Hindawi Publishing Corporation
Journal of Control Science and Engineering
Volume 2016, Article ID 8167931, 14 pages
http://dx.doi.org/10.1155/2016/8167931

Research Article

Improved Distributed Model Predictive Control with Control Planning Set

Wei Chen

Department of Automation, Hefei University of Technology, Hefei 230009, China

Correspondence should be addressed to Wei Chen; windysunny0531@gmail.com

Received 6 June 2016; Revised 16 August 2016; Accepted 30 August 2016

Academic Editor: Yuanyuan Zou

We focus on distributed model predictive control algorithm. Each distributed model predictive controller communicates with the others in order to compute the control sequence. But there are not enough communication resources to exchange information between the subsystems because of the limited communication network. This paper presents an improved distributed model predictive control scheme with control planning set. Control planning set algorithm approximates the future control sequences by designed planning set, which can reduce the exchange information among the controllers and can also decrease the distributed MPC controller calculation demand without degrading the whole system performance much. The stability and system performance analysis for distributed model predictive control are given. Simulations of the four-tank control problem and multirobot multitarget tracking problem are illustrated to verify the effectiveness of the proposed control algorithm.

1. Introduction

Model predictive control (MPC), also referred to receding horizon control (RHC), is an attractive control strategy because of its ability to control systems with input and output constraints in the optimization problem. The input sequence is calculated by solving an optimization problem (minimization of a given performance index) over a prediction horizon. Once the optimization problem is solved, only the first input value is implemented into the system. In the next sampling time, a new optimization problem is solved repeatedly. MPC has been widely applied in various control areas over the past few decades [1–3].

Nowadays, systems are becoming more and more complex. In centralized MPC, all the inputs sequences are optimized with respect to one given performance index in a single optimization problem. However, when the number of the state variables and inputs of the system becomes larger and larger, the computation burden of the centralized optimization problem may increase significantly. Moreover, the entire system would be out of control if the centralized MPC controller fails. Therefore it is impractical to apply the centralized MPC to large-scale systems. In fact, a large-scale system is composed by physically parted subsystems. Many decentralized and distributed model predictive control (DMPC) algorithms have been recently proposed [4–7], which are some feasible alternatives to overcome the computational burden of the centralized MPC.

In DMPC architecture, subsystems communicate with each other via networks and the inputs are computed by solving more than one optimization problem in each subsystem in a coordinated fashion. There are many achievements on DMPC strategy and a survey of major DMPC algorithms is presented in [8, 9]. The existing DMPC algorithms can be divided into different categories.

Based on the topology of the communication network, DMPC can be divided into fully connected algorithms and partially connected algorithms. In fully connected algorithms, DMPC is able to communicate with the rest of the local controllers [10, 11]. In partially connected algorithms, local optimization problems are solved by taking into account the neighboring (not the whole system) interaction and solution, which is suitable for loosely connected subsystems [12, 13]. However, it will deteriorate the whole system performance.

Based on the exchange times among the distributed controllers, DMPC can be divided into noniterative algorithms and iterative algorithms. In iterative algorithms, information is transmitted among the DMPC controllers many times in the sampling interval [14, 15]. On the contrary, in noniterative algorithms DMPC controller communicates with the other controllers only once in the sampling interval [16, 17].

In this article, we consider that the DMPC controllers can exchange information only once while they are solving their local optimization problems at each sampling time and the connectivity of the communication is sufficient for the distributed controller to obtain information. This paper proposes an extension of the fully connected noniterative DMPC algorithm. However, the exchange information between subsystems is usually realized over a digital communication network. Thus, the local systems can only have limited communication resource. For example, in a networked environment, bandwidth limitations can restrict the amount of exchange information. Thus, it is necessary to restrict the distributed controllers to exchange information. The proposed DMPC in the paper reduces the communication information compared to the standard distributed MPC control scheme in complex large-scale systems and at the same time decreases computational burden of each controller. This algorithm also provides a reasonable trade-off between system performance and low communication requirements needed to reach a cooperative solution.

The rest of the paper is organized as follows. In Section 2, the centralized and distributed model predictive control problem is formulated. In Section 3, the improved distributed model predictive control with control planning set (CP-DMPC) is proposed. The stability and performance analysis is provided in Section 4. In Section 5, the simulations of the proposed controller to four-tank system and multirobot multitarget tracking system are presented. Finally, the conclusions of the work are given in Section 6.

2. Centralized and Distributed Model Predictive Control Formulation

Without loss of generality, suppose that the whole system is comprised of N interconnected subsystems. And consider that each subsystem only couples through the input [18]. The discrete-time state-space model for ith subsystem is as follows:

$$x_{m,i}(k+1) = A_{m,i}x_{m,i}(k) + B_{m,ii}u_i(k) + \sum_{j=1, j\neq i}^{N} B_{m,ij}u_j(k), \tag{1a}$$

$$y_i(k) = C_{m,i}x_{m,i}(k), \tag{1b}$$

where $i = 1, \ldots, N$. $x_{m,i}(k)$, $u_i(k)$, and $y_i(k)$ are the state vector, the control input vector, and the output vector of ith subsystem at kth sampling time. The model (1a), (1b) is changed to suit the model predictive control design with an embedded integrator. The augmented model of the ith subsystem state space model is

$$x_i(k+1) = A_ix_i(k) + B_{ii}\Delta u_i(k) + w_i(k), \tag{2a}$$

$$y_i(k) = C_ix_i(k), \tag{2b}$$

where a new state variable vector is chosen to be

$$x_i(k) = \left[\Delta x_{m,i}(k) \quad y_i(k)\right] \tag{3}$$

and a new control variable vector is chosen to be

$$\Delta u_i(k) = u_i(k) - u_i(k-1) \tag{4}$$

and the difference of the state variable is denoted by

$$\Delta x_{m,i}(k+1) = x_{m,i}(k+1) - x_{m,i}(k). \tag{5}$$

The state interaction vector is given by

$$w_i(k) = \sum_{j=1, j\neq i}^{N} B_{ij}\Delta u_j(k). \tag{6}$$

The triplet $A_i, [B_{ii}, B_{ij}],\ C_i$ is

$$\begin{gathered} A_i = \begin{bmatrix} A_{m,i} & O \\ C_{m,i}A_{m,i} & I \end{bmatrix}, \\ B_{ii} = \begin{bmatrix} B_{m,ii} \\ C_{m,i}B_{m,ii} \end{bmatrix}, \\ B_{ij} = \begin{bmatrix} B_{m,ij} \\ C_{m,i}B_{m,ij} \end{bmatrix}, \\ C_i = \left[O \quad I\right]. \end{gathered} \tag{7}$$

The model of the whole system (centralized model) can be expressed in compact way

$$x(k+1) = Ax(k) + B\Delta u(k), \tag{8a}$$

$$y(k) = Cx(k) \tag{8b}$$

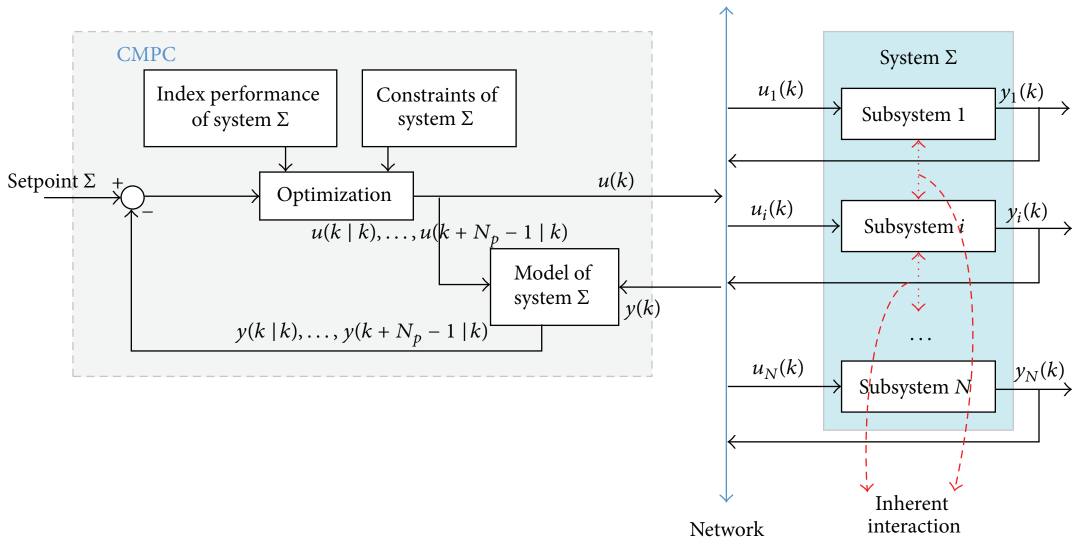


Figure 1: Centralized MPC control system architecture.

