frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 17 March 2025
DOI 10.3389/fnbot.2025.1553623

Check for updates

OPEN ACCESS

EDITED BY
Ming-Feng Ge,
China University of Geosciences Wuhan, China

REVIEWED BY
Bolin Liao,
Jishou University, China
Ameer Hamza Khan,
Hong Kong Polytechnic University, Hong Kong SAR, China
Dongsheng Guo,
Hainan University, China
Zhongbo Sun,
Changchun University of Technology, China
Shaofu Yang,
Southeast University, China
Mingzhi Mao,
Sun Yat-sen University, China
Sitian Qin,
Harbin Institute of Technology, China

*CORRESPONDENCE
Hui Cheng
✉ chengh9@mail.sysu.edu.cn

RECEIVED 31 December 2024
ACCEPTED 28 February 2025
PUBLISHED 17 March 2025

CITATION
He L, Cheng H and Zhang Y (2025) A distributed penalty-based zeroing neural network for time-varying optimization with both equality and inequality constraints and its application to cooperative control of redundant robot manipulators.
Front. Neurorobot. 19:1553623.
doi: 10.3389/fnbot.2025.1553623

A distributed penalty-based zeroing neural network for time-varying optimization with both equality and inequality constraints and its application to cooperative control of redundant robot manipulators

Liu He[1], Hui Cheng[1]* and Yunong Zhang[2]

[1]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China, [2]School of Intelligent Systems and Engineering, Sun Yat-sen University, Shenzhen, China

This study addresses the distributed optimization problem with time-varying objective functions and time-varying constraints in a multi-agent system (MAS). To tackle the distributed time-varying constrained optimization (DTVCO) problem, each agent in the MAS communicates with its neighbors while relying solely on local information, such as its own objective function and constraints, to compute the optimal solution. We propose a novel penalty-based zeroing neural network (PB-ZNN) to solve the continuous-time DTVCO (CTDTVCO) problem. The PB-ZNN model incorporates two penalty functions: The first penalizes agents for deviating from the states of their neighbors, driving all agents to reach a consensus, and the second penalizes agents for falling outside the feasible range, ensuring that the solutions of all agents remain within the constraints. The PB-ZNN model solves the CTDTVCO problem in a semi-centralized manner, where information exchange between agents is distributed, but computation is centralized. Building on the semi-centralized PB-ZNN model, we adopt the Euler formula to develop a distributed PB-ZNN (DPB-ZNN) algorithm for solving the discrete-time DTVCO (DTDTVCO) problem in a fully distributed manner. We present and prove the convergence theorems of the proposed PB-ZNN model and DPB-ZNN algorithm. The efficacy and accuracy of the DPB-ZNN algorithm are illustrated through numerical examples, including a simulation experiment applying the algorithm to the cooperative control of redundant manipulators.

KEYWORDS

distributed optimization, zeroing neural network, equality and inequality constraints, time-varying, cooperative control

1 Introduction

Recently, owing to the development of large-scale networks and advancements in big data theory, research on distributed optimization problems has garnered considerable attention due to its broad application prospects in science and engineering (Bahman and Jorge, 2014; Molzahn et al., 2017; Mao et al., 2023; Kang and Yang, 2023), such as resource allocation (Cai et al., 2024), energy management (Li et al., 2024), and economic dispatch

(Huang et al., 2022). In a typical distributed optimization problem, the objective function of a multi-agent system (MAS) results from summing the local sub-objectives of each individual agent (Jia et al., 2024). In addition, during the search for an optimal solution, each agent only has access to its local sub-objective and constraints while communicating its own state to neighboring agents. Overall system convergence is achieved through consensus among the agents.

The aforementioned studies focus on distributed optimization problems with static objective functions. However, distributed time-varying constrained optimization (DTVCO) problems, where the objective functions and constraints change over time, are encountered in practical scenarios, such as the formation control of multi-robot systems (Sun et al., 2022), Nash equilibrium seeking for time-varying non-cooperative games (Ye and Hu, 2015), and distributed time-varying resource allocation (Cherukuri and Cortés, 2016). The DTVCO problem is more intricate than distributed static optimization since the optimal solution evolves over time. As a result, researchers have increasingly turned their attention to DTVCO (Zheng et al., 2017; Sun et al., 2023; Zhang et al., 2021a; Zhu and Wang, 2024; He et al., 2021). For example, Zheng et al. (2017) introduced a consensus-based control scheme employing second-order optimization methods to address DTVCO problems characterized by local objective functions with identical Hessians. Sun et al. (2023), introduced a distributed control algorithm comprising a sliding-mode consensus component and a Hessian-based optimization component, integrated with log-barrier penalty functions, to address the DTVCO problem. Zhu and Wang (2024) utilized the finite-time stability theory and graph theory to propose a novel class of distributed finite-time optimization algorithms that are independent to time derivatives of the gradients and Hessian information.

Due to the parallel computing characteristics, hardware implementation capabilities, and parallel distributed properties (Yang et al., 2020; Jin et al., 2022b; Qiu et al., 2021; Sun et al., 2020; Song et al., 2024), neural network-based methods provide an powerful approach for solving various challenging computational problems in real time. Recent advancements in neural networks have shown significant progress in tackling time-varying problems. For fault-tolerant motion planning of redundant manipulators, Jin et al. (2017) developed a different-level simultaneous minimization scheme by utilizing a discrete-time recurrent neural network to solve quadratic programming problems. Khan et al. (2020b) have developed a metaheuristic-based control framework for simultaneous tracking control and obstacle avoidance of redundant manipulators, unifying these tasks into a single constrained optimization problem. Enhancements to the beetle antennae search algorithm, called BAS-ADAM, have been proposed by Khan et al. (2020a) to improve convergence behavior and avoid local minima in highly non-convex objective functions. A distributed competitive and collaborative coordination approach has been established by (Liu et al., 2024) for multi-robot systems, optimizing system stability and resource utilization through a fusion of recurrent neural dynamics and distributed solvers. In addition, Zhang et al. (2018a) have proposed a novel recurrent neural network for kinematic control of redundant manipulators, addressing periodic input disturbances and physical constraints while optimizing a general quadratic performance index. These studies collectively highlight the innovation and potential of neural networks in addressing time-varying challenges in various applications. Another exciting direction in neural network-based control scheme is the data-driven model predictive control (MPC). The data-driven MPC impacted the field of robotic control, particularly in handling complex and dynamic environments. For example, one study by Yan et al. (2024) proposed a data-driven MPC algorithm that integrates neural dynamics for trajectory tracking in redundant manipulators with unknown models. Another research by Jin et al. (2024) introduced a cerebellum-inspired learning and control scheme using echo state networks to achieve precise joint velocity control in redundant manipulators.

Introduced by Zhang et al. (2002), since its inception, zeroing neural network (ZNN) method has been widely utilized to address a range of time-varying problems (Liao et al., 2022; Peng et al., 2022; Lan et al., 2023; Zuo et al., 2022; Dai et al., 2023; Chen et al., 2024; Hu et al., 2024; Jin et al., 2022a; Tan et al., 2023; Ding et al., 2018), such as time-varying matrix inversion (Dai et al., 2023), time-varying optimization (Chen et al., 2024), and time-varying Lyapunov equation solving (Zuo et al., 2022). Recent advancements in ZNN have illustrated their effectiveness in various robotic applications. (Sun et al., 2024b) have successfully applied ZNN to human-robot interaction and force control, improving estimation accuracy and ensuring safety in complex environments. In the realm of motion estimation, (Wang et al., 2024) have integrated ZNN into a multi-task parallel learning framework to reconstruct missing sEMG signals and estimate joint angles with high accuracy. In addition, (Xie et al., 2025) have utilized ZNN in obstacle avoidance schemes for redundant robots, leveraging deep reinforcement learning to enhance obstacle avoidance capabilities. Furthermore, Sun et al. (2024a) have developed a hybrid orthogonal repetitive motion and obstacle avoidance scheme for omnidirectional mobile robotic arms, achieving accurate obstacle avoidance and repetitive motion tasks. These studies collectively highlight the innovation and potential of ZNN in revolutionizing various aspects of robotics.

Despite the significant success of ZNN method in addressing time-varying problems, the ZNN models presented in the aforementioned works are limited to solve those problems in a centralized manner. There is a scarcity of existing research focusing on distributed ZNN. The most typical work of distributed ZNN is by Jin et al. In Jin et al. (2018), a ZNN-based control scheme is introduced to address the cooperative motion generation problem. This scheme facilitates cooperative motion generation within a distributed network of multiple redundant manipulators. However, for an individual manipulator, the computation results of neighbors' control inputs are required to compute its own control input. Therefore, the proposed control method is not a fully distributed scheme, as the control inputs for the entire system need to be computed collectively by all manipulators, rather than being computed independently by each manipulator.

Driven by the above discussions, a novel penalty-based ZNN (PB-ZNN) model is proposed. The PB-ZNN model solves a continuous-time DTVCO (CTDTVCO) problem in a semi-centralized manner. Then, to extend the application of ZNN models to distributed time-varying problems, a distributed penalty-based ZNN (DPB-ZNN) algorithm is designed on the basis of the proposed PB-ZNN model. The DPB-ZNN algorithm solves a discrete-time DTVCO (DTDTVTO) problem in a fully distributed manner.

The remainder of this study is organized into six sections. In Section 2, some preliminaries concerning graph theory and mathematical notations are provided. Then, in Section 3, the main results of this study are presented. The formulation of the CTDTVCO problem is presented in Section 3.1, and the detailed design process of PB-ZNN model is presented in Section 3.2. The formulation for the DTDTVCO problem and the detailed design process of DPB-ZNN algorithm is presented in Section 3.3. In Section 4, the theoretical proofs of the convergence of both PB-ZNN model and DPB-ZNN algorithm are provided. The efficacy and efficiency of the DPB-ZNN algorithm are illustrated in Section 5 through numerical examples, including a simulation experiment applying it to the cooperative control of redundant manipulators. Finally, the conclusion of this study is given in Section 6.

Before ending this section, the main contributions of this study are listed as follows.

- A novel PB-ZNN model is designed to address the CTDTVCO problem. By incorporating two penalty functions, the PB-ZNN model successfully solves the CTDTVCO problem in a semi-centralized manner.
- On the basis of the PB-ZNN model, a DPB-ZNN algorithm is designed. By incorporating the Euler formula, the proposed DPB-ZNN algorithm effectively solves the DTDTVCO problem in a fully distributed manner.
- The convergence properties of both PB-ZNN model and DPB-ZNN algorithm are proved. Further theoretical analyses prove that the maximum steady-state residual error (MSSRE) for the proposed DPB-ZNN algorithm is of $O(\tau^2)$.

2 Preliminaries

The notation $\|\cdot\|_2$ represents the 2-norm of a vector or a matrix. Moreover, $(\cdot)^{\mathrm{T}}$ denotes the transpose of a matrix or a vector, I_n is the identity matrix of size $n \times n$, $\mathbf{1}_d = \{1, \cdots, 1\}^{\mathrm{T}} \in \mathbb{R}^d$, and $\otimes$ represents the Kronecker product (Golub and Loan, 2013).

Some principles from graph theory (Diestel, 2017) are introduced as follows.

The notation $\mathcal{G} = \{\mathcal{V}, \mathcal{Y}, A\}$ denotes a weighted graph, wherein $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ represents the set of vertices, and $\mathcal{Y} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges. The adjacency matrix is denoted by $A = (a_{ij})_{n\times n} \in \mathbb{R}^{n\times n}$, where $a_{ij} > 0$ when $(v_i, v_j) \in \mathcal{Y}$ and $a_{ij} = 0$ when $(v_i, v_j) \notin \mathcal{Y}$, respectively. Moreover, the notation $\mathcal{N}_i$ denotes the neighbor set of ith vertex. The graph $\mathcal{G}$ is said to be connected and undirected when a path between any given pair of vertices within $\mathcal{G}$ exists and when $(v_i, v_j) \in \mathcal{Y}$ also indicates $(v_j, v_i) \in \mathcal{Y}$. The degree matrix D is defined as $D = \mathrm{diag}\{d_1, d_2, \cdots, d_n\} \in \mathbb{R}^{n\times n}$ where $d_i = \sum_{j=1}^{n} a_{ij}$. Finally, the Laplacian matrix is defined as $L = D - A$.

Assumption 1: Graph $\mathcal{G}$ is undirected and connected.

3 Main results

In Subsection 3.1, the formulations of the CTDTVCO problem are provided. To address the CTDTVCO problem, two penalty functions are introduced. In Subsection 3.2, a PB-ZNN model is proposed to solve the CTDTVCO problem in a semi-centralized manner. Then, on the basis of the proposed PB-ZNN model, a novel DPB-ZNN algorithm is proposed to solve the DTDTVCO problem in a fully distributed manner in Subsection 3.3.

