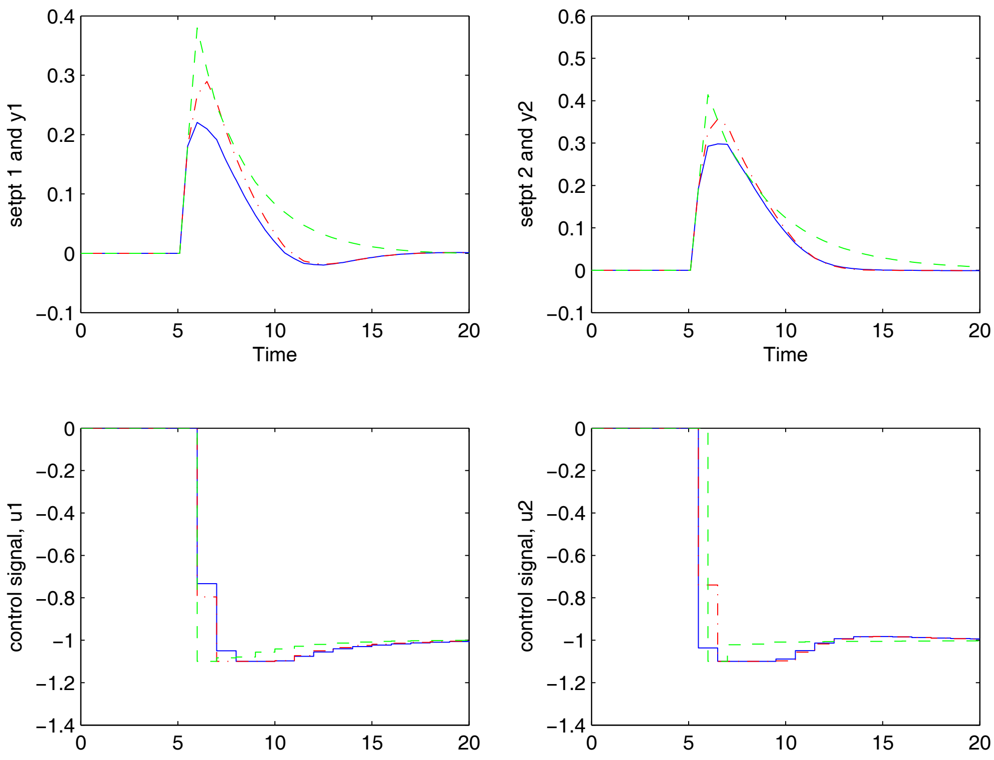
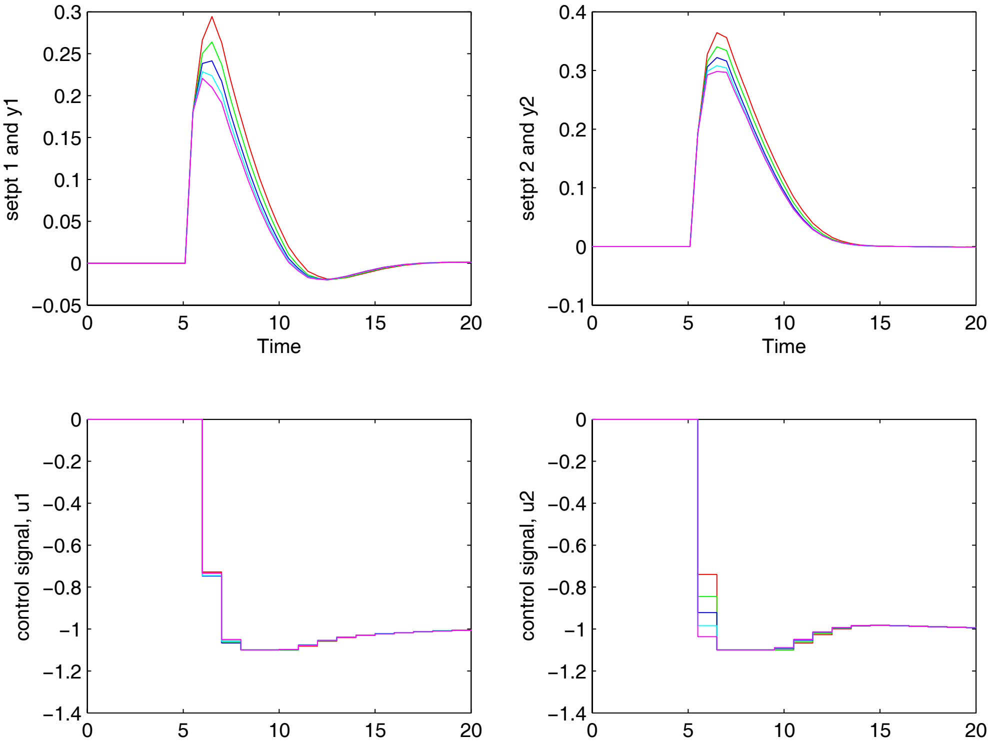