with state vector $x(k) \in R^{n_x}$, control input vector $\Delta u(k) \in R^{n_u}$, and output vector $y(k) \in R^{n_y}$. A, B, and C are the whole system matrices. This implies that

$$A = \begin{bmatrix} A_1 & & & & \\ & \ddots & & & \\ & & A_i & & \\ & & & \ddots & \\ & & & & A_N \end{bmatrix},$$

$$B = \begin{bmatrix} B_{11} & \cdots & B_{1j} & \cdots & B_{1N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ B_{i1} & \cdots & B_{ii} & \cdots & B_{iN} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ B_{N1} & \cdots & B_{Nj} & \cdots & B_{NN} \end{bmatrix},$$

$$C = \begin{bmatrix} C_1 & & & & \\ & \ddots & & & \\ & & C_i & & \\ & & & \ddots & \\ & & & & C_N \end{bmatrix}, \tag{9}$$

$$x(k) = [x_1(k), x_2(k), \ldots, x_N(k)]^T,$$

$$\Delta u(k) = [\Delta u_1(k), \Delta u_2(k), \ldots, \Delta u_N(k)]^T,$$

$$y(k) = [y_1(k), y_2(k), \ldots, y_N(k)]^T.$$

2.1. Centralized Model Predictive Control Formulation. The main idea of the centralized model predictive control formulation is one large-scale optimization with constraint. The centralized MPC control system architecture diagram is shown in Figure 1.

In the centralized model predictive control formulation, at each sampling time centralized MPC controller obtains the whole system measurement $y(k) = [y_1(k), y_2(k), \ldots, y_N(k)]$ and the control objective minimizes the following global performance index:

$$J(k) = \sum_{i=1}^{N} J_i(k),$$

$$J_i(k) = \sum_{l=1}^{N_p} \left\| y_i(k+l \mid k) - y_i^d(k+l) \right\|_{Q_i}^2 + \sum_{l=1}^{N_u} \left\| \Delta u_i(k+l-1 \mid k) \right\|_{R_i}^2 \tag{10a}$$

s.t.

$$\begin{aligned} x_i(k+l+1 \mid k) &= A_i x_i(k+l \mid k) + B_{ii} \Delta u_i(k+l \mid k) + w_i(k+l \mid k), \\ y_i(k+l \mid k) &= C_i x_i(k+l \mid k) \\ & i = 1, \ldots, N. \end{aligned} \tag{10b}$$

Here N_p is the prediction horizons and N_u is the control horizons. And $N_p \geq N_u$. Q_i and R_i are penalties on the output variables and control variables, respectively. y_i^d is the output set point. And because the central controller can handle all the information of the system, the interaction predictions $w_i(k+l \mid k)$ are known at time k.

This optimization problem (10a), (10b) can be solved by a standard quadratic program algorithm with constraints. The optimal control sequence $\Delta U^*(k, N_u \mid k) = [\Delta u^*(k \mid k),$

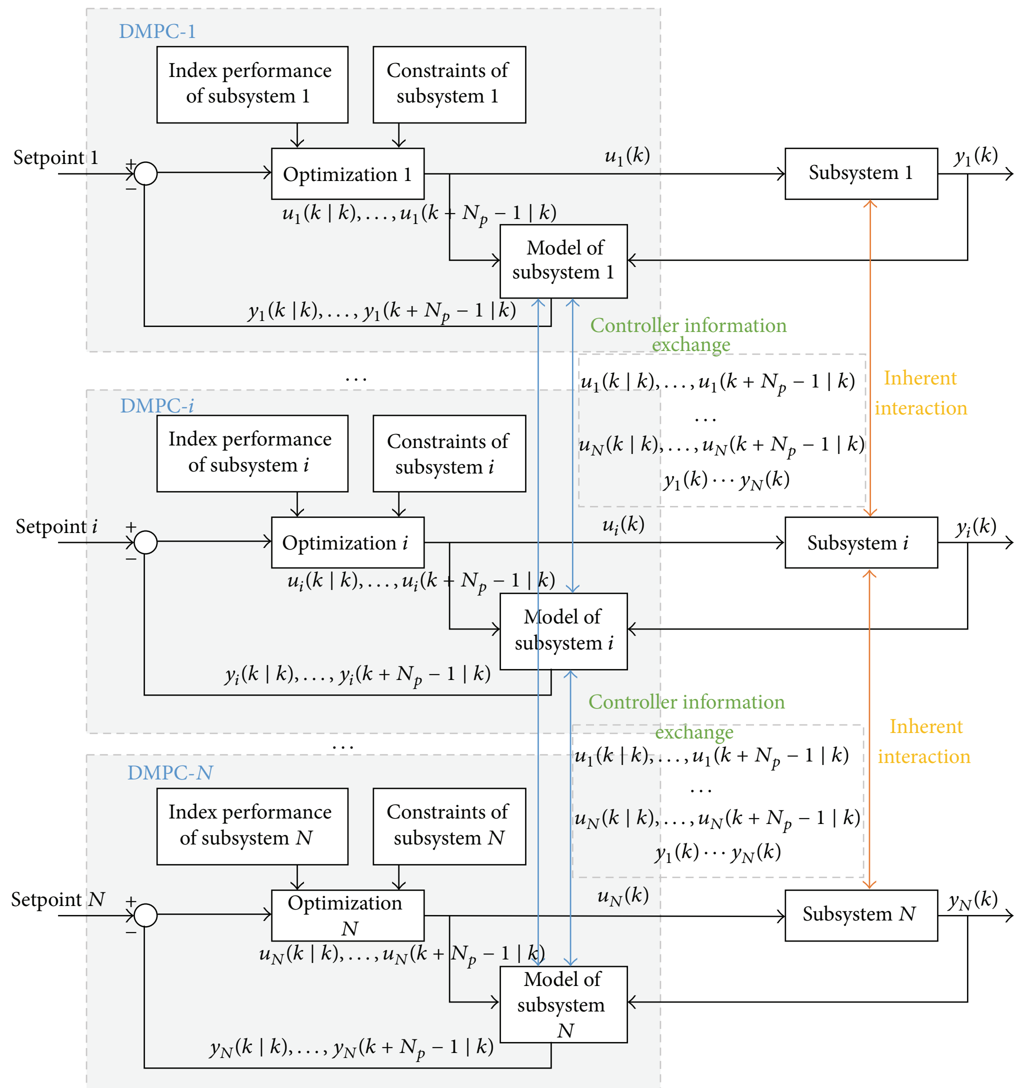


Figure 2: DMPC control system architecture.

$\Delta u^*(k+1 \mid k), \ldots, \Delta u^*(k+N_u-1 \mid k)]$ is calculated and only the first control signal $\Delta u^*(k \mid k) = [\Delta u_1^*(k \mid k), \Delta u_2^*(k \mid k), \ldots, \Delta u_N^*(k \mid k)]$ is applied to the whole system; after new measurements are available, a new optimization problem is solved in the next sampling time.

Many engineering applications such as power systems, unmanned aerial vehicles, sensor networks, economic system, transportation systems, and process control systems, have become larger and more complex. The overall number of inputs and states (outputs) is very large, and the optimized control sequence $\Delta U^*(k, N_u \mid k)$ is highly dimensional. A single optimization problem may require computational resources (CPU time, memory, etc.). In view of the above consideration, it is natural to look for distributed MPC algorithms.

2.2. Distributed Model Predictive Control Formulation. In the distributed model predictive control formulation, the large size optimization problem is replaced by N small ones that work cooperatively towards achieving the performance of centralized control system. And the following assumptions are made.

(a) Predictive horizons N_p and control horizons N_u are the same for each subsystem.

(b) Controllers are synchronous.

(c) Controllers communicate with each other only once within a sampling time interval.

(d) Controllers are interconnected and can obtain information which the controllers need.

And the DMPC control system architecture diagram is shown in Figure 2.

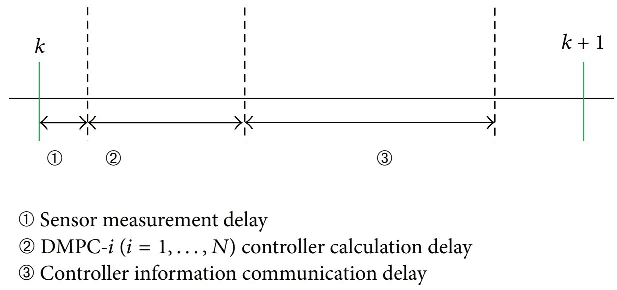


Figure 3: Delay time analysis per sampling interval.

