Nonlinear Dynamics and Systems Theory, 3(2) (2003) 163–176

Nash-Optimisation Enhanced Distributed Model Predictive Control

Shao-Yuan Li[1], Xiao-Ning Du[1] and Quan-Min Zhu[2]

[1] *Institute of Automation, Shanghai Jiao Tong University, 1954 Hua Shan Road, Shanghai 200030, P.R.China*

[2] *Faculty of CEMS, University of the West of England, Coldharbour Lane, Bristol, BS16 1QY, UK*

Received: May 28, 2003;   Revised: September 15, 2003

Abstract: This study presents an efficient distributed model predictive control scheme based on Nash optimality, in which the on-line optimisation of the whole system is decomposed into that of several small co-operative agents in distributed structures, thus it can significantly reduce computational complexity in model predictive control of large-scale systems. The relevant nominal stability and the performance on single-step horizon under the communication disturbance are investigated. A three input and three output linear model is simulated to test the effectiveness of the proposed control algorithm.

Keywords: *Model predictive control (MPC); distributed control system; Nash optimality; multi-agents.*

Mathematics Subject Classification (2000): 93C55, 93C10, 49K99.

1 Introduction

Model predictive control (MPC) is a popular technique and has been successfully used in the control of various linear and nonlinear dynamic systems (see [1, 7, 17]). However, an obvious drawback of MPC involved in the formidable on-line computational effort limits its applicability to relatively fast and/or large processes with moderate number of inputs ([5]). Practically, there exists a great number of complex high dimensional systems, in which the number of variables and constraints is of ten several dozens or even several hundreds. Thus it has become very important to develop computationally efficient control architectures and algorithms with less computational burden. Unfortunately, with the possible exception of the studies by [9, 13, −16]. Van Antwerp and Braatz [10] to reflect the large-scale nature of typical industrial plants, references for this topic are little in open literature. This probably is attributable to the inherent difficulties involved incomplex computation for large-scale processes. With the rapid development of communication network and the field-bus technology, centralised control has not been a sole structure in applications and has been gradually replaced by distributed control in large-scale systems. Distributed control structure brings new requirements to the traditional control field and allows the conceivability of new challenging control applications. For economic consideration and also no degrading performance, it is desirable to use several inexpensive microcomputers to replace a very high performance computer in control systems. The development of communication network and the field-bus technology has provided possibility for this distributed control. Xu, *et al.* [13] and Xi [12] proposed a decentralised predictive control algorithm. Zheng [14, 15], Zheng and Allgower [16], proposed a one-step approximation algorithm to reduce the on-line computation by decreasing the number of the decision variables. More recently Gurfil and Kasdin [3] developed an iterative ellipsoid algorithm to allow the quick computation of sub-optimal control moves. It should be pointed out that these approaches still take centralised computation and therefore need high cost computers. In this study, an efficient distributed optimisation scheme is developed based on Nash optimality for MPC of large-scale systems. Under this scheme, on-line optimisation of the whole system is decomposed into that of several small co-operative agents. These agents can co-operate and communicate each other in a distributed structure to achieve the objective of the whole system. Accordingly the computational complexity for such large-scale systems is significantly reduced. Since the protocol of mutual communication and information exchange is adequately taken into account, this approach can efficiently improve control performance and guarantee the Nash optimality ([6]). The second part of the study is to analyse the relevant performance of the developed method. The nominal stability and the convergent condition of this distributed control system are derived. The performance deviation on single-step horizon under the communication disturbance is also analysed with an assumption that the algorithm is convergent. The significance of this scheme is to reduce the computational burden in complex large-scale systems. Also it can be extended to the remote control and multi-agent systems. The main contents of the study is divided into five sections. In Section 2 distributed MPC algorithm based on Nash optimality is proposed. In Section 3 the convergent condition of the distributed predictive control algorithm for linear models is analysed. In Sections 4 and 5 the nominal stability and the performance deviation under disturbance are analysed respectively. In Section 6 a simulation example is presented to demonstrate the efficiency of the distributed MPC algorithm.

2 Distributed Model Predictive Control Algorithm Based on Nash Optimality

2.1 Model predictive control

Model predictive control (MPC) is formulated as resolving an on-lineopen-loop optimal control problem in moving horizon style. Using the current state, an input sequence is calculated to minimise a performance index while satisfying some specified constraints. Only the first element of the sequence is taken as controller output. At the next sampling time, the optimisation is resolved with new measurements from the plant. Thus both the control horizon and the prediction horizon move or recede ahead by one step at next sampling time. This is the reason why MPC is also sometimes referred to as receding horizon control (RHC) or moving horizon control (MHC). The purpose of taking new measurements at each sampling time is to compensate for unmeasured disturbances and model inaccuracy, both of which cause the system output to be different from its prediction. Suppose the prediction output model of the whole system is described as

