mathematics

MDPI

Article

# Enhancing Strong Neighbor-Based Optimization for Distributed Model Predictive Control Systems

**Shan Gao** [1,2], **Yi Zheng** [1,2,*] and **Shaoyuan Li** [1,2,*]

[1] Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China; gaoshan1117@sjtu.edu.cn

[2] Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China

* Correspondence: yizheng@sjtu.edu.cn (Y.Z.); syli@sjtu.edu.cn (S.L.)

Received: 1 April 2018; Accepted: 8 May 2018; Published: 22 May 2018

**Abstract:** This paper considers a class of large-scale systems which is composed of many interacting subsystems, and each of them is controlled by an individual controller. For this type of system, to improve the optimization performance of the entire closed-loop system in a distributed framework without the entire system's information or too-complicated network information, connectivity is always an important topic. To achieve this purpose, a distributed model predictive control (DMPC) design method is proposed in this paper, where each local model predictive control (MPC) considers the optimization performance of its strong coupling subsystems and communicates with them. A method to determine the strength of the coupling relationship based on the closed-loop system's performance and subsystem network connectivity is proposed for the selection of each subsystem's neighbors. Finally, through integrating the steady-state calculation, the designed DMPC is able to guarantee the recursive feasibility and asymptotic stability of the closed-loop system in the cases of both tracking set point and stabilizing system to zeroes. Simulation results show the efficiency of the proposed DMPC.

**Keywords:** model predictive control; distributed model predictive control; large-scale systems; neighborhood optimization

## 1. Introduction

There is a class of complex large-scale industrial control systems which are composed of many interacting and spatially distributed subsystems, and each subsystem is controlled by an individual controller (e.g., large-scale chemical process [1], smart micro-grid [2,3] systems, distributed generation systems [4]), where the controllers exchange information with each other through a communication network. The objective is to achieve a good global performance of the entire closed-loop system or a common goal of all subsystems by the controller network. This objective is usually to track setpoints with minimized total error or to stabilize the entire system to zeroes in the dynamic control layer.

Distributed model predictive control (DMPC) controls each subsystem by an individual local model predictive control (MPC), and is one of the most important distributed control or optimization algorithms [1,5–8], since it not only inherits MPC's ability to get good optimization performance and explicitly accommodate constraints [9,10], but also has the advantages of a distributed framework of fault-tolerance, less computation, and being flexible to system structure [7,11–14]. However, compared with the centralized control scheme, its performance is still not as good as that of centralized MPC for coupling systems in a peer-to-peer distributed control framework.

Many algorithms and design methods have appeared in the literature for different types of systems and for different problems in the design of DMPCs. For example, the design of DMPC for nonlinear systems [15,16], DMPC for uncertain systems [15,17], DMPC for networked systems with time delay [18], a decentralized optimization algorithm for solving DMPC [19], the design of cooperative strategies for improving the performance of DMPC [20], the design of an event-based communication DMPC for reducing the load on the communication network [21], as well as the design of a DPMC control structure [22]. Among these algorithms, several DMPC algorithms relate to the purpose of improving the closed-loop optimization performance while considering the information connectivity [5,21,23–26]. Information connectivity is considered because it directly affects the structural flexibility and error tolerance ability. Reference [27] proposed a DMPC where each subsystem-based MPC only communicates with its directly-impacted neighbors and uses an iterative algorithm to obtain the "Nash optimality". References [20,28,29] proposed cooperative DMPC, where each MPC considers the cost of the entire system and communicates with all the other MPCs to obtain "Pareto optimality". To reduce the information connectivity and increase the structural flexibility, Reference [30] proposed that each subsystem optimize all the subsystems impacted by it over the optimization horizon. The solution of this method is equal to the cooperative DMPC, while its communication efforts are less than the cooperative DMPC, especially for sparse systems. References [31,32] gave a strategy to dynamically adjust the weighting of performance index in cooperative MPC to avoid bad performance occurring in some subsystems. In an effort to achieve a trade-off between the optimization performance of the entire system and the information connectivity, an intuitively appealing strategy, called impacted-region cost optimization-based DMPC, is proposed in [33–35], where each subsystem-based MPC only considers the cost of its own subsystem and those of the subsystems directly impacted by it. Consequently, each MPC only communicates with its neighboring MPCs. In addition, in some papers, the control flexibility and information connectivity are paid more attention by researchers. References [14,36] provide a tube-based DMPC where all interactions are considered as disturbances and each subsystem-based MPC is solved independently. It does not exchange the state and input trajectory, but the interaction constraints, to avoid the interaction consistency problem. This method is able to improve the flexibility and fault tolerance ability of the control network [37]. References [25,37] proposed reconfigurable DMPC and plug-and-play DMPC based on dissipative theory, which focus on the problem of how to design a DMPC which allows the addition or deletion of subsystems without any change in existing controllers. It can be seen that the optimization performance of the entire system and structural flexibility are two conflicting key points in DMPC design. The selection of the range of each subsystem's neighbors to be optimized in each subsystem-based MPC is important in the design of DMPC in order to obtain good optimization performance without unnecessary information connections. Thus, the aim of this paper is to design an algorithm to determine the range of each subsystem optimized from the point of view of enlarging each subsystem MPC's feasible region, then to improve the entire system's optimization performance without too-complicated network connectivity. Then, based on the result of this algorithm, we aim to design a stabilized neighborhood optimization-based DMPC that handles state constraints and is able to be used in target tracking.

As for target tracking, the difficulty in DMPC is to guarantee the recursive feasibility. References [38–40] provide a tracking algorithm for a series of MPC systems, where a steady-state target optimizer (SSTO) is integrated in the design of the cost function. The proposed controller is able to drive the whole system to any admissible setpoint in an admissible way, ensuring feasibility under any change of setpoint. As for distributed systems, [38] gives a DMPC for tracking based on the method introduced in [39] and a cooperative DMPC strategy. Reference [41] proposes another method based on global calculations of targeting tracking. It does not require a feasible starting point of each distributed predictive controller. These methods provide good references and possible methods for designing a tracking DMPC that considers optimization performance improvement and network connectivity.

In this paper, strong coupling neighbor-based optimization DMPC is proposed. With this method, each local MPC coordinates and communicates with its strong coupling neighbors. It takes its strongly-coupling downstream subsystems cost function into account in its cost function to improve the performance of the entire closed-loop system. To reduce the unnecessary network connectivity, the interaction terms of weak coupling upstream neighbors are ignored in its predictive model and are considered as bounded disturbances. In addition, the closed-loop optimization performance is used to determine which interaction should be regarded as strong coupling and be considered in DMPC. The strategy proposed in [38] is used to guarantee the recursive feasibility and stability in the target tracking problem. An asymptotically-stable closed-loop system with state constraints is guaranteed.

The remainder of this paper is organized as follows. Section 2 describes the problem to be solved. Section 3 describes the design of the proposed DMPC. Section 4 analyzes the stability of the closed-loop system. Section 5 presents the simulation results to demonstrate the effectiveness of the proposed algorithm. Finally, a brief conclusion to the paper is drawn in Section 6.

## 2. Problem Description

Considering a large-scale discrete-time linear system which is composed of many interacting subsystems, the overall system model is:

$$\begin{cases} x^{+} = Ax + Bu, \\ \quad y = Cx, \end{cases} \tag{1}$$

where $x \in \mathcal{R}^{n_x}$ is the system state, $u \in \mathcal{R}^{n_u}$ is the system current control input, $y \in \mathcal{R}^{n_y}$ is the controlled output, and $x^{+}$ is the successor state. The state of the system and control input applied at sample time $t$ are denoted as $x(t)$,$u(t)$, respectively. Moreover, there are hard constraints in the system state and control input. That is, for $\forall t \geq 0$:

$$x(t) \in \mathcal{X}, u(t) \in \mathcal{U}, \tag{2}$$

where $\mathcal{X} \subset \mathcal{R}^{n_x}$ and $\mathcal{U} \subset \mathcal{R}^{n_u}$ are compact convex polyhedra containing the origin in their interior.

Given Model (1), without loss of generality, the overall system is divided in to $m$ subsystems, denoted as $\mathcal{S}_i, i \in \mathcal{I}_{0:m}$. Thus, $u = (u_1, u_2, ..., u_m)$ and $x = (x_1, x_2, ..., x_m)$, then the subsystem model for $\mathcal{S}_i, \forall i \in \mathcal{I}_{0:m}$ is:

$$x_i^{+} = A_{ii}x_i + B_{ii}u_i + \textstyle\sum_{j \in \mathcal{N}_i} B_{ij}u_j, \tag{3}$$

where $\mathcal{N}_i$ is the set of subsystems that send inputs to the current subsystem $\mathcal{S}_i$. For subsystem $\mathcal{S}_j, j \in \mathcal{N}_i$, $\mathcal{S}_j$ couples with $\mathcal{S}_i$ by sending control input $u_i$ to $\mathcal{S}_i$. In particular, $j \in \mathcal{N}_i$ if $B_{ij} \neq 0$. Given the overall system constraints set $\mathcal{X}, \mathcal{U}$, $x_i, u_i$ fit hard constraints $x_i(t) \in \mathcal{X}_i, u_i(t) \in \mathcal{U}_i$.

In this paper, for ease of analysis, here the definitions of neighbor (upstream-neighbor) and downstream neighbor are given.

**Definition 1.** *Given subsystem $\mathcal{S}_i$ with state evolution Equation (3), define $\mathcal{S}_j, \mathcal{S}_j \in \mathcal{N}_i$, which send input information to $\mathcal{S}_i$ as the neighbor (upstream neighbor) of $\mathcal{S}_i$. Moreover, for arbitrary $\mathcal{S}_j, \mathcal{S}_j \in \mathcal{N}_i$, since $\mathcal{S}_i$ receives input information from $\mathcal{S}_j$, $\mathcal{S}_i$ is defined as a downstream neighbor of $\mathcal{S}_j$.*

Denote the tracking target as $y_t$. Assume that $(A, B)$ is stabilizable and the state is measurable. The aim of a tracking problem given a target $y_t$ is to design a controller which enables $y(t) \rightarrow y_t$ in an admissible way when $t \rightarrow \infty$. Hence, the origin control objective function of the overall system is:

$$V_{N_{origin}}(x, y_t; u) = \sum_{k=0}^{N-1} (\|Cx(k) - \hat{y}_t\|_{Q_o}^2 + \|u(k) - \hat{u}_t\|_R^2) + \|Cx(N) - \hat{y}_t\|_{P_o}^2, \tag{4}$$

where $P_o > 0$, $Q_o > 0$, and $R > 0$ is the weighting coefficients matrix, and $u_t$ is steady input corresponding to $y_t$.

