

Research Article

Distributed Double-Layered Dynamic Matrix Control for Large-Scale System

Li Wang , Yuanli Cai , and Xin Zan 

Faculty of Electronic and Information, School of Automation, Xi'an Jiaotong University, Xi'an 710049, China

Correspondence should be addressed to Yuanli Cai; ylicai@mail.xjtu.edu.cn

Received 5 August 2021; Revised 5 April 2022; Accepted 16 April 2022; Published 30 May 2022

Academic Editor: Taseer Muhammad

Copyright © 2022 Li Wang et al. 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.

A distributed double-layered dynamic matrix control is proposed for large-scale systems. In the scheme, a large-scale system is first decomposed into several low-dimensional subsystems, and then a double-layered dynamic matrix control algorithm with distributed structure is developed for these subsystems. The distributed open-loop prediction equation of each subsystem is formed based on the predicted output of each local subsystem and effects of its interconnecting neighbor subsystems. Due to simultaneous optimization, at each prediction, the coupling effects of neighbor subsystems are not available in time. Thus, the assumed value is utilized instead. In the economic optimization stage, conflicts may occur among different economic optimization goals. Utopia-tracking strategy is introduced to optimize multiple steady-state targets. Then, the obtained steady-state target values are taken as reference values and tracked in subsequent dynamic control. The actual control move for each subsystem is finally calculated. The proposed algorithm is tested on Shell heavy oil fractionator benchmark, and the effectiveness is demonstrated by comparing with the typical double-layered dynamic matrix control algorithm.

1. Introduction

With the advantage of handling various constraints explicitly [1, 2], model predictive control (MPC) has become a standard method for solving the constrained multivariable control problem. Utilizing MPC, the production process can be operated on the boundaries of some important constraints, and the economic efficiency improves greatly. Recently, predictive control has been successfully applied to thousands of process control systems and achieved great economic efficiency.

In most modern industrial process control techniques, steady-state target calculation (SSTC) is incorporated into MPC to find the optimal set-point automatically; thus, a double-layered MPC is established [3–6]. In [7], the double-layered zone predictive control strategy was proposed to overcome the performance degradation caused by the mechanical wear. In [8], a method to guarantee the feasibility of optimization problem was presented for dynamic control tracking the steady-state targets. In the double-layered architecture, the economic optimization and dynamic control

are separately working at different rates. Moreover, the offset-free control is achieved even in the presence of disturbance [9, 10]. Therefore, the double-layered control has become the most effective and promising advanced industrial control technology.

However, it is challenge for the double-layered MPC to be applied to large-scale systems. As the size of the controlled process increases, the inherent computational complexity and communication limitation make it difficult for MPC to seek a satisfactory solution within an acceptable computational burden. Usually, the large-scale system is first decomposed into low order interconnected subsystems, and then a distributed control scheme is designed based on the subsystems [11–16]. In the distributed architecture, the interconnection influences among subsystems should be considered [11, 13, 17–19]. Besides, in practical application, many physical limitations such as input saturation, the physical limitation of actuators, and external disturbances also need to be considered. In [15], a distributed optimal control was proposed for the interconnected large-scale system with external disturbances and input constraints,

where by utilizing an adaptive dynamic programming technique, the worst-case disturbance policy and constrained-input optimal control policy can be approximate synchronously. An event-triggered feed-forward control policy was proposed in [16] to transform the control of large-scale systems into equivalent event-triggered control of isolated subsystems. By designing a novel event-triggering condition, three neural networks are eliminated for each subsystem, so that the computational complexity and resource are reduced. A distributed noncooperative MPC was proposed in [20], where input and state constraints are considered and convergence of the closed-loop control system is proved. According to the optimization problem and the types of information exchange among subsystems, the distributed MPCs are often classified as cooperative [21–23] and noncooperative [19, 20, 24]. In cooperative distributed MPC, all subsystems solve a global optimization problem and exchange information with each other. While in noncooperative MPC, each subsystem solves a local optimization problem and exchanges information iteratively with its neighbor subsystems and finally achieves Nash optimal solution [12, 24]. In [24], an efficient distributed MPC based on Nash optimality is presented for the large-scale system, which mainly aimed at reducing the computational burden in model predictive control. Moreover, as reviewed in [6], both distributed and hierarchical control architectures are proposed for large-scale systems. In [25], the cooperative distributed MPC is extended to the hierarchical structure so as to reduce the communication frequency among subsystems. A novel distributed MPC algorithm for large-scale systems proposed in [26] is very similar to the MPC algorithms employed in industry, where each subsystem needs the reference trajectories only of the variables of its interconnected neighbors.

Motivated by the above mentioned, in this paper, a distributed double-layered dynamic matrix control (DMC) is proposed for large-scale systems. In the distributed structure, when some economic objectives are in conflict at the economic optimization stage among different subsystems, it is tough to solve the optimal steady-state target values. Multiobjective optimization strategy is introduced to solve this knotty problem. A typical method for solving the multiobjective optimization with conflicting cost functions is to construct its Pareto front and then choose the appropriate point as the optimal value [27]. However, this method is computationally heavy and impractical in real-time application. Besides, an ideal set-point varies with changing operating conditions; thus, the corresponding steady-state targets need to update at each iteration, which makes it more complicate to calculate the steady-state target values. In [28], the steady-state compromise solution was proposed to avoid the computation of Pareto fronts in real-time environments. Therefore, by adopting this strategy, the steady-state target of each subsystem in the distributed double-layered DMC can be solved effectively.

This paper is organized as follows Section 2 presents the open-loop prediction model of each subsystem in the distributed scheme, where the estimation of coupling effects of neighbor subsystems is utilized. The error correction

between the actual value and the prediction value is included in the open-loop dynamic prediction equation. Section 3 analyzes the feasibility of the local optimization of each subsystem. The optimal steady-state target values are calculated based on the improved Pareto optimality. Section 4 designs the dynamic control for the distributed system and then calculates the control move and sent to the plant and implemented. Section 5 demonstrates the effectiveness of the proposed distributed double-layered DMC algorithm by applying it to Shell heavy oil fractionator benchmark and comparing with the centralized double-layered DMC algorithm.

2. Distributed Open-Loop Prediction Model

Consider a large-scale system with n_y outputs and n_u inputs, its dynamics is described by the following finite step response (FSR) model:

$$\begin{aligned} \nabla y(k) &= \sum_{t=1}^{N-1} S_t^u \Delta u(k-t) + S_N^u \nabla u(k-N) \\ &+ \sum_{t=1}^{N-1} S_t^f \Delta f(k-t) + S_N^f \nabla f(k-N), \end{aligned} \quad (1)$$

where N is the modeling horizon. $y \in \mathbb{R}^{n_y}$ and $u \in \mathbb{R}^{n_u}$ are the controlled variable and manipulated variable, respectively. $f \in \mathbb{R}^{n_f}$ is the disturbance variable. $\nabla y = y - y_{eq}$, where y_{eq} denotes the value of y at the equilibrium. For simplicity, ∇ is omitted. S_t^u and S_t^f are the step response coefficient matrices of the manipulated variable and disturbance variable, respectively. For all $t \geq 0$, $S_{N+t}^u = S_N^u$, $S_{N+t}^f = S_N^f$.

2.1. Decomposed Systems. Let system equation (1) be decomposed into M interconnected subsystems, where each subsystem a , $a \in \mathbb{M}$, $\mathbb{M} = \{1, 2, \dots, M\}$ has $u_a \in \mathbb{R}^{n_a}$ as input vector, i.e., $u = (u_1, \dots, u_M)$ and $\sum_{a=1}^M n_{ua} = n^u$. Similarly, it has $y_a \in \mathbb{R}^{n_y}$ as output vector, i.e., $y = (y_1, \dots, y_M)$, and $\sum_{a=1}^M n_{ya} = n^y$. In the decomposition, step response coefficient matrices $S^{u_{11}} \in \mathbb{R}^{n_1 \times n_1}, \dots, S^{u_{MM}} \in \mathbb{R}^{n_M \times n_M}$ of M subsystems are diagonal blocks of S^u . While the nondiagonal blocks of S^u (i.e., $S^{u_{al}}, a \neq l$) represent the interconnection influences among subsystems. $S^{u_{al}} \neq 0$ means that the input of subsystem l affects the dynamics of subsystem a . Therefore, the FSR model of each subsystem $a \in \mathbb{M}$ is expressed as follows:

$$\begin{aligned} \nabla y_a(k) &= \sum_{t=1}^{N-1} S_t^{u_a} \Delta u_a(k-t) + S_N^{u_a} u_a(k-N) \\ &+ \sum_{l \in \mathcal{N}_a} \left[\sum_{t=1}^{N-1} S_t^{u_{al}} \Delta u_l(k-t) + S_N^{u_{al}} u_l(k-N) \right] \\ &+ \sum_{t=1}^{N-1} S_t^f \Delta f(k-t) + S_N^f f(k-N), \end{aligned} \quad (2)$$

where $y_a \in \mathbb{R}^{n_{y_a}}$, $u_a \in \mathbb{R}^{n_{u_a}}$ are the output and input vectors of subsystem a , respectively. N denotes the modeling horizon for the large-scale system. N_c denotes the control horizon. $l \in \mathcal{N}_a$ denotes the neighbors of subsystem a , which excludes a , namely, $\mathcal{N}_a = \{1, 2, \dots, a-1, a+1, M\}$ denoting the collection of the subsystem a 's dynamic neighbor subsystems. S^{ul} represents the interconnection influence on subsystem a from neighbor subsystem l . $f \in \mathbb{R}^{n_f}$ is the disturbance variable.