$$Y(k+j \mid k) = f(Y(k), \Delta u(k|k)) \quad (j = 1, \dots, P), \tag{1}$$

where $\Delta u_M(k) = \left(\Delta u_{1,M}^T(k) \dots \Delta u_{m,M}^T(k)\right)^T$ is the increment of the manipulated (the controller output, also the input toplant) variables of the system, denotes the prediction horizon, denotes the control horizon, is the mapping function vector, where the element satisfied some smooth condition. The performance index of the whole system is

$$\min_{\Delta u_M(k|k)} J = \sum_{i=1}^{P} L[y(k+i \mid k), \Delta u_M(k \mid k)], \tag{2}$$

where L is the nonlinear function of input and output variables. The objective of the whole system is to regulate the system output to the expected values while keeping the performance minimal. For large-scale systems, because of the effect of control horizon M, the optimised variables $\Delta u_M(k)$ at each sampling time are highly dimensional, the computation is intensive, especially for nonlinear systems, which accordingly requires high performance computers or some advanced algorithms. To avoid the prohibitively high on-line computational demand, this study proposes a distributed scheme with inexpensive agent computers under network environment.

2.2 Distributed MPC strategy based on Nash optimality

The main idea of the distributed model predictive control algorithm is the on-line optimisation of MPC. Since an optimisation formulation in large-scale systems can be decomposed into a number of small-scale optimisations. These autonomous agents are connected via network with dynamic input coupling among them, share the common resources, communicate and co-ordinate each other in order to accomplish the whole objective. Suppose the behaviour of the whole system is described by m agents and the performance index (2) is separable for m agents. The local performance index for the i-th agent can be expressed as

$$\min_{\Delta u_i, M} J_i = \sum_{j=1}^{P} L_i[y_i(k+j \mid k), \Delta u_{i,M}(k \mid k)]. \tag{3}$$

This indicates the global performance index of the whole system is

$$\min J = \sum_{i=1}^{m} J_i. \tag{4}$$

At time instant k, the future predictive output of the i-th agent can be expressed as

$$y_i(k+j \mid k) = f_i[y_i(k), \Delta u_{1,M}(k \mid k), \cdots, \Delta u_{m,M}(k \mid k)], \quad (i = 1, \dots, P). \tag{5}$$

It can be seen that the global performance index can be decomposed into a number of local performance indices, but the output of each agent is still related to all the input variables due to the input coupling. Such distributed control problem with different goals can be resolved by means of Nash optimal concept ([6]). Concretely speaking, each agent optimises its objective (local performance index) only using its own input variables assuming that the other agent's optimal solutions are known, that is

$$\left.\frac{\partial J_i}{\partial \Delta u_{i,M}(k)}\right|_{\Delta u^*_{j,M}(k),\, j=1,\ldots,m,\, j\neq i} = 0 \quad (i=1,\ldots,m). \tag{6}$$

Thus the resulted Nash optimal solution satisfies the Nash optimality condition

$$\begin{aligned} J_i(\Delta u^*_{1,M}(k),\cdots,\Delta u^*_{m,M}(k)) \leq J_i(\Delta u^*_{1,M}(k),\cdots,\Delta u^*_{i-1,M}(k),\\ \Delta u_{i,M}(k),\Delta u^*_{i+1,M}(k),\cdots\Delta u^*_{m,M}(k)). \end{aligned} \tag{7}$$

Inspection of (5) to obtain the Nash optimal solution $\Delta u^*_{i,M}(k)$ of the i-th agent, it is necessary to know the other agent's Nash optimal solutions $\Delta u^*_{j,M}(k)$ $(j \neq i)$, so that the whole system could arrive at Nash optimal equilibrium in this coupling decision process. Here an iterative algorithm is proposed to seek the Nash optimal solution of the whole system at each sampling time. Each agent compares the newly computed optimal solution with that obtained in last iteration, and checks if its terminal condition is satisfied. If the algorithm is convergent, all the terminal conditions of the m agents will be satisfied, and the whole system will arrive at Nash equilibrium at this time. This Nash optimisation process will be repeated at next sampling time.

Algorithm:

Step 1: At sampling time k, each agent makes initial estimation of the input variables and announces it to the other agents, let the iterative index $l=0$,

$$\Delta\bar{u}^l_{i,M}(k) = [\Delta\bar{u}^l_i(k),\Delta\bar{u}^l_i(k+1),\cdots,\Delta\bar{u}^l_i(k+M-1)]^T, \quad (i=1,\cdots,m).$$

Step 2: Each agent resolves its optimal problem simultaneously to obtain its solution $\Delta u^*_{i,M}(k)$, $(i=1,\cdots,m)$.