The problem considered here is to design a DMPC algorithm to control a physical network, which coordinate with each other considering the following performance indicators:

- to achieve a good optimization performance of the entire closed-loop system.
- to guarantee the feasibility of target tracking.
- to simplify the information connectivity among controllers to guarantee good structural flexibility and error-tolerance of the distributed control framework.

To solve this problem, in this paper, an enhanced strong neighbor-based optimization DMPC is designed, and is detailed in the next section.

## 3. DMPC Design

In an interacting distributed system, the state evolution of each subsystem is affected by the optimal control decisions of its upstream neighbors. Each subsystem considers if these effects will help to improve the performance of entire closed-loop system. On the other hand, these impacts have different strengths for different downstream subsystems. Some of the effects are too small and can be ignored. If these weakly-coupling downstream subsystems' cost functions are involved in each subsystem's optimization problem, additional information connections arise with little improvement of the performance of the closed-loop system. The increase of information connections will hinder the error tolerance and flexibility of the distributed control system. Thus, each subsystem-based MPC takes the cost functions of its strongly-interacting downstream subsystems into account to improve the closed-loop performance of the entire system and receive information from its strong-coupling neighbors.

### *3.1. Strong-Coupling Neighbor-Based Optimization for Tracking*

Given that the coupling degrees between different subsystems differ substantially, here we enable the subsystem to cooperate with strong-coupling neighbors while treating the weak-coupling ones as disturbance. Define $\mathcal{N}_{i(strong)}$ as a set of strong-coupling neighboring subsystems and $\mathcal{N}_{i(weak)}$ as set of weak-coupling neighbors. The rule for deciding strong-coupling systems is detailed in Section 3.4.

Then, for $\mathcal{S}_i$, we have:

$$x_i^+ = A_{ii}x_i + B_{ii}u_i + \sum\nolimits_{j\in\mathcal{N}_{i(strong)}} B_{ij}u_j + w_i, \tag{5}$$

where

$$w_i = \sum\nolimits_{j\in\mathcal{N}_{i(weak)}} B_{ij}u_j,$$
$$w_i \in \mathcal{W}_i, \mathcal{W}_i = (\oplus B_{ij}\mathcal{U}_j),$$
$$\mathcal{N}_{i(weak)} \cup \mathcal{N}_{i(strong)} = \mathcal{N}_i = \{j|B_{ij} \neq 0, j \neq i\}.$$

The deviation $w_i$ represents the influence collection of weak-coupling upstream neighbors in $N_{i,(weak)}$. $w_i$ is contained in a convex and compact set $\mathcal{W}_i$ which contains the origin.

If the weak coupling influence $w_i$ is neglected, a simplified model based on $\mathcal{S}_i$ is acquired. That is:

$$\bar{x}_i^+ = A_{ii}\bar{x}_i + B_{ii}\bar{u}_i + \sum\nolimits_{j\in\mathcal{N}_{i(strong)}} B_{ij}\bar{u}_j. \tag{6}$$

Here $\bar{x}_i$ , $\bar{u}_i$, and $\bar{u}_j, j \in \mathcal{N}_{i(strong)}$ represent the state and input of a simplified subsystem model which neglects weak-coupling upstream neighbors' influence $w_i$.

The simplified overall system model with new coupling relation matrix $\bar{B}$ is:

$$\bar{x}^+ = A\bar{x} + \bar{B}\bar{u}, \tag{7}$$

where $\bar{x} = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_m)$ and $\bar{u} = (\bar{u}_1, \bar{u}_2, ..., \bar{u}_m)$ represent states and inputs in this simplified model.

Considering the target-tracking problem of the simplified model, in order to ensure the output track, given target $y_t$, constraints are given for terminal state prediction. If the current target $y_t$ is set as the tracking target through the controller optimization, when $y_t$ changes, the terminal constraints need to change immediately. The optimal solution at a previous time may not fit the terminal constraints brought by the changed $y_t$. This violates the recursive feasibility of the system. Thus, here a steady state optimization is integrated in the MPC for tracking where an artificial feasible tracking goal $y_s$ is proposed as a medium variable. This variable works as an optimized variable. With setting tracking point $y_s$ equal to the previous target, the recursive feasibility will not be violated by the target change.

The medium target $y_s$ and its state $\bar{x}_S$ and input $\bar{u}_s$ should satisfy the simplified system's steady state equations. It has

$$\begin{bmatrix} A - I_{n_x} & \bar{B} & 0 \\ C & 0 & -I \end{bmatrix} \begin{bmatrix} \bar{x}_s \\ \bar{u}_s \\ y_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{8}$$

$$\begin{bmatrix} \bar{x}_s & \bar{u}_s \end{bmatrix} = M_y y_s. \tag{9}$$

Here $M_y$ is a suitable matrix. That is, target $y_s$'s corresponding inputs $\bar{u}_s$ and states $\bar{x}_s$ in the simplified model can be expressed by $y_s$. The equation is based on the premise of Lemma 1.14 in [42]. If Lemma 1.14 does not hold, a $M_\theta$ and $\theta$ which fits $\begin{bmatrix} \bar{x}_s & \bar{u}_s \end{bmatrix} = M_\theta \theta$ can be found, which can replace the $y_s$ as a variable to be solved.

For the manual tracking target $y_s$ for the overall system, we have $y_s = \{y_{1,s}, \ldots, y_{i,s}, \ldots, y_{m,s}\}$. That is, given $y_s$, arbitrary subsystem $\mathcal{S}_i$ gets a subtracking target $y_{s,i}$. Similar to (9), $\bar{x}_{s,i}, \bar{u}_{s,i}$ are solved.

With the simplified model and artificial tracking target $y_{s,i}$, according to (9), in the strong-coupling neighbor-based optimization MPC algorithm, the objective function optimized in subsystem $\mathcal{S}_i, \forall i \in [1, m]$ is set as $V'_{iN}(x_i, y_t; x_i, u_{i,0:N-1}, y_s)$ as follows:

$$\begin{aligned} V'_{iN}(x_i, y_t; x_i, u_{i,0:N-1}, y_s) = & \sum_{k=0}^{N-1} (\|x_i(k) - \bar{x}_{i,s}\|^2_{Q_i} + \|u_i(k) - \bar{u}_{i,s}\|^2_{R_i}) + \|x_i(N) - \bar{x}_{i,s}\|^2_{P_i} + V_0(y_{i,s}, y_{t,i}) \\ & + \sum_{k=0}^{N-1} \sum_{h \in \mathcal{H}_i} \|x_h(k) - \bar{x}_{s,h}\|^2_{Q_h} + \|x_h(N) - \bar{x}_{s,h}\|_{P_h}, \end{aligned} \tag{10}$$

where $x_i, y_t$ is the given initial state and target, $u_{i,0:N-1}$ are input predictions in $0 : N-1$ sample time ahead. $y_s$ is the admissible target. $Q_i = C'_i Q_{o,i} C_i > 0$ and

$$\mathcal{H}_i = \{h | i \in \mathcal{N}_{h(strong)}, \forall \mathcal{S}_h, h \in [1, m], h \neq i\}. \tag{11}$$

Here, $\mathcal{S}_i$'s controller design takes the strong-coupling downstream neighbors' performances as part of its optimized objective. That is, the current subsystem $\mathcal{S}_i$'s optimal solution is decided by its own and downstream neighbors in set $\mathcal{H}_i$, which is strongly impacted by $\mathcal{S}_i$.

Next, we will use the simplified model in (6) with only strong couplings to solve the tracking problem (10) for each subsystem. To guarantee control feasibility and stability, the following definitions and assumptions are given.

One important issue is to deal with the deviation caused by neglecting weak-coupling neighbor inputs. Here robust positively invariantt sets are adopted to enable the deviation of states to be bounded and the real system's states to be controlled in $\mathcal{X}$.

**Definition 2.** *(Robust positively invariant set control law) Given $e = (x - \bar{x})$ which represents the dynamics of the error between the origin plant and the simplified model:*

$$e^+ = A_k e + w, \tag{12}$$

*with $A_k = (A + BK)$. A set $\phi$ is called a robust positively invariant set for system (12) if $A_k\phi \oplus \mathcal{W} \subseteq \phi$, and the control law is called a robust positively invariant set control law.*

The definition of a robust positively invariant set illustrates that for system $x = Ax + Bu + w$ if $\phi$ and robust positively invariant set control law $K$ exist, then for $e(0) = x(0) - \bar{x}(0)$, the trajectories of the original system at arbitrary time $t$ denoted as $x(t)$ can be controlled in $x(t) = \bar{x}(t) \oplus \phi$.

Based on this definition, in this paper the dynamics of deviation $(x_i - \bar{x}_i)$ introduced by neglecting weak-coupling neighbors can be solved. For subsystem $\mathcal{S}_i$ proposed as (5), the deviation is written as:

$$e_i^+ = A_{ii}e_i + B_{ii}u_{i,e} + w_i,$$

where $e_i = x_i - \bar{x}_i$ is the deviation from the simplified model to the original model and $u_{i,e}$ is the control law. There exists the set $\phi_i$ as a robust positively invariant set for $\mathcal{S}_i$ if $(A_{ii} + B_{ii}K_i)e_i \in \phi_i$ for all $e_i \in \phi_i$ and all $w_i \in \mathcal{W}_i$. Here $u_{i,e} = K_i e_i$ is a feedback control input and we denote $K_i$ as the robust positively invariant set control law for $\mathcal{S}_i$. Then, it is easy to obtain $x_i(t) = \bar{x}_i(t) \oplus \phi_i$ for time $t$. Let $(\bar{x}_i(t), \bar{u}_i(t)) \in \mathcal{F}_i$, where $\mathcal{F}_i = (\mathcal{X}_i \times \mathcal{U}_i) \ominus (\phi_i \times K_i\phi_i)$, the origin system state and input satisfy $(x_i(t), K_i(x_i(t) - \bar{x}_i(t)) + \bar{u}_i(t)) \in \mathcal{X}_i \times \mathcal{U}_i$. Thus, with the help of a robust positively invariant set, the original system optimization is transferred to a simplified model. For the overall system, we have $K = diag(K_1, K_2, ..., K_m)$.

