Article

On the Role of Matrix-Weights Elements in Consensus Algorithms for Multi-Agent Systems

Joshua Ogbebor [†] and Xiangyu Meng ^{*,†}

Division of Electrical & Computer Engineering, Louisiana State University, Baton Rouge, LA 70803, USA; jogbeb1@lsu.edu

* Correspondence: xmeng5@lsu.edu

† These authors contributed equally to this work.

Abstract: This paper examines the roles of the matrix weight elements in matrix-weighted consensus. The consensus algorithms dictate that all agents reach consensus when the weighted graph is connected. However, it is not always the case for matrix weighted graphs. The conditions leading to different types of consensus have been extensively analysed based on the properties of matrix-weighted Laplacians and graph theoretic methods. However, in practice, there is concern on how to pick matrix-weights to achieve some desired consensus, or how the change of elements in matrix weights affects the consensus algorithm. By selecting the elements in the matrix weights, different clusters may be possible. In this paper, we map the roles of the elements of the matrix weights in the systems consensus algorithm. We explore the choice of matrix weights to achieve different types of consensus and clustering. Our results are demonstrated on a network of three agents where each agent has three states.

Keywords: matrix-weighted graphs; multi-agent systems; clustered consensus; global consensus

Citation: Ogbebor, J.; Meng, X. On the Role of Matrix-Weights Elements in Consensus Algorithms for Multi-Agent Systems. *Network* **2021**, *1*, 233–246. <https://doi.org/10.3390/network1030014>

Academic Editors: Dakshnamoorthy Manivannan and Alexey Vinel

Received: 20 August 2021

Accepted: 9 October 2021

Published: 15 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 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

Multi-agent systems (MASs) consists of multiple autonomous agents [1], which can be used to solve problems that are difficult or impossible for an individual agent or a monolithic system. Communication and interaction between individual agents are fundamental characteristics of MASs, which allow agents to achieve a global objective despite having access to only local neighbourhood information [2], or edge information [3,4]. Having multiple agents could speed up a system's operation by providing a method for parallel computation, offering advantages in terms of extensibility and flexibility compared to single agent systems [5]. Applications of MASs also extends into cloud computing and social networks [6].

Consensus is one of the fundamental problems in multi-agent coordination. In consensus, agents interact with their neighbors according to a local protocol to ensure that a common value in terms of the state components, is agreed upon globally, by all the agents. This sweeping form of consensus is generally referred to as a global consensus. A clustered consensus is also possible, where some agents agree on some values, different from the consensus value of some other agents [7,8]. MAS is widely studied using the graph theory, in which the vertices and edges represent agents and the inter-agent links, respectively. Conventionally, the inter-agent links have been modelled by scalar weights [9,10] and the consensus algorithm is widely applied at the agent level [11–13]. Matrix-weights can however be used to capture the complexity at the state level for MASs. Particularly, in matrix-weighted consensus, the agreement of all agents in corresponding states may be affected by other states. There are a number of works in the literature that extend the conventional scalar weighted graph to the matrix weighted graph for the consensus problem, including in [14–16] where the conditions for achieving consensus in matrix-weighted discrete and continuous-time consensus algorithms are presented using the properties

of the graph Laplacian. In [15], discrete-time matrix-weighted consensus is studied over undirected and connected graphs considering symmetric matrix weights and a special case of non-symmetric matrix-weights that can achieve consensus control. Lyapunov stability theory for discrete-time systems is also employed to show the system's convergence to consensus. In the same way, in [14], matrix-weighted consensus algorithm is studied with fixed undirected graphs and a necessary and sufficient condition for exponentially reaching a global average consensus and clustered consensus is presented based on the null-space of the matrix-weighted Laplacian.

Generally, when the agent-to-agent link is weighted by positive semi-definite (PSD) matrices, it can be difficult to tell what form of consensus will be present. Clustered consensus can happen even when the graph is connected in the agent level, due to the existence of PSD matrix weights. A typical example, where all the elements of each of the matrix-weights are set to one, for a complete graph on five vertices, there will be no form of consensus for the agents in any of the states.

In this paper, we study how the choice of matrix weights for the inter-agent links affect the presence of clusters in a matrix-weighted graph. A linear algebraic approach is used in [17,18] to examine the properties of the Laplacian matrix and the necessary and sufficient conditions are given for the existence of consensus. We adopt a matrix-element-mapping approach to study how the matrix-weights alter the consensus dynamics. Our approach compares the matrix-based approach to the conventional scalar weighted graphs which helps view the network of each state, and offers hints on the choice of the matrix weights.