Step 3: Each agent checks if its terminal iteration condition is satisfied, that is, for the given error accuracy ε_i, $(i=1,\cdots,m)$, if there exist

$$\|\Delta u^{l+1}_{i,M}(k) - \Delta\bar{u}^l_{i,M}(k)\| \leq \varepsilon_i \quad (i=1,\cdots,m).$$

If all the terminal conditions are satisfied, then end the iteration and go to step 4; otherwise, let $l=l+1$, $\Delta\bar{u}^l_{i,M}(k) = \Delta u^*_{i,M}(k)$, $(i=1,\cdots,m)$ all agents communicate to exchange this information, and take the latest solution to step 2.

Step 4: Compute the instant control law

$$\Delta u_i(k) = [I \ \cdots \ \mathbf{0}]\Delta u^*_{i,M}(k) \quad (i=1,\cdots,m)$$

and take the first element as the controller output from each agent.

Step 5: Move horizon to the next sampling time, that is, $k+1 \to k$, and go to step 1.

3 Computational Convergence for Linear Systems

Consider this distributed model predictive control of linear dynamic plants. At sampling time k, the output prediction model of the i-th agent can be described as

$$\tilde{y}_{i,PM}(k) = \tilde{y}_{i,P0}(k) + A_{ii}\Delta u_{i,M}(k) + \sum_{\substack{j=1 \\ j\neq i}}^{m} A_{ij}\Delta u_{j,M}, \qquad (i = 1,\cdots,m), \tag{8}$$

where A_{ii} and A_{ij} are the dynamic matrix of the i-th agent and the step response matrix of the i-th agent stimulated by the j-th agent respectively. They are expressed in terms of matrix

$$A_{ij} = \begin{bmatrix} a_{ij}(1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ a_{ij}(M) & \dots & a_{ij}(1) \\ \vdots & \vdots & \vdots \\ a_{ij}(P) & \cdots & a_{ij}(P-M+1) \end{bmatrix}, \qquad A = \begin{bmatrix} A_{11} & \cdots & A_{1m} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mm} \end{bmatrix},$$

where $a_{ij}(k)$, $(k = 1,2,\dots,\ i,\ j = 1,\dots,m)$ is the step response matrix array. The local performance index for the i-th agent can be expressed as

$$\min J_i = \|\varpi_i(k) - \tilde{y}_{i,PM}(k)\|^2_{Q_i} + \|\Delta u_{i,M}(k)\|^2_{R_i} \quad (i = 1,\cdots,m), \tag{9}$$

where $\varpi_i(k) = [\varpi_i(k+1) \ \cdots \ \varpi_i(k+P)]^T$, $(i = 1,\cdots,m)$ is the expected output of the i-th agent, and

$$\begin{aligned} \tilde{y}_{i,PM}(k) &= [\tilde{y}_{i,M}(k+1 \mid k) \ \cdots \ \tilde{y}_{i,M}(k+P \mid k)]^T, \\ \tilde{y}_{i,P0}(k) &= [\tilde{y}_{i,0}(k+1 \mid k) \ \cdots \ \tilde{y}_{i,0}(k+P \mid k)]^T, \\ \Delta u_{i,M}(k) &= [\Delta u_i(k|k) \ \cdots \ \Delta u_i(k+M-1 \mid k)]^T. \end{aligned}$$

According to Nash optimality, at sampling time k, the Nash optimal solution of the i-th agent can be derived as

$$\Delta u^*_{i,M}(k) = D_{ii}[\varpi_i(k) - \tilde{y}_{i,P0}(k) - \sum_{\substack{j=1 \\ j\neq i}}^{m} A_{ij}\Delta u_{j,M}(k)] \quad (i = 1,\cdots,m), \tag{10}$$

where $D_{ii} = (A^T_{ii}QA_{ii} + R_i)^{-1}A^T_{ii}Q_i$. If the algorithm is convergent, the Nash optimal solution of the whole system can be written as

$$\Delta u_M(k) = D_1[\varpi(k) - \tilde{y}_{P0}(k)] + D_0\Delta u_M(k), \tag{11}$$

where

$$D_1 = \begin{bmatrix} D_{11} & & & \\ & D_{22} & & \\ & & \ddots & \\ & & & D_{mm} \end{bmatrix},$$

$$D_0 = \begin{bmatrix} 0 & -D_{11}A_{12} & \dots & -D_{11}A_{1m} \\ -D_{22}A_{21} & 0 & \dots & -D_{22}A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ -D_{mm}A_{m1} & \dots & \dots & 0 \end{bmatrix}.$$

In the iteration procedure, equation (10) can be expressed as

$$\Delta u_M^{l+1}(k) = D_1[\varpi(k) - \tilde{y}_{P0}(k)] + D_0 \Delta u_M^l(k) \quad (l = 0, 1, \dots). \tag{12}$$

At time instant k, $\varpi(k)$ and $\tilde{y}_{P0}(k)$ are known in advance, hence $D_1[\varpi(k) - \tilde{y}_{P0}(k)]$ is the constant term irrelevant to the iteration. The convergence of expression (11) is then equivalent to that of the following

$$\Delta u_M^{l+1}(k) = D_0 \Delta u_M^l(k). \tag{13}$$

From the above analysis the convergence condition for the algorithm in application to distributed linear model predictive control is

$$|\rho(D_0)| < 1. \tag{14}$$

That is the spectrum radius must be less than 1 to guarantee a convergent computation.

