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

Copyright © 2016 Wei Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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 $i$th subsystem is as follows:

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

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

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 $i$th subsystem at $k$th sampling time. The model (1a), (1b) is changed to suit the model predictive control design with an embedded integrator. The augmented model of the $i$th subsystem state space model is

\[
x_i(k + 1) = A_{i}x_i(k) + B_{ii}u_i(k) + w_i(k), \tag{2a}
\]

\[
y_i(k) = C_{i}x_i(k), \tag{2b}
\]

where a new state variable vector is chosen to be

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

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

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

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

\[
A_i = \begin{bmatrix}
A_{m,i} & O \\
C_{m,i}A_{m,i} & I
\end{bmatrix},
\]

\[
B_{ii} = \begin{bmatrix}
B_{m,ii} \\
C_{m,i}B_{m,ii}
\end{bmatrix},
\]

\[
B_{ij} = \begin{bmatrix}
B_{m,ij} \\
C_{m,ij}B_{m,ij}
\end{bmatrix},
\]

\[
C_i = [O \quad I].
\]

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

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

\[
y(k) = Cx(k) \tag{8b}
\]
with state vector $x(k) \in \mathbb{R}^n$, control input vector $\Delta u(k) \in \mathbb{R}^n$, and output vector $y(k) \in \mathbb{R}^n$. $A, B,$ and $C$ are the whole system matrices. This implies that

\[
A = \begin{bmatrix}
A_1 & \cdots & A_i & \cdots & A_N
\end{bmatrix},
\]

\[
B = \begin{bmatrix}
B_{11} & \cdots & B_{ij} & \cdots & B_{1N} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
B_{i1} & \cdots & B_{ii} & \cdots & B_{IN} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
B_{N1} & \cdots & B_{Nj} & \cdots & B_{NN}
\end{bmatrix},
\]

\[
C = \begin{bmatrix}
C_1 \\
\vdots \\
C_i \\
\vdots \\
C_N
\end{bmatrix},
\]

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

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

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

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:

\[
\begin{align*}
J(k) &= \sum_{i=1}^{N} J_i(k), \\
J_i(k) &= \sum_{l=1}^{N_p} \|y_i(k+l|k) - y_i^d(k+l)\|_Q^2 + \sum_{l=1}^{N_u} \|\Delta u_i(k+l-1|k)\|_R_i^2 \quad \text{(10a)}
\end{align*}
\]

s.t.

\[
x_i(k+l+1|k) = A_i x_i(k+l|k) \\
+ B_{ii} \Delta u_i(k+l|k) \\
+ w_i(k+l|k),
\]

\[
y_i(k+l|k) = C_i x_i(k+l|k) \quad \text{for } i = 1, \ldots, N.
\]

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 $w_i(k+l|k)$ 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 $\Delta U^*(k, N_u | k) = [\Delta u^*(k | k)$,
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.
The \( i \)th subsystem minimizes the following local performance index, which is the \( i \)th optimization problem [19]:

\[
J_i(k) = \sum_{l=1}^{N_u} \left\| y_i^d(k+l|k) - y_i(k+l|k) \right\|_{Q_i}^2 + \sum_{l=1}^{N_u} \left\| \Delta u_i(k+l-1|k) \right\|_{R_i}^2
\]

subject to

\[
x_i(kk + l + 1) = A_i x_i(kk + l) + B_i \Delta u_i(kk + l) + w_i(kk + l - 1),
\]

\[
y_i(kk+l) = C_i x_i(kk+l).
\]

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 \( w_i(k + l | k) \) are unknown for the \( i \)th subsystem. And only the prediction \( w_i(k + l | k - 1) \) 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 \( \Delta U^*_i(k, N_u | k) \) and broadcast \( \Delta U^*_i(k | k) \) 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.