The main contributions of this paper are summarized as follows:

- to examine the matrix-weighted graph and show how each element influences the consensus algorithm;
- to study how different forms of consensus can be achieved by the choice of matrix weights;
- to be able to determine the number of clusters and the elements in each of the cluster by looking at the matrix-weight set.

2. Problem Formulation

2.1. Agent Dynamics

The dynamics of agent v_i , $i = 1, \dots, n$ are modeled by a single integrator given by

$$\dot{x}_i(t) = u_i(t), \quad (1)$$

where $x_i(t) \in \mathbb{R}^m$ and $u_i(t) \in \mathbb{R}^m$ denote the states and the inputs of agent v_i , respectively, $i = 1, \dots, n$. Let us define $x_i^{[k]}$ as the k th entry of the column vector x_i for $i = 1, \dots, n$ and $k = 1, \dots, m$.

2.2. Graph Theory

The communication topology of the n agents is modeled by an n -partite graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$ [19]. The vertex set $\mathcal{V}$ is partitioned into n classes $\mathcal{V} = \{v_1, \dots, v_n\}$, where each class represents an agent. Every partition class contains exactly m vertices, which represent the m -dimensional state of an agent. Every edge in $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ has its ends in different classes: vertices in the same partition class are not adjacent. The matrix weight

$$\mathcal{W} = \{W_{ij} | i = 1, \dots, n, j = 1, \dots, n\}$$

models the connections between every two different classes, where $W_{ij} \succeq 0 \in \mathbb{R}^{m \times m}$ shows the connections of the states between agents v_i and v_j . We use $v_j^{[l]} v_i^{[k]}$ to represent an edge from the l th state in agent v_j to the k th state in agent v_i , which means that $x_i^{[k]}$ of agent v_i is affected by $x_j^{[l]}$ of agent v_j . The strength of this connection is characterized by $[W_{ij}]_{kl} = w_{kl}^{ij}$, the (k, l) entry of W_{ij} . If $w_{kl}^{ij} = 0$, $x_j^{[l]}$ of agent v_j has no influence on $x_i^{[k]}$ of agent v_i . If there is at least one entry in W_{ij} in which $w_{kl}^{ij} > 0$, we say agent v_j is a neighbor of agent v_i and

$v_j v_i \in \mathcal{E}$. The set of neighbors of agent v_i is defined as $\mathcal{N}_i \triangleq \{v_j \in \mathcal{V} : \exists w_{kl}^{ij} > 0\}$. An induced subgraph $\mathcal{G}^{[k]}$ of $\mathcal{G}$ with the vertex set

$$\mathcal{V}^{[k]} = \{v_1^{[k]}, \dots, v_n^{[k]}\}$$

is called the k -state graph. A path in $\mathcal{G}^{[k]}$ is given by a sequence of distinct vertices in $\mathcal{V}^{[k]}$ connected by an edge in $\mathcal{E}$. If the induced subgraph $\mathcal{G}^{[k]}$ contains a spanning arborescence, then we say the k -state graph is *connected*. If the k -state graph is not connected, k th-states of all agents can be partitioned into distinct connected components. The graph is connected in the agent level if for any two distinct agents, there is a path to connect at least one state in each agent.

The degree or valency of a given vertex $d(v_i) \in \mathbb{R}^{m \times m}$ is defined as

$$d(v_i) \triangleq \sum_{v_j \in \mathcal{N}_i} W_{ij}$$

that is, the sum of weights on the links from each neighbouring node for all states of agent v_i .

We can define the $nm \times nm$ degree matrix of $\mathcal{G}$

$$\Delta_{\mathcal{W}}(\mathcal{G}) = \text{blkdiag}\{d(v_i), i = 1, \dots, N\}, \quad (2)$$

as the block diagonal matrix of the degrees of all agents in $\mathcal{G}$. The $nm \times nm$ adjacency matrix $A_{\mathcal{W}}(\mathcal{G})$ can be defined as a block matrix, where the (i, j) block is an m -by- m matrix with

$$[A_{\mathcal{W}}]_{ij} = \begin{cases} W_{ij} & \text{if } v_j v_i \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The weighted Laplacian matrix of the graph, $\mathcal{L}_{\mathcal{W}}(\mathcal{G}) \in \mathbb{R}^{nm \times nm}$ is given by

$$\mathcal{L}_{\mathcal{W}} = \Delta_{\mathcal{W}} - A_{\mathcal{W}}. \quad (4)$$

The Laplacian matrix of the induced subgraph $\mathcal{G}^{[k]}$ is denoted by $\mathcal{L}_{\mathcal{W}}^{[k]}$.

Remark 1. For the special case when $m = 1$, the weight W_{ij} for each edge in $\mathcal{E}$ is a scalar $w_{ij} \in \mathbb{R}_{\geq 0}$.

