

Article

Redundancy-Reduction-Based Hierarchical Design in Synchronization of Multi-Agent Systems

Haoyi Que ^{1,*} , Zhaowen Xu ¹  and Hongye Su ²¹ Institute of Intelligence Science and Engineering, Shenzhen Polytechnic, Shenzhen 518055, China² The National Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Yuquan Campus, Hangzhou 310027, China

* Correspondence: fatqcrab@szpt.edu.cn

Abstract: In this paper, a layered, undirected-network-structure, optimization approach is proposed to reduce the redundancy in multi-agent information synchronization and improve the computing rate. Based on the traversing binary tree and aperiodic sampling of the complex delayed networks theory, we proposed a network-partitioning method for pinning dynamic networks, with a more simplified, analyzable structure, and all of the traversed nodes are mathematically asymptotically synchronized at the same time. Moreover, a systematic implementable approach is proposed for application. The approach could be separated into two main steps: the first is a method of network partition that reduces the trivial interaction, which does not affect the information traversal, and the second involves applying the time-dependent functional approach of Lyapunov to give global exponential conditions, under the criteria for the synchronization of multiple agents, with a lower conservatism of the decision variables compared to some other results, so that the information available could fully benefit from the actual discrete-time communication pattern. Both mathematical proofs and numerical example evidence are presented to demonstrate the effectiveness of the implemented approach. This class contains a number of practically interesting systems, for instance, unmanned aerial vehicle (UAV) formation systems or the ground-air coordinated unmanned aerial system.

Keywords: multi-agent system; synchronization; stability; nonlinear systems



Citation: Que, H.; Xu, Z.; Su, H.

Redundancy-Reduction-Based Hierarchical Design in Synchronization of Multi-Agent Systems. *Appl. Sci.* **2023**, *13*, 2486. <https://doi.org/10.3390/app13042486>

Academic Editor: Dimitris Mourtzis

Received: 28 November 2022

Revised: 20 December 2022

Accepted: 21 December 2022

Published: 15 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The internal communications and the refraction of information between individuals constitute numerous multi-agent systems; thus, they are important attributes of the properties that constitutes numerous natural systems. When exploring the multi-agent systems, synchronization plays an indispensable role in the power-grid networks, system biology, climatology, sociology, and technology, and especially in the consensus problem of multi-agent systems, all of which have been widely studied topic and applied in plenty of scenarios. Dynamic and self-dynamic networks, which contain linear and nonlinear parts, make the process of synchronization time-varying. These attributes make it shine in some cases, and the excessive activation and synchronization of neural networks have been found to associate with a series of dysfunctions and disorders of the brain, as well as tumor and neural diagnoses such as Parkinson's disease, epileptic seizure, and Alzheimer's disease. In engineering, synchronization also has important applications in smart-grid networking [1], the industrial Internet [2,3], data fusion [4,5], and multi-agent systems [6–9]. So far, based on the computer network analysis and control theory, researchers have proposed a variety of models, algorithms, and approaches for solving the synchronization problem. The recent and specific problems are concentrated on the following directions: (1) The controllability and observability of multi-agent systems; (2) the robustness of synchronization in multi-agent systems; (3) disturbance and other restricted states; (4) the natural attributes of multi-agent systems; (5) the synchronization rate; and (6) the amount of calculation. Since we are focused

on the control-oriented investigations, and simultaneously, neural networks have faced their boom era, a huge amount of computational cost pressure makes the last two directions mentioned above especially important. The mathematical analysis of synchronization is strengthened for reducing conversation [10,11], optimizing calculation [12], and maximizing available resources [13,14] and limited application scenarios. With the deepening of research, it is becoming increasingly difficult to rely on mathematical algorithms to improve efficiency. In view of this, researchers gradually turned to other approaches.

By measuring the similarity between different data sources, and classified with multiple clusters, the pre-processing by cluster analysis can analyze various methods for different types of data, which greatly simplifies the data analysis process and is one of the most visible and explainable concepts in computer science. By applying the clustering analysis method to the large-scale complex dynamic network structure decomposition, the nodes with large correlations of properties are classified so that large-scale networks can be simplified and analyzed for a limited number of small-scale networks; thus, the network model could be simplified and the number of calculations could be reduced. The main structure of the network is also made clearer by the extraction of the main nodes. Louis M. Pecora et al. [15] created a framework of an analysis method for network dynamics to show the connection between cluster formation and network symmetries. This finding provides the fundamental contributions to the analysis of the clustering of the dynamic network. Then, some early related applications have emerged. In [16], based on eigenvalue decomposition, the authors proposed a centralized adaptive intermittent control approach to make cluster synchronization. By applying the adaptive approach, Tianping Chen et al. [17] proposed sufficient conditions to guarantee global cluster synchronization for linearly coupled networks, which could be applied in neural calculation to save computations. Oriented from the pinning control strategy, the authors [18] also found a new way to drive a general network to a selected cluster synchronization pattern. Considering an array of hybrid coupled neural networks with delay, based on Lyapunov stability theory and linear matrix inequality (LMI) method, Jinde Cao and Lulu Li [19] proposed several sufficient conditions for cluster synchronization. It should be pointed out that compared with existing work, the coupling configuration matrix and inner coupling matrix have greater applicability. Additionally, relative works could also be seen in [20–25]. It is worth pointing out that there is some good work being done in some emerging and intersecting areas, such as AI-driven packet forwarding [26] and the services of the customized industrial Internet of things [27].

Considering that the algorithms mentioned above have made great progress in mathematics and application, the algorithms may be more complex, and many solutions are only suitable for some specific scenarios, which has led to the development of a wide range of applications and better operability. Therefore, to the author's knowledge, there are few works on the implementation of the physical level of simplification that at the same time give the corresponding mathematical proof; this low-cost and widely used approach is also an applied method used to solve problems. In this paper, with the study of traversing binary tree theory, and the technology of cluster analysis, we proposed a network-partitioning method for pinning dynamic networks. In particular, a relative theory is proposed to guarantee all traversed nodes asymptotically synchronize to the pinning node. We have standardized this method and given one kind of specific implementation step; therefore, the main structure of the network is also highlighted. Additionally, the requirement of the application is only related to the structure of the natural structural properties of the networks and can be freely adjusted according to the actual use. The method focuses on reducing the amount of calculation, which could be applied as a pre-treatment process before most of networked synchronization algorithms. Simulations with the same algorithm show the effectiveness of the method.