The ith subsystem minimizes the following local performance index, which is the ith optimization problem [19]:

$$J_i(k) = \sum_{l=1}^{N_p} \left\| y_i(k+l \mid k) - y_i^d(k+l) \right\|_{Q_i}^2 + \sum_{l=1}^{N_u} \left\| \Delta u_i(k+l-1 \mid k) \right\|_{R_i}^2 \tag{11a}$$

s.t.

$$\begin{aligned} x_i(kk+l+1) &= A_i x_i(kk+l) + B_{ii} \Delta u_i(kk+l) \\ &\quad + w_i(kk+-l1), \\ y_i(kk+l) &= C_i x_i(kk+l). \end{aligned} \tag{11b}$$

It can be seen that the global performance index can be decomposed into a number of local performance indexes, but the output of each agent is still related to all the input variables due to the input coupling. Because controllers communicate with each other only once within a sampling time interval, the interaction predictions $w_i(k+l \mid k)$ are unknown for the ith subsystem. And only the prediction $w_i(k+l \mid k-1)$ based on the information broadcasted at time $k-1$ is available. A noniterative algorithm is developed to seek the distributed solution at each sampling time. Based on the information from other subsystems, each controller solves local optimization problems to determine the future sequence $\Delta U_i^*(k, N_u \mid k) = [\Delta u_i^*(k \mid k), \Delta u_i^*(k+1 \mid k), \ldots, \Delta u_i^*(k+N_u-1 \mid k)]$ and broadcast $\Delta U_i^*(k \mid k)$ by communication network to the other controllers.

3. Improved Distributed Model Predictive Control with Control Planning Set (CP-DMPC)

Besides the computational advantages of DMPC, the amount of data needs to be exchanged among distributed controllers. In the paper, fully connected noniterative DMPC algorithm is focused on. However, each system exchanges information with each other by both their initial state and their optimized input. And time delays exist in communication network. In Figure 3, we can see that time delay consists of three parts, sensor measurement delay, DMPC controller calculation delay, and controller information communication delay.

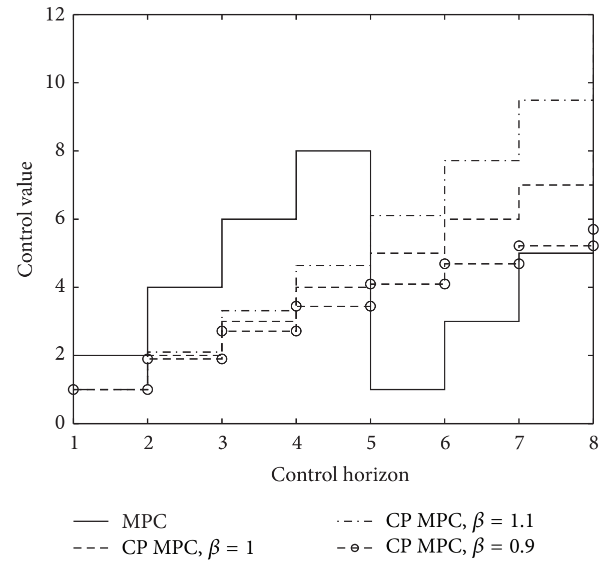


Figure 4: The comparison between traditional MPC and CP_MPC.

In this paper, a control planning set algorithm is combined with DMPC controller to reduce the controller information communication delay and meanwhile it also can decrease the DMPC controller calculation demand without degrading the whole system performance much. The control planning set method presented in the paper is inspired by the pulse-step control strategy [20]. Suboptimal strategies can be obtained by restricting the future control sequence

$$\Delta u(k+l \mid k) = f(\Delta u(k+l-1 \mid k)). \tag{12}$$

For specification and simplicity, we choose function f as a linear function:

$$\Delta u(k+l \mid k) = \beta \Delta u(k+l-1 \mid k). \tag{13}$$

In the control planning set algorithm, the future control sequence is restricted by one possibility. The parameter β is chosen to plan the future control sequence increases or decreases in the same direction, which is suitable for the experience of control engineering. And it will prevent the frequent oscillation of the control input; see Figure 4.

In a traditional MPC scheme, the optimized control sequence is calculated via the performance index, which may oscillate during the control horizon. In CP_MPC scheme, the optimized control sequence changes in one direction, which may not obtain the optimum solution but is suitable for the control engineering. In control engineering, in some time period control value does not change suddenly and frequently, and this is good for the control hardware device.

If $\beta = 1$, the control sequence is set in equal increase. If $\beta > 1$, the weight of the future control is larger than that of the current control.

Let one assume that

$$\widetilde{B}_i = [B_{i1}, \ldots, B_{ii-1}, O, B_{ii+1}, \ldots, B_{iN}], \tag{14a}$$

$$\Gamma_i = \begin{bmatrix} I_{n_{ui}\times n_{ui}} & & \\ & \ddots & \\ & & I_{n_{ui}\times n_{ui}} \\ O_{(N_p-N_u)\times n_{ui}} & \cdots & O_{(N_p-N_u)\times n_{ui}} \end{bmatrix}^T, \tag{14b}$$

$$\overleftrightarrow{B}_i = \operatorname{diag}_{N_p}\left(\widetilde{B}_i\right)\Gamma_i, \tag{14c}$$

$$\overleftrightarrow{B} = \left[\overleftrightarrow{B}_i, \ldots, \overleftrightarrow{B}_i, \ldots, \overleftrightarrow{B}_N\right], \tag{14d}$$

$$E_i = \left[I_{n_{ui}\times n_{ui}} \quad \beta_i I_{n_{ui}\times n_{ui}} \quad \cdots \quad \beta_i^{N_u-1} I_{n_{ui}\times n_{ui}}\right]^T, \tag{14e}$$

$$E = \operatorname{diag}\{E_1, E_2, \ldots, E_N\}, \tag{14f}$$

$$S_i = \left[(A_i)^T \quad \left(A_i^2\right)^T \quad \cdots \quad \left(A_i^{N_p}\right)^T\right]^T, \tag{14g}$$

$$S = \operatorname{diag}\{S_1, S_2, \ldots, S_N\}, \tag{14h}$$

$$T_i = \begin{bmatrix} A_i^0 & & 0 \\ \vdots & \ddots & \\ A_i^{N_p-1} & \cdots & A_i^0 \end{bmatrix}, \tag{14i}$$

$$T = \operatorname{diag}\{T_1, T_2, \ldots, T_N\}, \tag{14j}$$

$$\overline{B}_i = \operatorname{diag}_p\{B_{ii}, \ldots, B_{ii}\}\Gamma_i, \tag{14k}$$

$$\overline{B} = \operatorname{diag}\left\{\overline{B}_1, \overline{B}_2, \ldots, \overline{B}_N\right\}. \tag{14l}$$

Lemma 1. *The interaction predictions of ith subsystem at time k are given by*

$$W_i\left(k, N_p \mid k-1\right) = \overleftrightarrow{B}_i E_i \Delta U(k \mid k-1) \tag{15}$$

and the compact predictions have the following form:

$$W\left(k, N_p \mid k-1\right) = \overleftrightarrow{B} E \Delta U(k \mid k-1). \tag{16}$$

Proof. With (6) and (13), the prediction of the interaction vectors of time k is given by

$$\begin{aligned}
w_i(k \mid k-1) &= \sum_{j=1, j\neq i}^{N} B_{ij}\Delta u_j(k \mid k-1) \\
&= \widetilde{B}_i \Delta U(k \mid k-1) \\
w_i(k+1 \mid k-1) &= \sum_{j=1, j\neq i}^{N} B_{ij}\Delta u_j(k+1 \mid k-1) \\
&= \sum_{j=1, j\neq i}^{N} B_{ij}\beta\Delta u_j(k \mid k-1) = \beta_i \widetilde{B}_i \Delta U(k \mid k-1) \\
&\vdots \\
w_i(k+N_u \mid k-1) &= \sum_{j=1, j\neq i}^{N} B_{ij}\Delta u_j(k+N_u \mid k-1) \\
&= \sum_{j=1, j\neq i}^{N} B_{ij}\beta_i^{N_u-1}\Delta u_j(k \mid k-1) \\
&= \beta_i^{N_u-1}\widetilde{B}_i \Delta U(k \mid k-1)\, w_i(k+N_u+1 \mid k-1) \\
&= \sum_{j=1, j\neq i}^{N} B_{ij}\Delta u_j(k+N_u+1 \mid k-1) = 0 \\
&\vdots \\
w_i\left(k+N_p \mid k-1\right) \\
&= \sum_{j=1, j\neq i}^{N} B_{ij}\Delta u_j(k+N_u-1 \mid k-1) = 0.
\end{aligned} \tag{17}$$