With Definition 2, if the deviation brought by omitting weak-coupling neighbors is controlled in a robust positively invariant (RPI) set $\phi_i$ with control law $K_i$ and simplified model in (7) has control law and state $\bar{u}_i, \bar{x}_i$ confined in $\mathcal{U}_i \ominus K_i\phi_i, \mathcal{X}_i \ominus \phi_i$, respectively, the local subsystem will have a feasible solution for the optimization.

As for the manually-selected tracking target $y_s$, based on the overall simplified model in (7), the following definition is given:

**Definition 3.** *(Tracking invariant set control law). Consider that overall system (7) is controlled by the following control law:*

$$\bar{u} = \bar{K}(\bar{x} - \bar{x}_s) + \bar{u}_s = \bar{K}\bar{x} + Ly_s. \tag{13}$$

*Let $A + \bar{B}\bar{K}$ be Hurwitz, then this control law steers system (7) to the steady state and input $(\bar{x}_s, \bar{u}_s) = M_y y_s$. $\bar{K}$ is denoted as the tracking invariant set control law.*

Denote the set of initial state and steady output that can be stabilized by control law (13) while fulfilling the system constraints throughout its evolution as an invariant set for tracking $\Omega_{\bar{K}}$. For any $(x(0), y_s) \in \Omega_{\bar{K}}$, the trajectory of the system $\bar{x}^+ = A\bar{x} + B\bar{u}$ controlled by $\bar{u} = \bar{K}x + Ly_s$ is confined in $\Omega_{\bar{K}}$ and tends to $(x_s, u_s) = M_y y_s$.

Under Definitions 2 and 3, before introducing the enhancing strong neighbor-based optimization DMPC, some assumptions for the closed-loop system feasibility and stability are given as follows. The concrete theorem and an analysis of stability and feasibility are given in Section 4.

**Assumption 1.** *The eigenvalues of $A_{ii} + B_{ii}K_i$ are in the interior of the unitary circle. $\phi_i$ is an admissible robust positively invariant set for $\mathcal{S}_i$'s deviation $(x_i - \bar{x}_i)$ subject to constraints $\mathcal{F}_i$, and the corresponding feedback control law is $u_{i,e} = K_i e_i$.*

**Assumption 2.** *Let $\Omega_{\bar{K}}$ be a tracking invariant set for the simplified system (7) subject to constraints $\mathcal{F} = \{\{(\bar{x}_1, \bar{u}_1), ..., (\bar{x}_m, \bar{u}_m)\} | \forall i, (\bar{x}_i, \bar{u}_i) \in (\mathcal{X}_i \times \mathcal{U}_i) \ominus (\phi_i \times \bar{K}_i \phi_i)\}$ , and the corresponding feedback gain matrix is $\bar{K} = \{\bar{K}_1, \bar{K}_2, \dots, \bar{K}_m\}$.*

**Assumption 3.** *For $Q = block\text{-}diag\{Q_1, Q_2, \dots, Q_m\}$, $R = block\text{-}diag\{R_1, R_2, \dots, R_m\}$ and $P = block\text{-}diag(P_1, P_2, \dots, P_m)$, it has:*

$$(A + \bar{B}\bar{K})' P (A + \bar{B}\bar{K}) - P = -(Q + \bar{K}' R \bar{K}). \tag{14}$$

Assumption 1 ensures that with the feedback control law $u_{i,e} = K_i e_i$, $i \in \mathcal{I}_{0:m}$, the state estimated by the simplified model (7) is near to the real system's trajectory before the system reaches the target. In Assumption 2, $\Omega_{\bar{k}}$ is set as a terminal constraint of DMPC. Assumption 3 is used in the proof of the convergence of system presented in the Appendix A.

So far, the strong-coupling neighbor-based optimization DMPC algorithm, which is solved iteratively, can be defined as follows:

Firstly, denote the optimal objective of subsystem $\mathcal{S}_i$ as $V_{iN}$. According to (10), at iterating step $p$, $V_{iN}$ fits:

$$\begin{aligned}
&V_{iN}(x_i, y_t, p; \bar{x}_i, \bar{u}_{i,0:N-1}, y_{i,s}) \\
=& \sum_{k=0}^{N-1} (\|\bar{x}_i(k) - \bar{x}_{i,s}\|^2_{Q_i} + \|\bar{u}_i(k) - \bar{u}_{i,s}\|^2_{R_i}) + \|\bar{x}_i(N) - \bar{x}_{i,s}\|^2_{P_i} + V_0(y_{i,s}, y_{i,t}) \\
&+ \sum_{k=0}^{N-1} \sum_{h \in \mathcal{H}_i} \left\| \bar{x}_h(k) - \bar{x}^{[p-1]}_{h,s} \right\|^2_{Q_h} + \left\| \bar{x}_h(N) - \bar{x}^{[p-1]}_{h,s} \right\|_{P_h}.
\end{aligned} \tag{15}$$

Compute the optimization solution

$$(\bar{x}_i'(0), \bar{u}_{i,0:N-1}{}', y_{i,s}{}') = \arg\min V_{iN}(x_i, y_t, p; \bar{x}_i, \bar{u}_{i,0:N-1}, y_{i,s}), \tag{16}$$

Subject to constraints:

$$\bar{x}_{h_i}(k+1) = A_{h_i h_i} \bar{x}_{h_i}(k) + \textstyle\sum_{h_j \in \mathcal{N}_{h(strong)}} B_{h_j} \bar{u}^{[p]}_{h_j}(k) + B_{h_i h_i} \bar{u}_{h_i}(k),$$

$$(\bar{x}_{h_i}(k)\ \bar{u}_{h_i}(k)) \in \mathcal{F}, \mathcal{F} : (\mathcal{X}_{h_i}, \mathcal{U}_{h_i}) \ominus (\mathcal{W}_{h_i}, K_{h_i} \mathcal{W}_i), \tag{17a}$$

$$(\bar{x}(N), y_s) \in \Omega_{\bar{K}}, \tag{17b}$$

$$\bar{x}_i(0) \in x_i - \phi_i, \tag{17c}$$

$$M_y y_{i,s} = (\bar{x}_{i,s}, \bar{u}_{i,s}), \tag{17d}$$

with $h_i \in \mathcal{H}_i \cup \{i\}$, and $\phi_i, \Omega_{\bar{k}}$ defined in Assumptions 2 and 3, respectively. The optimization function (16) updates $\mathcal{S}_i$'s initial state, inputs in $N$ steps $\bar{u}_{i,0:N-1}$ and current tracking target $y_{i,s}$ based on the information from subsystems in $\mathcal{H}$.

Secondly, set

$$\bar{u}^{[p]}_{i,0:N-1} = \gamma_i \bar{u}'_{i,0:N-1} + (1 - \gamma_i) \bar{u}^{[p-1]}_{i,0:N-1}, \tag{18}$$

$$y^{[p]}_{i,s} = \gamma_i y'_{i,s} + (1 - \gamma_i) y^{[p-1]}_{i,s}, \tag{19}$$

$$\bar{x}^{[p]}_i(0) = \gamma_i x'_i(0) + (1 - \gamma_i) \bar{x}^{[p-1]}_i(0), \tag{20}$$

$$\sum_{i=1}^{m} \gamma_i = 1, \gamma_i > 0. \tag{21}$$

$\gamma_i \in \mathcal{R}, 0 < \gamma_i < 1$ is to guarantee the consistency of the optimization problem. That is, at the end of the current sample time, all shared variables converge.

After that, we take

$$p = p + 1$$

to iterate until the solutions convergence. Then, we have $\bar{x}_i^* = \bar{x}_i^{[p]}$, $\bar{u}_i^* = \bar{u}_{i,0:N-1}^{[p]}, y_{i,s}^* = y_{i,s}^{[p]}$ .

Finally, when the solution converges, according to Assumption 1, take the control law of $\mathcal{S}_i$ as

$$u_{i,0}^* = \bar{u}_{i,0}^* + K_i(x_i - \bar{x}_i^*), \tag{22}$$

where $K_i$ is the robust positively invariant set control law. $\bar{u}_{i,0}^*$ is the first element of $\bar{u}_i^*$. For better understanding, the algorithm is also presented in Algorithm 1.