4 Nominal Stability of Distributed Model Predictive Control System

In order to analyse the nominal stability, rewrite the prediction output model of (8) in terms of state space equation ([11]). The predictive state space model of the i-th agent at time instant can be written as

$$\begin{aligned} x_i(k+1) &= Sx_i(k) + a_{ii}\Delta u_i(k) + \sum_{\substack{j=1 \\ j \neq i}}^{m} a_{ij}\Delta u_j, \\ Y_i(k) &= GSx_i(k) + A_{ii}\Delta u_{i,M}(k) + \sum_{\substack{j=1 \\ j \neq i}}^{m} A_{ij}\Delta u_{j,M}, \\ & (i = 1, \cdots, m), \end{aligned} \tag{15}$$

where $\Delta u_i(k) = [1 \quad \cdots \quad 0]\Delta u_{i,M}(k)$

$$S = \begin{bmatrix} 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 \\ 0 & \dots & 0 & 1 \end{bmatrix}_{(N*N)},$$

where N is the modelling horizon, and

$$a_{ij} = [a_{ij}(1) \quad \cdots \quad a_{ij}(N)]^T, \qquad x_i(k) = [x_{i1}(k) \quad \cdots \quad x_{iN}(k)]^T,$$
$$Y_i(k) = [y_i(k+1) \quad \cdots \quad y_i(k+P)]^T.$$

$G = [I_{P*P} \ \ \mathbf{0}_{P*(N-P)}]$ denotes the operation of taking out the first P vectors from the N dimensional vectors. The Nash optimal solution in state space expression of the i-th agent at time instant k is

$$\Delta u_{i,M}(k) = D_{ii}[\varpi_i(k) - GSx_i(k) - \sum_{\substack{j=1 \\ j\neq i}}^{m} A_{ij}\Delta u_{j,M}(k)]. \tag{16}$$

The integral Nash optimal solution of the whole system provided that the algorithm is convergent at each sampling time can be written as

$$\Delta U(k) = (I - D_0)^{-1} D_1[\varpi(k) - F_2 X(k)]. \tag{17}$$

This is the state feedback control law. The instant control law of the whole system is $\Delta u(k) = L\Delta U(k)$, where

$$\begin{aligned}
&L = Block - diag(\underbrace{L_0 \quad \cdots \quad L_0}_{m}),\\
&L_0 = (1 \quad 0 \quad \cdots \quad 0)_{1*M},\\
&F_2 = Block - diag(\underbrace{GS, \cdots, GS}_{m}),\\
&\Delta U(k) = [\Delta u_{1,M}(k) \quad \cdots \quad \Delta u_{m,M}(k)]^T,\\
&\varpi(k) = [\varpi_1(k) \quad \cdots \quad \varpi_m(k)]^T,\\
&X(k) = [x_1(k) \quad \cdots \quad x_m(k)]^T.
\end{aligned}$$

Without loss of generality, let the expected output

$$\varpi_i(k+1) = 0, \quad (i = 1, \cdots, m).$$

Then the state space model of the whole system at time instant k can be expressed as

$$X(k+1) = F_1 X(k) + BL\Delta U(k) = [F_1 - BL(I - D_0)^{-1} D_1 F_2]X(k), \tag{18}$$

where

$$F_1 = Block - diag(\underbrace{S, \cdots, S}_{m}),$$

$$B = \begin{bmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mm} \end{bmatrix}.$$

The expression (17) shows the state mapping relationship of the distributed system between time instant k and time instant $k+1$. According to contraction mapping principle ([11]), the nominal stability of the whole distributed system can be guaranteed, if and only if

$$\lambda(F_1 - (I - D_0)D_1F_2) < 1. \tag{19}$$

That is, the eigen values of state mapping are less than 1.

5 Disturbance Analysis with Single-Step Horizon Control

In distributed control, each agent can work independently to achieve its local objective, but cannot accomplish the whole task on its own. These autonomous agents can communicate and co-ordinate each other, exchange information through network in order to accomplish the whole task or objective. If a distributed system is subjected to disturbance, does this strategy work well and what does the performance of the whole system change? In this section, the performance deviation on *single-step horizon* under the communication disturbance is discussed. Because MPC takes a receding-horizon control policy in which the optimisation is resolved on-line at each sampling time with updated measurements, it is reasonable to focus on single-step horizon.