3.1 CTDTVCO with distributed time-varying equality and inequality constraints

In this study, the CTDTVCO problem within graph $\mathcal{G}$ is formulated as

$$\text{minimize} \quad f(x(t), t) = \sum_{i=1}^{n} f_i(x(t), t), \tag{1}$$

$$\text{subject to} \quad K_i(t)x(t) = b_i(t), \quad i \in \{1, 2, \ldots, n\}, \tag{2}$$

$$J_i(t)x(t) \le c_i(t), \quad i \in \{1, 2, \ldots, n\}, \tag{3}$$

where $x(t) \in \mathbb{R}^d$, and $f_i(x(t), t) : \mathbb{R}^d \times \mathbb{R} \mapsto \mathbb{R}$ denotes the time-varying local objective function of the agent i and f_i should be second-order differentiable and strongly convex for all $i = 1, 2, \cdots, n$. Moreover, $K_i(t)x(t) = b_i(t)$ and $J_i(t)x(t) \le c_i(t)$ are the time-varying equality and inequality constraints associated with the agent i. For all $i \in \{1, 2, \ldots, n\}$, the coefficient matrix $K_i(t) \in \mathbb{R}^{k_i\times d}$ is of full row rank and coefficient matrix $J_i(t)$ is defined as $J_i(t) = [J_{i1}(t); \ldots; J_{im_i}(t)] \in \mathbb{R}^{m_i\times d}$. In addition, notations $b_i(t) \in \mathbb{R}^{k_i}$ and $c_i(t) \in \mathbb{R}^{m_i}$ denote the coefficient vectors. The above-mentioned matrices and vectors are assumed to be differentiable. It is worth pointing out that all contradictions among agents' equality constraints should be ruled out for the CTDTVCO problem to be solvable.

According to Assumption 1, the communication graph $\mathcal{G}$ is both connected and undirected. Hence, the CTDTVCO problem (Equations 1–3) is reformulated as an equivalent problem described by the following Lemma 1.

Lemma 1: If graph $\mathcal{G}$ is connected and undirected, then the CTDTVCO problem (Equations 1–3) is equivalent to the following optimization problem (Bahman and Jorge, 2014).

$$\text{minimize} \quad \varphi(\mathbf{x}(t), t) = \sum_{i=1}^{n} f_i(x_i(t), t), \tag{4}$$

$$\text{subject to} \quad \mathbf{J}(t)\mathbf{x}(t) \le \mathbf{c}(t), \tag{5}$$

$$\mathbf{K}(t)\mathbf{x}(t) = \mathbf{b}(t), \tag{6}$$

$$\mathbf{L}\mathbf{x}(t) = \mathbf{0}_{nd}, \tag{7}$$

where

$$\mathbf{K}(t) = \begin{bmatrix} K_1(t) & 0 & \cdots & 0 \\ 0 & K_2(t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K_n(t) \end{bmatrix} \in \mathbb{R}^{k\times nd},$$

$$\mathbf{J}(t) = \begin{bmatrix} J_1(t) & 0 & \cdots & 0 \\ 0 & J_2(t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & J_n(t) \end{bmatrix} \in \mathbb{R}^{m\times nd},$$

with $\mathbf{x}(t) = [x_1(t); x_2(t); \ldots; x_n(t)] \in \mathbb{R}^{nd}$, $k = \sum_{i=1}^{n} k_i$, and $m = \sum_{i=1}^{n} m_i$. The coefficient vectors $\mathbf{b}(t)$ and $\mathbf{c}(t)$ are defined as $\mathbf{b}(t) = [b_1(t); b_2(t); \ldots; b_n(t)] \in \mathbb{R}^k$, and $\mathbf{c}(t) = [c_1(t); c_2(t); \ldots; c_n(t)] \in \mathbb{R}^m$, and the matrix $\mathbf{L} = L \otimes I_d \in \mathbb{R}^{nd\times nd}$. In addition, $\mathbf{0}_{nd}$ denotes an nd dimensional vector whose elements are 0.

Remark 1: Through Lemma 1, the consensus problem for the CTDTVCO problem (Equations 1–3) is reformulated as an equality constraint (Equation 7). Following the conventional ZNN design method, equality constraints are typically handled by applying the Lagrange function. However, this approach is not suitable for equality constraint (Equation 7), because the Laplacian matrix L is inherently rank-deficient. Consequently, setting $\mathbf{L} = L \otimes I_d$ also results in a rank-deficient matrix.

To obtain the optimal solution for the CTDTVCO problem (Equations 1–3), two penalty functions are introduced in this study. The definitions of these two penalty functions are as follows:

$$p_1(\mathbf{x}(t)) = \frac{1}{4}\sigma_1 \sum_{i=1}^{n} \sum_{j\in\mathcal{N}_i} \|x_i(t) - x_j(t)\|_2^2, \tag{8}$$

$$p_2(\mathbf{x}(t)) = \rho \sum_{i=1}^{n} \sum_{j=1}^{m_i} \mathrm{e}^{-\sigma_2 N_j(x_i(t))}, \tag{9}$$

where $N_j(x_i(t)) = c_{ij}(t) - J_{ij}(t)x_i(t)$ with $c_{ij}(t)$ being the jth element of vector $c_i(t)$ and $J_{ij}(t)$ being the jth row of matrix $J_i(t)$. In addition, $\sigma_1, \sigma_2 > 0$ are two positive parameters that are sufficiently larege and parameter $\rho > 0$ is positive and near zero.

- The first penalty function intuitively manages information exchange between agents and their neighbors, driving all agents to reach a consensus.
- The second penalty function is from Zhang et al. (2021b), and it addresses the distributed inequality constraints, ensuring that each agent's solution remains bounded.

It is evident that when $\mathbf{L}\mathbf{x}(t) \rightarrow \mathbf{0}_{nd}$, the value of $p_1(\mathbf{x}(t))$ approaches zero. Furthermore, when inequality constraints (Equation 5) are met, the value of $p_2(\mathbf{x}(t))$ becomes a positive number very close to zero. This soft penalty allows slight violations of constraints while still promoting feasible solutions. Conversely, by choosing sufficiently large enough positive parameters σ_1 and σ_2, the values of $p_1(\mathbf{x}(t))$ and $p_2(\mathbf{x}(t))$ are magnified to serve as punishments for solution $\mathbf{x}(t)$ that fall outside the feasible range. Thus, the CTDTVCO problem (Equations 4–6) is transformed into an equivalent CTDTVCO problem with no inequality constraints as follows:

$$\text{minimize} \quad \varphi(\mathbf{x}(t), t) + p_1(\mathbf{x}(t)) + p_2(\mathbf{x}(t)), \tag{10}$$

$$\text{subject to} \quad \mathbf{K}(t)\mathbf{x}(t) = \mathbf{b}(t). \tag{11}$$

Therefore, by solving the CTDTVCO problem (Equations 10, 11), one acquires an approximate solution to the original CTDTVCO problem (Equations 1–3). To solve the CTDTVCO problem (Equations 10, 11), a Lagrange function is introduced:

$$\begin{aligned} L(\mathbf{x}(t), \boldsymbol{\lambda}(t), t) \; = \; & \varphi\left(\mathbf{x}(t), t\right) + p_1\left(\mathbf{x}(t)\right) + p_2(\mathbf{x}(t)) \\ & + \boldsymbol{\lambda}^{\mathrm{T}}(t)\left(\mathbf{K}(t)\mathbf{x}(t) - \mathbf{b}(t)\right), \end{aligned} \tag{12}$$

where the Lagrange multiplier $\boldsymbol{\lambda}(t)$ is defined as $\boldsymbol{\lambda}(t) = [\lambda_1(t); \lambda_2(t); \ldots; \lambda_n(t)] \in \mathbb{R}^k$, with $\lambda_i(t) \in \mathbb{R}^{k_i}$ being the Lagrange multiplier corresponding to the agent i. It is assumed that both $\partial L(\mathbf{x}(t), \boldsymbol{\lambda}(t), t)/\partial\mathbf{x}(t)$ and $\partial L(\mathbf{x}(t), \boldsymbol{\lambda}(t), t)/\partial\boldsymbol{\lambda}(t)$ exist and are continuous. The optimal solution must satisfy the following equations:

$$\begin{cases} \frac{\partial L(\mathbf{x}(t),\boldsymbol{\lambda}(t),t)}{\partial\mathbf{x}(t)} = \nabla_x\varphi(\mathbf{x}(t), t) + \sigma_1\mathbf{L}\mathbf{x}(t) + \Phi(\mathbf{x}(t)) + \mathbf{K}^{\mathrm{T}}(t)\boldsymbol{\lambda}(t) = \mathbf{0}_{nd}, \\ \frac{\partial L(\mathbf{x}(t),\boldsymbol{\lambda}(t),t)}{\partial\boldsymbol{\lambda}(t)} = \mathbf{K}(t)\mathbf{x}(t) - \mathbf{b}(t) = \mathbf{0}_k, \end{cases} \tag{13}$$

where

$$\begin{aligned} \nabla_x\varphi(\mathbf{x}(t), t) \; = \; & \frac{\partial\varphi(\mathbf{x}(t))}{\partial\mathbf{x}(t)} = [\nabla_x f_1(x_1(t), t); \nabla_x f_2(x_2(t), t); \ldots; \\ & \nabla_x f_n(x_n(t), t)] \in \mathbb{R}^{nd}, \end{aligned}$$

with the elements being

$$\nabla_x f_i(x_i(t), t) = \frac{\partial f_i(x_i(t), t)}{\partial x_i(t)} \in \mathbb{R}^d.$$

In addition, $\Phi(\mathbf{x}(t))$ is defined as

$$\Phi(\mathbf{x}(t)) = \frac{\partial p_2(\mathbf{x}(t))}{\partial\mathbf{x}(t)} = [\phi_1(x_1(t)); \phi_2(x_2(t)); \ldots; \phi_n(x_n(t))] \in \mathbb{R}^{nd},$$

with

$$\phi_i(x_i(t)) = \rho\sigma_2 \sum_{j=1}^{m_i} \mathrm{e}^{-\sigma_2 N_j(x_i(t))} J_{ij}^{\mathrm{T}}(t) \in \mathbb{R}^d.$$

For convenience in computation, (Equation 13) is expressed as the following matrix equation:

$$\mathbf{A}(t)\mathbf{y}(t) = \mathbf{g}(t), \tag{14}$$

where

$$\mathbf{A}(t) = \begin{bmatrix} \sigma_1\mathbf{L} & \mathbf{K}^{\mathrm{T}}(t) \\ \mathbf{K}(t) & \mathbf{0}_{k\times k} \end{bmatrix} \in \mathbb{R}^{(nd+k)\times(nd+k)},$$

$$\mathbf{y}(t) = \begin{bmatrix} \mathbf{x}(t) \\ \boldsymbol{\lambda}(t) \end{bmatrix} \in \mathbb{R}^{(nd+k)},$$

$$\mathbf{g}(t) = \begin{bmatrix} -\nabla_x\varphi(\mathbf{x}(t), t) - \Phi(\mathbf{x}(t)) \\ \mathbf{b}(t) \end{bmatrix} \in \mathbb{R}^{(nd+k)}.$$

The vector $\mathbf{y}(t)$ needs to be solved at all time. One sees that the CTDTVCO problem (Equations 1–3) is solved if the matrix equation (Equation 14) is solved.

3.2 PB-ZNN Model for CTDTVCO Problem Solving

In this subsection, a PB-ZNN model is proposed for the entire system to solve the matrix equation (Equation 14).

To obtain the solution of (Equation 14), an error function is defined as

$$\boldsymbol{\varepsilon}(t) = \mathbf{A}(t)\mathbf{y}(t) - \mathbf{g}(t) \in \mathbb{R}^{nd+k}. \tag{15}$$

To minimize the value of the error function $\boldsymbol{\varepsilon}(t)$ and approach zero, it is essential that the derivative of the error function (Equation 15) with respect to time t remains negative (Zhang et al., 2018b). Consequently, the ZNN design formula is employed as follows:

$$\frac{\mathrm{d}\boldsymbol{\varepsilon}(t)}{\mathrm{d}t} = -\gamma\Psi(\boldsymbol{\varepsilon}(t)), \tag{16}$$

where $\Psi(\cdot): \mathbb{R}^{nd+k} \mapsto \mathbb{R}^{nd+k}$ is an array consisting of activation functions $\psi(\cdot)$ and a positive parameter $\gamma > 0$ is used to adjust the convergence rate. It is crucial that the activation function $\psi(\cdot)$ satisfies two properties: It must be monotonically increasing and an odd function. By substituting (Equation 16) into (Equation 15), the following model is obtained:

$$\mathbf{A}(t)\dot{\mathbf{y}}(t) = -\dot{\mathbf{A}}(t)\mathbf{y}(t) - \gamma\Psi(\mathbf{A}(t)\mathbf{y}(t) - \mathbf{g}(t)) - \dot{\mathbf{g}}(t), \tag{17}$$

where $\dot{\mathbf{A}}(t)$ and $\dot{\mathbf{g}}(t)$ denote the derivatives with respect to time t of the matrix $\mathbf{A}(t)$ and the vector $\mathbf{g}(t)$, respectively.

