Neural Processing Letters (2022) 54:125–144
https://doi.org/10.1007/s11063-021-10623-6

A Finite-Time Convergent Neural Network for Solving Time-Varying Linear Equations with Inequality Constraints Applied to Redundant Manipulator

Ying Kong[1] · Tanglong Hu[1] · Jingsheng Lei[1] · Renji Han[2]

Accepted: 5 August 2021 / Published online: 20 August 2021

Abstract

Zhang neural network (ZNN), a special recurrent neural network, has recently been established as an effective alternative for time-varying linear equations with inequality constraints (TLEIC) solving. Still, the convergent time produced by the ZNN model always tends to infinity. In contrast to ZNN, a finite-time convergent neural network (FCNN) is proposed for the TLEIC problem. By introducing a non-negative slack variable, the initial form of the TLEIC has been transformed into a system of time-varying linear equation. Afterwards, the stability and finite-time performance of the FCNN model is substantiated by the theoretical analysis. Then, simulation results further verify the effectiveness and superiority of the proposed FCNN model as compared with the ZNN model for solving TLEIC problem. Finally, the proposed FCNN model is successfully applied to the trajectory planning of redundant manipulators with joint limitations, thereby illustrating the applicability of the new neural network model.

Keywords Finite-time convergent neural network · Time-varying linear equations with inequality constraints · Redundant manipulators · Joint constraints

1 Introduction

Finding solutions for systems of linear equations with inequality constraints is the basis of many mathematical or engineering topics [1–5], such as robot kinematics [2], data processing [3] and control optimization [4]. Many achievements have been studied the solutions of linear equations with inequality constraints [6–11]. For example, Pang *et al.* [6] proposed an ABS-MPVT algorithm to solve linear equations and inequality systems. Murav'eva [7] studied the

✉ Ying Kong
kongying-888@163.com

Renji Han
renjihan@csu.edu.cn

[1] Department of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou, China

[2] School of Science, Zhejiang University of Science and Technology, Hangzhou, China

consistency and inconsistent radius of linear equations and inequality solutions. Moreover, other iterative algorithms were proposed to obtain the solutions of linear equalities with bound constraints, including the abstract algorithm [10] and Γ-algorithm [11].

Due to the parallel processing characteristics of recurrent neural network (RNN), many researchers are committed to solve mathematical problems with RNN [12–15]. A special RNN, the classical Hopfield neural network (HNN), was presented in literature [12]. Following the aspiring work, Xia *et al.* [13] investigated two RNN models based on continuous time and discrete time for time-invariant mathematical problems. Liang *et al.* [14] proposed an improved discrete time RNN model. In [15], various RNN dynamics were developed for the solution of linear underdetermined systems with inequality constraints. Significantly, all the methods mentioned above are used to solve time-invariant systems of linear equations and inequalities. In practical applications, most systems are not static but changing over time [2]. However, there are large errors and relatively delay in solving time-varying mathematical problems with the mentioned methods.

Different from the aforementioned neural networks, Zhang neural network (ZNN), also a special RNN, was presented by Zhang *et al.* aiming zero convergence with infinity time for solving time-varying mathematical problems [16–21]. For instance, Zhang *et al.* proposed two newly-designed ZNN models, which aims at time-varying underdetermined linear systems in [21]. Simulation results of trajectory planning of redundant manipulators verified the validity of the two newly-designed ZNN models. However, in many real applications, the limitation of solutions should be considered when time-varying linear equations solving. For example, each joint is physically constrained within a fixed moving range in inverse-kinematics of redundant manipulators. The unexpected joint-angle position will lead to wrong trajectory of the manipulator, and even even cause damage to itself. Thus, keeping the joint variables within its physical limits is greatly important. For this purpose, two new constructed ZNN models incorporating bound constraints with time-varying linear equations were proved to have perfect convergence as long as time is infinity [22]. However, the long convergent time of time-varying linear equations with inequality constraints (TLEIC) will affect the executive effectiveness, which tends to add heavy burden for time-varying systems and is incapable to complete the given task.

To accelerate convergent speed and shorten convergent time, many studies have been done about finite-time convergence [23–33]. For instance, Li *et al.* [23] first presented a particular activation function, which can make ZNN model converge to zero within finite time. In [24], this activation function was utilized for establishing a ZNN model with finite-time characteristics to solve time-varying inequality problems. Aiming at the solution of time-varying nonlinear equation, Lin et al. also proposed two novel nonlinear activation functions in [25] for enhancing the convergence performance of the ZNN model. Jin *et al.* [26] used a power-versatile activation function in ZNN model to solve dynamic matrix inversion (DMI), and proved the excellent convergence performance and great robustness of the proposed model for solving DMI problem under various interference environments. Different from [23–26] (i.e., improving activation functions), some new design formulas were designed in [28–32] to accelerate ZNN model. For example, Jin *et al.* [28] replaced the error function $\boldsymbol{e}(t)$ in the original ZNN design formula with a Bernoulli equation, thereby obtaining a new design formula and constructing a improved ZNN (IZNN) model to solve time-invariant and time-varying nonlinear equation. In [30–32], a new design formula with linear activation function was investigated by Lin *et al.*, and established three different finite-time RNN (FTRNN) models for the online solutions of time-varying matrix inversion, time-varying matrix square root and time-varying Sylvester matrix equation. Comparative results were visualized to determine the superiority of FTRNN models. In [33], a class of new evolution formula was exploited for quadratic programming problems, and its finite-time convergence and robustness were also substantiated by the application of k-winner-take-all (k-WTA). However, at present, no research has focused on solving TLEIC directly in finite time.

Aspired by the above RNN designing rules for time-varying problems, a new finite-time convergent neural network (FCNN) is proposed in this paper to solve TLEIC. Theoretical analysis proves the stability and finite-time convergence of the proposed FCNN model in detail. The superiority of the FCNN model is substantiated in the simulation experiment. This work is attempted to show the proposed FCNN can directly handle the time-varying linear equations considering the corresponding inequality constraints. Furthermore, we apply the proposed FCNN model to trajectory planning of redundant manipulators and solve its TLEIC problem by referring to each joint variable limitations. Simulations on six-degrees-of-freedom (six-DOF) planar manipulator and PUMA560 redundant manipulator show the application of the proposed FCNN model. With its motion planning of redundant manipulators, this work serves as an improvement in the finite-time research of redundant manipulators with joint physical constraints.

The rest of this paper consists of six sections. Sect. 2 describes the specific form of TLEIC and gives out the key steps of transforming TLEIC into time-varying linear equations. In Sect. 3, the new design formula of FCNN is presented, and the corresponding FCNN model is established for TLEIC. Sect. 4 proves the stability and finite-time convergence of the proposed FCNN model. The FCNN model is further simulated and compared with the ZNN model for the TLEIC problem solving in Sect. 5. Section 6 demonstrates the application of the proposed FCNN model to trajectory planning of redundant manipulators. Finally, Sect. 7 gives a summary of this paper. It should be pointed out that the main contributions of this paper are as follows.

1) This paper mainly proposes a FCNN model via a novel design formula for directly solving TLEIC within finite time.
2) The stability and finite-time convergence of the FCNN model for TLEIC is proved in detail. Besides, the finite convergent time of the FCNN model is derived theoretically, which further proves the effectiveness in finite time.
3) Numerical simulation verifies the validity and finite-time convergence of the proposed FCNN model. Furthermore, the superiority of the FCNN model for solving TLEIC problem is verified, as compared with the ZNN model under various activation functions.
4) The feasibility and practicability of FCNN model in the application of trajectory planning for redundant manipulators with joint constraints (namely 6-DOF planar manipulator and PUMA560 redundant manipulator) are further demonstrated.

2 Problem Description and Transformation

This section mainly describes the mathematical expression of TLEIC and the transformation steps required in the solving procedure.

2.1 Problem Description

In this subsection, mathematical form of the TLEIC problem can be described as

$$\begin{cases} \mathcal{W}(t)\boldsymbol{x}(t) = \boldsymbol{l}(t) \\ \boldsymbol{x}_1 \leq \boldsymbol{x}(t) \leq \boldsymbol{x}_2 \end{cases} \tag{1}$$

where $\mathcal{W}(t) \in R^{m\times n}$ is a given non-zero time-varying coefficient matrix with $m < n$, and $\boldsymbol{l}(t) \in R^m$ is a given non-zero time-varying vector. $\boldsymbol{x}(t) \in R^n$ is an unknown vector to be solved in TLEIC problem (1). $\boldsymbol{x}_1 \in R^n$ and $\boldsymbol{x}_2 \in R^n$ denote the lower and upper bounds of $\boldsymbol{x}(t)$, both of which are constant vectors. Besides, to ensure the existence of $\boldsymbol{x}(t)$, we assume $\mathcal{W}(t)$ is nonsingular in (1) at all $t \in [0, +\infty)$ in this paper.

Subsequently, the inequality constraints on $\boldsymbol{x}(t)$ in TLEIC problem (1) can be transformed into a system of linear inequalities, i.e.,

$$\boldsymbol{U}\boldsymbol{x}(t) \leq \boldsymbol{c}$$

where $\boldsymbol{U} = [-\boldsymbol{I}, \boldsymbol{I}]^{\mathrm{T}} \in R^{2n\times n}$, with $\boldsymbol{I} \in R^{n\times n}$ denotes an identity matrix. $\boldsymbol{c} \in R^{2n}$ as a vector composed of the $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, and the specific form is $\boldsymbol{c} = \left[-\boldsymbol{x}_1^{\mathrm{T}}, \boldsymbol{x}_2^{\mathrm{T}}\right]^{\mathrm{T}}$. Note that superscript $^{\mathrm{T}}$ as the transpose operation. Then, the following equations can be obtained

$$\begin{cases} \mathcal{W}(t)\boldsymbol{x}(t) = \boldsymbol{l}(t) \\ \boldsymbol{U}\boldsymbol{x}(t) \leq \boldsymbol{c} \end{cases}. \tag{2}$$

which is equivalent to TLEIC problem (1).