2.2. Distributed Open-Loop Prediction. A large-scale system is decomposed into M dynamically coupled subsystems, and the distributed control is established based on the subsystems. The following symbols are used to distinguish different prediction information:

$u(k+t|k)$ is the real prediction of input

$\hat{u}(k+t|k)$ is the assumed predictive value of input

$u^*(k+t|k)$ is the optimal predictive value of input

For interconnected subsystems, the open-loop prediction output of subsystem a is affected not only by the input of local subsystem a itself but also by some information stemming from its neighbors. For free prediction, the local input of each subsystem does not change from time k onwards, that is, $\Delta u_a(t|k) = 0$ for all $t \geq k$. However, the inputs u_l to the neighbor subsystems keep changing. Due to communication delay, the inputs from its neighbors cannot be obtained immediately. The assumed value is used. Here, the assumed predictive input $\Delta \hat{u}_l(t|k)$ of each subsystem l at time k is chosen by the following formula:

$$\Delta \hat{u}_l(t|k) = \begin{cases} \Delta u_l^*(t|k-1), & t = k, \dots, N-1, \\ \Delta u_l(t), & t = 1, 2, \dots, k-1. \end{cases} \quad (3)$$

Suppose that the inputs from time k onwards do not change, $\Delta u(k+t-1|k) = 0$ ($1 \leq t \leq p$), the corresponding prediction of $y(k+p)$ is called free prediction, denoted as $y^{fr}(k+p|k)$, where the superscript "fr" means free. In the distributed structure, the free prediction of each subsystem a is established by assuming that the input of local subsystem is unchanged from time k , whereas the inputs from neighbor subsystems may vary. Thus, the free prediction of subsystem a can be expressed as follows:

$$\begin{aligned} & y_a^{fr}(k+p|k) \\ &= \sum_{t=1}^{N+N_c-1} S_t^{u_a} \Delta u_a(k-t+p|k) \\ &+ S_{N+N_c}^{u_a} u_a(k-N-N_c+p|k) \\ &+ \sum_{l \in \mathcal{N}_a} \left[\sum_{t=1}^{N+N_c-1} S_t^{ul} \Delta \hat{u}_l(k-i+p|k) \right. \\ & \left. S_{N+N_c}^{ul} u_l(k-N-N_c+p|k) \right] \\ &+ \sum_{t=1}^{N-1} S_t^f \Delta f(k-t) + S_N^f f(k-N). \end{aligned} \quad (4)$$

The free prediction at time $k-1$ is

$$\begin{aligned} & y_a^{fr}(k+p|k-1) \\ &= \sum_{t=1}^{N+N_c-1} S_t^{u_a} \Delta u_a(k-t+p|k-1) \\ &+ S_{N+N_c}^{u_a} u_a(k-N-N_c+p|k-1) \\ &+ \sum_{l \in \mathcal{N}_a} \left[\sum_{t=1}^{N+N_c-1} S_t^{ul} \Delta \hat{u}_l(k-t+p|k-1) \right. \\ & \left. + S_{N+N_c}^{ul} \hat{u}_l(k-N-N_c+p|k-1) \right] \\ &+ \sum_{t=1}^{N-1} S_t^f \Delta f(k-t+p) + S_N^f f(k-N+p). \end{aligned} \quad (5)$$

Suppose that $f(k)$ is measurable and will remain unchanged in the future. Then, one has

$$\begin{aligned} & y_a^{fr}(k+p|k) - y_a^{fr}(k+p|k-1) \\ &= S_{p+1}^{u_a} \Delta u_a(k-1) \\ &+ \sum_{l \in \mathcal{N}_a} \left\{ \sum_{t=1}^{N_c-1} S_t^{ul} [\Delta \hat{u}_l(k-t+p|k) \right. \\ & \left. - \Delta \hat{u}_l(k-t+p|k-1)] \right\}. \end{aligned} \quad (6)$$

Suppose that the control horizon of neighbor subsystem l is N_c . M coupling inputs from neighbor subsystem l , $l \in \mathcal{N}_a$ will keep changing from time k onwards. According to equation (3), substituting the assumed values into equation (6), the one-step prediction error is obtained by

$$\begin{aligned} & \sum_{t=1}^{N_c-1} S_t^{ul} [\Delta \hat{u}_l(k-t+p|k) - \Delta \hat{u}_l(k-i+p|k-1)] \\ &= \sum_{t=1}^p S_t^{ul} [\Delta u_l^*(k-t+p|k-1) - \Delta u_l^*(k-t+p|k-2)] \\ &+ S_1^{ul} [\Delta u_l(k-1) - \Delta u_l^*(k-1|k-2)]. \end{aligned} \quad (7)$$

As shown in equation (4), the influences of interconnection inputs from neighbors are considered in free prediction $y_a^{fr}(k+p|k)$ and then the real prediction for subsystem a can be calculated by

$$\begin{aligned} & y_a(k+p|k) \\ &= y_a^{fr}(k+p|k) + \sum_{t=1}^p S_t^{u_a} \Delta u_a(k-t+p|k). \end{aligned} \quad (8)$$

Seen from equation (8), at time k , the real output prediction is determined by only the effect of subsystem a itself. Thus, the coupling influences from neighbors can be neglected in the subsequent controller design. Notice that, in equations (6) and (8), the prediction step p satisfies $p \leq 0.001$. By taking $S_0^f = 0$, it is allowed to take $p \leq 0.001$. Then, we have $y^{fr}(k|k) = y(k|k) \neq y(k)$.

Feedback correction is added to equation (8) and then the open-loop prediction y_a^{ol} is obtained. Here, the superscript "ol" denotes open-loop. Based on equation (6), the open-loop prediction of subsystem a can be calculated by

$$\begin{aligned} y_a^{ol}(k|k) &= y_a^{ol}(k|k-1) + S_1^{u_a} \Delta u_a(k-1) \\ &+ \sum_{l \in \mathcal{N}'_a} \{S_1^{u_{al}} [\Delta u_l(k-1) \\ &- \Delta u_l^*(k-1|k-2)]\}. \end{aligned} \quad (9)$$

At time $k > 0$, the actual input $\Delta u_a(k-1)$ of each subsystem a is measurable. Suppose $y_a(k)$ is measurable. The open-loop prediction error is obtained by

$$\begin{aligned} \epsilon_a(k) &= y_a(k) - y_a^{ol}(k|k), \\ &= y_a(k) - y_a^{ol}(k|k-1) - S_1^{u_a} \Delta u(k-1), \\ &- \sum_{l \in \mathcal{N}'_a} \{S_1^{u_{al}} [\Delta u_l(k-1), \\ &- \Delta u_l^*(k-1|k-2)]\}. \end{aligned} \quad (10)$$

Let $\epsilon_i(k) = \Delta u_i(k) - \Delta \hat{u}_i(k)$, then $\sum_{l \in \mathcal{N}'_a} S_1^{u_{al}} [\Delta u_l(k-1) - \Delta u_l^*(k-1|k-2)] = \sum_{l \in \mathcal{N}'_a} S_1^{u_{al}} \epsilon_l(k-1)$. Let $e_i \in \mathbb{E}_i = \sum_{i \in \mathcal{N}'_a} S_1^{u_{al}} \epsilon_l$, and e_i can be considered as a bounded disturbance. The feedback correction $\epsilon_a(k)$ is added into the open-loop prediction equation to compensate the prediction errors. Let $N' = N + N_c$, the submodel of subsystem a can be written in vector form as follows:

$$\begin{aligned} \mathbf{Y}_{N',a}(k|k) &= \mathcal{A} \left[Y_{N',a}^{ol}(k|k-1) + \begin{bmatrix} S_1^{u_a} \\ S_2^{u_a} \\ \vdots \\ S_{N'}^{u_a} \end{bmatrix} \Delta u_a(k-1) \right. \\ &+ \left. \sum_{l \in \mathcal{N}'_a} (S_t^{u_{al}} \times \Delta \hat{\mathcal{U}}_l) \right] + \begin{bmatrix} S_1^f \\ S_2^f \\ \vdots \\ S_{N'}^f \end{bmatrix} \Delta f_a(k) + \begin{bmatrix} e_a(k+1|k) \\ e_a(k+2|k) \\ \vdots \\ e_a(k+N'|k) \end{bmatrix}, \\ \mathcal{A} &= \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & I & 0 \\ 0 & \cdots & 0 & 0 & I \\ 0 & \cdots & 0 & 0 & I \end{bmatrix}, \\ S_t^{u_{al}} &= \begin{bmatrix} S_1^{u_{al}} & S_2^{u_{al}} & \cdots & S_{N_c+1}^{u_{al}} \\ S_2^{u_{al}} & S_3^{u_{al}} & \cdots & S_{N_c+2}^{u_{al}} \\ \vdots & \vdots & \ddots & \vdots \\ S_{N'}^{u_{al}} & S_{N'+1}^{u_{al}} & \cdots & S_{N'+N_c}^{u_{al}} \end{bmatrix}, \\ \Delta \hat{\mathcal{U}}_l &= \begin{bmatrix} \Delta \hat{u}_l(k-1|k) - \Delta \hat{u}_l(k-1|k-1) \\ \Delta \hat{u}_l(k|k) - \Delta \hat{u}_l(k|k-1) \\ \vdots \\ \Delta \hat{u}_l(k+N_c-1|k) - \Delta \hat{u}_l(k+N_c-1|k-1) \end{bmatrix}. \end{aligned} \quad (11)$$

Notice that the assumption $y_a^{ol}(k+N'|k-1) = y_a^{ol}(k+N'-1|k-1)$ is utilized in deducing equation (11).