Moreover, according to Equation 14, one obtains

$$\dot{\mathbf{g}}(t) = \begin{bmatrix} \dot{\boldsymbol{\delta}}(t) \\ \dot{\mathbf{b}}(t) \end{bmatrix} \in \mathbb{R}^{nd+k}, \tag{18}$$

where

$$\dot{\boldsymbol{\delta}}(t) = -\dot{\mathbf{c}}(t) - \mathbf{H}(\mathbf{x}(t), t)\dot{\mathbf{x}}(t) - \dot{\Phi}(\mathbf{x}(t)) \in \mathbb{R}^{nd}, \tag{19}$$

with $\dot{\mathbf{b}}(t) = \mathrm{d}\mathbf{b}(t)/\mathrm{d}t$ and

$$\dot{\Phi}(\mathbf{x}(t)) = \left[\frac{\partial^2 p_{21}(x_1(t))}{\partial x_1(t)\partial t}; \frac{\partial^2 p_{22}(x_1(t))}{\partial x_2(t)\partial t}; \ldots; \frac{\partial^2 p_{2n}(x_1(t))}{\partial x_n(t)\partial t}\right] \in \mathbb{R}^{nd},$$

in which

$$\begin{aligned} \frac{\partial^2 p_{2i}(x_1(t))}{\partial x_i(t)\partial t} &= \rho\sigma_2 \sum_{j=1}^{m_i} \mathrm{e}^{-\sigma_2 N_j(x_i(t))}(\sigma_2 J_{ij}^{\mathrm{T}}(t)\dot{J}_{ij}(t)x_i(t) \\ &+ \sigma_2 J_{ij}^{\mathrm{T}}(t)J_{ij}(t)\dot{x}_i(t) - \sigma_2 J_{ij}^{\mathrm{T}}(t)\dot{c}_{ij} + \dot{J}_{ij}^{\mathrm{T}}(t)), \end{aligned}$$

with $\dot{J}_{ij}(t) = \mathrm{d}J_{ij}(t)/\mathrm{d}t$ and $\dot{c}_{ij}(t) = \mathrm{d}c_{ij}/\mathrm{d}t$. In addition,

$$\mathbf{H}(\mathbf{x}(t), t) = \begin{bmatrix} H_1(x_1(t), t) & 0 & \cdots & 0 \\ 0 & H_2(x_2(t), t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H_n(x_n(t), t) \end{bmatrix} \in \mathbb{R}^{nd \times nd},$$

where $H_i(x_i(t), t) \in \mathbb{R}^{d \times d}$ for all $i = 1, 2, \ldots, n$ denotes the Hessian matrix of the agent i and the definition of $H_i(x_i(t), t)$ is as follows:

$$H_i(x_i(t), t) = \begin{bmatrix} \frac{\partial^2 f_i(x_i(t),t)}{\partial x_{i1}^2} & \frac{\partial^2 f_i(x_i(t),t)}{\partial x_{i1}\partial x_{i2}} & \cdots & \frac{\partial^2 f_i(x_i(t),t)}{\partial x_{i1}\partial x_{id}} \\ \frac{\partial^2 f_i(x_i(t),t)}{\partial x_{i2}\partial x_{i1}} & \frac{\partial^2 f_i(x_i(t),t)}{\partial x_{i2}^2} & \cdots & \frac{\partial^2 f_i(x_i(t),t)}{\partial x_{i2}\partial x_{i3}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f_i(x_i(t),t)}{\partial x_{id}\partial x_{i1}} & \frac{\partial^2 f_i(x_i(t),t)}{\partial x_{id}\partial x_{i2}} & \cdots & \frac{\partial^2 f_i(x_i(t),t)}{\partial x_{id}^2} \end{bmatrix} \in \mathbb{R}^{d \times d}.$$

One notices the simultaneous presence of $\mathbf{x}(t)$ and $\dot{\mathbf{x}}(t)$ in $\dot{\boldsymbol{\delta}}(t)$ in Equation 19. This reflects the combined influence of both the state and its time derivative on the dynamics of the system. When numerically computing Equation 17, it becomes necessary to consolidate similar terms. Define two matrices, $\mathbf{M}_1(t)$ and $\mathbf{M}_2(t)$, and a vector $\mathbf{h}(t)$, as follows:

$$\mathbf{M}_1(t) = \begin{bmatrix} M_{11}(t) & 0 & \cdots & 0 \\ 0 & M_{12}(t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & M_{1n}(t) \end{bmatrix} \in \mathbb{R}^{nd \times nd},$$

$$\mathbf{M}_2(t) = \begin{bmatrix} M_{21}(t) & 0 & \cdots & 0 \\ 0 & M_{22}(t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & M_{2n}(t) \end{bmatrix} \in \mathbb{R}^{nd \times nd},$$

$$\mathbf{h}(t) = [h_1; h_2; \ldots; h_n] \in \mathbb{R}^{nd},$$

where for all $i = 1, 2, \ldots, n$,

$$M_{1i} = \rho\sigma_2^2 \sum_{j=1}^{m_i} \mathrm{e}^{-\sigma_2 N_j(x_i(t))} J_{ij}^{\mathrm{T}}(t)J_{ij}(t) \in \mathbb{R}^{d \times d},$$

$$M_{2i} = \rho\sigma_2^2 \sum_{j=1}^{m_i} \mathrm{e}^{-\sigma_2 N_j(x_i(t))} J_{ij}^{\mathrm{T}}(t)\dot{J}_{ij}(t) \in \mathbb{R}^{d \times d},$$

$$h_i = \rho\sigma_2 \sum_{j=1}^{m_i} \mathrm{e}^{-\sigma_2 N_j(x_i(t))}(-\sigma_2 J_{ij}^{\mathrm{T}}(t)\dot{c}_{ij} + \dot{J}_{ij}^{\mathrm{T}}(t)) \in \mathbb{R}^{d},$$

Hence, after a restructuring of the ZNN model (Equation 17), one has

$$\mathbf{Q}(t)\dot{\mathbf{y}}(t) = -\mathbf{S}(t)\mathbf{y}(t) - \gamma\Psi(\mathbf{A}(t)\mathbf{y}(t) - \mathbf{g}(t)) + \mathbf{u}(t),$$

where

$$\mathbf{Q}(t) = \begin{bmatrix} \mathbf{H}(\mathbf{x}(t), t) + \sigma_1\mathbf{L} + \mathbf{M}_1(t) & \mathbf{K}^{\mathrm{T}}(t) \\ \mathbf{K}(t) & \mathbf{0}_{k \times k} \end{bmatrix} \in \mathbb{R}^{(nd+k)\times(nd+k)},$$

$$\mathbf{S}(t) = \begin{bmatrix} \mathbf{M}_2(t) & \dot{\mathbf{K}}^{\mathrm{T}}(t) \\ \dot{\mathbf{K}}(t) & \mathbf{0}_{k \times k} \end{bmatrix} \in \mathbb{R}^{(nd+k)\times(nd+k)},$$

$$\mathbf{u}(t) = \begin{bmatrix} -\mathbf{h}(t) - \nabla_{xt}\varphi(\mathbf{x}(t), t) \\ \dot{\mathbf{b}}(t) \end{bmatrix} \in \mathbb{R}^{nd+k},$$

with $\nabla_{xt}\varphi(\mathbf{x}(t), t) = [\partial^2 f_1(x_1(t), t)/\partial x_1\partial t; \partial^2 f_2(x_2(t), t)/\partial x_2\partial t; \ldots; \partial^2 f_n(x_n(t), t)/\partial x_n\partial t]$. In this study, the linear function (i.e., $\Psi(x(t)) = x(t)$) is chosen as the activation function for simplicity. Therefore, the PB-ZNN model for the whole system is given as

$$\dot{\mathbf{y}}(t) = -\mathbf{Q}^{-1}(t)\left(\mathbf{S}(t)\mathbf{y}(t) - \gamma\left(\mathbf{A}(t)\mathbf{y}(t) - \mathbf{g}(t)\right) + \mathbf{u}(t)\right). \tag{20}$$

Thus, the design of PB-ZNN model that solves the CTDTVCO problem (Equations 1–3) is completed.

It is worth pointing out that, although the PB-ZNN model (Equation 20) is designed to address the CTDTVCO problem, it solves Equation 14 in a semi-decentralized manner. Upon examining matrix $\mathbf{Q}(t)$ in Equation 20, one observes that for agent i, due to the presence of $\mathbf{L}$ in $\mathbf{Q}(t)$, solving the time derivative $\dot{x}_i(t)$

analytically requires information from its neighbors: $\dot{x}_j(t)$ for $j \in \mathcal{N}_i$. Therefore, if to be analytically solved, the information exchange among agents is distributed, but the PB-ZNN model (Equation 20) is computed in a centralized way.

3.3 Distributed PB-ZNN algorithm for DTDTVCO problem solving

In this subsection, a distributed PB-ZNN (DPB-ZNN) algorithm is developed for each agent to solve the DTDTVCO problem in a fully distributed manner.

First, let us consider the following DTDTVCO problem with a computational time interval $[t_k, t_{k+1})$:

$$\text{minimize} \quad f(x_{k+1}, t_{k+1}) = \sum_{i=1}^{n} f_i(x_{k+1}, t_{k+1}), \tag{21}$$

$$\text{subject to} \quad K_{i_{k+1}} x_{k+1} = b_{i_{k+1}}, i \in \{1, 2, \ldots, n\}, \tag{22}$$

$$J_{i_{k+1}} x_{k+1} \leq c_{i_{k+1}}, i \in \{1, 2, \ldots, n\}, \tag{23}$$

where $f(x_{k+1}, t_{k+1})$ is generated or measured from the smoothly time-varying signal $f(x(t), t)$ by sampling at the time instant $t = (k+1)\tau$ (which is denoted as t_{k+1}), and τ denotes the sampling gap.

The DPB-ZNN algorithm is designed on the basis of the continuous-time PB-ZNN model (Equation 20). Therefore, the distributed form of PB-ZNN model (Equation 20) is given to lay the basis for the DPB-ZNN algorithm.

Through simple matrix computation, the distributed form of equation (Equation 14) for agent i to solve in continuous-time is given as

$$A_i(t) y_i(t) = g_i(t), \tag{24}$$

where

$$A_i(t) = \begin{bmatrix} \sigma_1 D_i & K_i^{\mathrm{T}}(t) \\ K_i(t) & \mathbf{0}_{k_i \times k_i} \end{bmatrix} \in \mathbb{R}^{(d+k_i)\times(d+k_i)},$$

$$y_i(t) = \begin{bmatrix} x_i(t) \\ \lambda_i(t) \end{bmatrix} \in \mathbb{R}^{d+k_i},$$

$$g_i(t) = \begin{bmatrix} -\nabla_x f_i(x_i(t), t) - \phi(x_i(t)) + \sigma_1 e_i(t) \\ b_i(t) \end{bmatrix} \in \mathbb{R}^{d+k_i},$$

with $D_i = d_i I_{d\times d}$ and d_i denotes the degree of the Laplacian matrix L associated with the agent i. In addition, $e_i(t) = \sum_{j\in\mathcal{N}_i} a_{ij} x_j(t)$. It is evident that the distributed solution $[\dot{x}_1(t); \dot{x}_2(t); \ldots; \dot{x}_n(t)]$ for equation (Equation 24) is equivalent to $\dot{\mathbf{x}}(t)$ in equation (Equation 14).

