Design and Analysis of a Novel Distributed Gradient Neural Network for Solving Consensus Problems in a Predefined Time

Lin Xiao, Lei Jia, Jianhua Dai, Yingkun Cao, Yiwei Li, Quanxin Zhu, Jichun Li and Min Liu

Abstract—In this paper, a novel distributed gradient neural network (DGNN) with predefined-time convergence is proposed to solve consensus problems widely existing in multi-agent systems. Compared with previous gradient neural networks (GNNs) for optimization and computation, the proposed DGNN model works in a non-fail-connected way, of which each neuron only needs the information of neighbor neurons to converge to the equilibrium point. The convergence and asymptotic stability of the DGNN model are proved according to the Lyapunov theory. In addition, based on a relatively loose condition, three novel nonlinear activation functions are designed to speed up the DGNN model to predefined-time convergence, which is proved by rigorous theory. Computer numerical results further verify the effectiveness, especially the predefined-time convergence, of the proposed nonlinearly activated DGNN model to solve various consensus problems of multi-agent systems. Finally, a practical case of the directional consensus is presented to show the feasibility of the DGNN model and a corresponding connectivity-testing example is given to verify the influence on the convergence speed.

Index Terms—Gradient neural network, asymptotic stability, predefined-time convergence, consensus.

I. INTRODUCTION

The consensus problem of Multi-Agent System (MAS), which is a branch of the control field, aims to make the state of the multi-agent system consistent [1], [2], such as the speed and direction, when each agent can only receive information from its neighbors and there is no global control [3]. It enables the internal state of a multi-agent network consistent through a distributed protocol that only utilizes local information [4], [5]. Nowadays, this kind of algorithm shines in the fields of formation control [6], [7], satellite group coordination [8], network congestion control [9], manipulator coordinated control [10], [11], communication network synchronization [12], large-scale machine learning problems [13] and so on.

This work was supported in part by the National Natural Science Foundation of China under Grant 61866013, Grant 61976089, and Grant 61966014; and in part by the Natural Science Foundation of Hunan Province of China under Grant 2021JJ20005, Grant 18A289, Grant 2018TP1018, and Grant 2018RS3065. (Corresponding authors: Lin Xiao and Jianhua Dai).

L. Xiao, L. Jia, J. Dai, Y. Cao and Y. Li are with Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, Changsha 410081, China, and also with MOE-LCSM, Hunan Normal University, Changsha 410081, China (e-mails: xiaolin5@hunnu.edu.cn; jhdai@hunnu.edu.cn).

Q. Zhu is with the MOE-LCSM, and also with School of Mathematics and Statistics, Hunan Normal University, Changsha 410081, China.

J. Li is with School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K.

M. Liu is with College of Electric and Information Technology, Hunan University, Changsha 410082 China.

In [14], a consensus protocol that converges asymptotically to the average of the initial values of the states is proposed. From the stability conclusion of the switching system, it can be known that, even on dynamic networks, this protocol can be reached by arbitrary switching between high-connected graphs [15]. Based on the theory of stability in a finite or predefined time [16], [17], [18], a consensus protocol with enhanced convergence is proposed. In [19], continuous and discrete algorithms with finite-time convergence (FTC) for multi-agent system consensus are proposed. But the least upper bound of convergence time is related to the initial state. Naturally, an important generalization of the above approaches is the predefined-time convergent consensus. In this case, there exists a supremum of convergence time independent of the initial state. Therefore, the property of predefined-time convergence is an ideal criterion for designing a consensus protocol. According to the standard of predefined-time convergence [17], several consensus schemes are designed, such as [20], [21], [22], [23].

Although there have already been many clever or complex algorithms proposed to solve consensus problems, this fact does not preclude the possibility of studying this issue from a different perspective. Hopfield-like neural network (HNN) [24], as a time-varying problem-solving system, is naturally suitable for automatic control fields, such as robot or manipulator trajectory tracking [25], [26], adaptive predictive control [27], chaos synchronization [28], etc. The greatest advantage of the neural network is that it can flexibly change different activation functions to adapt to different application scenarios. Therefore, the authors make efforts to use the theoretical framework of the HNN to design a class of consensus algorithms and explain its effectiveness.

There are two main design frameworks for the HNN model: one is the gradient neural network (GNN) [29], [30], [31] based on gradient descent optimization, and the other is the zeroing neural network (ZNN) [32], [33] based on the error settlement. They are both the generators of a dynamic model and provide some rationality and explanation for constructing a dynamic model. Besides, they have a mature theory system about convergence analysis. It’s important to note that one advantage of GNN over the ZNN model is that it is easy to implement distributed computation in practice. The purpose of this study is to design a distributed GNN (DGNN) to solve consensus problems. For HNN or GNN, convergence performance is one of the most important indicators. Generally, the predefined-time convergence (PTC) is the best performance for
HNN, and exponential convergence or finite-time convergence is the premise to achieve PTC. Therefore, another major work of this study is to design several activation functions to accelerate DGNN to possess the predefined-time convergence. In [7], two classes of consensus protocols for networks composed of nodes with first-order dynamics are proposed, and they are applied to the multi-agent formation control problem guaranteeing predefined-time convergence. In this paper, we are committed to extending the application scope of GNN to the field of multi-agent consistency. Especially, based on the GNN framework, a novel DGNN model is proposed to solve consensus problems of multi-agent systems. Compared with [7], this work is a novel perspective for consensus problems and has certain research and application value.

