Distributed MPC for Reconfigurable Architecture Systems via Alternating Direction Method of Multipliers

Ting Bai, Shaoyuan Li, *Senior Member, IEEE*, and Yuanyuan Zou, *Member, IEEE*

***Abstract*—This paper investigates the distributed model predictive control (MPC) problem of linear systems where the network topology is changeable by the way of inserting new subsystems, disconnecting existing subsystems, or merely modifying the couplings between different subsystems. To equip live systems with a quick response ability when modifying network topology, while keeping a satisfactory dynamic performance, a novel reconfiguration control scheme based on the alternating direction method of multipliers (ADMM) is presented. In this scheme, the local controllers directly influenced by the structure realignment are redesigned in the reconfiguration control. Meanwhile, by employing the powerful ADMM algorithm, the iterative formulas for solving the reconfigured optimization problem are obtained, which significantly accelerate the computation speed and ensure a timely output of the reconfigured optimal control response. Ultimately, the presented reconfiguration scheme is applied to the level control of a benchmark four-tank plant to illustrate its effectiveness and main characteristics.**

***Index Terms*—Alternating direction method of multipliers (ADMM) algorithm, distributed control, model predictive control (MPC), reconfigurable architecture systems.**

I. Introduction

IN the past few decades, the advanced distributed model predictive control (DMPC) has received significant attention in both academia and industry for its outstanding advantages of dealing with multiple input/state constraints, better fault tolerance capabilities and a lower computational load. For geographically or physically distributed systems, DMPC provides an effective tool for control by effectively coordinating the optimal decisions of different local controllers with a guaranteed global performance. To date, numerous achievements in the research of DMPC strategies have been reached under fixed distributed frameworks [1]–[7].

Manuscript received May 10, 2019; revised September 4, 2019; accepted October 24, 2019. This work was support by the National Natural Science Foundation of China (61833012, 61773162, 61590924) and the Natural Science Foundation of Shanghai (18ZR1420000). Recommended by Associate Editor Xiaoping Liu. *(Corresponding author: Shaoyuan Li.)*

Citation: T. Bai, S. Y. Li, and Y. Y. Zou, "Distributed MPC for reconfigurable architecture systems via alternating direction method of multipliers," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 7, pp. 1336–1344, Jul. 2021.

The authors are with the Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China (e-mail: number127578@sjtu.edu.cn; syli@sjtu.edu.cn; yuanyzou@sjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2020.1003195

However, along with the system scale expansion and the continual function improvement, various application demands for the system structure reconstruction have risen, which incurs the issue of reconfiguration control. For instance, some partial failures in distributed systems [8], [9] may trigger the fault isolation to avoid a further fault propagation. Another example is the function extension to existing systems [10], which may bring in the insertion of several new components to existent systems. In order to achieve a dependable and renewed system functionality, it is of vital importance to study how to perform an efficient distributed control in the reconfigured architecture while maintaining a satisfactory dynamic performance.

Indeed, reconfiguration control with non-fixed system architectures has become an essential capability [11], [12] for networked systems to adapt to various external environments and a switching control task. Nowadays, research on the high-efficiency distributed reconfiguration control strategies is still lacking. In the existing study [13], a reconfigurable control scheme for decentralized systems was proposed, where the coupling changes between different subsystems were handled by resorting to robust control invariant sets. For distributed systems with strong couplings, [14] presented an iterative reconfiguration scheme to cope with the significant interactions among subsystems and a class of time-varying terminal sets was designed. Notwithstanding, most of the existing reconfiguration control methodologies are developed based on the assumption that every reconfigurable control signal can be obtained within each sampling time. That is to say, it is based on the assumption that each control period is shorter than each sampling period, and the elapsed time spent in calculating the reconfiguration control input can be ignored. However, the computation of the optimal reconfigurable control law is time consuming in substantial real-world applications (especially for large-scale MPC optimization problems with multiple constraints), and the calculation efficiency of local controllers directly affects the distributed coordination of reconfiguration control. For the sake of avoiding a serious trajectory deviation in the context of system architecture realignment, the quick response ability of every local controller is of particular importance in the distributed reconfiguration control design.

As is known, the alternating direction method of multipliers (ADMM) is a high-efficiency iterative algorithm, which was originally introduced by Gabay and Mercier [15] in the 1970s and is well suited to large-scale convex optimization problems arising in statistics, machine learning, and other related areas. ADMM blends the benefits of dual decomposition and augmented Lagrangian algorithms for the constrained optimization and provides improved convergence properties under very mild hypotheses [16]–[20]. More precisely, the major advantages of ADMM can be summarized as follows: 1) in theory, the convergence of the algorithm can be guaranteed for any convex cost functions and constraints; 2) in practice, the augmented Lagrangian term often speeds up the convergence and computation. Although the powerful ADMM has been widely applied to solving convex optimization problems in the field of statistics, there is little research on the combination of ADMM and the reconfigurable DMPC scheme [21] to improve the computation efficiency and dynamic performance of reconfiguration control. Motivated by these reasons, in this technical note, we aim at proposing an efficient DMPC reconfiguration control scheme based on the powerful ADMM algorithm for linear networked systems with reconfigurable architectures. First, the distributed reconfiguration control strategy is presented with the consideration of three typical scenarios on the changeable system architecture. Next, the reconfigured controller design method is proposed and the iterative formulas based on ADMM algorithm for addressing the reconfiguration optimization problem are derived. Finally, the efficiency of the developed reconfigurable DMPC scheme is verified on the level control of a benchmark four-tank system.

The rest of the paper is organized as follows. A formal problem formulation is provided in Section II. In Section III, we introduce the distributed reconfigurable control scheme, including the reconfiguration control strategy, the way of reconfigured controller redesign and the resolving procedure of the reconfiguration control problem based on ADMM algorithm. Section IV applies the proposed reconfigured control scheme to a four-tank system, followed by some concluding remarks in Section V.

Notations: Let $\mathbb{R}$, $\mathbb{R}^n$, $\mathbb{R}^{n\times m}$ denote the set of real numbers, the vector space of real n-vectors and the set of $n\times m$ real matrices, respectively. For any given symmetric positive-definite matrix Q, $\|x\|_Q$ denotes the weighted Euclidean norm of x, i.e., $\|x\|_Q = \sqrt{x^TQx}$. By $\mathrm{diag}\{S_1,\ldots,S_k\}$, we refer to a block-diagonal matrix with matrices S_1 to S_k on the main diagonal while zeros elsewhere. $\times$ denotes the Cartesian product of two sets while $\otimes$ represents the Kronecker product of two matrices. The symbols $\mathcal{I}^m \in \mathbb{R}^m$ and $\mathcal{O}^n \in \mathbb{R}^n$ stand for the column vectors with all elements equal to one and zero, respectively.

II. Problem Formulation

Consider a distributed LTI (linear time-invariant) system composed by M subsystems which are interconnected with each other through states and inputs. The local subsystem $\mathcal{S}_i$ under a fixed system architecture is described as

$$\Sigma_{\mathcal{S}_i} : x_{[i]}^+ = \sum_{j=1}^{M}(A_{ij}x_{[j]} + B_{ij}u_{[j]}) \tag{1}$$

where $x_{[i]} \in \mathbb{R}^{n_i}$ and $u_{[i]} \in \mathbb{R}^{m_i}$, $\forall i \in \mathcal{M} = \{1,\ldots,M\}$ are the state and input of ith subsystem. $x_{[i]}^+$ denotes the successor state of $x_{[i]}$ at time $(k+1)$. In terms of each subsystem, the local subsystem state $x_{[i]}$ is directly effected by the states of $\{\mathcal{S}_j\}_{j\in\mathcal{P}_i^x}$ and the inputs of $\{\mathcal{S}_r\}_{r\in\mathcal{P}_i^u}$. $\mathcal{P}_i^x$ and $\mathcal{P}_i^u$ represent the state parent set and input parent set of subsystem $\mathcal{S}_i$, respectively, which are determined by

$$\mathcal{P}_i^x = \{j : A_{ij} \neq 0, j \neq i, \ \forall i, j \in \mathcal{M}\}$$
$$\mathcal{P}_i^u = \{r : B_{ir} \neq 0, r \neq i, \ \forall i, r \in \mathcal{M}\}.$$