The other parts are organized as follows. Section 2 shows some preliminary knowledge of our model and introduces a description of the network partitioning problem. In Section 3,

we proposed a theorem that proves that all of the layers of the nodes in the sub-networks could synchronize to the central node. Section 4 involves numerical simulation.

2. Preliminary Knowledge

Consider an undirected graph $\mathcal{G}(\mathcal{N}, \mathcal{E}, \mathcal{A})$, where $\mathcal{N} = \{n_1, n_2, \dots, n_N\}$ represents N nodes in the graph, $\mathcal{A} = [e_{ij}]_{N \times N}$ is the adjacency matrix, and e_{ij} represents an undirected edge in $\mathcal{G}(\mathcal{N}, \mathcal{E}, \mathcal{A})$ of the ordered pair of nodes n_i and n_j . $\mathcal{E}$ is a set of undirected edges and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$. Without orientation, the edge of node (n_i, n_j) is identical to the edge (n_j, n_i) , and the information from different nodes n_i and n_j share the same weight. Diagonal element e_{ii} is defined as $e_{ii} = -\sum_{j=1, j \neq i}^N e_{ij}$. It is easy to surmise that $\sum_{j=1}^N e_{ij} = 0$, $\sum_{i=1}^N e_{ij} = 0$.

We consider a common CDN with N identical nodes of the form

$$\begin{aligned} \dot{x}_i(t) &= Ax_i(t) + \bar{f}(x_i(t), t) + c \sum_{j=1}^N e_{ij} \Gamma x_j(t_k) \\ t &\in [t_k, t_{k+1}), i = 1, 2, \dots, N \end{aligned} \quad (1)$$

where we have Γ is an inner-coupling matrix of appropriate dimensions, $t_0 = 0$ as initial instant. t_k are the communication instants and satisfy

$$\begin{aligned} 0 = t_0 < t_1 < \dots < t_k < \dots < \lim_{k \rightarrow +\infty} t_k = +\infty \\ k &= 0, 1, 2, \dots \end{aligned} \quad (2)$$

$x_i(t) = [x_{i1}(t), x_{i2}(t), \dots, x_{in}(t)]^T \in \mathbb{R}^n$ is the state vector of node i . d is a constant of time delay. When $t < 0$, $x_{ij}(t-d) = 0$ holds at the initial condition.

To satisfy the synchronization requirement, all nodes must track the central node. In this paper, node 1 is always chosen as the central node.

Let

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_N(t) \end{bmatrix} \in \mathbb{R}^{Nn}, g(t) = \begin{bmatrix} \bar{f}(x_1(t), t) \\ \bar{f}(x_2(t), t) \\ \vdots \\ \bar{f}(x_N(t), t) \end{bmatrix} \in \mathbb{R}^{Nn}$$

Therefore, we could rewrite Equation (1) as follows:

$$\dot{x}(t) = (I_N \otimes A)x(t) + g(t) + c(\mathcal{E} \otimes \Gamma)x(t_k), t \in [t_k, t_{k+1}) \quad (3)$$

Let

$$y(t) = \begin{bmatrix} x_1(t) - x_2(t) \\ x_1(t) - x_3(t) \\ \vdots \\ x_1(t) - x_N(t) \end{bmatrix} \quad (4)$$

then the synchronization process could be described as $\lim_{t \rightarrow \infty} y(t) = 0$. To achieve the tracking goal, following assumption and lemma would be used.

Assumption 1 ([28]). A diagonal matrix $\Lambda > 0$ exists, and then the nonlinear part of the node system satisfies the following condition:

$$(\bar{g}(x_1, t) - \bar{g}(x_2, t))^T (\bar{g}(x_1, t) - \bar{g}(x_2, t)) \leq (x_1 - x_2)^T \Lambda (x_1 - x_2) \quad (5)$$

Lemma 1 ([28]). The following equation holds if matrix $G \in \mathbb{R}^{N \times N}$ satisfies the same sum for each row.

$$\tilde{M}G = \tilde{M}G\Phi\tilde{M} \quad (6)$$

where

$$\tilde{M} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 1 & 0 & -1 & 0 & \cdots & 0 \\ \vdots & & & \ddots & & \vdots \\ 1 & 0 & \cdots & 0 & -1 & 0 \\ 1 & 0 & \cdots & & 0 & -1 \end{bmatrix} \in \mathbb{R}^{(N-1) \times N}$$

$$\Phi = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ -1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & \vdots \\ 0 & 0 & \ddots & \ddots & 0 \\ \vdots & & \ddots & -1 & 0 \\ 0 & \cdots & 0 & 0 & -1 \end{bmatrix} \in \mathbb{R}^{N \times (N-1)}$$

Equation (3) could be represented as

$$\begin{aligned} \dot{y}(t) &= (I_{N-1} \otimes A)y(t) + g(t) \\ &+ cM(\mathcal{E}\Phi \otimes \Gamma)y(t_k), t \in [t_k, t_{k+1}) \end{aligned} \quad (7)$$

where $M = \tilde{M} \otimes I_n$.

In practical neural networks, usually N would be a huge number; therefore, there would be a tremendous amount of $\mathcal{E}$. The computation burden would be enormous, although mathematical work could alleviate the problem, but the ceiling may have been encountered. Therefore, some structural approaches may be applied that may be more effective than extensive algorithms.

In our model, the structure of network is known and the central node is confirmed. For a certain single node in the networks, it is obvious that the connected edge with greater weight plays a more influential role than the smaller one. Therefore, by evaluating the influence of different connection weights of individual nodes without breaking the integrity of network structure, we could establish a rule that focuses on prior connections. The weighted edges with a strong influence could be selected and retained, and the edges with smaller weights that have trivial influence on the global synchronization target could be ignored. The structure will further save on computing resources. Based on that, the CDN with a tremendous number of nodes and connections could be partitioned by a finite number of sub-networks.