For a stable process, the feedback correction is time-invariant, that is, $e_i(k+t|k) = \epsilon_i(k)$, for $t \geq 0$. Then, the open-loop steady-state prediction of each subsystem can be expressed as

$$y_{a,ss}^{ol} = y_a^{ol}(k+N'+t|k), \quad t \geq 0. \quad (12)$$

When replacing k with $k-1$ in equation (12), Equation (11) is still highly consistent with equation (12). In this case, the final open-loop dynamic prediction is equivalent to the steady-state prediction. Thus, the open-loop prediction equation can predict the dynamics of each subsystem effectively.

3. Calculating Steady-State Target Values Based on Improved Pareto Optimality

For the control of industrial process, the manipulated variables and controlled variables do not always hit the desired external targets given by the real-time optimization (RTO). In order to ensure that the manipulated variables operate as close to the desired set-point as possible, steady-state optimization is incorporated into the DMC algorithm, thus the double-layered DMC algorithm is formed, in which the optimal set-point is calculated by steady-state target calculation (SSTC). SSTC involves two stages: the first stage aims at to analyze the feasibility of the solution which satisfies various constraints and the second one is economic optimization stage, which is designed to seek the optimal steady-state target values.

Based on equation (8), the steady-state prediction equation for each subsystem a can be calculated by

$$y_{a,ss}(k) = S_{N'}^{u_a} \delta u_{a,ss}(k) + y_{a,ss}^{ol}(k), \quad (13)$$

where $\delta u_{a,ss}$ denotes the increment variable of the steady-state target of u for subsystem a , $\delta u_{a,ss}(k) = u_{a,ss}(k) - u_a(k-1)$. $y_{a,ss}$ denotes the steady-state target value of controlled variable y for subsystem a .

3.1. Handling the Constraints in SSTC. In practice, some operation requirements and physical restrictions need to be met. The constraints imposed on controlled variables and manipulated variables include soft constraints and hard constraints. In general, the hard constraints cannot be violated, while the soft constraint can be relaxed appropriately according to the situations. Besides, some variables have their own external target (ET) values given by the real-time optimization control engineers or operators.

The steady-state target of each manipulated variable needs to satisfy the corresponding hard magnitude constraints:

$$\underline{u}_a \leq u_{a,ss}(k) \leq \bar{u}_a, \quad k \geq 0, \quad (14)$$

where \underline{u} denotes the lower bound of u and \bar{u} denotes the upper bound of u .

In dynamic control module, the increment of manipulated variable has rate constraint $|\Delta u_a(k+j|k)| \leq \Delta \bar{u}$ ($0 \leq j \leq M-1$) and then $\delta u_{a,ss}$ should further satisfy

$$|\delta u_{a,ss}(k)| \leq M\Delta \bar{u}_a, \quad k \geq 0, \quad (15)$$

where $\delta u_{ss}(k) = u_{ss}(k) - u(k-1)$ denotes steady-state target increment of u at time k . $\Delta \bar{u}$ denotes the maximum allowable absolute change rate of u .

Combining equations (14) and (15), the hard constraint of $\delta u_{a,ss}$ is

$$\underline{u}'_a \leq \delta u_{a,ss}(k) \leq \bar{u}'_a(k), \quad k \geq 0, \quad (16)$$

where

$$\begin{aligned} \underline{u}'_a &= \max\{\underline{u}_a - u_a(k-1), -M\Delta \bar{u}_a\}, \\ \bar{u}'_a &= \min\{\bar{u}_a - u_a(k-1), M\Delta \bar{u}_a\}. \end{aligned} \quad (17)$$

The steady-state target of the controlled variables in each subsystem needs to satisfy the engineering limits, referred to as hard magnitude constraints as follows:

$$\underline{y}_{0,h} \leq y_{a,ss}(k) \leq \bar{y}_{0,h}, \quad k \geq 0, \quad (18)$$

where $\underline{y}_{0,h}$ denotes the lower engineering limit and $\bar{y}_{0,h}$ denotes the upper engineering limit.

Besides, the controlled variable target of each subsystem has to satisfy the operating limit:

$$\underline{y}_0 \leq y_{a,ss}(k) \leq \bar{y}_0, \quad k \geq 0, \quad (19)$$

where \underline{y}_0 denotes lower operating limit of y and \bar{y}_0 denotes upper operating limit of y .

The engineering limits are more stringent than the operating limits $\underline{y}_{0,h} \leq \underline{y}_0$ and $\bar{y}_0, h \geq \bar{y}_{0,h}$.

Substituting the steady-state prediction equation (13) into equation (19), we yield

$$\underline{y}_0 \leq S_{N'}^u \delta u_{a,ss}(k) + y_{a,ss}^{ol}(k) \leq \bar{y}_0, \quad k \geq 0. \quad (20)$$

Then, we have

$$\underline{y}(k) \leq S_{N'}^u \delta u_{a,ss}(k) \leq \bar{y}(k), \quad (21)$$

where

$$\begin{aligned} \underline{y}(k) &= \min\{\max\{\underline{y}_0 - y_{ss}^{ol}(k), \underline{y}_h(k)\}, \bar{y}_h(k)\}, \\ \bar{y}(k) &= \max\{\min\{\bar{y}_0 - y_{ss}^{ol}(k), \bar{y}_h(k)\}, \underline{y}_h(k)\}. \end{aligned} \quad (22)$$

The ETs of $u_{i,ss}$ and $y_{j,ss}$ are denoted as $u_{i,t}$ and $y_{j,t}$, respectively. $i \in \mathbb{I}_t$, $j \in \mathbb{J}_t$, \mathbb{I}_t and \mathbb{J}_t represent the corresponding set of all the ETs of $u_{i,ss}$ and $y_{j,ss}$. Denote the expected allowable range of $u_{i,t}(k)$ as $u_{i,ss,range}$. The limits of ET values for $u_{i,ss}$ can be taken as follows:

$$\bar{u}_{i,ss}(k) = u_{i,t}(k) + \frac{1}{2}u_{i,ss,range} - u_i(k-1), \quad i \in \mathbb{I}_t, \quad (23)$$

$$\underline{u}_{i,ss}(k) = u_{i,t}(k) - \frac{1}{2}u_{i,ss,range} - u_i(k-1), \quad i \in \mathbb{I}_t.$$

Then, for $u_{j,ss}(k)$, check the following soft constraint:

$$\underline{u}_{i,ss}(k) \leq \delta u_{i,ss}(k) \leq \bar{u}_{i,ss}(k), \quad i \in \mathbb{I}_t. \quad (24)$$

Denote the expected allowable range of $y_{j,t}(k)$ as $y_{j,ss,range}$. The limits of ET values for $y_{j,ss}$ are given by

$$\bar{y}_{j,ss}(k) = y_{j,t}(k) + \frac{1}{2}y_{j,ss,range} - y_{j,ss}^{ol}(k), \quad j \in \mathbb{J}_t, \quad (25)$$

$$\underline{y}_{j,ss}(k) = y_{j,t}(k) - \frac{1}{2}y_{j,ss,range} - y_{j,ss}^{ol}(k), \quad j \in \mathbb{J}_t.$$

If the external target of $y_{j,ss}(k)$ is considered, the following soft constraints need to be further satisfied, that is,

$$\begin{aligned} S_{N'}^u \delta u_{ss}(k) &\leq \bar{y}(k), \quad j \in \mathbb{J}_t, \\ S_{N'}^u \delta u_{ss}(k) &\geq \underline{y}(k), \quad j \in \mathbb{J}_t. \end{aligned} \quad (26)$$

Unifying all the ETs into the constraint of δu_{ss} , the corresponding constraint for $u_{i,t}(k)$ is denoted as

$$\delta u_{i,ss}(k) = u_{i,t}(k) - u_i(k-1), \quad i \in \mathbb{I}_t. \quad (27)$$

