



OPEN ACCESS

EDITED BY

Mei Liu,
Multi-scale Medical Robotics Center Limited,
China

REVIEWED BY

Bolin Liao,
Jishou University, China
Dongsheng Guo,
Hainan University, China
Cheng Hua,
Jishou University, China
Fujie Wang,
Dongguan University of Technology, China
Chenfu Yi,
Guangdong Polytechnic Normal University,
China
Yuting Xue,
Tongji University, China
Tinglei Wang,
Jishou University, China
Zhuosong Fu,
Jishou University, China

*CORRESPONDENCE

Zhisheng Ma
✉ mazhisheng0001@126.com

RECEIVED 14 February 2025

ACCEPTED 18 June 2025

PUBLISHED 11 July 2025

CITATION

Ma Z and Huang S (2025) Design and analysis of combined discrete-time zeroing neural network for solving time-varying nonlinear equation with robot application. *Front. Neurobot.* 19:1576473. doi: 10.3389/fnbot.2025.1576473

COPYRIGHT

© 2025 Ma and Huang. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Design and analysis of combined discrete-time zeroing neural network for solving time-varying nonlinear equation with robot application

Zhisheng Ma* and Shaobin Huang

College of Computer Science and Technology, Harbin Engineering University, Harbin, Heilongjiang, China

Zeroing neural network (ZNN) is viewed as an effective solution to time-varying nonlinear equation (TVNE). In this paper, a further study is shown by proposing a novel combined discrete-time ZNN (CDTZNN) model for solving TVNE. Specifically, a new difference formula, which is called the Taylor difference formula, is constructed for first-order derivative approximation by following Taylor series expansion. The Taylor difference formula is then used to discretize the continuous-time ZNN model in the previous study. The corresponding DTZNN model is obtained, where the direct Jacobian matrix inversion is required (being time consuming). Another DTZNN model for computing the inverse of Jacobian matrix is established to solve the aforementioned limitation. The novel CDTZNN model for solving the TVNE is thus developed by combining the two models. Theoretical analysis and numerical results demonstrate the efficacy of the proposed CDTZNN model. The CDTZNN applicability is further indicated by applying the proposed model to the motion planning of robot manipulators.

KEYWORDS

discrete-time zeroing neural network, time-varying nonlinear equation, Taylor difference formula, matrix inversion, robot manipulators

1 Introduction

Nonlinear equation (NE) has an important implication in many fields of science and engineering, such as signal and image processing, pattern recognition, and robot motion planning. Many solutions for the NE have been reported (Liao et al., 2020; Wu et al., 2021; Zeng et al., 2024; Mathews and Fink, 2004; Ramos and Monteiro, 2017; Sharma et al., 2017; Zhang and Yi, 2011; Xiao and Lu, 2015; Zhang et al., 2017). The Newton method, which possesses the quadratic convergence, was the classic iteration method to find the solution of the NE (Mathews and Fink, 2004). On the basis of Newton method, a new approach was developed by Ramos and Monteiro (2017) to solve the NE. A class of multipoint methods for solving the NE was presented by Sharma et al. (2017). Different from these iteration methods, the methods based on recurrent neural network (RNN) were designed to solve the NE. The gradient neural network (GNN), which possesses the exponential convergence, was the classic RNN model for determining the roots of the NE (Zhang and Yi, 2011). As shown by Xiao and Lu (2015), the GNN was accelerated to finite convergence through a special activation function. Another typical RNN model termed the zeroing neural network (ZNN) was provided by Zhang et al. (2017), and it was indicated to have a connection to Davidenko method in solving the NE. Notably, the abovementioned studies were reported on computing the static NE.

Focusing on solving the time-varying NE (TVNE), a continuous-time ZNN (CTZNN) model was developed by Zhang et al. (2012a), of which the effectiveness was substantiated by theoretical analysis and simulations. The nonlinearly-activated CTZNN model with finite-time convergence was designed by Xiao (2016) to solve the TVNE. Considering the existence of noise in the process of solving the TVNE, Jin et al. (2017b) presented an integration-enhanced CTZNN model, Li et al. (2020) provided a finite-time convergent and noise-rejection CTZNN model, and Dai et al. (2024) developed a norm-based CTZNN model with strong robustness. For possible digital hardware implementation, the discrete-time ZNN (DTZNN) model was investigated by Zhang et al. (2015) using the Euler difference formula to discretize the original CTZNN model. The DTZNN model was theoretically analyzed to have a square characteristic. That is, the steady-state computational error (SSCE) was expressed in the form of $O(\sigma^2)$, where σ denotes the sampling gap. Furthermore, the Broyden method was utilized to avoid the matrix inversion involved in the DTZNN model, and the modified DTZNN model was presented by Zhang et al. (2012b). This model was immune to singularity and still had the square characteristic.

Being different from the Euler difference formula, various Taylor-type difference formulas have been constructed (Liao et al., 2016; Guo et al., 2018; Xiang et al., 2019; Tang and Zhang, 2023; Cang et al., 2024; Chen et al., 2024; Xiang et al., 2025; Guo et al., 2025). For example, the Taylor difference formula with the $O(\sigma^2)$ truncation error was presented in Liao et al. (2016), while the one with $O(\sigma^3)$ truncation error was provided in Tang and Zhang (2023). These difference formula have been proven to be effective on the ZNN discretization. Specific for the TVNE solving, the Taylor difference formula in Guo et al. (2018) was used to discretize the CTZNN model (Zhang et al., 2012a). The resultant DTZNN model was developed, of which the SSCE is expressed in the form of $O(\sigma^4)$ (Guo et al., 2018). In Cang et al. (2024), the DTZNN model with the $O(\sigma^5)$ SSCE mode was designed to solve the TVNE. Notably, the shortcoming of such two DTZNN models is the requirement of calculating the inverse of Jacobian matrix, which could be (very) time consuming. In this sense, the DTZNN models in Guo et al. (2018) and Cang et al. (2024) may be less effective for the online solution of TVNE. Thus, an effective DTZNN model with great computational performance but without requirement of direct Jacobian matrix inversion should be designed and studied.

In this paper, motivated by the inspiring studies (Zhang et al., 2015, 2012b; Guo et al., 2018; Cang et al., 2024), a further investigation is provided by proposing a novel combined DTZNN (CDTZNN) model to solve the TVNE. Specifically, for the possible discretization of a continuous-time model, a new difference formula, which is called the Taylor difference formula, is developed by following Taylor series expansion (Mathews and Fink, 2004). Using the new formula to discretize the original CTZNN model (Zhang et al., 2012a) yields the resultant DTZNN model. On the basis of Guo et al. (2017), another DTZNN model is established for the Jacobian matrix inversion. With the combination of two models, the novel CDTZNN model for solving the TVNE is thus developed. Such a model does not require calculating the inverse of Jacobian matrix and can thus reduce the amount of calculation.

Theoretical analysis and numerical results substantiate the efficacy of the proposed CDTZNN model.

In recent years, robot manipulators have attracted considerable attention in engineering applications (Li et al., 2020; Guo et al., 2018; Cang et al., 2024; Guo et al., 2017; Jin et al., 2017a, 2024; Yan et al., 2024; Zhang et al., 2025; Li et al., 2019; Xiao et al., 2019). The motion planning of robot manipulators, as a fundamental issue, has been widely investigated. Mathematically, the purpose of motion planning can be achieved when the corresponding TVNE is solved (Li et al., 2020; Guo et al., 2018; Cang et al., 2024). In consideration of this point, the proposed CDTZNN model is further studied to the motion planning of robot manipulators through solving the TVNE. On the basis of a DOBOT manipulator (Xiao et al., 2019), the simulation results are provided to validate the applicability of the proposed CDTZNN model.

The rest of the paper consists of five parts. Section 2 presents the previous studies on TVNE solving. Section 3 shows the Taylor difference formula. Section 4 describes the CDTZNN model and analyzes its computational performance. Section 5 provides the numerical results of the proposed CDTZNN model and applies such a model to robot manipulators. Section 6 concludes the study. The main contributions are listed as below.

- 1) A new Taylor difference formula is developed via the Taylor series expansion. Such a formula differs from the formulas used in Zhang et al. (2015, 2012b); Guo et al. (2018); Cang et al. (2024) for the original CTZNN discretization on solving the TVNE.
- 2) On the basis of the new Taylor difference formula, a novel CDTZNN model is proposed and studied to solve the TVNE. To the best of the authors' knowledge, such a CDTZNN model has not been reported yet.
- 3) The computational performance of the proposed CDTZNN model is analyzed theoretically. The CDTZNN efficacy is further substantiated by numerical results. The results also express the $O(\sigma^4)$ mode in the SSCE of the proposed model.
- 4) The proposed CDTZNN model is effectively applied to the motion planning of robot manipulators. The simulations under the DOBOT manipulator show the potential application of CDTZNN.

2 Previous studies on TVNE solving

This section shows the problem formulation of solving the TVNE. Then, the CTZNN and DTZNN models in the previous studies are presented.

2.1 Problem statement

In this study, the following TVNE is considered (Zhang et al., 2012a):

$$\mathbf{f}(\mathbf{x}^*(t), t) = \mathbf{0} \in \mathbb{R}^n, \quad \forall t \in [0, +\infty), \quad (1)$$

where $\mathbf{f}(\cdot): \mathbb{R}^n \rightarrow \mathbb{R}^n$ denotes a differentiable nonlinear mapping and $\mathbf{x}^*(t) \in \mathbb{R}^n$ denotes the theoretical solution of Equation 1.

This study aims to find the numerical solution $\mathbf{x}(t) \in \mathbb{R}^n$ that holds the TVNE (Equation 1) true; specifically, as time t evolves, $\mathbf{x}(t) \rightarrow \mathbf{x}^*(t)$.

2.2 CTZNN model

To effectively solve the TVNE (Equation 1), the following error function must converge to zero as time evolves:

$$\mathbf{e}(t) = \mathbf{f}(\mathbf{x}(t), t) \in \mathbb{R}^n. \quad (2)$$

In Zhang et al. (2012a), the decay formula $\dot{\mathbf{e}}(t) = -\gamma \mathbf{e}(t)$ is used, and the CTZNN model for solving the TVNE (Equation 1) is designed as follows:

$$\dot{\mathbf{x}}(t) = -J^-(\mathbf{x}(t), t)(\mathbf{f}_t(\mathbf{x}(t), t) + \gamma \mathbf{f}(\mathbf{x}(t), t)), \quad (3)$$

where $\dot{\mathbf{e}}(t)$ and $\dot{\mathbf{x}}(t)$ correspond to the time derivatives of $\mathbf{e}(t)$ and $\mathbf{x}(t)$, $J^-(\mathbf{x}(t), t) \in \mathbb{R}^{n \times n}$ denotes the inverse of Jacobian matrix $J(\mathbf{x}(t), t) = \partial \mathbf{f} / \partial \mathbf{x}$, $\mathbf{f}_t(\mathbf{x}(t), t) = \partial \mathbf{f} / \partial t \in \mathbb{R}^n$ denotes the system time-derivative vector, and $\gamma > 0 \in \mathbb{R}$ denotes the parameter for scaling the CTZNN convergence rate.

Lemma 1 (Zhang et al., 2012a). *When a solvable TVNE (Equation 1) is considered, the state vector $\mathbf{x}(t)$ of the CTZNN model (Equation 3), starting from an initial state $\mathbf{x}(0)$ close sufficiently to $\mathbf{x}^*(0)$, converges to the theoretical solution $\mathbf{x}^*(t)$ of (Equation 1). That is, $\mathbf{x}(t) \rightarrow \mathbf{x}^*(t)$ as t evolves.*

2.3 DTZNN models

To discretize the CTZNN model (Equation 3), the following Euler difference formula (Mathews and Fink, 2004) can be used:

$$\dot{\mathbf{x}}_k = \dot{\mathbf{x}}(t_k = k\sigma) = \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\sigma} + \mathbf{O}(\sigma), \quad (4)$$

where $\mathbf{x}_k = \mathbf{x}(t_k = k\sigma)$ with the subscript $k \in N$ and sampling gap $\sigma \in (0, 1)$, and $\mathbf{O}(\sigma) \in \mathbb{R}^n$ denotes the truncation error vector with every element as $O(\sigma)$. By using Equation 4 to discretize Equation 3, the Euler DTZNN model for solving the TVNE (Equation 1) is formulated as follows (Zhang et al., 2015):

$$\mathbf{x}_{k+1} = \mathbf{x}_k - J^-(\mathbf{x}_k, t_k)(\sigma \mathbf{f}_t(\mathbf{x}_k, t_k) + h \mathbf{f}(\mathbf{x}_k, t_k)), \quad (5)$$

where $h = \gamma\sigma \in \mathbb{R}$ denotes the step size with γ defined as in Equation 3.

