

Article

Fixed-Time Distributed Optimization Without Matrix Inversion via Zeroing Neural Networks

Xintian Liang¹, Luke Li^{2,*} and Qintao Gan²¹ School of Science, Hebei University of Technology, Tianjin 300401, China² Shijiazhuang Campus, Army Engineering University of PLA, Shijiazhuang 050003, China

* Correspondence: luke0258@aeu.edu.cn

Abstract

This paper focuses on the fixed-time distributed optimization problem with time-varying cost functions in multi-agent systems (MASs). For the scenario where all agents share the same Hessian matrix, a control framework for minimizing the global cost function is proposed based on the zero-gradient-sum (ZGS) algorithm. Specifically, integral sliding mode is employed to achieve the initial optimality of local objective functions, thereby satisfying the initial conditions required by the ZGS algorithm. Furthermore, for the case with heterogeneous Hessian matrices among agents, a distributed optimization algorithm integrating distributed gradient estimation and the prediction-correction method (PCM) is designed. Specifically, the distributed gradient estimation enables each agent to reconstruct the global objective gradient information of the system within a fixed time. Both proposed algorithms can achieve fixed-time convergence, and the inverse of the Hessian matrix is approximated online using a zeroing neural network (ZNN), which significantly improves computational efficiency. Finally, numerical simulations are conducted based on the charge balance problem in smart grids to verify the effectiveness and superiority of the proposed algorithms.

Keywords: time-varying optimization; sliding mode control; fixed-time stability; zeroing neural networks

MSC: 34H05; 49N05; 93A16

1. Introduction

As a vital research area in networked systems and cooperative control, distributed cooperative optimization for multi-agent systems (MASs) has attracted growing research attention [1]. Without relying on any centralized coordinator, each agent interacts only with its neighboring peers to collaboratively minimize a time-varying global cost function. Relevant theories and algorithms have been widely applied in unmanned aerial vehicle (UAV) swarm operations, dynamic scheduling of integrated energy systems [2], and large-scale wireless sensor networks [3]. In contrast to traditional centralized optimization approaches, distributed schemes offer strong fault tolerance against central node failures, reduced communication overhead, and excellent scalability, making them highly suitable for large-scale dynamic networked systems [4].

Current distributed optimization algorithms are predominantly developed for time-invariant objective functions, where the global cost function remains unchanged throughout



Academic Editor: Jonathan Blackledge

Received: 17 June 2026

Revised: 17 July 2026

Accepted: 17 July 2026

Published: 24 July 2026

Copyright: © 2026 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.

the optimization process [5–8]. Within such static settings, agents iteratively share information with neighbors and update their states, eventually converging to a unified optimal solution [9]. A variety of effective approaches have been developed over the years, such as distributed (sub)gradient descent [10,11], the zero-gradient-sum (ZGS) algorithm [9], alternating direction method of multipliers [12], and push-sum algorithms [13], and substantial progress has been made in analyzing their convergence performance and convergence rates. Nevertheless, these existing methods are inherently tailored for static cost functions and cannot be directly applied to dynamic scenarios. In practical engineering tasks including real-time target tracking by UAV swarms [14,15], dynamic resource allocation [16], and distributed estimation of time-varying signals via mobile sensor networks, the global cost function evolves continuously over time. For these cases, algorithms are required to not only approach the instantaneous optimal solution at each time instant but also track the time-varying optimal trajectory in real time. To address this practical demand, research on distributed time-varying optimization has attracted extensive attention [17–19]. Its primary objective is to construct distributed strategies that enable multi-agent systems to stably follow the dynamic optimal trajectory, rather than achieving asymptotic convergence to a stationary optimal point.

In distributed time-varying optimization for multi-agent systems, the convergence time serves as a critical metric determining the system's real-time performance, which directly impacts the safety and reliability of engineering applications. Compared with asymptotically convergent algorithms, finite-time convergence methods offer faster response [20]. However, their convergence time bounds typically depend on the initial states of the agents [21]. When the initial states deviate greatly from the optimal trajectory, the convergence time may increase significantly, thereby failing to meet the strict real-time demands of applications such as UAV obstacle avoidance and real-time energy dispatch. As a special case of finite-time convergence, fixed-time convergence provides a uniform convergence time bound that is independent of initial states, ensuring the system's real-time responsiveness under uncertain initial conditions [22]. This desirable property has recently made fixed-time convergence a research hotspot in the field of distributed time-varying optimization. Motivated by this, researchers have developed various fixed-time distributed optimization algorithms. For example, Wu et al. investigated a fixed-time distributed time-varying optimization problem with affine formation constraints [23]. Yang et al. addressed the problem of global time-varying optimal consensus for UAV swarms under state-coupled inequality constraints [24]. However, these methods essentially only guarantee fixed-time consensus, whereas the optimization convergence remains asymptotic. As a result, in dynamic scenarios such as target tracking, the target can easily escape from the tracking agents. Unlike the above approaches, the authors of [25] established an intrinsic connection between zeroing neural networks (ZNNs) and time-varying optimization, and they proposed a novel fixed-time distributed time-varying optimization algorithm. Nevertheless, its two-stage structure leads to conservative estimation of the convergence time. Specifically, the switching instant is predetermined by the worst-case upper bound of the local optimization rather than the actual convergence time, which is often significantly shorter, and consequently incurs substantial waiting redundancy. Meanwhile, the work in [19] achieved simultaneous fixed-time consensus and optimization for multi-agent systems based on the ZGS algorithm. However, this approach relies on the restrictive assumption that all agents share identical Hessian matrices, which limits its applicability.

On the other hand, Hessian matrix inversion is a key step in solving time-varying optimization problems and improving convergence accuracy. However, the computational complexity of Hessian matrix inversion exhibits a cubic scaling with the dimension

of decision variables for each agent. With the increasing number of agents and system dimensionality, the overall computational burden rises sharply, which degrades the solution efficiency and real-time capability, and may even result in algorithm breakdown in extreme scenarios [26]. Furthermore, cost functions are inherently dynamic in time-varying environments, and the corresponding Hessian matrices evolve continuously over time. In this case, matrix inversion has to be executed repeatedly at every time step, which further increases computational and communication overheads. This drawback greatly hinders the practical implementation of optimization algorithms based on explicit Hessian inversion. Therefore, circumventing the computational bottleneck induced by explicit Hessian inversion while maintaining favorable tracking accuracy and fixed-time convergence performance has become a crucial issue for the design of distributed time-varying optimization algorithms.

In view of the existing research limitations, this work presents two novel ZNN-based inverse-free fixed-time distributed time-varying optimization methods for multi-agent systems with identical and distinct Hessian matrices. The key merits of the proposed approaches are demonstrated as follows:

- By integrating the ZGS algorithm with sliding mode control, a time-varying distributed optimization framework is developed for the case where all agents share identical Hessian matrices. The presented framework extends the original ZGS algorithm to time-varying scenarios and removes constraints associated with system local optima [9,27,28].
- A time-varying distributed optimization framework is developed for heterogeneous Hessian matrices using a prediction-correction method (PCM) and a distributed estimator. Compared with [24,29,30], this framework eliminates the need for strong convexity of local cost functions and boundedness of their partial derivatives, thereby significantly relaxing algorithmic restrictions.
- A ZNN-based matrix approximation method is proposed to avoid explicit Hessian inversion, effectively reducing the computational burden of multi-agent systems. Moreover, the proposed method features fixed-time convergence, allowing consensus and optimization to be reached simultaneously within a fixed time.

2. Preliminaries and Problem Formulation

2.1. Preliminaries

Notations: R^m , R , and R^+ denote the sets of m -dimensional real vectors, real numbers, and non-negative real numbers, respectively. Let 1_m (0_m) denote the m -dimensional column vectors of all ones (zeros); $\text{sign}(\cdot)$ is the signum function. For $x = [x_1, \dots, x_m]^T \in R^m$, x^T represents its transpose, while $\text{sign}(x) = (\text{sign}(x_1), \dots, \text{sign}(x_m))$ and $\|x\|_1$, $\|x\|$ denote its 1-norm and Euclidean norm, respectively. $\mathcal{A} \otimes \mathcal{B}$ represents the Kronecker product of matrices $\mathcal{A}$ and $\mathcal{B}$.

Graph Theory: Denote a graph as $\mathcal{G} = \{\varepsilon, \nu, \mathcal{A}\}$, where $\nu = \{\nu_1, \dots, \nu_N\}$ is the set of nodes, $\varepsilon \in \nu \times \nu$ is the set of edges, and $A = a_{ij} \in R^{N \times N}$ is the corresponding weighted adjacency matrix with $a_{ij} > 0$ if $(j, i) \in \varepsilon$ and $a_{ij} = 0$ otherwise. We assume that $a_{ii} = 0$. An edge $(\nu_i, \nu_j) \in \varepsilon$ indicates that the information of node ν_i is accessible to node ν_j . $\mathcal{N}_i = \{j \in \nu | (j, i) \in \varepsilon\}$ represents the neighbor set of agent i . The graph $\mathcal{G}$ is undirected if and only if $(j, i) \in \varepsilon$ is equivalent to $(i, j) \in \varepsilon$, and connected if there exists at least a path from every node to every other node. The Laplacian matrix $L = l_{ij} \in R^{N \times N}$ of $\mathcal{G}$ is defined as $l_{ij} = -a_{ij}$, $l_{ii} = \sum_{j=1, j \neq i}^N a_{ij}$.

Convex Analysis: A function $f : R^m \rightarrow R$ is convex if it is defined on convex set $\mathcal{D}$ and for $\forall x, y \in \mathcal{D}$, $0 \leq a \leq 1$, there exists $f(ax + (1-a)y) \leq af(x) + (1-a)f(y)$. For $\forall x, y \in \mathcal{D}$, $\mu > 0$, the continuously differentiable convex function f is μ -strongly convex if

$$(\nabla f(y) - \nabla f(x))^T(y - x) \geq \mu \|y - x\|^2. \quad (1)$$

2.2. Problem Formulation

Consider an MAS with N agents:

$$\dot{x}_i(t) = u_i(t), \quad i = 1, 2, \dots, N, \quad (2)$$

where $x_i(t) = (x_{i1}(t), \dots, x_{im}(t))^T \in R^m$ and $u_i(t) = (u_{i1}(t), \dots, u_{im}(t))^T \in R^m$ are the i th agent state and control input, respectively.