2.2 Problem Transformation

Previously, we have introduced the specific form of the TLEIC. Then, by introducing an unknown slack variable $\boldsymbol{s}^{\cdot 2}$, which is greater than or equal to $\boldsymbol{0}$ [16], the inequality constraints in (2) can be rewritten as the following time-varying nonlinear equations:

$$\boldsymbol{U}\boldsymbol{x}(t) + \boldsymbol{s}^{\cdot 2}(t) - \boldsymbol{c} = \boldsymbol{0} \tag{3}$$

where the superscript $^{\cdot 2}$ in $\boldsymbol{s}^{\cdot 2}(t)$ means the square operation of each element in $\boldsymbol{s}(t)$. For ease of understanding, the relationship between $\boldsymbol{s}(t)$ and $\boldsymbol{s}^{\cdot 2}(t)$ are given here:

$$\boldsymbol{s}(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \\ \vdots \\ s_{2n}(t) \end{bmatrix} \in R^{2n},\ \boldsymbol{s}^{\cdot 2}(t) = \begin{bmatrix} s_1^2(t) \\ s_2^2(t) \\ s_3^2(t) \\ \vdots \\ s_{2n}^2(t) \end{bmatrix} \in R^{2n}.$$

The following system including time-varying linear and nonlinear equations can be derived from (2) through analysis above:

$$\begin{cases} \mathcal{W}(t)\boldsymbol{x}(t) = \boldsymbol{l}(t) \\ \boldsymbol{U}\boldsymbol{x}(t) + \boldsymbol{s}^{\cdot 2}(t) = \boldsymbol{c} \end{cases}. \tag{4}$$

In addition, vector $\boldsymbol{s}(t)$ is determined in parallel with the solution of (1). For further discussion, we first convert $\boldsymbol{s}^{\cdot 2}(t)$ into the following form:

$$\boldsymbol{s}^{\cdot 2}(t) = \boldsymbol{V}(t)\boldsymbol{s}(t) \tag{5}$$

where $\boldsymbol{V}(t) \in R^{2n\times 2n}$ is a time-varying diagonal matrix with all the elements in $\boldsymbol{s}(t)$ as the main diagonal element, i.e.,

$$\boldsymbol{V}(t) = \operatorname{diag}\left(s_1(t), s_2(t), s_3(t), \cdots, s_{2n}(t)\right).$$

Then, according to Eq. (5), we reconstruct the the time-varying linear and nonlinear equations (4) as

$$\mathcal{K}(t)z(t) = \boldsymbol{q}(t) \tag{6}$$

where $\mathcal{K}(t) \in R^{(2n+m)\times 3n}$, $\boldsymbol{q}(t) \in R^{2n+m}$ and $z(t) \in R^{3n}$ are respectively expressed as

$$\mathcal{K}(t) = \begin{bmatrix} \mathcal{W}(t) & \boldsymbol{0} \\ \boldsymbol{U} & V(t) \end{bmatrix}, \ \boldsymbol{q}(t) = \begin{bmatrix} \boldsymbol{l}(t) \\ \boldsymbol{c} \end{bmatrix}, \ z(t) = \begin{bmatrix} \boldsymbol{x}(t) \\ s(t) \end{bmatrix}.$$

Remark 1 To guarantee the transformed Eq. (6) is solvable, we assume that Eq. (6) satisfies the following condition in this paper:

$$\text{rank}\ (\mathcal{K}(t)) = \text{rank}\ \left(\begin{bmatrix} \mathcal{K}(t) \ \boldsymbol{q}(t) \end{bmatrix}\right) = 3n$$

where operator rank$(\cdot)$ denotes the rank of a matrix.

Based on the above transformation, solving TLEIC problem (1) is equivalent to the solution of Eq. (4) or Eq. (6) at any $t \in [0, +\infty)$.

3 FCNN Description and Model Establishment

In this section, a finite-time convergent neural network (FCNN) model with finite-time characteristics is presented to solving the TLEIC problem (1).

Based on Eq. (6), we first define a indefinite vector-valued time-varying error function $\boldsymbol{e}(t)$ as

$$\boldsymbol{e}(t) = \mathcal{K}(t)z(t) - \boldsymbol{q}(t) \tag{7}$$

By the above definition, the time-derivative of $\boldsymbol{e}(t)$, i.e., $\dot{\boldsymbol{e}}(t)$, is given as

$$\dot{\boldsymbol{e}}(t) = \mathcal{K}(t)\dot{z}(t) + \dot{\mathcal{K}}(t)z(t) - \dot{\boldsymbol{q}}(t) \tag{8}$$

where $\dot{\mathcal{K}}(t)$, $\dot{\boldsymbol{q}}(t)$ and $\dot{z}(t)$ are the time-derivative of $\mathcal{K}(t)$, $\boldsymbol{q}(t)$ and $z(t)$, respectively. Then, in order to zeroing $\boldsymbol{e}(t)$, a novel design formula of FCNN is given as

$$\dot{\boldsymbol{e}}(t) = \boldsymbol{\Phi}\left(-\xi_1 \boldsymbol{e}^{\cdot d/b}(t) - \xi_2 \boldsymbol{e}(t)\right) \tag{9}$$

where $\xi_1 > 0 \in R$ and $\xi_2 > 0 \in R$ are design parameters. The superscript $^{\cdot d/b}$ in $\boldsymbol{e}^{\cdot d/b}(t)$ indicates that the exponential term d/b is added to each element in $\boldsymbol{e}(t)$, and the specific expression is

$$\boldsymbol{e}^{\cdot d/b}(t) = \begin{bmatrix} e_1^{d/b}(t) \\ e_2^{d/b}(t) \\ e_3^{d/b}(t) \\ \vdots \end{bmatrix}.$$

Moreover, d and b as positive odd numbers in Eq. (9), and satisfy $d < b$. $\boldsymbol{\Phi}(\cdot)$: $R^{2n+m} \rightarrow R^{2n+m}$ denotes an activation function, which can only be a monotone increasing odd function, i.e., $\boldsymbol{\Phi}(\cdot) = -\boldsymbol{\Phi}(-\cdot)$, such as linear function, sigmoid function, hyperbolic-sine function, etc. Note that this paper focus on the study of a design formula (i.e., Eq. (9)) to establish FCNN model for solving TLEIC problem (1). Therefore, in the experimental simulation, only linear activation function is used in the FCNN model (11).

On the basis of Eq. (5), the time-derivative of $\boldsymbol{s}^{\cdot 2}(t)$, i.e., $\dot{\boldsymbol{s}}^{\cdot 2}(t)$, can be obtained as

$$\dot{\boldsymbol{s}}^{\cdot 2}(t) = 2\boldsymbol{V}(t)\dot{\boldsymbol{s}}(t). \tag{10}$$

According to the Eqs. (7), (8) and (10), (9) can be expanded into the following equation:

$$\begin{bmatrix} \mathcal{W}(t) & \mathbf{0} \\ \boldsymbol{U} & 2\boldsymbol{V}(t) \end{bmatrix}\begin{bmatrix} \dot{\boldsymbol{x}}(t) \\ \dot{\boldsymbol{s}}(t) \end{bmatrix} + \begin{bmatrix} \dot{\mathcal{W}}(t) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{s}(t) \end{bmatrix} - \begin{bmatrix} \dot{\boldsymbol{l}}(t) \\ \mathbf{0} \end{bmatrix}$$
$$= \boldsymbol{\Phi}\left(-\xi_1\left(\begin{bmatrix} \mathcal{W}(t) & \mathbf{0} \\ \boldsymbol{U} & \boldsymbol{V}(t) \end{bmatrix}\begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{s}(t) \end{bmatrix} - \begin{bmatrix} \boldsymbol{l}(t) \\ \boldsymbol{c} \end{bmatrix}\right)^{.d/b} - \xi_2\left(\begin{bmatrix} \mathcal{W}(t) & \mathbf{0} \\ \boldsymbol{U} & \boldsymbol{V}(t) \end{bmatrix}\begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{s}(t) \end{bmatrix} - \begin{bmatrix} \boldsymbol{l}(t) \\ \boldsymbol{c} \end{bmatrix}\right)\right).$$

where $\dot{\mathcal{W}}(t)$, $\dot{\boldsymbol{x}}(t)$, $\dot{\boldsymbol{s}}(t)$ and $\dot{\boldsymbol{l}}(t)$ are the time-derivative of $\mathcal{W}(t)$, $\boldsymbol{x}(t)$, $\boldsymbol{s}(t)$ and $\boldsymbol{l}(t)$, respectively. Defining the time-varying matrices $\boldsymbol{G}(t)$ and $\boldsymbol{H}(t)$ as

$$\boldsymbol{G}(t) = \begin{bmatrix} \mathcal{W}(t) & \mathbf{0} \\ \boldsymbol{U} & 2\boldsymbol{V}(t) \end{bmatrix} \in R^{(2n+m)\times 3n}, \quad \boldsymbol{H}(t) = \begin{bmatrix} \dot{\mathcal{W}}(t) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in R^{(2n+m)\times 3n}$$

following implicit dynamic equation can be obtained:

$$\boldsymbol{G}(t)\dot{\boldsymbol{z}}(t) + \boldsymbol{H}(t)\boldsymbol{z}(t) - \dot{\boldsymbol{q}}(t) = \boldsymbol{\Phi}(-\xi_1(\mathcal{K}(t)\boldsymbol{z}(t) - \boldsymbol{q}(t))^{.d/b} - \xi_2(\mathcal{K}(t)\boldsymbol{z}(t) - \boldsymbol{q}(t))).$$

Furthermore, FCNN model can be expressed as follows:

$$\begin{aligned}\dot{\boldsymbol{z}}(t) = \boldsymbol{G}^{\dagger}(t)(&-\boldsymbol{H}(t)\boldsymbol{z}(t) + \dot{\boldsymbol{q}}(t) + \boldsymbol{\Phi}(-\xi_1(\mathcal{K}(t)\boldsymbol{z}(t) - \boldsymbol{q}(t))^{.d/b} \\ &-\xi_2(\mathcal{K}(t)\boldsymbol{z}(t) - \boldsymbol{q}(t)))\end{aligned} \tag{11}$$

where $\boldsymbol{G}^{\dagger}(t)$ is the right pseudo-inverse of $\boldsymbol{G}(t)$, and its concrete expression is $\boldsymbol{G}^{\dagger}(t) = \boldsymbol{G}(t)^{\mathrm{T}}(\boldsymbol{G}(t)\boldsymbol{G}^{\mathrm{T}}(t))^{-1}$.

To facilitate understanding, Fig. 1 presents the nteuron structure of FCNN model (11).

