



# New discrete-time zeroing neural network for solving time-dependent linear equation with boundary constraint

Naimeng Cang<sup>1</sup> · Feng Qiu<sup>1</sup> · Shan Xue<sup>1,2</sup> · Zehua Jia<sup>1</sup> · Dongsheng Guo<sup>1</sup> · Zhijun Zhang<sup>3</sup> · Weibing Li<sup>4</sup>

Accepted: 6 March 2024  
© The Author(s) 2024

## Abstract

Recently, continuous- and discrete-time models of a zeroing neural network (ZNN) have been developed to provide online solutions for the time-dependent linear equation (TDLE) with boundary constraint. This paper presents a novel approach to address the bound-constrained TDLE (BCTDLE) problem by proposing a new discrete-time ZNN (DTZNN) model. The proposed DTZNN model is designed using the Taylor difference formula to discretize the previous continuous-time ZNN (CTZNN) model. Theoretical analysis indicates the computational property of the proposed DTZNN model, and numerical results further demonstrate its validity. The applicability of the proposed DTZNN model is finally confirmed via its application to the motion planning of a PUMA560 robotic arm.

**Keywords** Discrete-time zeroing neural network · Time-dependent linear equation · Boundary constraint · Taylor difference formula · Robotic arm

## 1 Introduction

Linear and nonlinear equations are fundamental mathematical tools that play essential roles in several industrial applications, such as robot path planning, image recognition, and signal processing (Zhao 2013; Tsiligianni et al. 2015; Zhang and Jin 2017; Li et al. 2019). Extensive work has been conducted on solving linear and nonlinear equations, typically using numerical algorithms (Sharma 2005; Neta et al. 2014; Abdelmalek 1977; Donoho et al. 2012; Zeng et al. 2014; Spedicato et al. 2000; Morigi and Sgallari 2001) and

---

✉ Dongsheng Guo  
gdongsh2022@hainanu.edu.cn

<sup>1</sup> School of Information and Communication Engineering, Hainan University, Haikou, China

<sup>2</sup> School of System Design and Intelligent Manufacturing, South China University of Technology, Shenzhen, China

<sup>3</sup> School of Automation Science and Engineering, South China University of Technology, Guangzhou, China

<sup>4</sup> School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China

neurodynamic methods (Kumar 2022a, b; Kumar 2023; Cichocki et al. 1992; Xia et al. 1999; Liang and Tso 2002). Among the numerical algorithms, direct methods and iterative methods are mainly used. In Abdelmalek (1977), the minimum  $L_\infty$  solution of the linear equation was first proposed. In Donoho et al. (2012), the sparse approximate solution of the linear equation was obtained by using the stagewise orthogonal matching pursuit algorithm. Many iterative algorithms have also been developed to bring solutions to the linear equation, such as adaptive iterative thresholding algorithms (Zeng et al. 2014), abstraction algorithms (Spedicato et al. 2000), and regularizing Lanczos iterative algorithms (Morigi and Sgallari 2001). However, these numerical algorithms may encounter challenges such as the accumulation of numerical errors, slow convergence rates, and low stability (Zhang et al. 2013).

In recent years, neural network models have garnered widespread attention for addressing modeling and control problems in both linear and nonlinear dynamic systems (Zhang et al. 2018; Li et al. 2019; Xiao et al. 2020; Liao et al. 2024; Kumar 2022a, b; Kumar 2023). Particularly, the introduction of novel neural network models such as the Lyapunov-stability-based context-layered recurrent pi-sigma neural network (CLRPSNN) (Kumar 2022a), the memory recurrent Elman neural network (MRENN) (Kumar 2022b), and the higher-order recurrent neural network (HORNN) (Kumar 2023) have provided new approaches for solving the identification and control problems of nonlinear dynamic systems. The CLRPSNN model effectively addresses the issue of nonlinear system identification by introducing an additional layer of context nodes, demonstrating significant performance advantages in simulation results. However, stability has always been a crucial consideration in the design of neural network models. The MRENN and HORNN models innovate in stability design, ensuring stability and convergence of the models by combining Lyapunov stability criteria and recursive learning rate schemes. This emphasis on stability makes these models more reliable and practical when dealing with nonlinear dynamic systems.

Similar to numerical algorithms, neurodynamic methods also significantly improve the efficiency of solving the linear equation. Cichocki et al. (1992) proposed different recurrent neural networks (RNNs) to address the linear equation with an inequality constraint. Xia et al. (1999) designed an RNN model that can converge faster and provide more accurate solutions to the linear equation with an inequality constraint. The discrete-time form of such an RNN model was further deduced by Liang and Tso (2002). It is worth noting that these studies are based on the assumption of time-invariant when solving linear equations. However, many systems in practical engineering applications are always time-varying. Therefore, directly applying these methods to solve time-varying linear equations may yield poor results. Moreover, the mathematical domain of the variables involved in a linear equation must be set during the solution process (Zhang et al. 2018; Park et al. 1998). For example, it is assumed that many robotic arms can lead to task failure or even damage if the physical constraints of the joints are exceeded. Thus, appropriate methods must be investigated to ensure that joint angle, velocity, and/or acceleration are within the proper mathematical intervals (Guo et al. 2018; Zhang et al. 2013). In other words, solving time-varying underdetermined linear equations with boundary constraint is of importance in practical engineering applications.

In many industrial applications, there is a high demand for the real-time performance of linear equation, making them time-dependent. Existing studies are less concerned with methods for solving time-dependent linear equation (TDLE) with boundary constraint. To solve a given time-dependent problem, a representative RNN was presented and refined by Zhang and Guo (2015); Zhang and Yi (2011); Zhang et al. (2012, 2015). The main idea of this model was to define the error control equation and then derive the RNN (depicted in an ordinary equation) so that the computational error can converge globally and eventually become 0, hence the name zeroing neural network (ZNN). Focusing on solving the bound-constrained TDLE (BCTDLE), Xu et al. (2019a) developed the continuous-time ZNN (CTZNN) model, which they successfully employed to solve time-varying linear equations and inequality systems (Xu et al. 2019b). Sun and Liu (2020) designed a novel noise-resistant CTZNN model for solving time-varying Lyapunov equations. However, directly implementing the CTZNN model into hardware in practical engineering applications is challenging.

Therefore, for the purpose of possible hardware implementation, researchers discretize the CTZNN model using difference equations. The traditional numerical differentiation algorithm for discretizing continuous systems is typically implemented using the Euler difference formula. Specifically, in Guo and Zhang (2012), a one-step iteration description of the DTZNN model is proposed for dynamic matrix inversion. The simulation results show that this model has an error mode of  $O(\varepsilon^2)$ , that is, when the sampling interval  $\varepsilon$  decreases by a factor of 10, the steady-state calculation error (SSCE) decreases by a factor of 100. Unlike the traditional Euler difference formula, in Guo et al. (2017), a new Taylor-type difference rule is constructed. For the approximation of first-order derivatives, this rule has been proven to have a smaller SSCE than the Euler difference rule and possesses an error mode of  $O(\varepsilon^3)$ , that is, when the sampling interval  $\varepsilon$  decreases by a factor of 10, the SSCE decreases by a factor of 1000. Using the new Taylor difference rule, researchers have successively developed Taylor difference formulas with the error modes  $O(\varepsilon^4)$  and  $O(\varepsilon^5)$ . Specifically, Huang et al. (2022) developed a Taylor difference formula with an error mode of  $O(\varepsilon^4)$  to solve time-variant underdetermined nonlinear systems under bound constraint. Different from the proposed DTZNN model in this paper, their model focuses on solving nonlinear systems and may not necessarily be applicable to linear equations, whereas DTZNN model proposed in this paper exhibits superior accuracy. Cai et al. (2021) also discretized the CTZNN model based on Taylor series expansion. However, their model mainly focuses on solving a time-varying system of linear equation and inequality, and does not consider boundary constraint. A recent study (Ma and Guo 2021) showed that the DTZNN model can effectively solve the BCTDLE with an SSCE of the order  $O(\varepsilon^4)$ , where  $\varepsilon$  is the sampling distance.

Following but differing from the previous work (Ma and Guo 2021), this paper establishes a new DTZNN model that can effectively solve the TDLE with boundary constraint. The validity of such a model is supported by both theoretical and numerical results. Considering the importance of robotic arms in industrial applications (Xu et al. 2019a; Ma and Guo 2021; Zhang et al. 2019), this paper further applies the proposed DTZNN model to

the PUMA560 robotic arm to demonstrate the applicability of the model. The primary contributions of this paper can be summarized as follows:

- (1) The new DTZNN model, which has not been previously reported, is studied to solve the TDLE with boundary constraint. Notably, the proposed model is clearly different from the previous DTZNN model (Ma and Guo 2021), and achieves better performance on computing and solving the BCTDLE.
- (2) Theoretical analysis denotes the computational characteristic of the proposed DTZNN model, and numerical results verify its validity. More importantly, these results show that the DTZNN calculation error at steady state is of order  $O(\epsilon^5)$ . This is the first time that a computing model with  $O(\epsilon^5)$  mode is presented to solve the BCTDLE.
- (3) The proposed DTZNN model is utilized for robotic arms by solving the linear kinematic equation that involves joint physical constraint. Simulation results using the PUMA560 robotic arm with different examples demonstrate the effectiveness and practicality of the proposed DTZNN model.

The upcoming sections of this paper are structured as follows. Section 2 presents the problem statement and the ZNN models for the TDLE with boundary constraint. Section 3 describes the proposed DTZNN composition and theoretical analysis. Section 4 provides numerical validation experiments with the proposed DTZNN model. Section 5 shows the applicability of the DTZNN model to robotic arms with joint constraints. Section 6 concludes this paper.

## 2 Preliminary

This section outlines the problem to address the TDLE with boundary constraint. To probe the matter further, the continuous- and discrete-time models of ZNN are shown as basis.

### 2.1 Problem statement

The TDLE with boundary constraint considered in this paper is formulated as follows (Lu et al. 2019; Xu et al. 2019a):

$$\begin{cases} G(t)x(t) = h(t), \\ x^- \leq x(t) \leq x^+, \end{cases} \quad (1)$$

where matrix  $G(t) \in R^{m \times n}$  (with  $m < n$ ) is time-dependent and full-rank, vector  $h(t) \in R^m$  is time-dependent and smooth, vector  $x(t) \in R^n$  is unknown and must be obtained by solving (1), and  $x^\pm$  is the bounds of  $x(t)$ . This paper aims to find a viable  $x(t)$  that satisfies both the linear equation and the boundary constraint outlined in (1).