To avoid the calculation for the inverse of Jacobian matrix, the Euler DTZNN model (Equation 5) is improved in Zhang et al. (2012b) by using Broyden method (Broyden, 1965). Specifically, $J^-(\mathbf{x}_k, t_k)$ is calculated and approximated through a matrix $M_k = M(t_k = k\sigma) \in \mathbb{R}^{n \times n}$, where the recursion for updating M_k is given by

$$M_k = M_{k-1} + \frac{(\mathbf{u}_{k-1} - M_{k-1} \mathbf{v}_{k-1}) \mathbf{u}_{k-1}^T M_{k-1}}{\mathbf{u}_{k-1}^T M_{k-1} \mathbf{v}_{k-1}}, \quad (6)$$

with $\mathbf{u}_{k-1} = \mathbf{x}_k - \mathbf{x}_{k-1} \in \mathbb{R}^n$ and $\mathbf{v}_{k-1} = \mathbf{f}(\mathbf{x}_k, t_k) - \mathbf{f}(\mathbf{x}_{k-1}, t_k) \in \mathbb{R}^n$. On the basis of Equation 6, replacing $J^-(\mathbf{x}_k, t_k)$ by M_k in Equation 5 yields the modified DTZNN model as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - M_k(\sigma \mathbf{f}_t(\mathbf{x}_k, t_k) + h \mathbf{f}(\mathbf{x}_k, t_k)). \quad (7)$$

For further discussion, on the basis of the error function $\mathbf{e}(t)$ in Equation 2, the SSCE is defined as $\lim_{k \rightarrow \infty} \|\mathbf{e}_k\|_2 = \lim_{k \rightarrow \infty} \|\mathbf{f}(\mathbf{x}_k, t_k)\|_2$, where $\|\cdot\|_2$ returns the Euclidean norm of a vector. Then, the computational performances of the DTZNN models (Equations 5, 7), are shown by the following lemma.

Lemma 2 (Zhang et al., 2015, 2012b). *When a solvable TVNE (Equation 1) is considered, the SSCE of each of the DTZNN models (Equations 5, 7) is of $O(\sigma^2)$.*

Being different from the Euler difference formula, the following Taylor difference formulas are constructed (Tang and Zhang, 2023; Guo et al., 2017):

$$\dot{\mathbf{x}}_k = \frac{8\mathbf{x}_{k+1} + \mathbf{x}_k - 6\mathbf{x}_{k-1} - 5\mathbf{x}_{k-2} + 2\mathbf{x}_{k-3}}{18\sigma} + \mathbf{O}(\sigma^3), \quad (8)$$

$$\dot{\mathbf{x}}_k = \frac{24\mathbf{x}_{k+1} - 5\mathbf{x}_k - 12\mathbf{x}_{k-1} - 6\mathbf{x}_{k-2} - 4\mathbf{x}_{k-3} + 3\mathbf{x}_{k-4}}{48\sigma} + \mathbf{O}(\sigma^3), \quad (9)$$

where sampling gap $\sigma \in (0, 1)$, and $\mathbf{O}(\sigma^3) \in \mathbb{R}^n$ denotes the truncation error vector with every element as $O(\sigma^3)$. In Cang et al. (2024), another Taylor difference formula is expressed as follows:

$$\dot{\mathbf{x}}_k = \frac{110\mathbf{x}_{k+1} + 42\mathbf{x}_k - 89\mathbf{x}_{k-1} - 102\mathbf{x}_{k-2} + 12\mathbf{x}_{k-3} + 44\mathbf{x}_{k-4} - 17\mathbf{x}_{k-5}}{276\sigma} + \mathbf{O}(\sigma^4), \quad (10)$$

where $\mathbf{O}(\sigma^4) \in \mathbb{R}^n$ denotes the truncation error vector with every element as $O(\sigma^4)$. By using Equation 8–10 to discretize the CTZNN model (Equation 3), the following three Taylor DTZNN models for solving Equation 1 are obtained:

$$\begin{aligned} \mathbf{x}_{k+1} = & -\frac{1}{8}\mathbf{x}_k + \frac{3}{4}\mathbf{x}_{k-1} + \frac{5}{8}\mathbf{x}_{k-2} - \frac{1}{4}\mathbf{x}_{k-3} \\ & - J^-(\mathbf{x}_k, t_k) \left(\frac{9}{4}\sigma \mathbf{f}_t(\mathbf{x}_k, t_k) + h \mathbf{f}(\mathbf{x}_k, t_k) \right), \end{aligned} \quad (11)$$

$$\begin{aligned} \mathbf{x}_{k+1} = & \frac{5}{24}\mathbf{x}_k + \frac{1}{2}\mathbf{x}_{k-1} + \frac{1}{4}\mathbf{x}_{k-2} + \frac{1}{6}\mathbf{x}_{k-3} - \frac{1}{8}\mathbf{x}_{k-4} \\ & - J^-(\mathbf{x}_k, t_k) (2\sigma \mathbf{f}_t(\mathbf{x}_k, t_k) + h \mathbf{f}(\mathbf{x}_k, t_k)), \end{aligned} \quad (12)$$

$$\begin{aligned} \mathbf{x}_{k+1} = & -\frac{21}{55}\mathbf{x}_k + \frac{89}{110}\mathbf{x}_{k-1} + \frac{51}{55}\mathbf{x}_{k-2} - \frac{6}{55}\mathbf{x}_{k-3} - \frac{2}{5}\mathbf{x}_{k-4} \\ & + \frac{17}{110}\mathbf{x}_{k-5} - J^-(\mathbf{x}_k, t_k) \left(\frac{138}{55}\sigma \mathbf{f}_t(\mathbf{x}_k, t_k) + h \mathbf{f}(\mathbf{x}_k, t_k) \right), \end{aligned} \quad (13)$$

where step size h are given by $h = 9\gamma\sigma/4 \in \mathbb{R}$, $h = 2\gamma\sigma \in \mathbb{R}$, and $h = 138\gamma\sigma/55 \in \mathbb{R}$, respectively. Hereafter, Equation 11, 12, 13 are termed the Taylor DTZNN-I, DTZNN-II, and DTZNN-III models, respectively. Besides, the computational performance of such three models on solving the TVNE (Equation 1) is presented by the following lemma.

Lemma 3 (Guo et al., 2018; Cang et al., 2024). *When a solvable TVNE (Equation 1) is considered, the SSCE of the Taylor DTZNN models (Equations 11, 12, 13) are of $O(\sigma^4)$, $O(\sigma^4)$, and $O(\sigma^5)$, respectively.*

According to Lemmas 2 and 3, the Taylor DTZNN models (Equations 11, 12, 13) can possess better computational performance than the DTZNN models (Equations 5, 7) in solving the TVNE (Equation 1) [in terms of $O(\sigma^4)$ and $O(\sigma^5)$ vs. $O(\sigma^2)$]. However, the Taylor DTZNN models (Equation 11, 12, 13) requires calculating $J^-(\mathbf{x}_k, t_k)$ at each iteration, which could be very time consuming. Thus, a novel DTZNN model that does not require the direct Jacobian matrix inversion is proposed in this work to solve the TVNE (Equation 1).

3 New Taylor difference formula

In this section, a new difference formula termed the Taylor difference formula is constructed to further investigate on discretizing a continuous-time model.

3.1 Scalar-valued form

The new Taylor difference formula depicted in a scalar-valued form is presented via the following theorem.

Theorem 1. *For effectively approximating the first-order derivative of the function $\phi(\cdot)$, the new Taylor difference formula is expressed as follows:*

$$\dot{\phi}(t_k) = \frac{36\phi(t_{k+1}) - \phi(t_k) - 18\phi(t_{k-1}) - 27\phi(t_{k-2}) + 10\phi(t_{k-3})}{78\sigma} + O(\sigma^3), \quad (14)$$

with the subscript $k = 3, 4, 5, \dots$ and sampling gap $\sigma \in (0, 1)$.

Proof. Assuming that $\phi \in C^4[a, b]$ and that $t_{k+1}, t_k, t_{k-1}, t_{k-2}$, and $t_{k-3} \in [a, b]$. By following Taylor series expansion Mathews and Fink (2004), the expressions are obtained as follows:

$$\phi(t_{k+1}) = \phi(t_k) + \sigma\dot{\phi}(t_k) + \frac{\sigma^2}{2!}\ddot{\phi}(t_k) + \frac{\sigma^3}{3!}\dddot{\phi}(t_k) + O(\sigma^4), \quad (15)$$

$$\phi(t_{k-1}) = \phi(t_k) - \sigma\dot{\phi}(t_k) + \frac{\sigma^2}{2!}\ddot{\phi}(t_k) - \frac{\sigma^3}{3!}\dddot{\phi}(t_k) + O(\sigma^4), \quad (16)$$

$$\phi(t_{k-2}) = \phi(t_k) - 2\sigma\dot{\phi}(t_k) + \frac{4\sigma^2}{2!}\ddot{\phi}(t_k) - \frac{8\sigma^3}{3!}\dddot{\phi}(t_k) + O(\sigma^4), \quad (17)$$

$$\phi(t_{k-3}) = \phi(t_k) - 3\sigma\dot{\phi}(t_k) + \frac{9\sigma^2}{2!}\ddot{\phi}(t_k) - \frac{27\sigma^3}{3!}\dddot{\phi}(t_k) + O(\sigma^4), \quad (18)$$

where the symbol ! denotes the factorial operator, and $\dot{\phi}(t_k)$, $\ddot{\phi}(t_k)$, and $\dddot{\phi}(t_k)$ denote the first-, second-, and third-order derivatives of $\phi(t)$ at time t_k , respectively.

Using “ $36 \times (15) - 18 \times (16) - 27 \times (17) + 10 \times (18)$ ” yields the following expression:

$$36\phi(t_{k+1}) - 18\phi(t_{k-1}) - 27\phi(t_{k-2}) + 10\phi(t_{k-3}) = \phi(t_k) + 78\sigma\dot{\phi}(t_k) + O(\sigma^4),$$

of which the reformulation is given by

$$\dot{\phi}(t_k) = \frac{36\phi(t_{k+1}) - \phi(t_k) - 18\phi(t_{k-1}) - 27\phi(t_{k-2}) + 10\phi(t_{k-3})}{78\sigma} + O(\sigma^3). \quad (19)$$

The expression (Equation 19) is exactly the Taylor difference formula (Equation 14). The proof is thus completed. ■

Theorem 1 shows that the new Taylor difference formula (Equation 14) is constructed for first-order derivative approximation, and it further indicates that (Equation 14) has a truncation error of $O(\sigma^3)$.

Remark 1: On the one hand, the new Taylor difference formula (Equation 14) has a smaller truncation error than the Euler difference formula (Equation 4), that is, $O(\sigma^3)$ vs. $O(\sigma)$. Then, the DTZNN model derived by Equation 14 can exhibit a better computational performance than the DTZNN model derived by Equation 4 in solving the TVNE (Equation 1) under the same condition (which can be found in Section 5). On the other hand, though the mathematical derivation is similar (i.e., via Taylor series expansion), the new Taylor difference formula (Equation 14) has a different structure from those formulas (Equation 8, 9, 10), thereby possessing different truncation errors. By summarizing the Taylor difference formulas (Equation 8, 9, 10, 14), the link between Taylor series expansion and numerical differentiation can be obtained. The corresponding DTZNN models with great computational performances are established to solve the TVNE (Equation 1) by utilizing such difference formulas (see Sections 2 and 4). This finding can offer the possibility of developing more DTZNN models to solve different time-varying mathematical problems (Zeng et al., 2024; Zhang and Yi, 2011).