**Algorithm 1:** Enhancing Strong Neighbor-Based Optimization DMPC

```
Data: initial time $t_0$, inital state $x_i(t_0)$, and tracking target $y_{target}$ (target can be changed with
      time according to production demand) for subsystem $\mathcal{S}_i$
Result: the control law $u_i^*(t)$ for $t = t_0 : +\infty$
Firstly, determine the strong-coupling neighbors set $N_{i(strong)}$ by solving $\mathcal{C}_{i,(d^*)}$'s optimization
  in 3.4.
Secondly, confirm $\mathcal{S}_i$'s downstream neighbor set $\mathcal{H}_i$ which is the set of subsystems that $SS_i$
  has control influence on.
Set $t = t_0$.
while True do
    Select Warm Start (reference solution at iteration $p = 0$, details in Section 3.2) :
        $v_i^{[0]}(t) = (\bar{x}_i^{[0]}(0|t), \bar{u}_{i,0:N-1}^{[0]}(t), \bar{y}_{i,s}^{[0]}(t))$
    Set $x_i = x_i(t), y_t = y_{target}, p = 1$.
    while True do
        Set $(\bar{x}_i'(0), \bar{u}_{i,0:N-1}{}', y_{i,s}{}') = \arg\min\{V_{iN}(x_i, y_t, p; \bar{x}_i, \bar{u}_{i,0:N-1}, y_{i,s}) : s.t(17)\}$
        Get optimization solution at $p$:
            $\bar{u}_{i,0:N-1}^{[p]} = \gamma_i \bar{u}_{i,0:N-1}' + (1 - \gamma_i)\bar{u}_{i,0:N-1}^{[p-1]}$
            $y_{i,s}^{[p]} = \gamma_i y_{i,s}' + (1 - \gamma_i) y_{i,s}^{[p-1]}$
            $\bar{x}_i^{[p]}(0) = \gamma_i x_i'(0) + (1 - \gamma_i)\bar{x}_i^{[p-1]}(0)$
            $\sum_{i=1}^{m} \gamma_i = 1, \gamma_i > 0$
        if $||\bar{u}_i^{[p]} - \bar{u}_i^{[p-1]}|| \leq 1e^{-6}$ then
            Break
        end
        Set $p = p + 1$
    end
    Set $\bar{x}_i^* = \bar{x}_i^{[p]}(0), \bar{u}_i^* = \bar{u}_{i,0:N-1}^{[p]}, y_{i,s}^* = y_{i,s}^{[p]}$. $u_{i,0}^* = \bar{u}_{i,0}^* + K_i(x_i - \bar{x}_i^*)$
    Get $u_i^*(t) = u_{i,0}^*$. Set $t = t + 1$.
end
```

In this algorithm, we use an iterative strategy to guarantee the distributed control solution $(\bar{x}(0), \bar{u}_{0:N-1}, y_s)$ is consistent. Next, the selection of warm start, the given solution for each subsystem at initial iterative step 0, is proposed in the next section.

### 3.2. Warm Start

Considering a new sample time, with updated system states, the choice of a warm start is based on the principle that it fits the simplified system's constraints in (17), so that real subsystem solution's feasibility is guaranteed. The warm start is designed as the following algorithm:

**Algorithm 2:** Warm Start for Iterative Algorithm

```
Data: $x(t+1), y_t, \bar{u}^*_{i,0:N-1}(t), y^*_{i,s}(t)$
Result: the warm start $v_i^{[0]}(t+1) = (\bar{x}_i^{[0]}(0|t+1), \bar{u}^{[0]}_{i,0:N-1}(t+1), \bar{y}^{[0]}_{i,s}(t+1))$
For control inputs, two options are given:
Option 1: $\tilde{u}_{i,0:N-1}(t+1) = (\bar{u}^*_{i,1}(t), ..., \bar{u}^*_{i,N-1}(t), \bar{u}_i(N))$
where $\bar{u}(N) = (\bar{u}_1(N), ..., \bar{u}_m(N)) = \bar{K}\bar{x}^*(N) + Ly^*_s(t)$.
Option 2:
$\hat{u}_{i,0:N-1}(t+1) = (\hat{u}_i(0), ..., \hat{u}_i(N-1))$, where
        $\hat{x}_i(0) = \bar{x}^*_i(1,t)$, and $\hat{x}_i(k+1)$ fits
        $\hat{x}(k+1) = (\hat{x}_1(k+1), ..., \hat{x}_m(k+1)) = (A + B\bar{K})\hat{x}(k) + BLy^*_s(k), k \in I_{0:N-1}$
        $\hat{u}_i(k) = \bar{K}\hat{x}(k) + Ly^*_s(t), k \in I_{0:N-1}$
As for $\bar{x}_i^{[0]}(0|t+1), y_{i,s}^{[0]}(t+1)$. To ensure feasibility, denote:
        $\bar{x}_i^{[0]}(0|t+1) = \bar{x}^*_i(1|t)$
        $y_{i,s}^{[0]}(t+1) = y^*_{i,s}(t)$
Warm Start
if $(\bar{x}_i^{[0]}(t+1), y_{i,s}^{[0]}(t+1))$ in tracking invariant set, and
        $V_{iN}(x_i(t+1), y_t, 0; \bar{x}_i^{[0]}(0|t+1), \hat{u}_{i,0:N-1}(t+1), y_{i,s}^{[0]}(t+1)$
        $\leq V_{iN}(x_i(t+1), y_t, 0; \bar{x}_i^{[0]}(0|t+1), \tilde{u}_{i,0:N-1}(t+1), y_{i,s}^{[0]}(t+1)$
 then
    $v_i(t+1)^{[0]} = (\bar{x}_i^{[0]}(0|t+1), \hat{u}_{i,0:N-1}(t+1), y_{i,s}^{[0]}(t+1))$
end
else
    $v_i(t+1)^{[0]} = (\bar{x}_i^{[0]}(0|t+1), \tilde{u}_{i,0:N-1}(t+1), y_{i,s}^{[0]}(t+1))$
end
```

The algorithm illustrates that two choices are provided for the warm start. One is acquiring a solution from the tracking invariant set control law $\bar{K}$, with the simplified model prediction $(\bar{x}^*_i(1|t), y^*_{i,s}(t))$ as initial state and tracking target, respectively. The other is taking a solution from the simplified model prediction at time $t$. Both of them fit the constraints of (17). Note that the second option will only be considered when the subsystem enters the tracking invariant set.

### *3.3. RPI Control Law and RPI Set*

Here one constraint coupling subsystem is considered. Given that for $\mathcal{S}_i$ we have $x_i \in \mathcal{X}_i$ and $u_i \in \mathcal{U}_i$, express the constraints in inequalities: $\mathcal{X}_i = \{x_i | |l_i^T x_i| \leq 1\}$ and $\mathcal{U}_i = \{u_i | |h_i^T u_i| \leq 1\}$. The robust positively invariant set $\phi_i$ is denoted as $\phi_i = \{x_i : x_i^T P_i x_i \leq 1\}$.

With the definition a of robust positively invariant set in Definition 2, $\phi_i$ should ensure that $\forall x_i \in \phi_i$, $x_i \in \mathcal{X}_i$. That is:

$$|h_i^T x_i| \leq 1, \forall x_i \in \phi_i. \tag{23}$$

Based on definitions of $\mathcal{N}_{i(strong)}$ and $\mathcal{N}_{i(weak)}$, $\mathcal{W}_i$ is decided according to the constraints of $\mathcal{N}_{i(weak)}$. For deviation caused by neglecting the subsystem in $\mathcal{N}_{i(weak)}$, a minimization of robust positive invariant set $\phi_i$ by introducing a parameter $\gamma_i \in [0, 1]$ can be obtained.

The parameter $\gamma_i$ controls the size of the robust positive invariant set $\phi_i$ by further minimizing $\phi_i$ in $\phi_i \subseteq \sqrt{\gamma_i}\mathcal{X}$. That is:

$$\begin{aligned} &\min \gamma_i \\ s.t. \quad &|h_i^T x_i| \leq \sqrt{\gamma_i}, \forall x_i \in \phi_i. \end{aligned} \tag{24}$$

We should also consider the input constraint $\mathcal{U}_i$:

$$|l_i^T K_i x_i| \leq 1, \forall x_i \in \phi_i, \tag{25}$$

and the constraint brought by the property of robust positive invariant set $\phi_i$ itself should be considered.

Based on the above analysis, referring to [43], we can obtain $\gamma_i$ and $K_i$ by solving the following linear matrix inequality optimization problem:

$$\min_{W_i, Y_i, \gamma_i} \gamma_i, \tag{26}$$

$$\begin{bmatrix} \lambda_i W_i & * & * \\ 0 & 1 - \lambda_i & * \\ A_{ii} W_i + B_i Y_i & w_i & W_i \end{bmatrix} > 0, \forall w_i \in vert(\mathcal{W}_i), \tag{27}$$

$$\begin{bmatrix} 1 & * \\ Y_i^T l_i & W_i \end{bmatrix} > 0, \tag{28}$$

$$\begin{bmatrix} \gamma_i & * \\ W_i h_i & W_i \end{bmatrix} > 0, \tag{29}$$

and $K_i = Y_i W_i^{-1}$. Thus, we get RPI control law $K_i$ and $\gamma_i$, which illustrates the size of $\phi_i$. To get $\phi_i$, we use the procedure in Reference [43].

### *3.4. Determination of Strong Coupling*

There are many measurements to measure the strength of interactions among subsystems. Different measurements lead to different optimization performance. This paper focuses on the performance and connectivity of subsystems. Thus, the determination of strong-coupling neighbors is based on the influence on the size of current subsystem's robust positively invariant (RPI) set and subsystem connectivity.

On the one hand, as defined in Definition 2, $\phi_i$ is a robust positively invariant set for subsystem $\mathcal{S}_i$ described as $x_i^+ = A_{ii} x_i + B_{ii} u_i + \sum_{j \in \mathcal{N}_i} B_{ij} u_j$ when $u_j$ is set to zero. Given that $\phi_i$ deals with deviation caused by neglecting some of the inputs $u_j, j \in \mathcal{N}_i$, the size of $\phi_i$ is expected to be sufficiently small. The benefit is that the solution in (15) can get a larger feasible domain. Here we consider that a sufficiently large domain means the solution has more degrees of freedom and brings better subsystem performance. Based on the idea above, to decide to omit the weak-coupling neighbor set $N_{i(weak)}$, we choose a neighbor collection which results in a small size of robust positively invariant set $\phi_i$. The basis of measuring the robust positively invariant set $\phi_i$ by introducing $\gamma_i$ is mentioned in the previous section. On the other hand, connectivity, as the measurement of subsystem topology complexity, is easy to obtain. Next, we give the numerical analysis.