Here we use $\mathcal{P}_i = \mathcal{P}_i^x \cup \mathcal{P}_i^u$ to denote the parent set of subsystem $\mathcal{S}_i$. On this basis, we say subsystem $\mathcal{S}_j$ belongs to the state child set $\mathcal{C}_i^x$ of subsystem $\mathcal{S}_i$ if $i \in \mathcal{P}_j^x$ holds true. In a similar way, the input child set $\mathcal{C}_i^u$ of subsystem $\mathcal{S}_i$ is defined. The state and input child sets of subsystem $\mathcal{S}_i$ are characterized as follows

$$\mathcal{C}_i^x = \{j : i \in \mathcal{P}_j^x, \ \forall i, j \in \mathcal{M}\}$$
$$\mathcal{C}_i^u = \{j : i \in \mathcal{P}_j^u, \ \forall i, j \in \mathcal{M}\}.$$

Then, the child set $\mathcal{C}_i$ of subsystem $\mathcal{S}_i$ is expressed as $\mathcal{C}_i = \mathcal{C}_i^x \cup \mathcal{C}_i^u \cup i$. According to the above definitions, the local dynamics (1) can be reformulated as

$$\Sigma_{\mathcal{S}_i} : x_{[i]}^+ = A_{ii}x_{[i]} + B_{ii}u_{[i]} + \sum_{j\in\mathcal{P}_i^x} A_{ij}x_{[j]} + \sum_{j\in\mathcal{P}_i^u} B_{ij}u_{[j]} \tag{2}$$

where the local subsystem state $x_{[i]}$ and the input $u_{[i]}$ are subject to $\mathcal{X}_i$ and $\mathcal{U}_i$ with $\mathcal{X}_i = \{x_{[i]} : x_{[i,\min]} \leq x_{[i]} \leq x_{[i,\max]}\}$ and $\mathcal{U}_i = \{u_{[i]} : u_{[i,\min]} \leq u_{[i]} \leq u_{[i,\max]}\}$. Here $\mathcal{X}_i$ and $\mathcal{U}_i$ are compact convex sets with the origin contained as their interior points.

Constructed by the subsystem state and input variables, the global system state $x \in \mathbb{R}^n$ and input $u \in \mathbb{R}^m$ are denoted by $x = [x_{[1]}^T,\ldots,x_{[M]}^T]^T$ and $u = [u_{[1]}^T,\ldots,u_{[M]}^T]^T$. Then, the global system model can be represented as

$$\Sigma_{\mathcal{S}} : x^+ = Ax + Bu. \tag{3}$$

The global system state and input are confined by $x \in \mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_M$ and $u \in \mathcal{U} = \mathcal{U}_1 \times \cdots \times \mathcal{U}_M$. Meanwhile, the dynamic matrices A and B of the overall system are attained in light of the following form

$$A = \begin{bmatrix} A_{11} & \ldots & A_{1M} \\ \vdots & \ddots & \vdots \\ A_{M1} & \ldots & A_{MM} \end{bmatrix}, \ B = \begin{bmatrix} B_{11} & \ldots & B_{1M} \\ \vdots & \ddots & \vdots \\ B_{M1} & \ldots & B_{MM} \end{bmatrix}.$$

In the existing system architecture and control framework, we assume that the initial global system represented by (A, B) is controllable. Meanwhile, every local controller is designed according to the traditional non-cooperative DMPC strategy, i.e., a local performance index is minimized in determining the optimal control input. The initial DMPC optimization problem of subsystem $\mathcal{S}_i$, $i \in \mathcal{M}$ is expressed as

$$\min_{u_{[i]}(k+h|k)_{0\leq h\leq N-1}} \sum_{h=0}^{N-1}(\|x_{[i]}(k+h|k)\|_{Q_i} + \|u_{[i]}(k+h|k)\|_{R_i}) + \|x_{[i]}(k+N|k)\|_{P_i} \tag{4a}$$

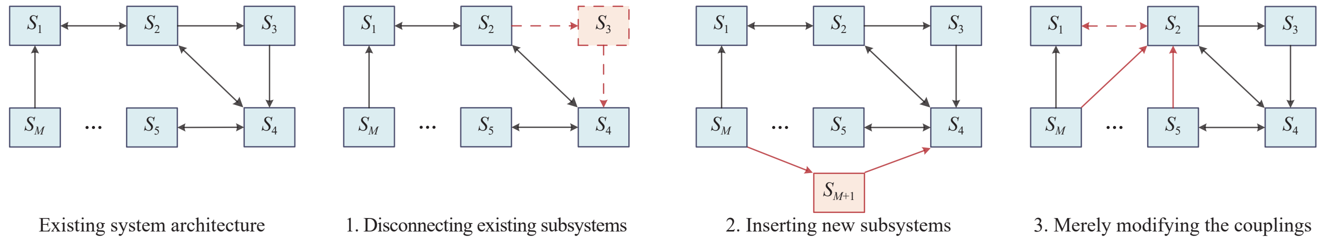


Fig. 1. Three typical reconfiguration scenarios. The red dotted lines denote the interconnections to be disconnected while the red solid lines represent the associated relationships to be newly established.

$$\text{s.t.} \quad x_{[i]}(k|k) = x_{[i]}(k) \tag{4b}$$

$$x_{[i]}^+ = A_{ii}x_{[i]} + B_{ii}u_{[i]} + \sum_{j\in\mathcal{P}_i^x} A_{ij}x_{[j]} + \sum_{j\in\mathcal{P}_i^u} B_{ij}u_{[j]} \tag{4c}$$

$$x_{[i]} \in \mathcal{X}_i \tag{4d}$$

$$u_{[i]} \in \mathcal{U}_i \tag{4e}$$

$$x_{[i]}(k+N|k) \in \mathcal{X}_i^f \tag{4f}$$

where $x_{[i]}(k+h|k)$, $u_{[i]}(k+h|k)$ denote the values of $x_{[i]}$ and $u_{[i]}$ at the future time $(k+h)$ predicted at the time k. The terminal invariant set $\mathcal{X}_i^f$ is devised with guaranteed stability of the global system in the existing system structure.

Throughout this context, the communication protocol in controller design is provided by the following assumption.

Assumption 1: The communication between two interconnected local subsystems $\mathcal{S}_i$ and $\mathcal{S}_j$ is permitted if and only if $i \in \mathcal{P}_j$ or $i \in \mathcal{C}_j$.

In line with different reconfiguration demands in real applications, we focus on three typical reconfigurable scenarios in this paper: inserting new subsystems, disconnecting existing subsystems, and modifying the couplings between different local subsystems. As described earlier, inserting several new subsystems generally maps the requirement of system function extension, troubleshooting or subsystem recovery. Disconnecting existing subsystems often corresponds to failure isolation; modifying the couplings between interconnected subsystems arises from the need of system structure improvement. A more clear illustration to the three typical reconfiguration scenarios is given by Fig. 1. Note that the three scenarios cover the basic reconfiguration modes in a distributed system framework, and other complicated reconfigurable architecture requirements can be viewed as a combination of them.

In the sequel, we propose to investigate the DMPC reconfiguration control scheme by employing the powerful ADMM algorithm with respect to the above three typical reconfiguration scenarios. It is attributed to achieving a quick control response capability and a satisfactory dynamic performance in the reconfigurable distributed control.

III. Reconfiguration Control Scheme

This section introduces the DMPC reconfiguration control scheme for linear systems with reconfigurable architectures. Specifically, in the first place, the reconfiguration control strategy in terms of three typical reconfiguration scenarios is presented. Secondly, we show how to redesign the local controllers for some selected subsystems in the reconfigured control. Then, the way to resolve the reconfiguration optimization problem via the powerful ADMM algorithm is provided, which equips local subsystems with a quick-response ability to dynamical structure changes and helps to achieve a good distributed coordination during the whole reconfiguration control.

A. Distributed Reconfiguration Strategy

Due to the realignment of system architectures, the dynamics and interconnections of partial local subsystems are changed compared with those in the original system architecture. Here, we first provide a detailed analysis to the influence of three typical reconfiguration scenarios to every local subsystem. Later, the reconfiguration control strategy is presented accordingly.