By definitions (14a)–(14l), this implies the relations (15) and the equivalent compact forms (16) hold. □

Lemma 2. *The state and output predictions of ith subsystem at time k are expressed by*

$$\begin{aligned}
X_i\left(k+1, N_p \mid k\right) &= S_i x_i(k \mid k) + T_i \overline{B}_i E_i \Delta u_i(k) \\
&\quad + T_i \overleftrightarrow{B}_i E \Delta U(k \mid k-1), \\
Y_i\left(k+1, N_p \mid k\right) &= C_i X_i\left(k+1, N_p \mid k\right)
\end{aligned} \tag{18}$$

and the compact predictions have the following form:

$$\begin{aligned}
X\left(k+1, N_p \mid k\right) &= Sx(k \mid k) + T\overline{B}E\Delta u(k) \\
&\quad + T\overleftrightarrow{B} E \Delta U(k \mid k-1), \\
Y\left(k+1, N_p \mid k\right) &= CX\left(k+1, N_p \mid k\right).
\end{aligned} \tag{19}$$

(1) Set initial parameter values
(2) Repeat
(3) ith CP-DMPC controller receives the output measurement $y_i(k)$ from the sensors, $i = 1, \ldots, N$.
(4) Obtain the control input $\Delta u_j(k \mid k-1)$ and control index parameters β_j $j = 1, \ldots, i-1, i+1, \ldots, N$ from the other CP-DMPC controllers.
(5) Compute the predictions of the interaction $W_i(k, N_p \mid k-1)$.
(6) Compute the optimal control input $\Delta u_i^*(k \mid k)$ and broadcast it by the communication network.
(7) Apply the control input $\Delta u_i^*(k)$ into each subsystem.
(8) until the control procedure ends

Algorithm 1: Algorithm CP-DMPC: CP-DMPC in a pseudo-algorithm format.

Proof. With (2a), (2b), and (13), the state and output predictions of ith subsystem at time k are expressed by

$$\begin{aligned}
x_i(k+1 \mid k) &= A_i x_i(k \mid k) + B_{ii}\Delta u_i(k \mid k) \\
&\quad + \sum_{j=1, j\neq i}^{N} B_{ij}\Delta u_j(k \mid k) \\
x_i(k+2 \mid k) &= A_i x_i(k+1 \mid k) + B_{ii}\Delta u_i(k+1 \mid k) \\
&\quad + \sum_{j=1, j\neq i}^{N} B_{ij}\Delta u_j(k+1 \mid k) = A_i^2 x_i(k \mid k) + [A_i B_{ii} \\
&\quad + B_{ii}\beta]\Delta u_i(k \mid k) + \sum_{j=1, j\neq i}^{N} [A_i B_{ij} + B_{ij}\beta] \\
&\quad \cdot \Delta u_j(k \mid k) \\
&\vdots \\
x_i(k+N_p \mid k) &= A_i^{N_p} x_i(k \mid k) + \left[A_i^{N_p-1} B_{ii} + \cdots \right. \\
&\quad \left. + A_i^{N_p-N_u} B_{ii}\beta^{N_u-1}\right]\Delta u_i(k \mid k) \\
&\quad + \sum_{j=1, j\neq i}^{N} \left[A_i^{N_p-1} B_{ij} + \cdots + A_i^{N_p-N_u} B_{ij}\beta^{N_u-1}\right] \\
&\quad \cdot \Delta u_j(k \mid k).
\end{aligned} \tag{20}$$

By definitions (14a)–(14l), this implies the relations (18) and the equivalent compact forms (19) hold. □

Remark 3. There are three parts in the state (output) predictions of ith subsystem $X_i(k+1, N_p \mid k)$. The first part is $S_i x_i(k \mid k)$, which can be obtained by the current state value. The second part $T_i \overline{B}_i E_i \Delta u_i(k)$ is the interaction item between ith subsystem and $\bar{i}$th system ($\bar{i} = \{1, \ldots, i-1, i+1, \ldots, N\}$). And the last part $T_i \overleftrightarrow{B}_i E \Delta U(k \mid k-1)$ is the future optimization item.

Lemma 4. *The ith subsystem at time k has to solve the following optimization problem:*

$$\begin{aligned}
J_i &= -G_i^T(k+1, N_p \mid k)\Delta U_i \\
&\quad + \Delta U_i^T\left(\Phi_{ii}^T Q_i \Phi_{ii} + R_i\right)\Delta U_i,
\end{aligned} \tag{21}$$

where $\Phi_{ii} = T_i \overline{B}_i E_i$, $G_i(k+1, N_p \mid k) = 2\Phi_{ii}^T Q_i(Y_i^d - S_i x_i(k \mid k) - T_i W_i(k, N_p \mid k-1))$.

Proof. Using the local performance index (11a), the cost function can be written in the equivalent form

$$J_i = \left(Y_i^d - Y_i\right)^T Q_i \left(Y_i^d - Y_i\right) + \Delta U_i^T R_i \Delta U_i. \tag{22}$$

Applying (18) into it, the local performance index J_i takes the form (21). □

Theorem 5. *For ith subsystem, the explicit form of the control law is given by*

$$\begin{aligned}
&\Delta u_i(k \mid k) \\
&\quad = \overline{K}_i\left(Y_i^d - S_i x_i(k \mid k) - T_i W_i\left(k, N_p \mid k-1\right)\right).
\end{aligned} \tag{23}$$

And the compact expression is

$$\begin{aligned}
\Delta U(k \mid k) &= \Xi Y^d + \Theta x(k \mid k) \\
&\quad + \Psi \Delta U(k-1 \mid k-1),
\end{aligned} \tag{24}$$

where $\overline{K}_i = (\Phi_{ii}^T Q_i \Phi_{ii} + R_i)^{-1}\Phi_{ii}^T Q_i$, $\Xi = \mathrm{diag}\{\overline{K}_1, \ldots, \overline{K}_N\}$, $\Theta = -\Xi S$, $\Psi = \Xi T \overleftrightarrow{B} E$

The distributed MPC algorithm with control planning set (CP-DMPC) can be summarized as shown in Algorithm 1.

4. Stability and Performance Analysis

4.1. Stability Analysis. We provide sufficient conditions that guarantee practical stability of the closed-loop system.

Theorem 6. *The closed-loop system with N subsystems is asymptotically stable if and only if*

$$\lambda\left\{\begin{bmatrix} A & 0 & B & 0 \\ S & 0 & T\overline{B}E & T\overleftrightarrow{B}E \\ \Theta A & 0 & \Theta B+\Psi & 0 \\ 0 & 0 & I & 0 \end{bmatrix}\right\}<1. \quad (25)$$

Proof. Combining the process (8a) and (8b) and control law (23), the closed-loop state-space representation is derived:

$$\begin{aligned} x(k) &= Ax(k-1) \\ &\quad + B\Delta u(k-1\mid k-1), \\ X\left(k,N_p\mid k-1\right) &= Sx(k-1) \\ &\quad + T\overline{B}E\Delta U(k-1\mid k-1) \\ &\quad + T\overleftrightarrow{B}E\Delta U(k-2\mid k-2), \\ \Delta U(k\mid k) &= \Xi Y^d + \Theta x(k\mid k) \\ &\quad + \Psi\Delta U(k-1\mid k-1), \\ y(k) &= Cx(k). \end{aligned} \quad (26)$$

Define the extended state

$$X_N(k) = \left[x^T(k) \quad X^T\left(k,N_p\mid k-1\right) \quad \Delta U^T(k\mid k) \quad \Delta U^T(k-1\mid k-1)\right],$$

$$\begin{bmatrix} x(k) \\ X\left(k,N_p\mid k-1\right) \\ \Delta U(k\mid k) \\ \Delta U(k-1\mid k-1) \end{bmatrix} = \begin{bmatrix} A & 0 & B & 0 \\ S & 0 & T\overline{B}E & T\overleftrightarrow{B}E \\ \Theta A & 0 & \Theta B+\Psi & 0 \\ 0 & 0 & I & 0 \end{bmatrix}\begin{bmatrix} x(k-1) \\ X\left(k-1,N_p\mid k-1\right) \\ \Delta U(k-1\mid k-1) \\ \Delta U(k-2\mid k-2) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \Xi \\ 0 \end{bmatrix}Y^d,$$

$$y(k) = \begin{bmatrix} C & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x(k) \\ X\left(k,N_p\mid k-1\right) \\ \Delta U(k\mid k) \\ \Delta U(k-1\mid k-1) \end{bmatrix}. \quad (27)$$

□

4.2. Performance Analysis

Remark 7 (exchange information). In traditional DMPC, the optimal variable is $\Delta U_i(k\mid k) = [\Delta u_i(k\mid k), \Delta u_i(k+1\mid k), \ldots, \Delta u_i(k+N_u-1\mid k)]$, whose dimension is $N_u^* n_{ui}$.