Denote an arbitrary option for deciding the strong-, weak-coupling neighbors as $\mathcal{C}_{i,(d)}, d \in D_i$. $D_i = \{1, ..., d_{max}\} \in I$ is the label set of ways of $\mathcal{S}_i$'s neighbors' distribution. $d_{max}$ represents the size of feasible distribution methods which fits $d_{max} \leq 2^{size(\mathcal{N}_i)}$. For better understanding of $\mathcal{C}_{i,(d)}$, here we take an arbitrary neighbor set $\mathcal{N}_i = \{j_1, j_2, j_3\}$ as an example. If we treat $j_i$ as a strong-coupling neighbor and $j_2, j_2$ as weak, we have $\exists d \in D_i$, $\mathcal{C}_{i,(d)}$, satisfying:

$$\mathcal{C}_{i,(d)} = \{(\mathcal{N}_{i(strong)}, \mathcal{N}_{i(weak)}) | \mathcal{N}_{i(strong)} = \{j_1\}, \mathcal{N}_{i(weak)} = \{j_2, j_3\}\}.$$

Option $\mathcal{C}_{i,(d)}$ results in a specified connectivity amount (normalized) $c_{i,(d)} \in [0,1]$ and an RPI set denoted as $\phi_{(i,d)} \subseteq \sqrt{\gamma_{i,(d)}}\mathcal{X}_i$. Here $c_{i,(d)} \in [0,1]$ are defined as:

$$c_{i,(d)} = \frac{size(\mathcal{N}_{i(strong)})}{size(\mathcal{N}_i)} \in [0,1]. \tag{30}$$

To find the optimal distribution $\mathcal{C}_{i,(d^*)}$ of strong- and weak-coupling neighbors, here we take:

$$\mathcal{C}_{i,(d^*)} = \underset{\mathcal{C}_{i,(d)}, W_{i,(d)}, Y_{i,(d)}, \gamma_{i,(d)}}{\text{argmin}} \left((\gamma_{i,(1)} + \mu_i c_{i,(1)}), ..., (\gamma_{i,(d)} + \mu_i c_{i,(d)}), ..., (\gamma_{i,(d_{max})} + \mu_i c_{i,(d_{max})})\right),$$

where for $d \in D_i$,

$$\begin{bmatrix} \lambda_i W_{i,(d)} & * & * \\ 0 & 1-\lambda_i & * \\ A_{ii}W_{i,(d)} + B_i Y_{i,(d)} & w_i & W_{i,(d)} \end{bmatrix} > 0, \forall w_i \in vert(\mathcal{W}_{i,(d)}), \tag{31}$$

$$\begin{bmatrix} 1 & * \\ Y_{i,(d)}^T l_i & W_{i,(d)} \end{bmatrix} > 0, \tag{32}$$

$$\begin{bmatrix} \gamma_{i,(d)} & * \\ W_{i,(d)} h_i & W_{i,(d)} \end{bmatrix} > 0, \tag{33}$$

$$0 \leq \gamma_{i,(d)} \leq 1. \tag{34}$$

In this equation, $\mu_i$ is a weight coefficient for the optimization. $\gamma_{i,(d)}, W_{i,(d)}, Y_{i,(d)}, X_{i,(d)}$ represent the $\gamma_i, W_i, Y_i, X_i$ under distribution $\mathcal{C}_{i,(d)}$. Moreover, $\mathcal{C}_{i,(d^*)}$ is the optimal solution.

This optimization means that in order to make the optimal decision on strong-coupling neighbors and weak-coupling neighbors while taking both connectivity and performance into account, the optimization that minimizes the combination of subsystem connectivity and size of $\phi_i$ should be solved. To decide whether a neighbor $\mathcal{S}_j, j \in \mathcal{N}_i$ is a strong-coupling neighbor or a weak one, the size of $\phi_i$ is expected to be small so that the solution in (15) can get a larger feasible domain. At the same time, the connectivity is expected to be small to reduce the system's topological complexity.

The optimization achieves the goal of choosing neighbors which result in smaller size of robust positively invariant set $\phi_i$ and connectivity. Solution $\mathcal{C}_{i,(d^*)}$ reflects the consideration of influence on RPI set $\phi_i$ and connectivity. With this method, even though "weak-coupling" neighbors are omitted and deviation is brought, the simplified model has a large degree of freedom to design the control law of tracking and reduces the connectivity at the same time. Thus, a good system performance and error tolerance can be obtained.

## 4. Stability and Convergence

In this section, the feasibility and stability theorem of strong-coupling neighbor-based DMPC are given. Denote

$$\mathcal{X}_N = \{x \in \mathcal{X} | \exists v = (x, u_{0:N-1}, y_s), u(k) \in \mathcal{U}, k \in \mathcal{I}_{0:N-1}, y_s \in \mathcal{Y}_s, s.t. v \in \mathcal{Z}_N\},$$
$$\mathcal{Z}_N = \{v | u(k) \in \mathcal{U}, k \in \mathcal{I}_{0:N-1}, y_s \in \mathcal{Y}_s, x(k; x, u) \in \mathcal{X}, k \in \mathcal{I}_{0:N-1}, x(N; x, u) \in \Omega_{\bar{K}}\}.$$

$x(k; x, u)$ represents the current time's state prediction after $k$ sample time. $\mathcal{Y}_s$ is the feasible tracking set based on hard constraints of $x$ and $u$.

**Theorem 1.** *Assume that Assumptions 1–3 hold. Then, for all initial state $x(0)$ with tracking target $y_t$ if $v(0) \in \mathcal{Z}_N$, the closed-loop system based on a strong-coupling neighbor-based DMPC algorithm is feasible and asymptotically stable and converges to $\hat{y}_s \oplus C\phi_k$, where $\hat{y}_s = (\hat{y}_{1,s}, ..., \hat{y}_{m,s})$, $\hat{y}_{i,s} = argmin V_0(y_{i,s}, y_{i,t})$ among feasible targets.*

**Proof.** Feasibility is proved by Lemmas A1, A2. Stability's proofs are in Lemmas A3, A4 in the Appendix A. □

## 5. Simulation

The simulation takes an industrial system model with five subsystems interacting with each other as an example. Between different subsystems, the coupling degrees vary substantially. The relationships of subsystems and the designed MPC are shown in Figure 1.

In Figure 1, dotted lines are used to represent weak coupling, while solid lines are used to represent strong coupling. With the strategy we have defined in our paper, weak couplings are neglected. As a result, it can be seen in Figure 1 that only parts of the subsystems are joint in cooperation. Subsystem models are also given as follows:

$$\mathcal{S}_1: \; x_{1,t+1} = \begin{bmatrix} 0.5 & 0.6 \\ 0 & 0.66 \end{bmatrix} x_{1,t} + \begin{bmatrix} 0.1 \\ 0.7 \end{bmatrix} u_{1,t} + \begin{bmatrix} 0 \\ 0.04 \end{bmatrix} u_{2,t},$$
$$y_{1,t} = \begin{bmatrix} 0 & 1 \end{bmatrix} x_{1,t}, \tag{35a}$$
$$\mathcal{S}_2: \; x_{2,t+1} = \begin{bmatrix} 0.6 & 0.1 \\ 0 & 0.71 \end{bmatrix} x_{2,t} + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u_{2,t} + \begin{bmatrix} 0 \\ 0.3 \end{bmatrix} u_{1,t} + \begin{bmatrix} 0 \\ 0.01 \end{bmatrix} u_{3,t},$$
$$y_{1,t} = \begin{bmatrix} 0 & 1 \end{bmatrix} x_{1,t}, \tag{35b}$$
$$\mathcal{S}_3: \; x_{3,t+1} = \begin{bmatrix} 0.7 & 0.2 \\ 0.1 & 0.4 \end{bmatrix} x_{3,t} + \begin{bmatrix} 0.9 \\ 1 \end{bmatrix} u_{3,t} + \begin{bmatrix} 0 \\ 0.4 \end{bmatrix} u_{2,t} + \begin{bmatrix} 0 \\ 0.05 \end{bmatrix} u_{4,t},$$
$$y_{1,t} = \begin{bmatrix} 0 & 1 \end{bmatrix} x_{1,t}, \tag{35c}$$
$$\mathcal{S}_4: \; x_{4,t+1} = \begin{bmatrix} 0.9 & 0.7 \\ 0 & 0.6 \end{bmatrix} x_{4,t} + \begin{bmatrix} 0.4 \\ 0.4 \end{bmatrix} u_{4,t} + \begin{bmatrix} 0.3 \\ 0.6 \end{bmatrix} u_{3,t} + \begin{bmatrix} 0 \\ 0.01 \end{bmatrix} u_{5,t},$$
$$y_{1,t} = \begin{bmatrix} 0 & 1 \end{bmatrix} x_{1,t}, \tag{35d}$$
$$\mathcal{S}_5: \; x_{5,t+1} = \begin{bmatrix} 0.8 & 0 \\ 0.5 & 0.78 \end{bmatrix} x_{5,t} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_{5,t} + \begin{bmatrix} 0.4 \\ 0.2 \end{bmatrix} u_{4,t},$$
$$y_{1,t} = \begin{bmatrix} 0 & 1 \end{bmatrix} x_{1,t}. \tag{35e}$$

By the strong-coupling neighbor-based DMPC, connections including $\mathcal{S}_2 \rightarrow \mathcal{S}_1$, $\mathcal{S}_3 \rightarrow \mathcal{S}_2$, $\mathcal{S}_4 \rightarrow \mathcal{S}_3$, $\mathcal{S}_5 \rightarrow \mathcal{S}_4$ are neglected. For the five subsystems in the given model, $\gamma_1, \gamma_2, \gamma_3, \gamma_4$, and $\gamma_5$, which evaluate the system performance, are obtained by optimization in Section 3.4, they are:

$$(\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5) = (0.54, 0.66, 0.72, 0.53, 0). \tag{36}$$

Among them, $\gamma_5 = 0$ illustrates that subsystem $\mathcal{S}_5$ has no weak-coupling upstream neighbors. Additionally, the robust positively invariant set feedback control laws are

$$\{K_1, K_2, K_3, K_4\} = \{[-0.119 - 0.762]^{\mathrm{T}}, [-0.171 - 0.434]^{\mathrm{T}}, [-0.316 - 0.251]^{\mathrm{T}}, [-0.724 - 0.966]^{\mathrm{T}}\}.$$

The optimization horizon $N$ is 10 sample time. Take $Q = I_{10\times 10}$ and $R = I_{5\times 5}$. To accelerate the iterative process, in both of these iterative algorithms, the terminal conditions of iteration are $||u_i^{[p]} - u_i^{[p-1]}||_2 \leq 10^{-3}$ or $p > 100$. If either of these two conditions is satisfied, iteration terminates.