As shown in Xu et al. (2019b), solving the BCTDLE (1) can be transformed into finding the solution of the system as follows:

$$\begin{cases} G(t)x(t) - h(t) = 0, \\ Ux(t) - \vartheta + V(t)y(t) = 0, \end{cases} \quad (2)$$

with  $U = [-I; I] \in R^{2n \times n}$ ,  $\vartheta = [-x^-; x^+] \in R^{2n}$ , and  $V(t) = \text{diag}\{y_1(t), \dots, y_{2n}(t)\} \in R^{2n \times 2n}$ .  $I \in R^{n \times n}$  is the identity matrix, and  $y(t)$  is an unknown vector that must still be determined when solving (2). The matrix–vector form of (2) is then expressed as follows:

$$Q(t)w(t) - r(t) = 0, \tag{3}$$

with  $Q(t) \in R^{(m+2n) \times (3n)}$ ,  $w(t) \in R^{(3n)}$ , and  $r(t) \in R^{(m+2n)}$  being

$$\begin{aligned} Q(t) &= \begin{bmatrix} G(t) & 0 \\ U & V(t) \end{bmatrix}, \\ w(t) &= \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}, \\ r(t) &= \begin{bmatrix} h(t) \\ g \end{bmatrix}. \end{aligned}$$

In this way, solving the BCTDLE (1) is equivalent to solving the matrix–vector equation presented in (3) for  $t \geq 0$ .

### 2.2 CTZNN model

In this subsection, to solve the BCTDLE (1), following the ZNN design principle (Xu et al. 2019b), the error equation  $e(t) \in R^{(m+2n)}$  is defined by

$$e(t) = Q(t)w(t) - r(t). \tag{4}$$

Then, a decay exponent formula is introduced to achieve the convergence of  $e(t)$  to 0 (Xu et al. 2019b), and the resultant CTZNN model is derived and formulated as follows (Lu et al. 2019):

$$\dot{w}(t) = -P^+(t)(\lambda(Q(t)w(t) - r(t)) + M(t)w(t) - \dot{r}(t)), \tag{5}$$

where vectors  $\dot{w}(t)$  and  $\dot{r}(t)$  denote the time derivatives of  $w(t)$  and  $r(t)$ , respectively, and matrices  $P(t) \in R^{(m+2n) \times (3n)}$  and  $M(t) \in R^{(m+2n) \times (3n)}$  are

$$P(t) = \begin{bmatrix} G(t) & 0 \\ U & 2V(t) \end{bmatrix}, \text{ and } M(t) = \begin{bmatrix} \dot{G}(t) & 0 \\ 0 & 0 \end{bmatrix},$$

with  $P^+(t) \in R^{(3n) \times (m+2n)}$  being the right pseudoinverse matrix of  $P(t)$ . The following theoretical conclusion regarding the CTZNN model (5) is given and proved in Xu et al. (2019b).

**Lemma 1** *When considering a solvable BCTDLE (1), the CTZNN model (5) can produce an exact time-dependent solution of (1).*

### 2.3 DTZNN model

The study of discrete-time models of ZNN is of practical interest because of hardware implementation and development of numerical algorithms (Guo et al. 2017; Mathews and Fink 2004). The widely-used method for deriving a discrete-time model is by using the Euler difference formula (Mathews and Fink 2004), which is given by

$$\dot{w}_k = \dot{w}(t_k = k\varepsilon) = \frac{w_{k+1} - w_k}{\varepsilon} + O(\varepsilon), \quad (6)$$

where  $w_k = w(t_k = k\varepsilon)$ ,  $\varepsilon = t_{k+1} - t_k$  represents the sampling interval,  $k = 0, 1, 2, \dots$  is the number of iterations, and  $O(\varepsilon)$  is the truncation error.

Evidently, the discretization of the CTZNN model (5) via Euler difference formula (6) yields the following expression:

$$w_{k+1} = w_k - \mu P_k^+(Q_k w_k - r_k) - \varepsilon P_k^+(M_k w_k - \dot{r}_k) + O(\varepsilon^2), \quad (7)$$

where  $P_k^+ = P^+(t_k = k\varepsilon)$ ,  $Q_k = Q(t_k = k\varepsilon)$ ,  $M_k = M(t_k = k\varepsilon)$ ,  $\dot{r}_k = \dot{r}(t_k = k\varepsilon)$ ,  $r_k = r(t_k = k\varepsilon)$ , and  $\mu = \lambda\varepsilon > 0$  is the step size.

By eliminating  $O(\varepsilon^2)$  from (7), the following DTZNN model to solve the BCTDLE (1) is derived below:

$$w_{k+1} = w_k - u P_k^+(Q_k w_k - r_k) - \varepsilon P_k^+(M_k w_k - \dot{r}_k). \quad (8)$$

In relation to the DTZNN model (7), it is a variation law with an  $O(\varepsilon^2)$  error form. That is, when the sampling interval  $\varepsilon$  is reduced by a factor of 10, the stationary error is reduced by a factor of 100. Such a DTZNN model may not satisfy the high-precision requirement in practice. Thus, a new DTZNN model with better performance on computing and solving the BCTDLE (1) is proposed in this paper.

### 3 New DTZNN model

In this section, on the basis of the Taylor difference formula (Cai et al. 2021), the new DTZNN model is developed to solve the BCTDLE (1). Theoretical analysis is also provided to denote the computational characteristic of the proposed DTZNN model.

#### 3.1 Model formulation

The approximation of a first order derivative via Taylor series expansion has been the subject of numerous studies (Zhang et al. 2019). In the previous study (Cai et al. 2021), the Taylor difference formula that has higher accuracy and lower steady-state error than the conventional Euler difference formula (6) has been constructed.

**Lemma 2** *The following Taylor difference formula (Cai et al. 2021) is shown to approximate the first order derivative:*

$$\begin{aligned} \dot{f}(t_k) = & (54f(t_{k+1}) - 5f(t_k) - 10f(t_{k-1}) - 55f(t_{k-2}) + 10f(t_{k-3}) \\ & + f(t_{k-4}) + 10f(t_{k-5}) - 5f(t_{k-6}))/ (120\varepsilon) + O(\varepsilon^4), \end{aligned} \quad (9)$$

with  $k = 6, 7, 8, \dots$  and  $f(\cdot)$  being the objective function.

To discretize the CTZNN model (5) using the above Taylor difference formula (9), the following vector form of (9) is presented for the approximation of  $\dot{w}_k$ :

$$\begin{aligned} \dot{w}_k = & (54w_{k+1} - 5w_k - 10w_{k-1} - 55w_{k-2} + 10w_{k-3} \\ & + w_{k-4} + 10w_{k-5} - 5w_{k-6}) / (120\varepsilon) + O(\varepsilon^4). \end{aligned} \tag{10}$$

By applying the Taylor difference formula (10) to the discretization of the CTZNN model (5), the discrete-time expression is given by

$$\begin{aligned} w_{k+1} = & \frac{5}{54}w_k + \frac{5}{27}w_{k-1} + \frac{55}{54}w_{k-2} - \frac{5}{27}w_{k-3} - \frac{1}{54}w_{k-4} - \frac{5}{27}w_{k-5} \\ & + \frac{5}{54}w_{k-6} - \frac{20}{9}P_k^+(\mu(Q_k w_k - r_k) + \varepsilon M_k w_k - \varepsilon \dot{r}_k) + O(\varepsilon^5). \end{aligned} \tag{11}$$

By eliminating  $O(\varepsilon^5)$  from (11), the DTZNN model proposed in this paper to solve the BCTDLE (1) is expressed as follows:

$$\begin{aligned} w_{k+1} = & \frac{5}{54}w_k + \frac{5}{27}w_{k-1} + \frac{55}{54}w_{k-2} - \frac{5}{27}w_{k-3} - \frac{1}{54}w_{k-4} - \frac{5}{27}w_{k-5} \\ & + \frac{5}{54}w_{k-6} - \frac{20}{9}P_k^+(\mu(Q_k w_k - r_k) + \varepsilon M_k w_k - \varepsilon \dot{r}_k). \end{aligned} \tag{12}$$

Regarding the proposed DTZNN model (12), it requires seven initial states (i.e.,  $w_0, w_1, w_2, w_3, w_4, w_5$  and  $w_6$ ) to activate the iterative computation. Thus, given an initial value  $w_0$ , the rest are found by the DTZNN model (8) and are computed as follows:

$$\begin{cases} w_1 = w_0 - \mu P_0^+(Q_0 w_0 - r_0) - \varepsilon P_0^+(M_0 w_0 - \dot{r}_0), \\ w_2 = w_1 - \mu P_0^+(Q_1 w_1 - r_1) - \varepsilon P_0^+(M_1 w_1 - \dot{r}_1), \\ w_3 = w_2 - \mu P_0^+(Q_2 w_2 - r_2) - \varepsilon P_0^+(M_2 w_2 - \dot{r}_2), \\ w_4 = w_3 - \mu P_0^+(Q_3 w_3 - r_3) - \varepsilon P_0^+(M_3 w_3 - \dot{r}_3), \\ w_5 = w_4 - \mu P_0^+(Q_4 w_4 - r_4) - \varepsilon P_0^+(M_4 w_4 - \dot{r}_4), \\ w_6 = w_5 - \mu P_0^+(Q_5 w_5 - r_5) - \varepsilon P_0^+(M_5 w_5 - \dot{r}_5). \end{cases}$$

The procedure of the proposed DTZNN model (12) to solve the BCTDLE problem (1) is as follows:

- (i) Initialization: Given the time duration  $T$ , sampling gap  $\varepsilon$ , step size  $\mu$ , constraint boundary  $[-x^-, x^+]$ . Initialize  $t_0, w_0, G_0, \dot{G}_0, h_0$ , and  $\dot{h}_0$ . Receive  $P_0^+, r_0, \dot{r}_0, Q_0$  and  $M_0$ . Compute  $\|e_0\|_2 = \|Q_0 w_0 - r_0\|_2$ .
- (ii) First Loop (with  $k = 0, 1, 2, 3, 4, 5$ ): Compute  $w_{k+1}$  through (8). Receive  $P_{k+1}^+, r_{k+1}, \dot{r}_{k+1}, Q_{k+1}$  and  $M_{k+1}$ . Compute  $\|e_{k+1}\|_2 = \|Q_{k+1} w_{k+1} - r_{k+1}\|_2$ .
- (iii) Second Loop (with  $k = 6, \dots, \text{int}(T/\varepsilon)$ ): Compute  $w_{k+1}$  through (12). Receive  $P_{k+1}^+, r_{k+1}, \dot{r}_{k+1}, Q_{k+1}$  and  $M_{k+1}$ . Compute  $\|e_{k+1}\|_2 = \|Q_{k+1} w_{k+1} - r_{k+1}\|_2$ .
- (iv) Output: Save  $w_k$  and  $\|e_k\|_2 = \|Q_k w_k - r_k\|_2$ , and plot figures.