Remark 1. In the undirected graph, unlike the directed graph, usually there are no necessary pinning control networks. Therefore, throughout our paper, the central node is free to choose, and the setting of node 1 is just for convenience.

A simple example of CDN with 8 nodes is offered in Figure 1. The weight allocation matrix is

$$\mathcal{E} = \begin{bmatrix} -68 & 12 & 8 & 41 & 0 & 0 & 0 & 7 \\ 12 & -140 & 0 & 0 & 125 & 3 & 0 & 0 \\ 8 & 0 & -69 & 61 & 0 & 0 & 0 & 0 \\ 41 & 0 & 61 & -214 & 0 & 112 & 0 & 0 \\ 0 & 125 & 0 & 0 & -149 & 24 & 0 & 0 \\ 0 & 3 & 0 & 112 & 24 & -145 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & -6 & 0 \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & -7 \end{bmatrix} \quad (8)$$

In Figure 1, we can observe information transfers between node 1 and node 2, 3 and 8. Node 3 exchanges information with node 1 and 4 at the same time. For node 3, the coupling

weight with node 4 is much bigger than with node 1, such that the impact from node 4 is even greater than node 1 to 3. Therefore, without affecting the complete transmission of topology, we could try to ignore the connection between node 1 and 3, to avoid useless calculations. By neglecting some trivial connections, we can realize the partitioning of the network to a finite number of sub-networks layer after layer, such that the computation burden could be relieved and the effectiveness could be boosted.

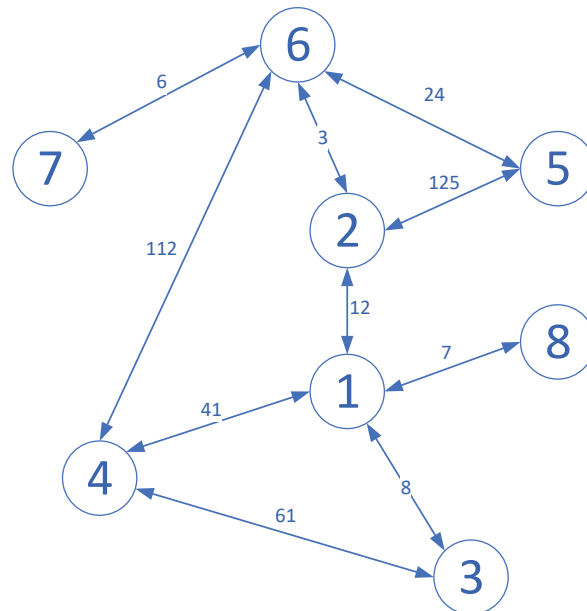


Figure 1. An example of CDN with 8 nodes.

In Figure 1, nodes 7 and 8 are the isolated nodes that only have one edge connecting to other nodes. Since node 1 is the central node, node 8 could be considered as a sub-network. As mentioned above, the connection weight between node 1 and node 3 is small. It does not affect the traversal of the whole network. Therefore, we could separate nodes 3, 4, 6, and 7 as another sub-network. Additionally, nodes 2 and 5 constitute the rest of the sub-network.

Consequently, the network of Figure 1 could be transferred into the following Figure 2, where the example with the form of system (7) could be rewritten as 3 parts:

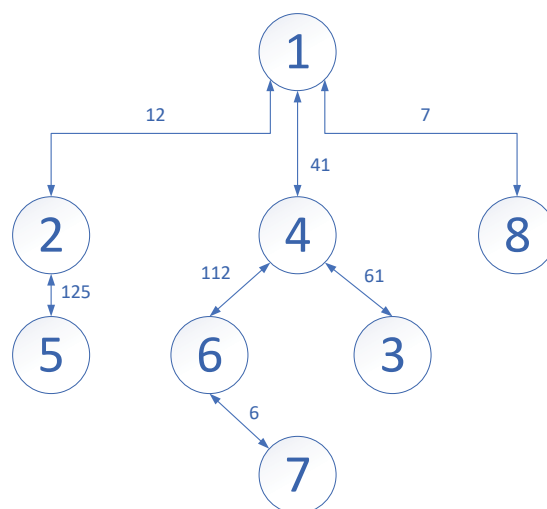


Figure 2. Structures of sub-networks in Figure 1.

The algorithm in [13] is applied here as reference. The total computation of scalar decision variables is $7.5m^2 + 1.5m + N - 1$, where $m = (N - 1)n$, $N = 8$ in the example. Therefore, without partitioning the CDN, in Example 1, the number is $367.5n^2 + 10.5n + 7$, while after partitioning the CDN, the number is $105n^2 + 9n + 6$, which is much less than the method without partitioning the CDN.

Moreover, the advantages of partitioning will become more obvious as N increases. Systematic process algorithms can be derived from the previous example.

3. Synchronization of Complex Dynamic Networks

Before beginning our systematic process, the integrity of information transmission should be ensured.

Throughout this paper, the synchronization algorithm is based on the following Lemma [14].