For $y_{j,t}(k)$, the constraint is

$$S_{N',j}^u \delta u_{ss}(k) = y_{j,t}(k) - y_{j,ss}^{ol}(k), \quad j \in \mathbb{J}_t. \quad (28)$$

3.2. Feasibility Analysis. For each subsystem a , the feasible solution of the economic optimization is determined by its constraint conditions including hard constraints and soft constraints, where the hard constraints are classified into manipulated variable constraints (equation (16)) and hard controlled variable constraints (equation (18)):

$$\begin{bmatrix} I \\ -I \\ S_{N'}^u \\ -S_{N'}^u \end{bmatrix} \delta u_{a,ss}(k) \leq \begin{bmatrix} \bar{u}'(k) \\ \underline{u}(k) \\ \bar{y}_h(k) \\ -\underline{y}_h(k) \end{bmatrix}. \quad (29)$$

The soft constraints do not have to be satisfied. In practical application, priority strategy is usually introduced to handle soft constraints. All soft constraints are prioritized according to importance. The rule for handling the soft constraints is that the soft constraints with higher priority are processed first and that with lower priority later. Soft constraints that are previously processed (possibly relaxed) are treated as hard constraints in subsequent priority processing. Some soft constraints are relaxed properly so that all

the constraints at the same priority are compatible. For convenience, at each rank of priority, either equality constraints or inequality constraints are dealt with. For the optimization problem with r rank of priorities, the constraint handled at the $(r - 1)$ th priority can be denoted as

$$C^{(r-1)} \delta u_{a,ss}(k) \leq c^{(r-1)}(k), \quad (30)$$

$$C_{eq}^{(r-1)} \delta u_{a,ss}(k) \leq c_{eq}^{(r-1)}(k). \quad (31)$$

Constraint equations (30) and (31) are treated as hard constraints in the r th priority. The following two cases are considered:

- (i) If the r th rank of priority is the desired value of external target, relax the r th equality constraint and then the considered constraints are as follows:

$$C^{(r-1)} \delta u_{a,ss}(k) \leq c^{(r-1)}(k), \quad (32a)$$

$$\begin{bmatrix} C_{eq}^{(r-1)} \\ \tilde{C}_{eq}^{(r)} \end{bmatrix} \delta u_{a,ss}(k) = \begin{bmatrix} c_{eq}^{(r-1)}(k) \\ \tilde{c}_{eq}^{(r)}(k) + \varepsilon_{eq}^{(r)}(k) \end{bmatrix}. \quad (32b)$$

- (ii) If the r th rank of priority is the desired upper bound or lower bound of target value, relax the r th rank of inequality constraint and then the considered constraints are as follows:

$$\begin{bmatrix} C^{(r-1)} \\ \tilde{C}^{(r)} \end{bmatrix} \delta u_{a,ss}(k) \leq \begin{bmatrix} c^{(r-1)}(k) \\ \tilde{c}^{(r)}(k) + \varepsilon^{(r)}(k) \end{bmatrix}, \quad (33a)$$

$$\varepsilon^{(r)}(k) \geq 0,$$

$$C_{eq}^{(r)} \delta u_{a,ss}(k) = c_{eq}^{(r-1)}, \quad (33b)$$

where $\varepsilon_{eq}^{(r)}(k)$ and $\varepsilon^{(r)}(k)$ are the slack variables.

In order to seek a smaller $|\varepsilon|$, either LP or QP is invoked. In each rank of priority, if more than one soft constraint needs to be relaxed, correspondingly, there are multiple ε 's need to be optimized. Suppose given them equal importance. Then, an equal concern error $\bar{\varepsilon}$ is assigned to each slack variable ε . For the upper bound constraints of controlled variables, $\bar{\varepsilon} = \underline{y}_h - \underline{y}_0$, and for the lower bound constraints, $\bar{\varepsilon} = \underline{y}_0 - \underline{y}_h$, and $\varepsilon = 1/2ET_{range}$ for the ET constraints or equality constraints. If some $\bar{\varepsilon} = 0$, the corresponding soft constraint cannot be relaxed and then the corresponding soft constraint can be removed.

Set

$$\varepsilon_{eq}^{(r)} = \varepsilon_{eq+}^{(r)}(k) - \varepsilon_{eq-}^{(r)}(k), \varepsilon_{eq+}^{(r)}(k) \geq 0, \varepsilon_{eq-}^{(r)}(k) \geq 0, \quad (34)$$

then the slack variables $\varepsilon_{eq+}^{(r)}(k)$ and $\varepsilon_{eq-}^{(r)}(k)$ can be obtained by solving the following LP:

$$\begin{aligned} \min_{\varepsilon_{eq+}^{(r)}, \varepsilon_{eq-}^{(r)}} \delta u_{a,ss}(k) & \sum_{\tau=1}^{d_{r+1}} \left(\bar{\varepsilon}_{eq,\tau}^{(r+1)} \right)^{-1} \\ & \times \left(\varepsilon_{eq+,\tau}^{(r)}(k) + \varepsilon_{eq-,\tau}^{(r)}(k) \right) \\ \text{s.t.} & (32)-(34), \end{aligned} \quad (35)$$

where τ indicates the τ th element of $\varepsilon_{eq}^{(r)}(k)$ and $d_{eq}^{(r+1)}$ denotes the dimension of $\varepsilon_{eq}^{(r)}(k)$.

3.3. Economic Optimization Stage of SSTC Based on Pareto Optimality. In the economic optimization stage, an economic cost function representing the profit/loss is constructed based on the change of $u_{ss}(k)$ and $y_{ss}(k)$. Suppose the steady-state change of input variable is bounded by

$$-U_i(k) \leq \delta u_{i,ss}(k) \leq U_i(k). \quad (36)$$

Then, minimizing the increment variable $|\delta u_{i,ss}(k)|$ is equivalent to minimizing its constraint bound $U_i(k)$. Having analyzed the feasibility, all the hard constraints and the relaxed soft constraints are simplified into the following formula:

$$\begin{aligned} u_i'(k) \leq \delta u_{i,ss}(k) & \leq \bar{u}_i'(k), \quad i \notin \mathbb{J}_t, \\ y_j(k) \leq S_{j,N'}^u \delta u_{ss}(k) & \leq \bar{y}_j(k), \quad j \notin \mathbb{J}_t, \\ S_{j_i,N'}^u \delta u_{ss}(k) = y_{j,ss}(k) - y_{j,ss}^o(k), & \quad j \in \mathbb{J}_t, \\ \delta u_{i,ss}(k) = u_{i,ss}(k) - u_i(k-1), & \quad i \in \mathbb{J}_t. \end{aligned} \quad (37)$$

By invoking LP, the economic optimization problem of each subsystem a is written as

$$\begin{aligned} \min_{\delta u_{a,ss}(k), U_i(k)} J_a = & \left[\sum_{i \notin \mathbb{J}_{mm} \cup \mathbb{J}_t} h_i \delta u_{a,ss}(k) \right. \\ & \left. + \sum_{i \in \mathbb{J}_{mm}} h_i U_i(k) \right], \end{aligned} \quad (38)$$

$$\text{s.t.} \quad (19), (24), (26)-(28),$$

$$i \in \mathbb{J}_{mm},$$

where h_i is the weighting coefficient.

Notice that $\delta u_{a,ss}$ calculated by equation (38) is local optimal for each subsystem itself, but not for the whole large-scale system. In this distributed control, if some economic objectives of different subsystems are in conflict, the steady-state target value δu_a of each subsystem is not necessarily optimal for the whole large-scale system. Define the steady-state target vector as $u_{a,ss} = [u_{1,ss}, u_{2,ss}, \dots, u_{M,ss}]$. According to the economic optimization problem of each subsystem, the global optimization problem can be chosen as

$$\min J_{\sum} = \min \sum_{a \in \mathbb{M}} J_a(\delta u_{a,ss}), \quad (39)$$

where $[J_1(\delta u_{1,ss}), J_2(\delta u_{2,ss}), \dots, J_M(\delta u_{a,ss})]$ represents the performance index of each subsystem. Their cost functions are assumed to be Lipschitz continuous. Seen from (39), it is a typical multiobjective optimization problem. The Pareto optimality strategy is utilized to handle the multiobjective problem with conflicting cost functions in [29]. The Pareto optimal solution is obtained by Definition 1.

Definition 1. (steady-state Pareto solution [29]). A feasible solution $(\delta u_{a,ss}^p)$ of multiobjective optimization problem (39) is said to be Pareto optimal if and only if there exists no other feasible solution $(\delta u_{a,ss}^f)$ such that $J_a(\delta u_{a,ss}^f) \leq J_a(\delta u_{a,ss}^p), \forall a \in \mathbb{M}$, and $J_a(\delta u_{a,ss}^f) < J_a(\delta u_{a,ss}^p)$ for at least one index $a \in \mathbb{M}$.