Consequently, for the agent i, the distributed form of the error function is defined as

$$\varepsilon_i(t) = A_i(t) y_i(t) - g_i(t) \in \mathbb{R}^{d+k_i}. \tag{25}$$

Therefore, following the same design process as the PB-ZNN model, one has

$$Q_i(t)\dot{y}_i(t) = -S_i(t) y_i(t) - \gamma \Psi(A_i(t) y_i(t) - g_i(t)) + u_i(t), \tag{26}$$

where

$$Q_i(t) = \begin{bmatrix} H_i(x_i(t)) + \sigma_1 D_i + M_{1i}(t) & K_i^{\mathrm{T}}(t) \\ K_i(t) & \mathbf{0}_{k_i \times k_i} \end{bmatrix} \in \mathbb{R}^{(d+k_i)\times(d+k_i)},$$

$$S_i(t) = \begin{bmatrix} M_{2i}(t) & \dot{K}_i^{\mathrm{T}}(t) \\ \dot{K}_i(t) & \mathbf{0}_{k_i \times k_i} \end{bmatrix} \in \mathbb{R}^{(d+k_i)\times(d+k_i)},$$

$$u_i(t) = \begin{bmatrix} -h_i(t) - \dot{\nabla}_x f_i(x_i(t), t) + \sigma_1 \dot{e}_i(t) \\ \dot{b}_i(t) \end{bmatrix} \in \mathbb{R}^{d+k_i},$$

with $\dot{e}_i(t) = \sum_{j\in\mathcal{N}_i} a_{ij} \dot{x}_j(t)$. In addition, $M_{1i}(t), M_{2i}(t)$, and $h_i(t)$ are defined in the same way as in Section 3.2. Therefore, for agent i, the distributed form of PB-ZNN model (Equation 20) is formulated as

$$\begin{aligned} \dot{y}_i(t) = & -\gamma Q_i^{-1}(t) \Psi\left(A_i(t) y_i(t) - g_i(t)\right) - Q_i^{-1}(t) S_i(t) y_i(t) \\ & + Q_i^{-1}(t) u_i(t). \end{aligned}$$

Hence, the distributed discrete-time PB-ZNN model is formulated as

$$\dot{y}_{i_k} = -\gamma Q_{i_k}^{-1} \Psi\left(A_{i_k} y_{i_k} - g_{i_k}\right) - Q_{i_k}^{-1} S_{i_k} y_i + Q_{i_k}^{-1} u_{i_k}, \tag{27}$$

where $Q_{i_k}, A_{i_k}, S_{i_k}, g_{i_k}$, and u_{i_k} are generated or measured from the smoothly time-varying signals $Q_i(t), A_i(t), S_i(t), g_i(t)$, and $u_i(t)$, respectively. Upon examining u_{i_k}, one notices that $\dot{e}_{i_k} = \sum_{j\in\mathcal{N}_i} a_{ij} \dot{x}_{j_k}$ is required for agent i to compute $\dot{x}_{i_k}$. However, $\dot{x}_{j_k}$ is unknown to agent i in a fully distributed manner. Therefore, an approximation for $\dot{e}_{i_k}$ is introduced.

In traditional methods that solve distributed optimization problems, agents exchange x_{i_k} with their neighbors to address the consensus problem. In this study, an alternative approach is proposed. Instead of merely exchanging the information x_{i_k}, each agent maintains a short memory of their neighbors' states. By using the Euler formula, agents approximate $\dot{x}_{j_k}$ for $j \in \mathcal{N}_i$ on the basis of the memories that they kept, and ultimately approximate $\dot{e}_{i_k}$ effectively.

The Euler formula used in this study is given as follows (Chen et al., 2024).

$$\dot{x}_{i_k} = \dot{\tilde{x}}_{i_k} + O(\tau) = \frac{x_{i_k} - x_{i_{k-1}}}{\tau} + O(\tau). \tag{28}$$

This formula approximates the derivative $\dot{x}_{i_k}$ using the backward difference of x_{i_k} over the time step τ. The term $O(\tau)$ represents the truncation error, indicating that the approximation becomes more accurate as τ approaches zero. Therefore, the approximation of $\dot{e}_i(t)$ is defined as

$$\dot{e}_{i_k} = \dot{\tilde{e}}_{i_k} + O(\tau) = \sum_{j\in\mathcal{N}_i} \dot{\tilde{x}}_{j_k} + O(\tau). \tag{29}$$

Here, $\dot{e}_{i_k}$ is approximated by summing the estimated derivatives of the neighbors' states $\dot{\tilde{x}}_{j_k}$, utilizing the stored state information to eliminate the need for continuous communication. Hence, the fully distributed DPB-ZNN algorithm that solves the DTDTVCO problem is given as

$$\dot{y}_{i_k} \doteq -\gamma Q_{i_k}^{-1} \Psi\left(A_{i_k} y_{i_k} - g_{i_k}\right) - Q_{i_k}^{-1} S_{i_k} y_i + Q_{i_k}^{-1} \tilde{u}_{i_k},$$

with

$$\tilde{u}_{i_k} = \begin{bmatrix} -h_{i_k} - \dot{\nabla}_x f_i(x_{i_k}, t_k) + \sigma_1 \dot{\tilde{e}}_{i_k} \\ \dot{b}_{i_k} \end{bmatrix}.$$

Hence, by choosing the linear function as the active function, one obtains

$$\dot{y}_{i_k} \doteq -\gamma Q_{i_k}^{-1}(A_{i_k} y_{i_k} - g_{i_k}) - Q_{i_k}^{-1} S_{i_k} y_{i_k} + Q_{i_k}^{-1} \tilde{u}_{i_k}.$$

By introducing the 2-step time-discretization (TD) formula (Chen et al., 2024).

$$\dot{\xi}_k = \frac{1}{\tau}\left(\xi_{k+1} - \xi_k\right) + O(\tau), \tag{30}$$

the DPB-ZNN algorithm is obtained as

$$y_{i_{k+1}} \doteq y_{i_k} - hQ_{i_k}^{-1}(A_{i_k} y_{i_k} - g_{i_k}) - \tau Q_{i_k}^{-1} S_{i_k} y_{i_k} + \tau Q_{i_k}^{-1} \tilde{u}_{i_k}, \tag{31}$$

where parameter $h = \tau\gamma$ is used to scale the convergence rate. The proposed DPB-ZNN algorithm (Equation 31) solves the DTDTVCO problem in a fully distributed manner. Notably, information exchange between agents is strictly limited to their solutions x_{i_k}. Local information, including each agent's own objective function and constraints, remains inaccessible to its neighbors.

4 Theoretical analyses

In this section, the convergence theorems for the PB-ZNN model (Equation 31) and DPB-ZNN algorithm (Equation 26) are established and proved.

4.1 Convergence Theorem of PB-ZNN Model

On the basis of the analyses presented in Subsection 3.1, the CTDTVCO problem (Equations 1–3) is reformulated as the matrix equation (Equation 15). Therefore, solving the CTDTVCO (Equations 1–3) is equivalent to solving the matrix equation (Equation 15).

Theorem 1: For the CTDTVCO problem (Equations 1–3), consider that a monotonically increasing odd activation function $\Psi(\cdot)$ is used. Starting from any initial state $\mathbf{y}(0) \in \mathbb{R}^{nd+k}$, the residual error $\|\boldsymbol{\varepsilon}(t)\|_2$ converges to zero, meaning $\lim_{t\to+\infty} \|\boldsymbol{\varepsilon}(t)\|_2 = 0$.

Proof: Define a Lyapunov candidate function as

$$V(t) = \frac{\|\boldsymbol{\varepsilon}(t)\|_2^2}{2} = \boldsymbol{\varepsilon}^{\mathrm{T}}(t)\boldsymbol{\varepsilon}(t) \geq 0. \tag{32}$$

Hence, the time derivative of $V(t)$ is

$$\dot{V}(t) = -\gamma \sum_{i=1}^{nd+k} \varepsilon_i(t)\psi(\varepsilon_i(t)). \tag{33}$$

With $\Psi(\cdot)$ being monotonically increasing and odd, the following conditions hold true for $\psi_i(\cdot)$:

$$\psi(\boldsymbol{\varepsilon}_i(t)) \begin{cases} > 0, & \text{if} \quad \varepsilon_i(t) > 0 \\ = 0, & \text{if} \quad \varepsilon_i(t) = 0 \\ < 0, & \text{if} \quad \varepsilon_i(t) < 0 \end{cases}. \tag{34}$$

Therefore, it is guaranteed that $\varepsilon_i(t)\psi(\varepsilon_i(t)) \geq 0$ always holds true, which means $\dot{V}(t) \leq 0$. According to the Lyapunov stability theorem (Khalil, 2002), one obtains that $\|\boldsymbol{\varepsilon}(t)\|_2$ converges to zero, meaning $\lim_{t\to+\infty} \|\boldsymbol{\varepsilon}(t)\|_2 = 0$. Thus, the proof is completed. ■

4.2 Convergence theorem of DPB-ZNN algorithm

In this subsection, detailed theoretical analyses about the DPB-ZNN algorithm (Equation 31) are given. The DPB-ZNN algorithm is deemed as a linear 2-step method. For better understanding, some lemmas are introduced as follows (Chen et al., 2024).

Lemma 2: A linear N-step method is formulated as $\sum_{j=0}^{N} \omega_j \alpha_{k+j} = \tau \sum_{j=0}^{N} \upsilon_j \beta_{k+j}$. The first and second characteristic polynomials of the linear multiple-step method are $z(\iota) = \sum_{j=0}^{N} \omega_j \iota^j$ and $\zeta(\iota) = \sum_{j=0}^{N} \upsilon_j \iota^j$, respectively. If all complex roots of the characteristic polynomial $z(\iota)$ ensure $|\iota| \leq 1$, and if there exists $|\iota| = 1$ with the root that ensures $|\iota| = 1$ is simple, then the corresponding N-step method is 0-stable.

Lemma 3: A linear N-step method is formulated as $\sum_{j=0}^{N} \omega_j \alpha_{k+j} = \tau \sum_{j=0}^{N} \upsilon_j \beta_{k+j}$. The order of truncation error when synthesizing the N-step method can be checked by computing $w_0 = \sum_{j=0}^{N} \omega_j$ and $w_j = \sum_{j=1}^{N} j^i/i!\omega_j - \sum_{j=0}^{N} j^{i-1}/(i-1)!\upsilon_j$. If $w_q \neq 0$ and $w_i = 0$ with $i < q$, the N-step method, then the multiple-step method has truncation error of $O(\tau^q)$.

Lemma 4: A linear N-step method is formulated as $\sum_{j=0}^{N} \omega_j \alpha_{k+j} = \tau \sum_{j=0}^{N} \upsilon_j \beta_{k+j}$. The first and second characteristic polynomials of the linear multiple-step method are $z(\iota) = \sum_{j=0}^{N} \omega_j \iota^j$ and $\zeta(\iota) = \sum_{j=0}^{N} \upsilon_j \iota^j$, respectively. If $z(1) = 0$ and $\dot{z}(1) = \zeta(1)$, with $\dot{z}$ being the derivative of $z(\iota)$, the N-step method is consistent.

Lemma 5: A linear N-step method is formulated as $\sum_{j=0}^{N} \omega_j \alpha_{k+j} = \tau \sum_{j=0}^{N} \upsilon_j \beta_{k+j}$. The N-step method is convergent, if and only if Lemmas 1 and 3 are satisfied. That is, an N-step method is convergent, if and only if it is 0-stable and consistent.

On the basis of the above lemmas, the theorem about the convergence property of DPB-ZNN algorithm (Equations 24) is proved.

Theorem 2: Consider the CTDTVO problem (Equations 1–3). Suppose that for every agent i, its local objective function $f_i(x_i(t), t)$ has continuous 2nd order derivatives. With design parameter $\gamma > 0$ and sufficiently small sampling interval gap, the DPB-ZNN algorithm (Equation 31) is convergent with a truncation error $O(\tau^2)$.

Proof: According to Lemma 5, if and only if the DPB-ZNN algorithm (Equation 24) is 0-stable and consistent, the DPB-ZNN algorithm (Equation 24) is convergent.

According to Lemma 2, the first characteristic polynomial of the DPB-ZNN algorithm (Equation 24) is formulated as

$$z_2(\iota) = \iota - 1. \tag{35}$$

The root of the characteristic polynomial is $\iota = 1$. One obtains that all roots ensure $|\iota| \leq 1$ and when$|\iota| = 1$ the root is simple. Therefore, the DPB-ZNN algorithm (Equation 24) is 0-stable.

On the basis of (Equations 30, 31), the following equation for the DPB-ZNN algorithm (Equation 31) is formulated:

$$\begin{aligned} y_{i_{k+1}} &= y_{i_k} + \tau\left(\dot{y}_{i_k} + \sigma_1 O(\tau)\right) + \tau O(\tau) \\ &= y_{i_k} + \tau \dot{y}_{i_k} + O(\tau^2). \end{aligned} \tag{36}$$

According to Lemma 3, $w_j = \sum_{j=1}^{N} j^i/i!\omega_j - \sum_{j=0}^{N} j^{i-1}/(i-1)!\upsilon_j$. Hence, w_2 is obtained as

$$w_2 = \frac{2^2}{2!} - \frac{1^2}{2!} - \frac{1^1}{1!} = \frac{1}{2} \neq 0.$$

One computes that $w_j = 0$ for $j = 0, 1$. Therefore, according to Lemma 3, the DPB-ZNN algorithm (Equation 31) has a truncation error order $O(\tau^2)$. In addition, the second characteristic polynomial of the DPB-ZNN algorithm (Equation 31) is formulated as $\zeta(\iota) = 1$. Hence, one has $\dot{z}(1) = \zeta(1) = 1$. Thus, according to Lemma 4, the DPB-ZNN algorithm (Equation 31) is consistent order $O(\tau^2)$.