An important criterion for measuring the performance of numerical algorithms is computational complexity. For (12), the overall computational complexity is  $O((m + 2n) \times (3n))$  (with  $m < n \in R$ ). Thus, the proposed DTZNN model (12) has low computational complexity, i.e.,  $O(n^2)$ .

### 3.2 Theoretical analysis

In this subsection, the computational characteristic of the proposed DTZNN model (12) is analyzed and proved theoretically.

**Lemma 3** *The proposed DTZNN model (12) is characterized by zero stability and consistency, and thus has convergence property.*

**Proof** By analyzing the characteristic polynomial, it has been determined that the proposed DTZNN model in Equation (12) exhibits zero stability (Guo et al. 2018). Then, from the derivation of (12), its truncation error is  $O(\varepsilon^5)$ , reflecting its consistency. Referring to the findings of Griffiths and Higham (2010), the zero stability and consistency are essential for the convergence of the proposed DTZNN model (12). With that, the proof is now fully established.  $\square$

**Lemma 4** *The proposed DTZNN model (12) can produce a precise time-dependent solution of the solvable BCTDLE (1).*

**Proof** By following Lemma 3, the solution  $w_k$  computed by the proposed DTZNN model (12) can converge to a theoretical solution  $w_k^* = w^*(t_k = k\varepsilon)$  of (3). Mathematically,  $w_k \rightarrow w_k^*$  if  $k$  is large enough (Huang et al. 2022). By virtue of  $Q_k w_k - r_k = 0$  and the given definitions of  $Q_k$  and  $r_k$ , the following equation can be deduced:

$$\lim_{k \rightarrow \infty} \left( \begin{bmatrix} G_k x_k - h_k \\ U x_k - \vartheta \end{bmatrix} \right) = \lim_{k \rightarrow \infty} \left( - \begin{bmatrix} 0 \\ V_k y_k \end{bmatrix} \right). \quad (13)$$

$\square$

Knowing that  $V_k y_k \geq 0$ ,  $U = [-I; I] \in R^{2n \times n}$ , and  $\vartheta = [-x^-; x^+] \in R^{n \times n}$ , (13) is reformulated as stated below:

$$\begin{cases} \lim_{k \rightarrow \infty} (G_k x_k - h_k) = 0, \\ x^- \leq \lim_{k \rightarrow \infty} (x_k) \leq x^+. \end{cases} \quad (14)$$

(14) indicates that, when  $k$  is large enough, the DTZNN model (12) has  $x_k \rightarrow x_k^*(t_k = k\varepsilon)$  for the BCTDLE (1) to hold. This statement further shows that the proposed DTZNN model (12) can offer a precise time-dependent solution of the solvable BCTDLE (1). With that, the proof is now fully established.  $\square$

**Lemma 5** *Considering the BCTDLE (1) is solved by the proposed DTZNN model (12), the SSCE varies in the mode of  $O(\varepsilon^5)$ .*

**Proof** Based on the error function (4), the SSCE of (12) is given by

$$\lim_{k \rightarrow \infty} \|e_k\|_2 = \lim_{k \rightarrow \infty} \|Q_k w_k - r_k\|_2.$$



According to Lemmas 3 and 4,  $w_k = w_k^* + O(\epsilon^5)$  with a large enough  $k$ . Then, the following result on the SSCE of (12) is obtained:

$$\begin{aligned} \lim_{k \rightarrow \infty} \|e_k\|_2 &= \lim_{k \rightarrow \infty} \|Q_k w_k - r_k\|_2 \\ &= \lim_{k \rightarrow \infty} \|Q_k(w_k^* + O(\epsilon^5)) - r_k\|_2 \\ &= \lim_{k \rightarrow \infty} \|Q_k O(\epsilon^5)\|_2 \\ &= O(\epsilon^5). \end{aligned} \tag{15}$$

□which means that the DTZNN calculation error at steady state is of order  $O(\epsilon^5)$ . Therefore, considering the BCTDLE (1) is solved by the proposed DTZNN model (12), the SSCE varies in the mode of  $O(\epsilon^5)$ . With that, the proof is now fully established. □

To summarize, Lemmas 3–5 provide theoretical guarantees that the proposed DTZNN model (12) can effectively solve the BCTDLE (1).

### 4 Numerical verification and comparison

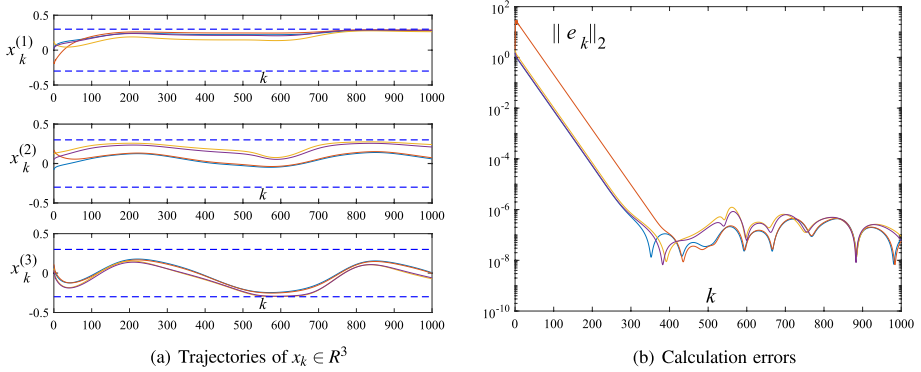
This section showcases numerical simulations to verify the superiority and effectiveness of the proposed DTZNN model (12) in solving the BCTDLE (1). Notably, the recent research (Guo et al. 2017) indicates that the DTZNN model via Taylor difference formula can outperform the DTZNN model (8) via Euler difference formula. Thus, this section only offers the comparative numerical results of using the proposed DTZNN model (12) and the previous DTZNN model in Guo et al. (2017). For convenience, the previous DTZNN model in Guo et al. (2017) is expressed as follows:

$$\begin{aligned} w_{k+1} &= \frac{5}{24}w_k + \frac{1}{2}w_{k-1} + \frac{1}{4}w_{k-2} + \frac{1}{6}w_{k-3} - \frac{1}{8}w_{k-4} \\ &\quad - P_k^+(\mu(Q_k w_k - r_k) + 2\epsilon(M_k w_k - \dot{r}_k)). \end{aligned} \tag{16}$$

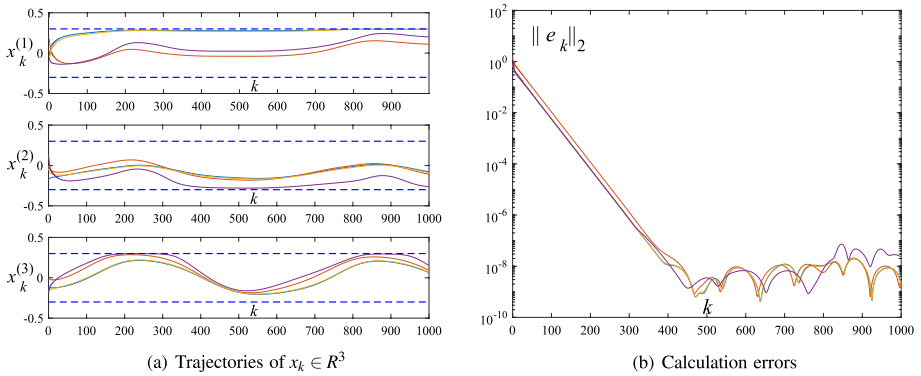
These simulations are also performed via MATLAB R2021b on a digital computer equipped with an AMD Ryzen 7 5800 H processor with Radeon Graphics @3.20 GHz, 32 GB of memory, and Windows 10 OS.

#### 4.1 Model comparison

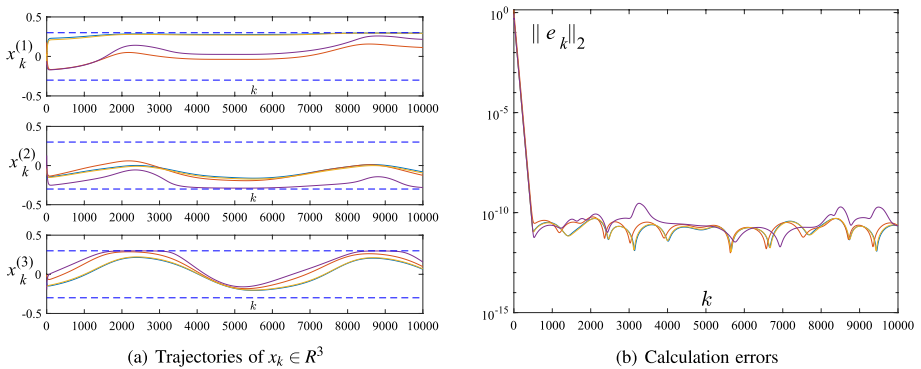
In this subsection, numerical simulation comparative experiments are conducted using the following example to demonstrate the effectiveness and superiority of the proposed DTZNN model (12) in solving BCTDLE (1). In the ensuing numerical experiments, the DTZNN models (12) and (16) are used to solve the BCTDLE (1) with the following coefficients:



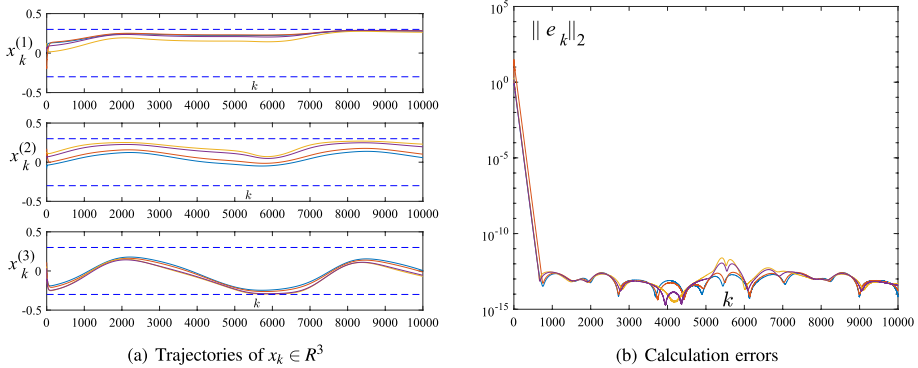
**Fig. 1** Numerical results for the previous DTZNN model (16) with  $\epsilon = 0.01$  and  $\mu = 0.1$  to solve the BCT-DLE (1)