The objective is to formulate some control protocols $u_i(t)$ that enable all agents to cooperatively solve the following optimization problem:

$$\text{minimize } f(x, t) = \sum_{i=1}^N f_i(x_i, t) \text{ s.t. } x_i(t) = x_j(t), \quad (3)$$

where $i, j \in \nu$, $x = (x_1(t), \dots, x_N(t))^T \in R^{Nm}$, $f_i(x_i, t)$ and $f(x, t)$ are agent's local time-varying cost function and global cost function, respectively.

Assumption 1. The interaction graph $\mathcal{G}$ is undirected and connected.

Remark 1. The undirected connected graph assumption is technically necessary within the present analytical framework, as the symmetry of the Laplacian plays a key role in the convergence proof via Lyapunov stability analysis. Extensions to directed graph settings are deferred to future work.

Assumption 2 ([19]). The function f_i is twice differentiable, μ -strongly convex and l Lipschitz gradient-continuous with respect to x for $\forall t \geq 0$, where $\mu, l > 0$.

Assumption 3 ([19]). The Hessian matrices of all local cost functions are identical for any state and the time partial derivatives of their gradients have an upper bound, i.e., $\nabla^2 f_i(x_i, t) = \nabla^2 f_j(x_j, t)$, $\|\frac{\partial}{\partial t} \nabla f_i(x_i, t)\|_\infty \leq \hbar$, $\forall i, j \in \nu$ and $t \geq 0$.

Remark 2. Assumption 3 is a relatively strong condition, but it is commonly adopted in the literature on distributed time-varying optimization and is naturally satisfied in a number of practical scenarios. For instance, in the context of energy optimization, quadratic functions of the form $f_i(x_i, t) = \frac{\gamma}{2} x_i^2 + \theta_i(t)x_i + c_i(t)$ are commonly used, and their Hessian $\nabla^2 f_i = \gamma$ is identical across all agents. In target tracking applications, the cost function $f_i(\omega_i, t) = \|\omega_i - m(t)\|_2^2$ has Hessian $\nabla^2 f_i = 2I$, which is also identical across all agents.

Assumption 4. The global cost function f is convex. For each i , the local cost function f_i is twice continuously differentiable with respect to x_i and first-order continuously differentiable with respect to t . Moreover, the Hessian matrices are invertible and not necessarily identical.

Lemma 1 ([31]). For a connected and undirected graph $\mathcal{G} = \{\epsilon, \nu, \mathcal{A}\}$, the eigenvalues of its positive semi-definite Laplacian matrix L are $0 \leq \lambda_{\min}(L) \leq \dots \leq \lambda_{\max}(L)$, where $\lambda_{\min}(L)$ is the algebraic connectivity of graph $\mathcal{G}$. Moreover, if $1_m^T x = 0$ for $x \in \mathbb{R}^N$, then $x^T L x \geq \lambda_{\min}(L) x^T x$.

Lemma 2 ([32]). If $\mu_m \geq 0$, $m = 1, 2, \dots, n$ and $0 < \varsigma < \zeta$, then $(\sum_{m=1}^n \mu_m^{\zeta})^{\frac{1}{\zeta}} \leq (\sum_{m=1}^n \mu_m^{\varsigma})^{\frac{1}{\varsigma}} \leq n^{\frac{1}{\varsigma} - \frac{1}{\zeta}} (\sum_{m=1}^n \mu_m^{\zeta})^{\frac{1}{\zeta}}$.

Lemma 3 ([33]). Consider the system $\dot{x}(t) = f(x(t))$ with $f(0_m) = 0_m$. If there is a continuously differentiable and radially unbounded function $V : \mathbb{R}^m \rightarrow \mathbb{R}^+$ satisfies $dV/dt \leq -\lambda_1 V^p - \lambda_2 V^q$ with $\lambda_1, \lambda_2 > 0, 0 < p < 1 < q$, then the origin of the system is fixed-time stable and the settling time T is estimated by $T = 1/(\lambda_1(1-p)) + 1/(\lambda_2(q-1))$.

3. ZNN Models

In this section, a fixed-time ZNN model with time-varying parameters is introduced, which will serve as the basis for the design of the distributed optimization algorithms in the following sections.

Consider a vector-type or matrix-type error function $e(t)$, whose dynamic can be described as follows:

$$\dot{e}(t) = -\Gamma(t)\Xi(e(t)), \quad (4)$$

where $\Gamma(\cdot)$ is a positive parameter, and $\Xi(\cdot)$ is the activation function (AF) array. The dynamic system (4) ensures that the error function $e(t)$ converges to zero if $\Xi(\cdot)$ is a monotonically increasing odd function [34–36]. It should be noted that the parameter $\Gamma(t)$ and the AF $\Xi(\cdot)$ are two important factors for the convergent performance of the neural network system (4). In this paper, the following parameter is considered:

- Time-Varying Parameter (TVP):

$$\Gamma_1(t) = \exp(\vartheta_1 t + \vartheta_0), \quad (5)$$

where the parameters $\vartheta_0, \vartheta_1 \geq 0$.

To converge the error function $e(t)$ in fixed time, the following AF will be employed throughout the article:

- Sign-Bi-Power Activation Function (SBP):

$$\Xi_1(\iota) = (\lambda_1 |\iota|^p + \lambda_2 |\iota|^q + \lambda_3) \text{sign}(\iota), \quad (6)$$

where the parameters $\lambda_{1-3} > 0, 0 < p < 1 < q$.

Remark 3. In this section, a ZNN model is introduced for solving the optimization problem. The model employs the Sign-bi-power activation function, which is capable of achieving fixed-time stability, meaning that the convergence time is uniformly bounded regardless of the initial conditions. It is worth noting that, unlike the results in [18,37,38] where a constant parameter was adopted, the parameter $\Gamma(t)$ considered in this paper is time-varying, which greatly contributes to the convergence of (4). In practical hardware implementation, numerical overflow can be effectively prevented by applying input saturation to the parameter, without affecting its fixed-time convergence performance.

Remark 4. It is worth noting that the discontinuous signum function may induce chattering in practical implementation. This issue can be alleviated by replacing it with a continuous approximation, such as a saturation function or a hyperbolic tangent function. Such an approximation will reduce the chattering effect and make the controller easier to implement in practical applications [39].

4. Main Results

4.1. ZGS-Based Optimization Algorithm Design

In this section, based on the ZGS method and the ZNN model, we design an optimization algorithm that drives the agent's state to track the time-varying optimal solution within fixed time. For agent i , we design the controller as

$$\begin{cases} u_i = p_i(-\Gamma_1(t)\Xi_1(z_i) - \sum_{j \in \mathcal{N}_i} \Xi_1(x_{ij}) - \frac{\partial}{\partial t} \nabla f_i(x_i, t)), & (7a) \\ \dot{p}_i = p_i \Gamma_1(t) \Xi_1(p_i^{-1} - H_i) p_i - p_i \dot{H}_i p_i, & (7b) \\ z_i = \nabla f_i(x_i, t) + \int_0^t \sum_{j \in \mathcal{N}_i} \Xi_1(x_{ij}) ds, & (7c) \end{cases}$$

where $i = 1, 2, \dots, N$, u_i is the designed optimal controller for agent i , while z_i is an auxiliary variable; $x_{ij} = x_i - x_j$, $H_i = \nabla^2 f_i(x_i, t)$; the other parameters are same as in the previous subsection.

Remark 5. A novel distributed optimization algorithm based on the ZNN model is presented in Equation (7). The proposed controller comprises three components, each serving a distinct role in achieving fixed-time convergence to the time-varying optimal solution. Specifically, z_i drives the global gradient of the distributed optimization problem to zero within a fixed time and maintains it at zero thereafter, thereby satisfying the initial condition required by the ZGS algorithm. Built upon the ZNN model, p_i enables each agent to approximate the inverse of its time-varying Hessian matrix in fixed time, providing a fundamental computational basis for the distributed optimization scheme.

Remark 6. The traditional ZGS algorithm drives the system state to slide along an invariant zero-gradient manifold and converge to the global optimal solution, provided that the initial state lies within the region of a local optimum. In contrast to the algorithms in [9,27], the proposed control law u_i extends the ZGS framework to time-varying problems. By incorporating sliding mode control, it guarantees that each agent, starting from arbitrary initial conditions, converges to the time-varying optimal solution within a fixed time.

Define $e_i = p_i^{-1} - H_i$; taking the derivative of e_i yields

$$\dot{e}_i = -\Gamma_1(t) \Xi_1(e_i). \quad (8)$$

Evidently, Equation (8) is a ZNN model. For matrix-type e_i , $\Xi_1(e_i) = [\Xi_1(e_i^{kl})]_{m \times m}$, where e_i^{kl} denotes the kl th element of matrix e_i . Next, we prove that system (8) can converge to zero in fixed time.

Theorem 1. Starting from any initial state $e_i(0)$, the matrix-type error function e_i in system (8) is guaranteed to converge to zero in fixed time T_1 , i.e., p_i approaches H_i^{-1} within the fixed time T_1 .

Proof. Consider the kl th subsystem of (8):

$$\dot{e}_i^{kl}(t) = -\Gamma_1(t) \Xi_1(e_{kl}(t)), k, l = 1, \dots, m.$$

Defining the Lyapunov functions $V_{kl} = (e_i^{kl})^2$, note that $V_{kl} \geq 0$, $V_{kl} = 0$ if and only if $e_i^{kl} = 0$. Taking the derivatives of V_{kl} yields

$$\begin{aligned} \dot{V}_{kl} &= 2e_i^{kl} \dot{e}_i^{kl} = -2e_i^{kl} \Gamma_1(t) \Xi_1(e_i^{kl}) \\ &= -2e_i^{kl} \Gamma_1(t) (\lambda_1 |e_i^{kl}|^p + \lambda_2 |e_i^{kl}|^q + \lambda_3) \text{sign}(e_i^{kl}) \\ &\leq -2\Gamma_1(t) (\lambda_1 |e_i^{kl}|^{p+1} + \lambda_2 |e_i^{kl}|^{q+1}) \\ &= -2\Gamma_1(t) (\lambda_1 (|e_i^{kl}|^2)^{\frac{p+1}{2}} + \lambda_2 (|e_i^{kl}|^2)^{\frac{q+1}{2}}) \\ &= -2\lambda_1 \Gamma_1(t) V_{kl}^{\frac{p+1}{2}} - 2\lambda_2 \Gamma_1(t) V_{kl}^{\frac{q+1}{2}}. \end{aligned} \quad (9)$$

Since $\Gamma_1(t) = \exp(\vartheta_1 t + \vartheta_0) > 1$ for $\vartheta_0, \vartheta_1 \geq 0$, following (9), one has

$$\dot{V}_{kl} \leq -2\lambda_1 V_{kl}^{\frac{p+1}{2}} - 2\lambda_2 V_{kl}^{\frac{q+1}{2}}. \quad (10)$$

According to Lemma 3, the kl th subsystem is fixed-time stable and the settling time is $T_1 = 1/\lambda_1(1-p) + 1/\lambda_2(q-1)$. That is, $\lim_{t \rightarrow T_1} e_i^{kl} = 0$, which implies that $\lim_{t \rightarrow T_1} e_i = 0_{m \times m}$ because of the arbitrariness of e_i^{kl} . $\square$

Remark 7. The ZNN dynamics (7b) ensures that the approximation error e_i converges to zero within a fixed time T_1 , i.e., $p_i(t) = H_i^{-1}(t)$ holds exactly for all $t \geq T_1$. Therefore, the approximation error has no influence on the steady-state optimization accuracy.