According to Lemma 5, the DPB-ZNN algorithm (Equation 31) is convergent since it is 0-stable and consistent order $O(\tau^2)$. Thus, the proof is completed. ■

Theorem 3: Consider the CTDTVO problem (Equations 1–3). Suppose that, for every agent i, its local objective function $f_i(x_i(t), t)$ have continuous 2nd order derivatives. With design parameter $\gamma > 0$ and sufficiently small sampling interval gap, the maximum steady-state residual error (MSSRE) synthesized by the DPB-ZNN algorithm (Equation 26) $\lim_{k\to+\infty} \sup\|\varepsilon_{i_k}\|_2$ is of $O(\tau^2)$.

Proof: Let $y^*_{i_{k+1}}$ denotes the actual solution to the problem, i.e., $\varepsilon_{i_{k+1}} = 0$ when $y_{i_{k+1}} = y^*_{i_{k+1}}$. According to Theorem 3, the DPB-ZNN algorithm (Equation 31) has a truncation error of $O(\tau^2)$, meaning $y_{i_{k+1}} = y^*_{i_{k+1}} + O(\tau^2)$. By applying the Taylor expansion, one further has

$$\begin{aligned} \varepsilon_{i_{k+1}}(y_{i_{k+1}}, t_{k+1}) =& \varepsilon_{i_{k+1}}\left(y^*_{i_{k+1}} + O(\tau^2), t_{k+1}\right) \\ =& \varepsilon_{i_{k+1}}\left(y^*_{i_{k+1}}, t_{k+1}\right) + \frac{\partial \varepsilon_{i_{k+1}}(y_{i_{k+1}}, t_{k+1})}{\partial y_{i_{k+1}}} O(\tau^2) \\ &+ O(\tau^4) \\ \leq& \frac{\partial \varepsilon_{i_{k+1}}(y_{i_{k+1}}, t_{k+1})}{\partial y_{i_{k+1}}} O(\tau^2). \end{aligned}$$

Hence, the MSSRE is deducted as

$$\begin{aligned} \lim_{k\to+\infty} \sup\left\|\varepsilon_{i_{k+1}}(y_{i_{k+1}}, t_{k+1})\right\|_2 &\leq \lim_{k\to+\infty} \sup\left\|\frac{\partial \varepsilon_{i_{k+1}}(y^*_{i_{k+1}}, t_{k+1})}{\partial y^*_{i_{k+1}}} O(\tau^2)\right\|_2 \\ &\leq \lim_{k\to+\infty} \sup\left\|\frac{\partial \varepsilon_{i_{k+1}}(y^*_{i_{k+1}}, t_{k+1})}{\partial y^*_{i_{k+1}}}\right\|_F O(\tau^2), \end{aligned}$$

where$\|\cdot\|_F$ represents the Frobenius norm of a matrix. Since $y^*_{i_{k+1}}$ is the actual solution, the partial derivative $\|\partial\varepsilon^i_{k+1}(y^*_{i_{k+1}}, t_{k+1})/\partial y^*_{i_{k+1}}\|_F$ evaluates to a constant matrix, bounded due to the continuity of the second-order derivatives of f_i. Therefore, the MSSRE synthesized by the DPB-ZNN algorithm (Equation 31) is of $O(\tau^2)$. Thus, the proof is completed. ■

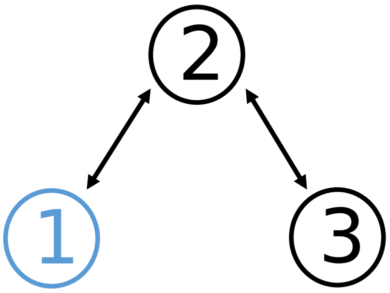

FIGURE 1
Communication topology of graph $\mathcal{G}_1$ in Example 1, where $\mathcal{G}_1$ is connected and undirected.

TABLE 1 Expressions of time-varying objective functions of all agents in Example 1.

Objective function	Expression
$f_1(x_k, t_k)$	$\frac{\sin(t_k)}{4}x^2_{1_k} + x^2_{2_k} + \cos(t_k)x_{1_k}x_{2_k}$
$f_2(x_k, t_k)$	$x^2_{1_k} + \frac{\sin(t_k)}{4}x^2_{2_k} + \cos(3t_k)x_{1_k}$
$f_3(x_k, t_k)$	$x^2_{1_k} + x^2_{2_k} + \sin(3t_k)x_{2_k}$

5 Illustrative examples

In this section, two numerical examples are presented. These examples serve to validate the effectiveness of the proposed DPB-ZNN algorithm (Equation 31) discussed in this study.

5.1 Example 1: DTDTVCO problem by DPB-ZNN algorithm

Consider a DTDTVCO problem with a MAS consisting of three agents. The formulation of the DTDTVCO problem is given as follows:

$$\begin{aligned} &\text{minimize} \quad f(\mathbf{x}_{k+1}, t_{k+1}) = \sum_{i=1}^{3} f_i(x_{i_{k+1}}, t_{k+1}), \\ &\text{subject to} \quad K_{i_{k+1}}\mathbf{x}_{k+1} = b_{i_{k+1}}, \quad i = 1, \\ &\qquad\qquad\quad J_{i_{k+1}}\mathbf{x}_{k+1} \leq c_{i_{k+1}}, \quad i = 2, 3, \end{aligned} \tag{37}$$

where $x_{i_k} \in \mathbb{R}^2$. The communication topology of the network $\mathcal{G}_1$ is shown in Figure 1. The detailed time-varying local objective functions f_i and the corresponding time-varying local constraints are given through Tables 1, 2.

Before conducting the experiment, the parameters of the DPB-ZNN algorithm (Equation 31) must be properly configured. The total solving time is set to $T = 10$ s, the sampling gap τ is set to $\tau = 0.001$ s, and h is set to 0.2. In this experiment, for the first penalty function, σ_1 is set to 50. To guarantee that x_{i_k} remains within the feasible region of the inequality constraints for any agent, σ_2 is set to 100, while the parameter ρ is set to a value close to zero, specifically 0.001.

The initial states x_{i_0} are set as $x_{1_0} = [-0.6, 0.6]^{\mathrm{T}}$, $x_{2_0} = [0, 0]^{\mathrm{T}}$, and $x_{3_0} = [0.6, -0.6]^{\mathrm{T}}$. The experimental results are illustrated through Figures 2, 3. In Figures 2, 3, the dashed red lines denote the inequality constraints $J_{2_k}x_{2_k} \leq c_{2_k}$ associated with the agent 2, and dashed purple lines denote the inequality constraints $J_{3_k}x_{3_k} \leq c_{3_k}$ associated with the agent 3.

From Figures 2, 3, one sees that the three agents achieve consensus from different initial states. In Figures 2, 3, the distributed time-varying inequality constraints are denoted by the red and purple dashed lines. As pointed out by the blue circles in Figures 2, 3, for each agent, their solutions are constrained by the distributed time-varying inequality constraints.

In addition, several experiments are conducted with DPB-ZNN algorithm (Equation 31) solving the DTDTVCO problem (Equation 37) with different τ. The corresponding MSSREs $\|\boldsymbol{\varepsilon}_k\|_2$ are presented in Figure 4, Table 3. In Figure 4, the MSSREs $\|\boldsymbol{\varepsilon}_k\|_2$ with different τ are presented. Figure 4 includes three distinct trajectories corresponding to $\tau = 0.001$ s (blue dashed line), $\tau = 0.0001$ s (orange dashed line), and $\tau = 0.00001$ s (green dashed line). The blue dashed line, for $\tau = 0.001$ s, shows a decreasing MSSRE starting from approximately 10^2 to approximately 10^{-2}. The orange dashed line, for $\tau = 0.0001$ s, reflects a faster convergence with MSSRE reducing to 10^{-4}. The green dashed line, for $\tau = 0.00001$ s, presents the most rapid reduction, with MSSRE decreasing from 10^2 to 10^{-6}, highlighting the advantage of very small time steps for precise updates. The figure reveals a clear trend: As τ decreases, the MSSRE also decreases, indicating improved performance of the algorithm with smaller τ values. From Figure 4, Table 3, one sees that the MSSRE $\|\boldsymbol{\varepsilon}_k\|_2$ is of $O(\tau^2)$, which corroborates the theoretical analyses.

TABLE 2 Expressions of time-varying constraints of all agents in Example 1.

Constraint	Expressions
$K_{1_k}x_{1_k} = b_{1_k}$	$\sin(t_k)x_{1_k} + \cos(t_k)x_{2_k} = \cos(2t_k)$
$J_{2_k}x_{2_k} \leq c_{2_k}$	$x_{1_k} \leq 0.5\cos(t_k - 4.4) + 1.5$ $x_{2_k} \leq 0.5\cos(t_k - 4) + 0.9$
$J_{3_k}x_{3_k} \leq c_{3_k}$	$x_{1_k} \geq -0.5\cos(t_k - 3.6) - 1$ $x_{2_k} \geq -0.5\cos(t_k - 3.6) - 1$

Moreover, the gradient neural network (GNN) algorithm is often used to solve time-varying problems with constraints. Therefore, a comparison experiment between the GNN and the proposed DPB-ZNN algorithm is conducted.

First, for the agent i, a scalar-valued energy function is designed as follows:

$$e^G_{i_k} = \frac{1}{2}\|\varepsilon_{i_k}\|_2^2, \tag{38}$$

where $\varepsilon_{i_k} = A_{i_k}y_{i_k} - g_{i_k}$. Then, by exploiting the gradient information of the energy function (Equation 38), one obtains

$$\dot{y}_{i_k} = -\gamma^G \frac{\partial e^G_{i_k}}{\partial y_{i_k}} = -\gamma^G \left(\frac{\partial \varepsilon_{i_k}}{\partial y_{i_k}}\right)^{\mathrm{T}} \varepsilon_{i_k} = -\gamma^G A^{\mathrm{T}}_{i_k}(A_{i_k}y_{i_k} - g_{i_k}),$$

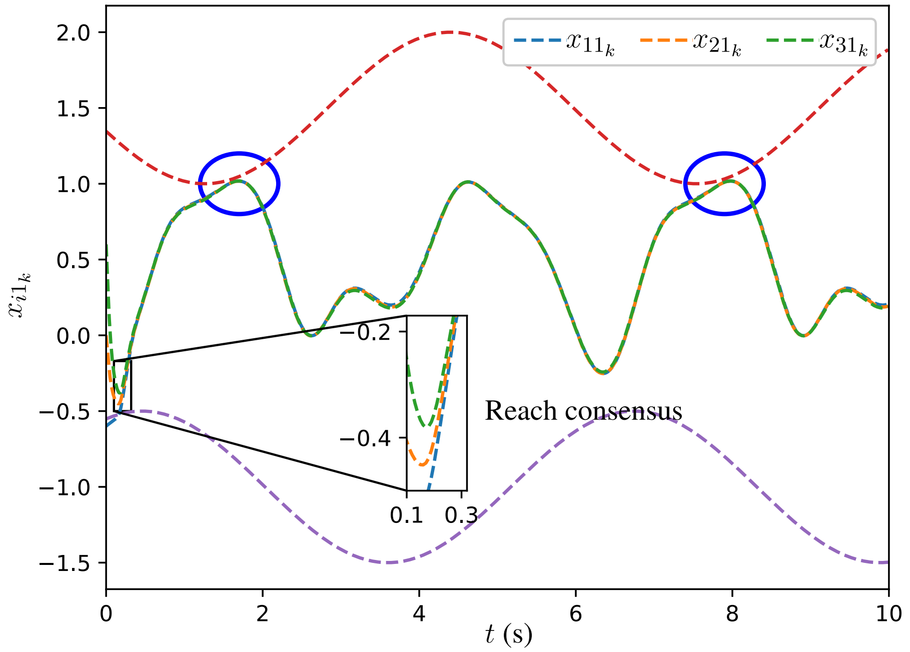

FIGURE 2
Trajectories of x_{i1_k} for all agents synthesized by DPB-ZNN algorithm (Equation 31).