The structure of the rest of this paper is as follows. In Section II, we will describe the concepts of consensus and the related graph theory. In Section III, according to the GNN framework, the novel DGNN model is proposed to solve consensus problems of multi-agents. In Section IV, the asymptotic stability and conditions for the DGNN model with PTC are first discussed through theoretical analysis, and then three novel PTC functions are proposed. In Section V, three numerical experiments for the proposed functions are first presented, and a practical application case of the directional multi-agent consistency, and the feasibility of designing a proxy agent and all of its neighbors in the system. It allows a proxy network to make the state of agents consistent in a distributed manner, only using local information. For a multi-agent system with \( n \) nodes, dynamics of the \( i \)-th agent is defined by

\[
\dot{x}_i(t) = \mu_i(t), \quad i \in \mathcal{V}(\mathcal{G}) = \{1, 2, ..., n\},
\]

where \( \mathcal{V}(\mathcal{G}) \) denotes the set of all nodes of Graph \( \mathcal{G} \), \( x_i(t) \) is the state of the node \( i \), and \( \mu_i(t) \) is the control input for the node \( i \).

**Definition 1**: Given a consensus protocol \( \mu_i, i \in \mathcal{V}(\mathcal{G}) \), if there exists \( \lim_{t \to \infty} |x_i(t) - x_j(t)| = 0 \) for \( \forall x_i(0) \) and \( \forall i, j \in \mathcal{V}(\mathcal{G}) \), or to say \( \mathbf{x} \in \mathbb{R}^n|y = c \cdot (1, 1, ..., 1)^T, c \in \mathbb{R} \), where \( \mathbb{R}^n \) is used to represent the consensus solution set, the states of multi-agent system in Eq. (1) are said to reach or achieve consensus.

### II. Preliminaries

In this section, we will describe the concept of consensus and its tools: graph theory [34], as well as the GNN framework [30].

#### A. Consensus Concept

In general, an autonomous and adaptive node is called an agent, such as an intelligent device or computer system. A multi-agent system has the following characteristics.

1) Each node can only obtain the local information (i.e., the information of the nearest neighbor).
2) There is no global control over the group behavior of all agents in the network.
3) The behavior of each agent is autonomous.

The DGNN model contracts the interaction between each agent and all of its neighbors in the system. It allows a proxy network to make the state of agents consistent in a distributed manner, only using local information. For a multi-agent system with \( n \) nodes, dynamics of the \( i \)-th agent is defined by

\[
\dot{x}_i(t) = \mu_i(t), \quad i \in \mathcal{V}(\mathcal{G}) = \{1, 2, ..., n\},
\]

where \( \mathcal{V}(\mathcal{G}) \) denotes the set of all nodes of Graph \( \mathcal{G} \), \( x_i(t) \) is the state of the node \( i \), and \( \mu_i(t) \) is the control input for the node \( i \).

**Definition 1**: Given a consensus protocol \( \mu_i, i \in \mathcal{V}(\mathcal{G}) \), if there exists \( \lim_{t \to \infty} |x_i(t) - x_j(t)| = 0 \) for \( \forall x_i(0) \) and \( \forall i, j \in \mathcal{V}(\mathcal{G}) \), or to say \( \mathbf{x} \in \mathbb{R}^n|y = c \cdot (1, 1, ..., 1)^T, c \in \mathbb{R} \), where \( \mathbb{R}^n \) is used to represent the consensus solution set, the states of multi-agent system in Eq. (1) are said to reach or achieve consensus.

#### B. Graph Theory

This subsection explains the basis of graph theory involved in this paper [34]. The directed graph network of the system with \( n \) linked nodes is expressed as follows:

\[
\mathcal{G} = (\mathcal{V}, \mathcal{E}),
\]

where \( \mathcal{V} = \{v_1, v_2, ..., v_n\} \) denotes a finite set of vertices (or nodes), \( \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \) represents a set of the link between two vertices. Such a graph is called a connected graph if they have a link path through 0 or more nodes between any two nodes. The adjacency matrix \( \mathcal{A} \) of graph \( \mathcal{G} \) with \( n \) vertices is represented by \( \mathcal{A}(\mathcal{G}) = [a_{ij}] \), where \( a_{ij} \) is represented as follows:

\[
a_{ij} = \begin{cases} 
1, & j \in \mathcal{N}_i, \\
0, & i \text{ or otherwise}
\end{cases}
\]

where \( \mathcal{N}_i \) denotes a set of neighbor nodes about node \( i \). The number of neighbors of \( i \)-th node is calculated by the formula

\[
d_i = \sum_{j \in \mathcal{N}_i} a_{ij},
\]

which is called the degree of node \( i \). The degree matrix of the graph \( \mathcal{G} \) is denoted as

\[
\mathcal{D}(\mathcal{G}) = \Lambda(d_1, d_2, ..., d_n),
\]

where \( \Lambda(\cdot) \) is the vector diagonalization function.

### Table I: The main terms and notations used in the work

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGNN</td>
<td>distributed gradient neural network</td>
</tr>
<tr>
<td>PTC</td>
<td>predefined-time convergence</td>
</tr>
<tr>
<td>( \mathbb{R} )</td>
<td>the set of real numbers</td>
</tr>
<tr>
<td>( \mathbb{N} )</td>
<td>the set of natural numbers</td>
</tr>
<tr>
<td>( \mathbb{I} )</td>
<td>the set of ( n )-dimensional real-valued vector where all the elements are equal</td>
</tr>
<tr>
<td>( \mathcal{A} )</td>
<td>the adjacency matrix</td>
</tr>
<tr>
<td>( \mathcal{D} )</td>
<td>the degree matrix</td>
</tr>
<tr>
<td>( \mathcal{L} )</td>
<td>the Laplacian matrix</td>
</tr>
<tr>
<td>( \mathcal{N}_i )</td>
<td>the set of the neighbors of the ( i )-th node</td>
</tr>
<tr>
<td>( \Lambda(\cdot) )</td>
<td>the vector diagonalization function</td>
</tr>
<tr>
<td>( \lambda_i(\cdot) )</td>
<td>the ( i )-th smallest eigenvalue of a matrix</td>
</tr>
<tr>
<td>( \text{sign}(\cdot) )</td>
<td>the sign function</td>
</tr>
</tbody>
</table>
The Laplacian matrix of graph $G$ is denoted as

$$\mathcal{L}(G) = D(G) - A(G).$$  \hfill (5)