The surface of the Pareto solution set is usually called the Pareto frontier. In practice, it is time-consuming to construct the Pareto frontier. Since the optimal point selected along the Pareto frontier will change with the conditions, the ideal Pareto optimum may be unreachable. The utopia-tracking MPC [28] is utilized to handle this problem.

Definition 2 (steady-state Utopia point [30]). The steady-state utopia point is the ideal optimal solution $(\delta u_{a,ss}^L)$ under the condition of $J_a(\delta u_{a,ss}^L)$, with $a \in \mathbb{M}$. $\delta u_{a,ss}^L$ is given by solving the following problem:

$$\min_{\delta u_{a,ss}^L} J_a(\delta u_{a,ss}), \quad s.t. (13), a \in \mathbb{M} \quad (40)$$

where the utopia cost of each subsystem a is denoted as J_a^L .

Considering the conflict of economic goals among subsystems, the utopia value is not reachable. Here, a compromise solution is used instead, which is the closest point from the Pareto frontier to the utopia value. The compromise solution is defined as $\delta u_{a,ss}^C$, which can be easily calculated by the following distance problem:

$$\begin{aligned} \min_{\delta u_{a,ss}} \|J_a(\delta u_{a,ss}) - J_a^L\|_p, \\ s.t. (13), \quad a \in \mathbb{M}. \end{aligned} \quad (41)$$

Equation (41) is regarded as the steady-state utopia-tracking problem [28]. The schematic of the utopia-tracking approach is shown in Figure 1.

Notice that for the single-objective case, the compromise solution coincides with the utopia point, that is, $J_1(\delta u_{1,ss}) = J_1^L$.

The compromise solution $\delta u_{a,ss}^C$ calculated by equation (41) is taken as the steady-state target value of subsystem a . Substituting into equation (13), the set-point of $y_{a,ss}$ is obtained. Then, the steady-state value $(\delta u_{a,ss} = \delta u_{a,ss}^C, y_{a,ss})$ is sent to dynamic control.

4. Distributed Dynamic Control

In the dynamic control, update the actual control move at each time step so that the output predicted value closely

tracks the steady-state value $\Delta u_{a,ss}^C$. For each subsystem a , $Y_{a,N}^{ol}(k|k)$ is given. Combining the real prediction equation (8) and the open-loop dynamic prediction equation (11), the closed-loop prediction equation of each subsystem a is obtained:

$$Y_{a,N}(k|k) = Y_{a,N}^{ol}(k|k) + \mathcal{S}_a \Delta \tilde{\mathbf{u}}_a(k|k), a \in \mathbb{M} \quad (42)$$

where

$$\begin{aligned} Y_{a,N}(k|k) &= \begin{bmatrix} y_a(k+1|k) \\ y_a(k+2|k) \\ \vdots \\ y_a(k+N|k) \end{bmatrix}, \\ \Delta \tilde{\mathbf{u}}_a(k|k) &= \begin{bmatrix} \Delta u_a(k|k) \\ \Delta u_a(k+1|k) \\ \vdots \\ \Delta u_a(k+N_c-1|k) \end{bmatrix}, \\ \mathcal{S}_a &= \begin{bmatrix} S_1^{u_a} & 0 & \cdots & 0 \\ S_2^{u_a} & S_1^{u_a} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ S_{N_c}^{u_a} & \cdots & S_2^{u_a} & S_1^{u_a} \\ \vdots & & \vdots & \vdots \\ S_N^{u_a} & \cdots & S_{N-N_c+2}^{u_a} & S_{N-N_c+1}^{u_a} \end{bmatrix}, \end{aligned} \quad (43)$$

\mathcal{S}_a is called the dynamic matrix. N is the prediction horizon, and $N_c \leq N \leq N^f$. Equation (42) is the closed-loop prediction equation.

In order to ensure the output to track the steady-state targets as close as possible, meanwhile suppress the controlled variable increment $\delta u_{a,ss}$, the cost function is chosen as

$$\begin{aligned} J_a(k) &= \sum_{i=1}^N \|y_a(k+i|k) - y_{a,ss}(k)\|_{Q_i(k)}^2 \\ &\quad + \sum_{j=0}^{N_c-1} \|\Delta u_a(k+j|k)\|_{\Lambda}^2, \end{aligned} \quad (44)$$

where the weighting matrix Λ is given off-line. $Q_i(k) = \text{diag}\{q_{i1}(k)^2, q_{i2}(k)^2, \dots, q_{i,n_y}(k)^2\}$, updated online.

Besides, an additional constraint to limit $\delta u_{a,ss}^C(k)$ is added to the dynamic control optimization problem:

$$\mathcal{L} \Delta \tilde{\mathbf{u}}_a(k|k) = \delta u_{a,ss}^C(k), \quad (45)$$

where $\mathcal{L} = [I \ I \ \cdots \ I]$. In addition, the inequality constraints including manipulated variable magnitude, manipulated variable increment, and controlled variable magnitude constraints are considered:

$$\begin{aligned}
|\Delta u_a(k+j|k)| &\leq \Delta \bar{u}_a, \quad 0 \leq j \leq M-1, \\
\bar{u} &\leq u_a(k-1) + \sum_{l=0}^j \Delta u_a(k+l|k) \leq \bar{u}, \\
0 &\leq j \leq M-1, \\
y_{a,0}' &\leq y_a^{ol}(k+i|k) + \mathcal{S}_i \Delta \bar{u}_a(k|k) \leq \bar{y}_0'(k), \\
1 &\leq i \leq N,
\end{aligned} \tag{46}$$

where \mathcal{S}_i denotes the i th row of dynamic matrix \mathcal{S} :

$$\begin{aligned}
\bar{y}_{a,0}'(k) &= \max\{\bar{y}_{a,0}, y_{a,ss}(k)\}, \\
y_{a,0}'(k) &= \min\left\{y_{-a,0}, y_{a,ss}(k)\right\}.
\end{aligned} \tag{47}$$

At time k , the following optimization problem of each subsystem is firstly solved:

$$\min_{\Delta \bar{u}_a(k|k)} J_a(k), \text{ s.t. (45), (46), } a \in \mathbb{M}. \tag{48}$$

If the optimization is infeasible, slack variables $\underline{\varepsilon}_{dc}(k)$ and $\bar{\varepsilon}_{dc}(k)$ are added into the cost function and then the optimization problem is modified to

$$\begin{aligned}
J_a'(k) &= \sum_{i=1}^N \|y_a(k+i|k) - y_{a,ss}(k)\|_{Q_i}^2 \\
&+ \sum_{j=0}^{M-1} \|\Delta u_a(k+j|k)\|_{\Lambda}^2 + \|\underline{\varepsilon}_{dc}(k)\|_{\underline{\omega}}^2 \\
&+ \|\bar{\varepsilon}_{dc}(k)\|_{\bar{\omega}}^2,
\end{aligned} \tag{49}$$

where $\underline{\varepsilon}_{dc}(k)$ and $\bar{\varepsilon}_{dc}(k)$ are slack variables. The related constraints on slack variables are as follows:

$$\underline{\varepsilon}_{dc}(k) \leq \underline{y}_{a,0}'(k) - \underline{y}_{a,0,h}, \quad 1 \leq i \leq N, \tag{50}$$

$$\bar{\varepsilon}_{dc}(k) \leq \bar{y}_{a,0,h} - \bar{y}_{a,0}'(k), \quad 1 \leq i \leq N. \tag{51}$$

By adding the constraints equations (50) and (51), the optimization problem with slack variables is expressed as follows:

$$\begin{aligned}
\min_{\underline{\varepsilon}_{dc}(k), \bar{\varepsilon}_{dc}(k), \Delta \bar{u}_a(k|k)} J_a'(k), \\
\text{s.t. (45), (46), (50), (51)}.
\end{aligned} \tag{52}$$

Equations (48) and (52) are standard quadratic problem (QP). By solving equations (48) or (52), $\Delta \bar{u}_a^*(k|k)$ is obtained. Then, $u_a(k|k) = u_a(k|k-1) + \Delta \bar{u}_a^*(k|k)$ is sent to each subsystem and implemented.

Remark 1. In this paper, the considered distributed structured matrix dynamic control (DMC) is a heuristic algorithm. The dynamic is described by the finite step response (FSR) model, which is not accurate. The model error is compensated in a feedback correction term. The stability has not been stressed in the study of DMC since it was proposed by Garcia in 1986. The distributed double-layered DMC provides a solution for the predictive control of complex industrial systems with multiobjects and multilayered rates.

The authors in this paper focus on the economic efficiency and computation burden, while the ingredients for proving stability such as the terminal constraint set, terminal cost function, and feedback control law are not mentioned. On the other hand, the optimization problem, in this paper, is solved by linear programming (LP) or quadratic programming (QP); therefore, the global optimal can be obtained.