3.2 Vector- and matrix-valued forms

Many continuous-time models are depicted in vector- or matrix-valued forms. Thus, extending the new Taylor difference formula (Equation 14) in Theorem 1 to the vector- and matrix-valued forms is necessary.

Specifically, with regard to the approximation of a vector $\dot{\mathbf{x}}_k = \dot{\mathbf{x}}(t_k = k\sigma) \in \mathbb{R}^n$, (Equation 14) is reformulated as follows:

$$\dot{\mathbf{x}}_k = \frac{36\mathbf{x}_{k+1} - \mathbf{x}_k - 18\mathbf{x}_{k-1} - 27\mathbf{x}_{k-2} + 10\mathbf{x}_{k-3}}{78\sigma} + \mathbf{O}(\sigma^3), \quad (20)$$

where $\mathbf{O}(\sigma^3) \in \mathbb{R}^n$ denotes the vector with every element as $O(\sigma^3)$. With regard to the approximation of a matrix $\dot{M}_k = \dot{M}(t_k = k\sigma) \in \mathbb{R}^{n \times n}$, Equation 14 is reformulated as follows:

$$\dot{M}_k = \frac{36M_{k+1} - M_k - 18M_{k-1} - 27M_{k-2} + 10M_{k-3}}{78\sigma} + \mathbf{O}(\sigma^3), \quad (21)$$

where $\mathbf{O}(\sigma^3) \in \mathbb{R}^{n \times n}$ denotes the matrix with every element as $O(\sigma^3)$.

Evidently, each of the difference formulas (Equations 20, 21) has the truncation error of $\mathbf{O}(\sigma^3)$ and is better than the conventional difference formulas (Mathews and Fink, 2004).

Therefore, on the one hand, Equation 20 is used to discretize the CTZNN model (Equation 3) for solving the TVNE (Equation 1). On the other hand, by following Guo et al. (2017), Equation 21 is used to establish another DTZNN model for Jacobian matrix inversion. Based on the combination of such two models, the novel CDTZNN model is developed for solving (Equation 1) in this study.

4 Novel CDTZNN model

This section proposes the novel CDTZNN model to solve the TVNE (Equation 1) and analyzes the proposed CDTZNN model theoretically.

4.1 Model formulation

Discretizing the CTZNN model (Equation 3) via the difference formula (Equation 20) leads to the following expression:

$$\mathbf{x}_{k+1} = \frac{1}{36}\mathbf{x}_k + \frac{1}{2}\mathbf{x}_{k-1} + \frac{3}{4}\mathbf{x}_{k-2} - \frac{5}{18}\mathbf{x}_{k-3} - J^-(\mathbf{x}_k, t_k) \left(\frac{13}{6}\sigma \mathbf{f}_t(\mathbf{x}_k, t_k) + h\mathbf{f}(\mathbf{x}_k, t_k) \right) + \mathbf{O}(\sigma^4), \quad (22)$$

where step size $h = 13\gamma\sigma/6 \in \mathbb{R}$ with γ defined as in Equation 3, and $\mathbf{O}(\sigma^4) \in \mathbb{R}^n$ denotes the vector with every element as $O(\sigma^4)$. By eliminating $\mathbf{O}(\sigma^4)$ from Equation 22, the following DTZNN model for solving the TVNE (Equation 1) is obtained:

$$\mathbf{x}_{k+1} = \frac{1}{36}\mathbf{x}_k + \frac{1}{2}\mathbf{x}_{k-1} + \frac{3}{4}\mathbf{x}_{k-2} - \frac{5}{18}\mathbf{x}_{k-3} - J^-(\mathbf{x}_k, t_k) \left(\frac{13}{6}\sigma \mathbf{f}_t(\mathbf{x}_k, t_k) + h\mathbf{f}(\mathbf{x}_k, t_k) \right), \quad (23)$$

of which the truncation error is $\mathbf{O}(\sigma^4)$.

Furthermore, to avoid the calculation for $J^-(\mathbf{x}_k, t_k)$ in Equation 23, another DTZNN model for Jacobian matrix inversion is established on the basis of Guo et al. (2017). Such a model is given by

$$M_{k+1} = \frac{1}{36}M_k + \frac{1}{2}M_{k-1} + \frac{3}{4}M_{k-2} - \frac{5}{18}M_{k-3} - \frac{13}{6}\sigma M_k \dot{J}(\dot{\mathbf{x}}_k, \mathbf{x}_k, t_k)M_k - hM_k(J(\mathbf{x}_k, t_k)M_k - I), \quad (24)$$

where M_k denotes the state matrix that can converge to $J^-(\mathbf{x}_k, t_k)$ with k being sufficiently large (which will be presented in Section 5), $\dot{J}(\dot{\mathbf{x}}_k, \mathbf{x}_k, t_k) \in \mathbb{R}^{n \times n}$ denotes the time derivative of $J(\mathbf{x}_k, t_k)$, and $I \in \mathbb{R}^{n \times n}$ denotes the identity matrix. Notably, $\dot{\mathbf{x}}_k$ is required for calculating $\dot{J}(\dot{\mathbf{x}}_k, \mathbf{x}_k, t_k)$, and $\dot{\mathbf{x}}_k$ is determined by $\dot{\mathbf{x}}_k = \mathbf{f}(\mathbf{x}_k, t_k) + \mathbf{f}_t(\mathbf{x}_k, t_k)$. For completeness, the derivation of the DTZNN model (Equation 24) is presented in the Appendix, which also indicates that Equation 24 has a truncation error of $\mathbf{O}(\sigma^4)$.

Therefore, replacing $J^-(\mathbf{x}_k, t_k)$ in Equation 23 with M_k and combining Equation 23 with Equation 24 yield the expression as

follows:

$$\begin{aligned} \mathbf{x}_{k+1} &= \frac{1}{36}\mathbf{x}_k + \frac{1}{2}\mathbf{x}_{k-1} + \frac{3}{4}\mathbf{x}_{k-2} - \frac{5}{18}\mathbf{x}_{k-3} \\ &\quad - M_k \left(\frac{13}{6}\sigma \mathbf{f}_t(\mathbf{x}_k, t_k) + h\mathbf{f}(\mathbf{x}_k, t_k) \right), \\ M_{k+1} &= \frac{1}{36}M_k + \frac{1}{2}M_{k-1} + \frac{3}{4}M_{k-2} - \frac{5}{18}M_{k-3} \\ &\quad - \frac{13}{6}\sigma M_k \dot{J}(\dot{\mathbf{x}}_k, \mathbf{x}_k, t_k)M_k - hM_k(J(\mathbf{x}_k, t_k)M_k - I), \end{aligned} \quad (25)$$

which is the proposed CDTZNN model to solve the TVNE (Equation 1). Evidently, the proposed CDTZNN model (Equation 25) does not require calculating $J^-(\mathbf{x}_k, t_k)$ at each iteration. Thus, the model can reduce the amount of calculation.

Algorithm	Procedure of CDTZNN model (Equation 25) for solving TVNE (Equation 1)
Step 1: Initialization	Set up time duration T , sampling gap σ , and step size h . Initialize t_0 , $\mathbf{x}_0$, and M_0 . Receive $\mathbf{f}(\mathbf{x}_0, t_0)$, $\mathbf{f}_t(\mathbf{x}_0, t_0)$, and $J(\mathbf{x}_0, t_0)$. Compute $\ \mathbf{e}_0\ _2 = \ \mathbf{f}(\mathbf{x}_0, t_0)\ _2$.
Step 2: First Loop ($k = 0, 1, 2$)	Compute $\dot{\mathbf{x}}_k = \mathbf{f}(\mathbf{x}_k, t_k) + \mathbf{f}_t(\mathbf{x}_k, t_k)$, and $\dot{J}(\dot{\mathbf{x}}_k, \mathbf{x}_k, t_k)$. Compute $\mathbf{x}_{k+1}$ and M_{k+1} through (Equation 26). Receive $\mathbf{f}(\mathbf{x}_{k+1}, t_{k+1})$, $\mathbf{f}_t(\mathbf{x}_{k+1}, t_{k+1})$, and $J(\mathbf{x}_{k+1}, t_{k+1})$. Compute $\ \mathbf{e}_{k+1}\ _2 = \ \mathbf{f}(\mathbf{x}_{k+1}, t_{k+1})\ _2$.
Step 3: Second loop ($k = 3, \dots, T/\sigma$)	Compute $\dot{\mathbf{x}}_k = \mathbf{f}(\mathbf{x}_k, t_k) + \mathbf{f}_t(\mathbf{x}_k, t_k)$, and $\dot{J}(\dot{\mathbf{x}}_k, \mathbf{x}_k, t_k)$. Compute $\mathbf{x}_{k+1}$ and M_{k+1} through (Equation 25). Receive $\mathbf{f}(\mathbf{x}_{k+1}, t_{k+1})$, $\mathbf{f}_t(\mathbf{x}_{k+1}, t_{k+1})$, and $J(\mathbf{x}_{k+1}, t_{k+1})$. Compute $\ \mathbf{e}_{k+1}\ _2 = \ \mathbf{f}(\mathbf{x}_{k+1}, t_{k+1})\ _2$.
Step 4: Output	Save state vector $\mathbf{x}_k$ and computational error $\ \mathbf{e}_k\ _2$. Plot the corresponding figures.

For the proposed CDTZNN model (Equation 25), with the given initial states $\mathbf{x}_0$ and M_0 , three states (i.e., $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$) for the first recursion and three states (i.e., M_1 , M_2 , and M_3) for the second recursion are obtained via the following computation:

$$\begin{cases} \mathbf{x}_1 = \mathbf{x}_0 - M_0(\sigma \mathbf{f}_t(\mathbf{x}_0, t_0) + h\mathbf{f}(\mathbf{x}_0, t_0)), \\ M_1 = M_0 - \sigma M_0 \dot{J}(\dot{\mathbf{x}}_0, \mathbf{x}_0, t_0)M_0 - hM_0(J(\mathbf{x}_0, t_0)M_0 - I), \\ \mathbf{x}_2 = \mathbf{x}_1 - M_1(\sigma \mathbf{f}_t(\mathbf{x}_1, t_1) + h\mathbf{f}(\mathbf{x}_1, t_1)), \\ M_2 = M_1 - \sigma M_1 \dot{J}(\dot{\mathbf{x}}_1, \mathbf{x}_1, t_1)M_1 - hM_1(J(\mathbf{x}_1, t_1)M_1 - I), \\ \mathbf{x}_3 = \mathbf{x}_2 - M_2(\sigma \mathbf{f}_t(\mathbf{x}_2, t_2) + h\mathbf{f}(\mathbf{x}_2, t_2)), \\ M_3 = M_2 - \sigma M_2 \dot{J}(\dot{\mathbf{x}}_2, \mathbf{x}_2, t_2)M_2 - hM_2(J(\mathbf{x}_2, t_2)M_2 - I). \end{cases} \quad (26)$$

In addition, for better understanding, the procedure of using the proposed CDTZNN model (Equation 25) to solve the TVNE (Equation 1) is presented in the Algorithm.

4.2 Theoretical analysis

This subsection theoretically analyzes the proposed CDTZNN model (Equation 25) to solve the TVNE (Equation 1) via the following theorems.

Theorem 2. *The proposed CDTZNN model (Equation 25) is zero stable and consistent and thus possesses the convergence property.*

Proof. For the proposed CDTZNN model (Equation 25), the characteristic polynomials of the first and second recursions are the

same as each other, which are written as the following unified form:

$$\tilde{P}_1(\vartheta) = \vartheta^4 - \frac{1}{36}\vartheta^3 - \frac{1}{2}\vartheta^2 - \frac{3}{4}\vartheta + \frac{5}{18}.$$

The roots of $\tilde{P}_1(\vartheta) = 0$ are given by

$$\begin{aligned}\vartheta_1 &= -0.5211 + i0.4694, \\ \vartheta_2 &= -0.5211 - i0.4694, \\ \vartheta_3 &= 0.2372 + i0.8065, \\ \vartheta_4 &= 0.2372 - i0.8065,\end{aligned}$$

where i denotes the imaginary unit. All the roots lie in the unit disk. According to Griffiths and Higham (2010), Equation 25 is zero stable.