It should be emphasized that this article focuses only on undirected graphs, which have a symmetric Laplacian matrix. In addition, the Laplacian matrix possesses many excellent properties as follows.

**Lemma 1**: [14] Let $G$ be a graph containing $n$ nodes, $A$ its adjacency matrix, $D$ its degree matrix and $\mathcal{L}$ its Laplacian matrix. The equality

$$\mathcal{L}x = 0$$  \hfill (6)

holds for $x \in \mathbb{I} = \{ y \in \mathbb{R}^n | y = c \cdot (1, 1, \ldots, 1)^T, c \in \mathbb{R} \}$.

**Lemma 2**: [14] Let $G$ be a connected undirected graph containing $n$ nodes, $A$ its adjacency matrix, $D$ its degree matrix and $\mathcal{L}$ its Laplacian matrix. The equality

$$x^T \mathcal{L}x = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} (x_i - x_j)^2$$  \hfill (7)

holds for $\forall x \in \mathbb{R}^n$, and satisfies $x^T \mathcal{L}x = 0$ if and only if $x \in \mathbb{I}$.

It follows from Lemmas 1 and 2 that for Laplacian matrix $\mathcal{L}$ of any graph $G$, the smallest eigenvalue is $\lambda_1(\mathcal{L}) = 0$ and its corresponding eigenvector is $1 = [1, 1, \ldots, 1]^T$.

**Lemma 3**: [35] Let $G$ be a connected graph and $\mathcal{L}$ its Laplacian matrix. The second smallest eigenvalue of $\mathcal{L}$, denoted by $\lambda_2(\mathcal{L})$, satisfies $\lambda_2(\mathcal{L}) = \min_{x \notin \mathbb{I}} \frac{x^T \mathcal{L}x}{x^T x}$ and $\lambda_2(\mathcal{L}) > 0$.

The second smallest eigenvalue $\lambda_2(\mathcal{L})$ is known as the algebraic connectivity of the graph $G$.

**Lemma 4**: Let $G$ be a connected graph and $\mathcal{L}$ its Laplacian matrix. For every $x \in \mathbb{R}^n, x^T \mathcal{L}x \geq \lambda_2(\mathcal{L})x^T \mathcal{L}x$.

**Remark 1**: Let us consider a special case: an undirected graph has two nodes, and the corresponding Laplacian matrix is $\mathcal{L} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$. Then, we can calculate that $x^T \mathcal{L}x = (x_1 - x_2)^2$. If $x_1 = x_2 + v$ with $v$ being a bounded constant, according to the definition in (7), the expression $x^T \mathcal{L}x$ does not satisfy the radially unbounded condition used in Lyapunov theory. Thus, in this paper, we are only concerned with the multi-agent system whose nodes satisfy $n \geq 3$ to guarantee the theoretical analysis, which is also a common situation in practical applications.

### III. DISTRIBUTED GRADIENT NEURAL NETWORK

In this part, the GNN design framework is first summarized [30], [31]. Then, according to this design framework, the novel DGN model is proposed to solve consensus problems of multi-agents.

#### A. GNN Design Framework

The gradient neural network (GNN) is a class of typical neural network, which is widely used in the on-line solution of various matrix equations (including the Sylvester equation). The GNN design framework is summarized as follows.

1. The equation to be solved is transformed into the standard form $F(x) = 0$ for $n$ unknown elements $x$, with $x^*$ denoting its theoretical solution.

2. Construct a convex optimization problem: $\min_{x \in \mathbb{R}^n} J(x)$, which satisfies that $x^*$ is its optimal solution. Then obtain its gradient $\frac{\partial J}{\partial x}$ as well as the linear gradient neural network $\dot{x} = -\frac{\partial J}{\partial x}$, which can effectively solve the equation.

3. Nonlinear activation functions, which are usually monotone increasing odd functions, operates on a certain layer of neurons in the linear gradient neural network $\dot{x} = -\frac{\partial J}{\partial x}$ to obtain a nonlinear gradient neural network with enhanced performance. For example, activation function $\Phi(\cdot)$ acts on the last layer to obtain a nonlinear gradient neural network $\dot{x} = -\Phi(\frac{\partial J}{\partial x})$.

#### B. DGN Model

For connected graph $G$, consider the following optimization problem:

$$\min_{x \in \mathbb{R}^n} J = \frac{1}{4} \sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} (x_i - x_j)^2.$$  \hfill (8)

Obviously, the optimal solution $x^*$ of Eq. (8) belongs to the consensus solution set $\mathbb{I}$. According to Lemma 2, the equivalent expression of $J$ is as follows

$$J = \frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} (x_i - x_j)^2$$  \hfill (9)

$$= \frac{1}{2} x^T \mathcal{L}x,$$

where $\mathcal{L} = \mathcal{L}(G)$. Take the derivative of Eq. (9)

$$\frac{\partial J}{\partial x} = \mathcal{L}x := e,$$  \hfill (10)

where the notation $e$ is the state-error vector.