**Fig. 2** Numerical results for the proposed DTZNN model (12) with  $\epsilon = 0.01$  and  $\mu = 0.1$  to solve the BCT-DLE (1)



**Fig. 3** Numerical results for the previous DTZNN model (16) with  $\epsilon = 0.001$  and  $\mu = 0.1$  to solve the BCT-DLE (1)



**Fig. 4** Numerical results for the proposed DTZNN model (12) with  $\epsilon = 0.001$  and  $\mu = 0.1$  to solve the BCTDLE (1)

**Table 1** SSNEs of the proposed DTZNN model (12) and the previous DTZNN model (16) using different  $\epsilon$  and  $\mu$  to solve the BCTDLE (1)

#	Model	$\mu = 0.1$	$\mu = 0.15$	$\mu = 0.2$	Mode
$\epsilon = 0.01$	DTZNN (16)	$3.99467 \times 10^{-7}$	$2.83945 \times 10^{-7}$	$2.17668 \times 10^{-7}$	$O(\epsilon^4)$
	DTZNN (12)	$4.75485 \times 10^{-9}$	$3.22147 \times 10^{-9}$	$2.65792 \times 10^{-9}$	$O(\epsilon^5)$
$\epsilon = 0.001$	DTZNN (16)	$4.21474 \times 10^{-11}$	$2.78023 \times 10^{-11}$	$2.0733 \times 10^{-11}$	$O(\epsilon^4)$
	DTZNN (12)	$1.34232 \times 10^{-13}$	$9.53207 \times 10^{-14}$	$7.35412 \times 10^{-14}$	$O(\epsilon^5)$

$$\begin{aligned}
 G(t) &= [1 + \sin t, 2 + \cos(2t), 5 - \sin t], \\
 h(t) &= \sin(0.5t) - \cos(2t), \\
 -x^+ &= x^- = [-0.3, -0.3, -0.3].
 \end{aligned}$$

The relevant numerical results are provided in Figs. 1, 2, 3, 4 and Table 1.

Figure 1 presents the simulation results of the previous DTZNN model (16) using  $\epsilon = 0.01$  and  $\mu = 0.1$  and four different initial values to solve the BCTDLE (1), where  $x_k^{(i)}$  ( $i = 1, 2, 3$ ) indicates the  $i$ th element of the feasible solution vector  $x_k$  of (1). In Fig. 1a, the trajectories of  $x_k$  are time-dependent, and the changing values are within the constraint boundary  $[x^-, x^+]$ . These results mean that the exact solution of (1) can be obtained by applying the previous DTZNN model (16). In Fig. 1b, the calculation errors  $\|e_k\|_2$  of (16) converge, and the SSCEs are in the order of  $10^{-6}$ . According to these results, the previous DTZNN model (16) can effectively solve the BCTDLE (1).

Figure 2 presents the simulation results of the proposed DTZNN model (12) using  $\epsilon = 0.01$  and  $\mu = 0.1$  and four different initial values to solve the BCTDLE (1). In Fig. 2a, the trajectories of  $x_k$  are time-dependent as well and numerically bounded (or say, they satisfy  $x_k \in [x^-, x^+]$ ). In Fig. 2b, the calculation errors of (12) demonstrate convergence, with the SSCE values reaching the order of  $10^{-8}$ . These findings provide strong evidence for the validity of (12) in solving (1). More importantly, comparing the results in Figs. 1b and 2b,

the SSCE of the proposed model (12) is 100 times smaller than that of the previous model (16). This comparison indicates that (12) has smaller errors and higher accuracy than (16). Therefore, the proposed DTZNN model (12) offers greater advantages than the previous DTZNN model (16) in solving the TDLE with boundary constraint, i.e., (1).

By reducing the sampling interval  $\varepsilon$  by a factor of 10 and repeating the above numerical experiments, the corresponding results of using the DTZNN models (16) and (12) are shown in Figs. 3 and 4, respectively. In such two figures, as computed by (16) or (12), the trajectories of  $x_k$  are time-dependent and satisfy  $x_k \in [x^-, x^+]$ . The related computational errors converge with the SSCE being small enough. These simulation results verify again the effectiveness of the DTZNN models (16) and (12) in solving the BCTDLE (1). In particular, comparing Figs. 3b and 4b, the SSCEs of (16) and (12) for solving (1) are  $10^{-10}$  and  $10^{-13}$  orders of magnitude, respectively. The latter is about 1000 times smaller than the former. Therefore, the proposed DTZNN model (12) can be more advantageous than the previous DTZNN model (16) in solving the BCTDLE (1).

For further investigation, the SSCEs of the DTZNN models (16) and (12) are compared and validated using different values of  $\varepsilon$  and  $\mu$ , with a fixed initial state of  $w_0 = 0.2$  and the same sampling time. The detailed data are provided in Table 1, which demonstrates that the proposed model (12) is computationally better than the previous model (16). These results also verify that the calculation error variation mode of (12) is  $O(\varepsilon^5)$ . Specifically, when  $\varepsilon = 0.001$ , the SSCE can reach the order of  $10^{-14}$ . In addition to this, the performance of the proposed DTZNN model (12) on computing and solving (1) can be improved by decreasing the sampling interval  $\varepsilon$  or increasing  $\mu$  in the appropriate range.

Overall, the simulation results presented in Figs. 1, 2, 3, 4 and Table 1 confirm the superiority and efficacy of the proposed DTZNN model (12) in solving the BCTDLE (1) in comparison with the previous DTZNN model (16).

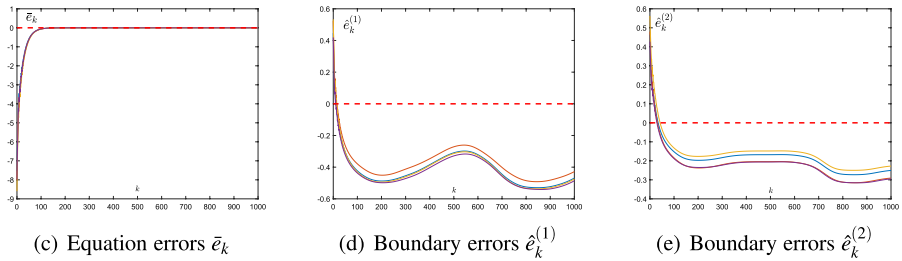
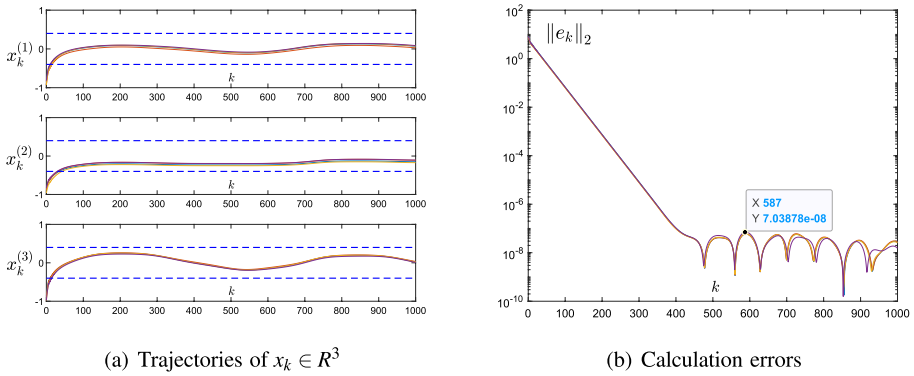
## 4.2 Influence of $\varepsilon$ and $\mu$

In the previous section, the effectiveness and superiority of the proposed DTZNN model (12) in solving the BCTDLE problem are demonstrated through simulation comparisons. The proposed DTZNN model (12), it has two adjustable parameters (i.e.,  $\varepsilon$  and  $\mu$ ). Since different values of  $\varepsilon$  and  $\mu$  can have an effect on the solution effectiveness, this section delves into the influence of these two adjustable parameters on the proposed DTZNN model (12) using a new example. The time period for the simulation is set to  $T = 10$  s. The coefficient matrix and the constraint boundary of the BCTDLE (1) to be solved are given as follows:

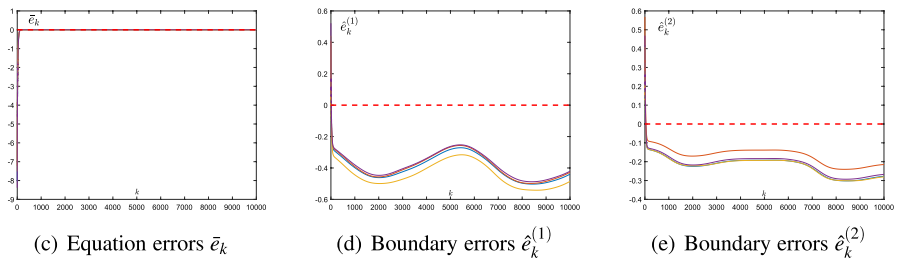
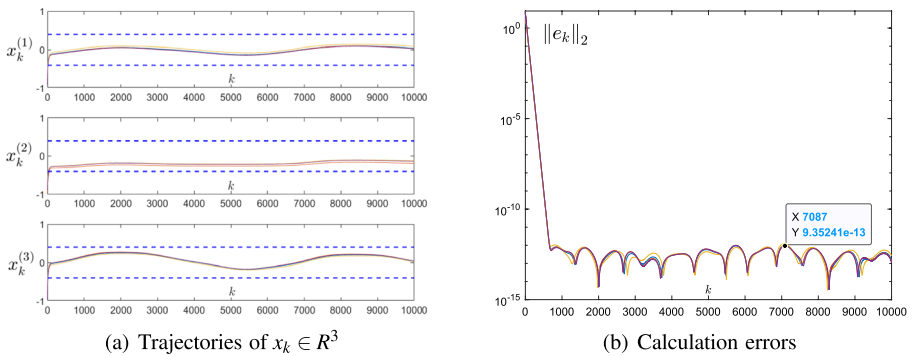
$$\begin{aligned} G(t) &= [3 + \cos(3t), 1 + \sin(t), 6 - \cos(t) - \sin(t)], \\ h(t) &= [\sin(t) - \cos(t)], \\ -x^+ &= x^- = [-0.4, -0.4, -0.4]. \end{aligned}$$

To better demonstrate the effectiveness of the proposed DTZNN model (12) in solving BCTDLE (1), two additional computational errors are introduced: the equation error  $\bar{e}_k = G(t)x(t) - h(t)$  and the constraint boundary error  $\hat{e}_k = Ux(t) - \vartheta$ .

