

RESEARCH

Open Access



# Noise-tolerant continuous-time Zhang neural networks for time-varying Sylvester tensor equations

Sun Min<sup>1\*</sup> and Liu Jing<sup>2</sup>

\*Correspondence:

ziyouxiaodou@163.com

<sup>1</sup>School of Mathematics and Statistics, Zaozhuang University, Zaozhuang, P.R. China

Full list of author information is available at the end of the article

## Abstract

In this paper, to solve the time-varying Sylvester tensor equations (TVSTEs) with noise, we will design three noise-tolerant continuous-time Zhang neural networks (NTCTZNNs), termed NTCTZNN1, NTCTZNN2, NTCTZNN3, respectively. The most important characteristic of these neural networks is that they make full use of the time-derivative information of the TVSTEs' coefficients. Theoretical analyses show that no matter how large the unknown noise is, the residual error generated by NTCTZNN2 converges globally to zero. Meanwhile, as long as the design parameter is large enough, the residual errors generated by NTCTZNN1 and NTCTZNN3 can be arbitrarily small. For comparison, the gradient-based neural network (GNN) is also presented and analyzed to solve TVSTEs. Numerical examples and results demonstrate the efficacy and superiority of the proposed neural networks.

**Keywords:** Time-varying Sylvester tensor equations; Noise-tolerant continuous-time Zhang neural network; Gradient-based neural network; Global convergence

## 1 Introduction

As is well known, tensors are higher order generalizations of matrices, which are common tools to construct the mathematical models of systems in high dimension. For example, a black and white image (including width and height) can be stored as a matrix, while an RGB image (including width, height and brightness) is often stored as a three-order tensor, and a color video image (including width, height, brightness and time) must be stored as a four-order tensor. An  $m$ th-order real tensor  $\mathcal{A} = (a_{i_1 i_2 \dots i_m})$  ( $a_{i_1 i_2 \dots i_m} \in \mathbb{R}$ ,  $1 \leq i_j \leq I_j$ ,  $j \in \langle m \rangle = \{1, 2, \dots, m\}$ ) is a multidimensional array with  $I_1 I_2 \dots I_m$  entries. Clearly, an order one tensor is a vector, and an order two tensor is a matrix. Let  $\mathbb{R}^{I_1 \times \dots \times I_n}$  be the set of order  $n$ , dimension  $I_1 \times \dots \times I_n$  tensors over  $\mathbb{R}$ . For any two tensors  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_n \times K_1 \times \dots \times K_n}$  and  $\mathcal{B} \in \mathbb{R}^{K_1 \times \dots \times K_n \times J_1 \times \dots \times J_m}$ , the Einstein product  $\mathcal{A} *_n \mathcal{B}$  is defined by [1]

$$(\mathcal{A} *_n \mathcal{B})_{i_1 \dots i_n j_1 \dots j_m} = \sum_{k_1 \dots k_n} a_{i_1 \dots i_n k_1 \dots k_n} b_{k_1 \dots k_n j_1 \dots j_m}, \quad (1)$$

which indicates that  $\mathcal{A} *_n \mathcal{B} \in \mathbb{R}^{I_1 \times \dots \times I_n \times J_1 \times \dots \times J_m}$ . It is obvious that, when  $m = n = 1$ , the Einstein product (1) reduces to a matrix product.

In practice, various kinds of tensor equations arise from physics, mechanics, Markov process and partial differential equations. In this paper, we are interested in the following time-varying Sylvester tensor equations (TVSTEs) via the Einstein product:

$$A(t) *_n \mathcal{X} + \mathcal{X} *_m B(t) = C(t), \quad t \geq 0, \tag{2}$$

where  $A(t) \in \mathbb{R}^{I_1 \times \dots \times I_n \times I_1 \times \dots \times I_n}$ ,  $B(t) \in \mathbb{R}^{K_1 \times \dots \times K_m \times K_1 \times \dots \times K_m}$ ,  $C(t) \in \mathbb{R}^{I_1 \times \dots \times I_n \times K_1 \times \dots \times K_m}$  are input tensors, and  $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_n \times K_1 \times \dots \times K_m}$  is an unknown tensor to be determined. TVSTEs can be used to discrete a linear partial differential equation by the finite element and finite difference. For example, for noncentrosymmetric materials in physics, the linear piezoelectric equation is expressed as [2]:

$$\mathcal{A} *_2 X = p,$$

where  $\mathcal{A} \in \mathbb{R}^{3 \times 3 \times 3}$  is a piezoelectric tensor, and  $X \in \mathbb{R}^{3 \times 3}$  is a stress tensor, and  $p \in \mathbb{R}^3$  is an electric charge density displacement. The two-dimensional (2D) Poisson problem is

$$\begin{cases} -\nabla^2 v = f, & \text{in } \Omega = \{(x, y) \mid 0 < x, y < 1\}, \\ v = 0, & \text{on } \partial\Omega, \end{cases} \tag{3}$$

where  $f$  is a given function, and the Laplace operator  $\nabla^2 v$  is defined as

$$\nabla^2 v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}.$$

By the standard central difference approximations, Poisson’s equation (3) in two dimensions can be depicted as the following four-order tensor representation [3]:

$$\mathcal{A} *_2 X = C,$$

where  $\mathcal{A} \in \mathbb{R}^{n \times n \times n \times n}$ ,  $X \in \mathbb{R}^{n \times n}$  and  $C \in \mathbb{R}^{n \times n}$  are the discretized functions  $v$  and  $f$  on the unit square mesh. Similarly, the three-dimensional (3D) discretized Poisson problem can be depicted as [3]:

$$\mathcal{A} *_3 \mathcal{X} = C,$$

where  $\mathcal{A} \in \mathbb{R}^{n \times n \times n \times n \times n \times n}$ ,  $\mathcal{X}$  and  $C \in \mathbb{R}^{n \times n \times n}$ .