A simple gradient descent network is obtained: $\dot{x} = -e$, which is a linear DGN model.

According to the nonlinear gradient neural network design approach, the following GNN-based consensus model is proposed:

$$\dot{x} = -\gamma \Phi(e),$$  \hfill (11)

s.t. $\Phi(e) = [\phi(e_1), \phi(e_2), \ldots, \phi(e_n)]^T,$

where $\gamma$ is a design parameter and $\phi(\cdot)$ is an odd activation function with monotone increasing. It can be rewritten as the distributed gradient neural network (DGNN) model:

$$\dot{x}_i = \mu_i := -\gamma \phi(e_i),$$  \hfill (12)

s.t. $e_i = \sum_{j=1}^{n} a_{ij} (x_i - x_j).$

If the activation function $\phi(\cdot)$ is nonlinear, then the DGNN model can perform better than the linear GNN model. For example, the linear GNN model does not have finite-time convergence [36], not to mention predefined-time convergence.

**Remark 2**: Linear GNN models can only achieve exponential convergence rate, while specific nonlinear activation functions can make the DGNN model to achieve finite-time convergence or predefined-time convergence (such functions will be discussed below). Besides, the expression ability of the linear GNN model is too weak. In the field of engineering
applications, it is often difficult to control the speed at different error levels by only adjusting the proportional parameter of the linear GNN model. It is well known that the error-settling-rate of linear GNN models is proportional to the absolute value of the state error, which results in the system state not being effectively adjusted when the error goes to infinity or goes to zero. The nonlinear GNN model overcomes it, and is able to sink faster in both extreme intervals.

It is worth pointing out that, unlike the traditional GNN model, the neurons in the DGNN model are not fully connected. That is, the weight of the non-neighbor neurons is independent of each other, which can satisfy the distributed calculation requirements of the consensus algorithm.

IV. THEORETICAL ANALYSES

In the section, the asymptotic stability and conditions for predefined-time convergence of DGNN model (12) are discussed in details.

A. Convergence Analysis of DGNN Model

Theorem 1: Let $\mathcal{G}$ be a communication network formed by a connected graph, and $\mathcal{L}$ its Laplacian matrix. Consider DGNN model (12) and its corresponding dynamic system:

$$\dot{x} = -\gamma \Phi(e),$$

s.t. $e = \mathcal{L}x,$

where $\Phi(\cdot)$ is a vector-valued function. If $\Phi(\cdot)$ can be decoupled as a group of odd functions:

$$\Phi(e) = [\phi(|e_1|) \cdot \text{sign}(e_1), \phi(|e_2|) \cdot \text{sign}(e_2), \ldots, \phi(|e_n|) \cdot \text{sign}(e_n)]^T,$$

(14)

where $\phi(\cdot)$ is defined as a smooth monotone increasing odd function on $(-\infty, +\infty)$ and $\text{sign}(\cdot)$ denotes a sign function:

$$\text{sign}(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{if } x = 0, \\ -1, & \text{if } x < 0. \end{cases}$$

Then the dynamic system (13) is asymptotically stable and the state-values tend to be consistent over time, i.e., $\lim_{t \to +\infty} x(t) \in \mathbb{R}.$

Proof: Construct the unbounded Lyapunov function candidate

$$\mathcal{J}(x(t)) = \frac{1}{2} x(t)^T \mathcal{L}x(t),$$

(16)

which satisfies that $\dot{\mathcal{J}}(x(t)) = 0$ when and only when $e(t) = \mathcal{L}x(t) = 0.$ The time-derivative of $\mathcal{J}(x(t))$ is derived as follows: $[\dot{\mathcal{J}}(x(t))]$ is abbreviated as $\mathcal{J}$ for presentation:

$$\frac{d\mathcal{J}}{dt} = \sum_{i=1}^{n} \frac{\partial \mathcal{J}}{\partial x_i} \cdot \dot{x}_i = -\gamma \sum_{i=1}^{n} e_i \text{sign}(e_i) \phi(|e_i|) \leq 0.$$

(17)

If $e \neq 0,$ $\frac{d\mathcal{J}}{dt} < 0$ holds. Thus, the dynamic system is asymptotically stable and $e(t)$ converges to 0. Finally, since $\mathcal{G}$ is connected, Lemmas 3 and 4 hold. The conclusion $\lim_{t \to +\infty} x(t) \in \mathbb{I}$ can be drawn from two conditions $\lim_{t \to +\infty} e(t) = 0$ and $\lambda_2(\mathcal{G}) \neq 0.$ Thus, the proof is completed.

Remark 3: Although the dynamic system (13) includes the discontinuous function $\text{sign}(\cdot),$ each whole term $\phi(|e_i|) \cdot \text{sign}(e_i)$ $(i = 1, 2, \ldots, n)$ in activation function array $\Phi(e) = [\phi(|e_1|) \cdot \text{sign}(e_1), \phi(|e_2|) \cdot \text{sign}(e_2), \ldots, \phi(|e_n|) \cdot \text{sign}(e_n)]^T$ is a continuously-differentiable function. Thus, the activation function array $\Phi(e)$ is a group of continuously-differentiable functions. To show this property more clearly, the specific proof process about the differentiability of $\phi(|e_i|) \cdot \text{sign}(e_i)$ is presented as below.