On the basis of Section 4.1, the DTZNN models (Equation 23, 24) have the truncation error of $\mathcal{O}(\sigma^4)$. With the combination of such two models, the proposed CDTZNN model (Equation 25) is derived. This statement shows that the truncation error of (Equation 25) is $\mathcal{O}(\sigma^4)$. According to Griffiths and Higham (2010), the proposed CDTZNN model (Equation 25) is consistent. The zero stability and consistency ensure the convergence property of Equation 25. The proof is thus completed. ■

Theorem 3. The proposed CDTZNN model (Equation 25) is a linear convergent method for solving the TVNE (Equation 1).

Proof. Let $\Delta \mathbf{x}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\mathbf{x}_k^* = \mathbf{x}^*(t_k = k\tau)$ be the theoretical solution of Equation 1 (i.e., $\mathbf{f}(\mathbf{x}_k^*, t_k) = \mathbf{0}$). Considering the truncation error, the first recursion of the proposed CDTZNN model (Equation 25) is rewritten as follows:

$$\begin{aligned}\Delta \mathbf{x}_k &= -\frac{35}{36}\Delta \mathbf{x}_{k-1} - \frac{17}{36}\Delta \mathbf{x}_{k-2} + \frac{5}{18}\Delta \mathbf{x}_{k-3} \\ &\quad - M_k \left(\frac{13}{6}\sigma \mathbf{f}_t(\mathbf{x}_k, t_k) + h\mathbf{f}(\mathbf{x}_k, t_k) \right) + \mathcal{O}(\sigma^4),\end{aligned}$$

which is equivalent to the following expression:

$$\begin{aligned}\Delta \mathbf{x}_k &= \frac{7}{13}\Delta \mathbf{x}_k - \frac{35}{78}\Delta \mathbf{x}_{k-1} - \frac{17}{78}\Delta \mathbf{x}_{k-2} + \frac{5}{39}\Delta \mathbf{x}_{k-3} \\ &\quad - M_k \left(\sigma \mathbf{f}_t(\mathbf{x}_k, t_k) + \frac{6}{13}h\mathbf{f}(\mathbf{x}_k, t_k) \right) + \mathcal{O}(\sigma^4).\end{aligned}\quad (27)$$

It follows from Taylor series expansion (Mathews and Fink, 2004) that

$$\begin{aligned}\mathbf{f}(\mathbf{x}_{k+1}, t_{k+1}) &= \mathbf{f}(\mathbf{x}_k + \Delta \mathbf{x}_k, t_k + \sigma) \\ &= \mathbf{f}(\mathbf{x}_k, t_k) + J(\mathbf{x}_k, t_k)\Delta \mathbf{x}_k + \sigma \mathbf{f}_t(\mathbf{x}_k, t_k) + \mathcal{O}(\sigma^2),\end{aligned}\quad (28)$$

with $\mathcal{O}(\|\Delta \mathbf{x}_k\|_2^2)$ being absorbed into $\mathcal{O}(\sigma^2)$. Let us assume that M_k computed by the second recursion of

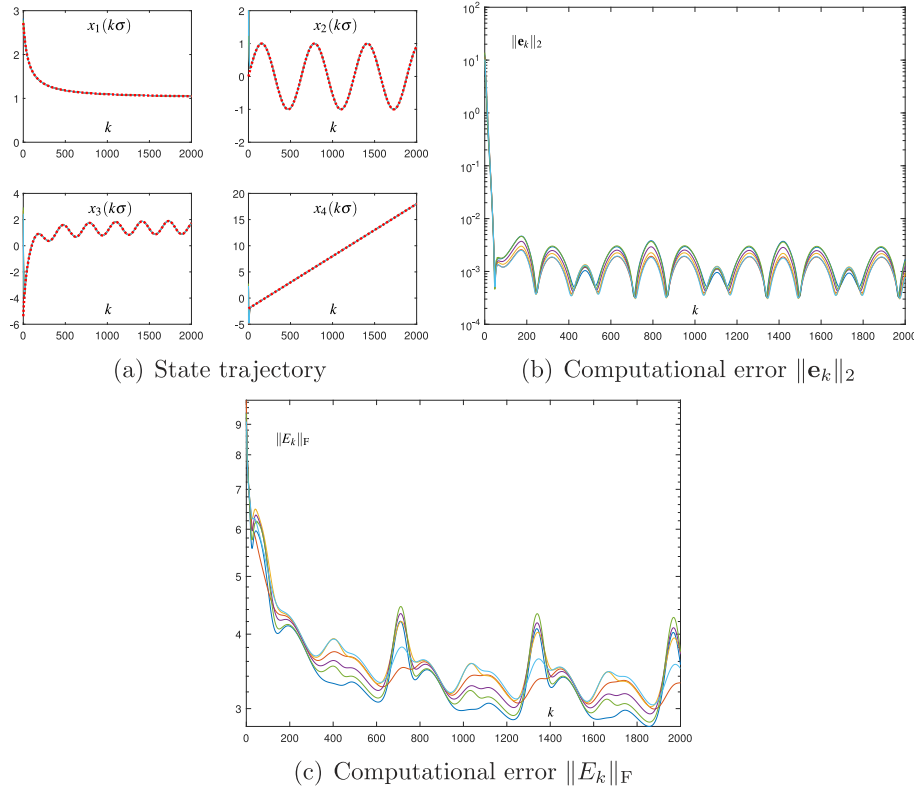


FIGURE 1

Numerical results of using the Broyden-aided DTZNN model (Equation 7) with $\sigma = 0.01$ and $h = 0.2$ for solving the TVNE (Equation 33). (A) State trajectory. (B) Computational error $\|e_k\|_2$. (C) Computational error $\|E_k\|_F$.

Equation 25 is exactly equal to $J^-(\mathbf{x}_k, t_k)$, that is, satisfying $J(\mathbf{x}_k, t_k)M_k = I$. Then, substituting Equation 27 into Equation 28 yields

$$\begin{aligned} \mathbf{f}(\mathbf{x}_{k+1}, t_{k+1}) = & J(\mathbf{x}_k, t_k) \left(\frac{7}{13} \Delta \mathbf{x}_k - \frac{35}{78} \Delta \mathbf{x}_{k-1} - \frac{17}{78} \Delta \mathbf{x}_{k-2} + \frac{5}{39} \Delta \mathbf{x}_{k-3} \right) \\ & + \left(1 - \frac{6}{13} h \right) \mathbf{f}(\mathbf{x}_k, t_k) + \mathbf{O}(\sigma^2) + \mathbf{O}(\sigma^4). \end{aligned} \quad (29)$$

Notably, Theorem 2 and its proof indicate that $\mathbf{x}_k$ computed by Equation 25 would converge to $\mathbf{x}_k^*$. In addition, when k is sufficiently large, $\mathbf{x}_k = \mathbf{x}_k^* + \mathbf{O}(\sigma^4)$, and $\mathbf{f}(\mathbf{x}_k, t_k) = \mathbf{f}(\mathbf{x}_k^*, t_k) + \mathbf{O}(\sigma^4)$. In view of that $\mathbf{f}(\mathbf{x}_k^*, t_k) = \mathbf{0}$, $J(\mathbf{x}_k, t_k)(42\Delta \mathbf{x}_k - 35\Delta \mathbf{x}_{k-1} - 17\Delta \mathbf{x}_{k-2} + 10\Delta \mathbf{x}_{k-3})/78 + \mathbf{O}(\sigma^2) = \alpha \mathbf{f}(\mathbf{x}_k, t_k) + \mathbf{O}(\sigma^4)$, with $\alpha \in \mathbb{R}$ denoting the correction parameter and k being sufficiently large. Thus, the reformulation of Equation 29 is given by

$$\mathbf{f}(\mathbf{x}_{k+1}, t_{k+1}) = \left(1 - \frac{6}{13} h + \alpha \right) \mathbf{f}(\mathbf{x}_k, t_k) + \mathbf{O}(\sigma^4). \quad (30)$$

The error function $\mathbf{e}(t)$ defined in Equation 2 is recalled, and the following result is obtained from Equation 30:

$$\begin{aligned} \mathbf{e}_k = \mathbf{f}(\mathbf{x}_k, t_k) &= \left(1 - \frac{6}{13} h + \alpha \right) \mathbf{e}_{k-1} + \mathbf{O}(\sigma^4) \\ &= \left(1 - \frac{6}{13} h + \alpha \right) \left(\left(1 - \frac{6}{13} h + \alpha \right) \mathbf{e}_{k-2} + \mathbf{O}(\sigma^4) \right) + \mathbf{O}(\sigma^4) \\ &= \left(1 - \frac{6}{13} h + \alpha \right)^2 \mathbf{e}_{k-2} + \left(1 - \frac{6}{13} h + \alpha \right) \mathbf{O}(\sigma^4) + \mathbf{O}(\sigma^4) \\ &\vdots \\ &= \left(1 - \frac{6}{13} h + \alpha \right)^k \mathbf{e}_0 + \beta \mathbf{O}(\sigma^4), \end{aligned} \quad (31)$$

where $\mathbf{e}_0 \in \mathbb{R}^n$ denotes the initial error and $\beta > 0 \in \mathbb{R}$ denotes a constant. As presented in Theorem 2, the proposed CDTZNN model (Equation 25) possesses the convergence property, which means that the error function $\mathbf{e}_k$ is convergent. With regard to Equation 31, the convergence of $\mathbf{e}_k$ implies that $-1 < 1 - 6h/13 + \alpha < 1$. Therefore, on the basis of Equation 30,

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{e}_{k+1}\|_2}{\|\mathbf{e}_k\|_2} = |1 - \frac{6}{13} h + \alpha| < 1, \quad (32)$$

which shows the linear convergence of the sequence $\{\mathbf{x}_k\}$ computed by Equation 25. Thus, the proposed CDTZNN model (Equation 25)

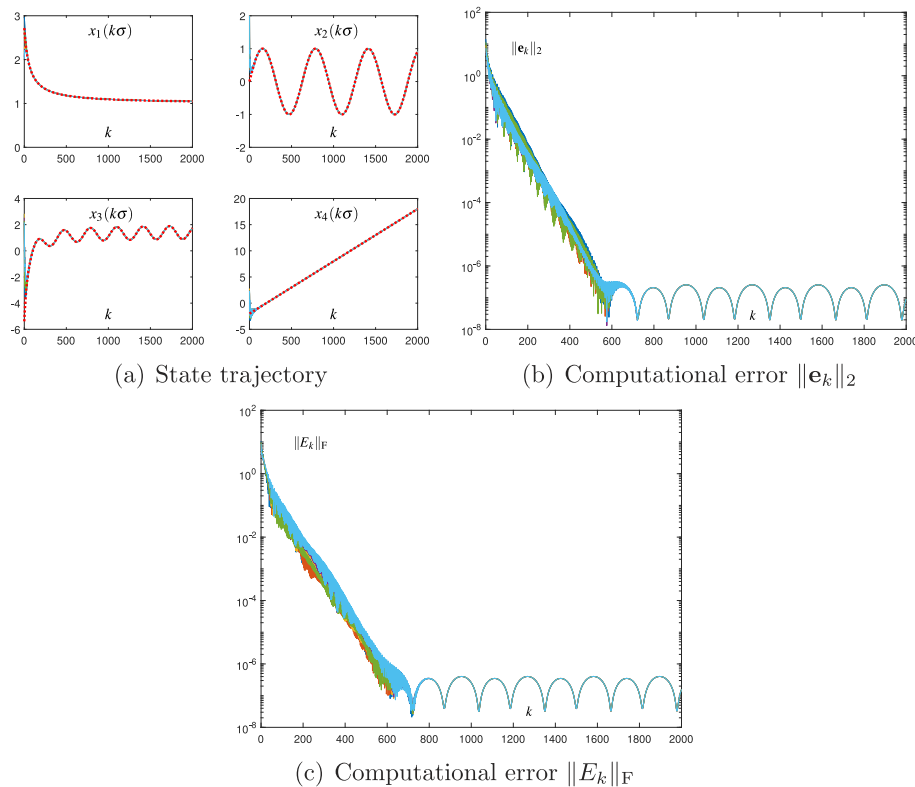


FIGURE 2

Numerical results of using the proposed CDTZNN model (Equation 25) with $\sigma = 0.01$ and $h = 0.2$ for solving the TVNE (Equation 33). (A) State trajectory. (B) Computational error $\|\mathbf{e}_k\|_2$. (C) Computational error $\|E_k\|_F$.

is a linear convergent method for solving the TVNE (Equation 1, which completes the proof. ■

Theorem 4. When a solvable TVNE (Equation 1) is considered, the SSCE of the proposed CDTZNN model (Equation 25) is of $O(\sigma^4)$.