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

\[
\Delta u(k + l | k) = f(\Delta u(k + l - 1 | k)).
\]

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

\[
\Delta u(k + l | k) = \beta \Delta u(k + l - 1 | k).
\]

In the control planning set algorithm, the future control sequence is restricted by one possibility. The parameter \( \beta \) 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

$$\tilde{B}_i = [B_{i1}, \ldots, B_{i(i-1)}, O, B_{i(i+1)}, \ldots, B_{iN}], \quad (14a)$$

$$\Gamma_i = \begin{bmatrix} I_{n_u \times n_u} & \vdots & I_{n_u \times n_u} \\ O_{(N_p-N_u) \times n_u} & \cdots & O_{(N_p-N_u) \times n_u} \end{bmatrix}^T, \quad (14b)$$

$$\tilde{B} = \text{diag}_{N_p} (\tilde{B}_i) \Gamma_i, \quad (14c)$$

$$\Gamma = \begin{bmatrix} \Gamma_1 & \vdots & \Gamma_N \end{bmatrix}, \quad (14d)$$

$$E_i = \begin{bmatrix} I_{n_u \times n_u} & \beta I_{n_u \times n_u} & \cdots & \beta^{N_p-1} I_{n_u \times n_u} \end{bmatrix}^T, \quad (14e)$$

$$E = \text{diag} \{E_1, E_2, \ldots, E_N\}, \quad (14f)$$

$$S_i = \begin{bmatrix} (A_i) & (A_i^2) & \cdots & (A_i^{N_p}) \end{bmatrix}^T, \quad (14g)$$

$$S = \text{diag} \{S_1, S_2, \ldots, S_N\}, \quad (14h)$$

$$T_i = \begin{bmatrix} A_i^0 & 0 \\ \vdots & \vdots \\ A_i^{N_p-1} & A_i^0 \end{bmatrix}, \quad (14i)$$

$$T = \text{diag} \{T_1, T_2, \ldots, T_N\}, \quad (14j)$$

$$\tilde{B}_i = \text{diag}_{N_p} (B_{i1}, \ldots, B_{i(i-1)}, O, B_{i(i+1)}, \ldots, B_{iN}) \Gamma_i, \quad (14k)$$

$$\tilde{B} = \begin{bmatrix} \tilde{B}_1 & \ldots & \tilde{B}_N \end{bmatrix}, \quad (14l)$$

**Lemma 1.** The interaction predictions of $i$th subsystem at time $k$ are given by

$$W_i (k, N_p | k-1) = \tilde{B}_i E \Delta U (k | k-1) \quad (15)$$

and the compact predictions have the following form:

$$W (k, N_p | k-1) = \tilde{B} E \Delta U (k | k-1). \quad (16)$$

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

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

$$= \tilde{B}_i \Delta U (k | k-1)$$

$$w_i (k+1 | k-1) = \sum_{j=1, j \neq i}^{N} B_{ij} \Delta u_j (k+1 | k-1)$$

$$= \sum_{j=1, j \neq i}^{N} B_{ij} \beta \Delta u_j (k | k-1) = \beta \tilde{B}_i \Delta U (k | k-1)$$

$$\vdots$$

$$w_i (k + N_u | k-1) = \sum_{j=1, j \neq i}^{N} B_{ij} \Delta u_j (k + N_u | k-1)$$

$$= \sum_{j=1, j \neq i}^{N} B_{ij} \beta^{N_u-1} \Delta u_j (k | k-1)$$

$$= \beta^{N_u-1} \tilde{B}_i \Delta U (k | k-1) \quad (17)$$

$$\vdots$$

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

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

$$X_i (k+1, N_p | k) = S_i X_i (k | k) + T_i \tilde{B}_i \Delta U_i (k)$$

$$+ T_i \tilde{B}_i E \Delta U (k | k-1) \quad (18)$$

$$Y_i (k+1, N_p | k) = C_i X_i (k+1, N_p | k)$$

and the compact predictions have the following form:

$$X (k+1, N_p | k) = S X (k | k) + T \tilde{B} E \Delta U (k | k-1) \quad (19)$$

$$Y (k+1, N_p | k) = C X (k+1, N_p | k).$$
Algorithm 1: Algorithm CP-DMPC: CP-DMPC in a pseudo-algorithm format.