In Theorem 1, $e_i = 0_{m \times m}$ indicates that $p_i H_i = H_i p_i = I_{N \times N}$ for all $i \in \nu$ when $t > T_1$, i.e., $p_i = (\nabla_{x_i}^2 f_i(x_i, t))^{-1}$ for $t \geq T_1$. Therefore, algorithm (7) can be rewritten as

$$\begin{cases} u_i = (H_i)^{-1}[-\Gamma_1(t)\Xi_1(z_i) - \sum_{j \in \mathcal{N}_i} \Xi_1(x_{ij}) - \frac{\partial}{\partial t} \nabla f_i(x_i, t)], \\ z_i = \nabla f_i(x_i, t) + \int_0^t \sum_{j \in \mathcal{N}_i} \Xi_1(x_{ij}) ds. \end{cases} \quad (11a)$$

$$(11b)$$

Theorem 2. If Assumptions 1–3 hold, and $\lambda_3 > (N-1)hl/\mu$, then system (2) under algorithm (7) can converge to the optimal trajectory $x^*(t)$ within the fixed time T_2 .

Proof. From (11a), differentiating (11b) yields

$$\begin{aligned} \dot{z}_i &= H_i \dot{x}_i + \frac{\partial}{\partial t} \nabla f_i(x_i, t) + \sum_{j \in \mathcal{N}_i} \Xi_1(x_{ij}) \\ &= -\Gamma_1(t)\Xi_1(z_i). \end{aligned} \quad (12)$$

The neuro-dynamic (12) is a ZNN model with a time-varying parameter $\Gamma_1(t) = \exp(\vartheta_1 t + \vartheta_0)$. In the following, we prove that z_i in Equation (12) can converge to zero in fixed time.

Defining the Lyapunov functions $V_i = z_i^2$, note that $V_i \geq 0$, $V_i = 0$ if and only if $z_i = 0$. Taking the derivatives of V_i yields

$$\begin{aligned} \dot{V}_i &= 2z_i \dot{z}_i = -2z_i \Gamma_1(t) \Xi_1(z_i) \\ &= -2z_i \Gamma_1(t) (\lambda_1 |z_i|^p + \lambda_2 |z_i|^q + \lambda_3) \text{sign}(z_i) \\ &\leq -2\Gamma_1(t) (\lambda_1 |z_i|^{p+1} + \lambda_2 |z_i|^{q+1}) \\ &= -2\Gamma_1(t) (\lambda_1 (|z_i|^2)^{\frac{p+1}{2}} + \lambda_2 (|z_i|^2)^{\frac{q+1}{2}}) \\ &= -2\lambda_1 \Gamma_1(t) V_i^{\frac{p+1}{2}} - 2\lambda_2 \Gamma_1(t) V_i^{\frac{q+1}{2}}. \end{aligned} \quad (13)$$

Next, we consider the following two cases' accordance with the initial value $e_{kl}(0)$:

(1) If the initial value $z_i(0) \leq 1$, (13) can be further converted into

$$\dot{V}_i \leq -2\lambda_1 \Gamma_1(t) V_i^{\frac{p+1}{2}}. \quad (14)$$

Define $M_i = V_i^{\frac{1-p}{2}}$; from (14), we have

$$\dot{M}_i(t) \leq -\lambda_1(1-p)\Gamma_1(t). \quad (15)$$

Integrating both sides of (15) simultaneously from 0 to t yields

$$\begin{aligned} M_i(t) &\leq M_i(0) - \lambda_1(1-p) \int_0^t \exp(\vartheta_1 s + \vartheta_0) ds \\ &\leq 1 - \lambda_1(1-p) \frac{\exp(\vartheta_0)}{\vartheta_1} [\exp(\vartheta_1 t) - 1]. \end{aligned} \quad (16)$$

From (16), we know that $M_i(t) \leq 0$ if $t \geq T_2' = \frac{1}{\vartheta_1} \ln(\frac{\vartheta_1}{\lambda_1(1-p)\exp(\vartheta_0)} + 1)$. Since $M_i(t) \geq 0$ for $\forall t \geq 0$, it gets $M_i(t) \equiv 0$ when $t \geq T_2'$, i.e., $z_i \equiv 0$ when $t \geq T_2'$. Therefore, z_i can converge to zero within T_2' .

(2) Considering the initial value $z_i(0) > 1$, we claim that there exists $\hat{t}$ such that $V_i(\hat{t}) \leq 1$, where

$$\hat{t} \leq T_2'' = \frac{1}{\vartheta_1} \ln(\frac{\vartheta_1}{\lambda_2(q-1)\exp(\vartheta_0)} + 1). \quad (17)$$

Otherwise, suppose that $V_i(t) > 1$ for all $t \in [0, T_2''];$ (13) can be further converted into

$$\dot{V}_i \leq -2\lambda_2\Gamma_1(t)V_i^{\frac{q+1}{2}}. \quad (18)$$

Define $W_i = V_i^{\frac{1-q}{2}}$; from (18), we have

$$\dot{W}_i(t) \geq \lambda_2(q-1)\Gamma_1(t). \quad (19)$$

Integrating both sides of (19) simultaneously from 0 to T_2'' gives

$$\begin{aligned} W_i(T_2'') &\geq W_i(0) + \lambda_2(q-1) \int_0^{T_2''} \exp(\vartheta_1 s + \vartheta_0) ds \\ &\geq \lambda_2(q-1) \frac{\exp(\vartheta_0)}{\vartheta_1} [\exp(\vartheta_1 T_2'') - 1]. \end{aligned} \quad (20)$$

From (20), we get $W_i(T_2'') \geq 1$. That is, $V_i(T_2'') \leq 1$ because of $q > 1$. This is contract with $V_i(T_2'') > 1$.

By combining case (1), we can conclude that z_i can converge to zero in fixed time $T_2 = T_2' + T_2''$.

Since the interaction graph $\mathcal{G}$ is undirected and connected, from (11b), we know that $\sum_{i=1}^N z_i = \sum_{i=1}^N f_i(x_i, t)$ always holds. Therefore, system (2) achieves ZGS in $t < T_2$ and does not violate it when $t > T_2$.

For $t > T_2$, algorithm (7) can be rewritten as

$$u_i = -(H_i)^{-1} \left(\sum_{j \in \mathcal{N}_i} \Xi_1(x_{ij}) + \frac{\partial}{\partial t} \nabla f_i(x_i, t) \right). \quad (21)$$

Next, we prove that all agents can reach consensus in fixed time. Define $e_i = x_i - \bar{x}$, $e = (e_1, \dots, e_N)^T$, $e_{ij} = e_i - e_j$, where $\bar{x} = \sum_{i=1}^N x_i / N$. Take the Lyapunov function as $V = \|e\|^2 / 2$; its derivative respect to t can be calculated as

$$\dot{V} = - \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (x_i - \bar{x})^T H_i^{-1} \Xi_1(x_{ij}) - \frac{1}{N} \sum_{i,j=1}^N x_{ij}^T H_i^{-1} \frac{\partial}{\partial t} \nabla f_i(x_i, t). \quad (22)$$

Substituting $e_i = x_i - \bar{x}$ and $e_{ij} = e_i - e_j = x_i - x_j$ into (22), the first term of $\dot{V}$ becomes

$$- \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (x_i - \bar{x})^T H_i^{-1} \Xi_1(x_{ij})$$

$$\begin{aligned}
&= \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} e_{ij}^T H_i^{-1} \left(-\lambda_1 \text{sign}^p(e_{ij}) - \lambda_2 \text{sign}^q(e_{ij}) - \lambda_3 \text{sign}(e_{ij}) \right) \\
&= -\frac{\lambda_1}{2} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} e_{ij}^T H_i^{-1} \text{sign}^p(e_{ij}) - \frac{\lambda_2}{2} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} e_{ij}^T H_i^{-1} \text{sign}^q(e_{ij}) \\
&\quad - \frac{\lambda_3}{2} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} e_{ij}^T H_i^{-1} \text{sign}(e_{ij}),
\end{aligned} \tag{23}$$

where the equality follows from the symmetry of the undirected graph $\mathcal{G}$ (i.e., exchanging the subscripts i and j).