TVSTEs include the following special cases:

- (1). If  $m = n = 1$ , TVSTEs reduce to the time-varying Sylvester matrix equations (TVSMEs) [4]:

$$A(t)X + XB(t) = C(t), \quad t \geq 0, \tag{4}$$

where  $A(t)$ ,  $B(t)$ ,  $C(t)$ , and  $X$  are matrices with compatible dimension; and if  $A(t)$ ,  $B(t)$ , and  $C(t)$  are further time invariant, TVSMEs reduce to the following classic Sylvester matrix equations (SMEs):

$$AX + XB = C, \tag{5}$$

which again includes the well-known Lyapunov matrix equations and the Stein matrix equations as its special cases [5–7]. The SMEs serves as a basic model arising from control theory, system theory, stability analysis etc.

- (2). If  $\mathcal{A}(t)$ ,  $\mathcal{B}(t)$  and  $\mathcal{C}(t)$  are time invariant, TVSTEs reduce to the Sylvester tensor equations (STEs) [8]:

$$\mathcal{A} *_n \mathcal{X} + \mathcal{X} *_m \mathcal{B} = \mathcal{C}, \tag{6}$$

which comes from the finite difference, finite element or spectral methods [9, 10] and plays an important role in discretization of the linear partial differential equations in high dimension.

A very basic and important problem in the study of the above equations concerns their solutions. In the past several decades, many researchers have carried out their work to find analytical and numerical solutions of SMEs, TVSMEs and STEs. For example, Ding and Chen [11] presented a gradient-based iterative algorithm for SMEs. Zhang *et al.* [4] introduced a recurrent neural network with implicit dynamics for the approximate solution of TVSMEs. Wang and Xu [12] developed some iterative algorithms for solving some tensor equations, which were generalized by Huang and Ma [13] to solve STEs. When the above equations are inconsistent, Lv and Zhang [14] designed an iterative algorithm to find the least squares solutions of SMEs. Sun and Wang [7] extended the conjugate gradient method to get the least squares solution with the least Frobenius norm of the generalized periodic SMEs. Specifically, during the past ten years, Hajarian et al. have conducted intensive research on the iterative method for solving various matrix equations, such as the generalized conjugate direction algorithm for solving the general coupled matrix equations over symmetric matrices [15], the finite algorithm for solving the generalized non-homogeneous Yakubovich-transpose matrix equation [16], and the symmetric solutions of general Sylvester matrix equations via the Lanczos version of the biconjugate residual algorithm [17]. A tensor form of the conjugate gradient method was given to solve inconsistent tensor equations [12, 13]. Furthermore, by virtue of the Moore–Penrose inverse and the  $\{1\}$ -inverse of the tensor, Sun *et al.* [8] obtained analytic solutions of some special linear tensor equations. Other iterative methods for matrix/tensor equations and their applications can be found in [18–27] and the references therein.

To the best of the authors’ knowledge, no research has been devoted to the solutions of the TVSTEs. In this paper, we are going to extend a special kind of recurrent neural networks, i.e., the continuous-time Zhang neural network (CTZNN), to solve TVSTEs. ZNN was proposed by Yunong Zhang in March 2001 [4], which is quite suitable to solve various time-varying problems, such as time-varying nonlinear optimization [28–30], time-varying convex quadratic programming [31], time-varying matrix pseudoinversion [32] and time-varying absolute value equations [33]. Motivated by [4, 12, 13], we design three noise-tolerant CTZNNs (NTCTZNN1-3) for solving TVSTEs (2). Convergence analysis shows that: (1) the residual error  $\|\mathcal{A}(t) *_n \mathcal{X} + \mathcal{X} *_m \mathcal{B}(t) - \mathcal{C}(t)\|$  generated by NTCTZNN1 and NTCTZNN3 can be arbitrarily small if their design parameters are large enough; (2) the residual error of NTCTZNN2 converges to zero globally no matter how large the unknown noise is, where the Frobenius norm  $\|\cdot\|$  of  $\mathcal{A}$  is defined as  $\|\mathcal{A}\| = (\sum_{i_1 \dots i_n j_1 \dots j_n} |a_{i_1 \dots i_n j_1 \dots j_n}|^2)^{1/2}$  for a tensor  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_n \times J_1 \times \dots \times J_n}$ .

The remainder of this paper is organized as follows. In Sect. 2, we introduce some necessary notations that are essential to derive main results of this paper. In Sect. 3, we propose

three NTCTZNNs and GNN methods to solve TVSTEs, and establish their convergence. In Sect. 4, to verify the effectiveness of NTCTZNN to solve TVSTEs, we present two numerical examples. Finally, some brief conclusions are drawn in Sect. 5.

## 2 Preliminaries

In this section, we introduce some useful notations and recall some known results.