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

\begin{align*}
x_i(k+1 | k) &= A_i x_i(k | k) + B_{ij} \Delta u_j(k | k) \\
&\quad + \sum_{j=1,j \neq i}^{N} B_{ij} \Delta u_j(k | k) \\
x_i(k+2 | k) &= A_i x_i(k+1 | k) + B_{ij} \Delta u_j(k+1 | k) \\
&\quad + \sum_{j=1,j \neq i}^{N} B_{ij} \Delta u_j(k+1 | k) = A_i^2 x_i(k | k) + [A_i B_{ij} \\
&\quad + B_{ij} \beta] \Delta u_j(k | k) + \sum_{j=1,j \neq i}^{N} [A_i B_{ij} + B_{ij} \beta] \\
&\quad \cdot \Delta u_j(k | k) \\
&\quad + \cdots \\
x_i(k + N_p | k) &= A_i^{N_p} x_i(k | k) + [A_i^{N_p - 1} B_{ij} + \cdots \\
&\quad + A_i^{N_p - N_p} B_{ij} \beta^{N_p - 1}] \Delta u_j(k | k) \\
&\quad + \sum_{j=1,j \neq i}^{N} [A_i^{N_p - 1} B_{ij} + \cdots + A_i^{N_p - N_p} B_{ij} \beta^{N_p - 1}] \\
&\quad \cdot \Delta u_j(k | k).
\end{align*}

By definitions (14a)-(14l), this implies the relations (18) and the equivalent compact forms (19) hold. \hfill \Box

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

Lemma 4. The $i$th subsystem at time $k$ has to solve the following optimization problem:

\begin{align*}
J_i &= -G_i^T(k + 1, N_p | k) \Delta U_i \\
&\quad + \Delta U_i^T \left( \Phi_i^T Q_i \Phi_i + R_i \right) \Delta U_i, \quad (21)
\end{align*}

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

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

\begin{align*}
J_i &= (Y_i^d - Y_i)^T Q_i (Y_i^d - Y_i) + \Delta U_i^T R_i \Delta U_i, \quad (22)
\end{align*}

Applying (18) into it, the local performance index $J_i$ takes the form (21). \hfill \Box

Theorem 5. For the $i$th subsystem, the explicit form of the control law is given by

\begin{align*}
\Delta u_i(k | k) &= \mathcal{K}_i \left( Y_i^d - S_i x_i(k | k) - T_i W_i(k, N_p | k-1) \right). \\
\Delta U(k | k) &= \Xi Y^d + \Theta x(k | k) \\
&\quad + \Psi \Delta U(k-1 | k-1), \quad (23)
\end{align*}

where $\mathcal{K}_i = (\Phi_i^T Q_i \Phi_{ii} + R_i)^{-1} \Phi_i^T Q_i$, $\Xi = \text{diag}(\mathcal{K}_1, \ldots, \mathcal{K}_N)$, $\Theta = -2S_i$, $\Psi = \Xi T_i \tilde{B}_i E$

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
\[
\lambda \left( \begin{bmatrix} A & 0 & B & 0 \\ S & 0 & TBE & TBE \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \right) \prec 1.
\] (25)

Proof. Combining the process (8a) and (8b) and control law (23), the closed-loop state-space representation is derived:
\[
x (k) = Ax (k-1) + B \Delta u (k-1 | k-1),
\]
Define the extended state
\[
X (k) = \begin{bmatrix} x^T (k) & X^T (k, N_p | k-1) & \Delta U^T (k | k) & \Delta U^T (k-1 | k-1) \end{bmatrix},
\]
\[
X (k, N_p | k-1) = Sx (k-1) + TBE \Delta U (k-1 | k-1) + TBE \Delta U (k-2 | k-2),
\]
\[
\Delta U (k | k) = \Xi Y^d + \Theta x (k | k) + \Psi \Delta U (k-1 | k-1),
\]
\[
y (k) = Cx (k).
\] (26)