5. Case Study

The distributed double-layered DMC algorithm is applied to the Shell benchmark problem for simulation investigation. In the operation of Shell heavy oil fractionator, the controlled variables and manipulated variables need to meet various requirements. As shown in Figure 2, there are three product extraction ports and three side cyclic refluxes. The product concentrations coming out of the top and side extraction ports are determined by the economics and operating requirements. For the cyclic refluxes, the product separation is performed by removing heat. The exchange heat flows back to reboil the columns in other parts of the plant. An enthalpy controller is added to regulate heat removal in the bottom while the heat exchanges of the other two refluxes are taken as the disturbances to the column.

A three-input-three-output model was presented in [31] for the heavy oil fractionator:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = G^u(s) \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + G^d(s) \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}, \tag{53}$$

where

$$G^u(s) = \begin{bmatrix} \frac{4.05e^{-27s}}{50s+1} & \frac{1.77e^{-28s}}{60s+1} & \frac{5.88e^{-27s}}{50s+1} \\ \frac{5.39e^{-18s}}{50s+1} & \frac{5.72e^{-14s}}{60s+1} & \frac{6.90e^{-15s}}{40s+1} \\ \frac{4.38e^{-20s}}{33s+1} & \frac{4.42e^{-22s}}{44s+1} & \frac{7.20}{19s+1} \end{bmatrix}, \tag{54}$$

$$G^d(s) = \begin{bmatrix} \frac{1.20e^{-27s}}{45s+1} & \frac{1.44e^{-27s}}{40s+1} \\ \frac{1.52e^{-15s}}{25s+1} & \frac{1.83e^{-15s}}{20s+1} \\ \frac{1.14}{27s+1} & \frac{1.26}{32s+1} \end{bmatrix},$$

where the manipulated variables are the top extraction u_1 , the side extraction u_2 , and the enthalpy controller for the bottom of the column u_3 . d_1 and d_2 are the middle reflux and the top reflux, respectively. The controlled variables include top product concentration y_1 , side product concentration y_2 , and bottom reflux temperature y_3 . All the variables are normalized, and the unit of time is minute.

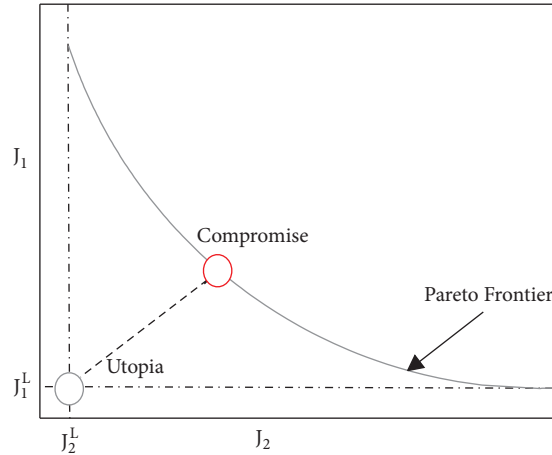


FIGURE 1: Schematic of the utopia-tracking approach.

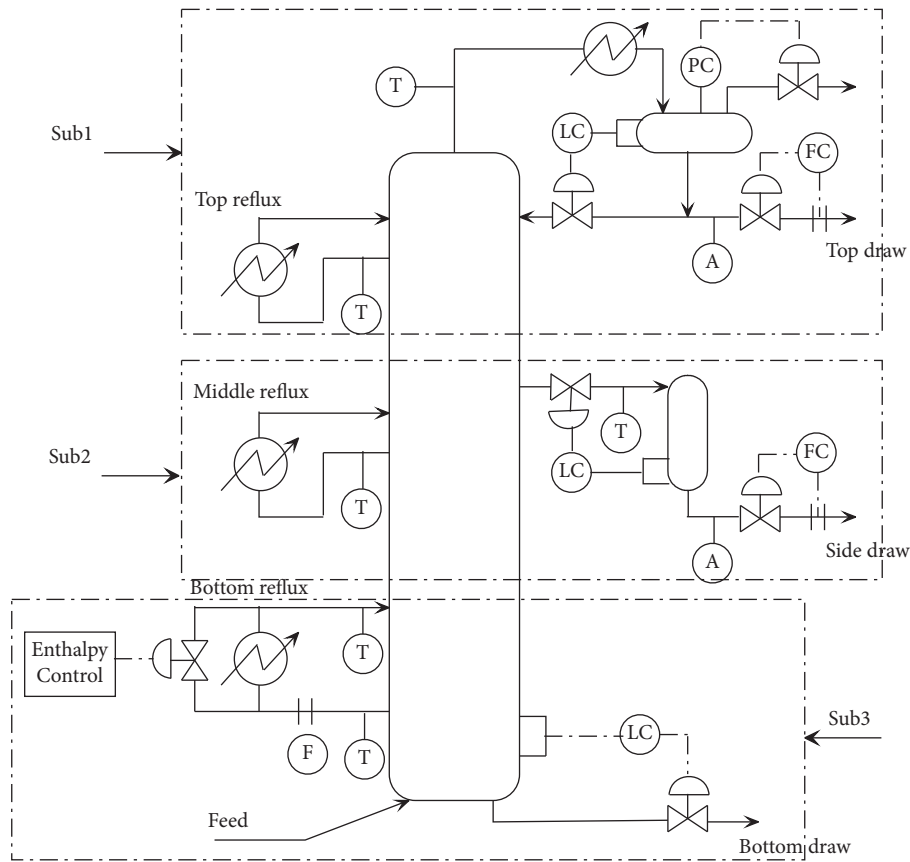


FIGURE 2: Shell heavy oil fractionator.

As seen from Figure 2, by applying the interconnection decomposition, the transfer function $G^u(s)$ in equation (53) can be decomposed into the following three subsystems:

Subsystem 1: $4.05e^{-27s}/50s + 1$

Subsystem 2: $5.72e^{-14s}/60s + 1$

Subsystem 3: $7.20/19s + 1$

In the distributed structure, the coupling effects from neighbor subsystems are considered in the control of each

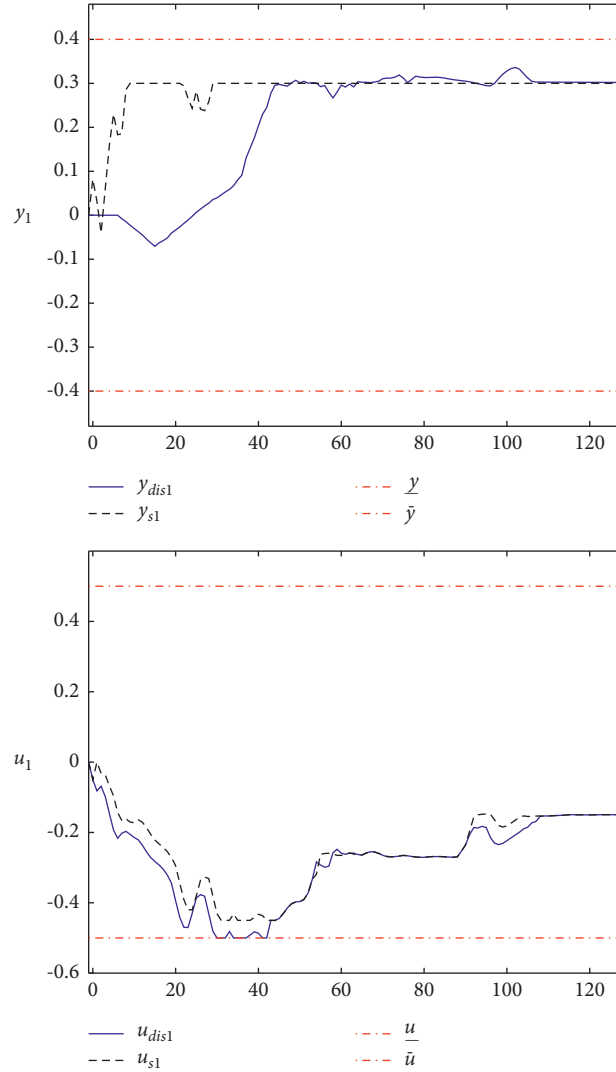


FIGURE 3: Closed-loop responses of subsystem 1.

subsystem. The sampling period is 4 minutes. The finite step response model of each subsystem can be obtained by

$$\begin{aligned}
 y_a(k) = & \sum_{i=1}^{N-1} S_i^{u_a} \Delta u_a(k-i) + S_N^{u_a} u_a(k-N) \\
 & + \sum_{l \neq a} \left[\sum_{i=1}^{N-1} S_i^{u_{al}} \Delta u_l(k-i) + S_N^{u_{al}} u_l(k-N) \right] \\
 & + \sum_{i=1}^{N-1} S_a^d \Delta d(k-i) + S_N^d d(k-N),
 \end{aligned} \quad (55)$$

where $a = 1, 2, 3$ denotes the number of the subsystem. $S_i^{u_a}$ and $S_i^{u_{al}}$ are the step response coefficient matrices on manipulated variables for local subsystem a and neighbor subsystem l , respectively. Denote $S_N^{u_a}$ and $S_N^{u_{al}}$ as the steady-state step response coefficient matrices.