### Definition 2.1 ([12, 34])

- (1) Let  $\mathcal{A} = (a_{i_1 \dots i_n j_1 \dots j_m}) \in \mathbb{R}^{I_1 \times \dots \times I_n \times J_1 \times \dots \times J_m}$ ,  $\mathcal{B} = (b_{j_1 \dots j_m i_1 \dots i_n}) \in \mathbb{R}^{J_1 \times \dots \times J_m \times I_1 \times \dots \times I_n}$ , if  $a_{i_1 \dots i_n j_1 \dots j_m} = b_{j_1 \dots j_m i_1 \dots i_n}$ , then the tensor  $\mathcal{B}$  is called the transpose of  $\mathcal{A}$ , denoted by  $\mathcal{A}^T$ .
- (2) A tensor  $\mathcal{D} = (d_{i_1 \dots i_n j_1 \dots j_n}) \in \mathbb{R}^{I_1 \times \dots \times I_n \times J_1 \times \dots \times J_n}$  is a diagonal tensor if  $d_{i_1 \dots i_n j_1 \dots j_n} = 0$  in the case that the indices  $i_1 \dots i_n$  are different from  $j_1 \dots j_n$ . Furthermore, in this case if all the diagonal entries  $d_{i_1 \dots i_n i_1 \dots i_n} = 1$ , then  $\mathcal{D}$  is called a unit tensor, denoted by  $\mathcal{I}$ . Specially, if all the entries of a tensor are zero, we say this tensor a zero tensor, denoted by  $\mathcal{O}$ .
- (3) The trace of a tensor  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_n \times I_1 \times \dots \times I_n}$  is defined as  $\text{tr}(\mathcal{A}) = \sum_{i_1 \dots i_n} a_{i_1 \dots i_n i_1 \dots i_n}$ .
- (4) Let  $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times \dots \times I_n \times I_1 \times \dots \times I_n}$ , the inner product of  $\mathcal{A}, \mathcal{B}$  is defined by

$$\langle \mathcal{A}, \mathcal{B} \rangle = \text{tr}(\mathcal{B}^T *_n \mathcal{A}). \tag{7}$$

In the following, we set  $I = I_1 I_2 \dots I_n$ ,  $J = J_1 J_2 \dots J_n$ ,  $K = K_1 K_2 \dots K_m$ . Motivated by Definition 2.3 in [12], we give the following definition.

**Definition 2.2** Define the transformation  $\Phi_{IK} : \mathbb{R}^{I_1 \times \dots \times I_n \times K_1 \times \dots \times K_m} \rightarrow \mathbb{R}^{I \times K}$  with  $\Phi_{IK}(\mathcal{A}) = A$  defined component-wise as

$$(\mathcal{A})_{i_1 \dots i_n k_1 \dots k_m} \rightarrow (A)_{st}, \tag{8}$$

where  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_n \times K_1 \times \dots \times K_m}$ ,  $A \in \mathbb{R}^{I \times K}$ ,  $s = i_n + \sum_{p=1}^{n-1} ((i_p - 1) \prod_{q=p+1}^n I_q)$  and  $t = k_m + \sum_{p=1}^{m-1} ((k_p - 1) \prod_{q=p+1}^m K_q)$ . Note we use the convention  $\sum_{p=1}^0 a_p = 0$ .

**Example 2.1** Let  $\mathcal{A} = (a_{i_1 i_2 k_1 k_2 k_3}) \in \mathbb{R}^{2 \times 2 \times 2 \times 2 \times 2}$  be a tensor such that

$$\begin{aligned} \mathcal{A}(:, :, 1, 1, 1) &= \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, & \mathcal{A}(:, :, 1, 1, 2) &= \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}, & \mathcal{A}(:, :, 1, 2, 1) &= \begin{bmatrix} 9 & 10 \\ 11 & 12 \end{bmatrix}, \\ \mathcal{A}(:, :, 1, 2, 2) &= \begin{bmatrix} 13 & 14 \\ 15 & 16 \end{bmatrix}, & \mathcal{A}(:, :, 2, 1, 1) &= \begin{bmatrix} 17 & 18 \\ 19 & 20 \end{bmatrix}, \\ \mathcal{A}(:, :, 2, 1, 2) &= \begin{bmatrix} 21 & 22 \\ 23 & 24 \end{bmatrix}, & \mathcal{A}(:, :, 2, 2, 1) &= \begin{bmatrix} 25 & 26 \\ 27 & 28 \end{bmatrix}, \\ \mathcal{A}(:, :, 2, 2, 2) &= \begin{bmatrix} 29 & 30 \\ 31 & 32 \end{bmatrix}. \end{aligned}$$

By Definition 2.2, we have

$$A = \begin{bmatrix} 1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 \\ 2 & 6 & 10 & 14 & 18 & 22 & 26 & 30 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 & 31 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 \end{bmatrix}.$$

*Remark 2.1* Given the positive integers  $I_1, \dots, I_n, K_1, \dots, K_m$ , we can define an inverse function of  $\Phi_{IK} : \mathbb{R}^{I_1 \times \dots \times I_n \times K_1 \times \dots \times K_m} \rightarrow \mathbb{R}^{I \times K}$  defined in Definition 2.2 as follows:

$$\Phi_{IK}^{-1} : \mathbb{R}^{I \times K} \rightarrow \mathbb{R}^{I_1 \times \dots \times I_n \times K_1 \times \dots \times K_m},$$

with  $(A)_{st} \rightarrow (A)_{i_1 \dots i_n k_1 \dots k_m}$ , where the  $j$ th column of the matrix  $A$  consists the  $j$ th element in the set  $\{A(:, \dots, :, k_1, \dots, k_m) \mid \forall k_1, \dots, k_m\}$ . Here we sort all the elements in this set in lexicographic order, that is, from  $(1, \dots, 1)$  to  $(K_1, \dots, K_m)$ .

It is well known that using the Kronecker product and the vectorization operator, one can transform a linear matrix equations into a linear equations. Similarly, the following proposition indicates that the transformation  $\Phi_{IK}$  can transform a linear tensor equations into a linear matrix equations.