First, in the renewed system architecture, A^{re} and B^{re} are utilized to represent the reconfigured system matrix and input matrix, respectively. Then, the dynamics of the reconfigured global system can be described as

$$\Sigma_{\mathcal{S}^{re}} : \tilde{x}^+ = A^{re}\tilde{x} + B^{re}\tilde{u} \tag{5}$$

where $\tilde{x} \in \mathbb{R}^{\tilde{n}}$ and $\tilde{u} \in \mathbb{R}^{\tilde{m}}$ are the global system state and input signals in the reconfigured system architecture, respectively. For the sake of achieving a control reconstruction to reconfigured systems, the following assumption is necessary.

Assumption 2: The global system in the reconfigured system architecture and control configuration is still controllable, i.e., (A^{re}, B^{re}) is stabilizable.

Now consider that, a reconfigured system $\mathcal{S}^{re}$ consists of T subsystems in the renewed system architecture. Meanwhile, we denote by $\tilde{\mathcal{P}}_i^x$ and $\tilde{\mathcal{P}}_i^u$ the updated state parent set and input parent set of subsystem $\mathcal{S}_i$, respectively. Taking this as a basis, the dynamics of every local subsystem in the reconfigured system architecture is denoted by

$$\begin{aligned}\Sigma_{\mathcal{S}_i^{re}} : \tilde{x}_{[i]}^+ &= \sum_{j=1}^{T}(A_{ij}^{re}\tilde{x}_{[j]} + B_{ij}^{re}\tilde{u}_{[j]}) \\ &= A_{ii}\tilde{x}_{[i]} + B_{ii}\tilde{u}_{[i]} + \sum_{j\in\tilde{\mathcal{P}}_i^x} A_{ij}^{re}\tilde{x}_{[j]} + \sum_{j\in\tilde{\mathcal{P}}_i^u} B_{ij}^{re}\tilde{u}_{[j]}\end{aligned} \tag{6}$$

where $\tilde{x}_{[i]} \in \mathbb{R}^{\tilde{n}_i}$ and $\tilde{u}_{[i]} \in \mathbb{R}^{\tilde{m}_i}$ are the local state and input of subsystem $\mathcal{S}_i^{re}$ in the renewed architecture, respectively. Note that, since the inner structure of every local subsystem is not changed, we have $A_{ii} = A_{ii}^{re}$ and $B_{ii} = B_{ii}^{re}$. Meanwhile, the reconfigured local state and input are respectively confined by $\tilde{x}_{[i]} \in \tilde{\mathcal{X}}_i$ and $\tilde{u}_{[i]} \in \tilde{\mathcal{U}}_i$, $\forall i \in \mathcal{T} = \{1,\ldots,T\}$. For every extraneous subsystem to be newly inserted into the existing system architecture, the assumption below holds true.

Assumption 3: The subsystems to be inserted into the initial system architecture are controllable by their independent controllers, i.e., (A_{ii}, B_{ii}) is stabilizable for any $i \in \mathcal{T} - \mathcal{M}$.

Next, aiming at three typical reconfiguration scenarios, a reconfiguration strategy is proposed which determines the local subsystems whose controllers need to be redesigned in the further reconfigured control. In order to minimize changes to the existing controller designs while realizing an effective control of the global reconfigured system, the directly influenced subsystems are selected to re-conceive their local controllers. Specifically, taking into account the structure modification by means of subsystems removal, subsystems addition or coupling realignments between different subsystems, the directly influenced subsystems are referred to as the subsystems whose reconfigured subsystem dynamics is different from the original subsystem dynamics in the initial system architecture. By (1) and (6), the set of directly influenced subsystems $\mathcal{R}$ is determined by

$$\mathcal{R} = \{i : \tilde{\mathcal{P}}_i^x \neq \mathcal{P}_i^x \text{ or } \tilde{\mathcal{P}}_i^u \neq \mathcal{P}_i^u, \forall i \in \mathcal{T}\}. \tag{7}$$

During the whole reconfiguration control, the controllers of directly influenced subsystems are redesigned while the other subsystems in the initial system structure keep using the original controller designs in the reconfigured control.

Remark 1: For every subsystem $\mathcal{S}_i^{re}$, $i \in \mathcal{T} - \mathcal{M}$ to be newly inserted into the existing system architecture, we have $\mathcal{P}_i^x = \mathcal{P}_i^u = \emptyset$ holds true. Moreover, if the state parent set and input parent set of subsystems $\mathcal{S}_i^{re}$ with $i \in \mathcal{T} - \mathcal{M}$ are empty in the reconfigured system structure, i.e., $\tilde{\mathcal{P}}_i^x = \tilde{\mathcal{P}}_i^u = \emptyset$, then these subsystems $\mathcal{S}_i^{re}$ are definitely not included in the set of directly influenced subsystems according to (7). In this case, the original controller design of the subsystem $\mathcal{S}_i^{re}$, $i \in \mathcal{T} - \mathcal{M}$ is employed in the further reconfiguration control.

B. Reconfigurable Controller Redesign

Based on the set of directly influenced subsystems determined above, in the following, the way of reconfigurable controller redesign for subsystems $\mathcal{S}_i^{re}$ with $i \in \mathcal{R}$ is proposed. For any subsystem $\mathcal{S}_i^{re}$ in the renewed system architecture, the reconfigured optimization problem of the local controller redesign is formulated by

$$\min_{\tilde{u}_{[i]}(k+h|k)_{0 \le h \le N-1}} \sum_{h=0}^{N-1} (\|\tilde{x}_{[i]}(k+h|k)\|_{\tilde{Q}_i} + \|\tilde{u}_{[i]}(k+h|k)\|_{\tilde{R}_i}) + \|\tilde{x}_{[i]}(k+N|k)\|_{\tilde{P}_i} \tag{8a}$$

$$\text{s.t.} \quad \tilde{x}_{[i]}(k|k) = \tilde{x}_{[i]}(k) \tag{8b}$$

$$\tilde{x}_{[i]}^+ = A_{ii}\tilde{x}_{[i]} + A_{ic}^{re}\tilde{x}_{ic} + B_{ii}\tilde{u}_{[i]} + B_{ic}^{re}\tilde{u}_{ic} \tag{8c}$$

$$\tilde{x}_{[i]} \in \tilde{\mathcal{X}}_i \tag{8d}$$

$$\tilde{u}_{[i]} \in \tilde{\mathcal{U}}_i \tag{8e}$$

$$\tilde{x}_{[i]}(k+N|k) \in \tilde{\mathcal{X}}_i^f \tag{8f}$$

where we denote by $\tilde{x}_{[i]}(k+h|k)$ the value of vector $\tilde{x}_{[i]}$ at the future time $(k+h)$ predicted at the time instant k. So is the input $\tilde{u}_{[i]}(k+h|k)$. $\tilde{Q}_i$ and $\tilde{R}_i$ are the weighting matrices of the directly influenced subsystems $\mathcal{S}_i^{re}$ with $i \in \mathcal{R}$. Meanwhile, $\tilde{\mathcal{X}}_i^f = \tilde{x}_{[i]}(k+N|k)^T \tilde{P}_i \tilde{x}_{[i]}(k+N|k) \le \tilde{\beta}_i$ is the terminal invariant set of the reconfigured subsystem $\mathcal{S}_i^{re}$, which is designed according to the same techniques used in the initial distributed control system for ensuring the controllability of the global reconfigured system. Moreover, in (8c), the associated items between different interconnected subsystems have the particular form of $A_{ic}^{re}\tilde{x}_{ic} = \sum_{j \in \tilde{\mathcal{P}}_i^x} A_{ij}^{re}\tilde{x}_{[j]}$ and $B_{ic}^{re}\tilde{u}_{ic} = \sum_{j \in \tilde{\mathcal{P}}_i^u} B_{ij}^{re}\tilde{u}_{[j]}$.

To equip live systems with a rapid response ability to various reconfiguration requirements and then achieve a good distributed coordination, we next provide an approach to transform the reconfigurable control problem (8) into a standard optimization formulation for employing the ADMM algorithm.