Remark 2 The i-th neuron of FCNN model (11) is expressed as

$$z_i = \int \sum_{j=1}^{2n+m} \tilde{g}_{ij}\left(-\sum_{p=1}^{3n} h_{jp}z_p + \dot{q}_j\phi\left(-\xi_1\left(\sum_{p=1}^{3n} k_{jp}z_p - q_j\right)^{d/b} - \xi_2\left(\sum_{p=1}^{3n} k_{jp}z_p - q_j\right)\right)\right)\mathrm{d}t$$

where $\tilde{g}_{ij}$, h_{jp}, k_{jp}, q_j, $\dot{q}_j$ and z_i are the ij-th element of $\boldsymbol{G}^{\dagger}(t)$, the jp-th element of $\boldsymbol{H}(t)$, the jp-th element of $\mathcal{K}(t)$, the j-th element of $\boldsymbol{q}(t)$, the j-th element of $\dot{\boldsymbol{q}}(t)$ and the i-th element of $z(t)$, respectively. $\phi(\cdot)$ is an element of $\boldsymbol{\Phi}(\cdot)$.

4 Theoretical Analysis

In this section, stability and finite-time convergence of the FCNN model (11) are analyzed, which verifies the feasibility of the FCNN model for the solution of TLEIC problem (1).

4.1 Stability Analysis

Theorem 1 *Given time-varying coefficient matrix $\mathcal{W}(t)$, time-varying vector $\boldsymbol{l}(t)$ and the bounded vector constraints $\boldsymbol{c}$, if $\boldsymbol{\Phi}(\cdot)$ is a monotone increasing odd function, the vector-valued $\boldsymbol{x}(t)$ of FCNN model (11) can globally converges to the theoretical solution $\boldsymbol{x}^*(t)$ of TLEIC problem (1).*

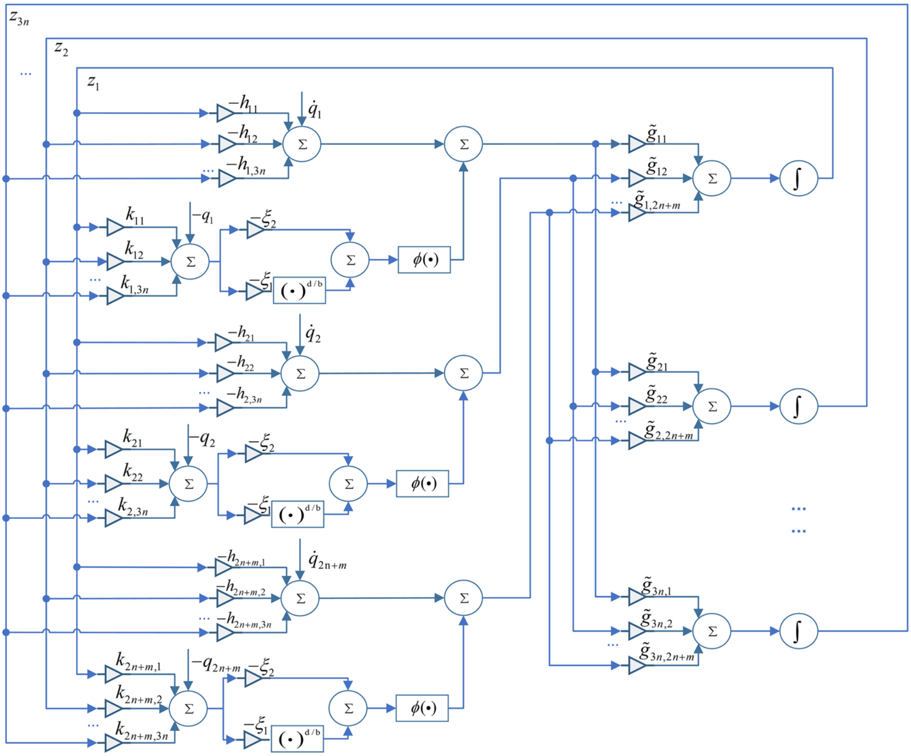

Fig. 1 Neuron structure of FCNN model (11) for TLEIC problem (1) solving

Proof Let's suppose $\boldsymbol{x}^*(t)$ be the theoretical solution of TLEIC problem (1), and the slack variable $\boldsymbol{s}^{*.2}(t)$ will be exists that satisfies the following equations:

$$\begin{cases} \mathcal{W}(t)\boldsymbol{x}^*(t) = \boldsymbol{l}(t) \\ \boldsymbol{U}\boldsymbol{x}^*(t) + \boldsymbol{s}^{*.2}(t) = \boldsymbol{c} \end{cases}. \tag{12}$$

Furthermore, the time-derivative of Eq. (12) can be obtained as

$$\begin{cases} \mathcal{W}(t)\dot{\boldsymbol{x}}^*(t) + \dot{\mathcal{W}}(t)\boldsymbol{x}^*(t) = \dot{\boldsymbol{l}}(t) \\ \boldsymbol{U}\dot{\boldsymbol{x}}^*(t) + \dot{\boldsymbol{s}}^{*.2}(t) = \boldsymbol{0} \end{cases}. \tag{13}$$

Then, expanding FCNN model (11) and converting into

$$\begin{cases} \mathcal{W}(t)\dot{\boldsymbol{x}}(t) + \dot{\mathcal{W}}(t)\boldsymbol{x}(t) - \dot{\boldsymbol{l}}(t) \\ = \boldsymbol{\Phi}(-\xi_1(\mathcal{W}(t)\boldsymbol{x}(t) - \boldsymbol{l}(t))^{.d/b} - \xi_2(\mathcal{W}(t)\boldsymbol{x}(t) - \boldsymbol{l}(t))) \\ \boldsymbol{U}\dot{\boldsymbol{x}}(t) + \dot{\boldsymbol{s}}^{.2}(t) \\ = \boldsymbol{\Phi}(-\xi_1(\boldsymbol{U}\boldsymbol{x}(t) + \boldsymbol{s}^{.2}(t) - \boldsymbol{c})^{.d/b} - \xi_2(\boldsymbol{U}\boldsymbol{x}(t) + \boldsymbol{s}^{.2}(t) - \boldsymbol{c})). \end{cases}$$

Substituting Eq. (12) and Eq. (13) into the above equations, it can be obtained that

$$\begin{cases} \boldsymbol{\mathcal{W}}(t)(\dot{\boldsymbol{x}}(t)-\dot{\boldsymbol{x}}^*(t))+\dot{\boldsymbol{\mathcal{W}}}(t)(\boldsymbol{x}(t)-\boldsymbol{x}^*(t)) \\ =\boldsymbol{\Phi}(-\xi_1(\boldsymbol{\mathcal{W}}(t)(\boldsymbol{x}(t)-\boldsymbol{x}^*(t)))^{.d/b}-\xi_2(\boldsymbol{\mathcal{W}}(t)(\boldsymbol{x}(t)-\boldsymbol{x}^*(t)))) \\ \boldsymbol{U}(\dot{\boldsymbol{x}}(t)-\dot{\boldsymbol{x}}^*(t))+\dot{\boldsymbol{s}}^{.2}(t)-\dot{\boldsymbol{s}}^{*.2}(t) \\ =\boldsymbol{\Phi}(-\xi_1(\boldsymbol{U}(\boldsymbol{x}(t)-\boldsymbol{x}^*(t))+\boldsymbol{s}^{.2}(t)-\boldsymbol{s}^{*.2}(t))^{.d/b} \\ \quad -\xi_2(\boldsymbol{U}\,(\boldsymbol{x}(t)-\boldsymbol{x}^*(t))+\boldsymbol{s}^{.2}(t)-\boldsymbol{s}^{*.2}(t))) \end{cases} \tag{14}$$

which can be further converted into

$$\begin{aligned} &\begin{bmatrix} \boldsymbol{\mathcal{W}}(t) & \boldsymbol{0} \\ \boldsymbol{U} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}}(t)-\dot{\boldsymbol{x}}^*(t) \\ \dot{\boldsymbol{s}}^{.2}(t)-\dot{\boldsymbol{s}}^{*.2}(t) \end{bmatrix} + \begin{bmatrix} \dot{\boldsymbol{\mathcal{W}}}(t) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t)-\boldsymbol{x}^*(t) \\ \boldsymbol{s}^{.2}(t)-\boldsymbol{s}^{*.2}(t) \end{bmatrix} \\ &= \boldsymbol{\Phi}\left(-\xi_1\left(\begin{bmatrix} \boldsymbol{\mathcal{W}}(t) & \boldsymbol{0} \\ \boldsymbol{U} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t)-\boldsymbol{x}^*(t) \\ \boldsymbol{s}^{.2}(t)-\boldsymbol{s}^{*.2}(t) \end{bmatrix}\right)^{.d/b}\right. \\ &\quad \left. -\xi_2\left(\begin{bmatrix} \boldsymbol{\mathcal{W}}(t) & \boldsymbol{0} \\ \boldsymbol{U} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t)-\boldsymbol{x}^*(t) \\ \boldsymbol{s}^{.2}(t)-\boldsymbol{s}^{*.2}(t) \end{bmatrix}\right)\right) \end{aligned} \tag{15}$$

where $\dot{\boldsymbol{s}}^{*.2}(t)$ and $\dot{\boldsymbol{x}}^*(t)$ are the time-derivative of $\boldsymbol{s}^{*.2}(t)$ and $\boldsymbol{x}^*(t)$, respectively. $\boldsymbol{I} \in R^{2n\times 2n}$ is a identity matrix.

Defining

$$\hat{\boldsymbol{e}}(t) = \begin{bmatrix} \boldsymbol{\mathcal{W}}(t) & \boldsymbol{0} \\ \boldsymbol{U} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t)-\boldsymbol{x}^*(t) \\ \boldsymbol{s}^{.2}(t)-\boldsymbol{s}^{*.2}(t) \end{bmatrix} \in R^{2n+m} \tag{16}$$

Eq. (15) can be reformulated as

$$\dot{\hat{\boldsymbol{e}}}(t) = \boldsymbol{\Phi}\left(-\xi_1\hat{\boldsymbol{e}}^{.d/b}(t)-\xi_2\hat{\boldsymbol{e}}(t)\right) \tag{17}$$

where $\dot{\hat{\boldsymbol{e}}}(t)$ is the time-derivative of $\hat{\boldsymbol{e}}(t)$. Then, the j-th element (with $j \in \{1, 2, \cdots, 2n+m\}$) in $\hat{\boldsymbol{e}}(t)$ can be given as

$$\dot{\hat{e}}_j(t) = \varphi(\hat{e}_j(t)) \tag{18a}$$

$$\varphi(\hat{e}_j(t)) = \phi\left(-\xi_1\hat{e}_j^{d/b}(t)-\xi_2\hat{e}_j(t)\right) \tag{18b}$$

where $\phi(\cdot)$ is an element of $\boldsymbol{\Phi}(\cdot)$, which is a monotone increasing odd function.