**Proposition 2.1** ([12]) *Let  $\Phi_{IJ}$  be defined as Definition 2.2. Then*

$$A *_n \mathcal{X} = C \iff \Phi_{IJ}(A)\Phi_{JK}(\mathcal{X}) = \Phi_{IK}(C), \tag{9}$$

where the tensors  $A \in \mathbb{R}^{I_1 \times \dots \times I_n \times J_1 \times \dots \times J_n}$ ,  $\mathcal{X} \in \mathbb{R}^{J_1 \times \dots \times J_n \times K_1 \times \dots \times K_m}$ ,  $C \in \mathbb{R}^{I_1 \times \dots \times I_n \times K_1 \times \dots \times K_m}$ .

The following lemma is a direct generalization of the derivative principle of a matrix.

**Lemma 2.1** *Let  $A(t) \in \mathbb{R}^{I_1 \times \dots \times I_n \times J_1 \times \dots \times J_n}$ ,  $\mathcal{X}(t) \in \mathbb{R}^{J_1 \times \dots \times J_n \times K_1 \times \dots \times K_m}$ . We have*

$$((A(t) *_n \mathcal{X}(t))'_t)_{i_1 \dots i_n k_1 \dots k_m} = (A(t))'_t *_n \mathcal{X}(t) + A(t) *_n (\mathcal{X}(t))'_t. \tag{10}$$

*Proof* Assume that  $\Phi_{IJ}(A(t)) = A(t)$ ,  $\Phi_{JK}(\mathcal{X}(t)) = X(t)$ . Then, by Definition 2.2, one has  $A(t) \in \mathbb{R}^{IJ}$ ,  $X(t) \in \mathbb{R}^{JK}$ . By the derivative principle of a matrix, we have

$$\begin{aligned} & ((A(t) *_n \mathcal{X}(t))'_t)_{i_1 \dots i_n k_1 \dots k_m} \\ &= ((A(t) *_n \mathcal{X}(t))_{i_1 \dots i_n k_1 \dots k_m})'_t \\ &= \left( \sum_{j_1 \dots j_n} a_{i_1 \dots i_n j_1 \dots j_n}(t) x_{j_1 \dots j_n k_1 \dots k_m}(t) \right)'_t \\ &= \left( \sum_{v=1}^J A_{sv}(t) X_{vw}(t) \right)'_t \\ &= \sum_{v=1}^J (A_{sv}(t) X_{vw}(t))'_t \end{aligned}$$

$$\begin{aligned}
 &= \sum_{\nu=1}^J ((A_{s\nu}(t))'_t X_{\nu w}(t) + A_{s\nu}(t)(X_{\nu w}(t))'_t) \\
 &= ((A(t)X(t))'_t)_{sw} \\
 &= ((A(t))'_t X(t) + A(t)(X(t))'_t)_{sw} \\
 &= (\Phi_{IK}^{-1}[(A(t))'_t X(t) + A(t)(X(t))'_t])_{i_1 \dots i_n k_1 \dots k_m} \\
 &= (\Phi_{IK}^{-1}[(\Phi_{IJ}(\mathcal{A}(t)))'_t \Phi_{JK}(\mathcal{X}(t)) + \Phi_{IJ}(\mathcal{A}(t))(\Phi_{JK}(\mathcal{X}(t))'_t)])_{i_1 \dots i_n k_1 \dots k_m} \\
 &= (\Phi_{IK}^{-1}[\Phi_{IK}((\mathcal{A}(t))'_t *_n \mathcal{X}(t) + \Phi_{IK}(\mathcal{A}(t)(\mathcal{X}(t))'_t *_n)])]_{i_1 \dots i_n k_1 \dots k_m} \\
 &= ((\mathcal{A}(t))'_t *_n \mathcal{X}(t) + \mathcal{A}(t) *_n (\mathcal{X}(t))'_t)_{i_1 \dots i_n k_1 \dots k_m},
 \end{aligned}$$

where  $s, \nu$  and  $w$  are defined as in Definition 2.1, and the second-to-last equality comes from Proposition 2.1. This completes the proof.  $\square$

### 3 NTCTZNNs and GNN for TVSTEs

In this section, we will present three noise-tolerant continuous-time Zhang neural networks (NTCTZNNs) and a gradient-based neural network (GNN) for TVSTEs, and we analyze their convergence property.

Firstly, following Zhang *et al.*'s design method [4], we define a tensor-valued indefinite error-function as follows:

$$\mathcal{E}(t) = \mathcal{A}(t) *_n \mathcal{X} + \mathcal{X} *_m \mathcal{B}(t) - \mathcal{C}(t) \in \mathbb{R}^{I_1 \times \dots \times I_n \times K_1 \times \dots \times K_m}, \tag{11}$$

where every entry denoted by  $e_{i_1 \dots i_n k_1 \dots k_m}(t) \in \mathbb{R}$  may be negative or unbounded.