Remark 2. For the special case when $m > 1$ and

$$W_{ij} = w_{ij} I_m$$

with $I_m \in \mathbb{R}^{m \times m}$ being the identity matrix, $\Delta_{\mathcal{W}}(\mathcal{G})$, the $nm \times nm$ degree matrix of $\mathcal{G}$, is the diagonal matrix of the degree of all the vertices in $\mathcal{G}$ given by

$$\Delta_{\mathcal{W}}(\mathcal{G}) = \Delta_w \otimes I_m \quad (5)$$

where

$$\Delta_w = \text{diag}\left\{\sum_{v_j \in \mathcal{N}_i} w_{ij}, i = 1, \dots, n\right\},$$

and the $nm \times nm$ adjacency matrix $A_{\mathcal{W}}(\mathcal{G})$ is defined as

$$A_{\mathcal{W}}(\mathcal{G}) = A_w \otimes I_m \quad (6)$$

where

$$[A_w]_{ij} = \begin{cases} w_{ij} & \text{if } v_j v_i \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

The matrix-weighted Laplacian of the graph, $\mathcal{L}_W(\mathcal{G}) \in \mathbb{R}^{nm \times nm}$ is given by

$$\mathcal{L}_W(\mathcal{G}) = \Delta_W - A_W = \mathcal{L}_w \otimes I_m, \quad (8)$$

where $\mathcal{L}_w = \Delta_w - A_w$.

Remark 3. Matrix weighted graphs are closely related to signed graphs [20]. Let us use a simple example of two agents, where each agent has two states, to illustrate the similarity. The matrix weight is given by $W_{12} = W_{21} = I_2$, and the Laplacian matrix is

$$\mathcal{L} = \begin{bmatrix} I_2 & -I_2 \\ -I_2 & I_2 \end{bmatrix}. \quad (9)$$

If we use the definition of Laplacian matrix for scalar weighted graphs, it is equivalent to say that the weight between $x_1^{[1]}$ and $x_1^{[2]}$ is -1 .

2.3. Cluster Consensus

Consensus and clustering is defined based on the similarity of the agents state values. Clustering configuration across the sates is also defined.

Definition 1. If the states $x_i^{[k]}$ and $x_j^{[k]}$ satisfy the condition

$$\lim_{t \rightarrow \infty} \|x_i^{[k]}(t) - x_j^{[k]}(t)\| = 0,$$

then we say they belong to the same cluster. All k -clusters $\mathcal{C}^k = \{\mathcal{C}_1^k, \dots, \mathcal{C}_{p(k)}^k\}$ form a partition of the k th entry of all states, where $p(k)$ is the number of clusters of the k th entry of all states. Then, all agents are said to reach k -cluster consensus (KCC).

It is easy to verify that all k -clusters satisfy the conditions

$$\bigcup_{i=1}^{p(k)} \mathcal{C}_i^k = \{x_1^{[k]}, \dots, x_n^{[k]}\}$$

and

$$\mathcal{C}_p^k \cap \mathcal{C}_q^k = \emptyset$$

for any $\mathcal{C}_p^k \in \mathcal{C}^k$, $\mathcal{C}_q^k \in \mathcal{C}^k$, and $p \neq q$.

Definition 2. If $p(k) = 1$ for some $k \in \{1, \dots, m\}$ then we say the k th entry of all states reach k -global consensus (KGC). When $p(k) = 1$ for all $k = 1, \dots, m$, then we say all states reach a global consensus (GC).

Note that for different entries i and j of all states, the i -clusters $\mathcal{C}^i$ and the j -clusters $\mathcal{C}^j$ may not be the same, that is,

$$\mathcal{C}^i \neq \mathcal{C}^j$$

for some $i, j \in \{1, \dots, m\}$. In other words, the k -cluster consensus may not be uniform across all entries of all states. However, when the clusters are uniform for all entries of all states, we have the following definition.

Definition 3. If the clusters of all entries satisfying

$$\mathcal{C}^i = \mathcal{C}^j$$

for all $i \neq j$, $i = 1, \dots, m$, and $j = 1, \dots, m$, then we say all states reach a global cluster consensus (GCC).

2.4. Objectives

Our main aim in this paper is to examine how the elements of the matrix-weights $\mathcal{W}$ could be chosen, so as to achieve the desired consensus (KCC, GCC, KGC, GC).