Since Assumption 2 guarantees $\mu I_n \leq \nabla^2 f_i(x_i, t) \leq l I_n$, we have $\frac{1}{\omega} I_n \leq H_i^{-1} \leq \frac{1}{\theta} I_n$. Applying this fact yields

$$\begin{aligned}
e_{ij}^T H_i^{-1} \text{sign}^p(e_{ij}) &\geq \frac{1}{l} \|e_{ij}\|_{p+1}^{p+1}, \\
e_{ij}^T H_i^{-1} \text{sign}^q(e_{ij}) &\geq \frac{1}{l} \|e_{ij}\|_{q+1}^{q+1}, \\
e_{ij}^T H_i^{-1} \text{sgn}(e_{ij}) &\geq \frac{1}{l} \|e_{ij}\|_1.
\end{aligned}$$

Substituting the above three inequalities into (23) gives

$$\begin{aligned}
& - \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (x_i - \bar{x})^T H_i^{-1} \Xi_1(x_{ij}) \\
& \leq -\frac{\lambda_1}{2l} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|e_{ij}\|_{p+1}^{p+1} - \frac{\lambda_2}{2l} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|e_{ij}\|_{q+1}^{q+1} - \frac{\lambda_3}{2l} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|e_{ij}\|_1.
\end{aligned} \tag{24}$$

We next consider the second term of $\dot{V}$ in (22). By exchanging the subscripts i and j , we obtain

$$-\frac{1}{N} \sum_{i,j=1}^N e_{ij}^T H_i^{-1} \frac{\partial}{\partial t} \nabla f_i(x_i, t) = -\frac{1}{2N} \sum_{i,j=1}^N e_{ij}^T H_i^{-1} \left(\frac{\partial}{\partial t} \nabla f_i(x_i, t) - \frac{\partial}{\partial t} \nabla f_j(x_j, t) \right).$$

Applying the Cauchy–Schwarz inequality and the triangle inequality gives

$$\begin{aligned}
& -\frac{1}{2N} \sum_{i,j=1}^N e_{ij}^T H_i^{-1} \left(\frac{\partial}{\partial t} \nabla f_i(x_i, t) - \frac{\partial}{\partial t} \nabla f_j(x_j, t) \right) \\
& \leq \frac{1}{2N} \sum_{i,j=1}^N \|e_{ij}\| \cdot \|H_i^{-1}\| \cdot \left\| \frac{\partial}{\partial t} \nabla f_i(x_i, t) - \frac{\partial}{\partial t} \nabla f_j(x_j, t) \right\| \\
& \leq \frac{1}{2N} \sum_{i,j=1}^N \|e_{ij}\|_1 \cdot \frac{1}{\mu} \cdot 2\hbar \\
& = \frac{\hbar}{N\mu} \sum_{i,j=1}^N \|e_{ij}\|_1,
\end{aligned} \tag{25}$$

where we have used $\|H_i^{-1}\| \leq 1/\mu$ from $\mu I_n \leq \nabla^2 f_i(x_i, t) \leq l I_n$, and $\left\| \frac{\partial}{\partial t} \nabla f_i(x_i, t) - \frac{\partial}{\partial t} \nabla f_j(x_j, t) \right\|_\infty \leq 2\hbar$ from Assumption 3. Combining (24) and (25) yields

$$\dot{V} \leq -\frac{\lambda_1}{2l} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|e_{ij}\|_{p+1}^{p+1} - \frac{\lambda_2}{2l} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|e_{ij}\|_{q+1}^{q+1} - \frac{\lambda_3}{2l} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|e_{ij}\|_1 + \frac{\hbar}{\mu N} \sum_{i,j=1}^N \|e_{ij}\|_1. \tag{26}$$

From (26), using the inequality $\sum_{i,j=1}^N \|e_{ij}\|_1 \leq \frac{N(N-1)}{2} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|e_{ij}\|_1$ in [19], we obtain

$$\dot{V} \leq -\frac{\lambda_1}{2l} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|e_{ij}\|_{p+1}^{p+1} - \frac{\lambda_2}{2l} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|e_{ij}\|_{q+1}^{q+1} - \left(\frac{\lambda_3}{2l} - \frac{\hbar(N-1)}{2\mu} \right) \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|e_{ij}\|_1.$$

Since $\lambda_3 > (N-1)\hbar l / \mu$, we have

$$\dot{V} \leq -\frac{\lambda_1}{2l} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|e_{ij}\|_{p+1}^{p+1} - \frac{\lambda_2}{2l} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|e_{ij}\|_{q+1}^{q+1}. \quad (27)$$

By Lemma 2 and the fact that $\sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|e_{ij}\|^2 = 2e^T \mathcal{L}_{\otimes} e \geq 2\lambda_{\min} \|e\|^2 = 4\lambda_{\min} V$, we have

$$\begin{aligned} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|e_{ij}\|_{\alpha_2+1}^{\alpha_2+1} &\geq 2^{\alpha_2+1} \lambda_{\min}^{\frac{\alpha_2+1}{2}} V^{\frac{\alpha_2+1}{2}}, \\ \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|e_{ij}\|_{\beta_2+1}^{\beta_2+1} &\geq (N(N-1)m)^{-\frac{\beta_2}{2}} 2^{\beta_2+1} \lambda_{\min}^{\frac{\beta_2+1}{2}} V^{\frac{\beta_2+1}{2}}. \end{aligned}$$

Substituting the above two inequalities into (27), one gets

$$\begin{aligned} \dot{V} &\leq -\eta_1 [e^T \otimes L \otimes e]^{\frac{p+1}{2}} - \eta_2 [e^T \otimes L \otimes e]^{\frac{q+1}{2}} \\ &\leq -2^{\frac{p+1}{2}} \lambda_{\min}^{\frac{p+1}{2}}(L) \eta_1 V^{\frac{p+1}{2}} - 2^{\frac{q+1}{2}} \lambda_{\min}^{\frac{q+1}{2}}(L) \eta_2 V^{\frac{q+1}{2}} \\ &= -a_1 V^{\frac{p+1}{2}} - b_1 V^{\frac{q+1}{2}}, \end{aligned} \quad (28)$$

where $a_1 = 2^{\frac{p+1}{2}} \lambda_{\min}^{\frac{p+1}{2}}(L) \eta_1$, $b_1 = 2^{\frac{q+1}{2}} \lambda_{\min}^{\frac{q+1}{2}}(L) \eta_2$, $\eta_1 = \frac{2^{\frac{p-1}{2}} \lambda_1}{l}$ and $\eta_2 = \frac{2^{\frac{q-1}{2}} \lambda_2}{[N(N-1)n]^{\frac{q-1}{2}} l}$.

Consequently, it is determined that each agent goes to the optimal trajectory, i.e., $x_i(t) \rightarrow x^*(t)$, $i \in \nu$ within a fixed time $T = T_1 + T_2 + T_3$, where

$$T_3 = \frac{2^{1-p} l}{\lambda_{\min}^{\frac{p+1}{2}}(L) \lambda_1 (1-p)} + \frac{2^{1-q} [N(N-1)n]^{\frac{q-1}{2}} l}{\lambda_{\min}^{\frac{q+1}{2}}(L) \lambda_2 (q-1)}. \quad (29)$$

□

Remark 8. The algorithm in [25,40] is also based on the ZGS framework and employs a two-stage architecture for fixed-time convergence, where the first stage drives arbitrary initial states to local optima to satisfy the ZGS prerequisites. Although theoretically sound, this staged estimation yields a conservative upper bound on the convergence time. In contrast, the proposed algorithm (7) extends the ZGS-based time-varying scheme to a single-stage update rule, eliminating the need for preliminary initialization optimization and effectively reducing the conservatism of the settling-time estimate.

Remark 9. The optimization algorithm described above requires that all agents share an identical Hessian matrix. This condition is satisfied in problems such as distributed energy optimization and target tracking [24]. Nevertheless, it fails in more general scenarios, e.g., when each agent minimizes a private cost function with distinct curvature information. To this end, the following section presents an improved algorithm that does not require this consistency assumption, allowing each agent to minimize its own private cost function with an independent Hessian matrix.

4.2. PCM-Based Optimization Algorithm Design

In this subsection, we present a novel distributed optimization algorithm that removes the requirement of identical Hessian matrices. To this end, distributed estimation techniques are employed in conjunction with the prediction-correction method. The proposed controller is designed as follows:

$$u_i = -p_i \left(\frac{\partial}{\partial t} \nabla f_i(x_i, t) + \Xi_1(\hat{\theta}_i) \right) - \Gamma_1(t) \Xi_1 \left(\sum_{j \in \mathcal{N}_i} a_{ij} x_{ij} \right) - \|p_i \left(\frac{\partial}{\partial t} \nabla f_i(x_i, t) + \Xi_1(\hat{\theta}_i) \right)\| \text{sign} \left(\sum_{j \in \mathcal{N}_i} a_{ij} x_{ij} \right), \quad (30)$$

where p_i satisfies (7b), and $\hat{\theta}_i$ is a distributed estimator driven by

$$\begin{cases} \theta_i = \zeta_i + \varsigma_i, \\ \dot{\zeta}_i = - \sum_{j \in \mathcal{N}_i} \Xi_1(\theta_{ij}) - \lambda_4 \sum_{j \in \mathcal{N}_i} \text{sign}(\theta_{ij}) \|\nabla f_i - \nabla f_j\|, \end{cases} \quad (31)$$

where $\hat{\theta}_i = \theta_i$, $\theta_{ij} = \theta_i - \theta_j$, $\zeta_i = \nabla f_i$.

Our purpose in developing the distributed estimator (31) is to estimate the global information $\sum_{i=1}^N \nabla f_i$. Next, we prove that the auxiliary variable θ_i satisfies $\theta_i = \sum_{i=1}^N \varsigma_i / N$ in a fixed time T_4 , i.e., $\hat{\theta}_i = \sum_{i=1}^N \nabla f_i / N$ for $t > T_4$.

Theorem 3. For system (31), the variable $\hat{\theta}_i$ will converge to $\sum_{i=1}^N \nabla f_i / N$ within a fixed time T_4 if $\sum_{i=1}^N \zeta_i(0) = 0$ and $\lambda_4 > 1/(2N)$, where

$$T_4 = \frac{1}{\lambda_1 2^{\frac{p-1}{2}} \lambda_{\min}^{\frac{p+1}{2}}(L)(1-p)} + \frac{1}{\lambda_2 2^{\frac{q-1}{2}} \lambda_{\min}^{\frac{q+1}{2}}(L) N^{\frac{1-q}{2}}(q-1)}.$$

Proof. Define $\tilde{e}_i = \theta_i - \sum_{j=1}^N \theta_j / N$, choose a Lyapunov function as $\tilde{V} = \frac{1}{2} \sum_{i=1}^N \tilde{e}_i^T \tilde{e}_i$; taking its derivatives yields

$$\begin{aligned} \dot{\tilde{V}} &= \sum_{i=1}^N \tilde{e}_i^T \dot{\tilde{e}}_i = - \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \tilde{e}_i^T \Xi_1(\theta_{ij}) - \sum_{i=1}^N \tilde{e}_i^T \nabla f_i - \lambda_4 \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \tilde{e}_i^T \text{sign}(\theta_{ij}) \|\nabla f_{ij}\| \\ &\leq -\lambda_1 \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|\tilde{e}_{ij}\|^{p+1} - \lambda_2 \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|\tilde{e}_{ij}\|^{q+1} + \left(\frac{1}{2N} - \lambda_4\right) \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|\tilde{e}_{ij}\| \|\nabla f_{ij}\| \\ &\leq -\lambda_1 2^{\frac{p+1}{2}} \lambda_{\min}^{\frac{p+1}{2}}(L) \tilde{V}^{\frac{p+1}{2}} - \lambda_2 2^{\frac{q+1}{2}} \lambda_{\min}^{\frac{q+1}{2}}(L) N^{\frac{1-q}{2}} \tilde{V}^{\frac{q+1}{2}} \\ &\leq a_2 \tilde{V}^{\frac{p+1}{2}} - b_2 \tilde{V}^{\frac{q+1}{2}}, \end{aligned} \quad (32)$$

where $a_2 = -\lambda_1 2^{\frac{p+1}{2}} \lambda_{\min}^{\frac{p+1}{2}}(L)$, $b_2 = \lambda_2 2^{\frac{q+1}{2}} \lambda_{\min}^{\frac{q+1}{2}}(L) N^{\frac{1-q}{2}}$. Combining Lemma 3, it is constructed that $\theta_i = \theta_j = \sum_{k=1}^N \theta_k / N$ holds in a fixed time T_4 . Moreover, it has $\sum_{i=1}^N \zeta_i(0) = 0$ and $\sum_{i=1}^N \dot{\zeta}_i = 0$ for all $t > 0$, so $\sum_{i=1}^N \theta_i = \sum_{i=1}^N \varsigma_i$ since $\theta_i = \zeta_i + \varsigma_i$. It follows that $\theta_i = \theta_j = \sum_{i=1}^N \varsigma_i / N$, i.e., $\hat{\theta}_i = \sum_{i=1}^N \nabla f_i / N$ for $t \geq T_4$.