Proof. Theorem 3 and its proof indicate that $-1 < 1 - 6h/13 + \alpha < 1$. Then, $\lim_{k \rightarrow \infty} (1 - 6h/13 + \alpha)^k = 0$. Therefore, the following result is obtained:

$$\lim_{k \rightarrow \infty} \|\mathbf{e}_k\|_2 = \lim_{k \rightarrow \infty} \|\mathbf{f}(\mathbf{x}_k, t_k)\|_2 = \lim_{k \rightarrow \infty} \|\beta \mathbf{O}(\sigma^4)\|_2 = O(\sigma^4),$$

which indicates that the SSCE of the proposed CDTZNN model (Equation 25) is of $O(\sigma^4)$. The proof is thus completed. ■

In summary, Theorems 2 through 4 theoretically guarantee the computational performance of Equation 25 in solving the TVNE (Equation 1). That is, the proposed CDTZNN model (Equation 25) can effectively generate an exact solution of Equation 1

5 Numerical verification and robot application

In this section, comparative numerical experiments are conducted to demonstrate the efficacy of the proposed CDTZNN model (Equation 25). Moreover, such a model is applied to the motion planning of robot manipulators to show the applicability of CDTZNN.

5.1 Model comparison

This subsection shows the comparative results of using the Broyden-aided DTZNN model (Equation 7) and the proposed CDTZNN model (Equation 25) to solve the following TVNE:

$$\mathbf{f}(\mathbf{x}(t), t) = \begin{bmatrix} \ln(x_1(t)) - 1/(t+1) \\ x_1(t)x_2(t) - \sin(t) \exp(1/(t+1)) \\ x_1^2(t) - \sin(t)x_2(t) + x_3(t) - 2 \\ x_1^2(t) - x_2^2(t) + x_3(t) + x_4(t) - t \end{bmatrix} = \mathbf{0}. \quad (33)$$

The theoretical solution to Equation 33 is

$$\mathbf{x}^*(t) = \begin{bmatrix} \exp(1/(t+1)) \\ \sin(t) \\ 2 - \exp(2/(t+1)) + \sin^2(t) \\ t - 2 \end{bmatrix},$$

which is given to validate the effectiveness of Equations 7, 25. Notably, the DTZNN models (Equations 5, 12) for solving Equation 33 have been investigated and compared in Guo et al. (2018), and their studies are thus omitted in this paper. The numerical results using Equations 7, 25 to solve Equation 33 are provided in Figures 1–4.

Figure 1 presents the results of using the Broyden-aided DTZNN model (Equation 7) with $\sigma = 0.01$ and $h = 0.2$, where the computational error $\|E_k\|_F$ is obtained by $\|E_k\|_F =$

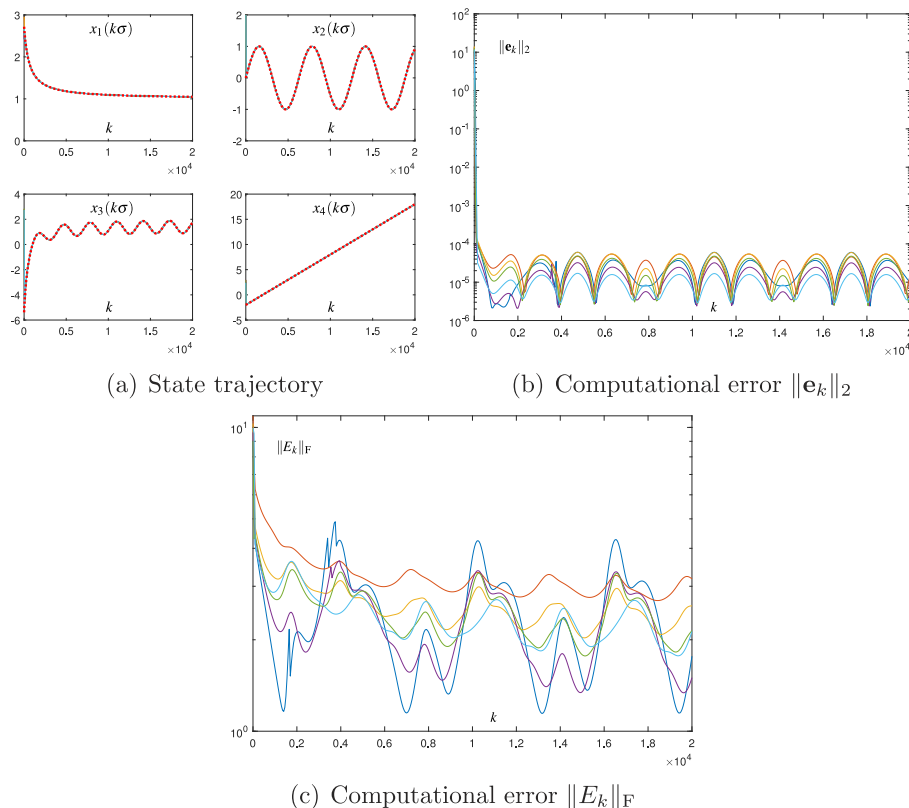


FIGURE 3

Numerical results of using the Broyden-aided DTZNN model (Equation 7) with $\sigma = 0.001$ and $h = 0.2$ for solving the TVNE (Equation 33). (A) State trajectory. (B) Computational error $\|e_k\|_2$. (C) Computational error $\|E_k\|_F$.

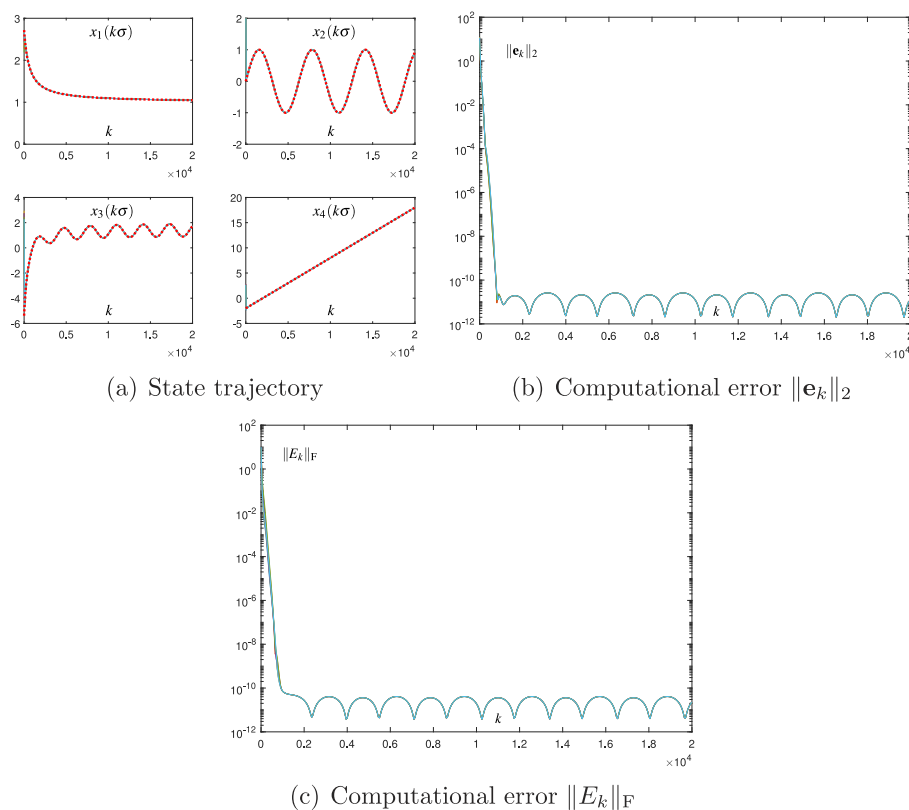


FIGURE 4

Numerical results of using the proposed CDTZNN model (Equation 25) with $\sigma = 0.001$ and $h = 0.2$ for solving the TVNE (Equation 33). (A) State trajectory. (B) Computational error $\|e_k\|_2$. (C) Computational error $\|E_k\|_F$.

$\|J(\mathbf{x}_k, t_k)M_k - I\|_F$ with $\|\cdot\|_F$ denoting the Frobenius norm of a matrix. As presented in Figure 1a, each state vector $\mathbf{x}_k$ of Equation 7 converges to the theoretical solution of Equation 33, that is, $\mathbf{x}_k \rightarrow \mathbf{x}_k^* = \mathbf{x}^*(t = k\sigma)$ as k evolves. As presented in Figure 1b, each computational error $\|e_k\|_2$ of Equation 7 presents the convergence characteristic, and the corresponding SSCE is of the order 10^{-3} . Notably, Figure 1c shows the computational error $\|E_k\|_F$ via the Broyden method (Equation 6) during the process of solving (Equation 33) for illustration and investigation. As shown in Figure 1c, $\|E_k\|_F$ at steady state is of the order 10^{-3} . These results indicate that the Broyden method (Equation 6) is effectively utilized to replace $J^-(\mathbf{x}_k, t_k)$ in Equation 5, and they verify that the Broyden-aided DTZNN model (Equation 7) can effectively solve the TVNE (Equation 33).

Figure 2 shows the results of using the proposed CDTZNN model (Equation 25) with $\sigma = 0.01$ and $h = 0.2$. As shown in Figure 2a, each state vector $\mathbf{x}_k$ of Equation 25 is convergent to $\mathbf{x}_k^*$ of Equation 33. As shown in Figure 2b, each computational error $\|e_k\|_2$ of the first recursion in Equation 25 also shows the convergence characteristic, and the corresponding SSCE is of the order 10^{-7} . As shown in Figure 2c, each computational error $\|E_k\|_F$ of the second recursion in Equation 25 at steady state is of the order 10^{-7} . These results verify that Equation 24 is successfully combined with Equation 23, and they substantiate that Equation 25 can effectively solve Equation 33. The comparison of Figure 1b with Figure 2b shows that the SSCE of Equation

25 is approximately 1,000 times smaller than the SSCE of Equation 7. In addition, the comparison of Figure 1c with Figure 2c reveals that the DTZNN model (Equation 24) is more effective than the Broyden method (Equation 6) in replacing $J^-(\mathbf{x}_k, t_k)$ at each iteration. These comparative results indicate that the proposed CDTZNN model (Equation 25) is advantageous over the Broyden-aided DTZNN model (Equation 7) in solving the TVNE (Equation 33).

The Broyden-aided DTZNN model (Equation 7) and the proposed CDTZNN model (Equation 25) are studied by decreasing σ from 0.01 to 0.001 and fixing the value of h . The related numerical results are presented in Figures 3 and 4, which demonstrate the effectiveness of Equation 7, 25 in terms of the state vector $\mathbf{x}_k$ converging to $\mathbf{x}_k^*$ and the computational errors $\|e_k\|_2$ and $\|E_k\|_F$ being sufficiently small. In particular, by observing Figures 1–4, the decrease in σ leads to the decrease in SSCEs. In addition, by decreasing σ , the performance of Equation 25 is improved more effectively than that of Equation 7. That is, when σ decreases by 10 times, $\|e_k\|_2$ of Equation 25 at steady state decreases by 10,000 times (i.e., from 10^{-7} to 10^{-11}), and $\|e_k\|_2$ of Equation 7 at steady state decreases only by 100 times (i.e., from 10^{-3} to 10^{-5}). Therefore, the proposed CDTZNN model (Equation 25) is superior to the Broyden-aided DTZNN model (Equation 7).