Lemma 2. For a given scalar h , matrices $P > 0$, $\begin{bmatrix} U_1 & U_2 \\ * & U_3 \end{bmatrix} > 0$, $X, X_1, H = [H_1 \ H_2 \ H_3]$ exist, and a diagonal matrix $\Delta > 0$; under Assumption 1, we have

$$\begin{bmatrix} P + h\frac{X + X^T}{2} + U_1 & -hX + hX_1 - U_1 & U_2 \\ * & \Pi & -U_2 \\ * & * & U_3 \end{bmatrix} > 0 \quad (9)$$

$$\Psi_1 = \begin{bmatrix} \Psi_{11} & \Psi_{12} & \Psi_{13} \\ * & \Psi_{22} & -H_2^T \\ * & * & \Psi_{33} \end{bmatrix} < 0 \quad (10)$$

$$\Psi_2 = \begin{bmatrix} \Psi_{11} & \Psi_{12} & \Psi_{13} & hH_1^T \\ * & \Psi_{22} & -H_2^T & hH_2^T \\ * & * & \Psi_{33} - hU_3 & hH_3^T \\ * & * & * & -hU_1 \end{bmatrix} < 0 \quad (11)$$

$$\Psi_3 = \begin{bmatrix} \Psi_{11} + Y_{11} & \Psi_{12} + Y_{12} & \Psi_{13} + Y_{13} \\ * & \Psi_{22} + hU_1 & -H_2^T + Y_{23} \\ * & * & \Psi_{33} + Y_{33} \end{bmatrix} < 0 \quad (12)$$

where

$$\begin{aligned} \Psi_{11} &= P(I_{N-1} \otimes A) + (I_{N-1} \otimes A)^T P - \frac{X + X^T}{2} \\ &\quad + (\Delta \otimes I_n)(I_{N-1} \otimes \Lambda) + (I_{N-1} \otimes \Lambda)(\Delta \otimes I_n) \\ &\quad + H_1 + H_1^T \\ \Psi_{12} &= P + H_2 \\ \Psi_{13} &= cPM(G\Phi \otimes \Gamma) - U_2 + X - X_1 + H_3 - H_1^T \\ \Psi_{22} &= -2(\Delta \otimes I_n) \\ \Psi_{33} &= U_2 + U_2^T + X_1 + X_1^T - \frac{X + X^T}{2} - H_3 - H_3^T \\ Y_{11} &= h\frac{X + X^T}{2}(I_{N-1} \otimes A) + h(I_{N-1} \otimes A)^T \frac{X + X^T}{2} \\ &\quad + h(I_{N-1} \otimes A)^T U_1 (I_{N-1} \otimes A) \end{aligned}$$

$$\begin{aligned}
Y_{12} &= h \frac{X + X^T}{2} + h(I_{N-1} \otimes A)^T U_1 \\
Y_{13} &= h(I_{N-1} \otimes A)^T U_2 + ch \frac{X + X^T}{2} M(G\Phi \otimes \Gamma) \\
&\quad + h(I_{N-1} \otimes A)^T (-X + X_1) \\
&\quad + ch(I_{N-1} \otimes A)^T U_1 M(G\Phi \otimes \Gamma) \\
Y_{23} &= hU_2 - hX + hX_1 + chU_1 M(G\Phi \otimes \Gamma) \\
Y_{33} &= hU_3 + chU_2^T M(G\Phi \otimes \Gamma) + ch(G\Phi \otimes \Gamma)^T M^T U_2 \\
&\quad + ch(-X^T + X_1^T) M(G\Phi \otimes \Gamma) \\
&\quad + ch(G\Phi \otimes \Gamma)^T M^T (-X + X_1) \\
&\quad + c^2 h(G\Phi \otimes \Gamma)^T M^T U_1 M(G\Phi \otimes \Gamma) \\
\Pi &= -hX_1 - hX_1^T + h \frac{X + X^T}{2} + U_1 \\
M &= \tilde{M} \otimes I_n
\end{aligned}$$

Then, CDN (2) is globally exponentially synchronized.

The proof process would be omitted here due to the limited space. The reader can see the whole proof process from the reference [14].

We partition the network into finite chains, of which no cycle is included in the sub-networks. Then, the process of CDN synchronization could be reduced to the finite chains and the central node. With Lemma 2, the following theorem is proposed.

Theorem 1. Consider a class of multi-agent systems with a known structural connection weight. Partition the CDN into multi-layers of sub-networks with according sub-central nodes and stay full-connected. Assuming that each layer of the sub-network asymptotically synchronizes to the corresponding sub-central node; then, the whole CDN achieves asymptotical synchronization.

Proof. Partition the CDN with $i = 1, 2, 3, \dots, n_l$ layers. Assuming that there would be s_i sub-networks in the i th layer, therefore there would be s_i sub-central nodes in the i th layer. s_i^j denotes the j th single sub-central node. By using the prior definition, the first sub-central node is named by the 1st node, such that all nodes in the CDN could be described as $x_{i,s_i^j,k}$. With that, we have that $x_{1,1,1}$ is the central node for all of the multi-agent systems. The sub-central node $x_{i,s_i^j,1}$ represents the j th sub-central node in the i th layer.

$t_0 = 0$ is the initial time. The aim is to prove that if $\delta > 0$, for any $\|x_{i,s_i^j,1}(t_0) - x_{i,s_i^j,k}(t_0)\| < \delta$ exists, and $\lim_{t \rightarrow \infty} \|x_{i,s_i^j,1}(t) - x_{i,s_i^j,k}(t)\| = 0$. Then, the $\lim_{t \rightarrow \infty} \|x_{1,1,1}(t) - x_{i,s_i^j,k}(t)\| = 0$ also set up.

The proof could be achieved by the iterative method. Firstly, with the last layer, sub-networks (n_l th layer) asymptotically synchronize to the sub-central node $x_{n_l,s_{n_l}^j,k}$; then, the corresponding Lyapunov functional could exist and be represented as $V_{n_l,s_{n_l}^j}(t)$, and we have

$$\dot{V}_{n_l,s_{n_l}^j}(t) < 0 \quad (13)$$

for all $s_{n_l}^j$.

Then, for the one up layer, a similar Lyapunov functional could be established with $V_{n_l-1,s_{n_l-1}^j}(t)$ to prove that for the $n_l - 1$ layer, the sub-network could achieve asymptotic synchronization.