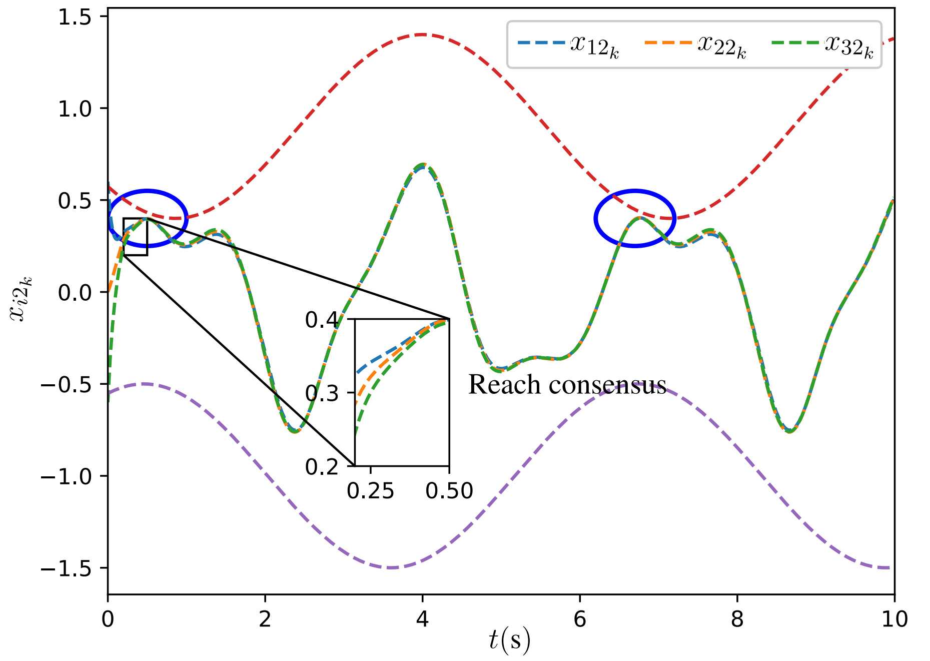

FIGURE 3
Trajectories of x_{i2_k} for all agents synthesized by DPB-ZNN algorithm (Equation 31).

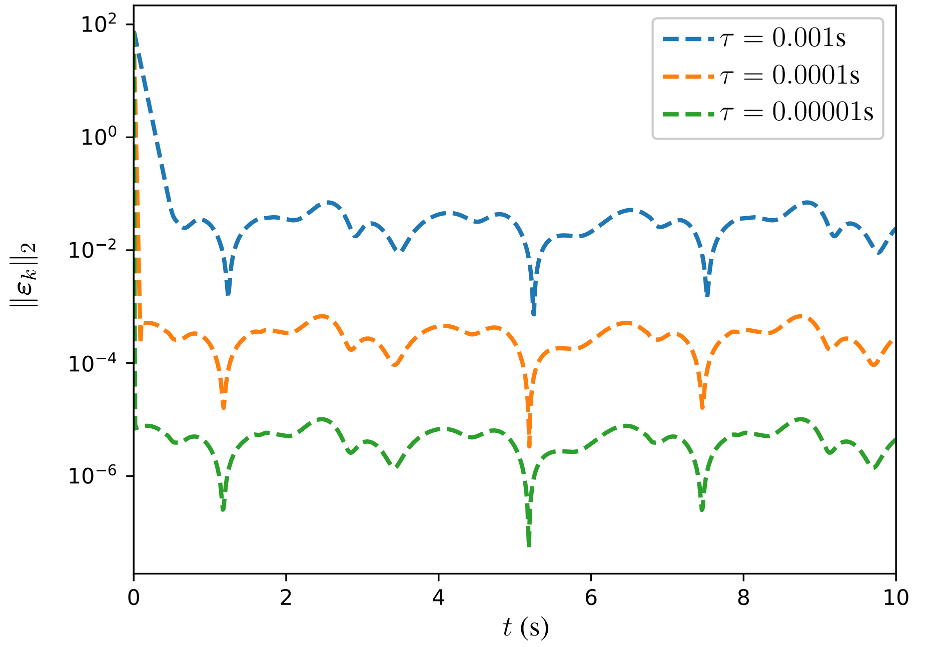

FIGURE 4
Trajectories of MSSRE $\|\varepsilon_k\|_2$ for DPB-ZNN algorithm in solving DTDTVCO problem (Equation 40), with different τ.

where the parameter γ^G is used to scale the convergence rate of the GNN algorithm (Zhang et al., 2021b). Hence, the GNN algorithm for solving the DTDTVCO problem is given as

$$y_{i_{k+1}} = y_{i_k} - \gamma^G \tau A_{i_k}^{\mathrm{T}}(A_{i_k} y_{i_k} - g_{i_k}), \tag{39}$$

Then, both the GNN algorithm (Equation 39) and the DPB-ZNN algorithm (Equation 31) are used to solve the DTDTVCO problem (Equation 37). For both the GNN algorithm and the DPB-ZNN algorithm, the parameters σ_1, σ_2, ρ, and the initial states are set the same. The corresponding experimental results are shown in Table 4, Figure 5.

The MSSRE $\|\boldsymbol{\varepsilon}_k\|_2$ along with the corresponding parameters for both GNN and DPB-ZNN algorithms are provided in Table 4. From Table 4 and Figure 5, one sees that, with the sampling time being the same $\tau = 0.0001$ s, DPB-ZNN algorithm (Equation 31) achieves a higher accuracy with the MSSRE $\|\boldsymbol{\varepsilon}_k\|_2$ being 3.54×10^{-4}, while the MSSRE $\|\boldsymbol{\varepsilon}_k\|_2$ of GNN algorithm is 2.16×10^{-1}. To sum up, the DPB-ZNN algorithm (Equation 31) has a higher accuracy when solving DTDTVCO problem than the GNN algorithm.

TABLE 3 MSSREs of DPB-ZNN algorithm (Equation 31) in solving DTDTVCO problem (Equation 37) with different τ.

τ	MSSRE $\|\boldsymbol{\varepsilon}_k\|_2$
0.001 s	3.76×10^{-2}
0.0001 s	5.51×10^{-4}
0.00001 s	6.44×10^{-6}

TABLE 4 Comparisons between GNN algorithm and DPB-ZNN algorithm in solving DTDTVCO problem.

Algorithm	Parameter	τ	MSSRE $\varepsilon_k\|_2$
GNN	$\gamma^G = 2,000$	0.0001s	2.16×10^{-1}
DPB-ZNN	$\gamma = 2,000$	0.0001s	3.54×10^{-4}

In addition, to investigate the impact of model inaccuracies and measurement noise, numerical experiments on a DTVCO problem with noise, solved by the DPB-ZNN algorithm, are conducted. To lay the basis for further investigation on the robustness of DPB-ZNN algorithm under the pollution of unknown noises, one obtains the following equation:

$$y_{i_{k+1}} \doteq y_{i_k} - hQ_{i_k}^{-1}(A_{i_k} y_{i_k} - g_{i_k}) - \tau Q_{i_k}^{-1} S_{i_k} y_{i_k} + \tau Q_{i_k}^{-1} \tilde{u}_{i_k} + \varrho_{i_k}, \tag{40}$$

where $\varrho_{i_k} \in [-0.5, 0.5]$ denotes a discrete-time bounded unknown random noise that is uniformly distributed within the range $[-0.5, 0.5]$. The corresponding experimental results are presented in Figure 6. Except for the bounded random noise ϱ_{i_k}, the initial states and corresponding parameters are set to be the same as those in Subsection 5.1. The trajectories of the residual error $\|\boldsymbol{\varepsilon}_k\|_2$ of the DPB-ZNN algorithm with different values of τ are illustrated in Figure 6.

From the Figure 6, one observes that the MSSRE converges over time for all three sampling intervals, indicating that the DPB-ZNN algorithm effectively reduces the residual errors in the presence of noise. The trajectory with $\tau = 0.00001$ s (green dash line) shows the fastest convergence, reaching the lowest error value within the shortest time, with MSSRE values dropping to approximately 10^{-4}. The trajectory with $\tau = 0.0001$ s (orange dashed line) also converges well but at a slightly slower rate, with MSSRE values reaching approximately 10^{-3}. The trajectory with $\tau = 0.001$ s (blue dashed line) converges the slowest, taking the longest time to reach a low error value, with MSSRE values approximately 10^{-2}.

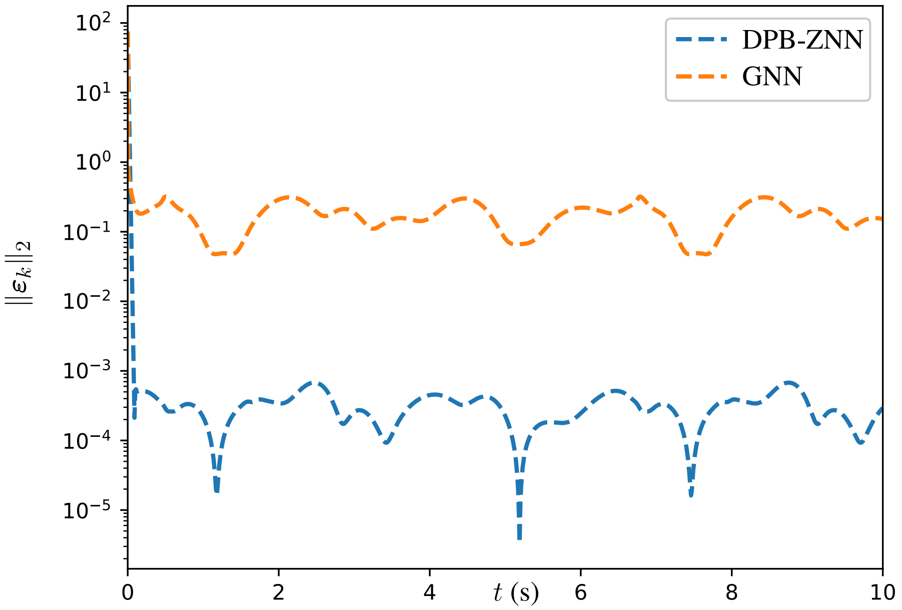

FIGURE 5
Trajectories of $\|\boldsymbol{\varepsilon}_k\|_2$ of GNN and DPB-ZNN algorithms in solving DTDTVCO problem.

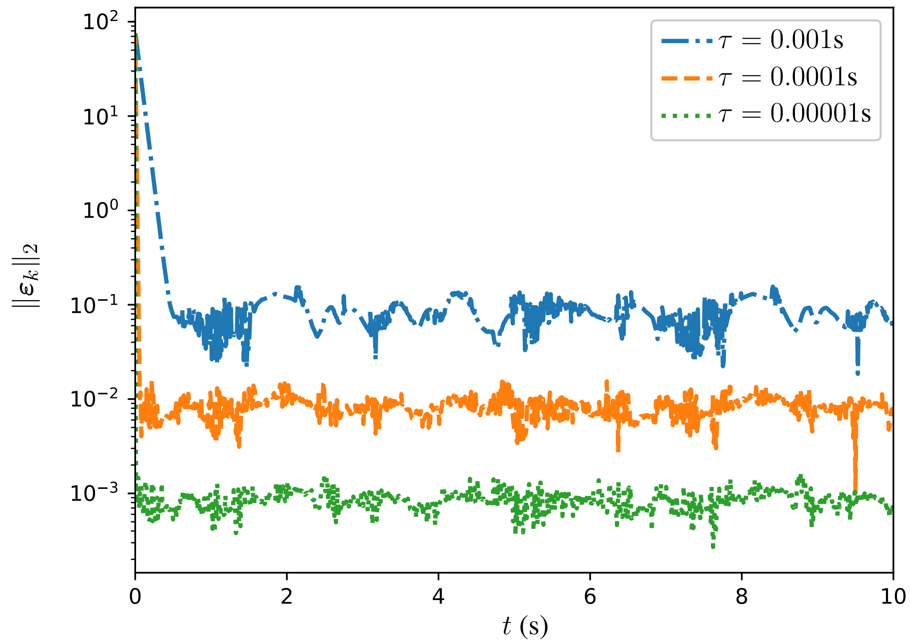

FIGURE 6
Trajectories of MSSRE $\|\varepsilon_k\|_2$ for DPB-ZNN algorithm in solving DTDTVCO problem with random noise and different τ.

These experimental results illustrate the robustness of the DPB-ZNN algorithm as it consistently minimizes residual errors and converges effectively even in the presence of noise. The smaller sampling intervals τ lead to faster convergence and lower residual errors, highlighting the algorithm's ability to handle noise polluted time-varying challenges with high accuracy and efficiency.

5.2 Example 2: DPB-ZNN algorithm application to cooperative control of redundant robot manipulators

To illustrate the efficacy of the proposed DPB-ZNN algorithm (Equation 31) in applications, a cooperative control experiment is simulated. In this experiment, we consider a group of four identical 5-joint planar redundant robot manipulators performing a cooperative control task. The communication topology of the four robot manipulators is shown in Figure 7.