The abovementioned numerical results in Figures 1–4 substantiate the efficacy of the proposed CDTZNN model (Equation 25) in solving the TVNE (Equation 33).

5.2 Effect of σ and h

In this subsection, the effect of σ and h on six different DTZNN models (Equations 5, 7, 11, 12, 13, 25) is studied. Table 1 shows the corresponding numerical results, where the initial state is $\mathbf{x}_0 = \mathbf{0.2} \in \mathbb{R}^4$.

It follows from Table 1 that the computational errors $\|\mathbf{e}_k\|_2$ at steady state, i.e., SSCE, are sufficiently small, and the effectiveness of Equations 5, 7, 11, 12, 13, 25) in solving Equation 33 is thus validated. By observing the numerical results in Table 1, the following summary is obtained.

- 1) The computational performance of four different DTZNN models (Equations 5, 7, 12, 25) is improved by decreasing the value of σ , which shows the importance of σ in such models.
- 2) The SSCE of the Euler DTZNN model (Equation 5) and the Broyden-aided DTZNN model (Equation 7) is expressed in the mode of $O(\sigma^2)$. That is, the decrease in σ by 10 times results in the decrease in SSCE by 100 times. This phenomenon indicates the $O(\sigma^2)$ mode of Equations 5 or 7 in solving Equation 33, which coincides with Lemma 2.
- 3) The SSCEs of the Taylor DTZNN-I model (Equation 11) and the Taylor DTZNN-II model (Equation 12) are expressed in the mode of $O(\sigma^4)$. That is, the decrease in σ by 10 times results in the decrease in SSCE by 10,000 times. This phenomenon indicates the $O(\sigma^4)$ mode of Equation 12 in solving Equation 33, which coincides with Lemma 3.
- 4) The SSCE of the Taylor DTZNN-III model (Equation 13) is expressed in the mode of $O(\sigma^5)$. That is, the decrease in σ by 10 times results in the decrease in SSCE by 100,000 times. This phenomenon indicates the $O(\sigma^5)$ mode of Equation 12 in solving Equation 33, which coincides with Lemma 3.
- 5) The SSCE of the proposed CDTZNN model (Equation 25) is also expressed in the mode of $O(\sigma^4)$. That is, the decrease in σ by 10 times results in the decrease in SSCE by 10,000 times. This phenomenon indicates the $O(\sigma^4)$ mode of Equation 25 in solving Equation 33, which coincides with Theorem 4.
- 6) The computational performance of four different DTZNN models (Equations 5, 7, 11, 12, 13, 25) is further improved in solving Equation 33 with the appropriate increase in h .

On the basis of these results, a small value of σ (e.g., $\sigma = 0.01$) and a relatively large value h (e.g., $h = 0.2$) can ensure the superior computational performance of the proposed CDTZNN model (Equation 25).

Remark 2: With regard to the existing DTZNN models (Equation 5, 11, 12, 13) that require calculating $J^-(\mathbf{x}_k, t_k)$ at each iteration, they have at least the computational complexity of $O(n^3)$. In this sense, the proposed CDTZNN model (Equation 25) has the same computational complexity as Equations 5, 11, 12, 13. Notably, the direct calculation of $J^-(\mathbf{x}_k, t_k)$ is not required in Equation 25, and the amount of calculation can thus be effectively reduced. Compared with the Broyden-aided DTZNN model (Equation 7), more multiplications and more additions are needed for the proposed CDTZNN model (Equation 25). However, Equation 25 has a better performance than Equation 7 in solving Equation 1,

TABLE 1 SSCE of six different DTZNN models (Equation 5, 7, 11, 12, 13, 25) with different σ and h for solving the TVNE (Equation 33).

#	h	$\sigma = 0.01$	$\sigma = 0.001$	Mode
Euler DTZNN (Equation 5)	0.10	1.399×10^{-3}	1.414×10^{-5}	$O(\sigma^2)$
	0.15	9.385×10^{-4}	9.422×10^{-6}	
	0.20	7.056×10^{-4}	7.070×10^{-6}	
Broyden-aided DTZNN (Equation 7)	0.10	3.660×10^{-3}	4.616×10^{-5}	$O(\sigma^2)$
	0.15	2.397×10^{-3}	3.066×10^{-5}	
	0.20	2.109×10^{-3}	2.927×10^{-5}	
Taylor DTZNN-I (Equation 11)	0.10	4.554×10^{-7}	4.945×10^{-11}	$O(\sigma^4)$
	0.15	3.173×10^{-7}	3.299×10^{-11}	
	0.20	2.420×10^{-7}	2.474×10^{-11}	
Taylor DTZNN-II (Equation 12)	0.10	6.612×10^{-7}	7.059×10^{-11}	$O(\sigma^4)$
	0.15	4.569×10^{-7}	4.712×10^{-11}	
	0.20	3.472×10^{-7}	3.535×10^{-11}	
Taylor DTZNN-III (Equation 13)	0.10	1.260×10^{-8}	1.447×10^{-13}	$O(\sigma^5)$
	0.15	8.913×10^{-9}	9.763×10^{-14}	
	0.20	6.837×10^{-8}	7.618×10^{-14}	
CDTZNN (Equation 25)	0.10	4.874×10^{-7}	5.184×10^{-11}	$O(\sigma^4)$
	0.15	3.362×10^{-7}	3.456×10^{-11}	
	0.20	2.554×10^{-7}	2.592×10^{-11}	

which can be concluded by Table 1. In summary, the proposed CDTZNN model (Equation 25) is advantageous over the existing DTZNN models (Equations 5, 7, 11, 12, 13) in solving the TVNE (Equation 1).

5.3 Application to robot manipulators

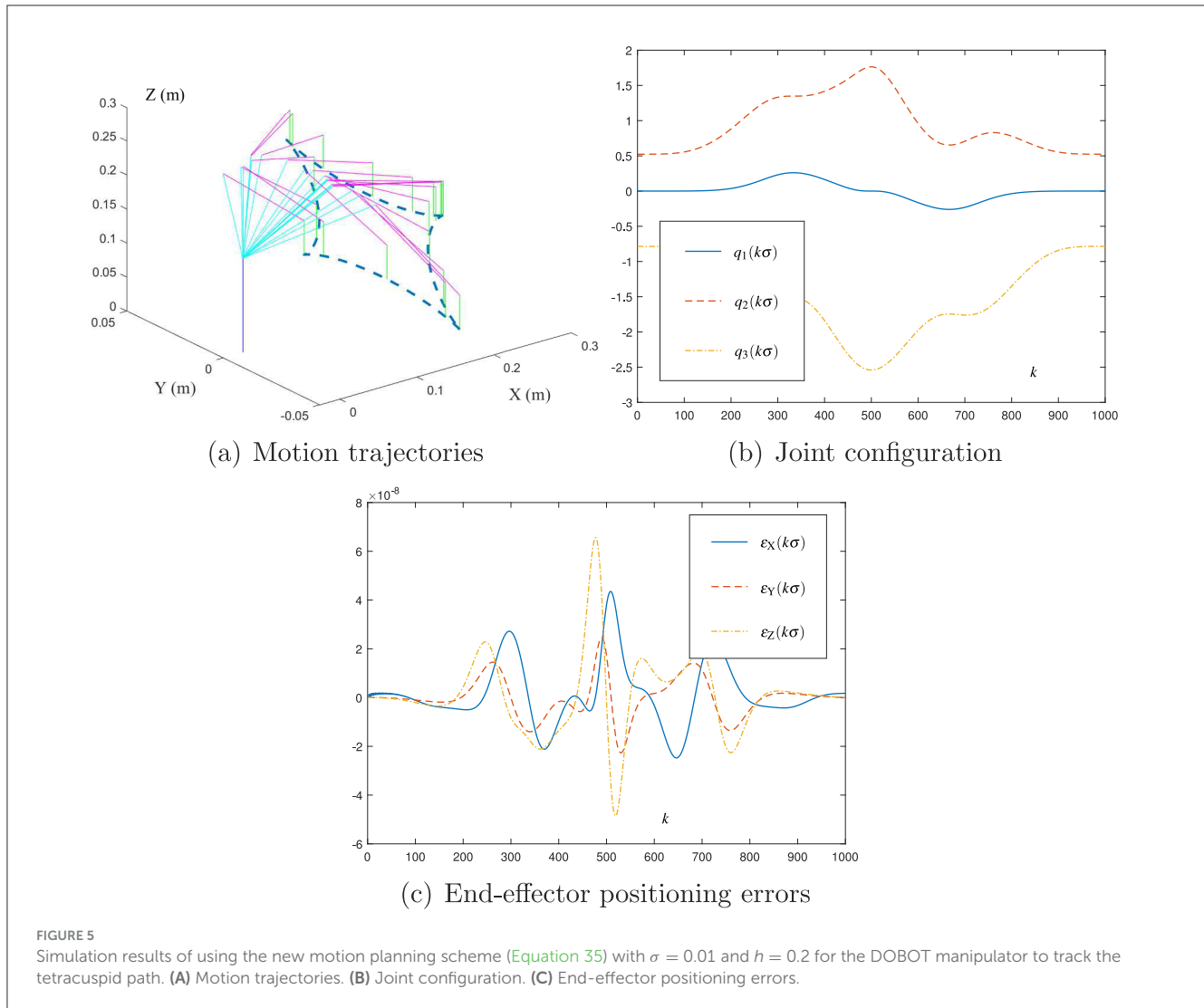
In general, the motion planning of a robot manipulator is described as that the joint configuration $\mathbf{q}(t) \in \mathbb{R}^n$ should be determined when giving the end-effector path $\mathbf{r}(t) \in \mathbb{R}^n$ (Li et al., 2019). Mathematically, the purpose of the robot's motion planning can be achieved by effectively solving the TVNE as follows:

$$\mathbf{f}(\mathbf{q}(t)) = \mathbf{r}(t), \quad (34)$$

where $\mathbf{f}(\cdot)$ denotes the differentiable nonlinear mapping with known parameters for a specific robot manipulator.

By extending the proposed CDTZNN model (Equation 25) to solve Equation 34, the following expression is obtained:

$$\begin{aligned}
 \mathbf{q}_{k+1} &= \frac{1}{36}\mathbf{q}_k + \frac{1}{2}\mathbf{q}_{k-1} + \frac{3}{4}\mathbf{q}_{k-2} - \frac{5}{18}\mathbf{q}_{k-3} \\
 &\quad + M_k\left(\frac{13}{6}\sigma\dot{\mathbf{r}}_k - h(\mathbf{f}(\mathbf{q}_k) - \mathbf{r}_k)\right), \\
 M_{k+1} &= \frac{1}{36}M_k + \frac{1}{2}M_{k-1} + \frac{3}{4}M_{k-2} - \frac{5}{18}M_{k-3} \\
 &\quad - \frac{13}{6}\sigma M_k J(\dot{\mathbf{q}}_k, \mathbf{q}_k)M_k - hM_k(J(\mathbf{x}_k)M_k - I),
 \end{aligned} \quad (35)$$