4.2. Performance Analysis

Remark 7 (exchange information). In traditional DMPC, the optimal variable is \( \Delta U_i (k | k) = [\Delta u_i (k | k), \Delta u_i (k+1 | k), \ldots, \Delta u_i (k+N_u-1 | k) ] \), whose dimension is \( N_p \times n_u \).

In CP-DMPC algorithm, the optimal variable is \( \Delta u_i (k | k) \) and the dimension of variable is \( n_u \), which decreases greatly. As a result, exchange information among the CP-DMPC controllers reduces from \( N_p \times n_u \) to \( n_u \).

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
\[
\Delta u_i (k | k) = K_i (Y^d_i - S_i x_i (k | k) - T_i W_i (k, N_p | k-1)).
\] (28)

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

In CP-DMPC algorithm, \( \Phi_i \) 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
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

\[
\begin{align*}
\frac{dh_1}{dt} &= -\frac{a_1}{S}\sqrt{2gh_1} + \frac{a_3}{S}\sqrt{2gh_3} + \frac{γ_a}{S}q_a, \\
\frac{dh_2}{dt} &= -\frac{a_2}{S}\sqrt{2gh_2} + \frac{a_4}{S}\sqrt{2gh_4} + \frac{γ_b}{S}q_b, \\
\frac{dh_3}{dt} &= -\frac{a_3}{S}\sqrt{2gh_3} + \frac{(1 - γ_b)}{S}q_b, \\
\frac{dh_4}{dt} &= -\frac{a_4}{S}\sqrt{2gh_4} + \frac{(1 - γ_a)}{S}q_a,
\end{align*}
\]

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

\[
\begin{align*}
x_i &= h_i - h_i^0, & i = 1, 2, 3, 4, \\
u_1 &= q_a - q_a^0, \\
u_2 &= q_b - q_b^0.
\end{align*}
\]

The following continuous-time linear model can be obtained:

\[
\frac{dx}{dt} = A_c x + B_c u, \\
y = C_c x,
\]

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

\[
A_c = \begin{bmatrix}
-1/	au_1 & 0 & 1/	au_3 & 0 \\
0 & -1/	au_2 & 0 & 1/	au_4 \\
0 & 0 & -1/	au_3 & 0 \\
0 & 0 & 0 & -1/	au_4
\end{bmatrix},
\]

\[
B_c = \begin{bmatrix}
\gamma_a/S & 0 \\
0 & \gamma_b/S \\
0 & (1 - \gamma_b)/S \\
1 - \gamma_a/S & 0
\end{bmatrix},
\]

\[
C_c = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix},
\]

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

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**

\[
\begin{align*}
x_{11}(k) &= (x_1(k), x_3(k))^T, \\
y_{11}(k) &= (x_1(k))^T, \\
u(k) &= (u_1(k), u_2(k))^T, \\
x_{11}(k + 1) &= A_{c1}x_{11}(k) + B_{c1}u(k) \\
&= A_{c1}x_{11}(k) + B_{c1}^{(1)}u_1(k) + B_{c1}^{(2)}u_2(k), \\
y_{11}(k) &= C_{c1}x(k).
\end{align*}
\]

**Subsystem 2**

\[
\begin{align*}
x_{12}(k) &= (x_2(k), x_4(k))^T, \\
y_{12}(k) &= (x_2(k))^T, \\
u(k) &= (u_1(k), u_2(k))^T, \\
x_{12}(k) &= A_{c2}x_{12}(k) + B_{c2}u(k) \\
&= A_{c2}x_{12}(k) + B_{c2}^{(1)}u_1(k) + B_{c2}^{(2)}u_2(k), \\
y_{12}(k) &= C_{c2}x(k).
\end{align*}
\]