In CP-DMPC algorithm, the optimal variable is $\Delta u_i(k\mid k)$ and the dimension of variable is n_{ui}, which decreases greatly. As a result, exchange information among the CP-DMPC controllers reduces from $N_u^* n_{ui}$ to n_{ui}.

Remark 8. However, the computation of the optimization problem is reduced greatly because of the dimension reduction of the optimal variables.

The control value is calculated as

$$\begin{aligned} &\Delta u_i(k\mid k) \\ &\quad = \overline{K}_i\left(Y_i^d - S_i x_i(k\mid k) - T_i W_i\left(k,N_p\mid k-1\right)\right). \end{aligned} \quad (28)$$

In the traditional DMPC algorithm, when the number of subsystem inputs and the control horizon becomes large, the optimized control sequence $\Delta U_i(k\mid k) = [\Delta u_i(k\mid k), \Delta u_i(k+1\mid k), \ldots, \Delta u_i(k+N_u-1\mid k)]$ is highly dimensional. The matrices Φ_{ii} have also high dimensions. The computation load of (10a) and (10b) is mainly to calculate the inverse of the matrix $(\Phi_{ii}^T Q_i \Phi_{ii} + R_i)^{-1}$, which may require significant computational resources.

In CP-DMPC algorithm, Φ_{ii} is a vector not a matrix. Compared with (10a) and (10b), the computation load of (21) is lower because of no calculation of the matrix inverse. As a result, the CP-DMPC controller decreases the computation demand greatly.

5. Simulations and Results

In this section the theoretical results are illustrated using two different examples. The first example is focused on the process control system, four-tank system whose sampling time interval is about several seconds. The second example is focused on the motion control, multirobot target tracking scenario whose sampling time interval is about milliseconds. All the simulations are run in MATLAB on the same computer with Intel(R) Core (TM) 2.6 GHz processor and 8 GB RAM.

5.1. Four-Tank Plant

5.1.1. System Description. The four-tank problem used in the section is described by [21–23] and the description of the

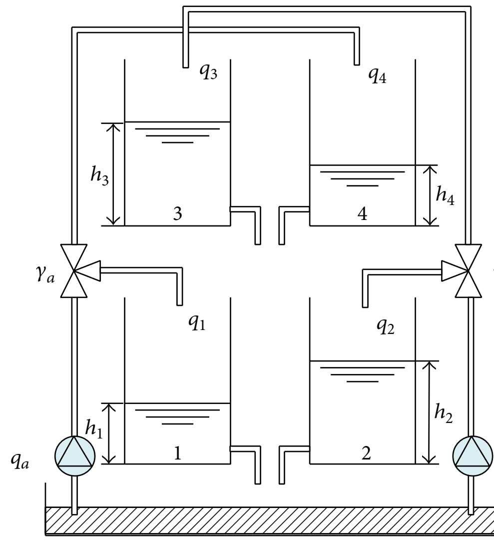


Figure 5: Description of the four-tank system.

system is shown in Figure 5. It is a multivariable system with two manipulates variables and four state variables. The differential equations that model the nonlinear dynamics of the system can be expressed as

$$\begin{aligned} \frac{dh_1}{dt} &= -\frac{a_1}{S}\sqrt{2gh_1} + \frac{a_3}{S}\sqrt{2gh_3} + \frac{\gamma_a}{S}q_a, \\ \frac{dh_2}{dt} &= -\frac{a_2}{S}\sqrt{2gh_2} + \frac{a_4}{S}\sqrt{2gh_4} + \frac{\gamma_b}{S}q_b, \\ \frac{dh_3}{dt} &= -\frac{a_3}{S}\sqrt{2gh_3} + \frac{(1-\gamma_b)}{S}q_b, \\ \frac{dh_4}{dt} &= -\frac{a_4}{S}\sqrt{2gh_4} + \frac{(1-\gamma_a)}{S}q_a, \end{aligned} \tag{29}$$

where the parameters in (29) can be found in Table 2.

For the predictive controllers to be tested, a linear predictive model is obtained by linearizing (29) at the operating point. Define the deviation variables

$$\begin{aligned} x_i &= h_i - h_i^0, \quad i = 1,2,3,4, \\ u_1 &= q_a - q_a^0, \\ u_2 &= q_b - q_b^0. \end{aligned} \tag{30}$$

The following continuous-time linear model can be obtained:

$$\begin{aligned} \frac{dx}{dt} &= A_c x + B_c u, \\ y &= C_c x, \end{aligned} \tag{31}$$

where $x = (x_1, x_2, x_3, x_4)^T$, $u = (u_1, u_2)^T$, $y = (x_1, x_2)^T$,

$$\begin{aligned} A_c &= \begin{bmatrix} \frac{-1}{\tau_1} & 0 & \frac{1}{\tau_3} & 0 \\ 0 & \frac{-1}{\tau_2} & 0 & \frac{1}{\tau_4} \\ 0 & 0 & \frac{-1}{\tau_3} & 0 \\ 0 & 0 & 0 & \frac{-1}{\tau_4} \end{bmatrix}, \\ B_c &= \begin{bmatrix} \frac{\gamma_a}{S} & 0 \\ 0 & \frac{\gamma_b}{S} \\ 0 & \frac{1-\gamma_b}{S} \\ \frac{1-\gamma_a}{S} & 0 \end{bmatrix}, \\ C_c &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \end{aligned} \tag{32}$$

where $\tau_i = S/a_i\sqrt{2h_i^0 g}$, $i = 1,2,3,4$.

The whole system can be divided into two input-coupled subsystems. Subsystem 1 consists of tanks 1 and 3 while subsystem 2 consists of tanks 2 and 4. And the two subsystems are discretized with a sampling time.

Subsystem 1

$$\begin{aligned} x_{s1}(k) &= (x_1(k), x_3(k))^T, \\ y_{s1}(k) &= (x_1(k))^T, \\ u(k) &= (u_1(k), u_2(k))^T, \\ x_{s1}(k+1) &= A_{c1}x_{s1}(k) + B_{c1}u(k) \\ &= A_{c1}x_{s1}(k) + B_{c1}^{(1)}u_1(k) + B_{c1}^{(2)}u_2(k), \\ y_{s1}(k) &= C_{c1}x(k). \end{aligned} \tag{33}$$

Subsystem 2

$$\begin{aligned} x_{s2}(k) &= (x_2(k), x_4(k))^T, \\ y_{s2}(k) &= (x_2(k))^T, \\ u(k) &= (u_1(k), u_2(k))^T, \\ x_{s2}(k) &= A_{c2}x_{s2}(k) + B_{c2}u(k) \\ &= A_{c2}x_{s2}(k) + B_{c2}^{(1)}u_1(k) + B_{c2}^{(2)}u_2(k), \\ y_{s2}(k) &= C_{c2}x(k). \end{aligned} \tag{34}$$

5.1.2. Simulations with Centralized MPC, DMPC, and CP-DMPC. The control objective in the four-tank system is to

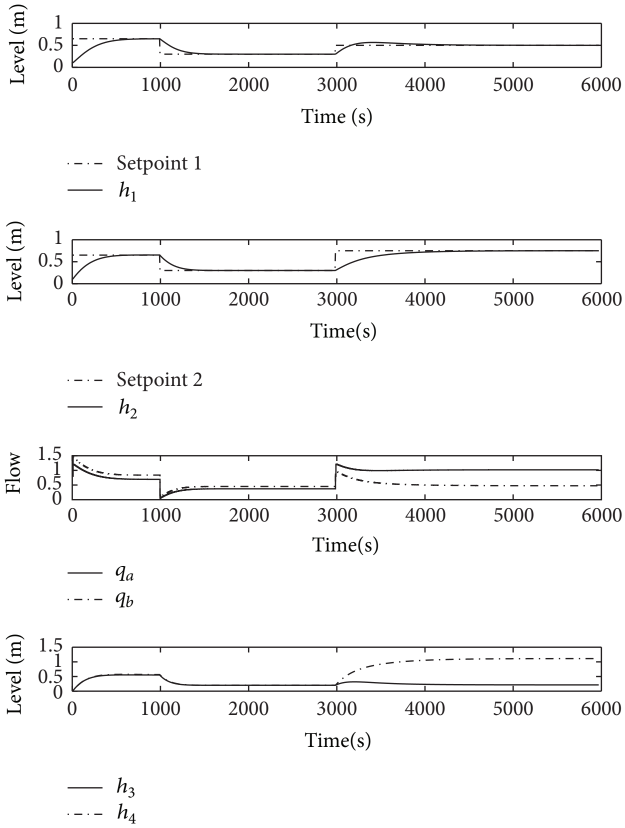


Figure 6: Dynamic response of the four-tank system of centralized MPC for tracking.