Then, let us recall the Zhang neural network (ZNN) design formula (also termed Zhang *et al.*'s design formula)

$$\dot{\mathcal{E}}(t) = -\gamma F(\mathcal{E}(t)),$$

where  $\gamma > 0$  is a design parameter,  $F(\cdot)$  is a tensor activation function defined on every entry of the error-function  $\mathcal{E}(t)$ , and  $\dot{\mathcal{E}}(t)$  is a component-wise derivative of the error-function  $\mathcal{E}(t)$ . By (10) in Lemma 2.1, substituting the error-function  $\mathcal{E}(t)$  into the above Zhang neural network (ZNN) design formula, we get a continuous-time ZNN (CTZNN):

$$\begin{aligned}
 &\mathcal{A}(t) *_n \dot{\mathcal{X}} + \dot{\mathcal{X}} *_m \mathcal{B}(t) \\
 &= -\gamma F(\mathcal{A}(t) *_n \mathcal{X} + \mathcal{X} *_m \mathcal{B}(t) - \mathcal{C}(t)) - \dot{\mathcal{A}}(t) *_n \mathcal{X} - \mathcal{X} *_m \dot{\mathcal{B}}(t) + \dot{\mathcal{C}}(t),
 \end{aligned} \tag{12}$$

where  $\dot{\mathcal{A}}(t), \dot{\mathcal{B}}(t), \dot{\mathcal{C}}(t)$  denote the time-derivatives of tensors  $\mathcal{A}(t), \mathcal{B}(t), \mathcal{C}(t)$ , respectively.

*Remark 3.1* Let  $f(\cdot)$  be the entry of the tensor-valued function  $F(\cdot)$ . The function  $f(\cdot)$  can be set as any odd and strictly monotone increasing function. There are six basic types of activation functions in the literature, i.e., the linear activation function, the power activation function, the bipolar sigmoid activation function, the power-sigmoid activation function, the sign-bi-power activation function and the power-sum activation function [35]. Note that different activation functions result in different numerical performance of

CTZNN (12). Generally speaking, the performance of CTZNN with power-sigmoid activation function is better than that of CTZNN with linear activation function, and CTZNN with sign-bi-power activation function often converges in a limited time.

CTZNN (12) is suitable to solve TVSTEs without noise. However, noise is ubiquitous, which should not be ignored in real life. In the following, we only consider the constant noise for simplicity, which is denoted by  $\eta(t) = \eta$ . Setting  $\bar{\eta} = (\eta) \in \mathbb{R}^{I_1 \times \dots \times I_n \times K_1 \times \dots \times K_m}$ , whose entries all equal to  $\eta$ . Based on the results about CTZNNs in the literature [36–38], we are going to propose three noise-tolerant CTZNNs for TVSTEs.

(1) Setting  $e(t) = \mathcal{E}(t)$  in the following improved Zhang design formula:

$$\dot{e}(t) = -\gamma e(t) + \bar{\eta}, \quad \gamma > 0, \tag{13}$$

we get the first noise-tolerant CTZNN (NTCTZNN1) for TVSTEs:

$$\begin{aligned} & \mathcal{A}(t) *_n \dot{\mathcal{X}} + \dot{\mathcal{X}} *_m \mathcal{B}(t) \\ & = -\gamma (\mathcal{A}(t) *_n \mathcal{X} + \mathcal{X} *_m \mathcal{B}(t) - \mathcal{C}(t)) - \dot{\mathcal{A}}(t) *_n \mathcal{X} - \mathcal{X} *_m \dot{\mathcal{B}}(t) + \dot{\mathcal{C}}(t) + \bar{\eta}, \end{aligned} \tag{14}$$

whose convergence is given in the following theorem.

**Theorem 3.1** *NTCTZNN1 (14) converges to the theoretical solution of TVSTEs with the limit of the residual error being  $\sqrt{IK}\eta/\gamma$ .*

*Proof* Obviously, NTCTZNN1 (14) can be written as

$$\dot{\mathcal{E}}(t) = -\gamma \mathcal{E}(t) + \bar{\eta},$$

which can be decoupled into the following  $IK$  differential equations:

$$\dot{e}_{i_1 \dots i_n k_1 \dots k_m}(t) = -\gamma e_{i_1 \dots i_n k_1 \dots k_m}(t) + \eta, \quad \forall i_p, k_q, p \in \langle n \rangle, q \in \langle m \rangle,$$

which has a closed-form solution:

$$e_{i_1 \dots i_n k_1 \dots k_m}(t) = e_{i_1 \dots i_n k_1 \dots k_m}(0) \exp(-\gamma t) + \frac{\eta}{\gamma}, \quad \forall i_p, k_q, p \in \langle n \rangle, q \in \langle m \rangle.$$

Furthermore, we can have

$$\lim_{t \rightarrow \infty} e_{i_1 \dots i_n k_1 \dots k_m}(t) = \frac{\eta}{\gamma}, \quad \forall i_p, k_q, p \in \langle n \rangle, q \in \langle m \rangle.$$

So

$$\lim_{t \rightarrow +\infty} \|\mathcal{E}(t)\| = \frac{\sqrt{IK}\eta}{\gamma}.$$

This completes the proof. □

(2) Setting  $e(t) = \mathcal{E}(t)$  in the following improved Zhang design formula [36]:

$$\dot{e}(t) = -\gamma_1 e(t) - \gamma_2 \int_0^t e(s) ds + \bar{\eta}, \quad \gamma_1 > 0, \gamma_2 > 0, \tag{15}$$

we get the second noise-tolerant CTZNN (NTCTZNN2) for TVSTEs:

$$\begin{aligned} &A(t) *_n \dot{\mathcal{X}} + \dot{\mathcal{X}} *_m B(t) \\ &= -\gamma_1 (A(t) *_n \mathcal{X} + \mathcal{X} *_m B(t) - C(t)) - \gamma_2 \int_0^t (A(s) *_n \mathcal{X} + \mathcal{X} *_m B(s) - C(s)) ds \\ &\quad - \dot{A}(t) *_n \mathcal{X} - \mathcal{X} *_m \dot{B}(t) + \dot{C}(t) + \bar{\eta}. \end{aligned} \tag{16}$$