Let $f(x) = \phi(|x|) \cdot \text{sign}(x).$ Since $\phi(\cdot)$ is a smooth monotone increasing odd function, it is easy to prove that $f(x)$ is a continuously-differentiable function, when $x > 0$ or $x < 0.$ Next, we will prove that $f(x)$ is also differentiable at $x = 0.$ First, let us solve the right-hand limit of $f(x)$ at $x = 0$:

$$\lim_{x \to 0^+} \frac{f(x) - f(0)}{x - 0} = \lim_{x \to 0^+} \frac{\phi(|x|) \cdot \text{sign}(x) - \phi(0) \cdot \text{sign}(0)}{x - 0} = \lim_{x \to 0^+} \frac{\phi(|x|)}{x}.$$

(18)

Then, we can solve the left-hand limit of $f(x)$ at $x = 0$:

$$\lim_{x \to 0^+} \frac{f(x) - f(0)}{x - 0} = \lim_{x \to 0^+} \frac{\phi(|x|) \cdot \text{sign}(x) - \phi(0) \cdot \text{sign}(0)}{x - 0} = \lim_{x \to 0^+} \frac{-\phi(|x|)}{x} = \lim_{x \to 0^+} \frac{\phi(|x|)}{x}.$$

(19)

Based on the above result, we can conclude that $f(x) = \phi(|x|) \cdot \text{sign}(x)$ is a continuously-differentiable function. Thus, we can further conclude that $\Phi(\cdot)$ is differentiable everywhere and the dynamic system (13) is a smooth Lipschitz continuous-time system.

Definition 2: For any DGNN model $\dot{x} = \mu(x(t)),$ there exists its corresponding state-error vector $e(t) = \mathcal{L}x(t).$ If the settling-time function (STF) [7], [19] $T(e_0)$ defined as

$$T(e_0) = \inf \left\{ T \in [0, +\infty) \mid \|e(t)||_{\mathcal{G} \geq T} = 0 \right\},$$

(20)

satisfies finiteness for any given initial state-error vector $e_0 = \mathcal{L}x(0),$ i.e., $T(e_0) < +\infty,$ then the DGNN model is called finite-time consensus [19].

Definition 3: For any DGNN model $\dot{x} = \mu(x(t)),$ if it satisfies Definition 2, and the STF is bounded on $\mathbb{R}^n,$ i.e., there exists a supremum $T_\infty := \lim_{\|e_0\| \to \infty} T(e_0) < +\infty,$ then it is called predefined-time consensus [7].

Definition 4: For any dynamics shaped like $\dot{x}_i = -\gamma \phi(|e_i|) \text{sign}(e_i),$ where $e_i = \sum_{j=1}^{n} a_{ij} (x_i - x_j),$ $\gamma$ is a constant and $\phi(\cdot)$ is an activation function, if it satisfies Definition 3, then $\phi(\cdot)$ is called PTC function.

Theorem 2: For DGNN model (12), if the expression $\psi(e) = e \cdot \phi(e)$ is a convex function, then there exists a radially unbounded Lyapunov function $\mathcal{J}(x(t))$ that satisfies $\mathcal{J}(x(t)) = 0$ if and only if $x \in \mathbb{I} = \{ y \in \mathbb{R}^n \mid y =$
\[ c \cdot (1, 1, ..., 1)^T, c \in \mathbb{R}, \] i.e., \( e(t) = \mathcal{L}x(t) = 0 \), such that the inequality
\[ \frac{d}{dt} J(x) \leq -\lambda \phi(J) \]
holds.

**Proof:** Based on the zoom method in Ref. [7], a more flexible proof method is presented as follow. For DGNN model (12), the equilibrium set is given by \( \mathcal{I} = \{ y \in \mathbb{R}^n | y = c \cdot (1, 1, ..., 1)^T, c \in \mathbb{R} \} \), i.e., at the equilibrium points, consensus is achieved. Construct the radially unbounded Lyapunov function candidate:
\[ J(x(t)) = c \sqrt{x(t)^T \mathcal{L} x(t)}, \] (21)
which satisfies that \( J(x(t)) = 0 \) if and only if \( x \in \mathbb{I} = \{ y \in \mathbb{R}^n | y = c \cdot (1, 1, ..., 1)^T, c \in \mathbb{R} \} \), i.e., \( e(t) = \mathcal{L}x(t) = 0 \). The time-derivative of \( J(x(t)) \) is
\[
\frac{d}{dt} J = \frac{c}{\sqrt{x(t)^T \mathcal{L} x(t)}} \sum_{i=1}^{n} \frac{\partial J}{\partial x_i} \frac{dx_i}{dt} \\
= -\frac{c^2 \gamma}{J} \sum_{i=1}^{n} \psi(|e_i|) \\
= -\frac{c^2 \gamma}{J} \sum_{i=1}^{n} \psi(|e_i|).
\]
(22)
Since \( \psi(\cdot) \) is a convex function, it satisfies Jensen inequality:
\[
\psi\left(\frac{\sum_{i=1}^{n} \mu_i x_i}{n}\right) \leq \frac{\sum_{i=1}^{n} \mu_i \psi(x_i)}{n}, \]
s.t. \( \sum_{i=1}^{n} \mu_i = 1 \).
(23)
The following inequality is obtained:
\[
\frac{d}{dt} J = -\frac{c^2 \gamma}{J} \sum_{i=1}^{n} \psi(|e_i|) \\
\leq -\frac{c^2 \gamma}{J} \psi\left(\frac{1}{n} \sum_{i=1}^{n} |e_i|\right) \\
\leq -\frac{c^2 \gamma}{J} \psi\left(\frac{1}{n} \|e\|_2\right).
\]
(24)
According to Lemma 4, we can obtain the following inequality:
\[
\|e\|_2^2 = \sqrt{x(t)^T \mathcal{L}^T \mathcal{L} x(t)} \\
\geq \sqrt{\frac{\lambda_2(\mathcal{L})}{c} \cdot \lambda_2(\mathcal{L})^T \mathcal{L} x(t)} \\
= \frac{\sqrt{\lambda_2(\mathcal{L})}}{c} \cdot \mathcal{L} x(t).
\]
(25)
Then, Eq. (24) can be further derived as
\[
\frac{d}{dt} J \leq -\frac{c^2 \gamma}{J} \psi\left(\frac{1}{n} \|e\|_2\right) \\
\leq -\frac{c^2 \gamma}{J} \psi\left(\frac{\lambda_2(\mathcal{L})}{cn}\right) \cdot J.
\]
(26)
Let \( c = \sqrt{\lambda_2(\mathcal{L})/n} \), and we have
\[
\frac{d}{dt} J \leq -\frac{c^2 \gamma}{J} \psi(J) \\
= -\frac{\lambda_2(\mathcal{L})}{n} \psi(J) \\
= -\frac{\lambda_2(\mathcal{L})}{n} \phi(J).
\]
(27)
Thus, the proof is completed. □