In the following analysis, assume that the prediction horizon and the control horizon are equal and the communication disturbance is confined within stable region. To indicate the communication connection between agents, define a connection matrix E. All elements in the main diagonal of E are zeros and other elements in the non-main diagonal of E are 1 or 0. 1 denotes no communication disturbance, and 0 shows communication disturbance existed and the corresponding communication channel is shut up. The output prediction model and the Nash optimal solution of the i-th agent at time instant k can be respectively rewritten as

$$\tilde{y}_{i,PM}(k) = \tilde{y}_{i,P0}(k) + A_{ii}\Delta u_{i,M}(k) + \sum_{\substack{j=1 \\ j\neq i}}^{m} e_{ij}A_{ij}\Delta u_{j,M} \qquad (i = 1, \cdots, m), \tag{20}$$

and

$$\Delta u^*_{i,M}(k) = (A_{ii}^T Q_i A_{ii} + R_i)^{-1} A_{ii}^T Q_i [\varpi_i - \tilde{y}_{i,P0}(k) - \sum_{\substack{j=1 \\ j\neq i}}^{m} G_{ij}\Delta u^*_{j,M}(k)] \qquad (i = 1, \cdots, m). \tag{21}$$

Here $G = E \cdot A$, "$\cdot$" denotes the dot multiplication. The Nash optimal solution of the whole system under convergent computation is

$$\Delta u^*_M(k) = (I - D_E)^{-1} D_1[\varpi(k) - \tilde{y}_{P0}(k)], \tag{22}$$

where

$$D_E = \begin{bmatrix} 0 & -D_{11}e_{12}A_{12} & \cdots & -D_{11}e_{1N}A_{1N} \\ -D_{22}e_{21}A_{21} & 0 & & -D_{22}e_{2N}A_{2N} \\ \vdots & & \ddots & \\ -D_{NN}e_{N1}A_{N1} & & & 0 \end{bmatrix}.$$

To analyse system performance deviation, define a disturbance matrix T. The disturbance matrix T is a diagonal matrix or block diagonal matrix. For diagonal matrix, define the elements of its main diagonal as 1 or 0. For block diagonal matrix, the elements of its main diagonal block are all 1s or all 0s. The value 0 corresponds to no disturbance, and 1 for the communication disturbance existed.

Remark 5.1 Here the communication disturbance is classified into three cases

(1) Row disturbance, that is, the disturbance happens on the receiving channels. In this case the agent cannot receive the information coming from other agents, equivalently the corresponding row of matrix G becomes 0 and G becomes G', or, $G' = G - G''$, $G'' = TG$ and the corresponding element of disturbance matrix T has changed from 0 to 1;

(2) Column disturbance, that is, the disturbance happens on the transmitting channels. In this case, the agent cannot send its information to other agents, equivalently the corresponding column of matrix G becomes 0 and G becomes G', or, $G' = G - G''$, $G'' = GT$;

(3) Mixed disturbance. In this case, both row and column disturbances exist and $G' = G - G'' = G - TGT$.

With these preliminaries a theorem is presented.

Theorem 5.1 *For a distributed system, assume that the prediction horizon and the control horizon are equal and the communication disturbance cannot affect the stability. Its performance at time instant k under the local communication disturbance is degrading. The degrading magnitude δ satisfies $0 \le \delta \le \delta_{\max}$, and the upper bound of this magnitude $\delta_{\max}$ is*

$$\delta_{\max} = \frac{t_W(W_{\max})}{\lambda_m(F)},$$

where $t_W(W_{\max})$ denotes the norm of $W_{\max}$ and $\lambda_m(F)$ is the minimal eigen value of F with

$$F = [D_1^{-1}(I - D_E) - A]^T Q [D_1^{-1}(I - D_E) - A] + R,$$

$$\begin{aligned} W_{\max} = {} & (A - \overline{A} - G)^T Q (A - \overline{A} - G) + [D_1^{-1}(I - D_E) - A]^T Q (A - \overline{A} - G) \\ & + (A - \overline{A} - G)^T Q [D_1^{-1}(I - D_E) - A], \end{aligned}$$

$$\overline{A} = \begin{bmatrix} A_{11} & & \\ & \ddots & \\ & & A_{NN} \end{bmatrix}.$$

Proof Without loss of generality, take the column disturbance as an example, it has

$$D''_E = D_E T \quad D'_E = D_E - D''_E = D_E - D_E T.$$

The Nash optimal solution of the whole system in this case is

$$\Delta u_M^{dis}(k) = (I - D_E + D_E T)^{-1} D_1 [\varpi(k) - \tilde{y}_{P0}(k)]. \tag{23}$$