First, we denote by $\tilde{\boldsymbol{x}}_{[i]} \in \mathbb{R}^{Nn_i}$ and $\tilde{\boldsymbol{u}}_{[i]} \in \mathbb{R}^{Nm_i}$ in MPC expression, which include the predicted states and inputs and have the following form

$$\tilde{\boldsymbol{x}}_{[i]}(k) = \begin{bmatrix} \tilde{x}_{[i]}(k+1|k) \\ \tilde{x}_{[i]}(k+2|k) \\ \vdots \\ \tilde{x}_{[i]}(k+N|k) \end{bmatrix}, \ \tilde{\boldsymbol{u}}_{[i]}(k) = \begin{bmatrix} \tilde{u}_{[i]}(k|k) \\ \tilde{u}_{[i]}(k+1|k) \\ \vdots \\ \tilde{u}_{[i]}(k+N-1|k) \end{bmatrix}.$$

By this form, the reconfigured subsystem dynamics (8c) can be represented by

$$\tilde{\boldsymbol{x}}_{[i]}(k+1) = \Psi_i \tilde{\boldsymbol{x}}_{[i]}(k) + \Psi_{ic}\tilde{\boldsymbol{x}}_{ic}(k) + \Phi_i \tilde{\boldsymbol{u}}_{[i]}(k) + \Phi_{ic}\tilde{\boldsymbol{u}}_{ic}(k)$$

where $\tilde{\boldsymbol{x}}_{ic}$ and $\tilde{\boldsymbol{u}}_{ic}$ denote the associated state and input of subsystem $\mathcal{S}_i^{re}$ in MPC expression. Additionally, all the system matrices involved in the above equation are determined by

$$\Psi_i = [A_{ii}^T, A_{ii}^2, \ldots, A_{ii}^N]^T$$

$$\Phi_i = \begin{bmatrix} B_{ii} & \mathbf{0} & \ldots & \mathbf{0} \\ A_{ii}B_{ii} & B_{ii} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ A_{ii}^{N-1}B_{ii} & A_{ii}^{N-2}B_{ii} & \ldots & B_{ii} \end{bmatrix}$$

$$\Psi_{ic} = \begin{bmatrix} A_{ic}^{re} & \mathbf{0} & \ldots & \mathbf{0} \\ A_{ii}A_{ic}^{re} & A_{ic}^{re} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ A_{ii}^{N-1}A_{ic}^{re} & A_{ii}^{N-2}A_{ic}^{re} & \ldots & A_{ic}^{re} \end{bmatrix}$$

$$\Phi_{ic} = \begin{bmatrix} B_{ic}^{re} & \mathbf{0} & \ldots & \mathbf{0} \\ A_{ii}B_{ic}^{re} & B_{ic}^{re} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ A_{ii}^{N-1}B_{ic}^{re} & A_{ii}^{N-2}B_{ic}^{re} & \ldots & B_{ic}^{re} \end{bmatrix}.$$

Then, in ADMM form, the reconfigurable control problem (8) can be transformed into the following standard expression

$$\min_{\tilde{\boldsymbol{u}}_{[i]}(k)} \ \frac{1}{2}\tilde{\boldsymbol{u}}_{[i]}^T(k) p_i \tilde{\boldsymbol{u}}_{[i]}(k) + q_i^T \tilde{\boldsymbol{u}}_{[i]}(k) + r_i \tag{9a}$$

$$\text{s.t.} \quad \tilde{\boldsymbol{u}}_{[i]} \in \tilde{\mathcal{U}}_i. \tag{9b}$$

In the cost function of the optimization problem (9a), the parameters p_i, q_i and r_i are respectively given by

$$\begin{aligned} p_i &= 2(\Phi_i^T \hat{Q}_i \Phi_i + \hat{R}_i) \\ q_i &= 2[F_i \tilde{x}_i(k) + \Lambda_i \tilde{\boldsymbol{u}}_{ic}(k) + E_i \tilde{\boldsymbol{x}}_{ic}(k)] \\ r_i &= \tilde{x}_i^T(k) G_i \tilde{x}_i(k) + 2[\tilde{x}_i^T(k) F_{ic}^T + \tilde{\boldsymbol{x}}_{ic}^T(k) E_{ic}^T] \tilde{\boldsymbol{u}}_{ic}(k) \\ &\quad + [2\tilde{x}_i^T(k) \Upsilon_i + \tilde{\boldsymbol{x}}_{ic}^T(k) \Upsilon_{ic}] \tilde{\boldsymbol{x}}_{ic}(k) + \tilde{\boldsymbol{u}}_{ic}^T(k) \Theta_i \tilde{\boldsymbol{u}}_{ic}(k). \end{aligned}$$

Note that, the block diagonal matrices $\hat{Q}_i$ and $\hat{R}_i$ have the form of $\hat{Q}_i = \mathrm{diag}\{\underbrace{\tilde{Q}_i, \ldots, \tilde{Q}_i}_{N-1}, \tilde{P}_i\}$ and $\hat{R}_i = \mathrm{diag}\{\underbrace{\tilde{R}_i, \ldots, \tilde{R}_i}_{N}\}$. In addition, the other matrices included in the specific expression of q_i and r_i are provided as follows:

$$\begin{aligned} &F_i = \Phi_i^T \hat{Q}_i \Psi_i, && \Lambda_i = \Phi_i^T \hat{Q}_i \Phi_{ic}, && E_i = \Phi_i^T \hat{Q}_i \Psi_{ic} \\ &G_i = \Psi_i^T \hat{Q}_i \Psi_i + \tilde{Q}_i, && F_{ic} = \Phi_{ic}^T \hat{Q}_i \Psi_i, && E_{ic} = \Phi_{ic}^T \hat{Q}_i \Psi_{ic} \\ &\Upsilon_i = \Psi_i^T \hat{Q}_i \Psi_{ic}, && \Upsilon_{ic} = \Psi_{ic}^T \hat{Q}_i \Psi_{ic}, && \Theta_i = \Phi_{ic}^T \hat{Q}_i \Phi_{ic}. \end{aligned}$$

In (9b), the constraint space $\tilde{\mathcal{U}}_i$ is denoted by $\tilde{\mathcal{U}}_i = \{\tilde{\boldsymbol{u}}_{[i]} : \Gamma_i \tilde{\boldsymbol{u}}_{[i]} \le \Omega_i\}$, where Γ_i and Ω_i are determined by

$$\Gamma_i = \begin{bmatrix} \Phi_i \\ \mathcal{I}_i^{Nm_i} \\ -\Phi_i \\ -\mathcal{I}_i^{Nm_i} \end{bmatrix}, \; \Omega_i = \begin{bmatrix} \tilde{\boldsymbol{x}}_{[i,\max]} - \Psi_i \tilde{x}_i(k) - \Xi_{ic} \tilde{\boldsymbol{u}}_{[i,\max]} \\ \Psi_i \tilde{x}_i(k) + \Xi_{ic} - \tilde{\boldsymbol{x}}_{[i,\min]} - \tilde{\boldsymbol{u}}_{[i,\min]} \end{bmatrix}$$

$$\Xi_{ic} = \Psi_{ic} \tilde{\boldsymbol{x}}_{ic}(k) + \Phi_{ic} \tilde{\boldsymbol{u}}_{ic}(k).$$

The lower and upper bounds to the reconfigured subsystem states are expressed as

$$\tilde{\boldsymbol{x}}_{[i,\min]} = \mathcal{I}^N \otimes \tilde{x}_{[i,\min]}, \quad \tilde{\boldsymbol{x}}_{[i,\max]} = \mathcal{I}^N \otimes \tilde{x}_{[i,\max]}.$$

Using a similar way, the bounds to every local input $\tilde{\boldsymbol{u}}_{[i,\min]}$ and $\tilde{\boldsymbol{u}}_{[i,\max]}$ can be denoted. It is worth mentioning that, the constraint $\tilde{\mathcal{U}}_i$ is time-varying since the associated states $\tilde{\boldsymbol{x}}_{ic}(k)$ and inputs $\tilde{\boldsymbol{u}}_{ic}(k)$ are dynamically changed in each communication cycle, which subsequently changes the value of Ξ_{ic} in solving the optimization problem (9a).

C. Solution of the Reconfigured Control via ADMM

Hereafter, the method for solving the reconfigured optimization problem (9) by resorting to the powerful ADMM algorithm is proposed. Generally, the constrained DMPC problem can be addressed by employing the fmincon function in Matlab if the available calculation time is sufficient. For many cases of the instant structure reconfiguration requirements, such as the isolation of several failed subsystems to prevent a further fault propagation, the time to prepare the reconfiguration control is quite limited. Furthermore, the slower the reconfiguration control response, the harder it is to achieve a satisfactory distributed coordination and the greater the system state deviation in the renewed system architecture. For these reasons, below we intend to solve the reconfigured DMPC problem (9) via the high-efficiency ADMM algorithm, which helps to significantly improve the computational efficiency of the reconfigured control response and then achieve a satisfactory reconfiguration control property.