We can be obtained that the equilibrium state of the dynamic equation (18a) is $\hat{e}_j = 0$. Then, we defined the Lyapunov function as

$$v\left(\hat{e}_j\right) = \frac{1}{2}\hat{e}_j^2(t)$$

which is a positive-definite function. Subsequently, the time-derivative of $v\left(\hat{e}_j\right)$ is

$$\dot{v}\left(\hat{e}_j\right) = \frac{\mathrm{d}\,v\left(\hat{e}_j\right)}{\mathrm{d}\,t} = \hat{e}_j(t)\dot{\hat{e}}_j(t).$$

Substituting Eq. (18a) into the above equation, $\dot{v}\left(\hat{e}_j\right)$ can be reformulated as

$$\dot{v}\left(\hat{e}_j\right) = \hat{e}_j(t)\varphi(\hat{e}_j(t)). \tag{19}$$

If $\phi(\cdot)$ is a monotone increasing odd function, then

$$\varphi(\hat{e}_j(t)) \begin{cases} >0, & \hat{e}_j<0 \\ =0, & \hat{e}_j=0 \\ <0, & \hat{e}_j>0 \end{cases}$$

Thus, we can obtained that

$$\dot{v}\left(\hat{e}_j,t\right)=\hat{e}_j(t)\varphi(\hat{e}_j(t))\begin{cases}=0, & \hat{e}_j=0\\ <0, & \hat{e}_j\neq 0\end{cases}$$

which demonstrates $\dot{v}\left(\hat{e}_j\right)$ is a negative-definite function.

Then, according to Lyapunov stability theory [34], it can be concluded that dynamical equation (18a) is asymptotically stable in the equilibrium state $\hat{e}_j=0$. Since $\hat{\boldsymbol{e}}(t)=\{\hat{e}_j(t)\}$, it's further proved that $\hat{\boldsymbol{e}}(t)=\boldsymbol{0}$ is the equilibrium state of dynamical equation (17) and is asymptotically stable.

Considering the previous definition of $\hat{\boldsymbol{e}}(t)$, the equilibrium state $\hat{\boldsymbol{e}}=\boldsymbol{0}$ can be expanded as

$$\hat{\boldsymbol{e}}(t)=\begin{bmatrix}\mathcal{W}(t)\left(\boldsymbol{x}(t)-\boldsymbol{x}^*(t)\right)\\ \boldsymbol{U}\left(\boldsymbol{x}(t)-\boldsymbol{x}^*(t)\right)+\boldsymbol{s}^{\cdot 2}(t)-\boldsymbol{s}^{*\cdot 2}(t)\end{bmatrix}=\begin{bmatrix}0\\0\end{bmatrix}$$

which proves that the $\boldsymbol{x}(t)$ asymptotes to the $\boldsymbol{x}^*(t)$ at the equilibrium state $\hat{\boldsymbol{e}}=\boldsymbol{0}$. That means the vector-valued $\boldsymbol{x}(t)$ of FCNN model (11) can globally converge to the theoretical solution $\boldsymbol{x}^*(t)$ of TLEIC problem (1). The proof of $Theorem$ 1 is completed. □

4.2 Convergence Analysis

In this subsection, finite-time convergence of FCNN model (11) without the specific activation function is analyzed. Meanwhile, the convergent time of FCNN model (11) under linear activation function is calculated.

Theorem 2 *Given time-varying coefficient matrix $\mathcal{W}(t)$, time-varying vector $\boldsymbol{l}(t)$ and the bounded vector constraints $\boldsymbol{c}$, if $\boldsymbol{\Phi}(\cdot)$ is a monotone increasing odd function, then the vector-valued $\boldsymbol{x}(t)$ of FCNN model (11) with any initial state $\boldsymbol{x}(0)$ can converge to the theoretical solution $\boldsymbol{x}^*$ of TLEIC problem (1) in finite time. Specifically, when the linear activation functions is used, the convergent time of FCNN model (11) is $t_{cv}=\frac{b}{\xi_2(b-d)}\ln\frac{\xi_2\hat{e}_j(0)^{(b-d)/b}+\xi_1}{\xi_1}$.*

Proof Finite-time convergence of the general monotone increasing activation functions $\boldsymbol{\Phi}(\cdot)$ has been proved in detail in [35], which is omitted here.

Then, we consider the activation function $\boldsymbol{\Phi}(\cdot)$ is a linear function, i.e., $\boldsymbol{\Phi}(\hat{\boldsymbol{e}}(t))=\hat{\boldsymbol{e}}(t)$, Eq. (18a) can be reformulated as

$$\dot{\hat{e}}_j(t)=-\xi_1\hat{e}_j^{d/b}(t)-\xi_2\hat{e}_j(t),\ \ j\in\{1,2,\cdots,2n+m\}. \tag{20}$$

Dividing both sides of Eq. (20) by $\hat{e}_j^{d/b}(t)$, it can be obtained that

$$\hat{e}_j^{-d/b}(t)\frac{\mathrm{d}\hat{e}_j}{\mathrm{d}t}+\xi_2\hat{e}_j^{1-d/b}(t)=-\xi_1. \tag{21}$$

Let $\hat{e}_j^{1-d/b}(t)=f(t)$, then

$$\frac{\mathrm{d}f(t)}{\mathrm{d}t}=\frac{b-d}{b}\hat{e}_j^{-d/b}(t)\frac{\mathrm{d}\hat{e}_j(t)}{\mathrm{d}t}.$$

Eq. (21) can be reformulated into

$$\frac{\mathrm{d}f(t)}{\mathrm{d}t}+\frac{b-d}{b}\xi_2 f(t)=-\frac{b-d}{b}\xi_1. \tag{22}$$

At this point, Eq. (22) is a first-order linear differential equation, then its general solution is

$$\begin{aligned} f(t) &= \mathrm{e}^{-\int_0^t \frac{b-d}{b}\xi_2\, \mathrm{d}t}\left(\int_0^t -\frac{b-d}{b}\xi_1 \mathrm{e}^{\int_0^t \frac{m-n}{m}\xi_2\, \mathrm{d}t}\, \mathrm{d}t + \mathcal{A}\right) \\ &= \mathrm{e}^{-\int_0^t \frac{b-d}{b}\xi_2 \mathrm{d}t}\left(\int_0^t -\frac{b-d}{b}\xi_1 \mathrm{e}^{\frac{b-d}{b}\xi_2 t}\, \mathrm{d}t + \mathcal{A}\right) \end{aligned} \tag{23}$$

where $\mathcal{A}$ is an arbitrary constant.

Since $t = 0$, $\mathcal{A} = f(0)$, then Eq. (23) can further expressed as

$$\begin{aligned} f(t) &= \mathrm{e}^{-\frac{b-d}{b}\xi_2 t}\left(-\frac{b-d}{b}\xi_1 \frac{b}{(b-d)\xi_2}\mathrm{e}^{\frac{b-d}{b}\xi_2 t}\Big|_0^t + f(0)\right) \\ &= -\frac{\xi_1}{\xi_2} + \frac{\xi_1}{\xi_2}\mathrm{e}^{-\frac{b-d}{b}\xi_2 t} + f(0)\mathrm{e}^{-\frac{b-d}{b}\xi_2 t}. \end{aligned} \tag{24}$$

When $\hat{e}_j(t) = 0$, $f(t) = \hat{e}_j^{1-d/b}(t) = 0$ and $t = t_{cv}$. Then, Eq. (24) can be converted into

$$\left(\frac{\xi_1}{\xi_2} + f(0)\right)\mathrm{e}^{-\frac{b-d}{b}\xi_2 t_{cv}} = \frac{\xi_1}{\xi_2}\,. \tag{25}$$

According to Eq. (25), the convergent time t_{cv} from any initial state $\hat{e}_j(0)$ to equilibrium state $\hat{e}_j = 0$ is

$$\begin{aligned} t_{cv} &= \frac{b}{\xi_2(b-d)}\ln\frac{\xi_2 f(0) + \xi_1}{\xi_1} \\ &= \frac{b}{\xi_2(b-d)}\ln\frac{\xi_2\hat{e}_j(0)^{(b-d)/b} + \xi_1}{\xi_1}\,. \end{aligned} \tag{26}$$

It is worth noting that $\hat{e}_j(0)$ is the j-th element of $\hat{\boldsymbol{e}}(0)$. Substituting Eq. (12) into Eq. (16), we can obtained that

$$\begin{aligned} \hat{\boldsymbol{e}}(t) &= \begin{bmatrix} \boldsymbol{\mathcal{W}}(t)\boldsymbol{x}(t) - \boldsymbol{\mathcal{W}}(t)\boldsymbol{x}^*(t) \\ \boldsymbol{U}\boldsymbol{x}(t) + \boldsymbol{s}^{\cdot 2}(t) - \left(\boldsymbol{U}\boldsymbol{x}^*(t) + \boldsymbol{s}^{*.2}(t)\right) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\mathcal{W}}(t)\boldsymbol{x}(t) - \boldsymbol{l}(t) \\ \boldsymbol{U}\boldsymbol{x}(t) + \boldsymbol{s}^{\cdot 2}(t) - \boldsymbol{c} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\mathcal{W}}(t) & \boldsymbol{0} \\ \boldsymbol{U} & \boldsymbol{V}(t) \end{bmatrix}\begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{s}(t) \end{bmatrix} - \begin{bmatrix} \boldsymbol{l}(t) \\ \boldsymbol{c} \end{bmatrix} \\ &= \boldsymbol{\mathcal{K}}(t)\boldsymbol{z}(t) - \boldsymbol{q}(t). \end{aligned}$$

Hence, $\hat{\boldsymbol{e}}(0) = \boldsymbol{\mathcal{K}}(0)\boldsymbol{z}(0) - \boldsymbol{q}(0)$. The proof of *Theorem* 2 is completed. □

Remark 3 In the convergence analysis, finite-time convergence of the general monotone increasing odd functions $\Phi(\cdot)$ has been discussed in [35]. Thus, in order to simplify the process, we only give the details of the finite-time convergence analysis of FCNN model (11) under linear activation function. Although the process of analysis is similar to the theoretical analysis presented in [30–32], this paper focuses on the solution of the TLEIC problem, and the corresponding solution model, i.e., FCNN model (11), is quite different from the solution model of time-varying matrix inversion, time-varying matrix square root and time-varying Sylvester matrix equation in [30–32].