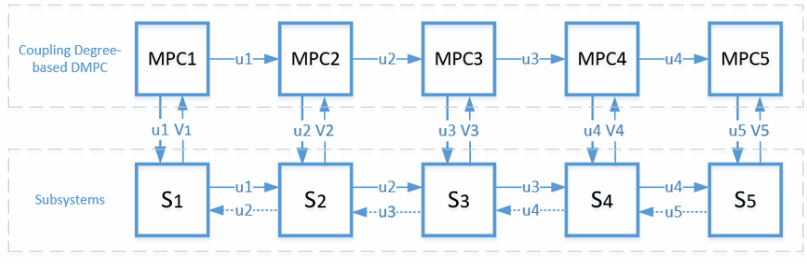

**Figure 1.** An illustration of the structure of a distributed system and its distributed control framework. MPC: model predictive control; DMPC: distributed MPC.

The following shows the system performance when the strong-coupling neighbor-based DMPC algorithm is applied. Here we chose different set-points to detect the system stability. Three groups of setpoints were given to verify the system's feasibility and stability. For a better understanding, cooperative DMPC strategy control results which cooperate with all neighbors are also introduced to make a comparison. The simulation took a total of 74.3 seconds for 90 sampling times. The performance comparison of strong-coupling neighbor-based DMPC (SCN-DMPC) with cooperative DMPC where each subsystem used the full system's information in their controller is shown in Figures 2–4.

Figure 2 shows the state evolution of each subsystem. The two curves of SCN-DMPC and cooperative are close to each other. This is because the weak couplings in the given example are tiny compared with the strong couplings and thus do not have much of an impact on system dynamics. Besides, SCN-DMPC optimization algorithm was always feasible and was able to keep stable with a changing tracking target. Figure 3 shows the input difference between these two algorithms. The control laws of these two algorithms are almost the same. Tracking results are shown in Figure 4. There was a small off-set in subsystem $\mathcal{S}_1, \mathcal{S}_3$, which could be eliminated by adding an observer. All other subsystems could track the steady-state target without steady-state off-set. From the simulation results of Figures 2–4, the stability and good optimization performance of the closed-loop system using SCN-DMPC is verified.

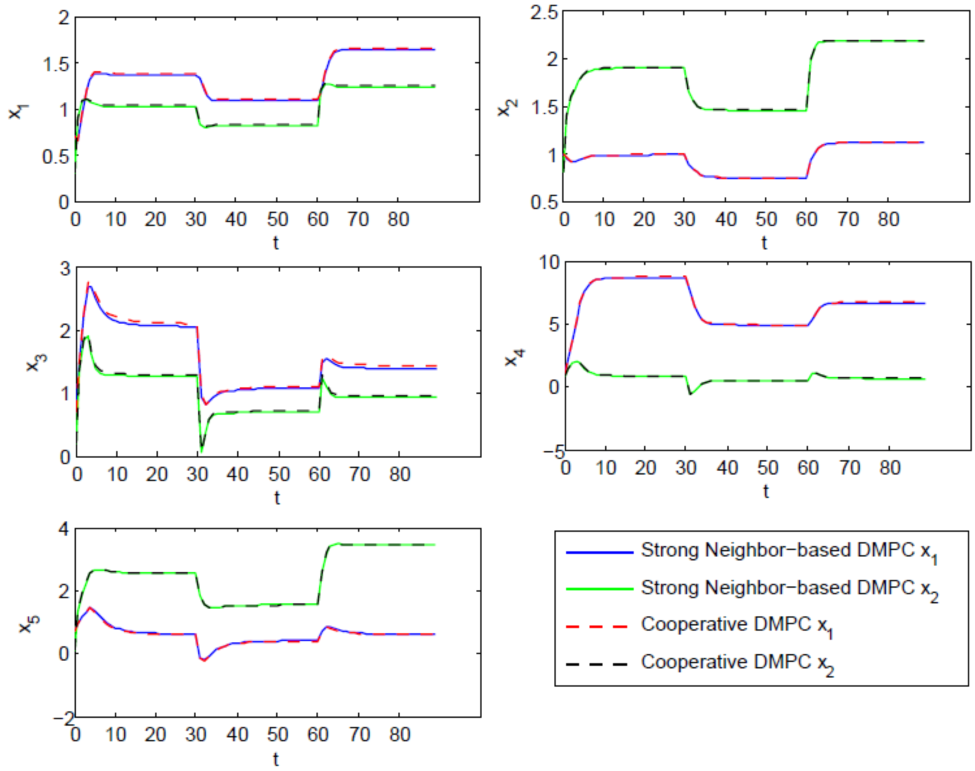

**Figure 2.** States of each subsystem under the control of strong-coupling neighbor-based DMPC (SCN-DMPC) and cooperative DMPC.

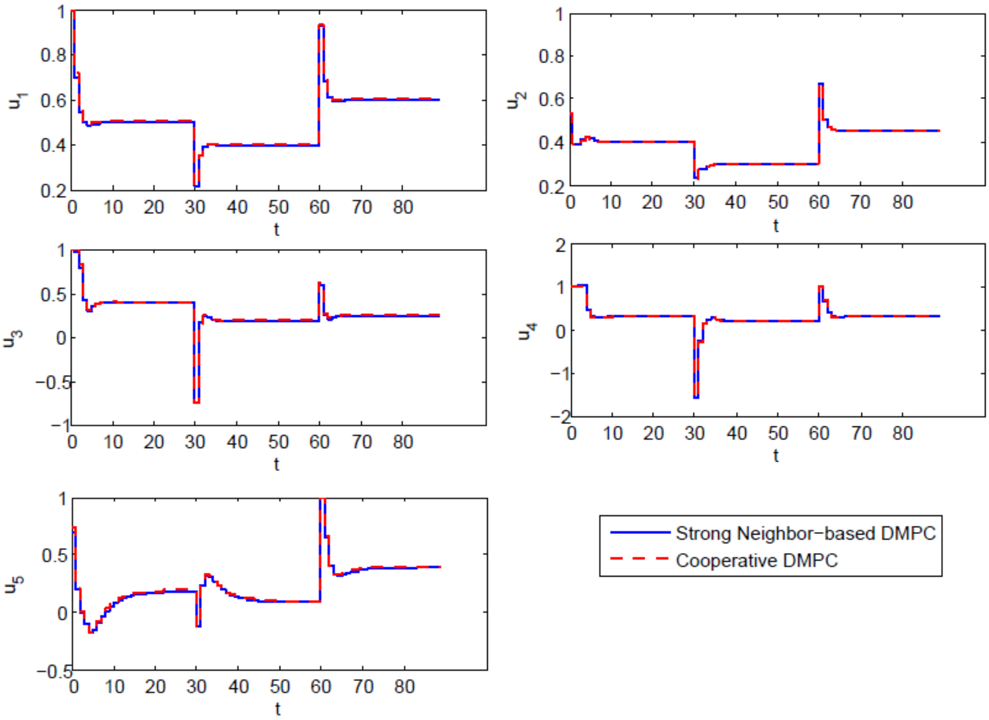

**Figure 3.** Inputs of each subsystem under the control of SCN-DMPC and cooperative DMPC.

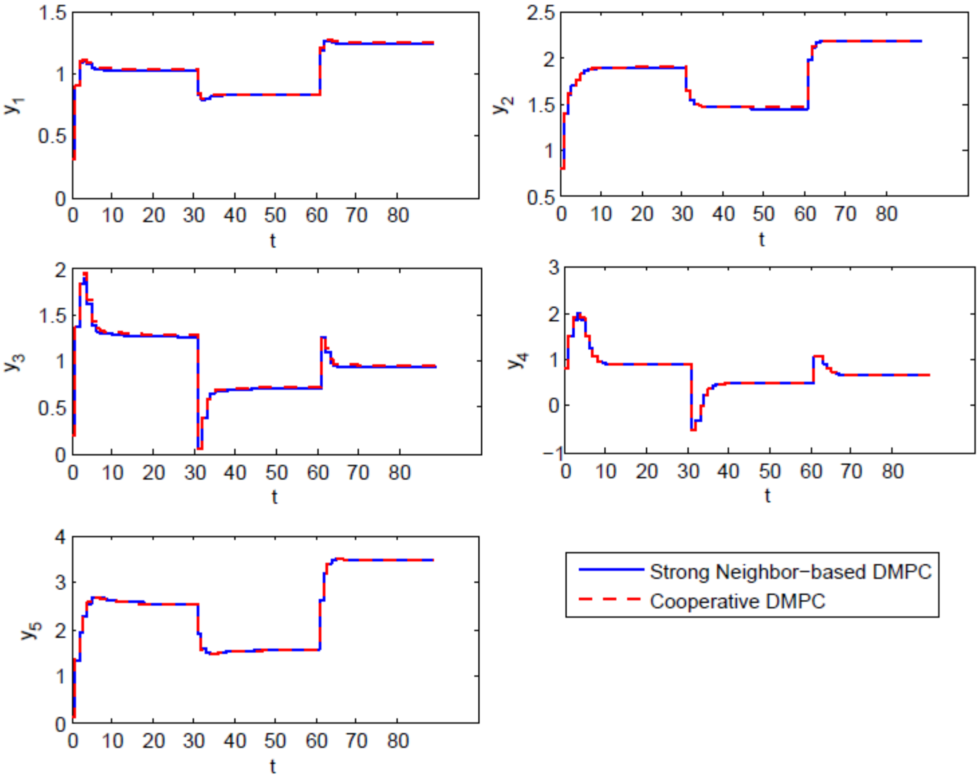

**Figure 4.** Output of each subsystem under the control of SCN-DMPC and cooperative DMPC.

In Figures 2–4, the curves of SCN-DMPC and cooperative DMPC are close to each other. The reason is that the weak couplings in the given example were tiny compared with the strong couplings, and thus they did not have much of an impact on system dynamics, even though a small difference existed. Specifically, given the state equation input weight coefficients in each subsystem, the deviations of the five subsystems fit $\|w_1\| \leq 0.04$, $\|w_2\| \leq 0.01$, $\|w_3\| \leq 0.05$, $\|w_4\| \leq 0.01$. The effects of these disturbances were very small compared with those of each subsystem's inputs. Under robust feedback control law, they do not have much of an influence on system dynamics. Besides, the performance of the simplified model under SCN-DMPC equals to that under the control law optimizing the global performance of simplified system. As a result, the system performance under SCN-DMPC was close to that in cooperative DMPC. Under circumstances where the weak interactions are close to the impact of each subsystem's inputs (which sacrifices part of the performance to achieve less network connectivity), omitting weak couplings may result in a greater influence on system dynamics, and the simulation results can differ.