To control the redundant robot manipulators performing a cooperative control, for the entire system, a motion planning scheme for the entire system on the basis of the DTDTVCO problem is described as follows (Jin et al., 2018):

$$\text{minimize} \quad \sum_{i=1}^{4} \frac{1}{2}\dot{\theta}_{i_k}^{\mathrm{T}}\dot{\theta}_{i_k}, \tag{41}$$

$$\text{subject to} \quad \mathcal{J}_{iw_k}\dot{\theta}_{i_k} = \dot{r}_{w_k}, \quad i = 2, 4, \tag{42}$$

$$\sum_{j\in\mathcal{N}_{i_k}} \|\dot{\theta}_{i_k} - \dot{\theta}_{j_k}\|_2 = 0, i = 1, 2, 3, 4, \tag{43}$$

$$\dot{\theta}_i^- \leq \dot{\theta}_{i_k} \leq \dot{\theta}_i^+, \quad i = 1, 2, 3, 4. \tag{44}$$

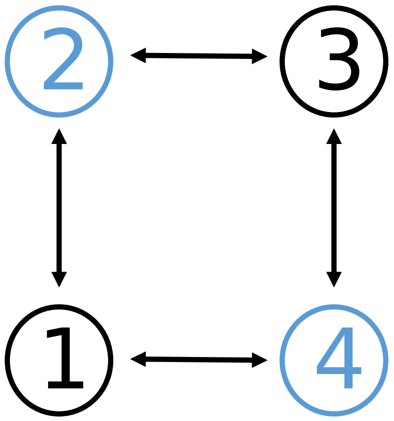

FIGURE 7
Communication topology of graph $\mathcal{G}_2$ in Example 3, where $\mathcal{G}_2$ is connected and undirected.

where $\dot{r}_{iw_k}$ denotes the time derivative of the desired trajectory of the end-effector position vector (i.e., reference), $\dot{\theta}_{i_k} \in \mathbb{R}^n$ denotes the joint velocities of the ith manipulator, and $\mathcal{J}_{iw_k} \in \mathbb{R}^{2\times5}$ denotes the Jacobian matrix of the manipulator i. In applications, limitations of joint velocities are often encountered. The boundary constraints are denoted as $\dot{\theta}_i^- \leq \dot{\theta}_i \leq \dot{\theta}_i^+$, where the lower and upper limits of the joint velocity $\dot{\theta}_i$ are denoted as $\dot{\theta}_i^- \in \mathbb{R}^5$ and $\dot{\theta}_i^+ \in \mathbb{R}^5$, respectively. The bound constraints (Equation 44) are easily transformed into inequality constraints $J_{i1}\dot{\theta}_{i_k} \leq \dot{\theta}_i^+$ and $J_{i2}\dot{\theta}_{i_k} \leq -\dot{\theta}_i^-$, in which $J_{i1} = I_{d\times d}$ and $J_{i2} = -I_{d\times d}$, respectively. It is worth pointing out that, as discussed in Section 3.1, the equality

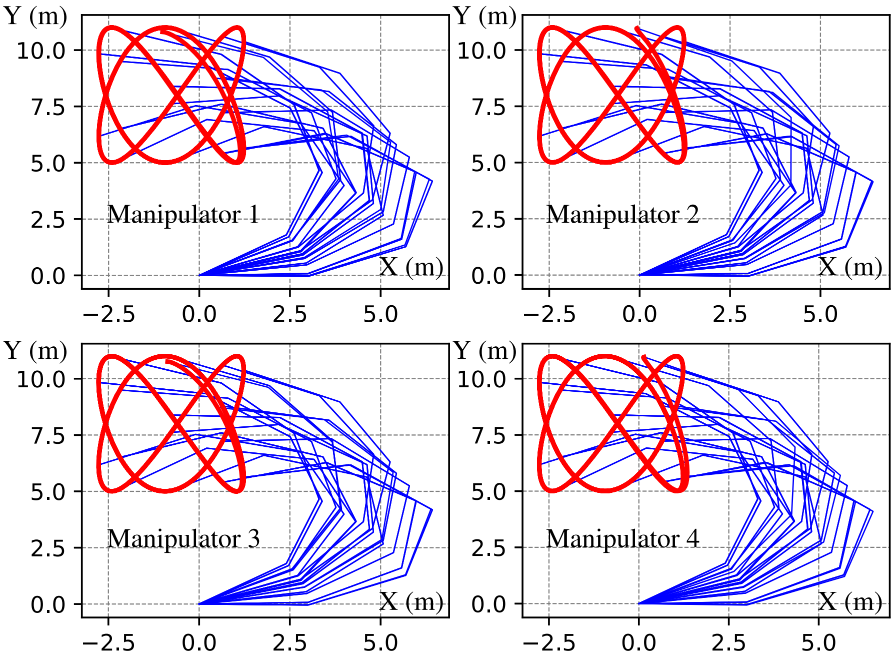

FIGURE 8
Trajectories of end effectors generated by DPB-ZNN algorithm (Equation 31) when robot manipulators track Lissajous pattern.

constraints of the robot manipulators should not be contradictory. In this particular experiment, manipulators 2 and 4 share the same equality constraint $\mathcal{J}_{iw_k}\dot{\theta}_{i_k} = \dot{r}_{w_k}$, whereas manipulators 1 and 3 do not have such constraints. This means that manipulators 2 and 4 have access to the desired path, while manipulators 1 and 3 do not. Then, the DPB-ZNN algorithm (Equation 31) is adopted to solve the aforementioned cooperative control problem with parameters $h = 0.2$, $\sigma_1 = 50$, and $\sigma_2 = 50$. The sampling time is set to $\tau = 0.001$ s. The desired trajectory for the robot manipulators follows a Lissajous pattern. In this simulation experiment, the upper and lower joint velocity limits for each joint were set to 1.5 and -1.5 rad/s, respectively. The experimental results are presented in Figure 8 through Figure 10.

The experimental results depicted in Figure 8 showcase the trajectories of the end effectors generated by the DPB-ZNN algorithm, as the manipulators track a Lissajous pattern. Each subplot, corresponding to Manipulator 1 through Manipulator 4, exhibits that all manipulators track the Lissajous pattern successfully. The experimental results depicted in Figure 9 reveal the effectiveness of the proposed DPB-ZNN algorithm in driving the residual errors $\|\epsilon_{i_k}\|_2 = \|\mathcal{J}_{iw_k}\dot{\theta}_{i_k} - \dot{r}_{w_k}\|_2$ of all robot manipulators toward zero over time. Figure 9 fully illustrates the efficiency of the proposed DPB-ZNN algorithm for solving the cooperative control problem of robot manipulators with joint velocity limits.

From Figure 9, one sees that the residual error trajectories, presented in a logarithmic scale reaches the level of 10^{-4} m/s to 10^{-5} m/s over a 10-s period, illustrate a rapid reduction in error magnitude, particularly during the initial phase of the simulation, highlighting the algorithm's swift convergence capabilities. While minor fluctuations in the early stages reflect the dynamic adjustments made by the algorithm to adhere to time-varying constraints and achieve consensus among agents, the overall trend consistently shows convergence across all manipulators. The slight variations in the error reduction rate among the manipulators are likely due to differences in their initial states and local objective functions, yet these discrepancies diminish over time, emphasizing the algorithm's distributed nature and ability to enforce consensus and feasibility. In addition, one sees that despite the lack of information of the desired path, manipulator 1 and manipulator 3 track the desired path successfully, meaning the cooperative control problem is solved by the DPB-ZNN algorithm (Equation 31) effectively in a fully distributed manner.

Furthermore, the joint velocities of the robot manipulators solved by the DPB-ZNN algorithm (Equation 31) are presented in Figure 10. As shown in Figure 10, for all agents, all the joint velocities remain within the upper and lower joint velocity limits, which verify the effectiveness of DPB-ZNN algorithm in applying to the cooperative control problem of robot manipulators. To sum up, this experiment verifies the effectiveness and high accuracy of the proposed DPB-ZNN algorithm (Equation 31).

6 Conclusion

Aiming at solving the CTDTVCO problem with both time-varying equality and inequality constraints, a novel PB-ZNN model has been designed in this study by incorporating two penalty functions. The proposed PB-ZNN model solves the CTDTVCO problem in a semi-centralized manner. Then, on the basis of

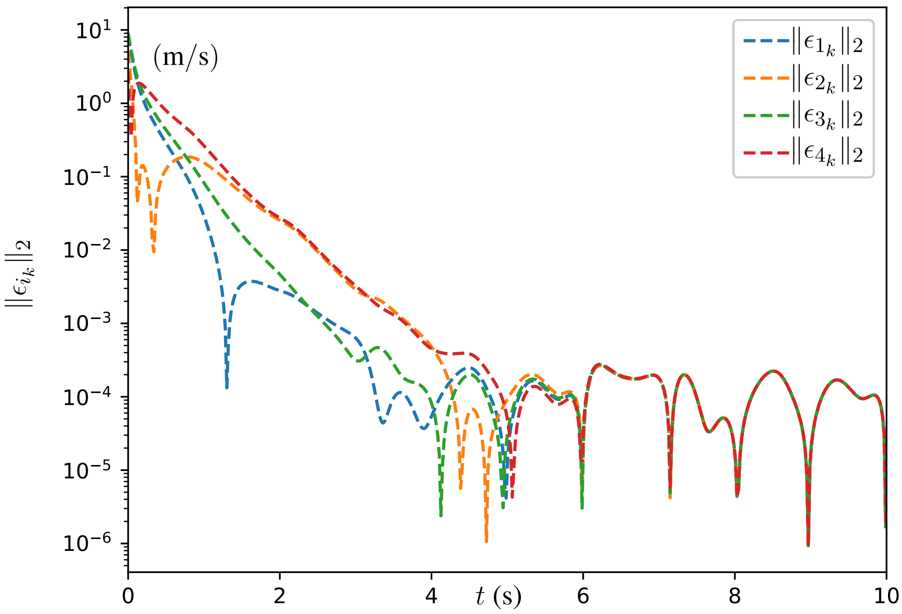

FIGURE 9
Trajectories of residual error $\|\epsilon_{i_k}\|_2$ of all robot manipulators.

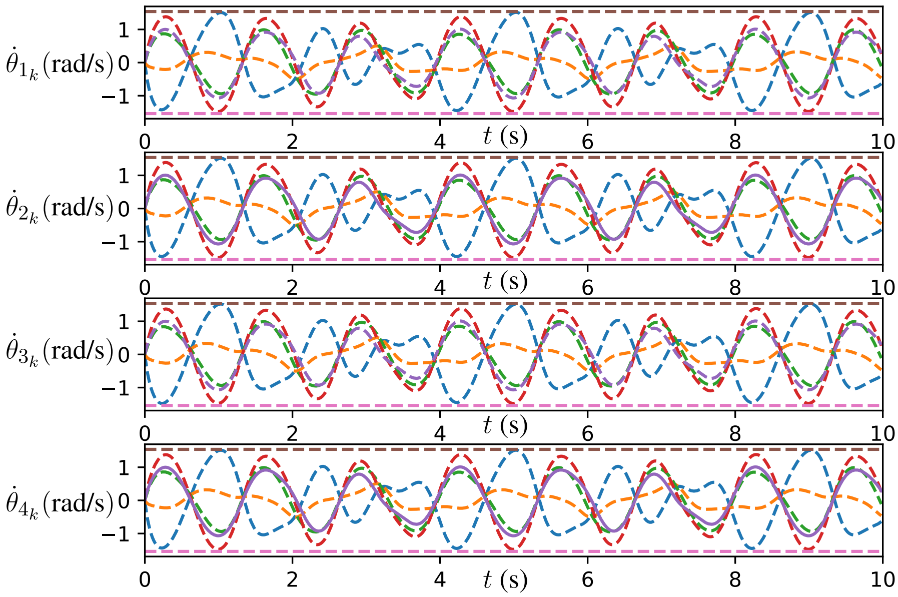

FIGURE 10
Trajectories of $\dot{\theta}_{i_k}$ for four robot manipulators generated by DPB-ZNN algorithm (Equation 31) when robot manipulators track Lissajous pattern.

the PB-ZNN model, a DPB-ZNN algorithm has been proposed. By adopting an approximation formula, the DPB-ZNN algorithm solves the DTDTVCO problem in a fully distributed manner. The global convergence theorems of the proposed PB-ZNN model and DPB-ZNN algorithm have been proved. Numerical experiment results have illustrated the efficacy and efficiency of the proposed DPB-ZNN algorithm, including a simulation experiment applying it to the cooperative control of redundant robot manipulators.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

LH: Conceptualization, Data curation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. HC: Conceptualization, Supervision, Validation, Writing – review & editing. YZ: Validation, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research and/or publication of this article. This research was supported by the China National Key R&D Program (with number 2022YFB3903804), the National Natural Science Foundation of China (with number 62376290), and Natural Science Foundation of Guangdong Province (with number 2024A1515011016).