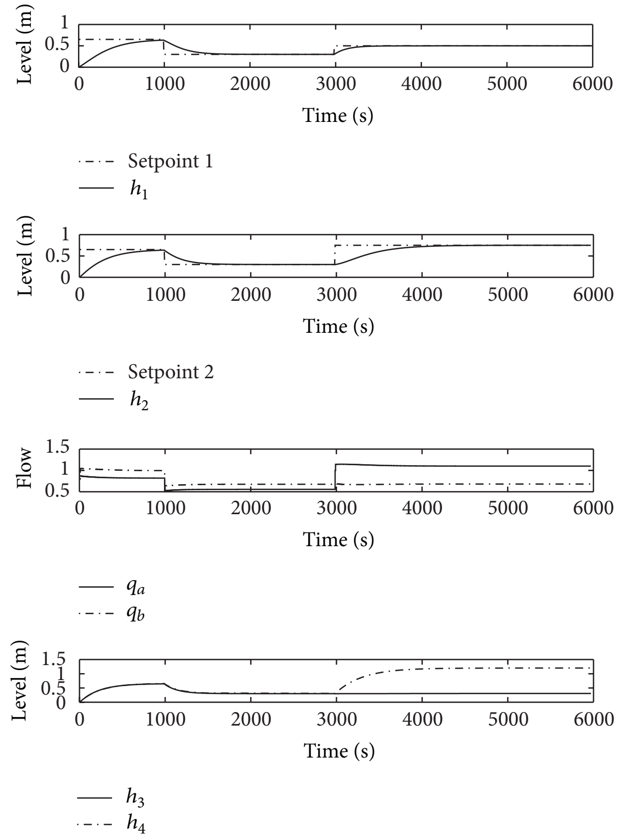


Figure 7: Dynamic response of the four-tank system of DMPC for tracking.

keep the levels of tank 1 and tank 2 at reference values. In this section, the system performance of three control algorithms is compared, which are centralized MPC, DMPC, and CP-DMPC. All of these strategies have the same input constraints, input and output weights, prediction, and control horizon. The parameters used in the simulations are $Q_i = 1$, $R_i = 0.01$, $N_p = 15$, $N_u = 4$, $i = 1, 2$. And the sampling time is 5 s. The parameter used in CP-DMPC is $\beta = 0.1$.

The set-point levels of tank 1 and tank 2 are as follows:

(1) From 0 s to 1000 s, the set-point of tank 1 is 0.65 m and the set-point of tank 2 is 0.65 m.

(2) From 1001 s to 3000 s, the set-point of tank 1 is 0.3 m and the set-point of tank 2 is 0.3 m.

(3) From 1001 s to 3000 s, the set-point of tank 1 is 0.5 m and the set-point of tank 2 is 0.75 m.

From Figures 6, 7 and 8, we can conclude that CMPC has the best control and that CP-DMPC can also have similar control performance as traditional DMPC (noniterative). But from Figure 9, we can see that the CMPC and traditional DMPC provide a higher optimization time than CP-DMPC algorithm.

5.2. Multirobot Target Tracking Scenario. In the section, N robots with sensors track a target and the motion model of each robot is

$$\begin{bmatrix} P_{x,i}(k+1) \\ P_{y,i}(k+1) \end{bmatrix} = \begin{bmatrix} P_{x,i}(k) \\ P_{y,i}(k) \end{bmatrix} + T_s \begin{bmatrix} v_{x,i}(k) \\ v_{y,i}(k) \end{bmatrix}, \tag{35}$$

where $P_i(k) = [P_{x,i}(k), P_{y,i}(k)]$ is the state of ith robot at time k. $P_{x,i}(k)$ and $P_{y,i}(k)$ are the x-coordinate position and y-coordinate position of ith robot at time k. $v_{x,i}(k)$ and $v_{y,i}(k)$ are the x-coordinate velocity and y-coordinate velocity of ith robot at time k. T_s is the time interval.

The target motion model is modeled by the constant velocity model, that is,

$$x_t(k+1) = F x_t(k), \tag{36}$$

where

$$F = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{37}$$

The objective of the whole system is to track a target with N robots and to keep the distant between the robots and the target. Meanwhile there will not be a collision among the robots during tracking the target. As a result, the local performance index of the ith robot can be selected as

$$J_i(k) = \left(\left\| P_i(k) - P_t(k) \right\| - R\right)^2. \tag{38}$$

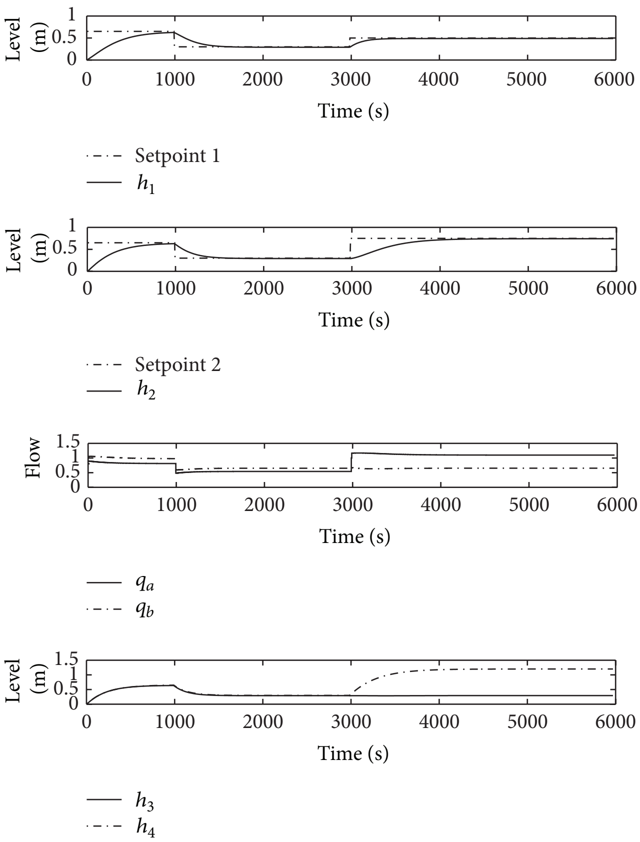


Figure 8: Dynamic response of the four-tank system of CP-DMPC for tracking.

We simulate the scenario from time $t = 1, \ldots, 30$ s. The target moves according to the dynamic (25) with the sampling time $T_s = 1$ in the area collectively monitored by the three robots states above. The initial positions of three robots is (20 m, 0 m), (−20 m, 0 m), and (0 m, −10 m). The maximum velocities of robots are 2 m/s of the x and y coordinates. The initial positions of the target is (−10 m, −10 m) and the target motion trajectory is illustrated in Figure 10.

5.2.1. Simulations with CP-DMPC Algorithm. In this section, the system performance of two control algorithms is compared, which are DMPC (noniterative) and CP-DMPC. Both of these strategies have the same input constraints, input and output weights, prediction, and control horizon. The parameters used in the simulations are $Q_i = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $R_i = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $N_p = 4$, $N_u = 4$ $(i = 1, 2, 3)$. The parameter used in CP-DMPC is $\beta = 1$. The traditional DMPC and CP-DMPC algorithms are applied to the scenario by the same parameters.

The trajectories of three robots and target and four typical snapshots at time = 1, 10, 20, 30 are depicted in Figure 11. The simulation results demonstrate that the multirobot system with the CP-DMPC controller can track the target well.

5.2.2. Comparisons between Traditional MPC and CP-DMPC Algorithm. In the section, we compare the computational

Table 1: Metrics comparisons among different algorithms.

Algorithm	Centralized MPC	Distributed MPC	
		Traditional DMPC	CP-DMPC
Solution	Optimal	Nash optimal	Suboptimal
Robustness	Central node failure leads to system down	Good	Good
Information	Large	Small	Smaller
Calculation load	Large	Small	Smaller

Table 2: Parameters of the four-tank system.

	Value	Unit	Description
h_1, h_2, h_3, h_4			Water level
a_1	$1.31 * 10^{-4}$	m^2	Discharge constant of tank 1
a_2	$1.51 * 10^{-4}$	m^2	Discharge constant of tank 2
a_3	$9.27 * 10^{-5}$	m^2	Discharge constant of tank 3
a_4	$8.82 * 10^{-5}$	m^2	Discharge constant of tank 4
S	0.06	m^2	Cross-section of the tanks
q_a	1.63	m^3/h	Flow a
q_b	2.00	m^3/h	Flow b
γ_a	0.3		Ratio of the three-way valve of pump a
γ_b	0.4		Ratio of the three-way valve of pump b
g	$9.8 * 3600 * 3600$	m/h^2	

complexity, communication energy, and optimal performance index value between the traditional DMPC and CP-DMPC (Table 1).