First, taking into account the general form of ADMM in solving optimization problems

$$\min \; f(x) + g(z) \tag{10a}$$

$$\text{s.t.} \; Ax + Bz = c. \tag{10b}$$

For the above general form of ADMM, the optimization problem (10a) is solvable in the precondition of the following two assumptions [22], which are used to guarantee the existence of x and z that can minimize the augmented Lagrangian function.

Assumption 4: The functions $f(x)$ and $g(z)$ are closed, proper and convex.

Assumption 5: For simple Lagrangian

$$L_0(x,z,y) = f(x) + g(y) + y^T(Ax + Bz - c)$$

there exists (x^*, z^*, y^*), not necessarily unique, for which $L_0(x^*, z^*, y) \le L_0(x^*, z^*, y^*) \le L_0(x, z, y^*)$ holds true for all x, z, y.

Under these two mild assumptions, the ADMM iteration is convergent and the augmented Lagrangian is organized by

$$\begin{aligned} L_\rho(x,z,y) = &f(x) + g(z) + y^T(Ax + Bz - c) \\ &+ \frac{\rho}{2}\|Ax + Bz - c\|_2^2. \end{aligned}$$

Then, by adopting the *scaled dual variable* $u = \frac{1}{\rho} y$, the ADMM iteration can be transformed into the following simplified form

$$x^{(k+1)} = \arg\min_x \left(f(x) + \frac{\rho}{2}\|Ax + Bz^{(k)} - c + u^{(k)}\|_2^2 \right) \tag{11a}$$

$$z^{(k+1)} = \arg\min_z \left(g(z) + \frac{\rho}{2}\|Ax^{(k+1)} + Bz - c + u^{(k)}\|_2^2 \right) \tag{11b}$$

$$u^{(k+1)} = u^{(k)} + Ax^{(k+1)} + Bz^{(k+1)} - c. \tag{11c}$$

On this basis, we aim to derive the iterative formulas that can be applied to solving the proposed reconfiguration control problem. In line with the general form of (10a) and (10b), the reconfigured DMPC optimization problem (9) can be re-described as the following form

$$\min_{\tilde{\boldsymbol{u}}_{[i]}} \quad J_i(\tilde{\boldsymbol{u}}_{[i]}) \tag{12a}$$

$$\text{s.t.} \; \Gamma_i \tilde{\boldsymbol{u}}_{[i]} + z_{[i]} = \boldsymbol{0} \tag{12b}$$

where $z_{[i]} = \boldsymbol{m}_{[i]} - \Omega_i$ is a newly introduced optimization variable and $\boldsymbol{m}_{[i]} \ge O_i$. In the sequel, the detailed procedures to derive the iterations for both the optimal variable and the dual variable based on the optimality conditions [20] are provided.

1) $\tilde{\boldsymbol{u}}_{[i]}$-update: For clarity, the scaled dual variable u is replaced by λ in the following derivation. Here we define that

$$\nu_i^{(k)} = -z_{[i]}^{(k)} - \lambda_i^{(k)}$$

which can be viewed as a constant in the minimization of $\tilde{\boldsymbol{u}}_{[i]}$. Then by minimizing (12a), the formula to update $\tilde{\boldsymbol{u}}_{[i]}$ is obtained as

$$\tilde{\boldsymbol{u}}_{[i]}^{(k+1)} = \arg\min_{\tilde{\boldsymbol{u}}_{[i]}} \left(J_i + \frac{\rho}{2}\|\Gamma_i \tilde{\boldsymbol{u}}_{[i]}^{(k)} - \nu_i^{(k)}\|_2^2 \right). \tag{13}$$

2) $z_{[i]}$-update: In the minimization of $z_{[i]}$, the constant variable is denoted by $\omega_i^{(k)}$, which has the form of

$$\omega_i^{(k)} = -\Gamma_i \tilde{\boldsymbol{u}}_{[i]}^{(k+1)} - \lambda_i^{(k)}.$$

Similarly, by minimizing (12a), we can obtain that

$$\begin{aligned} z_{[i]}^{(k+1)} &= \arg\min_{z_{[i]}}\left(\frac{\rho}{2}\|z_{[i]}^{(k)}-\omega_i^{(k)}\|_2^2\right) \\ &= \arg\min_{z_{[i]}}\left(\frac{\rho}{2}\|\boldsymbol{m}_{[i]}-\left(\Omega_i+\omega_i^{(k)}\right)\|_2^2\right). \end{aligned} \tag{14}$$

Since $\boldsymbol{m}_{[i]} \geq O_i$, there exist two possible consequences in the minimization of (14). More specifically, $z_{[i]}^{(k+1)}$ can be calculated in accordance with the following equations in different situations:

$$\begin{aligned} &\text{a) if } \ \omega_i^{(k)}+\Omega_i \geq O_i \implies z_{[i]}^{(k+1)}=\omega_i^{(k)} \\ &\text{b) if } \ \omega_i^{(k)}+\Omega_i < O_i \implies z_{[i]}^{(k+1)}=-\Omega_i. \end{aligned}$$

As a result, the iterative formulas to solve the reconfigurable DMPC optimization problem (12) via the powerful ADMM algorithm are obtained as

$$\tilde{\boldsymbol{u}}_{[i]}^{(k+1)} = (p_i+\rho\Gamma_i^T\Gamma_i)^{-1}(\rho\Gamma_i^T\nu_i^{(k)}-q_i) \tag{15a}$$

$$z_{[i]}^{(k+1)} = \max\{-(\Gamma_i\tilde{\boldsymbol{u}}_{[i]}^{(k+1)}+\lambda_i^{(k)}),-\Omega_i\} \tag{15b}$$

$$\lambda_i^{(k+1)} = \lambda_i^{(k)}+\Gamma_i\tilde{\boldsymbol{u}}_{[i]}^{(k+1)}+z_{[i]}^{(k+1)}. \tag{15c}$$

Furthermore, to improve the convergence property of ADMM [20], the quantity $\Gamma_i\tilde{\boldsymbol{u}}_{[i]}^{(k+1)}$ is generally substituted by

$$\alpha^{(k)}\Gamma_i\tilde{\boldsymbol{u}}_{[i]}^{(k+1)}-(1-\alpha^{(k)})z_{[i]}^{(k)}$$

where $\alpha^{(k)}>1$ is known as the over-relaxation parameter.

Meanwhile, as the iteration proceeds, the reasonable termination criterion is designed by

$$\|r_i^{(k)}\|_2 \leq \epsilon^{\text{pri}} \tag{16a}$$

$$\|s_i^{(k)}\|_2 \leq \epsilon^{\text{dual}} \tag{16b}$$

$$\tilde{x}_{[i]}(k+N|k) \in \tilde{\mathcal{X}}_i^f \tag{16c}$$

where $r_i^{(k)}$ is the primal residual and $s_i^{(k)}$ is the dual residual. $\|r_i^{(k)}\|_2 \in \mathbb{R}^p$ and $\|s_i^{(k)}\|_2 \in \mathbb{R}^q$. The iteration stops when all the conditions (16a)–(16c) are satisfied, where (16c) reflects the terminal constraint (8f). The residuals $r_i^{(k)}$ and $s_i^{(k)}$ are updated according to the equations given below

$$\begin{aligned} r_i^{(k+1)} &= \Gamma_i\tilde{\boldsymbol{u}}_{[i]}^{(k+1)}+z_{[i]}^{(k+1)} \\ s_i^{(k+1)} &= \rho\Gamma_i^T(z_{[i]}^{(k+1)}-z_{[i]}^{(k)}). \end{aligned}$$

Additionally, the thresholds of the primal residual ϵ^{pri} and the dual residual ϵ^{dual} are respectively designed as

$$\begin{aligned} \epsilon^{\text{pri}} &= \sqrt{p}\epsilon^{\text{abs}}+\epsilon^{\text{rel}}\max\{\|\Gamma_i\tilde{\boldsymbol{u}}_{[i]}^{(k)}\|_2,\|z_i^{(k)}\|_2\} \\ \epsilon^{\text{dual}} &= \sqrt{q}\epsilon^{\text{abs}}+\epsilon^{\text{rel}}\|\Gamma_i^T\rho\lambda_i^{(k)}\|_2 \end{aligned}$$

where $\epsilon^{\text{abs}}>0$ and $\epsilon^{\text{rel}}>0$ are the absolute and relative tolerance, respectively. Eventually, based on the above results, one can obtain the optimal control response by solving the reconfigurable DMPC optimization problem (9) via iterations (15a)–(15c), which contribute to achieving a satisfactory reconfiguration control performance with a high computational efficiency. To give a more clear illustration to the proposed reconfiguration control scheme, an algorithm flow chart is provided in Fig. 2.