Using the matrix decomposition technique, it gives

$$\begin{aligned}(I - D_E + D_E T)^{-1} &= [2(I - D_E) + (D_E + D_E T - I)]^{-1} \\ &= [2(I - D_E)]^{-1} - [2(I - D_E)]^{-1}\{[2(I - D_E)]^{-1} \\ &\quad + (D_E + D_E T - I)^{-1}\}^{-1}[2(I - D_E)]^{-1}.\end{aligned} \tag{24}$$

In general $(D_E + D_E T - I)^{-1}$ and $(I - D_E)^{-1}$ all exist, therefore the above equation holds. Substitute (24) into (23) to give

$$\begin{aligned}\Delta u_M^{dis}(k) &= \frac{1}{2}\Delta u_M^*(k) - \frac{1}{4}(I - D_E)^{-1}\frac{\Delta u_M^*(k)}{\frac{1}{2}(I - D_E)^{-1} + (D_E + D_E T - I)^{-1}} \\ &= S\Delta u_M^*(k)\end{aligned} \tag{25}$$

with

$$S = \frac{1}{2}I - \frac{1}{4}(I - D_E)^{-1}\left[\frac{1}{2}(I - D_E)^{-1} + (D_E + D_E T - I)^{-1}\right]^{-1}.$$

From $\Delta u_M^*(k) = (I - D_E)^{-1}D_1[\varpi(k) - \tilde{y}_{P0}(k)]$, it has

$$\varpi(k) - \tilde{y}_{P0}(k) = D_1^{-1}(I - D_E)\Delta u_M^*(k).$$

Then it gives

$$\begin{aligned}J^* &= \|\varpi(k) - \tilde{y}_{P0}(k) - A\Delta u_M^*(k)\|_Q^2 + \|\Delta u_M^*(k)\|_R^2 \\ &= \|D_1^{-1}(I - D_E)\Delta u_M^*(k) - A\Delta u_M^*(k)\|_Q^2 + \|\Delta u_M^*(k)\|_R^2 = \|\Delta u_M^*(k)\|_F^2\end{aligned} \tag{26}$$

with $F = [D_1^{-1}(I - D_E) - A]^T Q[D_1^{-1}(I - D_E) - A] + R$, let

$$\overline{A} = \begin{bmatrix} A_{11} & & \\ & \ddots & \\ & & A_{NN}\end{bmatrix}.$$

Then the prediction model of the whole distributed system under the column disturbance can be written as

$$\tilde{y}_M^{dis}(k) = \tilde{y}_{P0}(k) + (\overline{A} + G - GT)\Delta u_M^{dis}(k) = \tilde{y}_{P0}(k) + L\Delta u_M^{dis}(k), \tag{27}$$

where $L = \overline{A} + G - GT$.

Substitute (25) and (27) into (9), it can be derived

$$\begin{aligned}J^{dis} &= \|\varpi(k) - \tilde{y}_{P0}(k) - LS\Delta u_M^*(k)\|_Q^2 + \|S\Delta u_M^*(k)\|_R^2 \\ &= \|\varpi(k) - \tilde{y}_{P0}(k) - A\Delta u_M^*(k) + (A - LS)\Delta u_M^*(k)\|_Q^2 \\ &\quad + \|\Delta u_M^*(k) + (S - I)\Delta u_M^*(k)\|_R^2 \\ &= J^* + \|\Delta u_M^*(k)\|_W^2,\end{aligned} \tag{28}$$

where

$$W = (A - LS)^T Q(A - LS) + (S - I)^T R(S - I) + R(S - I) \\ + (S - I)^T R + (M - A)^T Q(A - LS) + (A - LS)^T Q(M - A).$$

Let $t_W(W)$ denotes the norm of W, it gives

$$\|\Delta u_M^*(k)\|_W^2 \le \Delta u_M^{*T}(k)\|W\|\Delta u_M^*(k) = t_W(W)\|\Delta u_M^*(k)\|^2 \\ \le \frac{t_W(W)}{\lambda_m(F)}\|\Delta u_M^*(k)\|_F^2 = \frac{t_W(W)}{\lambda_m(F)} J^*.$$

Here $\lambda_m(F)$ is the minimal eigen value of F. From the above derivations, the performance relationship between the free disturbance and disturbance can be expressed as

$$J^{dis} \le J^* + \frac{t_W(W)}{\lambda_m(F)} J^* = (1+\delta)J^*. \tag{29}$$

Inspection of (29), that $t_W(W)$ depends on G'' and D_E'', while G'' and D_E'' are affected by disturbance matrix T. So in case of free disturbance, $t_W(W)$ can arrive at the maximal value, at this time, $\|T\| = 0$, $G'' = 0$, $D_E'' = 0$, $L = \overline{A} + G$, $S = I$ and

$$W = W_{\max} = (A - \overline{A} - G)^T Q(A - \overline{A} - G) + [D_1^{-1}(I - D_E) - A]^T Q(A - \overline{A} - G) \\ + (A - \overline{A} - G)^T Q(D_1^{-1}(I - D_E) - A).$$