3. Results

3.1. Consensus Control Design

The consensus control protocol for the n agents is given by

$$u_i(t) = \sum_{v_j \in \mathcal{N}_i} W_{ij}(x_j(t) - x_i(t)). \quad (10)$$

where $W_{ij} \in \mathbb{R}^{m \times m}$ is the weight matrix and $[W_{ij}]_{kl} = w_{kl}^{ij}$ for $k = 1, \dots, m$, and $l = 1, \dots, m$. This control law can be written in a compact matrix form as

$$u(t) = -\mathcal{L}_{\mathcal{W}}(\mathcal{G})x(t), \quad (11)$$

where

$$u(t) = [u_1 \cdots u_i \cdots u_n]^T, \quad u_i = [u_i^{[1]} \cdots u_i^{[m]}]^T \\ x(t) = [x_1 \cdots x_j \cdots x_n]^T, \quad x_j = [x_j^{[1]} \cdots x_j^{[m]}]^T$$

Remark 4. For the case when $m = 1$, the consensus control law becomes:

$$u_i(t) = \sum_{v_j \in \mathcal{N}_i} w_{ij}(x_j(t) - x_i(t)),$$

and it can be written in a compact matrix form as:

$$u(t) = -\mathcal{L}_w(\mathcal{G})x(t).$$

Remark 5. For the case when $m > 1$ and

$$W_{ij} = w_{ij}I_m,$$

the consensus control law becomes:

$$u_i(t) = \sum_{v_j \in \mathcal{N}_i} w_{ij}I_m(x_j(t) - x_i(t)),$$

and it can be written in a compact matrix form as:

$$u(t) = -\mathcal{L}_W(\mathcal{G})x(t) = -(\mathcal{L}_w(\mathcal{G}) \otimes I_m)x(t).$$

In order to observe the evolution of $x_i^{[k]}$, we need to study the dynamics of $x_i^{[k]}$, which obeys

$$\dot{x}_i^{[k]} = u_i^{[k]}(t).$$

Here

$$\begin{aligned} u_i^{[k]}(t) &= \sum_{v_j \in \mathcal{N}_i} \sum_{l=1}^m w_{kl}^{ij} (x_j^{[l]}(t) - x_i^{[l]}(t)) \\ &= \sum_{v_j \in \mathcal{N}_i} w_{kk}^{ij} (x_j^{[k]}(t) - x_i^{[k]}(t)) + \sum_{v_j \in \mathcal{N}_i} \sum_{l=1, l \neq k}^m w_{kl}^{ij} (x_j^{[l]}(t) - x_i^{[l]}(t)), \end{aligned} \quad (12)$$

is the k th entry of the control input of agent v_i defined in Equation (10). The coefficient w_{kl}^{ij} is the element on the k th row and the l th column of the matrix weight W_{ij} . When $k \neq l$, w_{kl}^{ij} is an off-diagonal element on the k th row of the matrix weight W_{ij} , and it qualifies the effect of the l th states of the neighbors of agent v_i on its k th state. When $k = l$, w_{kk}^{ij} is a diagonal element of the matrix weight W_{ij} , and it qualifies the effect of the k th states of the neighbors of agent v_i on its k th state.

3.2. Non-Diagonal Matrix-Weights

When the matrix weight W_{ij} is a non-diagonal matrix, we examine two possible scenarios.

3.2.1. Control Law Depends on Other States in the Same Cluster

Theorem 1. Assume that the k -state graph has incoming links only from the same connected components in other state graphs, and the connected components with outgoing links to the k -state graph have no incoming links. The following statements hold:

1. If the k -state graph is connected, the k th states of all agents reach k -global consensus (KGC).
2. If the k -state agents are not connected, the k th states of all agents reach k -cluster consensus (KCC).

Proof of Theorem 1. Assume that the k -state graph has incoming links from the same connected components in the l -state graph. Then

$$x_j^{[l]}(t) - x_i^{[l]}(t) \rightarrow 0$$

as $t \rightarrow \infty$ since $x_j^{[l]}$ and $x_i^{[l]}$ are in the same connected components without incoming links. The dynamics of all k states can be written as

$$\dot{x}^{[k]}(t) = -\mathcal{L}_{\mathcal{W}}^{[k]} x^{[k]}(t) + u^{[k]}(t)$$

where $u^{[k]}(t) \rightarrow 0$ as $t \rightarrow \infty$. When the k -state graph is connected, the k th states of all agents reach KGC and KCC otherwise based on input-to-state stability. $\square$

3.2.2. Control Law Depends on Other States in Different Clusters

In this case, the evolution of all states follows

$$\dot{x}(t) = u(t)$$

where $u(t)$ is given in Equation (11). By solving the above differential equation, we have

$$x(t) = e^{-t\mathcal{L}_{\mathcal{W}}} x(0). \quad (13)$$

The choice of the weight matrix $\mathcal{W}$ will affect the rate of convergence of the consensus algorithm of the states of all agents. Since the Laplacian matrix $\mathcal{L}_{\mathcal{W}}$ is positive semi-definite, all states will converge to a constant value eventually.