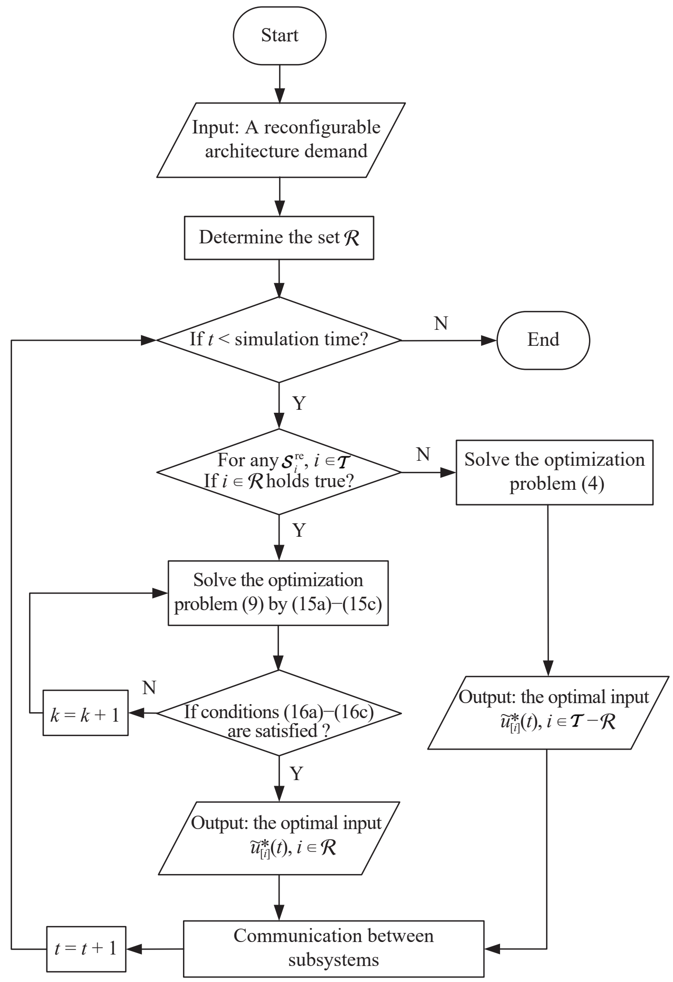


Fig. 2. An algorithm flow chart of the proposed reconfiguration control scheme.

IV. An Academic Example

This section verifies the effectiveness of the proposed reconfiguration control scheme on the level control problem of a benchmark four-tank system [23]. The multi-variable laboratory plant is composed by four interconnected tanks with nonlinear dynamics and is subject to both state and input constraints. There are two inputs (i.e., pump throughputs) which can be used in controlling the tank levels. A schematic of the process for the four-tank system in the existing architecture is shown in Fig. 3. The water in bottom reservoir is transfered by the pumps q_a and q_b to the upper tanks and the liquid levels of each tank can be measured by the local pressure sensors.

To proceed, the state space continuous-time model of the existing four-tank plant is described by the following equations

$$\begin{aligned} \frac{dh_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1}+\frac{a_3}{A_1}\sqrt{2gh_3}+\frac{\gamma_a}{A_1}q_a \\ \frac{dh_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2}+\frac{a_4}{A_2}\sqrt{2gh_4}+\frac{\gamma_b}{A_2}q_b \\ \frac{dh_3}{dt} &= -\frac{a_3}{A_3}\sqrt{2gh_3}+\frac{1-\gamma_a}{A_3}q_a \\ \frac{dh_4}{dt} &= -\frac{a_4}{A_4}\sqrt{2gh_4}+\frac{1-\gamma_b}{A_4}q_b. \end{aligned}$$

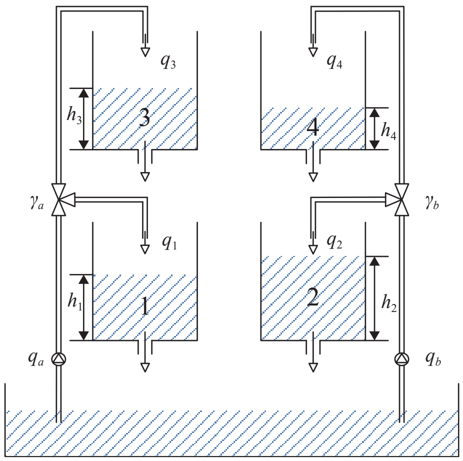


Fig. 3. A schematic description of the exiting four-tank system.

In this example, the water level of each tank is constrained by $h_i \in [0.2, 1.6]$, $\forall i \in \{1,2\}$ and $h_i \in [0.2, 1.3]$, $\forall i \in \{3,4\}$, where the minimum level is used to prevent eddy effects in discharge of the tank. Meanwhile, the flow of pump a and b are confined by $q_a \in [0, 3.26]$ and $q_b \in [0, 4]$. The detailed nominal operating conditions and the parameters estimated on the real four-tank system are provided in Table I.

TABLE I
THE NOMINAL OPERATING CONDITIONS AND THE PARAMETER VALUES OF THE FOUR-TANK SYSTEM

Symbol	Description	Value
A_i	cross-section of tank i	$0.06\ \mathrm{m}^2$
a_i	discharge constant of tank i	$1.310e^{-4}\mathrm{m}^2$, $1.507e^{-4}\mathrm{m}^2$, $9.267e^{-5}\mathrm{m}^2$, $8.816e^{-5}\mathrm{m}^2$
h_i^0	equilibrium level of tank i	0.6534m, 0.6521m, 0.6594m, 0.6587m
γ_a, γ_b	parameter of 3-ways valve	0.3, 0.4
g	gravitational acceleration	$9.81\ \mathrm{m/s}^2$
q_a^0, q_b^0	equilibrium flow	$1.63,\ 2\ \mathrm{m}^3/\mathrm{h}$

By defining the deviation variables $x_i = h_i - h_i^0$ and $u_j = q_j - q_j^0$ with $i \in \{1, \dots, 4\}$ and $j \in \{a, b\}$, the above nonlinear system model can be linearized at the operation point h_i^0 to obtain a discrete-time system model. The sampling frequency is $1s^{-1}$ and the linearized system model is denoted by

$$\frac{dx}{dt} = \begin{bmatrix} \frac{-1}{\tau_1} & 0 & \frac{A_3}{A_1\tau_3} & 0 \\ 0 & \frac{-1}{\tau_2} & 0 & \frac{A_4}{A_2\tau_4} \\ 0 & 0 & \frac{-1}{\tau_3} & 0 \\ 0 & 0 & 0 & \frac{-1}{\tau_4} \end{bmatrix} x + \begin{bmatrix} \frac{\gamma_a}{A_1} & 0 \\ 0 & \frac{\gamma_b}{A_2} \\ \frac{1-\gamma_a}{A_3} & 0 \\ 0 & \frac{1-\gamma_b}{A_4} \end{bmatrix} u$$

$$\tau_i = \frac{A_i}{a_i}\sqrt{\frac{2h_i^0}{g}}, \quad \forall i \in \{1, \dots, 4\}$$

where τ_i denotes the time constant of each tank. The control objective of the four-tank system is to keep the levels of every tank at the specified reference values. In the existing system architecture, as shown in Fig. 2, the global system is divided into two subsystems with $x_{[1]} = [x_1; x_3]$, $u_{[1]} = u_1$ and $x_{[2]} = [x_2; x_4]$, $u_{[2]} = u_2$. The two discretization subsystem models are expressed as

$$x_{[1]}^+ = \begin{bmatrix} 0.994 & 0.004 \\ 0 & 0.996 \end{bmatrix} x_{[1]} + \begin{bmatrix} 5.010 \\ 11.642 \end{bmatrix} u_{[1]}$$

$$x_{[2]}^+ = \begin{bmatrix} 0.993 & 0.004 \\ 0 & 0.996 \end{bmatrix} x_{[2]} + \begin{bmatrix} 6.664 \\ 9.980 \end{bmatrix} u_{[2]}.$$

To verify the presented reconfiguration control scheme, we modify the existing system architecture in Fig. 3 into the system structure shown in Fig. 4, where another branch from pump a is connected to tank 4 while a new branch from pump b becomes an input for tank 3.