We combine the $V_{n_l, s_{n_l}^j}(t)$ and $V_{n_l-1, s_{n_l-1}^j}(t)$ to be a new Lyapunov functional $V_{n_l, s_{n_l}^j}(t) + V_{n_l-1, s_{n_l-1}^j}(t)$, such that

$$\dot{V}_{n_l, s_{n_l}^j}(t) + \dot{V}_{n_l-1, s_{n_l-1}^j}(t) < 0 \quad (14)$$

By iterating this method over the entire network, the Lyapunov functionals could always exist

$$V(t) = \sum_{i=1}^{n_l} V_{i, s_i^j}(t) \quad (15)$$

and

$$\dot{V}(t) = \sum_{i=1}^{n_l} \dot{V}_{i, s_i^j}(t) < 0 \quad (16)$$

Such that there exist $\delta > 0$ to satisfy

$$\begin{aligned} & \|x_{1,1,1}(t_0) - x_{i, s_i^j, k}(t_0)\| < \delta \\ \Rightarrow \lim_{t \rightarrow \infty} \|x_{1,1,1}(t) - x_{i, s_i^j, k}(t)\| &= 0 \end{aligned} \quad (17)$$

□

Remark 2. In the proof, in Theorem 1, no concrete Lyapunov functional is needed to complete the proof; hence, it can be ignored.

Remark 3. Note that in the definition of $x_{i_1, s_{i_1}^j, k}$, $x_{i_2, s_{i_2}^j, k}$ could be the same node because nodes in the i_1 th layer could be the sub-central node in the i_2 th layer. The representation method does not affect the proof process.

One of the available partitioning network method is proposed with following steps:

- *Step 1:* Lists all of the nodes connected to the central node.
- *Step 2:* Find nodes with only one connection and list them out; name these nodes “initial points”.
- *Step 3:* Start the chain with the initial nodes. Find the node connected to the initial points, then continue the chain with the connected node to find the next step. The strategy is to connect the greater coupling weight. If the initial node only communicates with the central node, stop the search and mark the connection node as a sub-network.
- *Step 4:* If the two chains meet at one node (not the central node), then two chains could be combined to be one chain and continue the path.
- *Step 5:* When all of the chains are connected to the central node, mark all the chains and continue dealing with the rest of the nodes.
- *Step 6:* Start the path from each node of the marked chain to build the sub-chain; like *Step 1*, the sub-central node could be set here if needed. Use the greater coupling weight strategy to connect all the other nodes.
- *Step 7:* If two sub-chains connect to one node, the connection with stronger coupling weight could be kept as one node of the subchain. The weaker coupling weight connected node ignores this connection and searches for the second great coupling weight node.
- *Step 8:* Repeat *Step 6* until all nodes are traversed.
- *Step 9:* After traversing all nodes in the network, the search could stop and the sub-networks could be built.

The algorithm is only a simple solution; it is not optimal. Strategies such as dynamical programming, prim's algorithm, and others could be adopted here. Additionally, event-triggered samplers could be applied to offer a threshold. The effect of reducing the computation amount is more obvious when strategies consider more layers and a shorter chain; meanwhile, the efficiency of synchronization will be reduced by more layers. Moreover, coupling weight should also be taken into account because that will appreciably impact the synchronization rate. Therefore, the balance should be taken into account of equity simultaneously.

Remark 4. After the network-partitioning process, assuming that the new net has p_i and by re-arranging the weight allocation matrix, the following form would be shown:

$$\mathcal{E}_r = \begin{Bmatrix} a_{11} & P_1 & \cdots & P_{s_2} \\ P_1^T & P_{1,1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ P_{s_2}^T & \cdots & \cdots & P_{s_2,s_2} \end{Bmatrix} \quad (18)$$

where $P_1, \dots, P_{s_2}$ are the matrix block, and $P_{i,i}$ is the folded portion of the i th matrix block and is a symmetric matrix. With the re-arranged weight-allocation form, the connection between different nodes could be clarified.

In our results, all of the sub-networks could be grouped into finite sub-networks that are only connected to the central node. Proofs could be offered to show where the calculation savings are coming from. This is not merely an oversight of some unnecessary coupling weights.

Before the Proof, the following assumption should be offered to make the network-partitioning process clearer.

Assumption 2. Each node in the complex networks only has one route connecting to the central node after the process of network partitioning.

Corollary 1. Considering the fully connected complex dynamic network, with the projection of network partitioning, under the condition of Assumption 1, the number of decision variables could be reduced such that the calculation could be saved.

Proof. No matter what synchronization criterion we choose, the number of decision variables will always have following form:

$$\sum_{i=1}^M c_i (N-1)^i \quad (19)$$

where c_i is an integer coefficient greater than or equal to zero; also, n for the dimension of node is considered in the c_i . M is the max order. N is the number of nodes. For all of the partitioned network, the first layer only has the central node and is represented as $s_1 = 1$. In the $2, 3, \dots, l$ layer, the nodes are $s_2, s_3, \dots, s_l$, which are all sub-central nodes.

Let us start from the partitioned networks with layer 2, which means that all of the nodes are connected to the central node; therefore, we have $s_2 + 1 = N$. Additionally, the number of decision variables could be

$$\sum_{i=1}^M c_i (s_1 + s_2 - 1)^i \quad (20)$$

which is the same as the unpartitioned networks (19). Let us extend the situation into the n_l layer; for the i th layer, the s_i sub-central node has s_i sub-networks, and there are N_{m,s_i}

nodes in each sub-network accordingly. m is the serial number from $1, \dots, s_i$. Therefore, in the $l - 1$ layer, the number of decision variables we need to calculate is

$$\begin{aligned} & \sum_{i=1}^M \sum_{j=1}^{s_{l-1}} c_i (N_{j,s_{l-1}})^i \\ &= \sum_{i=1}^M c_i (N_{1,s_{l-1}})^i + \dots + c_i (N_{s_{l-1},s_{l-1}})^i \\ &\leq \sum_{i=1}^M c_i \left(\sum_{j=1}^{s_{l-1}} (N_{j,s_{l-1}}) \right)^i \end{aligned} \quad (21)$$

where $(a^i + b^i) \leq (a + b)^i$ is implemented here.

Furthermore, in each layer, the above proposition could be achieved. As the number of layers increases, the number of decision variables we need to deal with decreases at an accelerating rate.