**Lemma 5:** (Polyakov’s Theorem [17]) For any DGNN model \( \dot{x} = \mu(x(t)) \), if there exists a continuous radially unbounded function \( J(x(t)) \) which satisfies that 1) \( J(x(t)) = 0 \Rightarrow x \in \mathbb{I} \), and 2) the inequality
\[
\frac{d}{dt} J \leq -(aJ^p + bJ^q)^k, \\
\text{s.t. } a, b, p, q > 0, pk < 1, qk > 1,
\]
holds for any \( x \in \mathbb{R}^n \), then the state vector \( x(t) \) of the dynamic system converges to the steady-state \( x \in \mathbb{I} \) in a predefined time \( T_\infty \) that satisfies:
\[
T_\infty \leq \left( \frac{1}{a(1-p)} + \frac{1}{b(q-1)} \right)^k \cdot \|x(0)\|_2.
\]
(29)

**Theorem 3:** For DGNN model (12), if the activation function satisfies
(i) \( \psi(e) = e \cdot \phi(e) \) is a convex function;
(ii) \( \phi(z) \geq az^p + bz^q \) for \( z \geq 0, \exists a > 0, b > 0, 0 < p < 1, q > 1 \);
then, \( \phi(\cdot) \) is called PTC function.

**Proof:** For any activation function \( \phi(\cdot) \) that satisfies condition (i), according to Theorem 2, there exists a radially unbounded Lyapunov function
\[
J(x(t)) = \frac{\sqrt{\lambda_2(\mathcal{L})}}{c} \cdot \sqrt{x(t)^T \mathcal{L} x(t)}
\]
(30)
such that following inequality
\[
\frac{d}{dt} J \leq -\lambda \phi(J)
\]
(31)
holds. Furthermore, the inequality \( \frac{d}{dt} \|e\|_2 \leq -\lambda(aJ^p + bJ^q) \) holds because condition (ii) is satisfied. According to Lemma 5, the DGNN model converges to the steady-state in a predefined time
\[
T_\infty \leq \left( \frac{1}{a(1-p)} + \frac{1}{b(q-1)} \right)^k, \forall x(0) \in \mathbb{R}^n.
\]
(32)
and \( \phi(\cdot) \) is a PTC function. Thus, the proof is completed. □

**B. Activation Functions**

**Theorem 4:** The following functions defined in \([0, +\infty)\) are PTC functions:
(i) Sinh-Power activation function (SPAF):
\[
\phi(z) = \sinh(cz^p), c > 0, p \in (0, 1);
\]
(ii) Exponential-Power activation function (EPAF):
\[
\phi(z) = \exp(cz^p) - 1, c > 0, p \in (0, 1);
\]
(iii) Exponential-Bi-Power activation function (EBPAF):
\[
\phi(z) = \exp(cz^r)z^p, r > 0, p \in (0, 1).
\]
Proof: These three functions are obviously continuous monotone increasing functions and satisfy \(\phi(0) = 0\). Next, we prove that they satisfy the sufficient conditions in Theorem 3.

1. The second-order derivative of the function (i) \(\psi(z) = z\phi(z) = z\sinh(czp)\) can be calculated as follows:

\[
\frac{d^2}{dz^2}\psi(z) = c(p+1)z^{p-1}\cosh(czp) + c^2p^2z^{2(p-1)}\sinh(czp).
\]

Thus, \(\psi(z)\) is a convex function because the inequality \(\frac{d^2}{dz^2}\psi(z) \leq 0\) holds in its domain of definition.

Considering the power series expansion of function \(\phi(\cdot)\):

\[
\phi(z) = \sinh(czp) = \sum_{k=1}^{\infty} \frac{c^{2k-1}}{(2k-1)!} z^{(2k-1)p},
\]

there are \(a = c > 0, b = \frac{c^{2m-1}}{(2m-1)!} > 0, p \in (0, 1)\) and \(q = (2m-1)p > 1\) with \(\exists m = \lceil 1/p + 1 \rceil \in \mathbb{N}\) such that \(\phi(z) \geq az^p + b z^q\).

2. The second-order derivative of the function (ii) \(\psi(z) = z\phi(z) = z(\exp(czp) - 1)\) can be calculated as follows:

\[
\frac{d^2}{dz^2}\psi(z) = c \exp(czp) p z^{p-1}(1 + p + cpz^p).
\]

Thus, \(\psi(z)\) is a convex function because the inequality \(\frac{d^2}{dz^2}\psi(z) \leq 0\) holds in its domain of definition.