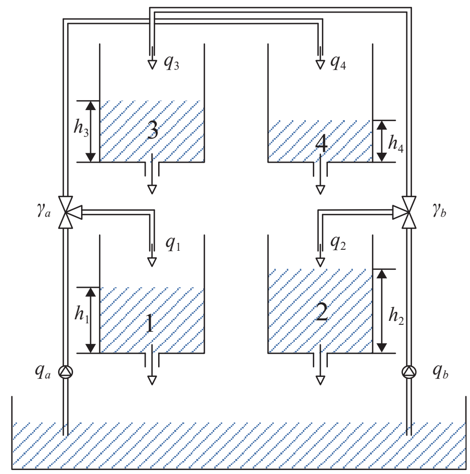


Fig. 4. The plant schema of the reconfigured four-tank system.

In the renewed distributed system architecture, the reconfigured discrete-time subsystem models are given below.

$$\tilde{x}_{[1]}^+ = \begin{bmatrix} 0.994 & 0.004 \\ 0 & 0.996 \end{bmatrix} \tilde{x}_{[1]} + \begin{bmatrix} 0.021 \\ 9.979 \end{bmatrix} \tilde{u}_{[1]} + \begin{bmatrix} 4.985 \\ 0 \end{bmatrix} \tilde{u}_{[2]}$$

$$\tilde{x}_{[2]}^+ = \begin{bmatrix} 0.993 & 0.004 \\ 0 & 0.996 \end{bmatrix} \tilde{x}_{[2]} + \begin{bmatrix} 6.644 \\ 0 \end{bmatrix} \tilde{u}_{[1]} + \begin{bmatrix} 0.023 \\ 11.643 \end{bmatrix} \tilde{u}_{[2]}$$

where the reconfigured subsystem states and inputs are $\tilde{x}_{[1]} = [\tilde{x}_1; \tilde{x}_3]$, $\tilde{u}_{[1]} = \tilde{u}_1$ and $\tilde{x}_{[2]} = [\tilde{x}_2; \tilde{x}_4]$, $\tilde{u}_{[2]} = \tilde{u}_2$.

In this example, we test the presented reconfiguration scheme in two cases: Case I, where the set point is constant and $x_i = 0$, and Case II, where the set point is a square wave and $x_i = \pm 0.1$. Before changing the system architecture, the local controllers are designed by solving the optimization problem (4), where $R_i = 1$ and Q_i is the identity matrix. In the terminal invariant set $\mathcal{X}_i^f = x_{[i]}(k+N|k)^T P_i x_{[i]}(k+N|k) \le \beta_i$, $\beta_i = 0.5$ and P_i are obtained as

$$P_1 = \begin{bmatrix} 170.2976 & -72.5529 \\ -72.5529 & 30.9110 \end{bmatrix}, \; P_2 = \begin{bmatrix} 22.9231 & -15.2461 \\ -15.2461 & 10.1402 \end{bmatrix}.$$

The prediction horizon is selected as $N = 5$. In the reconfigured controller design based on the ADMM algorithm, the over-relaxation parameter $\alpha^{(k)} = 1.8$ and $\rho = 0.2$. Meanwhile, the state and input weighting matrices are

$\tilde{Q}_i = \mathrm{diag}\{30, 1\}$ and $\tilde{R}_i = 1$. In the renewed system structure, $\tilde{P}_i$ are redesigned as

$$\tilde{P}_1 = \begin{bmatrix} 0.4637 & 1.8519 \times 10^{-5} \\ 1.8519 \times 10^{-5} & 0.4572 \end{bmatrix}$$

$$\tilde{P}_2 = \begin{bmatrix} 0.4845 & 1.0589 \times 10^{-5} \\ 1.0589 \times 10^{-5} & 0.4810 \end{bmatrix}.$$

The simulation results of the reconfiguration control in Case I are shown in Figs. 5 and 6. More precisely, Fig. 5 shows the dynamic performance of local subsystems without using the reconfiguration control scheme. During the control process of 0–200 steps, the system is controlled by initial controllers in the original system architecture. At iteration 200, the system structure changes in line with the architecture shown in Fig. 4. Similarly, in Case II, the simulation results are shown in Figs. 7 and 8, where the system architecture is modified at iteration 400. As can be seen from Figs. 5 and 7, the system dynamics in the existing controller design tends to be oscillating and divergent after the structure reconfiguration due to the strong input couplings between subsystems.

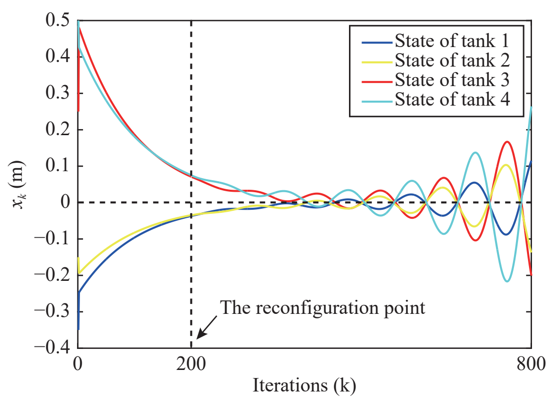


Fig. 5. System performance in the existing controller design in Case I.

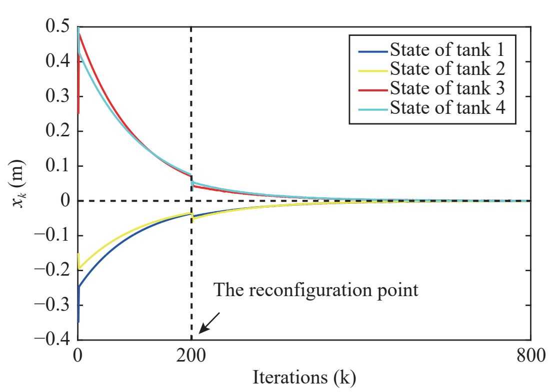


Fig. 6. System performance in the proposed reconfiguration control scheme in Case I.

For comparison, the control performance of the proposed reconfigurable DMPC scheme via the ADMM algorithm in both Cases I and II are depicted in Figs. 6 and 8. As is shown, although a small fluctuation appears at the reconfiguration point, the global system state converges to the steady point in a short period of time, which illustrates the effectiveness of

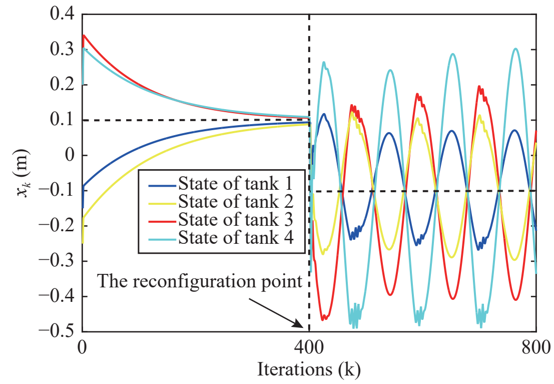


Fig. 7. System performance in the existing controller design in Case II.

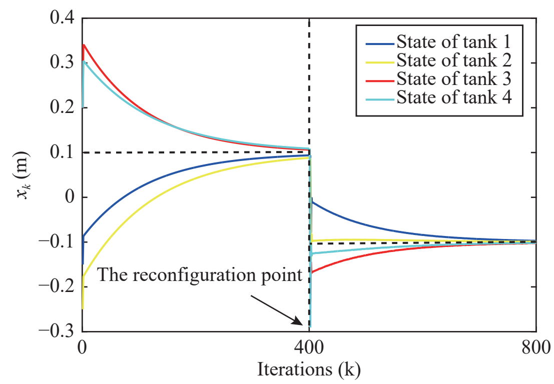


Fig. 8. System performance in the proposed reconfiguration control scheme in Case II.

the presented reconfiguration control scheme.