Based on the open-loop prediction of each subsystem a , the corresponding steady-state prediction is obtained by

$$y_{a,ss}(k) = S_N^{u_a} \delta u_{a,ss}(k) + y_{a,ss}^{ol}(k), \quad (56)$$

where $\delta u_{a,ss}$ denotes the steady-state increment of manipulated variable.

The constraints on manipulated variables and controlled variables are specified by $\underline{u}_a = [-0.5, -0.2, -0.35]$; $\bar{u}_a = [0.5, 0.2, 0.35]$; $\Delta \bar{u}_a = \delta \bar{u}_{a,ss} = [0.1, 0.1, 0.1]$; $\underline{y}_{j,h} = [-0.4, -0.4, -0.4]$; $\bar{y}_{j,h} = [0.4, 0.4, 0.4]$; $y_{j,0} = -0.5$; $\bar{y}_{j,0} = 0.5$, $j = 1, 2, 3$. The external targets of $y_{1,ss}$, $y_{2,ss}$, and $u_{3,ss}$ are 0.5. By invoking LP, the steady-state target value of each subsystem a is solved by the following optimization problem:

$$\min_{\delta u_{a,ss}, U_2(k)} J_a = \sum_{i=1,3} h_i \delta u_{i,ss}(k) + h_2 U_2(k). \quad (57)$$

Due to the conflicts of the optimization goals of different subsystems, the steady-state target value of each subsystem $\delta u_{a,ss}$ cannot be achieved at the same time. The utopia-

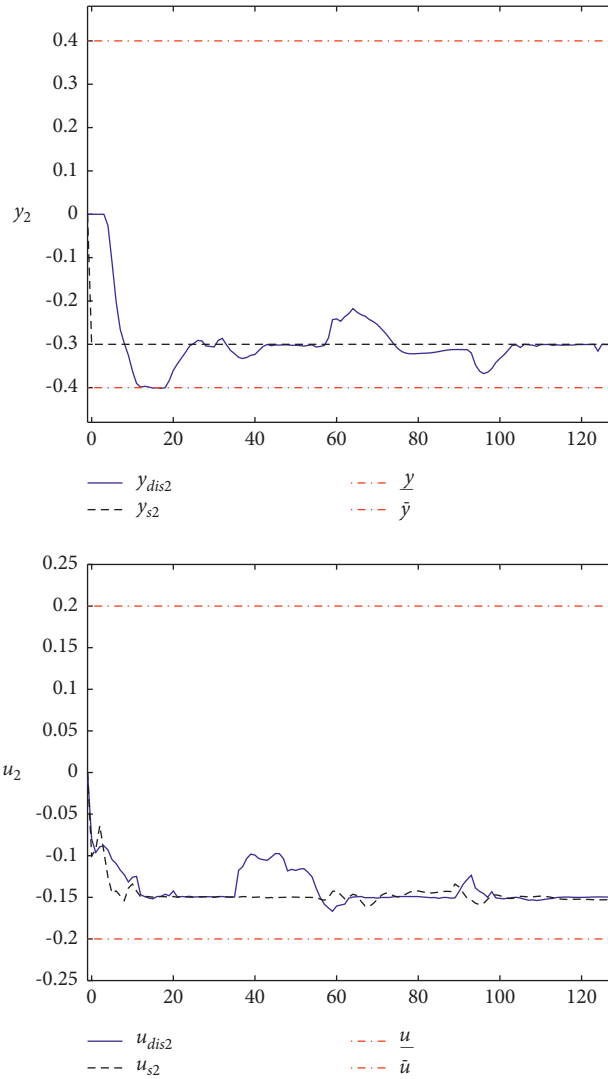


FIGURE 4: Closed-loop responses of subsystem 2.

tracking strategy [28] is adopted to solve the multiple optimization problem with conflicts. According to equations (40) and (41), the compromise solution of steady-state target is calculated. In the dynamic matrix control, by utilizing QP, the actual control move for each subsystem is obtained.

The first investigation is carried out with the performance of the proposed distributed double-layered DMC algorithm with constant disturbance. At time $55 \leq k \leq 90$, the value of disturbance is $[0.15; 0.2]$. In the distributed algorithm, the prediction and the control horizon for distributed double-layered DMC are 8 and 5, respectively. The weighting matrices in dynamic control are selected as $\Lambda = \text{diag}\{1; 4; 1\}$, $Q = \text{diag}\{4.0; 0.67; 1.0\}$. Take $u_1(-1) = u_2(-1) = u_3(-1) = 0$, $y_1(0) = y_2(0) = y_3(0) = 0$. The closed-

loop system responses of three subsystems are shown in Figures 3–5, respectively.

It can be observed that the proposed distributed dynamic prediction model can predict effectively the output of each subsystem at future time. In the distributed structure, the steady-state values of each subsystem obtained by Pareto-tracking satisfy all the constraints. When the calculated control move is implemented in each subsystem, even in the presence of disturbance, the corresponding output can be settled to the optimal steady-state value, which indicates that the control is offset-free. Meanwhile, all the outputs and control moves satisfy the physical constraints and operational restrictions. The results indicate that the proposed distributed double-layered DMC algorithm is effective.

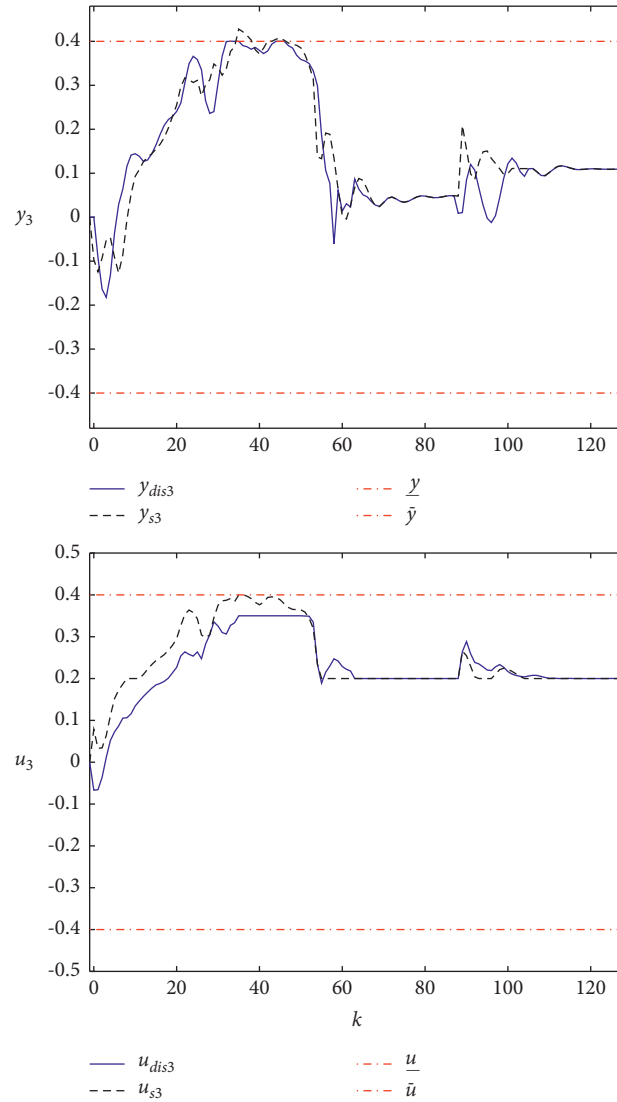


FIGURE 5: Closed-loop responses of subsystem 3.

Next, we will step into the study of the performance of the proposed algorithm. A comparison with the proposed distributed double-layered DMC algorithm and the centralized algorithm is carried out. In both of these algorithms, constant disturbance is considered, respectively. At time $55 \leq k \leq 90$, the value of disturbance is $[0.15; 0.2]$. For the centralized algorithm, by tuning parameters, $N = 8$, $N_c = 5$,

the tracking performance is best. The simulation results are shown in Figure 6, and the performance is quantified in Table 1. As seen from Figure 6, though the output of each subsystem starts with a slight bigger error from the steady-state target, as the iteration goes on, there is an obvious improvement in the tracking performance of the distributed algorithm compared with that of the centralized algorithm.