Figure 5a clearly illustrates that the feasible solution of the BCTDLE (1) under the influence of the proposed DTZNN (12) model, undergoes rapid convergence from an initially



**Fig. 5** Numerical results for the proposed DTZNN model (12) with  $\varepsilon = 0.01$  and  $\mu = 0.1$  to solve the BCT-DLE (1)



**Fig. 6** Numerical results for the proposed DTZNN model (12) with  $\varepsilon = 0.001$  and  $\mu = 0.1$  to solve the BCTDLE (1)

out-of-boundary state and reaches the boundary constraint range. This result clearly validates the effectiveness and convergence properties of the proposed DTZNN model (12). Figure 5b indicates that when the iteration count  $k$  is sufficiently large, the SSCE  $\|e_k\|_2$  converges, and the maximum value is in the smaller order of magnitude of  $10^{-8}$  after stabilization. This outcome suggests that the proposed DTZNN model (12) can effectively solve BCTDLE (1). From Fig. 5c, it can be seen that with increasing iteration count  $k$  and after reaching stability, the equation error  $\bar{e}_k$  converges to 0. The trend of the variation in  $\hat{e}_k$  in Fig. 5d and e also illustrates that under the influence of the proposed DTZNN model (12),  $x_k$  transitions from initially exceeding the boundary to returning within the boundary constraint range. Therefore, in Fig. 5a,  $x_k$  is the exact solution of BCTDLE (1), confirming that the proposed DTZNN model (12) can effectively solve BCTDLE (1).

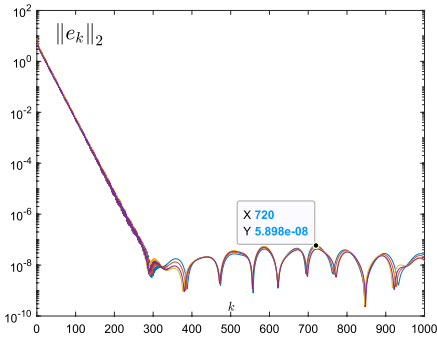
The above simulation results are obtained with the sampling interval  $\varepsilon = 0.01$ . To showcase the high precision characteristics of the proposed DTZNN model (12) and highlight the impact of the sampling interval on the solution effectiveness, the sampling interval is further reduced to  $\varepsilon = 0.001$ , while keeping other parameters constant. The relevant simulation results for solving BCTDLE (1) are presented below.

Figures 6a, d, and e reveal that even with four different initial values exceeding the boundary constraints,  $x_k$  can rapidly converge within the boundary constraint range under the influence of the proposed DTZNN model (12). Comparing Fig. 5b with Fig. 6b, it is also noticeable that as the sampling interval  $\varepsilon$  decreases, the computational performance improves. Specifically,  $(9.35241 \times 10^{-13}) / (7.03878 \times 10^{-8}) \approx 1.32870 \times 10^{-5}$ , indicating that the proposed DTZNN model (12) follows an  $O(\varepsilon^5)$  error model, that is, as the sampling interval  $\varepsilon$  is reduced by a factor of 10, the SSCE can be reduced by a factor of 100000. Therefore, it is possible to appropriately decrease the value of  $\varepsilon$  based on practical requirements to enhance computational accuracy and obtain more precise time-varying solutions.

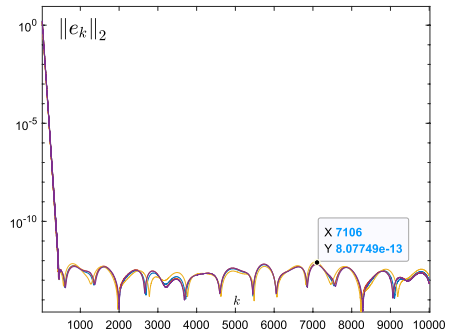
In addition to the sampling interval  $\varepsilon$ , the proposed DTZNN model (12) has another adjustable parameter,  $\mu$ . To further explore the influence of the two parameters on the proposed DTZNN model (12), six sets of simulation experiments are conducted. From Fig. 7a, c, and e, it can be observed that with the sampling interval  $\varepsilon = 0.01$ , the SSCEs of the proposed DTZNN model (12) for solving BCTDLE (1) decreases as the step size  $\mu$  increases. This similar conclusion can be obtained from Fig. 7b, d, and f. While the impact of increasing  $\mu$  on the SSCE is not as pronounced as decreasing  $\varepsilon$ , in practical situations, the sampling interval cannot be made arbitrarily small owing to some constraints. Therefore, when adjusting the sampling interval to the minimum achievable under practical conditions, increasing  $\mu$  can be employed to further reduce the SSCE and enhance precision.

By comparing the simulation results between the left and right of Fig. 7, it can be observed that, with the step size  $\mu$  held constant, as the sampling interval  $\varepsilon$  decreases from 0.01 to 0.001, the SSCEs reduce from the order of  $10^{-8}$  to  $10^{-13}$ . This magnitude reduction once again confirms that the proposed DTZNN model (12) exhibits an error model of  $O(\varepsilon^5)$ . Therefore, the numerical simulation results above strongly demonstrate the effectiveness and superiority of the proposed DTZNN model (12) in solving BCTDLE (1).

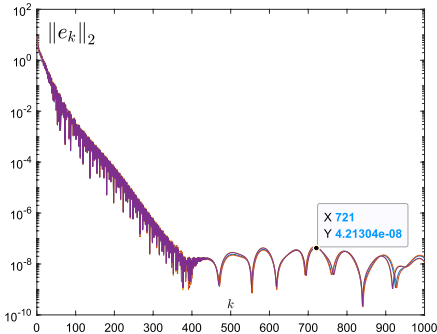
It is worth noting that the main difference in this section compared to the previous one lies in the fact that the initial values exceed the boundary constraint. Therefore, the comparison of simulation results show that while keeping the sampling interval  $\varepsilon$  and the step size  $\mu$  constant, the SSCE in this section is reduced by a factor of 10 compared to the



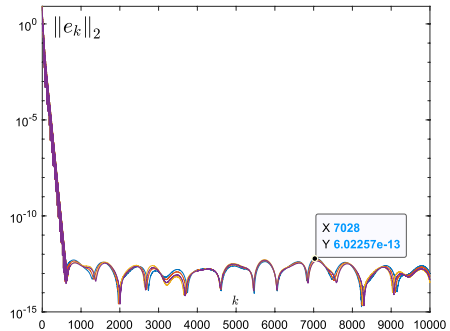
(a)  $\mu = 0.15, \varepsilon = 0.01$



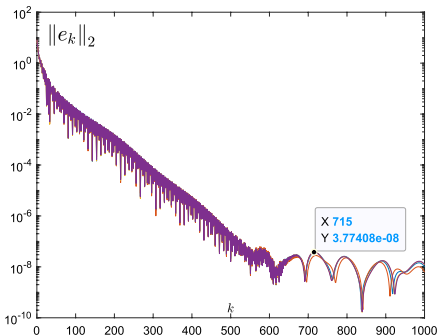
(b)  $\mu = 0.15, \varepsilon = 0.001$



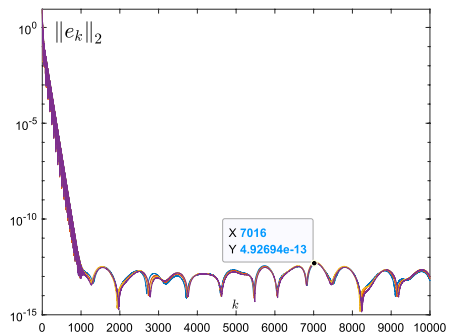
(c)  $\mu = 0.2, \varepsilon = 0.01$



(d)  $\mu = 0.2, \varepsilon = 0.001$



(e)  $\mu = 0.25, \varepsilon = 0.01$



(f)  $\mu = 0.25, \varepsilon = 0.001$

**Fig. 7** Calculation errors for the proposed DTZNN model (12) with different  $\varepsilon$  and  $\mu$  to solve the BCTDLE (1)

previous section. However, both sections still keep an  $O(\epsilon^5)$  error model. Such a phenomenon occurs not only in the proposed DTZNN model (12) but also in the previous DTZNN model (16), where model accuracy is sacrificed to ensure regression to within the boundary constraint. However, the proposed DTZNN model has smaller calculation errors and demonstrates greater fault tolerance, thus making it more widely applicable in practical engineering scenarios.

## 5 DTZNN application to robotics

This section presents the application of the proposed DTZNN model (12) to the robotic arms with joint physical constraints to demonstrate the applicability of the model.

### 5.1 Motion planning of robotic arms

The motion planning of a robotic arm involves generating the joint trajectory  $\theta(t) \in R^n$  in real-time to accurately follow the desired Cartesian path  $r(t) \in R^m$  of the end-effector (Li et al. 2019).

In particular, taking into consideration the feedback and joint limits, the motion planning of robotic arms is achieved by efficiently solving the TDLE with boundary constraints as follows (Xu et al. 2019b):

$$\begin{cases} J(\theta(t))\dot{\theta}(t) = \dot{r}(t) - k(\phi(\theta(t)) - r(t)), \\ \dot{\theta}^- \leq \dot{\theta}(t) \leq \dot{\theta}^+, \\ \theta^- \leq \theta(t) \leq \theta^+, \end{cases} \tag{17}$$

The above equation involves several key variables, including Jacobian matrix  $J(\theta(t)) \in R^{m \times n}$ , joint velocity  $\dot{\theta}(t) \in R^n$ , time derivative of  $r(t)$  denoted as  $\dot{r}(t) \in R^m$ , the feedback parameter  $k > 0 \in R$ , and the differentiable nonlinear mapping function  $\phi(\cdot)$ . In addition,  $\theta^\pm$  and  $\dot{\theta}^\pm$  represent the limits of  $\theta(t)$  and  $\dot{\theta}(t)$ , respectively.