According to Theorem 1, p_i can converge to H_i^{-1} within a fixed time T_1 . Therefore, for $t > \max\{T_1, T_4\}$, controller (11) can be rewritten as

$$u_i = -H_i^{-1} \left(\frac{\partial}{\partial t} \nabla f_i(x_i, t) + \Xi_1(\hat{\theta}_i) \right) - \Gamma_1(t) \Xi_1 \left(\sum_{j \in \mathcal{N}_i} a_{ij} x_{ij} \right) - \|H_i^{-1} \left(\frac{\partial}{\partial t} \nabla f_i(x_i, t) + \Xi_1(\hat{\theta}_i) \right)\| \text{sign} \left(\sum_{j \in \mathcal{N}_i} a_{ij} x_{ij} \right). \quad (33)$$

□

Now, we are in a position to present our main results in the following theorem:

Theorem 4. *If Assumptions 1 and 4 hold, all agents under controller (11) can converge to the optimal trajectory $x^*(t)$ within a fixed time $T' = \max\{T_1, T_4\} + T_5 + T_6$, and the time-varying optimization (3) is realized in a fixed time T' , where $T_5 = T'_5 + T''_5$,*

$$T'_5 = \frac{1}{\theta_1} \ln\left(\frac{\theta_1(2\lambda_{\min}(L))^{\frac{-(p+1)}{2}}}{\lambda_1(1-p)\exp(\theta_0)} + 1\right), T''_5 = \frac{1}{\theta_1} \ln\left(\frac{\theta_1 N^{\frac{q-1}{2}}(2\lambda_{\min}(L))^{\frac{-(q+1)}{2}}}{\lambda_2(q-1)\exp(\theta_0)} + 1\right),$$

$$T_6 = 1/(\lambda_1 N^{1-p} 2^{\frac{p-1}{2}}(1-p)) + 1/(\lambda_2 N^{1-q} 2^{\frac{q-1}{2}}(q-1)).$$

Proof. Consider the Lyapunov function as $\bar{V} = \frac{1}{2} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} a_{ij} \|x_{ij}\|^2$, where $x_{ij} = x_i - x_j$. Taking its derivative along the system trajectory and substituting the control law yields

$$\begin{aligned} \dot{\bar{V}} &= \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} a_{ij} (x_{ij})^T \dot{x}_i \\ &\leq - \sum_{i=1}^N z_i^T (H_i)^{-1} \left(\frac{\partial}{\partial t} \nabla f_i(x_i, t) + \Xi_1(\hat{\theta}_i) \right) \\ &\quad - \sum_{i=1}^N \|z_i\| \left\| (H_i)^{-1} \left(\frac{\partial}{\partial t} \nabla f_i(x_i, t) + \Xi_1(\hat{\theta}_i) \right) \right\| \\ &\quad - \Gamma_1(t) \lambda_1 \sum_{i=1}^N \|z_i\|^{p+1} - \Gamma_1(t) \lambda_2 \sum_{i=1}^N \|z_i\|^{q+1}, \end{aligned} \quad (34)$$

where $z_i \triangleq \sum_{j \in \mathcal{N}_i} a_{ij} x_{ij}$.

Applying the Cauchy–Schwarz inequality to the first term yields

$$-z_i^T (H_i)^{-1} \left(\frac{\partial}{\partial t} \nabla f_i(x_i, t) + \Xi_1(\hat{\theta}_i) \right) \leq \|z_i\| \left\| (H_i)^{-1} \left(\frac{\partial}{\partial t} \nabla f_i(x_i, t) + \Xi_1(\hat{\theta}_i) \right) \right\|.$$

Substituting this inequality into (34), we obtain

$$\dot{\bar{V}} \leq -\Gamma_1(t) \lambda_1 \sum_{i=1}^N \|z_i\|^{p+1} - \Gamma_1(t) \lambda_2 \sum_{i=1}^N \|z_i\|^{q+1}. \quad (35)$$

By virtue of Lemma 2, we have

$$\sum_{i=1}^N \|z_i\|^{p+1} \geq \left(\sum_{i=1}^N \|z_i\|^2 \right)^{\frac{p+1}{2}}, \quad \sum_{i=1}^N \|z_i\|^{q+1} \geq N^{\frac{1-q}{2}} \left(\sum_{i=1}^N \|z_i\|^2 \right)^{\frac{q+1}{2}}.$$

Substituting these inequalities into (35) yields

$$\dot{\bar{V}} \leq -\Gamma_1(t) \lambda_1 \left(\sum_{i=1}^N \|z_i\|^2 \right)^{\frac{p+1}{2}} - \Gamma_1(t) \lambda_2 N^{\frac{1-q}{2}} \left(\sum_{i=1}^N \|z_i\|^2 \right)^{\frac{q+1}{2}}. \quad (36)$$

Using the property of the graph Laplacian matrix L , we have $\sum_{i=1}^N \|z_i\|^2 = x^T (L \otimes I_n) x \geq 2\lambda_{\min}(L) \bar{V}$. Substituting this into (36) gives

$$\dot{\bar{V}} \leq -\Gamma_1(t) \lambda_1 (2\lambda_{\min}(L))^{\frac{p+1}{2}} \bar{V}^{\frac{p+1}{2}} - \Gamma_1(t) \lambda_2 N^{\frac{1-q}{2}} (2\lambda_{\min}(L))^{\frac{q+1}{2}} \bar{V}^{\frac{q+1}{2}}. \quad (37)$$

Following the same procedure as in Theorem 2, it can be concluded from (37) that x_{ij} reaches zero in a fixed time T_5 . Consequently, all agents can reach the optimal trajectory in fixed time.

For $t > \max\{T_1, T_4\} + T_5$, consider the following Lyapunov function:

$$\check{V} = \frac{1}{2} \left(\sum_{i=1}^N \nabla f_i \right)^T \left(\sum_{i=1}^N \nabla f_i \right). \quad (38)$$

Taking its time derivative and substituting the control law, we obtain

$$\begin{aligned} \dot{\check{V}} &= \left(\sum_{i=1}^N \nabla f_i \right)^T \left(\sum_{i=1}^N \left(\frac{\partial}{\partial t} \nabla f_i(x_i, t) + H_i \dot{x}_i \right) \right) \\ &= - \left(\sum_{i=1}^N \nabla f_i \right)^T \left[\sum_{i=1}^N (\lambda_1 N^{-p} |\sum_{i=1}^N \nabla f_i|^p + \lambda_2 N^{-q} |\sum_{i=1}^N \nabla f_i|^q) \right] \text{sign} \left(\sum_{i=1}^N \nabla f_i \right) \\ &\leq -\lambda_1 N^{1-p} 2^{\frac{1+p}{2}} \check{V}^{\frac{1+p}{2}} - \lambda_2 N^{1-q} 2^{\frac{1+q}{2}} \check{V}^{\frac{1+q}{2}}. \end{aligned} \quad (39)$$

Let $a_3 = \lambda_1 N^{1-p} 2^{\frac{1+p}{2}}$, $b_3 = \lambda_2 N^{1-q} 2^{\frac{1+q}{2}}$; it has

$$\dot{\check{V}} \leq -a_3 \check{V}^{\frac{1+p}{2}} - b_3 \check{V}^{\frac{1+q}{2}}. \quad (40)$$

From Lemma 3, the global optimization function approaches zero within a fixed time T' . $\square$

Remark 10. It is worth noting that our proposed method avoids direct Hessian matrix inversion. Theoretically, while conventional inversion algorithms incur a complexity of $O(N^3)$, the ZNN approach reduces the overall complexity to $O(N)$ due to its inherent parallel computing architecture. This advantage becomes increasingly significant as the system dimension N grows. A systematic numerical comparison of CPU time and memory usage is an important direction for our future work.

Remark 11. It is worth noting that both algorithm (7) and algorithm (30) are designed for Problem (3). Algorithm (7) requires all agents to share identical Hessian matrices, which limits its applicability to heterogeneous scenarios. In contrast, algorithm (30) incorporates a distributed estimator, where each agent updates itself relying solely on its local Hessian matrix, thereby removing the requirement for identical Hessian matrices. Nevertheless, this benefit comes at the cost of elevated computational complexity due to the gradient estimation procedure. There exists a trade-off between performance improvement and computational overhead for the two algorithms.

Remark 12. For distributed time-varying optimization problems, existing methods exhibit significant differences in convergence time performance. The prediction-correction strategy in [24] achieves fixed-time consensus, but its optimization process is only asymptotically convergent. The method in [41] can achieve fixed-time global time-varying optimization, yet it requires complete global gradient information. Even if global information is accessible in multi-AUV escorting scenarios, the algorithm remains inherently centralized. Unlike the above approaches, the two algorithms proposed in this paper simultaneously achieve fixed-time consensus and fixed-time optimization within a fully distributed framework.

Remark 13. The main objective of this paper is to enable each agent to minimize a global cost function. To overcome the reliance on global gradient information in existing approaches, this paper proposes a distributed estimator that allows all agents to reconstruct the global cost function information. Unlike existing distributed estimation mechanisms [18,40], the developed estimator is specifically designed to estimate the integral of the global gradient function. Based on this design, each

agent can indirectly obtain global gradient information, thereby further eliminating the requirements for boundedness of partial derivatives of the gradient function.

5. Numerical Simulations

In this section, a numerical example of battery energy storage balancing is provided to verify the theoretical fixed-time convergence properties and the effectiveness of the proposed distributed scheme.