Moreover, to illustrate the high efficiency of the proposed reconfigured control method combined with the ADMM algorithm, the time consumed in calculating the optimal reconfigured DMPC input via the traditional fmincon function (RDMPC(fmincon)) and the powerful ADMM (RDMPC (ADMM)) are compared. As is shown in Table II, the mean computational time in each iteration with using fmincon and ADMM are 0.145 s and 0.005 s, respectively. During the 600 iterations, the comparison of the accumulated computation time is shown in Fig. 9. From these results we can see that, the calculation efficiency via ADMM is greatly improved in computing the optimal reconfigured input, which helps to ensure a quick-response ability and a satisfactory dynamic performance in the reconfiguration control.

TABLE II
COMPARISON OF THE TIME CONSUMED IN SOLVING RDMPC VIA FMINCON AND ADMM

Algorithm	Iterations	Total time (s)	Mean time (s)
RDMPC (fmincon)	600	87.259	0.145
RDMPC (ADMM)	600	2.985	0.005

V. CONCLUSION

This paper proposed a novel reconfiguration DMPC scheme for linear systems combined with the high-efficiency ADMM algorithm. First, taking into account three typical scenarios, a distributed reconfiguration control strategy applying to any

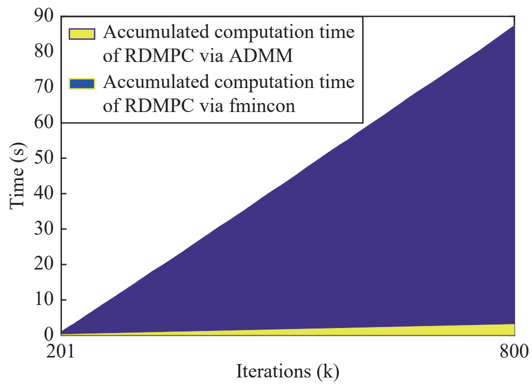


Fig. 9. Comparison of the accumulated computation time in solving the RDMPC via fmincon and ADMM.

reconfigurable requirements was presented. Secondly, based on the powerful ADMM algorithm, the way to redesign the reconfigured DMPC controller was provided and the iterative formulas employed in solving the reconfiguration optimization problem via ADMM algorithm were derived. Finally, the proposed reconfiguration control method was applied to a benchmark four-tank system to illustrate its higher efficiency and good reconfigured control performance.

REFERENCES

[1] P. D. Christofides, R. Scattolini, D. M. de la Pena, and J. F. Liu, "Distributed model predictive control: A tutorial review and future research directions," *Comput. Chem. Eng.*, vol. 51, pp. 21–41, 2013.

[2] Z. M. Wang and C. J. Ong, "Distributed model predictive control of linear discrete-time systems with local and global constraints," *Automatica*, vol. 81, pp. 184–195, 2017.

[3] Y. Zhang, X. J. Liu, and B. Qu, "Distributed model predictive load frequency control of multi-area power system with dfigs," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 1, pp. 125–135, 2017.

[4] A. Zakharov, E. Zattoni, M. Yu, and S.-L. Jämsä-Jounela, "A performance optimization algorithm for controller reconfiguration in fault tolerant distributed model predictive control," *J. Process Control*, vol. 34, pp. 56–69, 2015.

[5] Y. Wei, J. Wu, S. Li, and N. Li, "Distributed model predictive control based on neighbourhood optimization for high speed train," in *Proc. IEEE 10th Int. Conf. Control and Autom.* (*ICCA*). Hangzhou, China, 2013, pp. 664–669.

[6] P. Giselsson and A. Rantzer, "On feasibility, stability and performance in distributed model predictive control," *IEEE Trans. Autom. Control*, vol. 59, no. 4, pp. 1031–1036, 2013.

[7] S. Y. Li, Y. Zheng, and Z. L. Lin, "Impacted-region optimization for distributed model predictive control systems with constraints," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 4, pp. 1447–1460, 2014.

[8] J. Bendtsen, K. Trangbaek, and J. Stoustrup, "Plug-and-play control — modifying control systems online," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 1, pp. 79–93, 2011.

[9] S. Riverso, F. Boem, G. Ferrari-Trecate, and T. Parisini, "Plug-and-play fault detection and control-reconfiguration for a class of nonlinear largescale constrained systems," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 3963–3978, 2016.

[10] S. Riverso, M. Farina, and G. Ferrari-Trecate, "Plug-and-play decentralized model predictive control for linear systems," *IEEE Trans. Autom. Control*, vol. 58, no. 10, pp. 2608–2614, 2013.

[11] M. J. Tippett and J. Bao, "Reconfigurable distributed model predictive control," *Chem. Eng. Sci.*, vol. 136, pp. 2–19, 2015.

[12] D. J. Burns, C. Danielson, J. Q. Zhou, and S. Di Cairano, "Reconfigurable model predictive control for multievaporator vapor compression systems," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 3, pp. 984–1000, 2017.

[13] S. Riverso, M. Farina, and G. Ferrari-Trecate, "Plug-and-play model predictive control based on robust control invariant sets," *Automatica*, vol. 50, no. 8, pp. 2179–2186, 2014.

[14] M. N. Zeilinger, Y. Pu, S. Riverso, G. Ferrari-Trecate, and C. N. Jones, "Plug and play distributed model predictive control based on distributed invariance and optimization," in *Proc. 52nd IEEE Conf. Decision and Control* (*CDC*). Florence, Italy, 2013, pp. 5770–5776.

[15] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Comput. Math. Appl.*, vol. 2, no. 1, pp. 17–40, 1976.

[16] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (admm): quadratic problems," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 644–658, 2014.

[17] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan, "A general analysis of the convergence of admm," arXiv preprint arXiv: 1502.02009, 2015.

[18] B. X. Zhang and Z. B. Zhu, "Linearized proximal alternating direction method of multipliers for parallel magnetic resonance imaging," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 4, pp. 763–769, 2016.

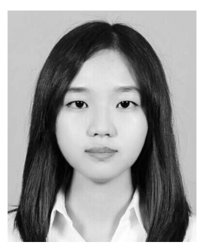
[19] Y. Wang, W. T. Yin, and J. S. Zeng, "Global convergence of admm in nonconvex nonsmooth optimization," *J. Sci. Comput.*, vol. 78, no. 1, pp. 29–63, 2019.

[20] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[21] F. Farokhi, I. Shames, and K. H. Johansson, "Distributed mpc via dual decomposition and alternative direction method of multipliers," in *Distributed Model Predictive Control Made Easy*. Springer, 2014, pp. 115–131.

[22] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *J. Mach. Learn. Res.*, vol. 11, pp. 1163–1707, 2010.

[23] E. P. Gatzke, E. S. Meadows, C. Wang, and F. J. Doyle Iii, "Model based control of a four-tank system," *Comput. Chem. Eng.*, vol. 24, no. 2–7, pp. 1503–1509, 2000.



Ting Bai received the B.S. degree in automation from Northwestern Polytechnical University in 2013, and the Ph.D. degree in control science and engineering from Shanghai Jiao Tong University, in 2019. Her research interests include distributed model predictive control, complex networked systems, minimum control configuration design, and distributed optimal control of reconfigurable architecture systems.

Shaoyuan Li (SM'05) received the B.S. and M.S. degrees in automation from Hebei University of Technology, in 1987 and 1992, respectively, and the Ph.D. degree from the Department of Computer and System Science, Nankai University, in 1997. Since July 1997, he has been with the Department of Automation, Shanghai Jiao Tong University, where he is currently a Chair Professor. His research interests include model predictive control and dynamic system optimization for large-scale networked systems. He is a Vice President of Asian Control Association (ACA).

Yuanyuan Zou (M'13) is a Professor with the Department of Automation, Shanghai Jiao Tong University. She received the B.S. degree and M.S. degree from Ludong University in 2002 and 2005, respectively, and the Ph.D. degree from Shanghai Jiao Tong University in 2009. Her research interests include predictive control, network-based control systems, and distributed control systems.