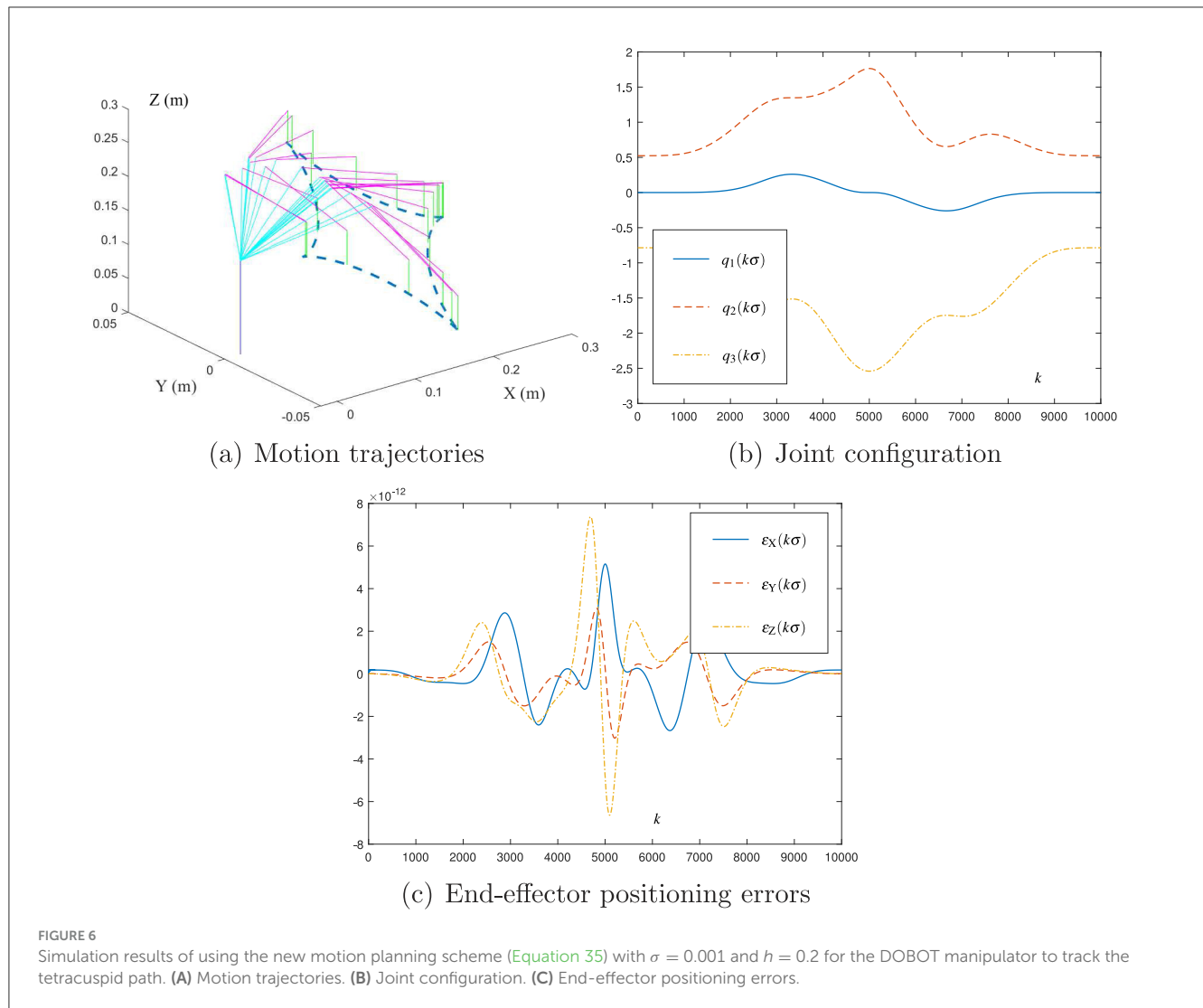
where the joint configuration $\mathbf{q}_k = \mathbf{q}(t = k\sigma)$, and the desired end-effector path $\mathbf{r}_k = \mathbf{r}(t = k\sigma)$ with $\dot{\mathbf{r}}_k$ as its time derivative. Therefore, the new motion planning scheme (Equation 35) without need for direct Jacobian matrix inversion is derived for robot manipulators.

It is worth mentioning here that the calculation procedure of the new motion planning scheme (Equation 35) is similar to that in the Algorithm. With the combined initial state $\{\mathbf{q}_0, M_0\}$, the three other combined states required for Equation 35, namely, $\{\mathbf{q}_1, M_1\}$, $\{\mathbf{q}_2, M_2\}$, and $\{\mathbf{q}_3, M_3\}$, are obtained via the calculation that is similar to Equation 26. Then, on the basis of the iterative calculation in Equation 35, the sequences of joint configuration $\{\mathbf{q}_k\}$ are obtained, where $k = 4, 5, \dots, T/\sigma$. Given that $\{\mathbf{q}_k\}$ is determined, the motion planning of robot manipulators is realized by Equation 35.

The following simulation results under the DOBOT manipulator (Xiao et al., 2019) are presented for validating the new motion planning scheme (Equation 35). In the simulations, the initial DOBOT joint configuration is $\mathbf{q}_0 = [0; \pi/6; -\pi/4]$ rad, and the motion task duration is $T = 10$ s.

Figure 5 presents the results of using the new motion planning scheme (Equation 35) with $\sigma = 0.01$ and $h = 0.2$ for the DOBOT manipulator to track a tetracuspid path, where the end-effector planning error is obtained as $\varepsilon_k = \mathbf{f}(\mathbf{q}_k) - \mathbf{r}_k \in \mathbb{R}^3$. As shown in Figures 5a, b, the DOBOT joint configuration $\{\mathbf{q}_k\}$ during the duration is determined via (Equation 35). Therefore, the motion planning task is successfully fulfilled. As shown in Figure 5c, the end-effector planning error is of order 10^{-8} m, which means that the DOBOT end-effector trajectory and the desired path match well. Notably, the initial point of the desired path is calculated and confirmed by $\mathbf{f}(\mathbf{q}_0)$, which means that the end-effector planning error ε_k at the initial time instant, i.e., ε_0 , is zero. Because of intrinsic DOBOT nonlinearity as well as the transient error computed by Equation 35, ε_k will increase in the transient phase. Since Equation 35 has the property of convergence, the magnitude of ε_k can be kept within the region of a small value, as presented in Figure 5c. These simulation results substantiate that the new scheme (Equation 35) is effective in the motion planning of the DOBOT manipulator.

The new motion planning scheme (Equation 35) is tested with the decrease in σ , and Figure 6 presents the related



simulation results. As presented in Figure 6, the DOBOT end-effector successfully follows the desired tetracuspid path with the maximal planning error being of order 10^{-12} m, which indicates the effectiveness of Equation 35). Moreover, the comparison of Figure 5c with Figure 6c shows that the decrease in σ by 10 times leads to the decrease in the maximal planning error by 10,000 times (i.e., from 10^{-8} m to 10^{-12} m). This phenomenon shows the important role of σ in the new motion planning scheme (Equation 35), and it expresses the $O(\sigma^4)$ form in the planning error via Equation refmp.new. Thus, σ in Equation 35 can be set as a small value to realize the required precision in robot applications.

Based on these simulation results, the new motion planning scheme (Equation 35) with $\sigma = 0.01$ and $h = 0.2$ is applied to the practical DOBOT manipulator, and the related experiment results are shown in Figure 7. Evidently, Figure 7 indicates that the DOBOT end-effector successfully tracks the desired tetracuspid path. Thus, the new motion planning scheme (Equation 35) is implemented effectively on the practical ROBOT manipulator.

Remark 3: As indicated by Figures 5, 6, the new motion planning scheme (Equation 35) possesses the $O(\sigma^4)$ form in the planning error. Then, even the sampling time σ is selected as a

large value (e.g., 0.1 s), the robot's planning error can be small enough (may be of order 10^{-4} m). Such an accuracy is still acceptable for robot manipulators. In practice, the sampling time σ of robot manipulators is generally selected to be 0.01s (and/or 0.02 s), as presented in Figure 7. With regard to this value of σ , the robot's planning error is kept within a sufficiently small value (e.g., of order 10^{-7} m or 10^{-8} m). In this sense, the planning accuracy and real-time feasibility can be both guaranteed by the new scheme (Equation 35) for the motion planning of robot manipulators.

In summary, the above simulation and experiment results in Figures 5–7 validate the efficacy of the new motion planning scheme (Equation 35) and show the applicability of the proposed CDTZNN model (Equation 25) to robot manipulators.

6 Conclusions

This study proposes a novel CDTZNN model (Equation 25) for solving the TVNE (Equation 1) on the basis of the discretization of the CTZNN model (Equation 3) using a new Taylor difference formula (Equation 14). Such a model is the combination of two

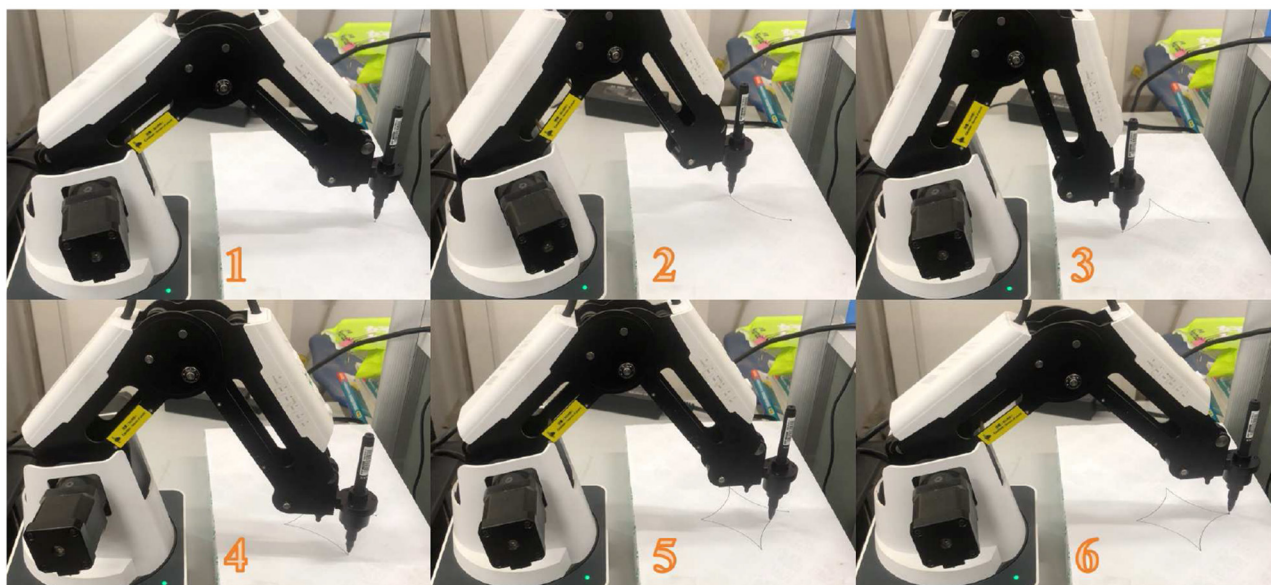


FIGURE 7

Experiment results of using the new motion planning scheme (Equation 35) with $\sigma = 0.01$ and $h = 0.2$ for the DOBOT manipulator to track the tetracuspid path.

DTZNN models (Equation 23) and Equation 24, and it does not require calculating the inverse of Jacobian matrix (thus reducing the amount of calculation). Theoretical analysis shows that the proposed CDTZNN model (Equation 25) possesses the convergence property, and it can generate the exact solution of Equation 1). Numerical results verify the efficacy of Equation 25 and show the $O(\sigma^4)$ form in the SSCE of Equation 25. Simulation results under the DOBOT manipulator demonstrate the efficacy of the new motion planning scheme (Equation 35) and the applicability of the proposed CDTZNN model (Equation 25). One future research direction is to extend the proposed CDTZNN model (Equation 25) to solve the TVNE with constraints (Guo et al., 2025). Another direction is to design new CDTZNN models to solve the TVNE with noise (Liao et al., 2022; Wei and Jin, 2024). By following this work, the proposed CDTZNN model (Equation 25) will be studied by implementing on practical robot manipulators. Besides, the CDTZNN method can be further investigated and extended to underwater acoustic sensor networks (Liu J. et al., 2024), multirobot systems (Liu M. et al., 2024), inter-robot management (Liao et al., 2024), and portfolio management (Cao et al., 2025).

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: <https://cam-can.mrc-cbu.cam.ac.uk/dataset>.

Ethics statement

Ethical approval was not required for the study involving humans in accordance with the local legislation and institutional

requirements. Written informed consent to participate in this study was not required from the participants or the participants' legal guardians/next of kin in accordance with the national legislation and the institutional requirements.

Author contributions

ZM: Investigation, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. SH: Conceptualization, Methodology, Supervision, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This study was supported by the Basic Scientific Research Project for University of Heilongjiang Province (with number being 135409611).

Acknowledgments

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

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.