Example: A power balancing task for a battery energy storage system (BESS) composed of six battery packages with the same capacity is considered herein. The interconnection topology of the BESS is presented in Figure 1. For each i -th battery packages, the state-of-charge (SOC) is associated with its power output P_i (KW) by the relationship $\text{SOC}_i = P_i$ [42], while the power output dynamics $\dot{P}_i(t)$ is governed by the controller input u_i . Once the optimal power trajectory $P^*(t)$ is determined, the SOC balancing objective can be effectively achieved by implementing the distributed control strategy proposed in [43]. Consequently, the core goal of this study is to distributively compute a consistent optimal solution for P_i on the basis of the local cost function of each individual battery package.

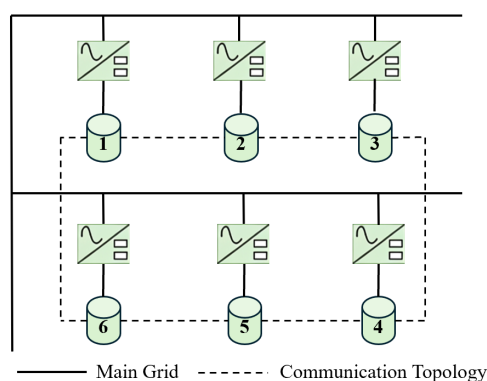


Figure 1. Communication topology of the BESS.

The optimization-based power balancing problem is formulated as follows:

$$\min \sum_{i=1}^6 f_i(P_i, t) \text{ s.t. } P_i = P_j, \quad (41)$$

where $i, j = 1, \dots, 6, i \neq j$, and the local cost function f_i of package i is $f_i(P_i, t) = \hbar(t)[(P_{i1} - 0.015i \cos(t))^2 + (P_{i2} - 0.015i \sin(t))^2]$ with $\hbar(t) = 1 + 0.3 \sin(t)$. Through straightforward mathematical derivation, the gradient, Hessian matrix, and the partial derivative of the gradient of the cost function are obtained as follows:

$$\begin{aligned} \nabla f_i(P_i, t) &= 2\hbar(t)[P_{i1} - 0.015i \cos(t); P_{i2} - 0.015i \sin(t)]; \\ \nabla^2 f_i(P_i, t) &= 2\hbar(t)[1, 0; 0, 1]; \hbar'(t) = 0.3 \cos(t); \\ \nabla_t f_i(P_i, t) &= 2[\hbar'(t)(P_{i1} - 0.015i \cos(t)) + \hbar(t)(0.015i \sin(t)); \\ &\quad \hbar'(t)(P_{i2} - 0.015i \sin(t)) - \hbar(t)(0.015i \cos(t))]. \end{aligned}$$

It has been verified that Assumptions 2 and 3 hold. For the proposed algorithm (7), the parameters are configured as follows: $\vartheta_0 = \vartheta_1 = 1$, $\lambda_1 = \lambda_2 = 1$, $\lambda_3 = 6$, $p = 0.5$, $q = 1.5$. The initial power output values of the six battery modules are set as $P_1(0) = (-1, -3.2)^T$, $P_2(0) = (-4.3, 1)^T$, $P_3(0) = (2.26, -2.3)^T$, $P_4(0) = (0.46, 0.6)^T$, $P_5(0) = (-2.4, -4.5)^T$, and $P_6(0) = (-1.8, 1.32)^T$, respectively.

Under the aforementioned parameter configurations and initial conditions, the algorithm (7) was applied to solve the BESS power balancing optimization problem (41), and the corresponding simulation results are displayed in Figures 2–4. Figure 2 shows that the ZNN model ensures that each p_i converges to the target inverse Hessian $H_i = [1/(2 + 0.6 \sin(t)), 0; 0, 1/(2 + 0.6 \sin(t))]$ in fixed time from distinct initial states. Based on the approximation results, the power output trajectories of each battery package and the evolution process of the optimizer are illustrated in Figures 3 and 4a, while the variation in the global cost function is shown in Figure 4b. Numerical simulation results demonstrate that all system states reach consensus within 0.3s, although the trajectories at this stage are not yet optimal. With the continuous action of the control strategy, all power outputs converge to the globally optimal trajectory $P^*(t) = 0.0525[\cos(t), \sin(t)]^T$ within a fixed time, indicating that the optimization problem (41) is fully resolved. Therefore, the proposed algorithm ensures that the power grid operates at its maximum capacity within a fixed time. Figure 5 compares the performance of algorithm (7) with that of the existing methods [25,40] in terms of state trajectories and the average tracking error of the time-varying optimal solution. The results show that although the two-stage algorithms in [25,40] can eventually track the time-varying optimal trajectory, they require a longer convergence time. In contrast, the proposed algorithm achieves rapid convergence and maintains a smaller tracking error throughout the entire process.

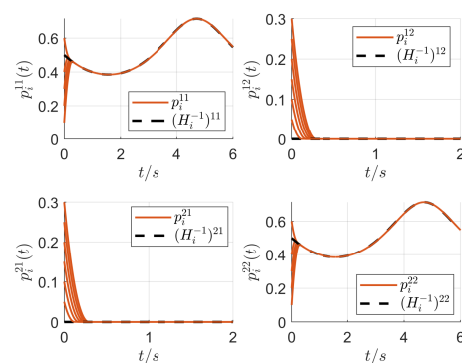


Figure 2. Trajectories of p_i with six random initial values.

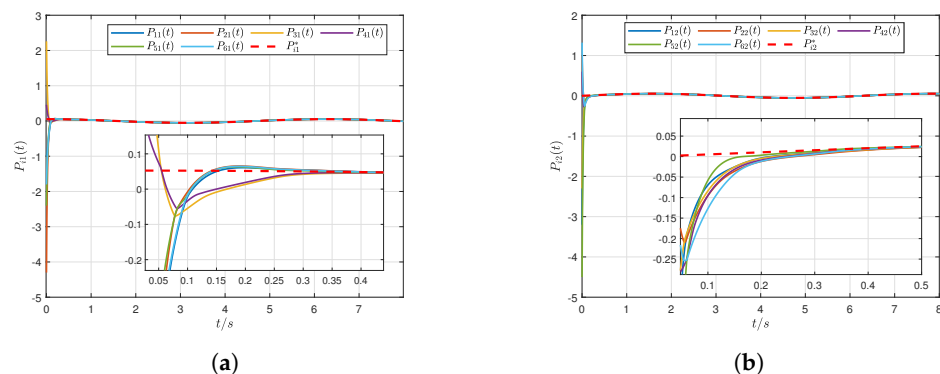


Figure 3. Dynamic evolutionary process of P_i under algorithm (7). (a) Evolutions of P_{i1} under algorithm (7). (b) Evolutions of P_{i2} under algorithm (7).

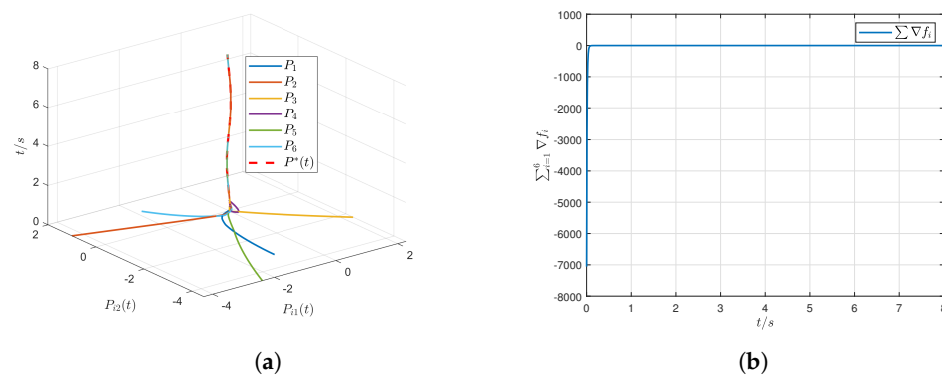


Figure 4. Evolutions of $P_i(t)$ and the global gradient under algorithm (7). (a) Trajectories of $P_i(t)$ and the optimizer $P^*(t)$ under algorithm (7). (b) Evolutions of the global gradient under algorithm (7).

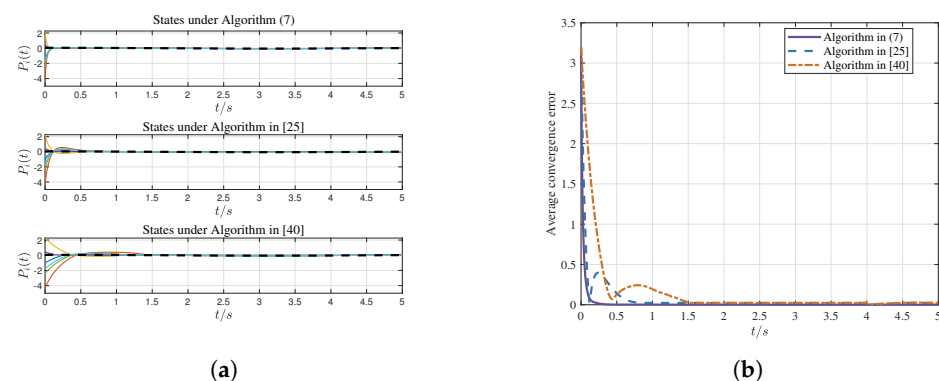


Figure 5. Comparisons of algorithm (7) and algorithms in [25] (with $\alpha_1 = \alpha_2 = \gamma_1 = \gamma_2 = 1$, $\mu_1 = 0.5$, $\nu_1 = 1.5$, $\alpha_3 = 6$) and [40] (with $\alpha_1 = \alpha_2 = \gamma_1 = \gamma_2 = 1$, $l = 0.25$). (a) Evolutions of P_i under algorithm (7) and algorithms in [25,40]. (b) Average convergence error under algorithm (7) and algorithms in [25,40].