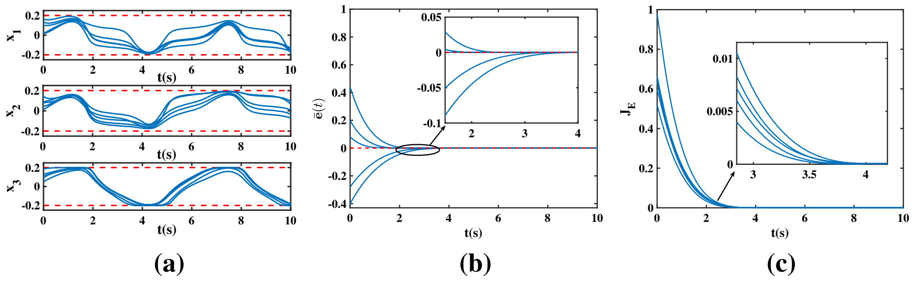

Fig. 2 Numerical simulation results using the proposed FCNN model (11) with $\xi_1 = 1$ and $\xi_2 = 1$ under linear activation function for TLEIC problem (1) solving. **a** State trajectories of $\boldsymbol{x}(t)$, **b** State trajectories of $\bar{\boldsymbol{e}}(t)$, **c** Residual errors J_E of FCNN model (11)

5 Numerical Verification of FCNN Model

In this section, the feasibility of the FCNN model (11) for TLEIC problem (1) is substantiated by numerically simulated. In addition, comparative simulation results with the ZNN model with various activation functions further verify the superiority of FCNN model (11).

5.1 Numerical Verification

In this numerical simulation, the specific forms of $\boldsymbol{\mathcal{W}}(t)$, $\boldsymbol{l}(t)$ and the constraints of $\boldsymbol{x}(t)$ (i.e., $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$) in TLEIC problem (1) are considered as

$$\boldsymbol{\mathcal{W}}(t) = \begin{bmatrix} 2+\sin(2t) \\ 2+\cos(t) \\ 2\cos(2t)-\sin(2t)+5 \end{bmatrix}^{\mathrm{T}} \in R^{1\times 3},$$

$$\boldsymbol{l}(t) = \cos(t)+\sin(t) \in R\ ,$$

$$\boldsymbol{x}_1 = \begin{bmatrix} -0.2 \\ -0.2 \\ -0.2 \end{bmatrix} \in R^3,\ \boldsymbol{x}_2 = \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \end{bmatrix} \in R^3.$$

Based on the above definition, FCNN model (11) is simulated and investigated with linear activation function and different values of ξ_1. In addition, design parameters $\xi_2 = 1$ and $d/b = 5/7$ are fixed in this simulation. The initial state $\boldsymbol{x}(0) \in R^3$ is set as "$0.2\times\mathrm{rand}(3,1)$". The relevant simulation results are displayed in Fig. 2-4.

Figure 2 shows the simulation results using FCNN model (11) with linear activation function and $\xi_1 = 1$, where the residual errors of FCNN model is defined as $J_E = \|\boldsymbol{\mathcal{K}}(t)\boldsymbol{z}(t) - \boldsymbol{q}(t)\|_2$ and the $\|\cdot\|_2$ denotes the two-norms. The state trajectories of $\boldsymbol{x}(t)$ are shown in Fig. 2awith five different initial states $\boldsymbol{x}(0)$. From this figure, we can see that all the state trajectories of $\boldsymbol{x}(t)$ do not exceed $[-0.2, 0.2]$. To further verify that the $\boldsymbol{x}(t)$ presented in Fig. 2a satisfy the linear equations of TLEIC problem (1), the state trajectories of $\bar{\boldsymbol{e}}(t) = \boldsymbol{\mathcal{W}}(t)\boldsymbol{x}(t) - \boldsymbol{l}(t)$ are given in Fig. 2b. As observed in Fig. 2b, all states of $\bar{\boldsymbol{e}}$ converges to zero in any initial state. In Fig. 2c, all the residual errors J_E of FCNN model (11) converge to zero within the range of 3.5 - 4 s. Combining with the analysis in Sect. 4, it can be concluded that the $\boldsymbol{x}(t)$ presented in Fig. 2a simultaneously satisfies time-varying linear equations (i.e., $\boldsymbol{\mathcal{W}}(t)\boldsymbol{x}(t) = \boldsymbol{l}(t)$) and the inequality constraints (i.e., $\boldsymbol{x}_1 \leq \boldsymbol{x}(t) \leq \boldsymbol{x}_2$). In other words, $\boldsymbol{x}(t)$ presented in Fig. 2a can converge to the theoretical solution $\boldsymbol{x}^*(t)$ of TLEIC problem

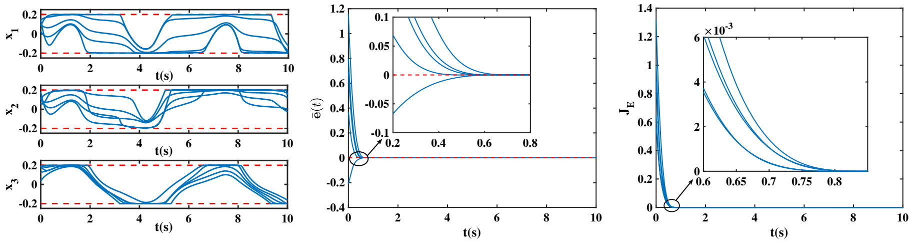

Fig. 3 Numerical simulation results using the proposed FCNN model (11) with $\xi_1 = 5$ and $\xi_2 = 1$ under linear activation function for TLEIC problem (1) solving. **a** State trajectories of $\boldsymbol{x}(t)$, **b** State trajectories of $\bar{\boldsymbol{e}}(t)$, **c** Residual errors J_E of FCNN model (11)

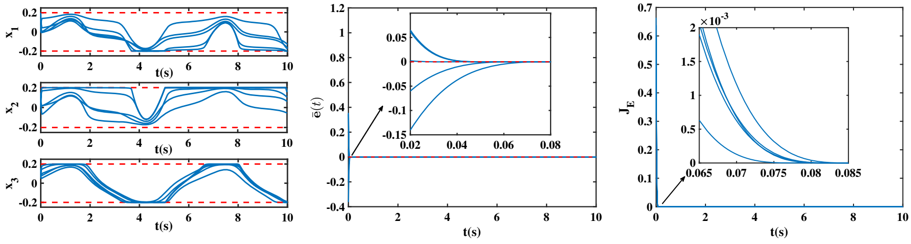

Fig. 4 Numerical simulation results using the proposed FCNN model (11) with $\xi_1 = 50$ and $\xi_2 = 1$ under linear activation function for TLEIC problem (1) solving. **a** State trajectories of $\boldsymbol{x}(t)$, **b** State trajectories of $\bar{\boldsymbol{e}}(t)$, **c** Residual errors J_E of FCNN model (11)

(1) within finite time. Therefore, the feasibility and finite-time convergence of FCNN model (11) with linear activation function and $\xi_1 = 1$ for TLEIC problem (1) is substantiated.

In order to verify the influence of different values of ξ_1 on the convergence performance of FCNN model (11), numerical simulations are carried out for $\xi_1 = 5$ and $\xi_1 = 50$, respectively. The relevant simulation results are displayed in Figs. 3 and 4.

As shown in Figs. 3a and 4a, all the state trajectories of $\boldsymbol{x}(t)$ obtained by FCNN model (11) with linear activation function are within $[-0.2, 0.2]$. The state trajectories of $\bar{\boldsymbol{e}}(t)$ in Figs. 3b and 4b both converge to zero. Besides, we can observe that the residual errors J_E of FCNN model (11) with $\xi_1 = 5$ converge to zero within the range of 0.75 - 0.85 s in Fig. 3c. Similarly, in Fig. 4c, the residual errors of FCNN model (11) with $\xi_1 = 50$ converge to zero within the range of 0.075 - 0.085 s. This indicates that the $\boldsymbol{x}(t)$ presented in Figs. 3a and 4a simultaneously hold $\boldsymbol{\mathcal{W}}(t)\boldsymbol{x}(t) = \boldsymbol{l}(t)$ and $\boldsymbol{x}_1 \leq \boldsymbol{x}(t) \leq \boldsymbol{x}_2$ true. Thus, the feasibility and finite-time convergence of FCNN model (11) with linear activation function for TLEIC problem (1) is substantiated again. Furthermore, the comparison of Figs. 2c, 3c and 4c shows that as the value of ξ_1 increases, the convergent rate of residual error J_E can be improved and the convergent time can be shortened, that is, increasing ξ_1 enhances the convergence performance of the proposed FCNN model (11). Note that the value of ξ_1 should be selected appropriately for experimental or simulation purpose.

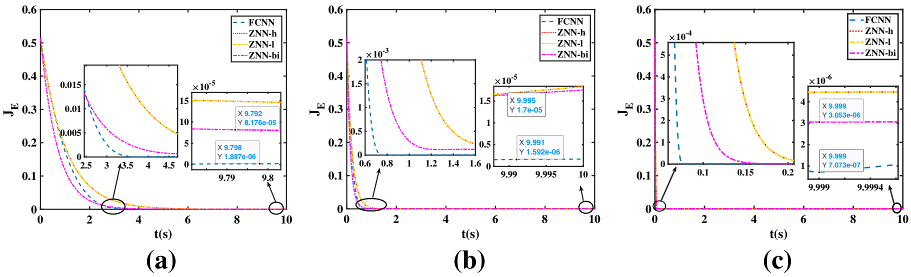

Fig. 5 Comparative results of residual errors $J_E = \|\boldsymbol{\mathcal{K}}(t)\boldsymbol{z}(t) - \boldsymbol{q}(t)\|_2$ using the proposed FCNN model (11) and ZNN model for solving TLEIC problem (1) with different values of ξ_1 and β_1 when $\xi_2 = 1$ is fixed. **a** $\xi_1 = 1$ and $\beta = 1$, **b** $\xi_1 = 5$ and $\beta = 5$, **c** $\xi_1 = 50$ and $\beta = 50$

5.2 Comparative Verification

For illustrate the superiority of the proposed FCNN model (11) for TLEIC problem (1), comparisons between ZNN model are stated in this subsection.