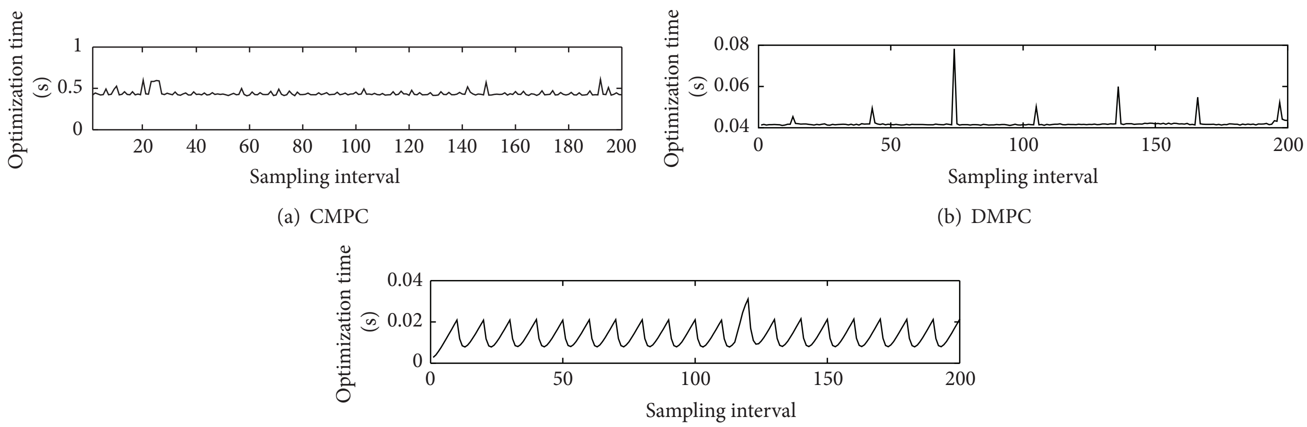
In [24], communication energy is made up of transmitting energy E_{tx} and receiving energy E_{rx}:

$$E_{tx}(i, j) = \left(\alpha_1 + \alpha_2 d(i, j)^n\right) r, \tag{39}$$

where $d(i, j)$ is the distance between the two robots, n is the path loss index, r is a transmitting data rate, and α_1, α_2 are constants (45 nJ/bit and 10 pJ/bit). And the receiving energy E_{rx} is constant, which is 135 nJ/bit.

The computational complexity corresponds to the number of operations required to complete the task, where an operation is defined as a combination of one addition and one multiplication. And model predictive control requires the solution of an open-loop optimal control problem at every sampling instant. In the paper, we use fast gradient method which has low implementation calculation and numerical robustness.

The two optimization problems between traditional DMPC and CP-DMPC algorithm are evaluated 50 times. The simulation results are shown in Figure 11. From Figure 12(a), the traditional DMPC provides a lower performance cost (better system performance) than CP-DMPC algorithm.



(a) CMPC

(b) DMPC

(c) CP-DMPC

Figure 9: Optimization time: CMPC versus DMPC versus CP-DMPC.

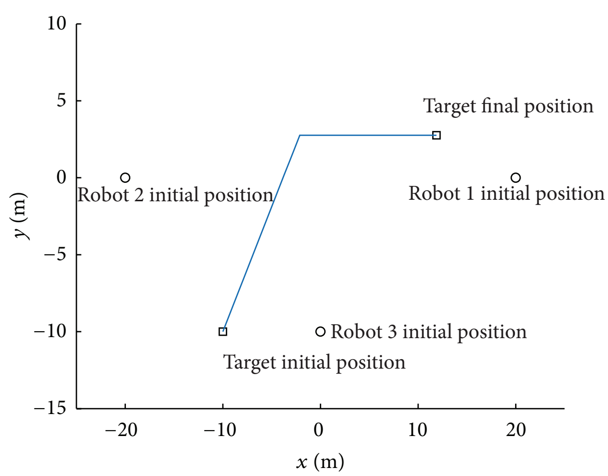


Figure 10: Scenario: three robots and one target.

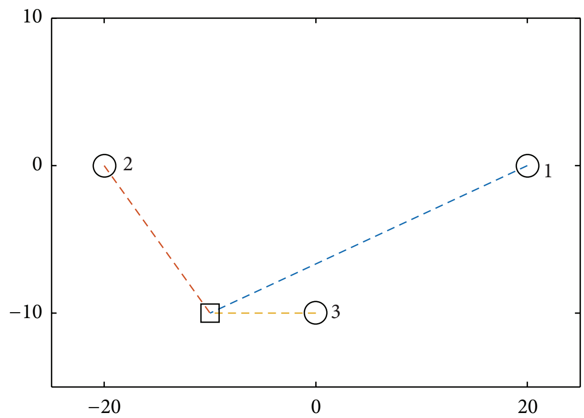
From Figure 12(b), the communication energy using traditional DMPC is generally larger than that of CP-DMPC algorithm. This is because the traditional DMPC transmits the optimal variable $\Delta U_i(k \mid k) = [\Delta u_i(k \mid k), \Delta u_i(k+1 \mid k), \ldots, \Delta u_i(k+N_u-1 \mid k)]$, and it has higher communication burden than the CP-DMPC algorithm. From Figure 12(c), the time needed to solve the traditional DMPC is much larger than the time needed to solve the CP-DMPC. It is because the traditional DMPC has to solve a much larger (in terms of decision variables) optimization problem than the CP-DMPC.

From Figure 12, we can see that the traditional DMPC provides a lower performance cost (better system performance) than CP-DMPC algorithm. But the CP-DMPC provides a lower calculation demand and communication data than the traditional DMPC.

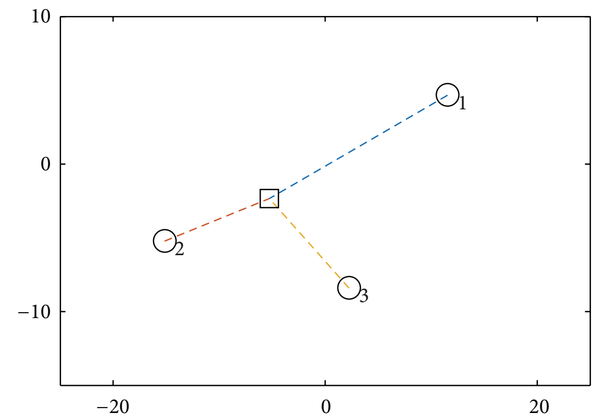
Obviously, a short prediction horizon would require a smaller amount of communication data and computational time, and a longer prediction horizon can prove the better effectiveness of CP-DMPC compared to the traditional DMPC. From Figure 13, the communication data in traditional DMPC increases as the prediction horizon increases. But the communication data in CP-DMPC do not change too much as the prediction horizon increases.

6. Conclusion

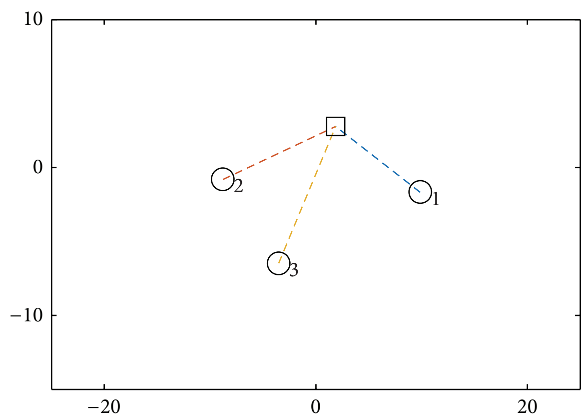
In the paper, a distributed model predictive control scheme with control planning set has been proposed. In the proposed scheme, the future control sequences are approximated by a set of planning set. It can reduce exchange information among the controller and at the same time also can reduce the distributed MPC controller calculation demand without degrading the whole system performance. Extensive simulations using a multirobot target tracking example have been carried out to compare the proposed distributed MPC with existing traditional DMPC algorithms from computational complexity, communication energy, and closed-loop system performance.