Therefore the upper bound of the performance deviation under the local communication disturbance is

$$\delta_{\max} = \frac{t_W(W)}{\lambda_m(F)}.$$

Theorem 5.2 *The convergence condition of the distributed linear model predictive control system under the communication disturbance is $|\rho(D_E)| < 1$. D_E is the same as defined before. This proof is similar to the analysis in Section 3.*

Remark 5.2 Under the communication disturbance, each agent cannot exchange information properly. In an extreme case the elements in matrix E are all 1s, D_E becomes null matrix, $|\rho(D_E)| < 1$ is always satisfied, which corresponds to the full decentralised architecture.

6 Simulation Study

Consider a linear continuous time dynamic plant model with three inputs and three outputs

$$G(s) = \begin{bmatrix} \frac{e^{-2s}}{100s+1} & \frac{e^{-6s}}{100s+1} & \frac{e^{-4s}}{200s+1} \\ \frac{-1.25e^{-2s}}{50s+1} & \frac{3.75e^{-6s}}{50s+1} & \frac{e^{-3s}}{50s+1} \\ \frac{-2e^{-2s}}{200s+1} & \frac{2e^{-4s}}{200s+1} & \frac{3.5e^{-2s}}{100s+1} \end{bmatrix}.$$

The expected output of this system is to follow a set of unit step reference signals. With the proposed distributed algorithm, first of all divide the whole system into three agents, they are

Agent 1: $G_1(x) = \frac{e^{-2s}}{100s+1}$.
Agent 2: $G_2(x) = \frac{3.75e^{-6s}}{50s+1}$.
Agent 3: $G_3(x) = \frac{3.5e^{-3s}}{100s+1}$.

The control parameters for each agent are set with $P = 8$, $M = 3$, $Q_i = I$, $R_i = 0.5I$, $(i = 1, 2, 3)$, sampling time of 20 sec, and $\varepsilon_i = 0.01$ $(i = 1, 2, 3)$. The Matlab based simulation results are shown in Figure 6.1. It can be observed that each agent in this distributed structure can properly arrive at the expected outputs while keeping the satisfactory performance to some extent. In addition, the design parameters for each agent such as prediction horizon, control horizon, weighting matrix and sample time etc. can all be designed and tuned separately, which is superior to the centralised control and significantly reduce the on-line computational burden. Notice that each agent is not necessary limited to SISO case and also it can be MIMO agent, whose dimension is still much lower than the whole system's.

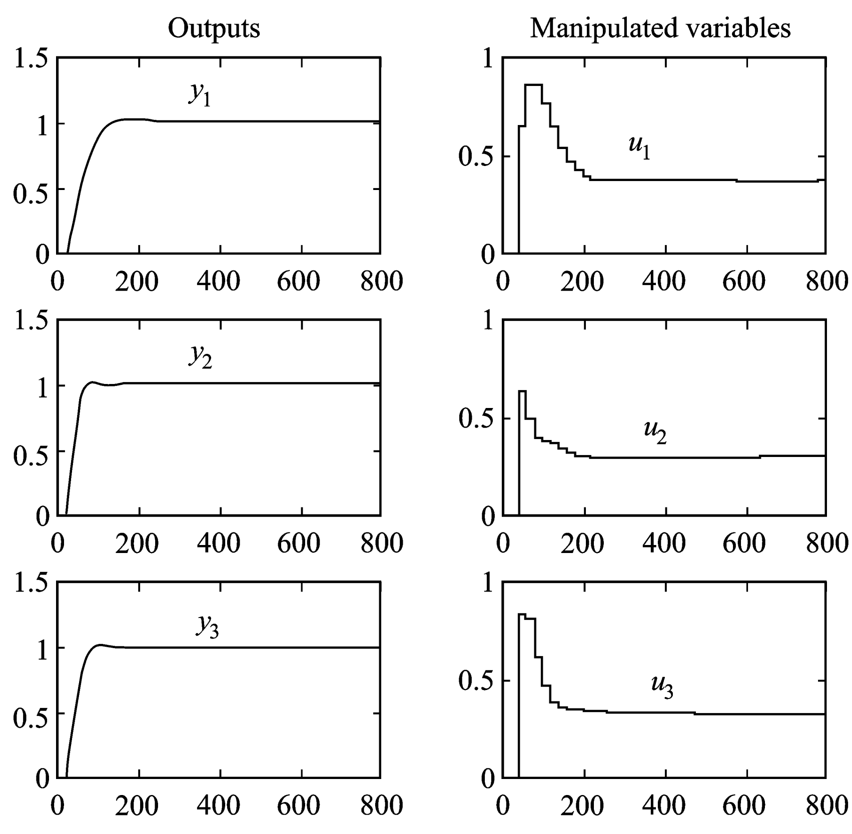

Figure 6.1. Output responses and manipulated/control signals on the experimental plant.