For comparison convenience, the design formula of ZNN is shown as

$$\dot{\boldsymbol{e}}(t) = -\beta\boldsymbol{\Phi}\left(\boldsymbol{e}(t)\right)$$

and the following three activation functions is used to ZNN.

1) The hyperbolic-sine activation function (ZNN-h):

$$\phi(e_i(t)) = \frac{\exp e_i(t) - \exp\left(-e_i(t)\right)}{2}.$$

2) The linear activation function (ZNN-l):

$$\phi(e_i(t)) = e_i(t).$$

3) The bisigmoid activation function (ZNN-bi):

$$\phi(e_i(t)) = \frac{1 - \exp(-\alpha e_i(t))}{1 + \exp(-\alpha e_i(t))}, \text{ with } \alpha > 2.$$

Note that $e_i(t)$ as the i−th element of $\boldsymbol{e}(t)$. The relevant simulation results are displayed in Fig. 5.

Figure 5 presents the residual errors $J_E = \|\boldsymbol{\mathcal{K}}(t)\boldsymbol{z}(t) - \boldsymbol{q}(t)\|_2$ of FCNN model (11) with linear activation function and ZNN model with above three activation functions (i.e., ZNN-h, ZNN-l, ZNN-bi). In addition, the value of ξ_2 and d/b are the same as defined in Sect. 5.1. The adjustable parameter ξ_1 and β are simultaneously set as 1, 5, 50 in Fig. 5a, b, c, respectively. As observed in Fig. 5a, b, c, the residual errors of FCNN model (11) with linear activation function converge to zero within 3.5 s, 0.75 s and 0.075 s, respectively, while at the same time, ZNN-h, ZNN-l and ZNN-bi converge slower. This comparison results indicate that the convergent time of FCNN model (11) is shorten than that of ZNN model. Furthermore, Table 1 shows the convergent accuracy of FCNN model (11) and ZNN model when residual errors converge to zero, which can be seen that the convergent accuracy of FCNN model (11) is better than that of ZNN-h, ZNN-l and ZNN-bi. Hence, the excellent convergence performance of FCNN model (11) with linear activation function as compared with ZNN-h, ZNN-l and ZNN-bi for solving TLEIC problem (1) is verified.

Table 1 Comparisons of convergent accuracy using the proposed FCNN model (11) with the ZNN model for TLEIC problem (1) solving

Neural network solver	Activation function	Value of ξ_1	Convergent accuracy
The proposed FCNN	Linear function	1	10^{-6}
		5	10^{-6}
		50	10^{-7}
ZNN	Hyperbolic sine function	1	10^{-4}
		5	10^{-5}
		50	10^{-6}
	Linear function	1	10^{-4}
		5	10^{-5}
		50	10^{-6}
	Bipolar sigmoid function	1	10^{-5}
		5	10^{-5}
		50	10^{-6}

6 Application of FCNN Model on Redundant Manipulators

6.1 Problem Description

In this subsection, we demonstrate the application of the proposed FCNN model to trajectory planning of redundant manipulators. The corresponding inverse-kinematics problem is shown as follows:

$$\begin{cases} \boldsymbol{J}(\boldsymbol{\theta}(t))\dot{\boldsymbol{\theta}}(t) = \dot{\boldsymbol{r}}_d(t) + f_b(\boldsymbol{r}_d - \boldsymbol{f}(\boldsymbol{\theta})) \\ \boldsymbol{\theta}^- \leq \boldsymbol{\theta}(t) \leq \boldsymbol{\theta}^+ \\ \dot{\boldsymbol{\theta}}^- \leq \dot{\boldsymbol{\theta}}(t) \leq \dot{\boldsymbol{\theta}}^+ \end{cases} \tag{27}$$

where $\boldsymbol{J}(\boldsymbol{\theta}(t)) \in R^{m\times n}$ is the Jacobian matrix. $\boldsymbol{\theta}(t) \in R^n$ is a joint-angle vector. $\dot{\boldsymbol{\theta}}(t) \in R^n$ is a joint-velocity vector. Moreover, $\boldsymbol{\theta}^\pm \in R^n$ and $\dot{\boldsymbol{\theta}}^\pm \in R^n$ represents the constraints of $\boldsymbol{\theta}(t)$ and $\dot{\boldsymbol{\theta}}(t)$, respectively. $\boldsymbol{r}_d \in R^m$ and $\boldsymbol{f}(\boldsymbol{\theta}) \in R^m$ are the expected trajectory and the actual trajectory of the end effector, respectively. $f_b(\boldsymbol{r}_d - \boldsymbol{f}(\boldsymbol{\theta}))$ is the feedback on the position, in which $f_b \in R$ (with $f_b > 0$) represents the feedback gain of the position.

Since the scheme (27) is established based on the velocity layer, the joint-angle constrains (i.e., $\boldsymbol{\theta}^- \leq \boldsymbol{\theta}(t) \leq \boldsymbol{\theta}^+$) need to be converted into the following joint-velocity constrains [36]:

$$\varrho\left(\boldsymbol{\theta}^- - \boldsymbol{\theta}(t)\right) \leq \dot{\boldsymbol{\theta}} \leq \varrho\left(\boldsymbol{\theta}^+ - \boldsymbol{\theta}(t)\right)$$

where large value of parameter $\varrho > 0$ may cause quick joint deceleration when a joint approaches its limits [5]. Then, by combining the above formula with $\dot{\boldsymbol{\theta}}^- \leq \dot{\boldsymbol{\theta}}(t) \leq \dot{\boldsymbol{\theta}}^+$, the inequality constraints on $\dot{\boldsymbol{\theta}}(t)$ can be transformed into

$$\boldsymbol{\vartheta}^- \leq \dot{\boldsymbol{\theta}} \leq \boldsymbol{\vartheta}^+$$

where the i-th element (with $i = 1, 2, \cdots, n$) in $\boldsymbol{\vartheta}^-$ and $\boldsymbol{\vartheta}^+$ can be expressed as

$$\begin{aligned} \vartheta_i^- &= \max\left\{\dot{\theta}_i^-, \varrho\left(\theta_i^- - \theta_i\right)\right\}, \\ \vartheta_i^+ &= \min\left\{\dot{\theta}_i^+, \varrho\left(\theta_i^+ - \theta_i\right)\right\}. \end{aligned}$$

Furthermore, scheme (27) can be re-written as

$$\begin{cases} \boldsymbol{J}(\boldsymbol{\theta}(t))\dot{\boldsymbol{\theta}}(t) = \dot{\boldsymbol{r}}_d(t) + f_b(\boldsymbol{r}_d - \boldsymbol{f}(\boldsymbol{\theta})) \\ \boldsymbol{\vartheta}^- \le \dot{\boldsymbol{\theta}} \le \boldsymbol{\vartheta}^+ \end{cases}. \tag{28}$$

Considering the TLEIC problem (1) described in Sect. 2, which is similar to the form of scheme (28). That is, $\boldsymbol{\theta}(t)$ corresponds to $\boldsymbol{x}(t)$, $\boldsymbol{J}(\boldsymbol{\theta}(t))$ corresponds to $\boldsymbol{\mathcal{W}}(t)$, $\dot{\boldsymbol{r}}_d(t) + f_b(\boldsymbol{r}_d - \boldsymbol{f}(\boldsymbol{\theta}))$ corresponds to $\boldsymbol{l}(t)$, $\boldsymbol{\vartheta}^-$ and $\boldsymbol{\vartheta}^+$ correspond to $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ respectively. Hence, the FCNN model (11) proposed in this paper can be used to solve scheme (28) with

$$\begin{aligned} \boldsymbol{G}(t) &= \begin{bmatrix} \boldsymbol{J}(\boldsymbol{\theta}(t)) & \boldsymbol{0} \\ \boldsymbol{U} & 2\boldsymbol{V}(t) \end{bmatrix}, \ \boldsymbol{H}(t) = \begin{bmatrix} \dot{\boldsymbol{J}}(\boldsymbol{\theta}(t)) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \\ \boldsymbol{\mathcal{K}}(t) &= \begin{bmatrix} \boldsymbol{J}(\boldsymbol{\theta}(t)) & \boldsymbol{0} \\ \boldsymbol{U} & \boldsymbol{V}(t) \end{bmatrix}, \quad \boldsymbol{z}(t) = \begin{bmatrix} \dot{\boldsymbol{\theta}}(t) \\ \boldsymbol{s}(t) \end{bmatrix}, \\ \boldsymbol{q}(t) &= \begin{bmatrix} \dot{\boldsymbol{r}}_d(t) + f_b(\boldsymbol{r}_d - \boldsymbol{f}(\boldsymbol{\theta})) \\ \boldsymbol{c} \end{bmatrix}. \end{aligned}$$

To further illustrate the practical feasibility of the FCNN model (11), FCNN model (11) will be applied to the six-degree-of-freedom (six-DOF) planar manipulator and PUMA560 redundant manipulator.

6.2 Six-DOF Planar Manipulator

In this simulation, the proposed FCNN model (11) with $\xi_1 = 100$ and linear activation function is applied to the circular trajectory planning of the six-DOF planar manipulator with joint constraints. Other adjustable parameters $\xi_2 = 1, d/b = 5/7$ and $f_b = 1$ in FCNN model (11). The initial joint-angles is set to $\boldsymbol{\theta}(0) = [\pi/4; \pi/4; \pi/4; \pi/4; \pi/4; \pi/4]$. In addition, the specific upper and lower bounds of $\boldsymbol{\theta}(t)$ and $\dot{\boldsymbol{\theta}}(t)$ (i.e., $\theta_i^{\pm}$ and $\dot{\theta}_i^{\pm}$) are given in Table 2. The entire period of motion in this simulation is 10 s. The relevant simulation results are displayed in Fig. 6.