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

- Broyden, C. G. (1965). A class of methods for solving nonlinear simultaneous equations. *Math. Comput.* 19, 577–593. doi: 10.1090/S0025-5718-1965-0198670-6
- Cang, N., Tang, H., Guo, D., Zhang, W., Li, W., Li, X., et al. (2024). Discrete-time zeroing neural network with quintic error mode for time-dependent nonlinear equation and its application to robot arms. *Appl. Soft Comput.* 157:111511. doi: 10.1016/j.asoc.2024.111511
- Cao, X., Lou, J., Liao, B., Peng, C., Pu, X., Khan, A. T., et al. (2025). based neural dynamics for portfolio management with tradeoffs of risks and profits under transaction costs. *Neural Netw.* 184:107090. doi: 10.1016/j.neunet.2024.107090
- Chen, J., and 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
- Dai, L., Xu, H., Zhang, Y., and Liao, B. (2024). Norm-based zeroing neural dynamics for time-variant non-linear equations. *CAAI Trans. Intell. Tech.* 9, 1561–1571. doi: 10.1049/cit2.12360
- Griffiths, D. F., and Higham, D. J. (2010). *Numerical Methods for Ordinary Differential Equations: Initial Value Problems*. London: Springer.
- Guo, D., Nie, Z., and Yan, L. (2017). Novel discrete-time Zhang neural network for time-varying matrix inversion. *IEEE Trans. Syst., Man, Cybern., Syst.* 47, 2301–2310. doi: 10.1109/TSMC.2017.2656941
- Guo, D., Xu, F., Li, Z., Nie, Z., and Shao, H. (2018). Design, verification and application of new discrete-time recurrent neural network for dynamic nonlinear equations solving. *IEEE Trans. Ind. Informat.* 14, 3936–3945. doi: 10.1109/TII.2017.2787729
- Guo, D., Yu, Y., Cang, N., Jia, Z., and Zhang, W. (2025). and Z. Ma, Discrete-time zeroing neural network for time-dependent constrained nonlinear equation with application to dual-arm robot system. *IEEE Trans. Syst., Man, Cybern., Syst.* 55, 1735–1745. doi: 10.1109/TSMC.2024.3512186
- 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., Liao, B., Liu, M., Xiao, L., Guo, D., Yan, X., et al. (2017a). Different-level simultaneous minimization scheme for fault tolerance of redundant manipulator aided with discrete-time recurrent neural network. *Front. Neurobot.* 11:50. doi: 10.3389/fnbot.2017.00050
- Jin, L., Zhang, Y., Li, S., and Zhang, Y. (2017b). Noise-tolerant ZNN models for solving time-varying zero-finding problems: a control-theoretic approach. *IEEE Trans. Autom. Control.* 62, 992–997. doi: 10.1109/TAC.2016.2566880
- Li, S., Jin, L., and Mirza, M. (2019). *A Kinematic Control of Redundant Robot Arms Using Neural Networks*. Hoboken: Wiley.
- Li, W., Xiao, L., and Liao, B. (2020). A finite-time convergent and noise-rejection recurrent neural network and its discretization for dynamic nonlinear equations solving. *IEEE Trans. Cybern.* 50, 3195–3207. doi: 10.1109/TCYB.2019.2906263
- Liao, B., Han, L., Cao, X., Li, S., and Double, J. L. (2022). integral-enhanced zeroing neural network with linear noise rejection for time-varying matrix inverse. *CAAI Trans. Intell. Tech.* 9, 197–210. doi: 10.1049/cit2.12161
- Liao, B., Hua, C., Xu, Q., Cao, X., and Li, S. (2024). Inter-robot management via neighboring robot sensing and measurement using a zeroing neural dynamics approach. *Expert Syst. Appl.* 244:122938. doi: 10.1016/j.eswa.2023.122938
- Liao, B., Zhang, Y., and Jin, L. (2016). Taylor $O(h^3)$ discretization of ZNN Models for dynamic equality-constrained quadratic programming with application to manipulators. *IEEE Trans. Neural Netw. Learn. Syst.* 27, 225–237. doi: 10.1109/TNNLS.2015.2435014
- Liao, Z., Gong, W., Yan, X., Wang, L., and Hu, C. (2020). Solving nonlinear equations system with dynamic repulsion-based evolutionary algorithms. *IEEE Trans. Syst. Man Cybern. Syst.* 50, 1590–1601. doi: 10.1109/TSMC.2018.2852798
- Liu, J., Du, X., and Jin, L. (2024). A localization algorithm for underwater acoustic sensor networks with improved newton iteration and simplified kalman filter. *IEEE Trans. Mobile Comput.* 23, 14459–14470. doi: 10.1109/TMC.2024.3443992
- 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
- Mathews, J. H., and Fink, K. D. (2004). *Numerical Methods Using MATLAB* (fourth ed.). Hoboken, NJ: Prentice Hall.
- Ramos, H., and Monteiro, T. T. (2017). Monteiro, A new approach based on the Newton's method to solve systems of nonlinear equations. *J. Comput. Appl. Math.* 318, 3–13. doi: 10.1016/j.cam.2016.12.019
- Sharma, J. R., Argyros, I. K., and Kumar, D. (2017). On a general class of optimal order multipoint methods for solving nonlinear equations. *J. Math. Analysis Appl.* 449, 994–1014. doi: 10.1016/j.jmaa.2016.12.051
- Tang, Z., and Zhang, Y. (2023). Continuous and discrete gradient-Zhang neuronet (GZN) with analyses for time-variant overdetermined linear equation system solving as well as mobile localization applications. *Neurocomputing* 561:126883. doi: 10.1016/j.neucom.2023.126883
- Wei, L., and Jin, L. (2024). Collaborative neural solution for time-varying nonconvex optimization with noise rejection. *IEEE Trans. Emerging Topics Comput. Intell.* 8, 2935–2948. doi: 10.1109/TETCI.2024.3369482
- Wu, J., Gong, W., and Wang, L. (2021). A clustering-based differential evolution with different crowding factors for nonlinear equations system. *Appl. Soft Comput.* 98:106733. doi: 10.1016/j.asoc.2020.106733
- Xiang, Q., Gong, H., and Hua, C. (2025). A new discrete-time denoising complex neurodynamics applied to dynamic complex generalized inverse matrices. *J. Supercomput.* 81:159. doi: 10.1007/s11227-024-06601-z
- Xiang, Q., Liao, B., Xiao, L., Lin, L., and Li, S. (2019). Discrete-time noise-tolerant Zhang neural network for dynamic matrix pseudoinversion. *Soft Comput.* 23, 755–766. doi: 10.1007/s00500-018-3119-8
- Xiao, L. (2016). A nonlinearly activated neural dynamics and its finite-time solution to time-varying nonlinear equation. *Neurocomputing.* 173, 1983–1988. doi: 10.1016/j.neucom.2015.08.031
- Xiao, L., Li, S., Lin, F., Tan, Z., and Khan, A. H. (2019). Zeroing neural dynamics for control design: Comprehensive analysis on stability, robustness, and convergence speed. *IEEE Trans. Ind. Informat.* 15, 2605–2616. doi: 10.1109/TII.2018.2867169
- Xiao, L., and Lu, R. (2015). Finite-time solution to nonlinear equation using recurrent neural dynamics with a specially-constructed activation function. *Neurocomputing.* 151, 246–251. doi: 10.1016/j.neucom.2014.09.047
- 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
- Zeng, Y., Liao, B., Li, Z., Hua, C., and Li, S. (2024). A comprehensive review of recent advances on intelligence algorithms and information engineering applications. *IEEE Access* 12, 135886–135912. doi: 10.1109/ACCESS.2024.3461756
- Zhang, X., Yu, Y., Cang, N., Guo, D., Li, S., Zhang, W., et al. (2025). Adaptive noise rejection strategy for cooperative motion control of dual-arm robots. *IEEE Robot. Autom. Lett.* 10, 868–874. doi: 10.1109/LRA.2024.3512992
- Zhang, Y., Peng, C., Li, W., and Shi, Y., and Ling, Y. (2012b). “Broyden-method aided discrete ZNN solving the systems of time-varying nonlinear equations,” in *Proc. Int. Conf. Control Eng. Commun. Tech* (Shenyang: IEEE), 492–495.
- Zhang, Y., Qiu, H., Peng, C., Shi, Y., and Tan, H. (2015). “Simply and effectively proved square characteristics of discrete-time ZD solving systems of time-varying nonlinear equations,” in *Proc. IEEE Int. Conf. Inf. Autom.*, 1457–1462. doi: 10.1109/ICInfA.2015.7279516
- Zhang, Y., Shi, Y., Xiao, L., and Mu, B. (2012a). “Convergence and stability results of Zhang neural network solving systems of time-varying nonlinear equations,” in *Proc. Int. Conf. Natural Comput* (Chongqing: IEEE), 150–154.
- Zhang, Y., and Yi, C. (2011). *Neural Networks and Neural-Dynamic Method*, NOVA Science. Commack, NY: Nova Science Publishers, Inc.
- Zhang, Y., Zhang, Y., Chen, D., Xiao, Z., and Yan, X. (2017). From Davidenko method to Zhang dynamics for nonlinear equation systems solving. *IEEE Trans. Syst. Man Cybern. Syst.* 47, 2817–2830. doi: 10.1109/TSMC.2016.2523917

Appendix

In this appendix, the derivation of the DTZNN model (Equation 24) for Jacobian matrix inversion is provided.

In Guo et al. (2017), the following CTZNN model is presented for computing the inverse of a time-varying matrix $A(t) \in \mathbb{R}^{n \times n}$:

$$\dot{M}(t) = -M(t)\dot{A}(t)M(t) - \mu M(t)(A(t)M(t) - I), \quad (36)$$

where $M(t) \in \mathbb{R}^{n \times n}$ denotes the state matrix with $\dot{M}(t)$ as its time derivative, $\dot{A}(t)$ denotes the time derivative of $A(t)$, and $\mu > 0 \in \mathbb{R}$ denotes the design parameter for scaling the convergence rate of Equation 36.

Using the difference formula (Equation 21) for discretizing the CTZNN model (Equation 36) yields the expression as follows:

$$\begin{aligned} M_{k+1} = & \frac{1}{36}M_k + \frac{1}{2}M_{k-1} + \frac{3}{4}M_{k-2} - \frac{5}{18}M_{k-3} \\ & - \frac{13}{6}\sigma M_k \dot{A}(t_k)M_k - hM_k(A(t_k)M_k - I) + \mathbf{O}(\sigma^4), \end{aligned} \quad (37)$$

where $\mathbf{O}(\sigma^4) \in \mathbb{R}^{n \times n}$ denotes the matrix with every element being $\mathbf{O}(\sigma^4)$. By eliminating $\mathbf{O}(\sigma^4)$ from Equation 37, the following DTZNN model for time-varying matrix inversion is obtained:

$$\begin{aligned} M_{k+1} = & \frac{1}{36}M_k + \frac{1}{2}M_{k-1} + \frac{3}{4}M_{k-2} - \frac{5}{18}M_{k-3} \\ & - \frac{13}{6}\sigma M_k \dot{A}(t_k)M_k - hM_k(A(t_k)M_k - I), \end{aligned} \quad (38)$$

which has a truncation error of $\mathbf{O}(\sigma^4)$.

With regard to the Jacobian matrix inversion in this study, $\dot{A}(t_k)$ and $A(t_k)$ in Equation 38 are replaced by $\dot{J}(\dot{\mathbf{x}}_k, \mathbf{x}_k, t_k)$ and $J(\mathbf{x}_k, t_k)$, respectively. Thus, the formulation is obtained as follows:

$$\begin{aligned} M_{k+1} = & \frac{1}{36}M_k + \frac{1}{2}M_{k-1} + \frac{3}{4}M_{k-2} - \frac{5}{18}M_{k-3} \\ & - \frac{13}{6}\sigma M_k \dot{J}(\dot{\mathbf{x}}_k, \mathbf{x}_k, t_k)M_k - hM_k(J(\mathbf{x}_k, t_k)M_k - I), \end{aligned}$$

which is exactly the DTZNN model (Equation 24) for Jacobian matrix inversion. On the basis of the aforementioned derivation, the truncation error of Equation 24 is $\mathbf{O}(\sigma^4)$.