To further verify the effectiveness of the distributed optimization strategy (30), the local objective function of the i -th battery package is adjusted to

$$f_i(P_i, t) = \tilde{h}_i(t)[(P_{i1} - 0.015i \cos(t))^2 + (P_{i2} - 0.015i \sin(t))^2], \tilde{h}_i(t) = i + 0.3 \sin(t).$$

Unlike the uniform time-varying coefficient adopted in algorithm (7), the time-varying weight coefficients $\tilde{h}_i(t)$ are independent across battery packages, allowing for a more accurate representation of the actual differences among BESS modules, such as variations in aging degree and rated capacity. Based on the modified local cost function, the gradient, Hessian matrix, and the partial derivative of $f_i(P_i, t)$ are re-derived as follows:

$$\nabla f_i(P_i, t) = 2\tilde{h}_i(t)[P_{i1} - 0.015i \cos(t); P_{i2} - 0.015i \sin(t)];$$

$$\nabla^2 f_i(P_i, t) = 2\tilde{h}_i(t)[1, 0; 0, 1]; \tilde{h}_i'(t) = 0.3 \cos(t);$$

$$\nabla_t f_i(P_i, t) = 2[\tilde{h}_i'(t)(P_{i1} - 0.015i \cos(t)) + \tilde{h}_i(t)(0.015i \sin(t)); \tilde{h}_i'(t)(P_{i2} - 0.015i \sin(t)) - \tilde{h}_i(t)(0.015i \cos(t))].$$

The initial power outputs of the six battery modules remain consistent with the previous settings. The parameters of algorithm (30) and estimator (31) are set as $\vartheta_0 = \vartheta_1 = 1$, $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1$, $p = 0.5$, and $q = 1.5$, and the simulation results are shown in Figures 6 and 7. Specifically, Figure 6 shows the trajectories of the estimated variables $\hat{\theta}_i$ under the proposed estimator. It can be observed that all estimated variables achieve

fixed-time consensus and converge to the average of the global gradient. Figures 7a and 8 illustrate the power output trajectories of six battery packages under algorithm (30), confirming the tracking performance of the optimal power trajectory. Figure 7b shows that the global gradient rapidly converges to zero, further validating the effectiveness and fast convergence of the proposed optimization algorithm. As illustrated in Figure 9, the performance of algorithm (30) was compared with that of the existing methods [24,41] in terms of state trajectories and average tracking error. It can be observed that the PCM-based algorithms in [24,41] either exhibit significant steady-state deviations from the optimal solution or suffer from slow convergence. Additionally, the algorithm in [41] requires global gradient information, making it essentially a centralized approach. By contrast, the proposed algorithm (30) not only converges rapidly but also maintains consistently low tracking error, demonstrating its superior tracking capability in distributed time-varying optimization.

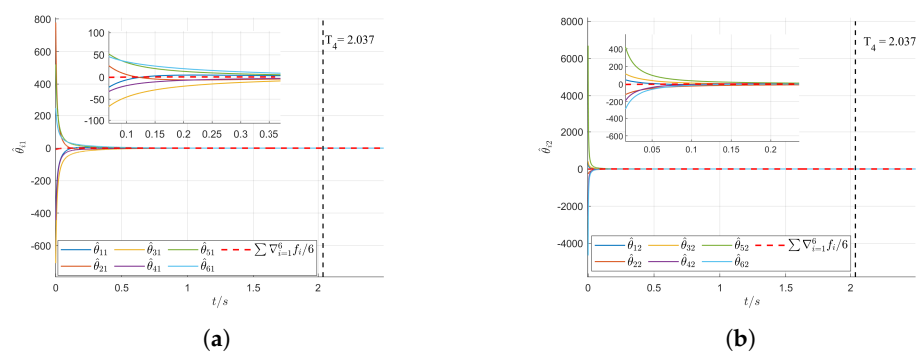


Figure 6. Estimations of the global average gradient $\frac{1}{6} \sum_{i=1}^6 \nabla f_i$ under algorithm (31). (a) Estimations of $\frac{1}{6} \sum_{i=1}^6 \nabla f_{i1}$ under algorithm (31). (b) Estimations of $\frac{1}{6} \sum_{i=1}^6 \nabla f_{i2}$ under algorithm (31).

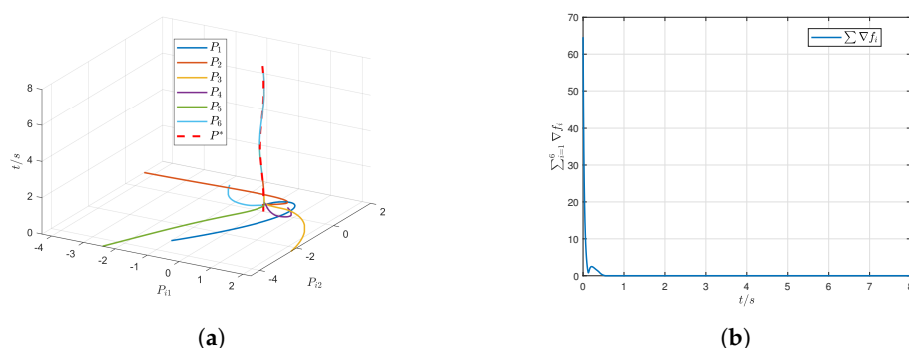


Figure 7. Evolutions of $P_i(t)$ and the global gradient under algorithm (30). (a) Trajectories of $P_i(t)$ and the optimizer $P^*(t)$ under algorithm (30). (b) Evolutions of the global gradient under algorithm (30).

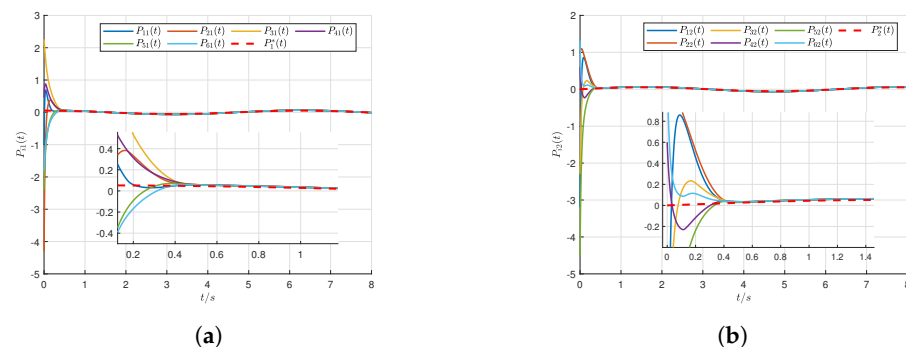


Figure 8. Dynamic evolutionary process of P_i under algorithm (30). (a) Evolutions of P_{i1} under algorithm (30). (b) Evolutions of P_{i2} under algorithm (30).

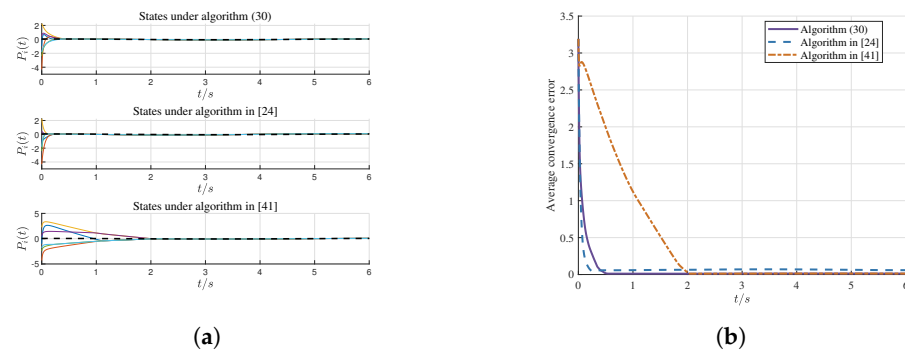


Figure 9. Comparisons of algorithm (30) and algorithms in [24,41]. (a) Evolutions of P_i under algorithm (30) and algorithms in [24,41]. (b) Average convergence error under algorithm (30) and algorithms in [24,41].

6. Conclusions

This paper proposes two distributed optimization algorithms for multi-agent systems with time-varying cost functions. The first algorithm integrates the ZGS method with sliding mode control and is applicable to scenarios where all agents share an identical Hessian matrix. The second algorithm extends the PCM to accommodate heterogeneous Hessian matrices across agents. Both algorithms achieve fixed-time convergence, meaning that the convergence time is independent of initial conditions. Furthermore, by employing a ZNN to approximate the inverse of the Hessian matrix in real time, the proposed algorithms achieve higher computational efficiency. Theoretical analysis based on Lyapunov methods and fixed-time stability theory rigorously establishes the convergence properties of the algorithms. Numerical simulations on a charge balancing problem in a BESS further confirm the theoretical results. Since the current work only considers ideal communication conditions, future work will explore the impact of communication delays and packet losses on the performance of the proposed algorithms. In addition, extending the current results to non-convex global objective functions is another important direction for future research.

Author Contributions: Conceptualization, X.L.; methodology, X.L. and L.L.; software, X.L. and L.L.; writing—original draft preparation, X.L. and L.L.; writing—review and editing, Q.G.; funding acquisition, Q.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. 61305076).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors appreciate the tremendous help given by the Associate Editor and Reviewers through their valuable comments.

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

Abbreviations

The following abbreviations are used in this manuscript:

MASs	Multi-agent systems
PCM	Prediction-correction method
ZNN	Zeroing neural network
ZGS	Zero-gradient-sum