Figure 6a presents the motion trajectory of the six-DOF planar manipulator tracking the circular trajectory. Figure. 6b shows the desired trajectory and the actual trajectory of the end-effector. As we can see from Fig. 6a, b, the actual trajectory of the end-effector coincides with the desired circular trajectory. In Fig. 6c, the tracking errors (i.e., $\varepsilon \boldsymbol{X}$, $\varepsilon \boldsymbol{Y}$) of the end-effector of six-DOF planar manipulator is shown. It can be seen that the maximum tracking error is not more than 3×10^{-5} m. This illustrates that the end-effector of six-DOF planar manipulator can successfully track the desired circular trajectory with a small error. Furthermore, Fig. 6d, e show the state trajectories of $\boldsymbol{\theta}(t)$ and $\dot{\boldsymbol{\theta}}(t)$ of six-DOF planar manipulator. The maximum and minimum values of each $\boldsymbol{\theta}(t)$ (i.e., $\max\theta_i$ and $\min\theta_i$) and $\dot{\boldsymbol{\theta}}(t)$ (i.e., $\max\dot{\theta}_i$ and $\min\dot{\theta}_i$) are listed in Table 2. It can be seen that the maximum and minimum values of $\boldsymbol{\theta}(t)$ and $\dot{\boldsymbol{\theta}}(t)$ of the six-DOF planer manipulator are kept within the given joint constraints. In conclusion, these results verify the effectiveness of FCNN model (11) to the trajectory planning of the six-DOF planar manipulator with joint constraints. This further illustrates the capability of the proposed FCNN.

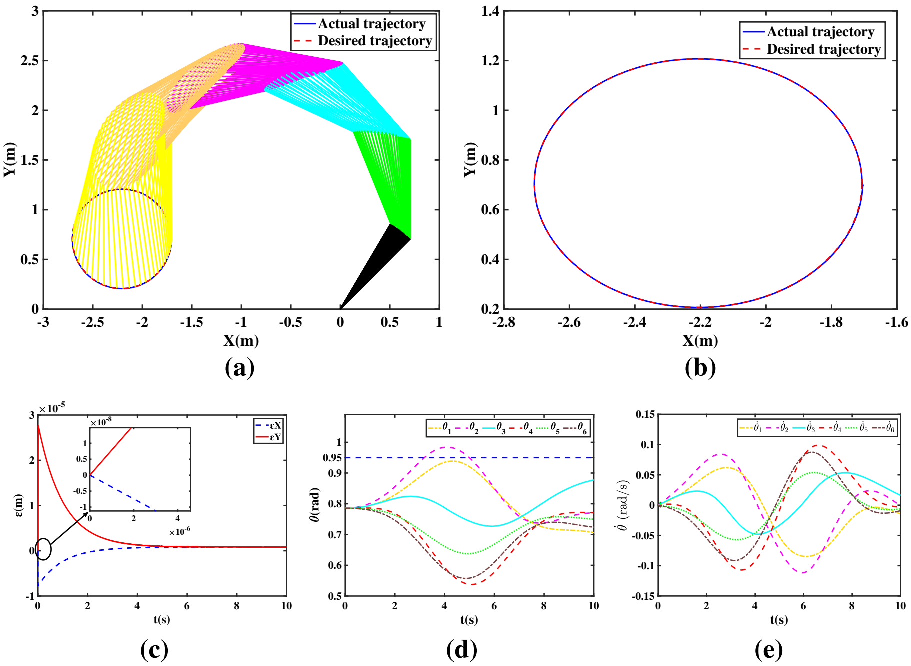

Fig. 6 Computational results of six-DOF planar manipulator tracking circular trajectory with joint-angle constrains and joint-velocity constrains by the FCNN model (11) with $\xi_1 = 100$ and $\xi_2 = 1$ under linear activation function. **a** Motion trajectory of the six-DOF planar manipulator, **b** End-effector trajectory, **c** End-effector tracking errors, **d** Trajectories of joint-angle, **e** Trajectories of joint-velocity

Table 2 Joint constrains used in the simulation of the six-DOF planar manipulator and the minimum and maximum of joint variables obtained by the FCNN model (11)

Joint	θ_i^+(rad)	θ_i^-(rad)	max θ_i	min θ_i	$\dot{\theta}_i^+$(rad/s)	$\dot{\theta}_i^-$(rad/s)	max $\dot{\theta}_i$	min $\dot{\theta}_i$
θ_1	0.95	-0.95	0.9391	0.7074	0.1	-0.1	0.0620	-0.0846
θ_2	$\pi/3$	$-\pi/3$	0.9845	0.7383	0.2	-0.2	0.0843	-0.1115
θ_3	$\pi/3$	$-\pi/3$	0.8763	0.7266	0.2	-0.2	0.0534	-0.0481
θ_4	$\pi/3$	$-\pi/3$	0.7861	0.5368	0.2	-0.2	0.0987	-0.1073
θ_5	$\pi/3$	$-\pi/3$	0.7854	0.6391	0.08	-0.08	0.0541	-0.0568
θ_6	$\pi/3$	$-\pi/3$	0.7854	0.5575	0.2	-0.2	0.0875	-0.0914

6.3 PUMA560 Redundant Manipulator

In this simulation, the proposed FCNN model (11) with $\xi_1 = 100$ and linear activation function is applied to the octagon trajectory planning of the PUMA560 redundant manipulator with joint constraints, where the length of the octagon is 1. The values of ξ_2, d/b and f_b are the same defined as Sect. 6.1. The initial joint-angles is set to $\boldsymbol{\theta}(0) = [\pi/4; 0; 0; -\pi/4; \pi/4; -\pi/4]$. In addition, the limitations of joint-angles and joint-velocities (i.e., $\theta_i^{\pm}$ and $\dot{\theta}_i^{\pm}$) are given in Table 3. Note that the entire period of motion in this simulation is 8 s. The relevant simulation results are displayed in Fig. 7.

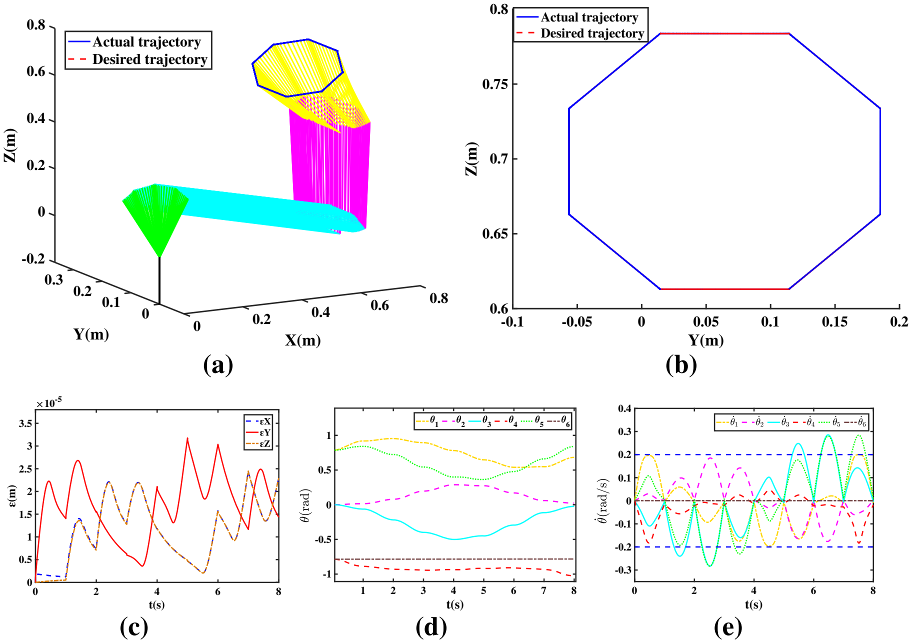

Fig. 7 Computational results of PUMA560 redundant manipulator tracking octagon trajectory with joint-angle constrains and joint-velocity constrains by the FCNN model (11) with $\xi_1 = 100$ and $\xi_2 = 1$ under linear activation function. **a** Motion trajectory of the PUMA560 redundant manipulator, **b** End-effector trajectory, **c** End-effector tracking errors, **d** Trajectories of joint-angle, **e** Trajectories of joint-velocity

Table 3 Joint limits used in the simulation of the PUMA560 redundant manipulator and the minimum and maximum of joint variables obtained by the FCNN model (11)

Joint	θ_i^+(rad)	θ_i^-(rad)	max θ_i	min θ_i	$\dot{\theta}_i^+$(rad/s)	$\dot{\theta}_i^-$(rad/s)	max $\dot{\theta}_i$	min $\dot{\theta}_i$
θ_1	2.151	− 2.512	0.9542	0.5392	0.2	− 0.2	0.1966	− 0.1966
θ_2	0.750	− 3.078	0.2881	0	0.3	− 0.3	0.1835	− 0.1853
θ_3	3.126	− 0.905	0	− 0.5006	0.3	− 0.3	0.2854	− 0.2834
θ_4	2.537	− 1.812	− 0.7854	− 1.0254	0.3	− 0.3	0.0477	− 0.1833
θ_5	1.651	− 1.651	0.8412	0.3638	0.3	− 0.3	0.2842	− 0.2827
θ_6	4.076	− 4.158	− 0.7825	− 0.7854	0.3	− 0.3	1e−03	3.58e−04

Figure 7a presents the movement trajectory of the PUMA560 redundant manipulator. Figure 7b shows the actual motion trajectory and the expected octagonal trajectory of the end-effector. As can be seen from these two figures, the actual trajectory of the end-effector coincides with the desired octagonal trajectory. As observed in Fig. 7c, the maximum tracking error of the end-effector of PUMA560 redundant manipulator, i.e., $\max\{\varepsilon \boldsymbol{X}, \varepsilon \boldsymbol{Y}, \varepsilon \boldsymbol{Z}\}$, is less than 3.1×10^{-5} m, which further verifies that the end-effector can successfully track the expected octagonal trajectory with minimal error. Subsequently, Figs. 7d, e show the state trajectories of $\boldsymbol{\theta}(t)$ and $\dot{\boldsymbol{\theta}}(t)$ of PUMA560 redundant manipulator. The maximum and minimum values of each $\boldsymbol{\theta}(t)$ (i.e., $\max \theta_i$ and $\min \theta_i$) and $\dot{\boldsymbol{\theta}}(t)$ (i.e., $\max \dot{\theta}_i$ and $\min \dot{\theta}_i$) are listed in Table 3. It can be seen that the $\boldsymbol{\theta}(t)$ and $\dot{\boldsymbol{\theta}}(t)$ in the PUMA560 redundant

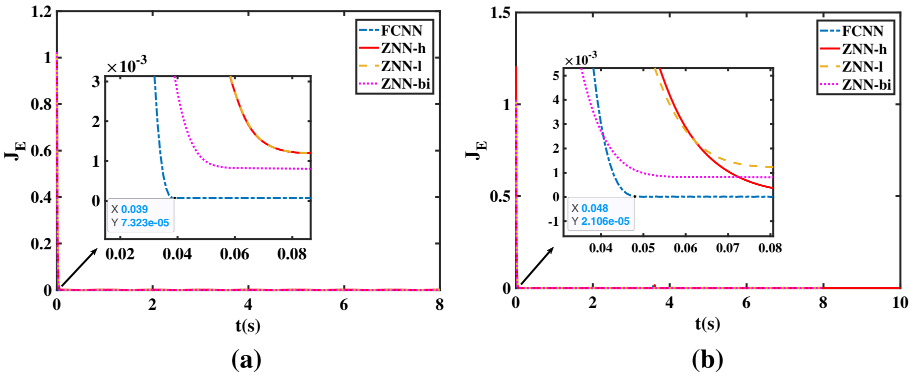

Fig. 8 Comparative results of residual errors $J_E = \|\mathcal{K}(t)z(t) - \boldsymbol{q}(t)\|_2$ using the proposed FCNN model (11) and ZNN model for trajectory planning of manipulator with joint limitations. **a** Residual errors in six-DOF planar redundant manipulator, **b** Residual errors in PUMA560 redundant manipulator

manipulator during movement are kept within the given joint constraints. In summary, these simulation results verify the effectiveness and practicability of FCNN model (11) for solving the trajectory planning problem of PUMA560 redundant manipulator with joint constraints.