**Theorem 3.2** *No matter how large the unknown noise  $\eta$  is, NTCTZNN2 (16) converges to the theoretical solution of TVSTEs globally.*

*Proof* Obviously, NTCTZNN2 (16) can be written as

$$\dot{\mathcal{E}}(t) = -\gamma_1 \mathcal{E}(t) - \gamma_2 \int_0^t \mathcal{E}(s) ds + \bar{\eta},$$

which can be decoupled into *IK* differential equations:

$$\dot{e}_{i_1 \dots i_n k_1 \dots k_m}(t) = -\gamma_1 e_{i_1 \dots i_n k_1 \dots k_m}(t) - \gamma_2 \int_0^t (e_{i_1 \dots i_n k_1 \dots k_m}(s)) ds + \eta. \tag{17}$$

Taking Laplace transformation on both sides of (17), one has

$$s e_{i_1 \dots i_n k_1 \dots k_m}(s) - e_{i_1 \dots i_n k_1 \dots k_m}(0) = -\gamma_1 e_{i_1 \dots i_n k_1 \dots k_m}(s) - \frac{\gamma_2}{s} e_{i_1 \dots i_n k_1 \dots k_m}(s) + \frac{\eta}{s}, \tag{18}$$

where  $e_{i_1 \dots i_n k_1 \dots k_m}(t)$  is the image function of  $e_{i_1 \dots i_n k_1 \dots k_m}(t)$ . From (18), we have

$$e_{i_1 \dots i_n k_1 \dots k_m}(s) = \frac{s e_{i_1 \dots i_n k_1 \dots k_m}(0) + \eta}{s^2 + s\gamma_1 + \gamma_2}.$$

Two poles of its transfer function are

$$s_{1,2} = \frac{-\gamma_1 \pm \sqrt{\gamma_1^2 - 4\gamma_2}}{2},$$

which are located on the left half-plane for any  $\gamma_1, \gamma_2 > 0$ . Thus the system (17) is stable and the final value theorem holds. That is,

$$\lim_{t \rightarrow \infty} e_{i_1 \dots i_n k_1 \dots k_m}(t) = \lim_{s \rightarrow 0} s e_{i_1 \dots i_n k_1 \dots k_m}(s) = \lim_{s \rightarrow 0} \frac{s^2 e_{i_1 \dots i_n k_1 \dots k_m}(0) + s\eta}{s^2 + s\gamma_1 + \gamma_2} = 0.$$

This completes the proof. □

*Remark 3.2* If  $\gamma_2 = 0$ , then NTCTZNN2 (16) reduces NTCTZNN1 (14). Though NTCTZNN2 (16) is more complicate than NTCTZNN1 (14) when  $\gamma_2 > 0$ , Theorems 3.1 and 3.2 indicate that NTCTZNN2 (16) is more stable than NTCTZNN1 (14) in this case.



*Remark 3.3* In the practical computation, to avoid the integral manipulation, we would better transform the integro-differential equations NTCTZNN1 (14) into the following differential equations:

$$\begin{cases} \mathcal{A}(t) *_n \dot{\mathcal{X}} + \dot{\mathcal{X}} *_m \mathcal{B}(t) = -\gamma_1(\mathcal{A}(t) *_n \mathcal{X} + \mathcal{X} *_m \mathcal{B}(t) - \mathcal{C}(t)) - \gamma_2 \mathcal{Y}(t) \\ \quad - \dot{\mathcal{A}}(t) *_n \mathcal{X} - \mathcal{X} *_m \dot{\mathcal{B}}(t) + \dot{\mathcal{C}}(t) + \bar{\eta}, \\ \dot{\mathcal{Y}}(t) = \mathcal{A}(t) *_n \mathcal{X} + \mathcal{X} *_m \mathcal{B}(t) - \mathcal{C}(t). \end{cases} \tag{19}$$

(3) Setting  $e(t) = \mathcal{E}(t)$  in the following improved Zhang design formula:

$$\dot{e}(t) = -\gamma_1 F(e(t)) - \gamma_2 G\left(e(t) + \gamma_1 \int_0^t F(e(s)) ds\right) + \bar{\eta}, \quad \gamma_1 \geq 0, \gamma_2 > 0, \tag{20}$$

where  $F(\cdot), G(\cdot)$  are two activation-function arrays, we get the third noise-tolerant CTZNN (NTCTZNN3) for TVSTEs:

$$\begin{aligned} & \mathcal{A}(t) *_n \dot{\mathcal{X}} + \dot{\mathcal{X}} *_m \mathcal{B}(t) \\ &= -\gamma_1 F(\mathcal{A}(t) *_n \mathcal{X} + \mathcal{X} *_m \mathcal{B}(t) - \mathcal{C}(t)) \\ & \quad - \gamma_2 G\left(\mathcal{A}(t) *_n \mathcal{X} + \mathcal{X} *_m \mathcal{B}(t) - \mathcal{C}(t) \right. \\ & \quad \left. + \gamma_1 \int_0^t F(\mathcal{A}(s) *_n \mathcal{X} + \mathcal{X} *_m \mathcal{B}(s) - \mathcal{C}(s)) ds\right) \\ & \quad - \dot{\mathcal{A}}(t) *_n \mathcal{X} + \mathcal{X} *_m \dot{\mathcal{B}}(t) + \dot{\mathcal{C}}(t) + \bar{\eta}. \end{aligned} \tag{21}$$