It follows from (Zhang et al. 2004) that the boundary constraints in (17) are unified as

$$\max \{ \rho(\theta^- - \theta), \dot{\theta}^- \} \leq \dot{\theta} \leq \min \{ \rho(\theta^+ - \theta), \dot{\theta}^+ \},$$

with  $\rho > 0 \in R$ . Then, the following reformulation of (17) is further obtained:

$$\begin{cases} J(\theta(t))\dot{\theta}(t) = \dot{r}(t) - k(\phi(\theta(t)) - r(t)), \\ \delta^- \leq \dot{\theta}(t) \leq \delta^+, \end{cases} \tag{18}$$

with  $\delta^- = \max \{ \rho(\theta^- - \theta), \dot{\theta}^- \}$  and  $\delta^+ = \min \{ \rho(\theta^+ - \theta), \dot{\theta}^+ \}$ . At this point, solving (18) is the same as solving the BCTDLE (1), with the correlation coefficients as follows:

$$\begin{aligned} G(t) &= J(\theta(t)), \\ h(t) &= \dot{r}(t) - k(\phi(\theta(t)) - r(t)), \\ x(t) &= \dot{\theta}(t). \end{aligned}$$



Moreover, the following augmented coefficient matrices and vectors are presented:

$$\begin{aligned}
 U &= \begin{bmatrix} -I \\ I \end{bmatrix}, \quad g = \begin{bmatrix} -\delta^- \\ \delta^+ \end{bmatrix}, \\
 P(t) &= \begin{bmatrix} G(t) & 0 \\ U & 2V(t) \end{bmatrix} = \begin{bmatrix} J(\theta(t)) & 0 \\ U & 2V(t) \end{bmatrix}, \\
 r(t) &= \begin{bmatrix} h(t) \\ g \end{bmatrix} = \begin{bmatrix} \dot{r}(t) - k(\phi(\theta(t)) - r(t)) \\ g \end{bmatrix}, \\
 Q(t) &= \begin{bmatrix} G(t) & 0 \\ U & V(t) \end{bmatrix} = \begin{bmatrix} J(\theta(t)) & 0 \\ U & V(t) \end{bmatrix}, \\
 M(t) &= \begin{bmatrix} \dot{G}(t) & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \dot{J}(\theta(t)) & 0 \\ 0 & 0 \end{bmatrix}, \\
 w(t) &= \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} \dot{\theta}(t) \\ y(t) \end{bmatrix}.
 \end{aligned}$$

Therefore, the proposed DTZNN model (12) for the BCTDLE (1) can be introduced to solve (18) and (17), where  $w_k = [\dot{\theta}_k; y_k] \in R^{3n}$ . Notably, the Taylor difference formula (10) is also employed to calculate the joint angle  $\theta_k = \theta(t_k = k\varepsilon)$  at every instance. On the basis of (12), the following detailed formulation for realizing the motion planning of robotic arms with joint physical constraints is provided:

$$\begin{cases}
 \theta_{k+1} = \frac{5}{54}\theta_k + \frac{5}{27}\theta_{k-1} + \frac{55}{54}\theta_{k-2} - \frac{5}{27}\theta_{k-3} - \\
 \quad \frac{1}{54}\theta_{k-4} - \frac{5}{27}\theta_{k-5} + \frac{5}{54}\theta_{k-6} - \frac{20}{9}\varepsilon\dot{\theta}_k, \\
 w_{k+1} = \frac{5}{54}w_k + \frac{5}{27}w_{k-1} + \frac{55}{54}w_{k-2} - \frac{5}{27}w_{k-3} \\
 \quad - \frac{1}{54}w_{k-4} - \frac{5}{27}w_{k-5} + \frac{5}{54}w_{k-6} - \\
 \quad \frac{20}{9}P_k^+(\mu(Q_k w_k - r_k) + \varepsilon M_k w_k - \varepsilon \dot{r}_k).
 \end{cases} \tag{19}$$

Hereafter, (19) is called the new motion planning scheme for physically-constrained robotic arms. With regard to such a motion planning scheme (19), seven initial values must still be determined to start the iterative computation. Similarly, given the initial joint angle  $\theta_0$  and  $w_0$ , the rest is obtained via the following iterative computation:

$$\begin{cases}
 \theta_{i+1} = \theta_i + \varepsilon\dot{\theta}_i, \\
 w_{i+1} = w_i - \varepsilon P_i^+(Q_i w_i - r_i) - \varepsilon P_i^+(M_i w_i - \dot{r}_i),
 \end{cases}$$

where  $i = 0, 1, \dots, 6$ .

### 5.2 Simulation verification

To demonstrate the effectiveness of the new motion planning scheme (19), simulations are conducted using the PUMA560 robotic arm for a set desired path [i.e., the tricuspid path (20), circular path (21), and Rhodonea path (22)]. In this way, the validity and practicality of the proposed DTZNN model (12) is thus confirmed.

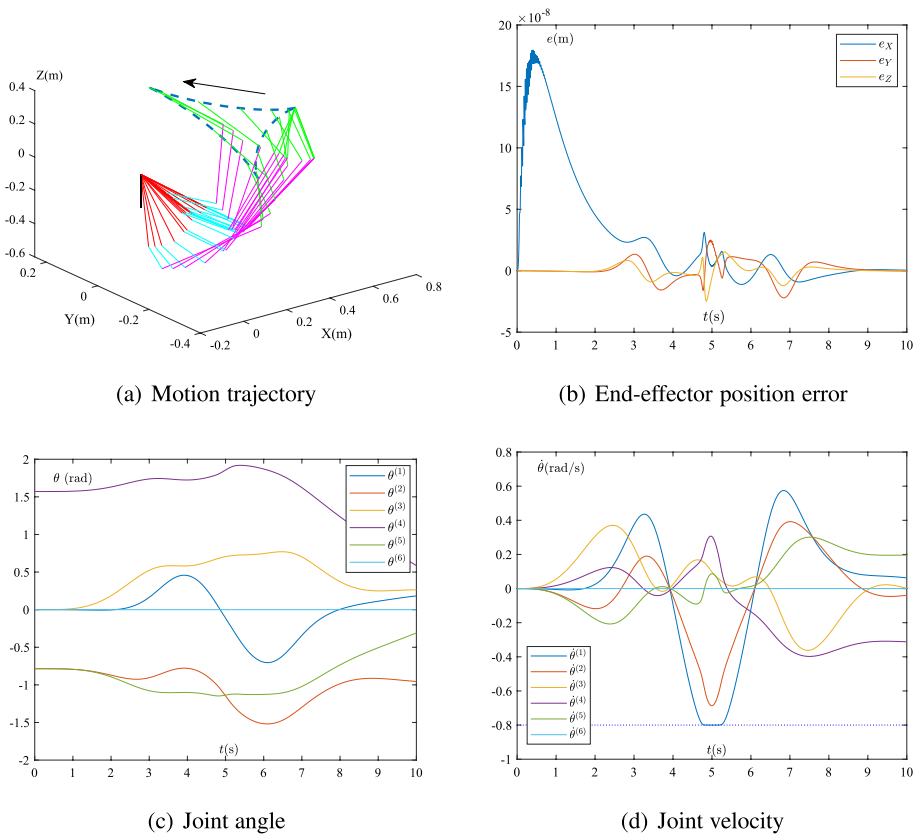
In these simulations, the limits of the PUMA560 robotic arm are defined as

$$\begin{aligned}
 -\dot{\theta}^+ &= \dot{\theta}^- = [-0.8; -0.8; -0.8; -0.8; -0.8; -0.8] \text{ rad/s,} \\
 \theta^+ &= [2.7754; 0.7506; 3.9274; 2.9674; 1.7455; 4.6256] \text{ rad,} \\
 \theta^- &= -[2.7754; 3.8925; 0.7855; 1.9201; 1.7455; 4.6256] \text{ rad.}
 \end{aligned}$$

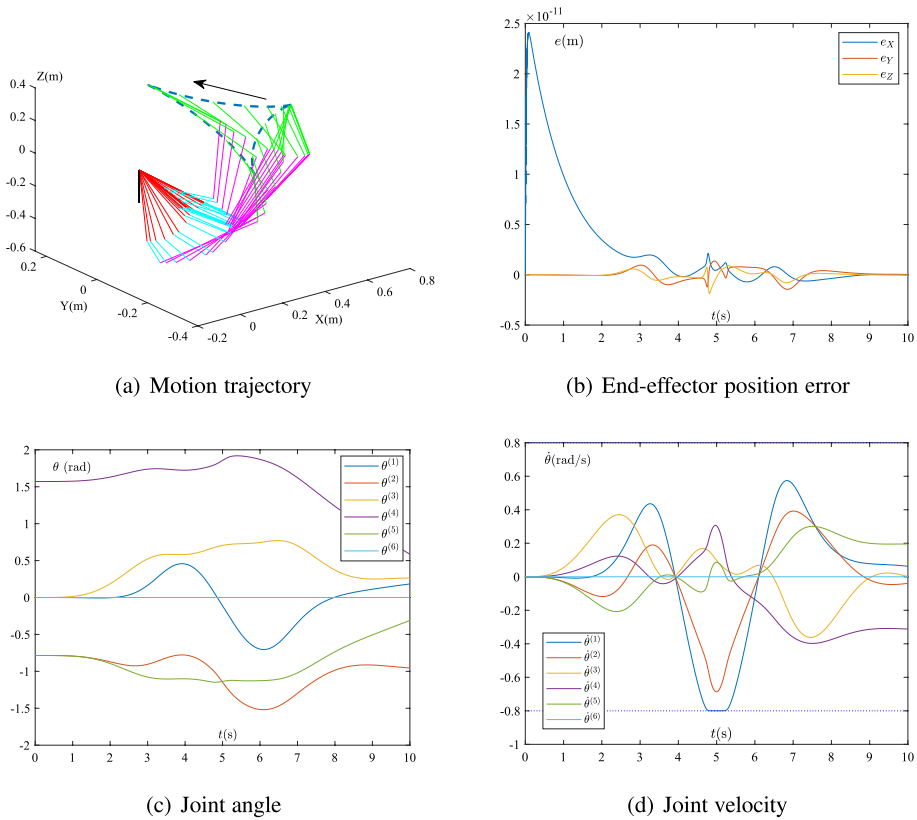
The desired end-effector position vectors for the tricuspid path, circular path, and Rhodonea path are respectively designed as

$$r_p = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} r(2 \cos(\phi) + \cos(2\phi)) - 3r + i_{x_0} \\ r(2 \sin(\phi) - \sin(2\phi)) \cos(\alpha) + i_{y_0} \\ r(2 \sin(\phi) - \sin(2\phi)) \sin(\alpha) + i_{z_0} \end{bmatrix}, \tag{20}$$

$$r_p = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} r \cos(\phi) - r + i_{x_0} \\ r \sin(\phi) \cos(\alpha) + i_{y_0} \\ r \sin(\phi) \sin(\alpha) + i_{z_0} \end{bmatrix}, \tag{21}$$