Considering the power series expansion of function \(\phi(\cdot)\):

\[
\phi(z) = \exp(czp) - 1 = \sum_{k=1}^{\infty} \frac{c^k}{k!} z^{kp},
\]

there are \(a = c > 0, b = \frac{c^m}{m!} > 0, p \in (0, 1)\) and \(q = mp > 1\) with \(\exists m = \lceil 1/p \rceil \in \mathbb{N}\) such that \(\phi(z) \geq az^p + b z^q\).

3. The second-order derivative of the function (iii) \(\psi(z) = z\phi(z) = z(\exp(cz^r) z^{p-1})\) can be calculated as follows:

\[
\frac{d^2}{dz^2}\psi(z) = \exp(cz^r) z^{p-1} \cdot (p + p^2 + crz^r + 2cprz^r + cr^2z^{2r} + c^2r^2z^{2r}).
\]

Thus, \(\psi(z)\) is a convex function because the inequality \(\frac{d^2}{dz^2}\psi(z) \leq 0\) holds in its domain of definition.

Considering the power series expansion of function \(\phi(\cdot)\):

\[
\phi(z) = \exp(cz^r) z^p = \sum_{k=0}^{\infty} \frac{c^k}{k!} z^{kr+p},
\]

there are \(a = 1 > 0, b = \frac{c^m}{m!} > 0, p \in (0, 1)\) and \(q = m + p > 1\) with \(\exists m = \lceil 1/p \rceil \in \mathbb{N}\) such that \(\phi(z) \geq az^p + b z^q\), where the notation \(\lceil \cdot \rceil\) is for the ceil operator.

In summary, the above three functions all meet two conditions of Theorem 3, so they are PTC functions.

V. EXAMPLES AND NUMERICAL VERIFICATION

In this section, the validity and PTC performance of DGNN based on three newly proposed AFs are shown through numerical simulation results. In addition, a practical example about the directional consensus is presented. Finally, a connectivity-testing example verifies that the connectivity of the communication network affects the convergence speed.

A. Numerical Simulation of Three Novel Activation Functions

Example 1: Given a communication network \(G_1\) composed of \(n = 8\) agents, as shown in Fig. 1, it possesses the cardinality of the edge set given by \(|\mathcal{E}(G_1)| = 7\) and an algebraic connectivity of \(\lambda_2(L(G_1)) = 0.15\). Let the initial condition be \(x_0 = [-2.82, -2.49, 3.93, 2.03, 0.56, -3.16, -2.88, -4.23]^{T}\). Then, under the control of DGNN model (12) using the activation function \(\phi(z) = \sinh(czp)\), the convergence of the 8 states is shown in Fig. 2 for the multi-agent system with communication network \(G_1\). According to Theorem 4(i), \(\phi(\cdot)\) is a PTC function, satisfying Lemma 5 with \(a = 1, b = \frac{1}{8}, p = 0.5, q = 1.5\) and \(k = 1\), so the predefined-time \(T_{\infty}^{(p)}\) is \(1\) and an algebraic...
connectivity of $\lambda_2(L(G_2)) = 1.23$. Let the initial condition be $x_0 = [4.14, 2.07, 0.58, -1.87, -3.34, 1.22, 4.88, -3.30]^T$. Then, under the control of DGNN model (12) using the activation function $\phi(z) = \exp(cz^p) - 1$ with $c = 1, p = 0.5, \gamma = 1$, the convergence of the 8 states is shown in Fig. 4 for the multi-agent system with communication network $G_2$. According to Theorem 4(ii), $\phi(\cdot)$ is a PTC function, satisfying Lemma 5 with $a = 1, b = \frac{1}{6}, p = 0.5, q = 1.5$ and $k = 1$, so the predefined-time $T_{\infty}^{\text{epb}}(\gamma) = 14$.

**Example 3:** Given a communication network $G_3$ composed of $n = 8$ agents, as shown in Fig. 5, it possesses the cardinality of the edge set given by $|E(G_3)| = 20$ and an algebraic connectivity of $\lambda_2(L(G_3)) = 2.63$. Let the initial condition be $x_0 = [-2.42, -1.03, -4.26, 1.84, -0.98, 4.83, -0.98, 1.21]^T$. Then, under the control of DGNN model (12) using the activation function $\phi(z) = \exp(cz^r)z^p$ with $c = 1, r = 0.5, p = 0.5, \gamma = 1$, the convergence of the 8 states is shown in Fig. 6 for the multi-agent system with communication network $G_3$. According to Theorem 4(iii), $\phi(\cdot)$ is a PTC function, satisfying Lemma 5 with $a = 1, b = \frac{1}{2}, p = 0.5, q = 1.5$ and $k = 1$, so the predefined-time $T_{\infty}^{\text{epb}}(\gamma) = 6$.

Notice that the actual convergence times for the consistent states of Examples 1, 2 and 3 are much less than the $T_{\infty}$ estimated by Theorem 3. There are at least three facts that contribute to this phenomenon:

1) $T_{\infty}$ is a rough estimate, slightly larger than the least upper bound;
2) the initial state starts with a finite value instead of infinity;
3) the convergence time of the consistent state is related to the algebraic connectivity of the network. That is, the greater the algebraic connectivity means the smaller the convergence time [37].
Remark 4: For the first point, we can, in fact, obtain a more accurate upper bound of settling-time by symbolic computation using the Wolfram Mathematica Function \texttt{Integrate}[f, x, x_{\text{min}}, x_{\text{max}}]^*, for example, $T_{\infty}^{(sp)} = \frac{\pi}{2} \approx 4.9348$, $T_{\infty}^{(ep)} = \frac{\pi^2}{3} \approx 3.2889$, and $T_{\infty}^{(ebp)} = 2.0$.