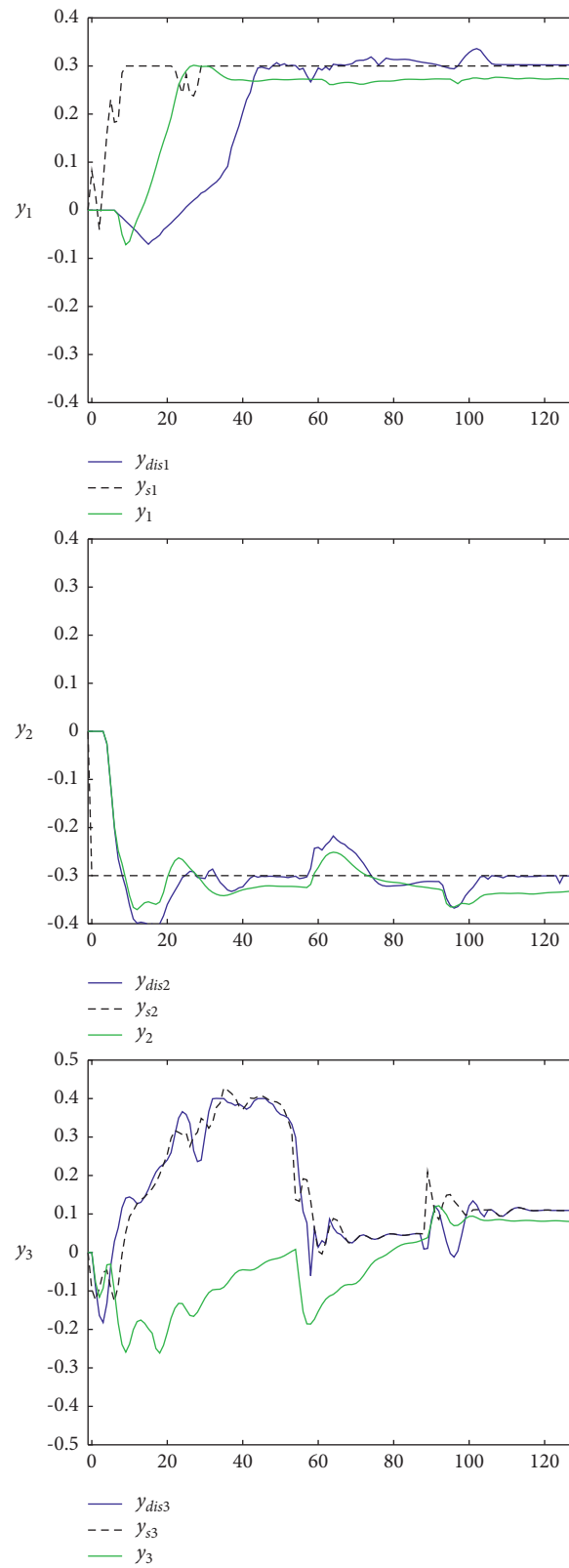


FIGURE 6: Tracking performance of the two algorithms.

TABLE 1: Performance comparison.

Algorithm	Track.-dev.	Time-consuming
Cen. DDMC	0.3811	24.0034
Dis. DDMC	0.3542	6.572

As indicated by the time-consuming in Table 1, the proposed distributed algorithm requires much less computation time than the centralized one. As seen from Table 1, there is a (7.06 % increase in RMSE) improvement in tracking performance as a whole.

6. Discussion and Conclusion

A distributed double-layered DMC scheme for the large-scale system is presented, where the coupling effects from interconnected subsystems are considered in the distributed open-loop prediction equation of each subsystem. The distributed economic optimization problem is treated as a multiple objective optimization problem. Then, the utopia-tracking strategy is adopted to calculate the optimal steady-state target values of each subsystem. The proposed distributed double-layered DMC algorithm is demonstrated through simulation studies. The results show that the computational burden is significantly reduced, and the tracking performance is improved. This algorithm can also be extended to the control of other industrial processes.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Key Research and Development Program of China under Grant 2018YFB1700100.

References

- [1] D. Q. Mayne, J. B. Rawlings, C. Rao, and P. O. M. Sokaert, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [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] D. E. Kassmann, T. A. Badgwell, and R. B. Hawkins, "Robust steady-state target calculation for model predictive control," *AIChE Journal*, vol. 46, no. 5, pp. 1007–1024, 2000.
- [4] K. R. Muske, "Steady-state target optimization in linear model predictive control," in *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*, pp. 3597–3601, Albuquerque, NM, USA, June 1997.
- [5] C. V. Rao and J. B. Rawlings, "Steady states and constraints in model predictive control," *AIChE Journal*, vol. 45, no. 6, pp. 1266–1278, 1999.
- [6] R. Scattolini, "Architectures for distributed and hierarchical model predictive control - a review," *Journal of Process Control*, vol. 19, no. 5, pp. 723–731, 2009.
- [7] H. Sun, T. Zou, J. Liu, and M. au, "Double-layer model predictive control integrated with zone control," *ISA Transactions*, vol. 114, no. 2021, pp. 206–216, 2021.
- [8] H. G. Pan and B. C. Ding, "A strategy of steady-state target calculation to guarantee feasibility of dynamic control in double-layered model predictive control," *Acta Automatica Sinica*, vol. 40, no. 10, pp. 2108–2114, 2014.
- [9] K. R. Muske and T. A. Badgwell, "Disturbance modeling for offset-free linear model predictive control," *Journal of Process Control*, vol. 12, no. 5, pp. 617–632, 2002.
- [10] U. Maeder, F. Borrelli, and M. Morari, "Linear offset-free model predictive control," *Automatica*, vol. 45, no. 10, pp. 2214–2222, 2009.
- [11] A. N. Venkat, J. B. Rawlings, and S. J. Wright, "Distributed model predictive control of large-scale systems," in *Assessment and Future Directions of Nonlinear Model Predictive Control* Springer, Berlin, Germany, 2007.
- [12] X. N. Du, Y. G. Xi, and S. Y. Li, "Distributed Model Predictive Control for Large-Scale Systems," in *Proceedings of the American Control Conference*, IEEE, Arlington, VA, USA, June 2001.
- [13] Y. Zheng, S. Li, and N. Li, "Distributed model predictive control over network information exchange for large-scale systems," *Control Engineering Practice*, vol. 19, no. 7, pp. 757–769, 2011.
- [14] R. A. Shalmani, M. Rahmani, and N. Bigdeli, "Nash-based robust distributed model predictive control for large-scale systems," *Journal of Process Control*, vol. 88, no. 88, pp. 43–53, 2020.
- [15] L. N. Tan, "Distributed H_{∞} optimal tracking control for strict-feedback nonlinear large-scale systems with disturbances and saturating actuators," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 11, pp. 4719–4731, 2020.
- [16] L. N. Tan, "Event-triggered distributed H_{∞} constrained control of physically interconnected large-scale partially unknown strict-feedback systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 4, pp. 2444–2456, 2021.
- [17] D. Jia and B. H. Krogh, "Distributed model predictive control," in *Proceedings of the American Control Conference*, pp. 25–27, Arlington, VA, USA, June 2001.
- [18] C. Wang and C.-J. Ong, "Distributed model predictive control of dynamically decoupled systems with coupled cost," *Automatica*, vol. 46, no. 12, pp. 2053–2058, 2010.
- [19] X. Liu, Y. Shi, and D. Constantinescu, "Distributed model predictive control of constrained weakly coupled nonlinear systems," *Systems & Control Letters*, vol. 74, pp. 41–49, 2014.
- [20] M. Farina and R. Scattolini, "Distributed predictive control: a non-cooperative algorithm with neighbor-to-neighbor communication for linear systems," *Automatica*, vol. 48, no. 6, pp. 1088–1096, 2012.
- [21] P. Trodden and A. Richards, "Cooperative distributed mpc of linear systems with coupled constraints," *Automatica*, vol. 49, no. 2, pp. 479–487, 2013.
- [22] C. Conte, C. N. Jones, M. Morari, and M. N. au, "Distributed synthesis and stability of cooperative distributed model predictive control for linear systems," *Automatica*, vol. 69, pp. 117–125, 2016.
- [23] D. D. Santana, M. A. Martins, and D. Odloak, "An efficient cooperative-distributed model predictive controller with

- stability and feasibility guarantees for constrained linear systems,” *Systems & Control Letters*, vol. 141, no. 2020, pp. 104701–104711, 2020.
- [24] 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.
- [25] B. T. Stewart, J. B. Rawlings, and S. J. Wright, “Hierarchical cooperative distributed model predictive control,” in *Proceedings of the American Control Conference Baltimore*, pp. 3963–3968, MD, USA, July 2010.
- [26] M. Farina and R. Scattolini, “Distributed non-cooperative MPC with neighbor-to-neighbor communication,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 404–409, 2011.
- [27] E. C. Kerrigan, A. Bemporad, D. Mignone, and J. M. au, “Multi-objective prioritisation and reconfiguration for the control of constrained hybrid systems,” *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, vol. 4, no. 2, pp. 290–296, 2000.
- [28] V. M. Zavala and A. Flores-Tlacuahuac, “Stability of multi-objective predictive control: a utopia-tracking approach,” *Automatica*, vol. 48, no. 10, pp. 2627–2632, 2012.
- [29] A. Chinchuluun and P. M. Pardalos, “A survey of recent developments in multiobjective optimization,” *Annals of Operations Research*, vol. 154, no. 1, pp. 29–50, 2007.
- [30] I. E. Grossmann, R. Drabbant, and R. K. Jain, “Incorporating toxicology in the synthessi of industrial chemical complexes,” *Chemical Engineering Communications*, vol. 17, no. 1-6, pp. 151–170, 1982.
- [31] D. M. Prett and M. Morari, *The Shell Process Control Workshop*, pp. 673–678, Butterworths, Boston, 1987.