Fig. 2. Disturbance at 5.1s, no constraint. SMPC(dash), MMPC scheme 1(dash-dot), MMPC scheme 2(solid)



Disturbance at 5.1s, Nu=5, SMPC(g−−), MMPC1(r−.),MMPC2(b−)



MMPC2, Disturbance at 5.1s, Nu=1:5, No Constraint

Fig. 3. Disturbance at 5.1s, with constraint. SMPC(dash), MMPC scheme 1(dash-dot), MMPC scheme 2(solid)

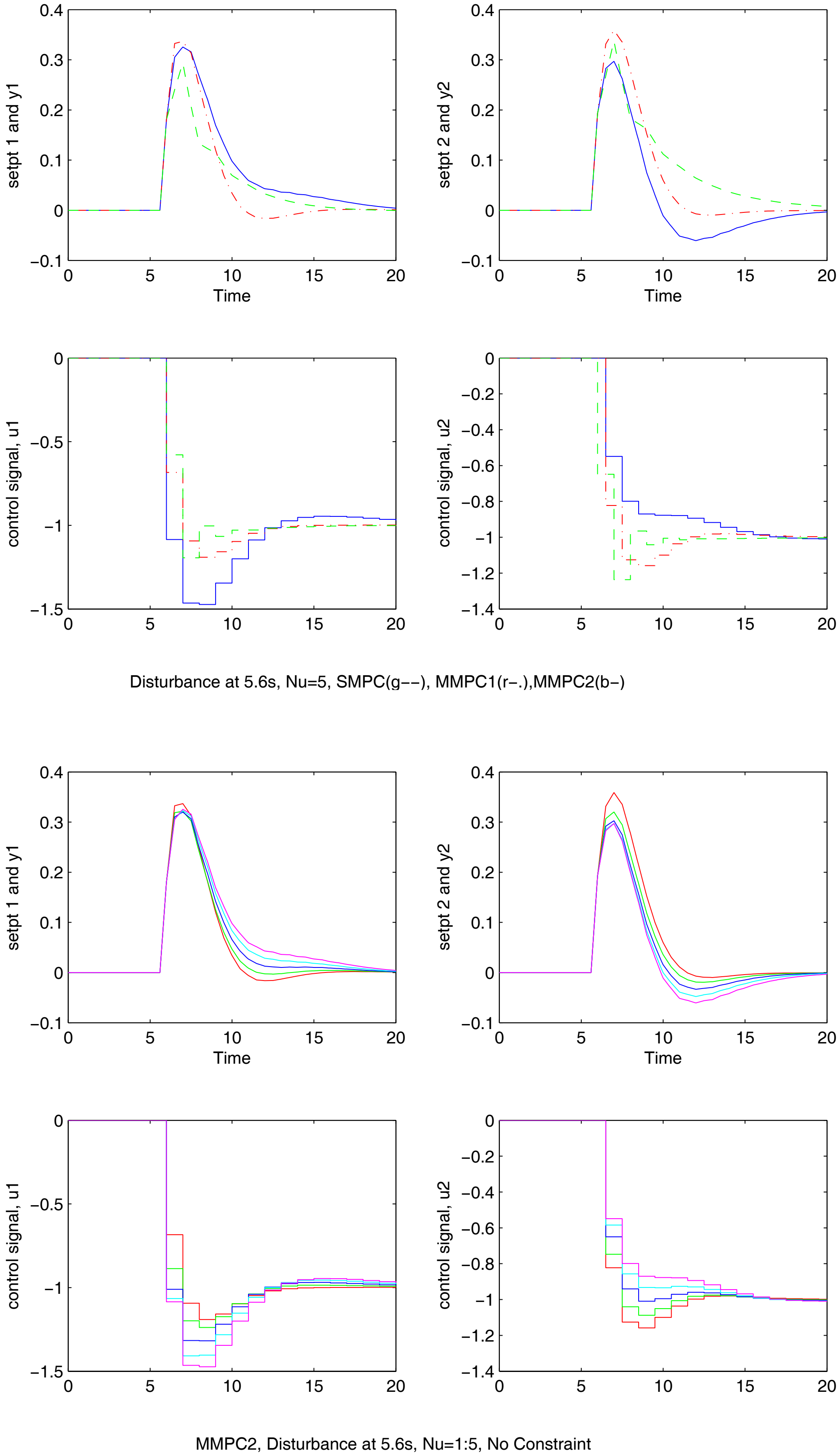


Fig. 4. Disturbance at 5.6s, no constraint. SMPC(dash), MMPC scheme 1(dash-dot), MMPC scheme 2(solid)