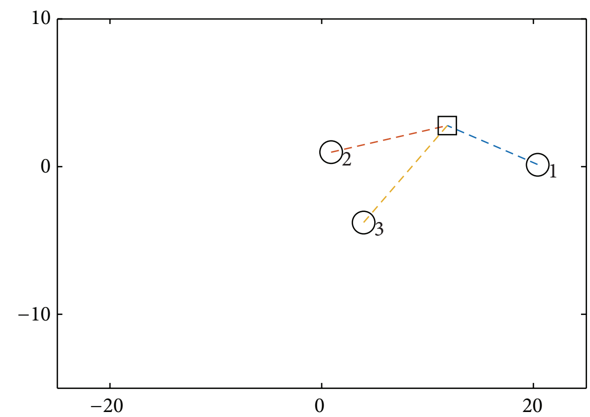
(a) Timestamp = 1 second



(a) Timestamp = 10 seconds

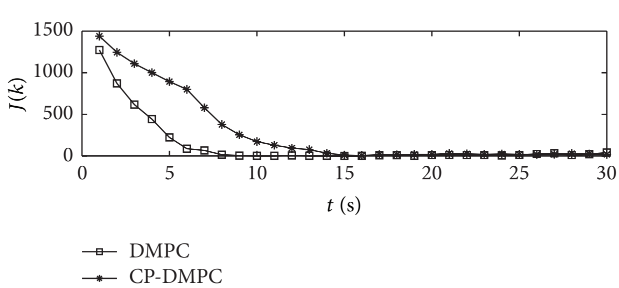


(a) Timestamp = 20 seconds

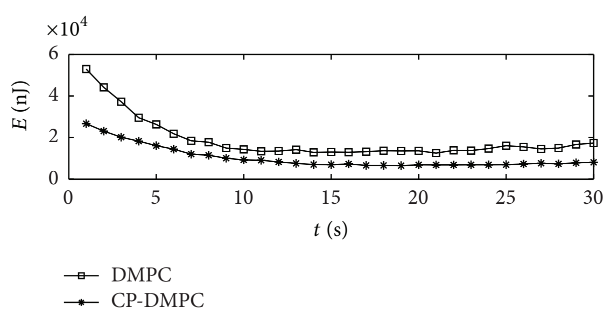


(a) Timestamp = 30 seconds

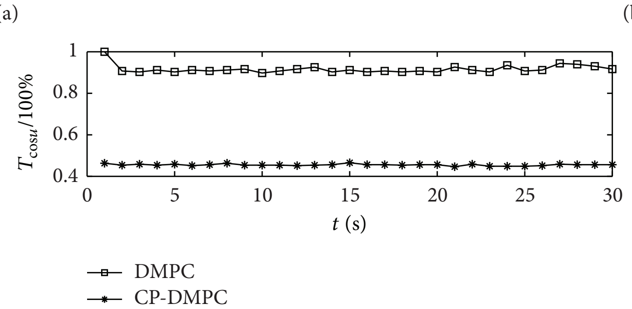
Figure 11: Four snapshots in target tracking scenario.



(a)



(b)



(c)

Figure 12: (a) Performance index: DMPC versus CP-DMPC. (b) Communication energy: DMPC versus CP-DMPC. (c) Relative computational time consumption in one robot: DMPC versus CP-DMPC.

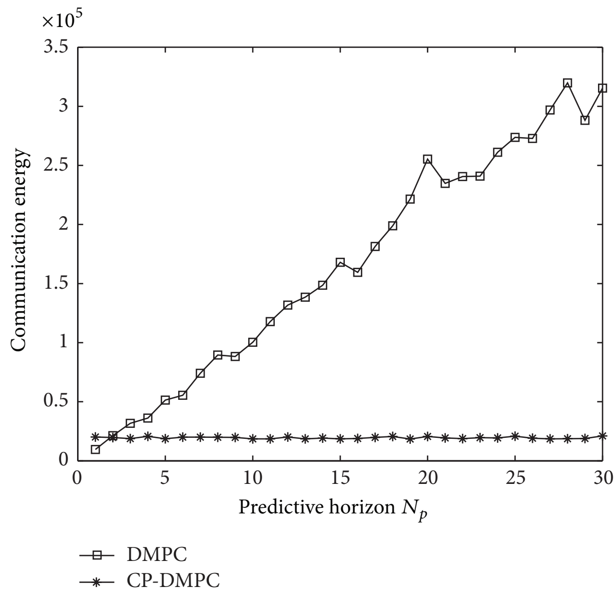


Figure 13: Communicate energy among robots with different prediction horizon.

Competing Interests

The author declares that she has no competing interests.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant no. 61004082.

References

[1] F. Eduardo and C. Bordons, *Model Predictive Control*, Springer, Berlin, Germany, 2nd edition, 2013.

[2] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[3] M. Morari and J. H. Lee, “Model predictive control: past, present and future,” *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, 1999.

[4] P. Giselsson and A. Rantzer, “On feasibility, stability and performance in distributed model predictive control,” *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 1031–1036, 2014.

[5] M. Farina, G. Betti, L. Giulioni, and R. Scattolini, “An approach to distributed predictive control for tracking-theory and applications,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1558–1566, 2014.

[6] A. Ferramosca, D. Limon, I. Alvarado, and E. F. Camacho, “Cooperative distributed MPC for tracking,” *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 49, no. 4, pp. 906–914, 2013.

[7] M. Zhao and B. Ding, “Distributed model predictive control for constrained nonlinear systems with decoupled local dynamics,” *ISA Transactions*, vol. 55, pp. 1–12, 2015.

[8] R. Scattolini, “Architectures for distributed and hierarchical model predictive control—a review,” *Journal of Process Control*, vol. 19, no. 5, pp. 723–731, 2009.

[9] R. R. Negenborn and J. M. Maestre, “Distributed model predictive control: an overview and roadmap of future research opportunities,” *IEEE Control Systems*, vol. 34, no. 4, pp. 87–97, 2014.

[10] J. Liu, X. Chen, D. M. de la Peña, and P. D. Christofides, “Sequential and iterative architectures for distributed model predictive control of nonlinear process systems,” *AIChE Journal*, vol. 56, no. 8, pp. 2137–2149, 2010.

[11] S. Li, Y. Zhang, and Q. Zhu, “Nash-optimization enhanced distributed model predictive control applied to the Shell benchmark problem,” *Information Sciences*, vol. 170, no. 2–4, pp. 329–349, 2005.

[12] Y. Zhang and S. Li, “Networked model predictive control based on neighbourhood optimization for serially connected large-scale processes,” *Journal of Process Control*, vol. 17, no. 1, pp. 37–50, 2007.

[13] J. B. Rawlings and B. T. Stewart, “Coordinating multiple optimization-based controllers: new opportunities and challenges,” *Journal of Process Control*, vol. 18, no. 9, pp. 839–845, 2008.

[14] L. Giovanini, “Game approach to distributed model predictive control,” *IET Control Theory and Applications*, vol. 5, no. 15, pp. 1729–1739, 2011.

[15] J. M. Maestre, D. Muñoz de la Peña, and E. F. Camacho, “Distributed model predictive control based on a cooperative game,” *Optimal Control Applications & Methods*, vol. 32, no. 2, pp. 153–176, 2011.

[16] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, “Distributed model predictive control,” *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 44–52, 2002.

[17] M. Farina and R. Scattolini, “Distributed non-cooperative MPC with neighbor-to-neighbor communication ?” in *Proceedings of the 18th IFAC World Congress*, pp. 404–409, Milano, Italy, September 2011.

[18] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia, “Cooperative distributed model predictive control,” *Systems & Control Letters*, vol. 59, no. 8, pp. 460–469, 2010.

[19] M. Vaccarini, S. Longhi, and M. Reze Katebi, “Unconstrained networked decentralized model predictive control,” *Journal of Process Control*, vol. 19, pp. 328–339, 2009.

[20] A. Wallen and K. J. Astrom, “Pulse-step control,” in *Proceedings of the 15th IFAC Triennial World Congress*, Barcelona, Spain, 2002.

[21] K. H. Johansson and J. L. R. Nunes, “A multivariable laboratory process with an adjustable zero,” in *Proceedings of the American Control Conference (ACC ’98)*, pp. 2045–2049, June 1998.

[22] M. Mercangöz and F. J. Doyle III, “Distributed model predictive control of an experimental four-tank system,” *Journal of Process Control*, vol. 17, no. 3, pp. 297–308, 2007.

[23] Y. Alipouri and J. Poshtan, “Optimal controller design using discrete linear model for a four tank benchmark process,” *ISA Transactions*, vol. 52, no. 5, pp. 644–651, 2013.

[24] M. Bhardwaj and A. P. Chandrakasan, “Bounding the lifetime of sensor networks via optimal role assignments,” in *Proceedings of the 21st IEEE Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1587–1596, New York, NY, USA, 2002.