Acknowledgments

The authors wish to express their sincere thanks to Sun Yat-sen University for its support and assistance in this study.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Bahman, G., and Jorge, C. (2014). Distributed continuous-time convex optimization on weight-balanced digraphs. *IEEE Trans. Automat. Contr.* 59, 781–786. doi: 10.1109/TAC.2013.2278132

Cai, Y., Yu, W., Nie, X., Cheng, Q., and Cui, T. (2024). Joint resource allocation for ris-assisted heterogeneous networks with centralized and distributed frameworks. *IEEE Trans. Circuits Syst. I Regul. Pap.* 71, 2132–2145. doi: 10.1109/TCSI.2023.3325262

Chen, J., Pan, Y., and Zhang, Y. (2024). Znn continuous model and discrete algorithm for temporally variant optimization with nonlinear equation constraints via novel td formula. *IEEE Trans. Syst. Man Cybern. Syst.* 54, 3994–4004. doi: 10.1109/TSMC.2024.3374754

Cherukuri, A., and Cortés, J. (2016). Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment. *Automatica* 74, 183–193. doi: 10.1016/j.automatica.2016.07.003

Dai, J., Tan, P., Xiao, L., Jia, L., He, Y., and Luo, J. (2023). A fuzzy adaptive zeroing neural network model with event-triggered control for time-varying matrix inversion. *IEEE Trans. Fuzzy Syst.* 31, 3974–3983. doi: 10.1109/TFUZZ.2023.3272653

Diestel, R. (2017). *Graph Theory*. Berlin: Springer.

Ding, L., Xiao, L., Liao, B., Lu, R., and Peng, H. (2018). An improved recurrent neural network for complex-valued systems of linear equation and its application to robotic motion tracking. *Front. Neurorobot* 11:45. doi: 10.3389/fnbot.2017.00045

Golub, G. H., and Loan, C. F. V. (2013). *Matrix Computations*. Baltimore: John Hopkins Univesity Press.

He, S., He, X., and Huang, T. (2021). A continuous-time consensus algorithm using neurodynamic system for distributed time-varying optimization with inequality constraints. *J. Franklin I.* 358, 6741–6758. doi: 10.1016/j.jfranklin.2021.07.007

Hu, Y., Zhang, C., Wang, B., Zhao, J., Gong, X., Gao, J., et al. (2024). Noise-tolerant znn-based data-driven iterative learning control for discrete nonaffine nonlinear mimo repetitive systems. *IEEE-CAA J. Automatic* 11, 344–361. doi: 10.1109/JAS.2023.123603

Huang, B., Zou, Y., Chen, F., and Meng, Z. (2022). Distributed time-varying economic dispatch via a prediction-correction method. *IEEE Trans. Circuits Syst. I Regul. Pap.* 69, 4215–4224. doi: 10.1109/TCSI.2022.3185398

Jia, W., Huang, T., and Qin, S. (2024). A collective neurodynamic penalty approach to nonconvex distributed constrained optimization. *Neural Netw.* 171, 145–158. doi: 10.1016/j.neunet.2023.12.011

Jin, J., Zhao, L., Chen, L., and Chen, W. (2022a). A robust zeroing neural network and its applications to dynamic complex matrix equation solving and robotic manipulator trajectory tracking. *Front. Neurorobot* 16:1065256. doi: 10.3389/fnbot.2022.1065256

Jin, J., Zhu, J., Gong, J., and Chen, W. (2022b). Novel activation functions-based znn models for fixed-time solving dynamirc sylvester equation. *Neural Comput. Applic.* 34, 14297–14315. doi: 10.1007/s00521-022-06905-2

Jin, L., Huang, R., Liu, M., and Ma, X. (2024). Cerebellum-inspired learning and control scheme for redundant manipulators at joint velocity level. *IEEE Trans. Cybern.* 54, 6297–6306. doi: 10.1109/TCYB.2024.3436021

Jin, L., Li, S., Xiao, L., Lu, R., and Liao, B. (2018). Cooperative motion generation in a distributed network of redundant robot manipulators with noises. *IEEE Trans. Syst. Man Cybern. Syst.* 48, 1715–1724. doi: 10.1109/TSMC.2017.2693400

Jin, L., Liao, B., Liu, M., Xiao, L., Guo, D., and Yan, X. (2017). Different-level simultaneous minimization scheme for fault tolerance of redundant manipulator aided with discrete-time recurrent neural network. *Front. Neurorobot.* 11:50. doi: 10.3389/fnbot.2017.00050

Kang, J., and Yang, G. G. G. (2023). Distributed optimization of uncertain multiagent systems with disturbances and actuator faults via exosystem observer-based output regulation. *IEEE Trans. Circuits Syst. I Regul. Pap.* 70, 897–909. doi: 10.1109/TCSI.2022.3221097

Khalil, H. K. (2002). *Nonlinear Systems*. Saddle River, New Jersey: Prentice Hall.

Khan, A. H., Cao, X., Li, S., Katsikis, V. N., and Liao, L. (2020a). BAS-ADAM: an adam based approach to improve the performance of beetle antennae search optimizer. *IEEE/CAA J. Autom. Sinica* 7, 461–471. doi: 10.1109/JAS.2020.1003048

Khan, A. H., Li, S., and Luo, X. (2020b). Obstacle avoidance and tracking control of redundant robotic manipulator: An rnn-based metaheuristic approach. *IEEE Trans. Ind. Informat.* 16, 123–134. doi: 10.1109/TII.2019.2941916

Lan, X., Jin, J., and Liu, H. (2023). Towards non-linearly activated znn model for constrained manipulator trajectory tracking. *Front. Phys*. 11:1159212. doi: 10.3389/fphy.2023.1159212

Li, J., Lu, J., Liu, Y., and Cao, J. (2024). Optimal energy management for networked microgrids via distributed neurodynamic. *IEEE Trans. Circuits Syst. I Regul. Pap*. 71, 2182–2192. doi: 10.1109/TCSI.2024.3351942

Liao, B., Wang, Y., Li, J., Guo, D., and He, Y. (2022). Harmonic noise-tolerant znn for dynamic matrix pseudoinversion and its application to robot manipulator. *Front. Neurorobot*. 16:928636. doi: 10.3389/fnbot.2022.928636

Liu, M., Li, Y., Chen, Y., Qi, Y., and Jin, L. (2024). A distributed competitive and collaborative coordination for multirobot systems. *IEEE Trans. Mobile Comput*. 23, 11436–11448. doi: 10.1109/TMC.2024.3397242

Mao, S., Yang, M., Yang, W., Tang, Y., Zheng, W., Gu, J., et al. (2023). Differentially private distributed optimization with an event-triggered mechanism. *IEEE Trans. Circuits Syst. I Regul. Pap*. 70, 2943–2956. doi: 10.1109/TCSI.2023.3266358

Molzahn, D. K., Dörfler, F., Sandberg, H., Low, S. H., Chakrabarti, S., Baldick, R., et al. (2017). A survey of distributed optimization and control algorithms for electric power systems. *IEEE Trans. Smart Grid* 8:2941–2962. doi: 10.1109/TSG.2017.2720471

Peng, J., Fan, B., and Liu, W. (2022). Penalty-based distributed optimal control of dc microgrids with enhanced current regulation performance. *IEEE Trans. Circuits Syst. I Regul. Pap*. 69, 3026–3036. doi: 10.1109/TCSI.2022.3167790

Qiu, B., Guo, J., Li, X., and Zhang, Y. (2021). New discretized zeroing neural network models for solving future system of bounded inequalities and nonlinear equations aided with general explicit linear four-step rule. *IEEE Trans. Industr. Inform*. 17, 5164–5174. doi: 10.1109/TII.2020.3032158

Song, F., Zhou, Y., Xu, C., and Sun, Z. (2024). A novel discrete zeroing neural network for online solving time-varying nonlinear optimization problems. *Front. Neurorobot*. 18:1446508. doi: 10.3389/fnbot.2024.1446508

Sun, S., Xu, J., and Ren, W. (2023). Distributed continuous-time algorithms for time-varying constrained convex optimization. *IEEE Trans. Automat. Contr*. 68, 3931–3946. doi: 10.1109/TAC.2022.3198113

Sun, S., Zhang, Y., Lin, P., Ren, W., and Farrell, J. A. (2022). Distributed timevarying optimization with state-dependent gains: algorithms and experiments. *IEEE Trans. Control Syst. Technol*. 30, 416–425. doi: 10.1109/TCST.2021.3058845

Sun, Z., Shi, T., Wei, L., Sun, Y., Liu, K., and Jin, L. (2020). Noise-suppressing zeroing neural network for online solving time-varying nonlinear optimization problem: a control-based approach. *Neural Comput. Appl*. 32, 11505–11520. doi: 10.1007/s00521-019-04639-2

Sun, Z., Tang, S., Fei, Y., Xiao, X., Hu, Y., and Yu, J. (2024a). An orthogonal repetitive motion and obstacle avoidance scheme for omnidirectional mobile robotic arm. *IEEE Trans. Ind. Electron*. 2024, 1–12. doi: 10.1109/TIE.2024.3451063

Sun, Z., Xu, C., Jin, L., Pang, Z., and Yu, J. (2024b). Human–robot interaction force control of series elastic actuator-driven upper limb exoskeleton robot. *IEEE Trans. Ind. Electron*. 71, 11449–11461. doi: 10.1109/TIE.2024.3468711

Tan, N., Yu, P., Zhong, Z., and Zhang, Y. (2023). Data-driven control for continuum robots based on discrete zeroing neural networks. *IEEE Trans. Industr. Inform*. 19, 7088–7098. doi: 10.1109/TII.2022.3204307

Wang, G., Jin, L., Zhang, J., Duan, X., Yi, J., Zhang, M., et al. (2024). Recurrent neural network enabled continuous motion estimation of lower limb joints from incomplete semg signals. *IEEE Trans. Neural Syst. Rehabil. Eng*. 32, 1–10. doi: 10.1109/TNSRE.2024.3459924

Xie, Z., Liu, M., Su, Z., Sun, Z., and Jin, L. (2025). A data-driven obstacle avoidance scheme for redundant robots with unknown structures. *IEEE Trans. Ind. Informat*. 21, 1793–1802. doi: 10.1109/TII.2024.3488775

Yan, J., Jin, L., and Hu, B. (2024). Data-driven model predictive control for redundant manipulators with unknown model. *IEEE Trans. Cybern*. 54, 5901–5911. doi: 10.1109/TCYB.2024.3408254

Yang, M., Zhang, Y., Hu, H., and Qiu, B. (2020). General 7-instant dcznn model solving future different-level system of nonlinear inequality and linear equation. *IEEE Trans. Neural Netw. Learn. Syst*. 31, 3204–3214. doi: 10.1109/TNNLS.2019.2938866

Ye, M., and Hu, G. (2015). Distributed seeking of time-varying nash equilibrium for non-cooperative games. *IEEE Trans. Automat. Contr*. 60, 3000–3005. doi: 10.1109/TAC.2015.2414817

Zhang, W., Zhao, Y., He, W., and Wen, G. (2021a). Time-varying formation tracking for multiple dynamic targets: Finite- and fixed-time convergence. *IEEE Trans. Circuits Syst. II Express Briefs* 68, 1323–1327. doi: 10.1109/TCSII.2020.3025199

Zhang, Y., Jiang, D., and Wang, J. (2002). A recurrent neural network for solving sylvester equation with time-varying coefficients. *IEEE Trans. Neural Netw*. 13, 1053–1063. doi: 10.1109/TNN.2002.1031938

Zhang, Y., Li, S., Kadry, S., and Liao, B. (2018a). Recurrent neural network for kinematic control of redundant manipulators with periodic input disturbance and physical constraints. *IEEE Trans. Cybern*. 49, 4194–4205. doi: 10.1109/TCYB.2018.2859751

Zhang, Z., Lin, Y., Li, S., Li, Y., Yu, Z., and Luo, Y. (2018b). Tricriteria optimization-coordination motion of dual-redundant-robot manipulators for complex path planning. *IEEE Trans. Control Syst. Technol*. 26, 1345–1357. doi: 10.1109/TCST.2017.2709276

Zhang, Z., Yang, S., and Zheng, L. (2021b). A penalty strategy combined varying-parameter recurrent neural network for solving time-varying multi-type constrained quadratic programming problems. *IEEE Trans. Neural Netw. Learn. Syst*. 32, 2993–3004. doi: 10.1109/TNNLS.2020.3009201

Zheng, Y., Liu, Q., and Wang, J. (2017). A specified-time convergent multiagent system for distributed optimization with a time-varying objective function. *IEEE Trans. Automat. Contr*. 62, 1590–1605. doi: 10.1109/TAC.2016.2593899

Zhu, W., and Wang, Q. (2024). Distributed finite-time optimization of multi-agent systems with time-varying cost functions under digraphs. *IEEE Trans. Netw. Sci. Eng*. 11, 556–565. doi: 10.1109/TNSE.2023.3301900

Zuo, Q., Li, K., Xiao, L., Wang, Y., and Li, K. (2022). On generalized zeroing neural network under discrete and distributed time delays and its application to dynamic lyapunov equation. *IEEE Trans. Syst. Man Cybern. Syst*. 52, 5114–5126. doi: 10.1109/TSMC.2021.3115555