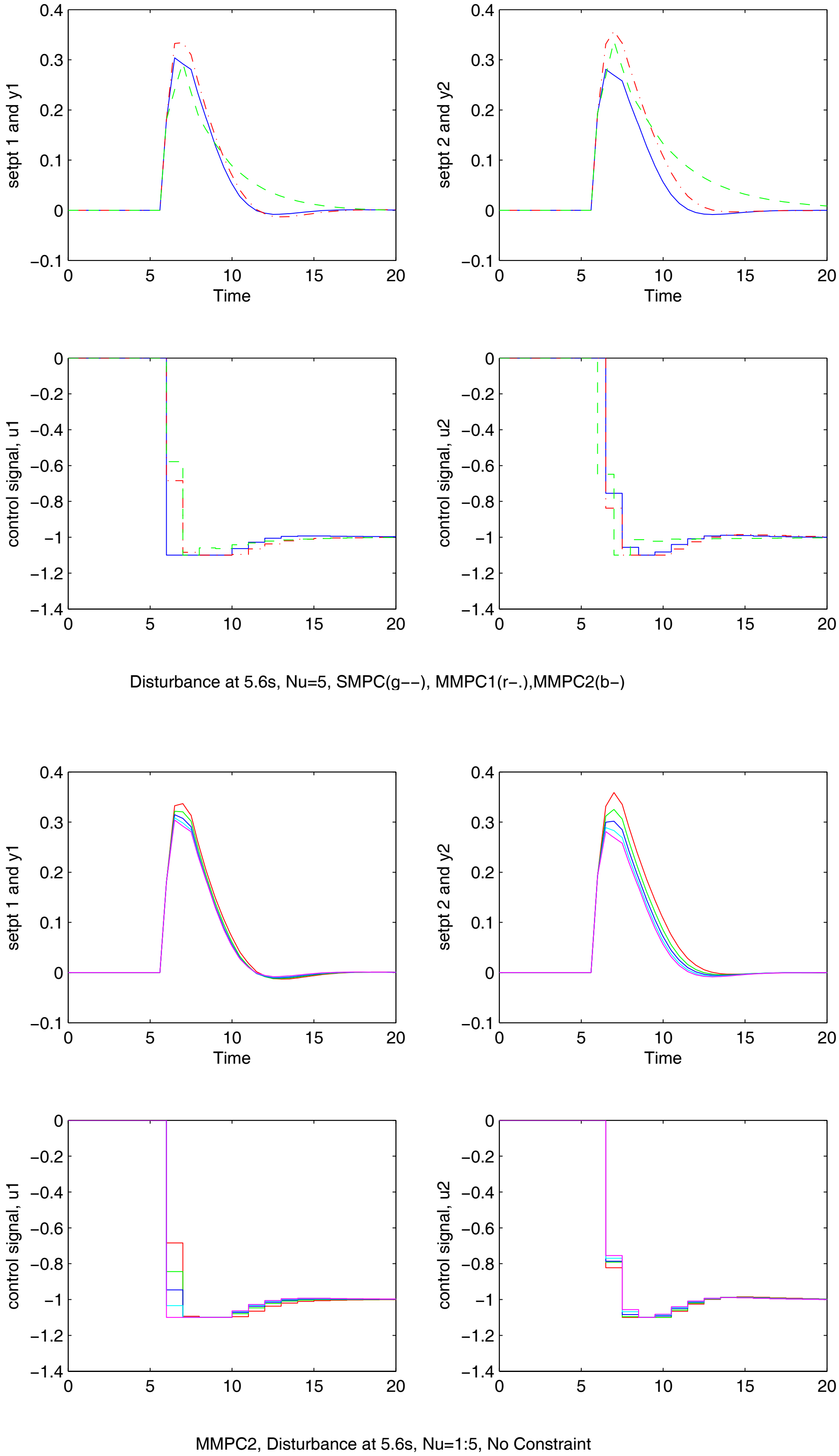


Fig. 5. Disturbance at 5.6s, with constraint. SMPC(dash), MMPC scheme 1(dash-dot), MMPC scheme 2(solid)