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

\[
\begin{bmatrix}
P_{x,i}(k+1) \\
P_{y,i}(k+1)
\end{bmatrix} = \begin{bmatrix}
P_{x,i}(k) \\
P_{y,i}(k)
\end{bmatrix} + T_s \begin{bmatrix}
v_{x,i}(k) \\
v_{y,i}(k)
\end{bmatrix},
\]

(35)

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

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

\[
x_i(k + 1) = Fx_i(k),
\]

(36)

where

\[
F = \begin{bmatrix}
1 & 0 & T_s & 0 \\
0 & 1 & 0 & T_s \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\]

(37)

The objective of the whole system is to track a target with \( N \) robots and to keep the distance 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 \( i \)th robot can be selected as

\[
J_i(k) = (\|P_i(k) - P_j(k)\| - R)^2.
\]

(38)
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 $Q_i = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}$, $R_i = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$, $N_{p_i} = 4$, $N_{u_i} = 4$ ($i = 1, 2, 3$). 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 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 $E_{rx}$:

$$E_{tx}(i, j) = (\alpha_1 + \alpha_2 d(i, j)^3) r,$$

where $d(i, j)$ is the distance between the two robots, $n$ is the path loss index, $r$ is a transmitting data rate, and $\alpha_1, \alpha_2$ 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.

### Table 1: Metrics comparisons among different algorithms.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Centralized MPC</th>
<th>Distributed MPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solution</td>
<td>Optimal</td>
<td>Nash optimal</td>
</tr>
<tr>
<td>Robustness</td>
<td>Central node failure leads to system down</td>
<td>Good</td>
</tr>
<tr>
<td>Information</td>
<td>Large</td>
<td>Small</td>
</tr>
<tr>
<td>Calculation load</td>
<td>Large</td>
<td>Smaller</td>
</tr>
</tbody>
</table>

### Table 2: Parameters of the four-tank system.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1, h_2, h_3, h_4$</td>
<td>Water level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$a_1$</td>
<td>$1.31 \times 10^{-3}$</td>
<td>m$^2$</td>
<td>Discharge constant of tank 1</td>
</tr>
<tr>
<td>$a_2$</td>
<td>$1.51 \times 10^{-3}$</td>
<td>m$^2$</td>
<td>Discharge constant of tank 2</td>
</tr>
<tr>
<td>$a_3$</td>
<td>$9.27 \times 10^{-4}$</td>
<td>m$^2$</td>
<td>Discharge constant of tank 3</td>
</tr>
<tr>
<td>$a_4$</td>
<td>$8.82 \times 10^{-5}$</td>
<td>m$^2$</td>
<td>Discharge constant of tank 4</td>
</tr>
<tr>
<td>$q_a$</td>
<td>$1.63 \times 10^{-3}$</td>
<td>m$^3$/h</td>
<td>Flow a</td>
</tr>
<tr>
<td>$q_b$</td>
<td>$2.00 \times 10^{-3}$</td>
<td>m$^3$/h</td>
<td>Flow b</td>
</tr>
<tr>
<td>$y_a$</td>
<td>0.3</td>
<td></td>
<td>Ratio of the three-way valve of pump a</td>
</tr>
<tr>
<td>$y_b$</td>
<td>0.4</td>
<td></td>
<td>Ratio of the three-way valve of pump b</td>
</tr>
<tr>
<td>$g$</td>
<td>$9.8 \times 3600 \times 3600$</td>
<td>m$/h^2$</td>
<td></td>
</tr>
</tbody>
</table>
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 $\Delta U_i(k | k) = [\Delta u_i(k | k), \Delta u_i(k + 1 | k), \ldots, \Delta u_i(k + N_u - 1 | k)]$, 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.
Figure 11: Four snapshots in target tracking scenario.

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**


Submit your manuscripts at http://www.hindawi.com