Theorem 2. Assume that the k -state graph has incoming links from y different connected components in the l -state graph. The following statements hold:

1. There will at least be y clusters for the corresponding agent states in the graph causing a k -cluster consensus (KCC).
2. In the k -state graph, if $x_j^{[k]}$ and $x_i^{[k]}$ belong to the same connected component, which has only one incoming link from the other state graphs, then $x_j^{[k]}, x_i^{[k]} \in \mathcal{C}_p^k$.
3. In the k -state graph, assume that $x_j^{[k]}, x_i^{[k]}$, and $x_p^{[k]}$ belong to the same connected components. If $x_i^{[k]}$ and $x_p^{[k]}$ have incoming links from different connected components in the l -state graph, and $x_j^{[k]}$ has no incoming links from other state graphs, then $x_j^{[k]}$ will be in a different cluster of $x_i^{[k]}$ and $x_p^{[k]}$.

Proof of Theorem 2. The y different connected components in the l -state graph will at least form y different clusters. The y different clusters will be y virtual leaders to states in the k -state graph. Therefore, there will be KCC with a minimum of y clusters.

When there is only one leader for a connected component in the k -state graph, the connected component will form a cluster.

When the connected component including $x_j^{[k]}, x_i^{[k]}$, and $x_p^{[k]}$ has incoming links from different connected components in other state graphs, it is equivalent to the case that all three states have two different leaders. Then all the three states will be in different clusters. $\square$

3.3. Diagonal Matrix Weights

The absence of off-diagonal elements in the matrix-weights W_{ij} guarantees that there is no cross-entry state dependence, that is, $w_{kl}^{ij} = 0$ when $k \neq l$. The control law reduces to:

$$u_i^{[k]}(t) = \sum_{v_j \in \mathcal{N}_i} w_{kk}^{ij} (x_j^{[k]}(t) - x_i^{[k]}(t)).$$

Here we discuss two different cases based on positive definite or positive semi-definite of matrices W_{ij} .

3.3.1. Positive Definite Matrix Weights

When diagonal positive definite (PD) matrices constitute the matrix-weights in the weight set $\mathcal{W} = \{W_{ij} : v_j v_i \in \mathcal{E}\}$, it implies that $w_{kk}^{ij} \neq 0$ for all $k = 1, \dots, m$, and for all $v_j v_i \in \mathcal{E}$.

Define

$$x^{[k]} = [x_1^{[k]} \dots x_n^{[k]}]^T$$

as the vector containing the k -state of all agents. Its dynamics follow

$$\dot{x}^{[k]}(t) = -\mathcal{L}_{\mathcal{W}}^{[k]} x^{[k]}(t) \quad (14)$$

The solution of the above differential equation is given by

$$x^{[k]}(t) = e^{-t\mathcal{L}_{\mathcal{W}}^{[k]}} x^{[k]}(0). \quad (15)$$

The choice of w_{kk}^{ij} will affect the rate of convergence of the consensus algorithm of the k -state of all agents.

Consider the case when the matrix weights are scalars multiples of the identity matrix. We know that for any square matrices $A_n \in \mathbb{R}^{n \times n}$ with $\lambda_1, \dots, \lambda_n$, and $B_m \in \mathbb{R}^{m \times m}$ with $\mu_1, \dots, \mu_m$ that the eigenvalues of $A_n \otimes B_m$ are $\lambda_i \mu_j$, $i = 1, \dots, n$, $j = 1, \dots, m$ [21]. Hence,

the eigenvalues of $\mathcal{L}_{\mathcal{W}} = \mathcal{L}_w(\mathcal{G}) \otimes I_m$ are the eigenvalues of $\mathcal{L}_w(\mathcal{G})$ repeated m times, and $\mathcal{L}_{\mathcal{W}}^{[k]} = \mathcal{L}_w$ for all $k = 1, \dots, m$. Then Equation (15) becomes

$$x^{[k]}(t) = e^{-\mathcal{L}_w t} x^{[k]}(0),$$

which implies GC.

Theorem 3. Assume that all the matrix-weights are diagonal PD matrices. The following statements hold:

1. If the k -state graph is connected, there will be a global consensus (GC) across the m states of all agents.
2. If the k -state graph is not connected, there will be a global cluster consensus (GCC) across the m states such that $C^i = C^j$ for all $i \neq j$, $i = 1, \dots, m$, and $j = 1, \dots, m$. Moreover, the number of clusters of the states is determined by the number of connected components of the k -state graph.

Proof of Theorem 3. If the k -state graph is connected, all agents are connected at the agent level. Since all matrix weights are diagonal PD matrices, all l -state graphs for all $l = 1, \dots, m$ are connected without incoming links. Hence, there will be a GC.