**Fig. 8** Simulation results of the motion planning scheme (19) with  $\epsilon = 0.01$  and  $\mu = 0.1$  for PUMA560 robotic arm tracking the tricuspid path



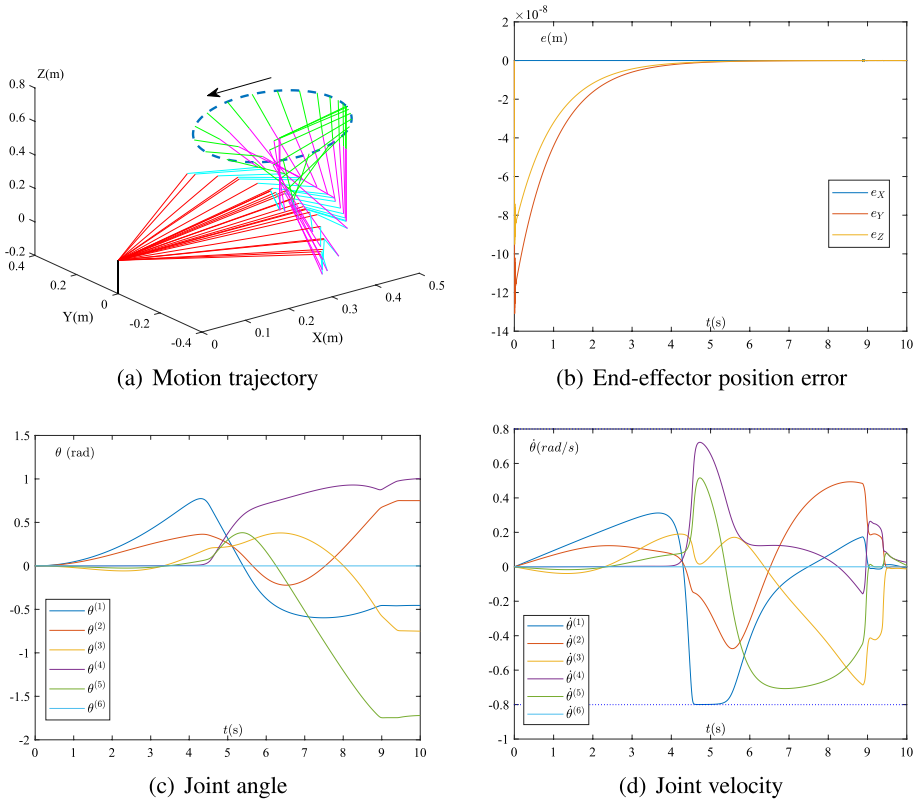
**Fig. 9** Simulation results of the motion planning scheme (19) with  $\epsilon = 0.001$  and  $\mu = 0.1$  for PUMA560 robotic arm tracking the tricuspid path

$$r_p = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} r \cos(2\phi) \cos(\phi) - r + i_{x_0} \\ r \cos(2\phi) \sin(\phi) \cos(\alpha) + i_{y_0} \\ r \cos(2\phi) \sin(\phi) \sin(\alpha) + i_{z_0} \end{bmatrix}, \tag{22}$$

where  $\phi = 2\pi \sin^2(\pi t/2T)$ , time  $t \in [0, T]$ , the parameter  $\alpha$  is constant, and  $r$  is the radius of the desired path. In addition,  $i_{x_0}$ ,  $i_{y_0}$  and  $i_{z_0}$  respectively denote the X-axis, Y-axis and Z-axis components of the initial position vector of the end-effector.

To better show the simulation effect, the PUMA560 end-effector path tracking periods are all set to  $T = 10s$ , and the initial joint configuration is set to  $\theta_0 = [0;0;0;0;0;0]$  rad. The related simulation results are depicted in Figs. 8, 9, 10, 11, where  $t \in \{0, \epsilon, 2\epsilon, \dots, 10\}$ ,  $e = \phi(\theta(t)) - r(t) \in R^3$ , and  $\dot{\theta}^{(i)}$  and  $\theta^{(i)}$  ( $i = 1, 2, \dots, 6$ ) represent the  $i$ th element of  $\dot{\theta}(t)$  and  $\theta(t)$ .

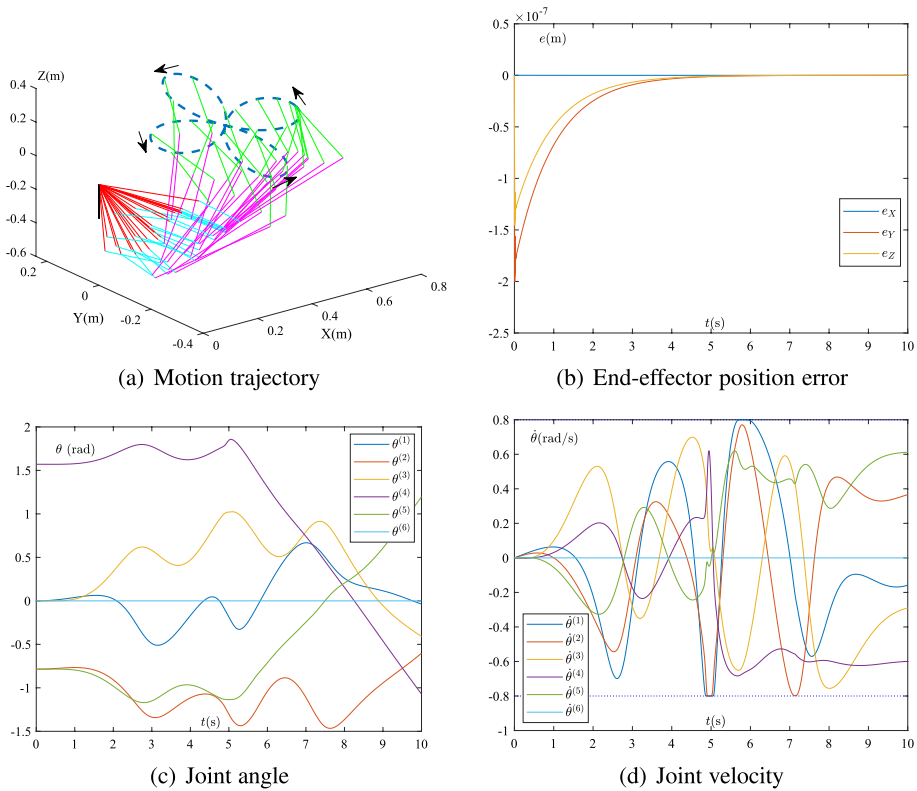
Figure 8 presents the results of the new motion planning scheme (19) with  $\epsilon = 0.01$  and  $\mu = 0.1$  for the PUMA560 robotic arm tracking the tricuspid path. In Figs. 8a and b, it can be observed that the PUMA560 end-effector trajectory closely follows the desired tricuspid path, and the maximum positioning error is  $6.11775 \times 10^{-7}$  m. In particular, Fig. 8c denotes that the value of joint angle  $\theta$  calculated by (19) remains within its limits. In Fig. 8d, the joint velocity  $\dot{\theta}$  stays within the limits even after the lower



**Fig. 10** Simulation results of the motion planning scheme (19) with  $\epsilon = 0.01$  and  $\mu = 0.1$  for PUMA560 robotic arm tracking the circular path

limit is reached. That is,  $\hat{\theta}^- \leq \dot{\theta}(t) \leq \hat{\theta}^+$  and  $\theta^- \leq \theta(t) \leq \theta^+$  are satisfied. These simulation results demonstrate that the new scheme (19) can effectively implement the motion planning of the PUMA560 robotic arm in the presence of joint physical constraints and further verify the practical application of the proposed DTZNN model (12).

To further present the high accuracy characteristics of the new motion planning scheme (19), the simulation is repeated with the sampling interval  $\epsilon$  reduced by a factor of 10 (i.e., from  $\epsilon = 0.01$  to  $\epsilon = 0.001$ ) and the rest of the condition parameters unchanged. The first two subfigures of Fig. 8 show that the PUMA560 end-effector successfully tracks the desired path again, with the maximum error is  $1.34149 \times 10^{-11}$  m. The rest of the subfigures of Figs. 8c and d denote that joint angle  $\theta$  and joint velocity  $\dot{\theta}$  also satisfy  $\hat{\theta}^- \leq \dot{\theta}(t) \leq \hat{\theta}^+$  and  $\theta^- \leq \theta(t) \leq \theta^+$ , respectively, thus demonstrating the effectiveness of the new motion planning scheme (19). Notably, the PUMA560 end-effector positioning error decreases by a factor of 10,000 as the sampling interval  $\epsilon$  decreases, that is,  $(1.34149 \times 10^{-11}) / (6.11775 \times 10^{-7}) \approx 2.19278 \times 10^{-5}$ . This finding underscores the importance of the sampling interval  $\epsilon$  for the new motion planning scheme (19) and reflects an  $O(\epsilon^5)$  error variation pattern. Therefore, in practical applications,  $\epsilon$  in the new motion planning scheme (19) should be set small enough to ensure the high planning accuracy required for robotic arms.



**Fig. 11** Simulation results of the motion planning scheme (19) with  $\epsilon = 0.01$  and  $\mu = 0.1$  for PUMA560 robotic arm tracking the Rhodonea path

For further investigation, the tracking path of the PUMA560 end-effector is set to a circular path and a Rhodonea path for the new motion planning scheme (19) with  $\epsilon = 0.001$  and  $\mu = 0.1$ . Figures 10 and 11 show the corresponding simulation results, which fully demonstrate the validity of the new motion planning scheme (19). In particular, the end-effector tracking trajectory closely follows the desired path, and the corresponding maximum error is of the order of  $10^{-10}$  m or  $10^{-11}$  m. Joint velocity  $\dot{\theta}$  and the joint angle  $\theta$  obtained through the new scheme (19) also remain within their respective limits, ensuring that they satisfy the conditions  $\dot{\theta}^- \leq \dot{\theta}(t) \leq \dot{\theta}^+$  and  $\theta^- \leq \theta(t) \leq \theta^+$ , respectively.

In summary, the simulation results provided in Figs. 8, 9, 10, 11 confirm the validity of the new motion planning scheme (19) for the PUMA560 robotic arm. Furthermore, these results underscore the effectiveness and practicality of the proposed DTZNN model (12).