To illustrate the excellent convergence performance of FCNN model (11), the residual errors $J_E = \|\mathcal{K}(t)z(t) - \boldsymbol{q}(t)\|_2$ using FCNN model (11) with linear activation function and ZNN model with three activation functions (i.e., ZNN-h, ZNN-l, ZNN-bi) are carried out in Fig. 8. Note that the ZNN-h, ZNN-l and ZNN-bi are defined in Sect. 5. In addition, the adjustable parameter of β is set as 100 in ZNN model. Figure 8a presents the residual errors J_E of FCNN model (11) and ZNN model for the circular trajectory planning of six-DOF planar manipulator with joint constraints is shown. Figure 8b shows the residual errors for the octagon trajectory planning of the PUMA560 redundant manipulator with joint constraints. From these two figures, it can be seen that the residual errors J_E of FCNN model (11) converge to zero within 0.04 s and 0.048 s, respectively, while the convergent rate of ZNN-h, ZNN-l and ZNN-bi is slower at $t = 0.04$ s and $t = 0.048$ s. These results illustrate that FCNN model (11) can solve the trajectory planning problem of manipulator with joint constraints in finite time, and the convergence performance of FCNN model (11) is better than that of ZNN-h, ZNN-l and ZNN-bi.

Above all, the conclusions obtained from the simulation results in Figs. 6, 7 and 8 can be summarized as follows:

1. The effectiveness and availability of FCNN model (11) for the trajectory planning of redundant manipulator with joint constraints is verified.
2. The convergence performance of FCNN model (11) with linear activation function is superior to ZNN model with hyperbolic activation function, linear activation function and bipolar sigmoid activation function for the trajectory planning of redundant manipulators with joint constraints.

7 Conclusion

For finding the solution of TLEIC, a FCNN model has been proposed in this paper. Specifically, by introducing a non-negative slack variable, the TLEIC is transformed into a class of time-varying linear equations. Then, the proposed FCNN model is used to solve the transformed equations. By using the Lyapunov stability theorem, the stability of the FCNN model (11) is proved concretely. Meanwhile, the convergent time of the FCNN model (11) with linear activation function is calculated, which proves its finite-time convergence. Compared with the ZNN model with three kinds of activation functions (i.e., ZNN-h, ZNN-l and ZNN-bi) in numerical simulation, the convergent speed, convergent time and convergent accuracy of the FCNN model (11) are significantly improved in solving TLEIC problem (1). Finally, the effectiveness and feasibility of the FCNN model (11) are further demonstrated through its application to two types of redundant manipulator with joint constraints. Further work may lie in the application of FCNN model (11) to a practical redundant manipulator, and the discrete-time form of the FCNN model (11) will be studied to solve the TLEIC.

Acknowledgements This study was partly supported by the National Natural Science Foundation of China (61803338, 61972357, 61672337), the Zhejiang Key R&D Program (Grant No. 2019C03135) and the Youth Foundation of Zhejiang University of Science and Technology (2021QN001, 2021QN046).

References

1. Li J, Zhang Y, Mao M (2020) Continuous and discrete zeroing neural network for different-level dynamic linear system with robot manipulator control. IEEE Trans Syst Man Cybern Syst 50(11):4633–4642. https://doi.org/10.1109/TSMC.2018.2856266
2. Zhang Y, Guo D (2015) Zhang functions and various models. Springer, Berlin
3. Zhao YB (2013) New and improved conditions for uniqueness of sparsest solutions of underdetermined linear systems. Appl Math Comput 224:58–73
4. Rump SM (2014) Improved componentwise verified error bounds for least squares problems and underdetermined linear systems. Numer Algorithms 66(2):309–322. https://doi.org/10.1007/s11075-013-9735-6
5. Chen D, Zhang Y (2015) A hybrid multi-objective scheme applied to redundant robot manipulators. IEEE Trans Autom Sci Eng 14(3):1337–1350
6. Pang LP, Spedicato E, Xia ZQ, Wang W (2007) A method for solving the system of linear equations and linear inequalities. Math Comput Model 46(5–6):823–836
7. Murav'eva OV, (2015) Consistency and inconsistency radii for solving systems of linear equations and inequalities. Comput Math Math Phys 55(3):366–377
8. Golikov AI, Evtushenko YG (2015) Regularization and normal solutions of systems of linear equations and inequalities. Proc Steklov Inst Math 289(S1):102–110
9. Li H, Luo J, Wang Q (2014) Solvability and feasibility of interval linear equations and inequalities. Linear Algebra Appl 463:78–94
10. Esmaeili H, Spedicato E (2004) Explicit ABS solution of a class of linear inequality systems and LP problems. Bull Iran Math Soc 30
11. Castillo E, Jubete F (2004) The Γ-algorithm and some applications. Int J Math Educ Sci Technol 35(3):369–389
12. Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci 79(8):2554–2558
13. 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
14. Liang Xue-Bin, Shiu Kit Tso (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
15. Cichocki A, Ramirez-Angulo J, Unbehauen R (1992) Architectures for analog VLSI implementation of neural networks for solving linear equations with inequality constraints
16. Guo D, Zhang Y (2015) ZNN for solving online time-varying linear matrix-vector inequality via equality conversion. Appl Math Comput 259:327–338. https://doi.org/10.1016/j.amc.2015.02.060
17. Shao S, Li H, Qin S, Li G, Luo C (2020) An inverse-free Zhang neural dynamic for time-varying convex optimization problems with equality and affine inequality constraints. Neurocomputing 412:152–166

18. Guo D, Zhang Y (2012) Novel recurrent neural network for time-varying problems solving [research Frontier]. IEEE Comput Intell Mag 7(4):61–65
19. Stanimirović PS, Srivastava S, Gupta DK (2018) From Zhang Neural Network to scaled hyperpower iterations. J Comput Appl Math 331:133–155. https://doi.org/10.1016/j.cam.2017.09.048
20. Guo D, Zhang Y (2014) Zhang neural network for online solution of time-varying linear matrix inequality aided with an equality conversion. IEEE Trans Neural Netw Learn Syst 25(2):370–382. https://doi.org/10.1109/TNNLS.2013.2275011
21. Zhang Y, Wang Y, Jin L, Mu B, Zheng H (2013) Different ZFs leading to various ZNN models illustrated via online solution of time-varying underdetermined systems of linear equations with robotic application. Adv Neural Netw - ISNN 2013. Springer, Berlin Heidelberg, pp 481–488
22. 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 Inf 15(4):2167–2176
23. Li S, Chen S, Liu B (2012) Accelerating a recurrent neural network to finite-time convergence for solving time-varying Sylvester equation by using a sign-bi-power activation function. Neural Process Lett 37(2):189–205
24. Guo D, Lin X (2020) Li-function activated Zhang neural network for online solution of time-varying linear matrix inequality. Neural Process Lett 52(1):713–726. https://doi.org/10.1007/s11063-020-10291-y
25. Xiao L, Zhang Z, Li S (2019) Solving time-varying system of nonlinear equations by finite-time recurrent neural networks with application to motion tracking of robot manipulators. IEEE Trans Syst Man Cybern Syst 49(11):2210–2220
26. Jin J, Zhao L, Li M, Yu F, Xi Z (2020) Improved zeroing neural networks for finite time solving nonlinear equations. Neural Comput Appl 32(9):4151–4160. https://doi.org/10.1007/s00521-019-04622-x
27. Shen Y, Miao P, Huang Y, Shen Y (2015) Finite-time stability and its application for solving time-varying sylvester equation by recurrent neural network. Neural Process Lett 42(3):763–784
28. Jin J, Gong J (2021) A noise-tolerant fast convergence ZNN for dynamic matrix inversion. Int J Comput Math. https://doi.org/10.1080/00207160.2021.1881498
29. Jin J (2021) An improved finite time convergence recurrent neural network with application to time-varying linear complex matrix equation solution. Neural Process Lett 53(1):777–786. https://doi.org/10.1007/s11063-021-10426-9
30. Xiao L (2016) A new design formula exploited for accelerating Zhang neural network and its application to time-varying matrix inversion. Theor Comput Sci 647:50–58. https://doi.org/10.1016/j.tcs.2016.07.024
31. Xiao L (2017a) Accelerating a recurrent neural network to finite-time convergence using a new design formula and its application to time-varying matrix square root. J Franklin Inst 354(13):5667–5677. https://doi.org/10.1016/j.jfranklin.2017.06.012
32. Xiao L (2017b) A finite-time recurrent neural network for solving online time-varying Sylvester matrix equation based on a new evolution formula. Nonlinear Dyn 90(3):1581–1591. https://doi.org/10.1007/s11071-017-3750-4
33. Li S, Li Y, Wang Z (2013) A class of finite-time dual neural networks for solving quadratic programming problems and its k-winners-take-all application. Neural Netw 39:27–39
34. Zhang Y, Yi C (2011) Zhang neural networks and neural-dynamic method. Nova Science Publishers Inc., Hauppauge
35. Kong Y, Jiang Y, Xia X (2020) Terminal recurrent neural networks for time-varying reciprocal solving with application to trajectory planning of redundant manipulators. IEEE Trans Syst Man Cybern Syst 1–13
36. Zhang Y, Zhang Z (2014) Repetitive motion planning and control of redundant robot manipulators. Springer, Berlin

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