Obviously, NTCTZNN3 (21) can be written as

$$\dot{\mathcal{E}}(t) = -\gamma_1 F(\mathcal{E}(t)) - \gamma_2 G\left(\mathcal{E}(t) + \gamma_1 \int_0^t F(\mathcal{E}(s)) ds\right) + \bar{\eta},$$

which can be decoupled into  $IK$  differential equations:

$$\begin{aligned} \dot{e}_{i_1 \dots i_n k_1 \dots k_m}(t) &= -\gamma_1 f(e_{i_1 \dots i_n k_1 \dots k_m}(t)) \\ & \quad - \gamma_2 g\left(e_{i_1 \dots i_n k_1 \dots k_m}(t) + \gamma_1 \int_0^t f(e_{i_1 \dots i_n k_1 \dots k_m}(s)) ds\right) + \eta. \end{aligned} \tag{22}$$

Define an intermediate variable  $y_{i_1 \dots i_n k_1 \dots k_m}(t)$  as follows:

$$y_{i_1 \dots i_n k_1 \dots k_m}(t) = e_{i_1 \dots i_n k_1 \dots k_m}(t) + \gamma_1 \int_0^t f(e_{i_1 \dots i_n k_1 \dots k_m}(s)) ds. \tag{23}$$

To establish the convergence of NTCTZNN3 (21), we make the following assumption about the constant noise  $\bar{\eta}$ .

**Assumption 3.1** The constant noise  $\bar{\eta} = (\eta) \in \mathbb{R}^{I_1 \times \dots \times I_n \times K_1 \times \dots \times K_m}$  satisfies

$$|\eta y_{i_1 \dots i_n k_1 \dots k_m}(t)| \leq \alpha \gamma y_{i_1 \dots i_n k_1 \dots k_m}(t) g(y_{i_1 \dots i_n k_1 \dots k_m}(t)), \quad t \geq 0,$$

where  $\alpha \in (0, 1)$ , and  $g(\cdot)$  is the entry of the activation function  $G(\cdot)$ .

**Theorem 3.3** *Under Assumption 3.1, NTCTZNN3 (21) converges to the theoretical solution of TVSTEs with the limit of the residual error being  $\sqrt{IK}g^{-1}(\eta/\gamma)$ .*

*Proof* Taking the time-derivative of (23) on both sides, one has

$$\dot{y}_{i_1 \dots i_n k_1 \dots k_m}(t) = \dot{e}_{i_1 \dots i_n k_1 \dots k_m}(t) + \gamma_1 f(e_{i_1 \dots i_n k_1 \dots k_m}(t)). \tag{24}$$

Then substituting (24) into (22), we have

$$\dot{y}_{i_1 \dots i_n k_1 \dots k_m}(t) = -\gamma_2 g(y_{i_1 \dots i_n k_1 \dots k_m}(t)) + \eta. \tag{25}$$

For the differential equation (25), let us define a Lyapunov function candidate

$$v_{i_1 \dots i_n k_1 \dots k_m}(t) = y_{i_1 \dots i_n k_1 \dots k_m}^2(t)/2 \geq 0,$$

whose time-derivative is

$$\begin{aligned} \dot{v}_{i_1 \dots i_n k_1 \dots k_m}(t) &= y_{i_1 \dots i_n k_1 \dots k_m}(t) \dot{y}_{i_1 \dots i_n k_1 \dots k_m}(t) \\ &= -\gamma y_{i_1 \dots i_n k_1 \dots k_m}(t) g(y_{i_1 \dots i_n k_1 \dots k_m}(t)) + \eta e_{i_1 \dots i_n k_1 \dots k_m}(t) \\ &\leq -(1 - \alpha) \gamma y_{i_1 \dots i_n k_1 \dots k_m}(t) g(y_{i_1 \dots i_n k_1 \dots k_m}(t)). \end{aligned}$$

Since  $g(\cdot)$  is an odd and monotone increasing function, we have  $y_{i_1 \dots i_n k_1 \dots k_m}(t) \times g(y_{i_1 \dots i_n k_1 \dots k_m}(t)) > 0$  for  $y_{i_1 \dots i_n k_1 \dots k_m}(t) \neq 0$ , and  $y_{i_1 \dots i_n k_1 \dots k_m}(t) g(y_{i_1 \dots i_n k_1 \dots k_m}(t)) = 0$  if and only if  $y_{i_1 \dots i_n k_1 \dots k_m}(t) = 0$ . This indicates the negative definiteness of  $\dot{v}_{i_1 \dots i_n k_1 \dots k_m}(t)$ . Then, by Lyapunov stability theory, equilibrium  $g^{-1}(\eta/\gamma_2)$  of (25) is globally asymptotically stable. Then, by the definition of  $y_{i_1 \dots i_n k_1 \dots k_m}(t)$  in (23), one has

$$\lim_{t \rightarrow \infty} \left( e_{i_1 \dots i_n k_1 \dots k_m}(t) + \gamma_1 \int_0^t f(e_{i_1 \dots i_n k_1 \dots k_m}(s)) ds \right) = g^{-1}(\eta/\gamma_2). \tag{26}$$

Since  $f(\cdot)$  is also an odd and monotone increasing function, one has

$$|e_{i_1 \dots i_n k_1 \dots k_m}(t)| \leq \left| e_{i_1 \dots i_n k_1 \dots k_m}(t) + \gamma_1 \int_0^t f(e_{i_1 \dots i_n k_1 \dots k_m}(s)) ds \right|.$$

This and (26),  $\eta \geq 0, \gamma_1, \gamma_2 > 0$  imply that

$$\lim_{t \rightarrow \infty} |e_{i_1 \dots i_n k_1 \dots k_m}(t)| \leq g^{-1}(\eta/\gamma_2).$$

So

$$\lim_{t \rightarrow \infty} \|\mathcal{E}(t)\| \leq \sqrt{IK}g^{-1}(\eta/\gamma).$$

This completes the proof. □

*Remark 3.4* Obviously, when  $F(\cdot)$  and  $G(\cdot)$  are identity mappings, NTCTZNN3 (21) reduces to NTCTZNN2 (16).