For the central node, N_{m,s_1} represents sub-networks connected to the central node (even if only one point is included) from $m = 1$ to $m = s_1$. Additionally, the number of decision variables we have after partitioning the network is as follows:

$$\begin{aligned} & \sum_{i=1}^M \sum_{j=1}^{s_1} c_i (N_{1,s_1} - 1)^i + \sum_{i=1}^M c_i (s_1 + 1 - 1)^i \\ &= \sum_{i=1}^M c_i \left((N_{1,1} - 1)^i + (N_{1,2} - 1)^i + \dots + (N_{1,s_1} - 1)^i \right) \\ &+ \sum_{i=1}^M c_i (s_1 + 1 - 1)^i \\ &\leq \sum_{i=1}^M c_i (N - 1)^i \end{aligned} \quad (22)$$

Such that the proof is complete. $\square$

4. Numerical Examples

The numerical example is chosen as Figure 1. With the projection we proposed, the connecting topology figure is depicted in Figure 2, and the weight allocation matrix is

$$\mathcal{E}_p = \begin{bmatrix} -60 & 12 & 0 & 41 & 0 & 0 & 0 & 7 \\ 12 & -137 & 0 & 0 & 125 & 3 & 0 & 0 \\ 0 & 0 & -61 & 61 & 0 & 0 & 0 & 0 \\ 41 & 0 & 61 & -214 & 0 & 112 & 0 & 0 \\ 0 & 125 & 0 & 0 & -125 & 0 & 0 & 0 \\ 0 & 0 & 0 & 112 & 0 & -118 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & -6 & 0 \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & -7 \end{bmatrix}$$

Select $g(t) = (\|x_1(t) + 1\| - \|x_1(t) - 1\|)$ as the nonlinear part. The inner coupling matrix Γ is I_N in the example, and various coupling strengths are selected as $c = 1.53$.

At the same time, a synchronization error is represented as

$$e(t) = \|y(t)\|$$

The trajectories of the state and error shown in Figure 3 adopted the network-partitioning method. As a comparison, the results are shown in Figure 4. It can be seen from the results

that the synchronization errors of the two methods converge exponentially. However, the speed of this method is greatly improved.

Under the same synchronization criterion, all of the trajectories in Figures 3 and 4 converge exponentially, but the tracking speed using the network-partitioning method is significant faster than the one without.

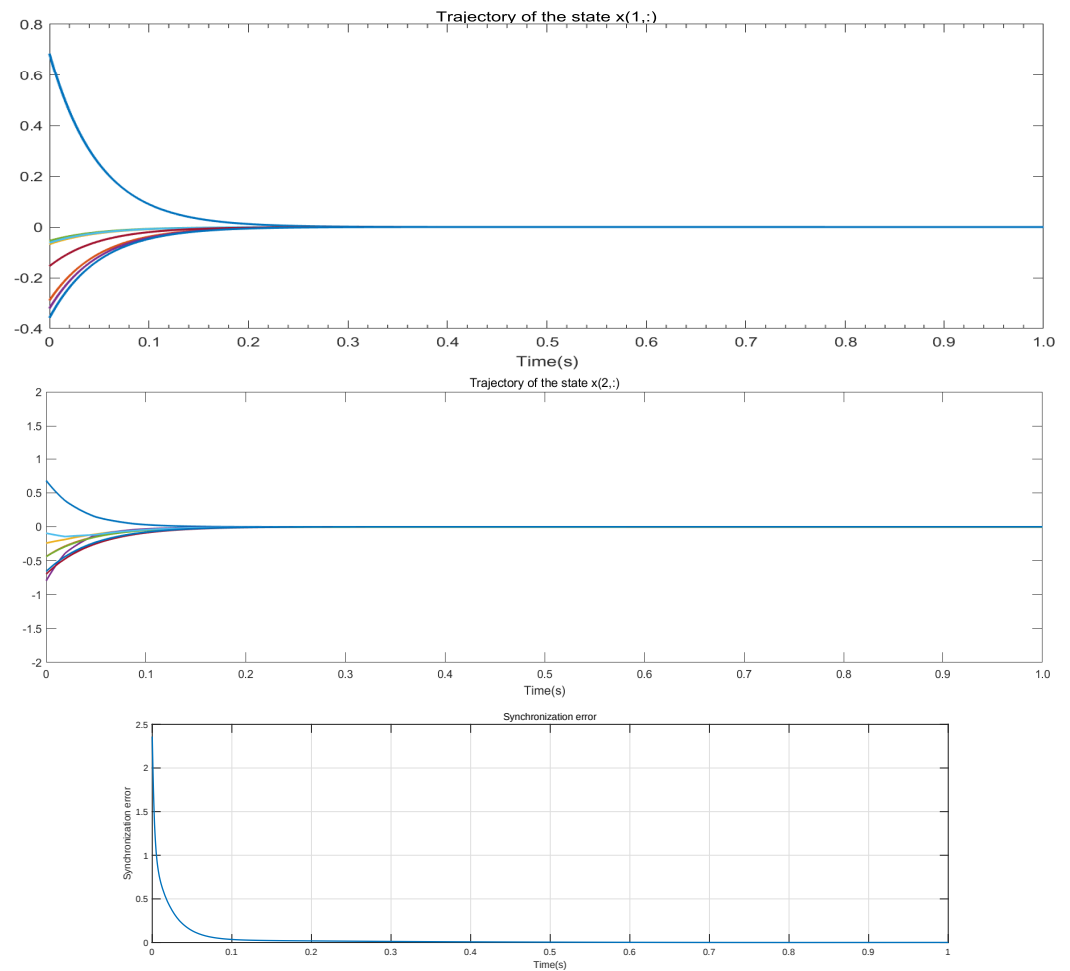


Figure 3. Trajectory using network-partitioning method.