Assume that the k -state graph is not connected. If the k th states of agents v_i and v_j are in different connected components, (that is, $w_{kk}^{ij} = 0$), then, v_i and v_j are not adjacent at the agent level, (that is, $W_{ij} = 0$) because all matrix weights are diagonal PD matrices. This implies that $w_{ll}^{ij} = 0$ for all $l = 1, \dots, m$. In other words, if agents v_i and v_j are not adjacent in the k th state, they are not adjacent in any other states. If the k th states of v_i and v_j are in the same connected components, then all other states of v_i and v_j are in the same connected components since all matrix weight are diagonal PD matrices. Hence, there will be a GCC. $\square$

3.3.2. Positive Semi-Definite Matrix Weights

When diagonal positive semi-definite (PSD) matrices constitute the matrix-weights in the weight set $\mathcal{W} = \{W_{ij} : v_j v_i \in \mathcal{E}\}$, it implies that for any $W_{ij} \neq 0$, there exist k and l such that $w_{kk}^{ij} = 0$ and $w_{ll}^{ij} \neq 0$. In this case, the edge $v_j^{[k]} v_i^{[k]}$ does not exist in the k th states between agents v_i and v_j , which leads to $C^k \neq C^l$ in general.

Theorem 4. Assume that the matrix weights are diagonal PSD matrices, then for any state k . The following statements hold:

1. If the k -state graph is connected, there will be k -global consensus (KGC).
2. If the k -state graph is not connected, then there is a k -cluster consensus (KCC). Moreover, the number of clusters of the states is determined by the number of connected components of the k -state graph.

Proof of Theorem 4. For a diagonal PSD matrix weight, the k -state graph has no incoming links from other state graphs. Hence, there will be KGC if the k -state agents are connected and a KCC otherwise. $\square$

Hence, by the choice of the diagonal entries of the matrix-weights as zero or non-zero, the graph structure of the states can be made different for different states.

4. Simulations

To demonstrate the effects of matrix weights on the consensus algorithm, we choose a network of three agents, where each agent has three states. The initial conditions are given by

$$x_1(0) = [4 \quad 25 \quad 30]^T, x_2(0) = [13 \quad 2 \quad 12]^T, x_3(0) = [21 \quad 18 \quad 10]^T.$$

4.1. Non-Diagonal Matrix-Weights

4.1.1. Control Law Dependent on Other State Values in the Same Cluster

For the graph topology shown in Figure 1, the state trajectory is presented in Figure 2. It can be seen from Figure 1 that the 2-state graph is connected, and is not dependent on any other state. Therefore, there is a 2-global consensus (2GC): $\mathcal{C}_1^2 = \{x_1^{[2]}, x_2^{[2]}, x_3^{[2]}\}$. The 1-state graph is connected but is dependent on agent states from cluster $\mathcal{C}_1^2$. According to Theorem 1, there is a 1-global consensus (1GC) since all 2-states belong to the same cluster. For 3-state, the 3-state graph is disconnected and is dependent on 2-state values from the cluster $\mathcal{C}_1^2$. Therefore, $\{x_1^{[3]}, x_3^{[3]}\} \in \mathcal{C}_1^3$ form a first cluster and $\{x_2^{[3]}\} \in \mathcal{C}_2^3$ forms another cluster based on Theorem 1. Hence, there is a 3-cluster consensus (3CC).

Figure 1. Graph topology with non-diagonal matrix weights: 1GC, 2GC and 3CC.

Figure 2. State trajectory for non-diagonal matrix weighted graph: 1GC, 2GC and 3CC.

4.1.2. Control Law Dependent on Other State Values in Different Clusters

The graph topology shown in Figure 3 is obtained from Figure 1 by setting $w_{22}^{32} = 0$, and $w_{33}^{32} = 3$. The state trajectory is presented in Figure 4. It can be seen from Figure 3, the 2-state graph is disconnected and do not depend on any other state values. Thus, for the 2-state, there are now two clusters $\mathcal{C}_1^2 = \{x_1^{[2]}, x_2^{[2]}\}$, and $\mathcal{C}_2^2 = \{x_3^{[2]}\}$.

Figure 3. State graphs for non-diagonal matrix weights: 1CC, 2CC and 3CC.

Figure 4. State trajectory for non-diagonal matrix weighted graph: 1CC, 2CC and 3CC.

For the 1-state, though the graph is connected, 1-state global consensus is not guaranteed since it depends on agent states from different clusters in $\mathcal{C}^2$. There will be at least 2 clusters according to Theorem 2. Moreover, since $x_1^{[1]}$ is linked to states $x_2^{[1]} \in \mathcal{C}_1^1$ and $x_3^{[1]} \in \mathcal{C}_2^1$, it will be in a different cluster: $x_1^{[1]} \in \mathcal{C}_3^1$ according to Theorem 2.