Remark 5: For the third point, a strict connectivity-testing experiment will be conducted in the next subsection to illustrate how the connectivity of communication networks affects the convergence rate.

B. Practical Application of DGNN

Example 4: Given a system consisting of seven agents shown in Fig. 7, their initial position coordinates and direction angles are shown in Table II. To simplify the experiment, the directional consensus is the object of study in this case, while the communication network $\mathcal{G}_4$ is considered to be static and the motion speeds of all agents are constant and set at 1m/s. In addition, it should be noted that the values of the direction are represented by the angle between the direction of motion and the horizontal axis. The communication network $\mathcal{G}_4$ composed of $n = 7$ agents, as shown in Fig. 7(a), it possesses the cardinality of the edge set given by $|\mathcal{E}(\mathcal{G}_4)| = 12$ and an algebraic connectivity of $\lambda_2(\mathcal{L}(\mathcal{G}_4)) = 1.49$. Let the initial state vector be $x_0 = [45.0, 60.0, 30.0, 55.0, 57.0, 35.0, 15.0]^T$. Then, under the control of DGNN model (12) using following

Table II: Initial state of the system

<table>
<thead>
<tr>
<th>Agent</th>
<th>Position</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>A_1</td>
<td>(6.3640, 6.3640)</td>
<td>45.0</td>
</tr>
<tr>
<td>A_2</td>
<td>(2.1213, 6.3640)</td>
<td>60.0</td>
</tr>
<tr>
<td>A_3</td>
<td>(6.3640, 2.1213)</td>
<td>30.0</td>
</tr>
<tr>
<td>A_4</td>
<td>(-1.4142, 5.6569)</td>
<td>55.0</td>
</tr>
<tr>
<td>A_5</td>
<td>(-0.3536, 1.7678)</td>
<td>57.0</td>
</tr>
<tr>
<td>A_6</td>
<td>(1.7678, -0.3536)</td>
<td>35.0</td>
</tr>
<tr>
<td>A_7</td>
<td>(5.6569, -1.4142)</td>
<td>15.0</td>
</tr>
</tbody>
</table>

six activation functions:
(i) SPAF: $\phi(z) = \sinh(c z^p)$,
(ii) EPAP: $\phi(z) = \exp(c z^p) - 1$,
(iii) EBPAF: $\phi(z) = \exp(c z^p)z^r$,$r = 0.5$,
(iv) Linear activation function (LAF): $\phi(z) = z$,
(v) Power activation function (PAF): $\phi(z) = z^p$,
(vi) Bi-Power activation function (BPAF): $\phi(z) = (z^p + z^q)/2$, $q = 1.5$.
where the common parameters are set to $c = 1$, $p = 0.5$ and $\gamma = 1$. The standard deviations $\sigma(t)$ of the $n$ state values at each time $t$ is shown in Fig. 8(a) for the multi-agent system with communication network $G_4$, where $\sigma(t)$ is given by

$$\sigma(t) = \left(\frac{1}{n} \sum_{i=1}^{n} (x_i(t) - \frac{1}{n} \sum_{i=1}^{n} x_i(t))^2\right)^{1/2}.$$

**Remark 6:** The simulation result of Example 4 is shown in Fig. 8(a). The figure shows that 1) DGNN using these six activation functions can eventually stabilize the state value to a consistent level; 2) Compared with Linear or Power submodels, DGNN models activated by PTC functions can maintain a faster settling-rate at the extremes of system error tendency to infinity and tending to zero; 3) Three DGNN models using novel AFs make the agents of the system reach consensus (alignment of direction) in a predefined time.

**Remark 7:** For continuous-time GNN models with PTC, it is meaningless to compare their overall settling-time, since any small settling-time can be achieved by adjusting the parameters. In the discrete-time models it is necessary to discuss because of the precision constraints.

**Example 5:** This is a connectivity-testing example from the four-node network. As shown in Fig. 7(b), only the communication network changed in the system (with weaker connectivity), while the initial position and direction remain unchanged. The network $G_3$ possesses the cardinality of the edge set given by $|E(G_3)| = 7$ and an algebraic connectivity of $\lambda_2(L(G_4)) = 0.75$.

**Remark 8:** The simulation result of Example 5 is shown in Fig. 8(b). Compared with Fig. 8(a), the settling-time of all activation functions is increased, which means that the less connectivity, the slower the system converges. Finally, it is worth noting that three more stringent upper bounds on settling-time calculated by Mathematica in the previous subsection are verified in Examples 4 and 5.

**VI. Conclusions**

In this paper, based on the gradient neural network (GNN) framework, a novel distributed GNN (DGNN) model is proposed to solve consensus problems of multi-agent systems. This method unifies the design idea of the gradient neural network and the zeroing neural network. Therefore, existing activation functions for the DGNN model still have similar properties, such as asymptotic stability, exponential convergence, finite-time convergence, and predefined-time convergence. Three new activation functions are proposed to make the DGNN model have the strongest predefined-time convergence among the convergence types. In addition, the new activation functions are compared with the existing activation functions in a practical directional consensus problem. The efficiency and superiority of the DGNN model for solving consensus problems of multi-agent systems are thus verified. Besides, a connectivity-testing example is presented to verify the effectiveness of the DGNN model for different communication networks. It is worth mentioning that this paper does not fully exploit the potentialities of the DGNN model because the same activation functions are used for all agents, and the stability of a multi-agent system with two nodes should be enhanced from the theoretical aspect. Therefore, these two directions will be our future research work.

**References**