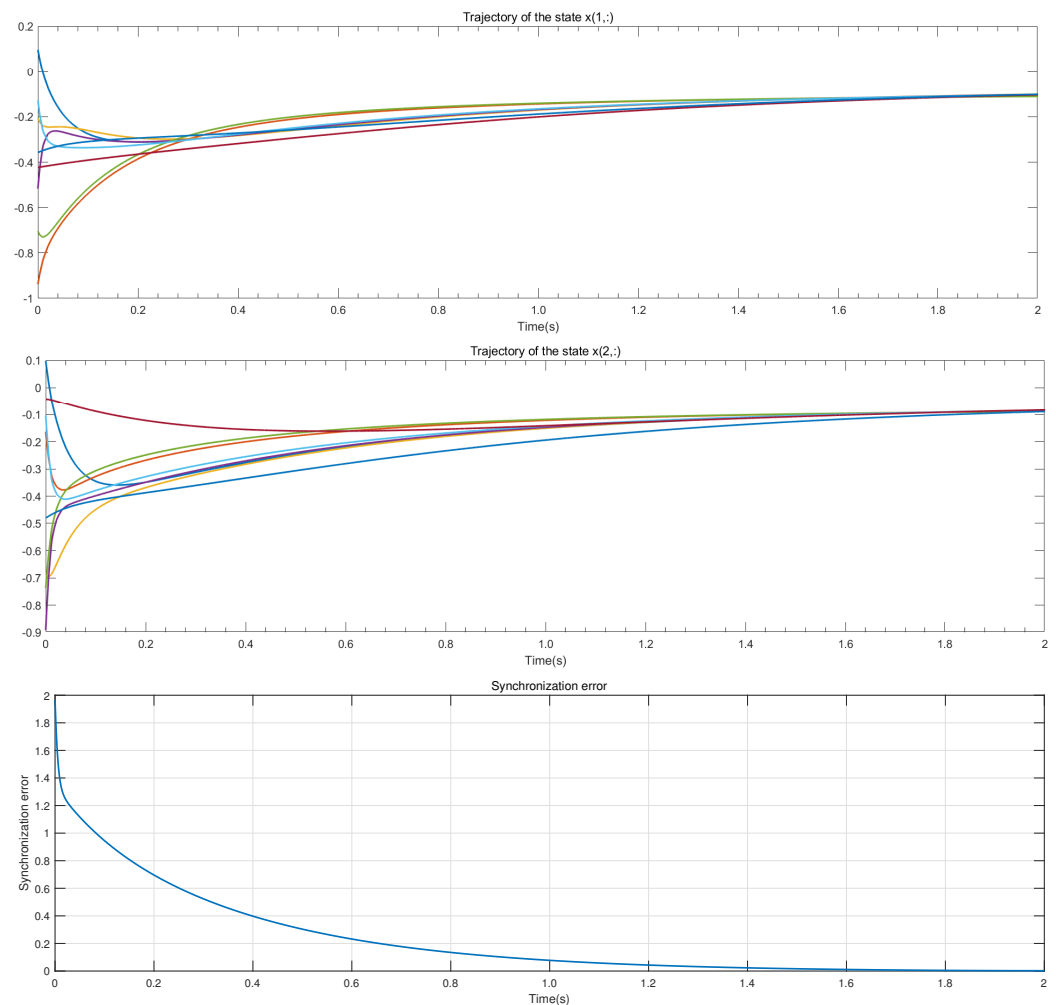


Figure 4. The comparison results of the trajectory graph of synchronization methods in [14].

5. Conclusions

In this paper, we present a general network-partitioning method for the synchronization of multi-agent systems. While using global exponential synchronization criteria conditions, we decompose the computational burden by slightly trivial connection weights that do not affect the global synchronization target. Under the projection, some unnecessary edges and information are ignored in the networks, such that the amount of calculation is reduced; the efficiency of synchronization has also been improved, and the main structure of the network has also been highlighted and decomposed into multiple sub-network chains, offering convenient conditions for further analysis. Moreover, after applying this method, the information transfer and synchronization characteristics between nodes are not affected. Both a mathematical proof and numerical examples demonstrate the effectiveness of the method. Future results will be achieved in the form of coordinated missions in swarms of UAVs.

Author Contributions: Conceptualization, H.Q.; methodology, H.Q.; software, Z.X.; validation, H.Q. and Z.X.; formal analysis, H.Q.; investigation, H.Q.; data curation, H.S.; writing—original draft preparation, H.Q.; writing—review and editing, H.Q. and Z.X.; supervision, H.S.; project administration, H.S.; and funding acquisition, H.Q., Z.X. and H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (61903260, 62003219) and the Stable Supporting Programme for the Universities of Shenzhen (GXWD20201231165806004, 20201019173610001).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to express their sincere gratitude to Zhengguang Wu and Hong Lin for helping to finish this paper.