The same reasoning applies to the 3-state graph. The graph is connected and depends on agent states from different clusters in $\mathcal{C}^2$. There will be at least 2 clusters according to

Theorem 2. Since $u_2^{[3]}$ is not dependent on any state and is connected to only $v_3^{[3]}$, there are two clusters in $\mathcal{C}^3$ (Theorem 2).

4.2. Diagonal Matrix-Weights

The consensus algorithm of the graph with all matrix weights equal to some scalar multiples of the identity matrix is simulated and the state trajectory is plotted in Figure 5. The matrix weights are chosen as

$$W_{12} = 3I_{3 \times 3}, \quad W_{32} = 1.9I_{3 \times 3}, \quad W_{13} = 5I_{3 \times 3}. \quad (16)$$

The graph structure per state is the same for all states with similar convergence properties when the matrix-weights are scalars multiplying the identity matrices. When the matrix weights are instead chosen to be general PD diagonal matrices as

$$W_{12} = \text{diag}\{1.3, 4.0, 6.1\}, \quad W_{32} = \text{diag}\{0.8, 2.1, 3.1\}, \quad W_{13} = \text{diag}\{8.1, 1.4, 3.1\},$$

the consensus algorithm is simulated and the state trajectory is shown in Figure 6. The convergence properties are different from state to state. The choice of w_{kk}^{ij} will affect the rate of convergence of the consensus algorithm of the k th state according to Equation (15). In all cases, a global consensus (GC) is achieved based on Theorem 3.

Figure 5. State trajectory for diagonal matrix weighted graph: scalar multiple of the identity matrix (GC with similar convergence properties across the states).

When the matrix weights are diagonal PSD matrices chosen as

$$W_{12} = \text{diag}\{1.3, 4.0, 0\}, \quad W_{32} = \text{diag}\{0, 0, 0\}, \quad W_{13} = \text{diag}\{8.1, 0, 3.1\}.$$

the consensus algorithm is simulated and the state trajectory is shown in Figure 7. Since the 1-state graph is connected due to $w_{11}^{12} \neq 0$ and $w_{11}^{13} \neq 0$, there is a 1-global consensus (1GC). For 2-states, there is a 2-cluster consensus (2CC) because $w_{22}^{13} = w_{22}^{32} = 0$. There are no links between $\{x_1^{[2]}, x_2^{[2]}\} \in \mathcal{C}_1^2$ and $\{x_3^{[2]}\} \in \mathcal{C}_2^2$. The same analogy applies for the 3-cluster consensus (3CC) in 3-states: $\{x_1^{[3]}, x_3^{[3]}\} \in \mathcal{C}_1^3$ and $\{x_2^{[3]}\} \in \mathcal{C}_2^3$ because $w_{33}^{12} = w_{33}^{32} = 0$.

Figure 6. State trajectory for general diagonal PD matrix weighted graph (GC with different convergence properties across the states).

Figure 7. State trajectory for general diagonal PSD matrix weighted graph (KGC and KCC).

5. Discussion

Matrix weights allow for coupling the consensus of a state to state values of another agent. These inter-dependencies may get complex if for instance each agent state in a system is dependent on every other state. However, when there are no much complexities, the nature of the consensus can be determined based on the number of connected components of the state graph and the clusters of the agent states linked to it. Further work includes the prediction of the number of clusters when there are complex inter-dependencies among the states in the multi-agent system.

6. Conclusions

In this paper, the role of each element in the matrix-weights of a multi-agent system has been studied under two classifications: diagonal and non-diagonal elements. By selecting the elements in the matrix weights, different clusters across each of the states may be possible. In particular, with diagonal positive-definite matrix weights, global consensus is guaranteed for each state if the graph is connected at the agent level; otherwise, a similar clustered consensus for each state is achieved. Diagonal positive semi-definite matrix weights result in a global consensus for a state if the graph is connected at the state level, and a clustered otherwise. For non-diagonal matrix weights, there will be global consensus for a state k if the graph is connected at the state level and the k -state graph is independent of agent states in different clusters; otherwise, there will be cluster consensus. Analyses have been carried out for determining the number of clusters per state and the results have further been demonstrated via simulation of a multi-agent system consisting of three agents.