Moreover, mean square errors between the closed-loop systems with strong-coupling neighbor-based optimization DMPC and cooperative DMPC outputs are listed in Table 1. The total error of five subsystems was only 3.5, which illustrates the good optimization performance of SCN-DMPC.

**Table 1.** Mean square error (MSE) of outputs between SCN-DMPC and cooperative DMPC.

| **Item** | $\mathcal{S}_1$ | $\mathcal{S}_2$ | $\mathcal{S}_3$ | $\mathcal{S}_4$ | $\mathcal{S}_5$ |
|---|---|---|---|---|---|
| MSE | 0.5771 | 1.1512 | 0.7111 | 0.1375 | 0.9162 |

Connectivities are compared in the following table.

Table 2 shows that when strong-coupling neighbor-based DMPC was applied, the total information connections reduced to eight, which means that five connections were avoided compared with cooperative DMPC.

**Table 2.** Comparison of system connectivity with different control methods.

| System | SCN-DMPC | Cooperative DMPC |
|---|---|---|
| $\mathcal{S}_1$ | 1 | 2 |
| $\mathcal{S}_2$ | 2 | 3 |
| $\mathcal{S}_3$ | 2 | 4 |
| $\mathcal{S}_4$ | 2 | 3 |
| $\mathcal{S}_5$ | 1 | 1 |
| $\mathcal{S}$ | 8 | 13 |

Above all, the simulation results show that the proposed SCN-DMPC achieved a good performance close to the cooperative DMPC with a significant reduction of information connectivity.

## 6. Conclusions

In this paper, a strong-coupling neighbor-based optimization DMPC method is proposed to decide the cooperation among subsystems, where each subsystem's MPC considers the optimization performance and evolution of its strong-coupling downstream subsystems and communicates with them. For strongly-coupled subsystems, the influence on state and objective function are considered. For weakly-coupled subsystems, influence is neglected in the cooperative design. A closed-loop system's performance and network connectivity-based method is proposed to determine the strength of coupling relationships among subsystems. The feasibility and stability of the closed-loop system in the case of target-tracking are analyzed. Simulation results show that the proposed SCN-DMPC was able to achieve similar performance in comparison to the DMPC which did not neglect the information or influence of weakly coupling subsystems. At the same time, connectivity was significantly decreased.

**Author Contributions:** Shan Gao developed the main algorithm, contributed to the stability analysis, designed the simulation and prepared the draft of the paper. Yi Zheng and Shaoyuan Li proposed the idea of Enhancing Strong Neighbor-based coordination strategy. They contributed to the main theory of the work and gave the inspiration and guidance of the strong-coupling neighbors' determination, the algorithm design and stability analysis.

**Acknowledgments:** This work is supported by the National Nature Science Foundation of China (61673273, 61590924).

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix

In the strong-coupling neighbor-based optimization DMPC algorithm proposed in this paper, the optimal solution of each subsystem equals to the solution of optimizing the overall system objective function. That is:

$$\arg\min V_N(x_i, y_t, \infty; \bar{x}_i, \bar{u}_{i,0:N-1}, y_s) = \arg\min V_{iN}(x_i, y_t; \bar{x}_i, \bar{u}_{i,0:N-1}, y_s), \tag{A1}$$

where

$$V_N(x_i, y_t; \bar{x}_i, \bar{u}_{i,0:N-1}, y_s) = \sum_{k=0}^{N-1} (\|x(k) - \bar{x}_s\|_Q^2 + \|u(k) - \bar{u}_s\|_R^2) + \|x(N) - \bar{x}_s\|_P^2 + V_0(y_s, y_t). \tag{A2}$$

Thus, for easy analysis, here we take the overall objective function $V_N$ to prove the feasibility and stability, and define:

$$v^{[p]} = \{\bar{x}_1^{[p]}, ..., \bar{x}_m^{[p]}, \bar{u}_{1,0:N-1}^{[p]}, ..., \bar{u}_{m,0:N-1}^{[p]}, y_{1,s}^{[p]}, ..., y_{1,m}^{[p]}\}, \tag{A3}$$

$$v^{*} = \{\bar{x}_1^{*}, ..., \bar{x}_m^{*}, \bar{u}_{1,0:N-1}^{*}, ..., \bar{u}_{m,0:N-1}^{*}, y_{1,s}^{*}, ..., y_{1,m}^{*}\}. \tag{A4}$$

Denote $v_i'^{[p]}$ as

$$v_i'^{[p]} = (\bar{x}_1^{[p-1]}, ..., \bar{x}_i'^{[p]}, ..., \bar{x}_m^{[p-1]}, \bar{u}_{1,0:N}^{[p-1]}, ..., \bar{u}_{i,0:N}'^{[p]}, ..., \bar{u}_{m,0:N}^{[p-1]}, y_{1,s}^{[p-1]}, ..., y_{i,s}'^{[p]}, ..., y_{m,s}^{[p-1]}). \tag{A5}$$

**Lemma A1.** *Feasibility. Feasibility can be proved by assuming* $v(t) \in \mathcal{Z}_n$*. Then, we get* $v(t+1) \in \mathcal{Z}_n$*.*

**Proof.** The feasibility of the model is proved by analyzing the simplified model.

Refer to the warm start algorithm. At time $t+1$, the warm start for arbitrary $\mathcal{S}_i : \bar{v}_i^{[0]}(t+1) = (\bar{x}_i^{[0]}(0|t+1), \bar{u}_{i,0:N}^{[0]}(t+1), y_{i,s}^{[0]}(t+1))$. With the warm start, all constraints in (17) are satisfied.

From $p^{th}$ to $(p+1)^{th}$ iterations, given that

$$(v_1'^{[p+1]}, v_2'^{[p+1]}, ..., v_m'^{[p+1]})$$

are feasible, obviously their convex sum $v^{[p+1]}$ is feasible. As a result, the converged simplified solution $v^*(t+1)$ is feasible. Based on this, the real system solution is around the invariant set of $v^*(t+1)$, and fits the system constraints. That is, $(x(t+1), v(t+1)) \in \mathcal{Z}_n$. □

**Lemma A2.** *Convergence. Here we have*

$$V_N(x(t), y_t; v^{[p+1]}(t)) \leq V_N(x(t), y_t; v^{[p]}(t)).$$

**Proof.** Since it has

$$\begin{aligned} V_N(x, y_t, v^{[p+1]}) &\leq \gamma_1 V_N(x, y_t; v_1'^{[p+1]}) + ... + \gamma_m V_N(x, y_t; v_m'^{[p+1]}) \\ &\leq \gamma_1 V_N(x, y_t; v^{[p]}) + ... + \gamma_m V_N(x, y_t; v^{[p]}) \\ &= V_N(x, y_t, v^{[p]}), \end{aligned} \tag{A6}$$

convergence is proved. □

According to the Lemma above, we have:

$$V_N(x(t), y_t; v^{[p+1]}(t)) \leq V_N(x(t), y_t; v^{[0]}(t)). \tag{A7}$$

**Lemma A3.** *Local Bounded. When* $(x(t), y_s^{[0]}(t) \in \Omega_{\bar{K}}$*, then*

$$V_N(x(t), y_t; v^{[p]}(t)) \leq \left\| x(t) - \bar{x}_s^{[0]}(t) \right\|_P^2 + V_0(y_s^{[0]}(t), y_t).$$

**Proof.** Firstly,

$$V_N(x(t), y_t; v^{[0]}(t)) \leq \left\| x(t) - \bar{x}_s^{[0]}(t) \right\|_P^2 + V_0(y_s^{[0]}(t), y_t)$$

will be proved.

According to the definition of warm start and Assumption 3, here we have:

$$\begin{aligned}
& V_N(x(t),y_t;v^{[0]}(t)) \\
\leq & \sum_{k=0}^{N-1} \left\|x(k;t)-\bar{x}_s^{[0]}(t)\right\|_Q + \left\|u(k)-\bar{u}_s^{[0]}(t)\right\|_R + \left\|x(N)-\bar{x}_s^{[0]}(t)\right\|_P^2 + V_0(y_s^{[0]}(t),y_t) \\
= & \sum_{k=0}^{N-1} \left\|x(k;t)-\bar{x}_s^{[0]}(t)\right\|_{Q+\bar{K}'R\bar{K}} + \left\|x(N)-\bar{x}_s^{[0]}(t)\right\|_P^2 + V_0(y_s^{[0]}(t),y_t) \\
= & \left\|x(N)-\bar{x}_s^{[0]}(t)\right\|_P^2 + V_0(y_s^{[0]}(t),y_t).
\end{aligned} \tag{A8}$$

Thus, we have:

$$\begin{aligned}
V_N(x(t),y_t;v^{[p]}(t)) \leq & V_N(x(t),y_t;v^{[0]}(t)) \\
\leq & \left\|x(t)-\bar{x}_s^{[0]}(t)\right\|_P^2 + V_0(\bar{y}_s^{[0]}(t),y_t).
\end{aligned} \tag{A9}$$

□

**Lemma A4.** *Convergence. Let Assumption 3 hold, for any feasible solution $z(0)=(x(0),v(0)) \in \mathcal{Z}_N$, the system converges to equilibrium point $z_s$. That is,*

$$V_N(x(t+1),y_t;\bar{v}^*(t+1)) - V_N(x(t),y_t;\bar{v}^*(t)) \leq \|x(t)-\bar{x}_s(t)\|_Q^2. \tag{A10}$$