7 Conclusions

In this study a distributed model predictive control method based on Nash optimality is developed for large-scale linear systems. To avoid the prohibitively high on-line computational demand, the MPC is implemented in distributed scheme with the inexpensive agents within the network environment. These agents can co-operate and communicate each other to achieve the objective of the whole system. Coupling effects between the agents are fully taken into account in this scheme, which is superior to other traditional decentralised control methods. The main advantage of this scheme is that the on-line optimisation of a large-scale system can be converted to that of several small-scale systems, thus can significantly reduce the computational complexity while keeping satisfactory performance. In addition, the design parameters for each agent such as prediction horizon, control horizon, weighting matrix and sample time etc. can all be designed and tuned separately, which provides more flexibility for the analysis and applications. The second part of this study is to investigate the performance of the distributed control scheme. The nominal stability and the performance deviation on the single-step horizon under the communication disturbance are analysed. These will provide users better understanding of the developed algorithm and sensible guidance in applications. As the method is also expandable to complex large-scale nonlinear model predictive control with certain constraints, a further study to control nonlinear Hammertien models is under investigation.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (under Grant: 60074004), and High Technology Research and Development Program of China (under Grant: 2002AA412130).

References

[1] Clark, D.W., Mohtadi, C. and Tuffs, P.S. General predictive control. *Automatica* **23**(3) (1987) 137–160.

[2] Du, X.N., Xi, Y.G. and Li, S.Y. Distributed model predictive control for large-scale systems. *Proceedings of American Control Conference, Arlington, Virginia*, 2001, 3142–3143.

[3] Gurfil, P. and Kasdin, N. J. Characterization and design of out-of-ecliptic trajectories using deterministic crowding genetic algorithms. *Computer Methods in Applied Mechanics and Engineering* **191**(19–20) (2002) 2141–2158.

[4] Katebi, M.R. and Johnson, M.A. Predictive controldesign for large-scale systems. *Automatica* **33**(3) (1997) 421–425.

[5] Mosca, E. *Optimal, Predictive and Adaptive Control*. Prentice Hall, Englewood Cliffs, New Jersey, 1995.

[6] Nash, S.G. A survey of truncated-Newton methods. *J. of Computational and Applied Mathematics* **124** (2000) 45–59.

[7] Qin, S.J. and Badgwell, T.A. An overview of industrial model predictive control technology. In: *Chemical process control-assessment and new directions for research*, (Eds.: Y.C. Kantor, C.E. Garcia and B.C. Garcia), AIChE Symposium series, AIChE and CACHE, **93**(316) (1997) 232-256.

[8] Richalet, J. Industrial applications of model based predictive control. *Automatica* **29**(5) (1993) 1251–1274.

[9] Ricker, N.L. and Lee, J.H. Nonlinear model predictive control of the Tennessee Eastman challenge process. *Computer and Chemical Engineering* **19**(9) (1995) 961–981.

[10] Van Antwerp, J.G. and Braatz, R.D. Model predictive control of large scale processes. *Journal of Process Control* **10**(1) (2000) 1–8.

[11] Xi, Y.G. New design method for discrete-time multi-variable predictive controllers. *Int. J. Control* **49**(1) (1989) 45–56.

[12] Xi, Y.G. *Predictive Control*. National defense and industry press, Beijing, 1993.

[13] Xu, X. M., Xi, Y.G. and Zhang, Z.J. Decentralised predictive control (DPC) of large-scale systems. *Information and Decision Technologies* **14**(3) (1988) 307–322.

[14] Zheng, A. A computationally efficient nonlinear MPC algorithm. *Proceedings of the American Control Conference, Albuquerque, New Mexico*, 1997, 1623–1627.

[15] Zheng, A. Reducing on-line computational demands in model predictive control by approximating QP constraints. *Journal of process control* **9**(4) (1999) 279–290.

[16] Zheng, A. and Allgower, F. Towards a practical nonlinear predictive control algorithm with guaranteed stability for large-scale systems. *Proceedings of the American Control Conference, Philadelphia, Pennsylvania*, 1998, 2534–2538.

[17] Zhu, Q.M., Warwick, K. and Douce, J.L. Adaptive general predictive controller for nonlinear systems. *IEE Proc. – Control Theory Appl.* **138**(1) (1991) 33–40.