Author Contributions: Conceptualization, J.O.; methodology, J.O. and X.M.; software, J.O. and X.M.; validation, X.M. and J.O.; formal analysis, X.M. and J.O.; writing—original draft preparation, J.O.; writing—review and editing, X.M.; visualization, X.M.; supervision, X.M.; project administration, X.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Abbreviations

The following abbreviations are used in this manuscript:

MASs	Multi-agent Systems
PD	Positive definite
PSD	Positive semi-definite
GC	Global consensus
GCC	Global clustered consensus
KCC	Cluster consensus for state k
KGC	Global consensus for state k

References

1. Wooldridge, M. *An Introduction to Multiagent Systems*; John Wiley & Sons: Hoboken, NJ, USA, 2009.
2. Mesbahi, M.; Egerstedt, M. *Graph Theoretic Methods in Multiagent Networks*; Princeton University Press: Princeton, NJ, USA, 2010; Volume 33.
3. Zhao, H.; Meng, X.; Wu, S. Distributed edge-based event-triggered coordination control for multi-agent systems. *Automatica* **2021**, *132*, 109797. [[CrossRef](#)]
4. Shi, C.X.; Yang, G.H.; Li, X.J. Data-Based Fault-Tolerant Consensus Control for Uncertain Multiagent Systems Via Weighted Edge Dynamics. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 2548–2558. [[CrossRef](#)]
5. Zhu, B.; Xie, L.; Han, D.; Meng, X.; Teo, R. A survey on recent progress in control of swarm systems. *Sci. China Inf. Sci.* **2017**, *60*, 070201. [[CrossRef](#)]
6. Dorri, A.; Kanhere, S.S.; Jurdak, R. Multi-Agent Systems: A Survey. *IEEE Access* **2018**, *6*, 28573–28593. [[CrossRef](#)]
7. Qin, J.; Ma, Q.; Shi, Y.; Wang, L. Recent advances in consensus of multi-agent systems: A brief survey. *IEEE Trans. Ind. Electron.* **2016**, *64*, 4972–4983. [[CrossRef](#)]
8. Han, Y.; Lu, W.; Chen, T. Cluster consensus in discrete-time networks of multiagents with inter-cluster nonidentical inputs. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *24*, 566–578. [[PubMed](#)]
9. Meng, X.; Chen, T. Event based agreement protocols for multi-agent networks. *Automatica* **2013**, *49*, 2125–2132. [[CrossRef](#)]
10. Meng, X.; Xie, L.; Soh, Y.C.; Nowzari, C.; Pappas, G.J. Periodic event-triggered average consensus over directed graphs. In Proceedings of the 2015 54th IEEE Conference on Decision and Control (CDC), Osaka, Japan, 15–18 December 2015; pp. 4151–4156.
11. Meng, X.; Xie, L.; Soh, Y.C. Asynchronous periodic event-triggered consensus for multi-agent systems. *Automatica* **2017**, *84*, 214–220. [[CrossRef](#)]
12. Li, Y.; Tan, C. A survey of the consensus for multi-agent systems. *Syst. Sci. Control Eng.* **2019**, *7*, 468–482. [[CrossRef](#)]
13. Chen, Y.; Lu, J.; Yu, X.; Hill, D.J. Multi-agent systems with dynamical topologies: Consensus and applications. *IEEE Circ. Syst. Mag.* **2013**, *13*, 21–34. [[CrossRef](#)]

14. Trinh, M.H.; Van Nguyen, C.; Lim, Y.H.; Ahn, H.S. Matrix-weighted consensus and its applications. *Automatica* **2018**, *89*, 415–419. [[CrossRef](#)]
15. Van Tran, Q.; Trinh, M.H.; Ahn, H.S. Discrete-Time Matrix-Weighted Consensus. *arXiv* **2020**, arXiv:2006.15530.
16. Pan, L.; Shao, H.; Li, D.; Xi, Y. Dynamic Event-Triggered Consensus of Multi-agent Systems on Matrix-weighted Networks. *arXiv* **2021**, arXiv:2106.06198.
17. Su, H.; Chen, J.; Yang, Y.; Rong, Z. The Bipartite Consensus for Multi-Agent Systems With Matrix-Weight-Based Signed Network. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 2019–2023. [[CrossRef](#)]
18. Liu, B.; Su, H.; Wu, L.; Shen, X. Controllability for multi-agent systems with matrix-weight-based signed network. *Appl. Math. Comput.* **2021**, *411*, 126520.
19. Diestel, R. *Graph Theory*; Springer: Berlin, Germany, 2006.
20. Yu, H.; Chen, X.; Chen, T.; Hao, F. Event-triggered bipartite consensus for multiagent systems: A Zeno-free analysis. *IEEE Trans. Autom. Control* **2019**, *65*, 4866–4873. [[CrossRef](#)]
21. Laub, A.J. *Matrix Analysis for Scientists and Engineers*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2004; Volume 91.