## 6 Conclusion

In this paper, utilizing the Taylor difference formula (9) to discretize the CTZNN model (5), the new DTZNN model (12) is proposed and studied to address the BCTDLE (1). Theoretical analysis demonstrates that the proposed DTZNN model (12) has convergence property and can generate the exact time-dependent solution of (1). Numerical results indicate the validity and superiority of the proposed DTZNN model (12) and further point to the SSCE variation being of the  $O(\varepsilon^5)$  mode. Such a DTZNN model is finally applied to robotic arms, and the related new motion planning scheme (19) is derived. Simulation results obtained from the PUMA560 robotic arm denote the validity and reliability of the new motion planning scheme (19) for different desired path tracking examples. The applicability of the proposed DTZNN model (12) is confirmed as well.

One future research directions involves utilizing the proposed DTZNN model (12) to solve the BCTDLE (1) in a noisy environment. Another direction is to explore the application of the proposed DTZNN model (12) in other tasks of redundant robot manipulators, such as repetitive motions and obstacle avoidance. As a continuation of this paper, further efforts will focus on designing and selecting different activation functions to enhance the robustness of the proposed DTZNN model (12).

**Acknowledgements** The authors would like to thank the editors and reviewers for their time and effort in evaluating this paper and for the constructive comments for the improvement of its presentation and quality.

**Author contributions** All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Naimeng Cang, Dongsheng Guo and Shan Xue. The first draft of the manuscript was written by Naimeng Cang and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

**Funding** This work is partly supported by Scientific Research Fund of Hainan University under Grant (KYQD(ZR)23025), Shenzhen Science and Technology Program under Grant (JCYJ20230807093513027), the National Science and Technology Major Project (Grant No. 2022ZD0119900), Shanghai Science and Technology program (Grant No. 22015810300), Hainan Province Science and Technology Special Fund (Grant No. ZDYF2024GXJS003), the Hainan Provincial Natural Science Foundation of China (Grant No. 620QN284), National Natural Science Foundation (Grant Nos. 61976096 and 62373157), and National High-Level Talents Special Support Program (Youth Talent of Technological Innovation of Ten-Thousands Talents Program) (Grant No. C7220060).

**Code or data availability** The code or data were available in this manuscript.

## Declarations

**Conflict of interest** The authors have no relevant financial or non-financial interests to disclose

**Ethical approval** This paper does not contain any studies with human participants or animals performed by any of the authors.

**Consent to participate** Informed consent was obtained from all individual participants included in the study.

**Consent for publication** The authors affirm that human research participants provided informed consent for publication of the images in all figures.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article

are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Abdelmalek NN (1977) Minimum  $L_\infty$  solution of underdetermined systems of linear equations. *J Approx Theory* 20(1):57–69
- Cichocki A, Ramirez-Angulo J, Unbehauen R (1992) Architectures for analog VLSI implementation of neural networks for solving linear equations with inequality constraints. *Proc IEEE Int Symp Circuits Syst* 3:1529–1532
- Donoho DL, Tsai Y, Drori I, Starck JL (2012) Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *IEEE Trans Inf Theory* 58(2):1094–1121
- Griffiths DF, Higham DJ (2010) Numerical methods for ordinary differential equations initial value problems. Springer, London
- Guo D, Zhang Y (2012) Zhang neural network, Getz-Marsden dynamic system, and discrete-time algorithms for time-varying matrix inversion with application to robots' kinematic control. *Neurocomputing* 97:22–32
- Guo D, Nie Z, Yan L (2017) Novel discrete-time Zhang neural network for time-varying matrix inversion. *IEEE Trans Syst Man Cybern: Syst* 47(8):2301–2310
- Guo D, Li K, Liao B (2018) Bi-criteria minimization with MWVN-INAM type for motion planning and control of redundant robot manipulators. *Robotica* 36(5):655–675
- Guo D, Xu F, Li Z, Nie Z, Shao H (2018) Design, verification, and application of new discrete-time recurrent neural network for dynamic nonlinear equations solving. *IEEE Trans Ind Informat* 14(9):3936–3945
- Huang S, Ma Z, Yu S, Han Y (2022) New discrete-time zeroing neural network for solving time-variant underdetermined nonlinear systems under bound constraint. *Neurocomputing* 487:214–227
- Kumar R (2022) A Lyapunov-stability-based context-layered recurrent pi-sigma neural network for the identification of nonlinear systems. *Appl Soft Comput* 122:108836
- Kumar R (2022) Memory recurrent Elman neural network-based identification of time-delayed nonlinear dynamical system. *IEEE Trans Syst Man Cybern: Syst* 53(2):753–762
- Kumar R (2023) Double internal loop higher-order recurrent neural network-based adaptive control of the nonlinear dynamical system. *Soft Comput* 27:17313–17331
- Li S, Jin L, Mirza MA (2019) Kinematic control of redundant robot arms using neural networks. Wiley, Hoboken
- Li W, Xiao L, Liao B (2019) A finite-time convergent and noise-rejection recurrent neural network and its discretization for dynamic nonlinear equations solving. *IEEE Trans Cybern* 50(7):3195–3207
- Liang XB, Tso SK (2002) Improved upper bound on step-size parameters of discrete-time recurrent neural networks for linear inequality and equation system. *IEEE Trans Circuits Syst I Fundam Theory Appl* 49(5):695–698
- Liao B, Han L, Cao X, Li S, Li J (2024) Double integral-enhanced Zeroing neural network with linear noise rejection for time-varying matrix inverse. *CAAI Trans Intell Technol* 9(1):197–210
- Lu H, Jin L, Luo X, Liao B, Guo D, Xiao L (2019) RNN for solving perturbed time-varying underdetermined linear system with double bound limits on residual errors and state variables. *IEEE Trans Indus Inf* 15(11):5931–5942
- Ma Z, Guo D (2021) Discrete-time recurrent neural network for solving bound-constrained time-varying underdetermined linear system. *IEEE Trans Ind Informat* 17(6):3869–3878
- Mathews JH, Fink KD (2004) Numerical methods using MATLAB, 4th edn. Prentice-Hall, Upper Saddle River
- Morigi S, Scallari F (2001) A regularizing L-curve Lanczos method for underdetermined linear systems. *Appl Math Comput* 121(1):55–73
- Neta B, Chun C, Scott M (2014) Basins of attraction for optimal eighth order methods to find simple roots of nonlinear equations. *Appl Math Comput* 227:567–592
- Park K, Chang P, Kim S (1998) The enhanced compact QP method for redundant manipulators using practical inequality constraints. *Proc IEEE Int Conf Robot Autom* 1:107–114
- Sharma JR (2005) A composite third order Newton–Steffensen method for solving nonlinear equations. *Appl Math Comput* 169(1):242–246

- Spedicato E, Xia Z, Zhang L (2000) ABS algorithms for linear equations and optimization. *J Comput Appl Math* 124(1/2):155–170
- Sun M, Liu J (2020) A novel noise-tolerant Zhang neural network for time-varying Lyapunov equation. *Adv Diff Equ* 1:1–15
- Tsiligianni E, Kondi LP, Katsaggelos AK (2015) Preconditioning for underdetermined linear systems with sparse solutions. *IEEE Signal Process Lett* 22(9):1239–1243
- Xia Y, Wang J, Hung DL (1999) Recurrent neural networks for solving linear inequalities and equations. *IEEE Trans Circuits Syst I: Fundam Theory Appl* 46(4):452–462
- Xiao L, Dai J, Lu R, Li S, Li J, Wang S (2020) Design and comprehensive analysis of a noise-tolerant ZNN model with limited-time convergence for time-dependent nonlinear minimization. *IEEE Trans Neural Netw Learn Syst* 31(12):5339–5348
- Xu F, Li Z, Nie Z, Shao H, Guo D (2019) New recurrent neural network for online solution of time-dependent underdetermined linear system with bound constraint. *IEEE Trans Ind Informat* 15(4):2167–2176
- Xu F, Li Z, Nie Z, Shao H, Guo D (2019) Zeroing neural network for solving time-varying linear equation and inequality systems. *IEEE Trans Neural Netw Learn Syst* 30(8):2346–2357
- Zeng J, Lin S, Xu Z (2014) Sparse solution of underdetermined linear equations via adaptively iterative thresholding. *Signal Process* 97:152–161
- Zhang Y, Guo D (2015) Zhang functions and various models. Springer, Germany
- Zhang Y, Jin L (2017) Robot manipulator redundancy resolution. Wiley, Hoboken
- Zhang Y, Yi C (2011) Zhang neural networks and neural-dynamic method. Nova, New York
- Zhang Y, Ge SS, Lee TH (2004) A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators. *IEEE Trans on Syst Man Cybern* 34(5):2126–2132
- Zhang Y, Li W, Guo D, Ke Z (2013) Different Zhang functions leading to different ZNN models illustrated via time-varying matrix square roots finding. *Expert Syst Appl* 40(11):4393–4403
- Zhang Y, Wang Y, Jin L (2013) Different ZFs leading to various ZNN models illustrated via online solution of time-varying underdetermined systems of linear equations with robotic application. *Lect Notes Comput Sci* 7952:481–488
- Zhang Z, Zheng L, Li L, Deng X, Xiao L, Huang G (2018) A new finite-time varying-parameter convergent-differential neural-network for solving nonlinear and nonconvex optimization problems. *Neurocomputing* 319:74–83
- Zhang Y, Li S, Gui J, Luo X (2018) Velocity-level control with compliance to acceleration-level constraints: a novel scheme for manipulator redundancy resolution. *IEEE Trans Ind Informat* 14(3):921–930
- Zhang Y, Yang M, Huang H, Xiao M, Hu H (2019) New discrete solution model for solving future different-level linear inequality and equality with robot manipulator control. *IEEE Trans Ind Informat* 15(4):1975–1984
- Zhao YB (2013) New and improved conditions for uniqueness of sparsest solutions of underdetermined linear systems. *Appl Math Comput* 224:58–73
- Zhang Y, Shi Y, Xiao L, Mu B (2012) Convergence and stability results of Zhang neural network solving systems of time-varying nonlinear equations. *Proc 8th Int Conf Natural Comput* 143–147
- Zhang Y, Qiu H, Peng C, Shi Y, Tan H (2015) Simply and effectively proved square characteristics of discrete-time zd solving systems of time-varying nonlinear equations. *Proc IEEE Int Conf Robot Autom* 1457–1462
- Cai J, Feng Q, Guo D (2021) New discrete-time zeroing neural network for solving time-varying system of linear equation and inequality. *Proc 33rd Chinese Control Decision Conf* 3806–3811

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.