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

References

1. Almas, M.S.; Vanfretti, L.; Singh, R.S.; Jonsdottir, G.M. Vulnerability of synchrophasor-based WAMPAC applications' to time synchronization spoofing. *IEEE Trans. Smart Grid* **2017**, *9*, 4601–4612. [\[CrossRef\]](#)
2. Qiu, T.; Zhang, Y.; Qiao, D.; Zhang, X.; Wymore, M.L.; Sangaiah, A.K. A robust time synchronization scheme for industrial internet of things. *IEEE Trans. Ind. Inf.* **2017**, *14*, 3570–3580. [\[CrossRef\]](#)
3. Tan, A.; Peng, Y.; Su, X.; Tong, H.; Deng, Q. A novel synchronization scheme based on a dynamic superframe for an industrial internet of things in underground mining. *Sensors* **2019**, *19*, 504. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Chou, C.A.; Jin, X.; Mueller, A.; Ostadabbas, S. Multimodal Data Fusion-Moving From Domain-Specific Algorithms to Transdomain Understanding for Accelerated Solution Development. *IEEE Sens. Lett.* **2019**, *3*, 7100104. [\[CrossRef\]](#)
5. Zhang, H.; Feng, B.; Tian, A. A systematic review for smart identifier networking. *Sci. China Inf. Sci.* **2022**, *65*, 100584. [\[CrossRef\]](#)
6. Wen, S.; Liu, W.; Yang, Y.; Huang, T.; Zeng, Z. Generating realistic videos from keyframes with concatenated GANs. *IEEE Trans. Circ. Syst. Video Technol.* **2018**, *29*, 2337–2348. [\[CrossRef\]](#)
7. Wen, S.; Hu, R.; Yang, Y.; Huang, T.; Zeng, Z.; Song, Y.D. Memristor-based echo state network with online least mean square. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *49*, 1787–1796. [\[CrossRef\]](#)
8. Wen, S.; Xiao, S.; Yang, Y.; Yan, Z.; Zeng, Z.; Huang, T. Adjusting learning rate of memristor-based multilayer neural networks via fuzzy method. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **2018**, *38*, 1084–1094. [\[CrossRef\]](#)
9. Gong, X.; Cui, Y.; Shen, J.; Feng, Z.; Huang, T. Necessary and sufficient conditions of formation-containment control of high-order multiagent systems with observer-type protocols. *IEEE Trans. Cybern.* **2022**, *52*, 7002–7016. [\[CrossRef\]](#)
10. Zhang, C.K.; He, Y.; Jiang, L.; Wang, Q.G.; Wu, M. Stability analysis of discrete-time neural networks with time-varying delay via an extended reciprocally convex matrix inequality. *IEEE Trans. Cybern.* **2017**, *47*, 3040–3049. [\[CrossRef\]](#)
11. Zhang, C.K.; He, Y.; Jiang, L.; Wu, M.; Wang, Q.G. An extended reciprocally convex matrix inequality for stability analysis of systems with time-varying delay. *Automatica* **2017**, *85*, 481–485. [\[CrossRef\]](#)
12. Zhang, R.; Zeng, D.; Liu, X.; Zhong, S.; Shi, K. A new method for quantized sampled-data synchronization of delayed chaotic Lur'e systems. *Appl. Math. Model.* **2019**, *70*, 471–489. [\[CrossRef\]](#)
13. Que, H.; Wu, Z.G.; Su, H. Globally exponential synchronization for dynamical networks with discrete-time communications. *J. Frank. Inst.* **2017**, *354*, 7871–7884. [\[CrossRef\]](#)
14. Que, H.; Fang, M.; Wu, Z.G.; Su, H.; Huang, T.; Zhang, D. Exponential synchronization via aperiodic sampling of complex delayed networks. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1399–1407. [\[CrossRef\]](#)
15. Pecora, L.M.; Sorrentino, F.; Hagerstrom, A.M.; Murphy, T.E.; Roy, R. Cluster synchronization and isolated desynchronization in complex networks with symmetries. *Nat. Commun.* **2014**, *5*, 4079. [\[CrossRef\]](#)
16. Allefeld, C.; Müller, M.; Kurths, J. Eigenvalue decomposition as a generalized synchronization cluster analysis. *Int. J. Bifurc. Chaos* **2007**, *17*, 3493–3497. [\[CrossRef\]](#)
17. Liu, X.; Chen, T. Cluster synchronization in directed networks via intermittent pinning control. *IEEE Trans. Neural Netw.* **2011**, *22*, 1009–1020.
18. Wu, W.; Zhou, W.; Chen, T. Cluster synchronization of linearly coupled complex networks under pinning control. *IEEE Trans. Circ. Syst. I: Regul. Pap.* **2008**, *56*, 829–839. [\[CrossRef\]](#)
19. Wang, Z.; Jiang, G.; Yu, W.; He, W.; Cao, J.; Xiao, M. Synchronization of coupled heterogeneous complex networks. *Journal of the Franklin Institute. J. Frankl. Inst.* **2017**, *354*, 4102–4125. [\[CrossRef\]](#)
20. Liu, X.; Ho, D.W.C.; Song, Q.; Xu, W. Finite/fixed-time pinning synchronization of complex networks with stochastic disturbances. *IEEE Trans. Cybern.* **2018**, *49*, 2398–2403. [\[CrossRef\]](#)
21. Wang, J.L.; Wu, H.N.; Huang, T.; Ren, S.Y. Analysis and Pinning Control for Output Synchronization and H_∞ Output Synchronization of Multi-weighted Complex Networks. In *Analysis and Control of Output Synchronization for Complex Dynamical Networks*; Springer: Singapore, 2019; pp. 175–205.
22. Li, Q.; Shen, B.; Wang, Z.; Huang, T.; Luo, J. Synchronization control for a class of discrete time-delay complex dynamical networks: A dynamic event-triggered approach. *IEEE Trans. Cybern.* **2018**, *49*, 1979–1986. [\[CrossRef\]](#) [\[PubMed\]](#)

23. Park, M.; Lee, S.H.; Kwon, O.M.; Seuret, A. Closeness-centrality-based synchronization criteria for complex dynamical networks with interval time-varying coupling delays. *IEEE Trans. Cybern.* **2017**, *48*, 2192–2202. [[CrossRef](#)] [[PubMed](#)]
24. Wang, Z.; He, H.; Jiang, G.P.; Cao, J. Quasi-synchronization in heterogeneous harmonic oscillators with continuous and sampled coupling. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *51*, 1267–1277. [[CrossRef](#)]
25. Wang, Y.W.; Xiao, J.W.; Wen, C.; Guan, Z.H. ynchronization of continuous dynamical networks with discrete-time communications. *IEEE Trans. Neural Netw.* **2011**, *22*, 1979–1986. [[CrossRef](#)] [[PubMed](#)]
26. Quan, W.; Xu, Z.; Liu, M.; Cheng, N.; Liu, G.; Gao, D.; Zhuang, W. AI-driven Packet Forwarding with Programmable Data Plane: A Survey. *IEEE Commun. Surv. Tutor.* **2022**. [[CrossRef](#)]
27. Song, F.; Ma, Y.; You, I.; Zhang, H. Smart Collaborative Evolvment for Virtual Group Creation in Customized Industrial IoT. *IEEE Trans. Netw. Sci. Eng.* **2022**, 1–11.. [[CrossRef](#)]
28. Cao, J.; Li, L. Cluster synchronization in an array of hybrid coupled neural networks with delay. *Neural Netw.* **2009**, *22*, 335–342. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.