References

1. Zhang, J.; Liu, S.; Xie, L. Distributed Convex Optimization Over Nonlinear Networks Under Set Constraints. *IEEE Trans. Autom. Control* **2025**, *70*, 4735–4742. <https://doi.org/10.1109/TAC.2025.3535578>.
2. Firouzbahrami, M.; Nobakhti, A. Finite-Time Distributed Economic Dispatch Over Network Systems With Coupled Local Costs. *IEEE Control Syst. Lett.* **2023**, *7*, 325–330. <https://doi.org/10.1109/LCSYS.2022.3188743>.
3. Mahmoudi, A.; Ghadikolaei, H.S.; Fischione, C. Cost-efficient Distributed optimization In Machine Learning Over Wireless Networks. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–7. <https://doi.org/10.1109/ICC40277.2020.9149216>.
4. Guo, G.; Zhang, R.; Zhou, Z.D. A local-minimization-free zero-gradient-sum algorithm for distributed optimization. *Automatica* **2023**, *157*, 111247. <https://doi.org/10.1016/j.automatica.2023.111247>.
5. Chen, J.; Yang, Y.; Qin, S. A Distributed Optimization Algorithm for Fixed-Time Flocking of Second-Order Multiagent Systems. *IEEE Trans. Netw. Sci. Eng.* **2024**, *11*, 152–162. <https://doi.org/10.1109/TNSE.2023.3292860>.
6. Wang, X.; Wang, G.; Li, S. Distributed Finite-Time Optimization for Integrator Chain Multiagent Systems with Disturbances. *IEEE Trans. Autom. Control* **2020**, *65*, 5296–5311. <https://doi.org/10.1109/TAC.2020.2979274>.
7. Gong, X.; Cui, Y.; Shen, J.; Xiong, J.; Huang, T. Distributed Optimization in Prescribed-Time: Theory and Experiment. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 564–576. <https://doi.org/10.1109/TNSE.2021.3126154>.
8. Fang, X.; Wen, G. Distributed optimal coordination of multi-agent systems with coupled objective functions: A fixed-time estimation-based approach. *Automatica* **2025**, *175*, 112213. <https://doi.org/10.1016/j.automatica.2025.112213>.
9. Lu, J.; Tang, C.Y. Zero-Gradient-Sum Algorithms for Distributed Convex Optimization: The Continuous-Time Case. *IEEE Trans. Autom. Control* **2012**, *57*, 2348–2354. <https://doi.org/10.1109/TAC.2012.2184199>.
10. Zhu, Y.; Yu, W.; Wen, G.; Chen, G.; Ren, W. Continuous-Time Distributed Subgradient Algorithm for Convex Optimization with General Constraints. *IEEE Trans. Autom. Control* **2019**, *64*, 1694–1701. <https://doi.org/10.1109/TAC.2018.2852602>.
11. Nedic, A.; Ozdaglar, A. Distributed Subgradient Methods for Multi-Agent Optimization. *IEEE Trans. Autom. Control* **2009**, *54*, 48–61. <https://doi.org/10.1109/TAC.2008.2009515>.
12. Makhdoumi, A.; Ozdaglar, A. Convergence Rate of Distributed ADMM Over Networks. *IEEE Trans. Autom. Control* **2017**, *62*, 5082–5095. <https://doi.org/10.1109/TAC.2017.2677879>.
13. Jahvani, M.; Guay, M. A Modified Distributed Gradient Dynamics for Multiagent Optimization on Directed Networks. *IEEE Trans. Control Netw. Syst.* **2025**, *12*, 1454–1465. <https://doi.org/10.1109/TCNS.2025.3526343>.
14. Zhang, B.; Hou, Y.; Yin, H.; Lv, M.; Yang, A.; Wu, L. Cooperative Dynamic Target Tracking: Distributed Time-Varying Optimization for Multi-UAV System. *IEEE Trans. Aerosp. Electron. Syst.* **2025**, *61*, 12245–12257. <https://doi.org/10.1109/TAES.2025.3574701>.
15. Yang, A.; Liang, X.; Zhang, J.; Hou, Y.; Wang, N. Global Optimal Consensus for UAV Swarms With Time-Varying Objective Functions and Bounded Input Constraints. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 3822–3836. <https://doi.org/10.1109/TAES.2024.3368996>.
16. Wang, B.; Fei, Q.; Wu, Q. Distributed Time-Varying Resource Allocation Optimization Based on Finite-Time Consensus Approach. *IEEE Control Syst. Lett.* **2021**, *5*, 599–604. <https://doi.org/10.1109/LCSYS.2020.3004764>.
17. Ding, Z. Distributed Time-Varying Optimization—An Output Regulation Approach. *IEEE Trans. Cybern.* **2024**, *54*, 2155–2165. <https://doi.org/10.1109/TCYB.2022.3219295>.
18. Li, H.; Qin, S. A Neurodynamic Approach for Solving Time-Dependent Nonlinear Equation System: A Distributed Optimization Perspective. *IEEE Trans. Ind. Inform.* **2024**, *20*, 10031–10039. <https://doi.org/10.1109/TII.2024.3383508>.
19. Guo, G.; Zhou, Z.; Zhang, R. Distributed Fixed-Time Optimization With Time-Varying Cost: Zero-Gradient-Sum Scheme. *IEEE Trans. Circuits Syst. II Express Briefs* **2024**, *71*, 3086–3090. <https://doi.org/10.1109/TCSII.2024.3355158>.
20. Li, G.; Wang, X.; Li, S. Finite-Time Distributed Approximate Optimization Algorithms of Higher Order Multiagent Systems via Penalty-Function-Based Method. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 6174–6182. <https://doi.org/10.1109/TSMC.2021.3138109>.
21. Shi, X.; Wen, G.; Cao, J.; Yu, X. Finite-Time Distributed Average Tracking for Multiagent Optimization With Bounded Inputs. *IEEE Trans. Autom. Control* **2023**, *68*, 4948–4955. <https://doi.org/10.1109/TAC.2022.3209406>.
22. Song, Y.; Cao, J.; Rutkowski, L. A Fixed-Time Distributed Optimization Algorithm Based on Event-Triggered Strategy. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 1154–1162. <https://doi.org/10.1109/TNSE.2021.3133541>.
23. Wu, C.; Fang, H.; Zeng, X.; Yang, Q.; Wei, Y.; Chen, J. Distributed Continuous-Time Algorithm for Time-Varying Optimization With Affine Formation Constraints. *IEEE Trans. Autom. Control* **2023**, *68*, 2615–2622. <https://doi.org/10.1109/TAC.2022.3190054>.
24. Yang, A.; Liang, X.; Zhang, J.; Hou, Y.; Wang, N. Distributed time-varying optimization with coupled constraints: Application in UAV swarm predefined-time cooperative consensus. *Aerosp. Sci. Technol.* **2024**, *147*, 109034. <https://doi.org/10.1016/j.ast.2024.109034>.
25. He, X.; Li, Y.; Zhang, M.; Huang, T. Distributed Fixed-Time Algorithms for Time-Varying Constrained Optimization Problems. *IEEE Trans. Artif. Intell.* **2025**, *6*, 2656–2668. <https://doi.org/10.1109/TAI.2025.3556095>.

26. Xiao, L.; Jia, L. *Zeroing Neural Networks: Finite-Time Convergence Design, Analysis and Applications*; Wiley-IEEE Press: New York, NY, USA, 2022; pp. 1–432.
27. Wu, Z.; Li, Z.; Yu, J. Designing Zero-Gradient-Sum Protocols for Finite-Time Distributed Optimization Problem. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 4569–4577. <https://doi.org/10.1109/TSMC.2021.3098641>.
28. Hu, Z.; Yang, J. Distributed finite-time optimization for second order continuous-time multiple agents systems with time-varying cost function. *Neurocomputing* **2018**, *287*, 173–184. <https://doi.org/10.1016/j.neucom.2018.01.082>.
29. Rahili, S.; Ren, W. Distributed Continuous-Time Convex Optimization With Time-Varying Cost Functions. *IEEE Trans. Autom. Control* **2017**, *62*, 1590–1605. <https://doi.org/10.1109/TAC.2016.2593899>.
30. Wu, J.; Tang, W.; Lei, W.; Xie, Y.; Ali, T. Free-will arbitrary time optimization for multi-agent systems with time-varying function. *ISA Trans.* **2025**, *162*, 1–10. <https://doi.org/10.1016/j.isatra.2025.04.004>.
31. Ma, L.; Hu, C.; Yu, J.; Wang, L.; Jiang, H. Distributed Fixed/Preassigned-Time Optimization Based on Piecewise Power-Law Design. *IEEE Trans. Cybern.* **2023**, *53*, 4320–4333. <https://doi.org/10.1109/TCYB.2022.3163623>.
32. Hardy, G.H.; Littlewood, J.E.; Pólya, G. *Inequalities*; Cambridge University Press: Cambridge, UK, 1952.
33. Polyakov, A. Nonlinear Feedback Design for Fixed-Time Stabilization of Linear Control Systems. *IEEE Trans. Autom. Control* **2012**, *57*, 2106–2110. <https://doi.org/10.1109/TAC.2011.2179869>.
34. Zhang, Y.; Yi, C.; Guo, D.; Zheng, J. Comparison on Zhang neural dynamics and gradient-based neural dynamics for online solution of nonlinear time-varying equation. *Neural Comput. Appl.* **2011**, *20*, 1–7. <https://doi.org/10.1007/s00521-010-0452-y>.
35. Cao, S.; Jin, J.; Zhang, D.; Chen, C. A review of Zeroing neural network: Theory, algorithm and application. *Neurocomputing* **2025**, *643*, 130425. <https://doi.org/10.1016/j.neucom.2025.130425>.
36. Jin, J.; Zhu, J.; Zhao, L.; Chen, L.; Chen, L.; Gong, J. A Robust Predefined-Time Convergence Zeroing Neural Network for Dynamic Matrix Inversion. *IEEE Trans. Cybern.* **2023**, *53*, 3887–3900. <https://doi.org/10.1109/TCYB.2022.3179312>.
37. Si, Y.; Wang, D.; Chou, Y.; Fu, D. Non-convex activated zeroing neural network model for solving time-varying nonlinear minimization problems with finite-time convergence. *Knowl.-Based Syst.* **2023**, *274*, 110633. <https://doi.org/10.1016/j.knosys.2023.110633>.
38. Jin, J.; Zhu, J.; Zhao, L.; Chen, L. A fixed-time convergent and noise-tolerant zeroing neural network for online solution of time-varying matrix inversion. *Appl. Soft Comput.* **2022**, *130*, 109691. <https://doi.org/10.1016/j.asoc.2022.109691>.
39. Zhang, R.; Guo, G.; Zhou, Z.D. Balance of Communication and Convergence: Predefined-Time Distributed Optimization Based on Zero-Gradient-Sum. *IEEE Trans. Cybern.* **2025**, *55*, 661–671. <https://doi.org/10.1109/TCYB.2024.3498323>.
40. Li, L.; Yang, J.; Kang, Q.; Liu, Y.; Gan, Q. Neurodynamic approaches for distributed constrained optimization problems with time-varying cost functions. *Neurocomputing* **2026**, *686*, 133712. <https://doi.org/10.1016/j.neucom.2026.133712>.
41. Zhang, P.; He, X.; Yu, J. A Distributed Fixed-Time Neurodynamic Algorithm and Its Application in Multi-Autonomous Underwater Vehicle Collaborative Escorting. *IEEE Trans. Netw. Sci. Eng.* **2025**, *12*, 4140–4151. <https://doi.org/10.1109/TNSE.2025.3569300>.
42. Ning, B.; Han, Q.L.; Zuo, Z. Distributed Optimization for Multiagent Systems: An Edge-Based Fixed-Time Consensus Approach. *IEEE Trans. Cybern.* **2019**, *49*, 122–132. <https://doi.org/10.1109/TCYB.2017.2766762>.
43. Deng, C.; Wang, Y.; Wen, C.; Xu, Y.; Lin, P. Distributed Resilient Control for Energy Storage Systems in Cyber-Physical Microgrids. *IEEE Trans. Ind. Informat.* **2021**, *17*, 1331–1341. <https://doi.org/10.1109/TII.2020.2981549>.

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.