*The final tracking points of the simplified system (the optimal solution of $V_N$) are $(\bar{x}^*(x_s,y_t),\bar{u}^*(x_s,y_t)) = (x_s,u_s)$, which are the centralized optimal solution.*

**Proof.** For simplified system optimization, we have

$$V_N(x(t+1),y_t,v^*(t+1)) \leq V_N(x(t+1),y_t;v(0;t+1)), \tag{A11}$$

and also

$$V_N(x(t+1),y_t;v(0;t+1)) \leq V_N(x(t),y_t,v^*(t)) - \|x(t)-\bar{x}_s(t)\|_Q^2 - \|u(t)-\bar{u}_s(t)\|_R. \tag{A12}$$

According to (A11) and (A12), we have

$$\begin{aligned}
& V_N(x(t+1),y_t;v^*(t+1)) - V_N(x(t),y_t;v^*(t)) \\
\leq & -\|x(t)-\bar{x}_s(t)\|_Q^2 - \|u(t)-\bar{u}_s(t)\|_R \\
\leq & -\|x(t)-\bar{x}_s(t)\|_Q^2.
\end{aligned} \tag{A13}$$

Since the robust positively invariant set feedback control law $K = diag(K_1, K_2, ..., K_m)$ ensures the real states in the invariant set of the simplified model, the real system's stability is proved. □

## References

1. Christofides, P.D.; Scattolini, R.; Muñoz de la Peña, D.; Liu, J. Distributed model predictive control: A tutorial review and future research directions. *Comput. Chem. Eng.* **2013**, *51*, 21–41. [CrossRef]
2. Del Real, A.J.; Arce, A.; Bordons, C. Combined environmental and economic dispatch of smart grids using distributed model predictive control. *Int. J. Electr. Power Energy Syst.* **2014**, *54*, 65–76. [CrossRef]
3. Zheng, Y.; Li, S.; Tan, R. Distributed Model Predictive Control for On-Connected Microgrid Power Management. *IEEE Trans. Control Syst. Technol.* **2018**, *26*, 1028–1039. [CrossRef]

4. Yu, W.; Liu, D.; Huang, Y. Operation optimization based on the power supply and storage capacity of an active distribution network. *Energies* **2013**, *6*, 6423–6438. [CrossRef]
5. Scattolini, R. Architectures for distributed and hierarchical model predictive control-a review. *J. Process Control* **2009**, *19*, 723–731. [CrossRef]
6. Du, X.; Xi, Y.; Li, S. Distributed model predictive control for large-scale systems. In Proceedings of the 2001 American Control Conference, Arlington, VA, USA, 25–27 June 2001; Volume 4, pp. 3142–3143.
7. Li, S.; Yi, Z. *Distributed Model Predictive Control for Plant-Wide Systems*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
8. Mota, J.F.C.; Xavier, J.M.F.; Aguiar, P.M.Q.; Püschel, M. Distributed Optimization With Local Domains: Applications in MPC and Network Flows. *IEEE Trans. Autom. Control* **2015**, *60*, 2004–2009. [CrossRef]
9. Qin, S.; Badgwell, T. A survey of industrial model predictive control technology. *Control Eng. Pract.* **2003**, *11*, 733–764. [CrossRef]
10. Maciejowski, J. *Predictive Control: With Constraints*; Pearson Education: London, UK, 2002.
11. Vaccarini, M.; Longhi, S.; Katebi, M. Unconstrained networked decentralized model predictive control. *J. Process Control* **2009**, *19*, 328–339. [CrossRef]
12. Leirens, S.; Zamora, C.; Negenborn, R.; De Schutter, B. Coordination in urban water supply networks using distributed model predictive control. In Proceedings of the American Control Conference (ACC), Baltimore, MD, USA, 30 June–2 July 2010; pp. 3957–3962.
13. Wang, Z.; Ong, C.J. Distributed Model Predictive Control of linear discrete-time systems with local and global constraints. *Automatica* **2017**, *81*, 184–195. [CrossRef]
14. Trodden, P.A.; Maestre, J. Distributed predictive control with minimization of mutual disturbances. *Automatica* **2017**, *77*, 31–43. [CrossRef]
15. Al-Gherwi, W.; Budman, H.; Elkamel, A. A robust distributed model predictive control algorithm. *J. Process Control* **2011**, *21*, 1127–1137. [CrossRef]
16. Kirubakaran, V.; Radhakrishnan, T.; Sivakumaran, N. Distributed multiparametric model predictive control design for a quadruple tank process. *Measurement* **2014**, *47*, 841–854. [CrossRef]
17. Zhang, L.; Wang, J.; Li, C. Distributed model predictive control for polytopic uncertain systems subject to actuator saturation. *J. Process Control* **2013**, *23*, 1075–1089. [CrossRef]
18. Liu, J.; Muñoz de la Peña, D.; Christofides, P.D. Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements. *Automatica* **2010**, *46*, 52–61. [CrossRef]
19. Cheng, R.; Fraser Forbes, J.; Yip, W.S. Dantzig–Wolfe decomposition and plant-wide MPC coordination. *Comput. Chem. Eng.* **2008**, *32*, 1507–1522. [CrossRef]
20. Zheng, Y.; Li, S.; Qiu, H. Networked coordination-based distributed model predictive control for large-scale system. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 991–998. [CrossRef]
21. Groß, D.; Stursberg, O. A Cooperative Distributed MPC Algorithm With Event-Based Communication and Parallel Optimization. *IEEE Trans. Control Netw. Syst.* **2016**, *3*, 275–285. [CrossRef]
22. Walid Al-Gherwi, H.B.; Elkamel, A. Selection of control structure for distributed model predictive control in the presence of model errors. *J. Process Control* **2010**, *20*, 270–284. [CrossRef]
23. Camponogara, E.; Jia, D.; Krogh, B.; Talukdar, S. Distributed model predictive control. *IEEE Control Syst. Mag.* **2002**, *22*, 44–52.
24. Conte, C.; Jones, C.N.; Morari, M.; Zeilinger, M.N. Distributed synthesis and stability of cooperative distributed model predictive control for linear systems. *Automatica* **2016**, *69*, 117–125. [CrossRef]
25. Tippett, M.J.; Bao, J. Reconfigurable distributed model predictive control. *Chem. Eng. Sci.* **2015**, *136*, 2–19. [CrossRef]
26. Zheng, Y.; Wei, Y.; Li, S. Coupling Degree Clustering-Based Distributed Model Predictive Control Network Design. *IEEE Trans. Autom. Sci. Eng.* **2018**, 1–10. doi:10.1109/TASE.2017.2780444. [CrossRef]
27. Li, S.; Zhang, Y.; Zhu, Q. Nash-optimization enhanced distributed model predictive control applied to the Shell benchmark problem. *Inf. Sci.* **2005**, *170*, 329–349. [CrossRef]
28. Venkat, A.N.; Hiskens, I.A.; Rawlings, J.B.; Wright, S.J. Distributed MPC Strategies with Application to Power System Automatic Generation Control. *IEEE Trans. Control Syst. Technol.* **2008**, *16*, 1192–1206. [CrossRef]
29. Stewart, B.T.; Wright, S.J.; Rawlings, J.B. Cooperative distributed model predictive control for nonlinear systems. *J. Process Control* **2011**, *21*, 698–704. [CrossRef]

30. Zheng, Y.; Li, S. N-Step Impacted-Region Optimization based Distributed Model Predictive Control. *IFAC-PapersOnLine* **2015**, *48*, 831–836. [CrossRef]
31. de Lima, M.L.; Camponogara, E.; Marruedo, D.L.; de la Peña, D.M. Distributed Satisficing MPC. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 305–312. [CrossRef]
32. de Lima, M.L.; Limon, D.; de la Pena, D.M.; Camponogara, E. Distributed Satisficing MPC With Guarantee of Stability. *IEEE Trans. Autom. Control* **2016**, *61*, 532–537. [CrossRef]
33. Zheng, Y.; Li, S.; Wang, X. Distributed model predictive control for plant-wide hot-rolled strip laminar cooling process. *J. Process Control* **2009**, *19*, 1427–1437. [CrossRef]
34. Zheng, Y.; Li, S.; Li, N. Distributed model predictive control over network information exchange for large-scale systems. *Control Eng. Pract.* **2011**, *19*, 757–769. [CrossRef]
35. Li, S.; Zheng, Y.; Lin, Z. Impacted-Region Optimization for Distributed Model Predictive Control Systems with Constraints. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 1447–1460. [CrossRef]
36. Riverso, S.; Ferrari-Trecate, G. Tube-based distributed control of linear constrained systems. *Automatica* **2012**, *48*, 2860–2865. [CrossRef]
37. Riverso, S.; Boem, F.; Ferrari-Trecate, G.; Parisini, T. Plug-and-Play Fault Detection and Control-Reconfiguration for a Class of Nonlinear Large-Scale Constrained Systems. *IEEE Trans. Autom. Control* **2016**, *61*, 3963–3978. [CrossRef]
38. Ferramosca, A.; Limon, D.; Alvarado, I.; Camacho, E. Cooperative distributed MPC for tracking. *Automatica* **2013**, *49*, 906–914. [CrossRef]
39. Limon, D. MPC for tracking of piece-wise constant references for constrained linear systems. In Proceedings of the 16th IFAC World Congress, Prague, Czech Republic, 3–8 July 2005; p. 882.
40. Limon, D.; Alvarado, I.; Alamo, T.; Camacho, E.F. Robust tube-based MPC for tracking of constrained linear systems with additive disturbances. *J. Process Control* **2010**, *20*, 248–260. [CrossRef]
41. Shao, Q.M.; Cinar, A. Coordination scheme and target tracking for distributed model predictive control. *Chem. Eng. Sci.* **2015**, *136*, 20–26. [CrossRef]
42. Rawlings, J.B.; Mayne, D.Q. *Model Predictive Control: Theory and Design*; Nob Hill Publishing: Madison, WI, USA, 2009; pp. 3430–3433.
43. Alvarado, I. On the Design of Robust Tube-Based MPC for Tracking. In Proceedings of the 17th World Congress The International Federation of Automatic Control, Seoul, Korea, 6–11 July 2008; pp. 15333–15338.