At the end of this section, we develop a gradient-based neural network (GNN) to solve TVSTEs for comparison. Define a scalar-valued norm-based energy function:

$$\xi(\mathcal{X}) = \|\mathcal{A}(t) *_n \mathcal{X} + \mathcal{X} *_m \mathcal{B}(t) - \mathcal{C}(t)\|^2/2.$$

Then, based on Theorem 3.1 in [13], the gradient of the energy function  $\xi(\mathcal{X})$  is

$$\frac{\partial \xi(\mathcal{X})}{\partial \mathcal{X}} = \mathcal{A}^\top(t) *_n \mathcal{E}(t) + \mathcal{E}(t) *_m \mathcal{B}^\top(t),$$

where  $\mathcal{E}(t) = \mathcal{A}(t) *_n \mathcal{X} + \mathcal{X} *_m \mathcal{B}(t) - \mathcal{C}(t)$ . Thus, we get the following GNN:

$$\dot{\mathcal{X}} = -\gamma(\mathcal{A}^\top(t) *_n \mathcal{E}(t) + \mathcal{E}(t) *_m \mathcal{B}^\top(t)) + \bar{\eta}, \tag{27}$$

where  $\gamma > 0$  is a design parameter.

*Remark 3.5* The most difference between GNN (27) and NTCTZNN1-3 lies in the former do not make use of the time-derivative information  $\dot{\mathcal{A}}(t)$ ,  $\dot{\mathcal{B}}(t)$  and  $\dot{\mathcal{C}}(t)$  to enhance its efficiency.

In the following, the following theorem establishes the convergence of GNN (27) for static Sylvester tensor equations, i.e.,  $\dot{\mathcal{A}} = \mathcal{O}$ ,  $\dot{\mathcal{B}} = \mathcal{O}$ ,  $\dot{\mathcal{C}} = \mathcal{O}$ , whose proof is motivated by Theorems 1 and 2 in [38].

**Theorem 3.4** *As for the convergence of GNN (27), we have the following conclusions:*

- (1) *GNN (27) exponentially converges to the theoretical solution of static Sylvester tensor equations without noise;*
- (2) *the computational error of GNN (27) for static Sylvester tensor equations with noise is upper bounded. Furthermore, if the design parameter  $\gamma$  tends to positive infinite, the steady-state solution-error diminishes to zero.*

*Proof* Set  $\Phi_{II}(\mathcal{A}) = A$ ,  $\Phi_{IK}(\mathcal{X}(t)) = X(t)$ ,  $\Phi_{KK}(\mathcal{B}) = B$ ,  $\Phi_{IK}(\mathcal{C}) = C$ ,  $\Phi_{IK}(\mathcal{E}) = E$ ,  $\Phi_{IK}(\bar{\eta}) = \theta$ . Then TVSTEs can be written as

$$E = AX + XB - C,$$

i.e.,

$$\text{vec}(E) = (I_K \otimes A + B^\top \otimes I_I) \text{vec}(X(t)) - \text{vec}(C) = (A \oplus B^\top) \text{vec}(X(t)) - \text{vec}(C).$$

So GNN (27) can be vectorized as

$$\begin{aligned} \text{vec}(\dot{X}(t)) &= -\gamma(I_K \otimes A^\top + B \otimes I_I)((A \oplus B^\top) \text{vec}(X(t)) - \text{vec}(C)) + \text{vec}(\theta) \\ &= -\gamma(A^\top \oplus B)((A \oplus B^\top) \text{vec}(X(t)) - \text{vec}(C)) + \text{vec}(\theta). \end{aligned}$$

Assume  $\mathcal{X}^*$  is one solution of TVSTEs, and set  $\Phi_{IK}(\mathcal{X}^*) = \mathcal{X}^*$ . Then

$$(A \oplus B^\top) \text{vec}(\mathcal{X}^*(t)) = \text{vec}(C).$$

From the above equations, one has

$$\text{vec}(\dot{\mathcal{X}}(t)) = -\gamma(A^\top \oplus B)(A \oplus B^\top)(\text{vec}(\mathcal{X}(t)) - \text{vec}(\mathcal{X}^*)) + \text{vec}(\theta),$$

i.e.,

$$\text{vec}(\dot{\mathcal{X}}(t)) - \text{vec}(\dot{\mathcal{X}}^*) = -\gamma M^\top M(\text{vec}(\mathcal{X}(t)) - \text{vec}(\mathcal{X}^*)) + \text{vec}(\theta),$$

where  $M = A \oplus B^\top$ . Setting  $y(t) = \text{vec}(\mathcal{X}(t)) - \text{vec}(\mathcal{X}^*)$ , we have

$$\dot{y}(t) = -\gamma M^\top M y(t) + \text{vec}(\theta).$$

Suppose  $\alpha$  is the minimum eigenvalue of  $M^\top M$ , which is assumed to be positive definite.

(1) For the static Sylvester tensor equations without noise,

$$\dot{v}(y(t), t) \leq -\gamma \alpha y^\top(t) y(t) = -\gamma \alpha v(y(t), t).$$

Thus,

$$v(y(t), t) = \frac{\|\text{vec}(\mathcal{X}(t)) - \text{vec}(\mathcal{X}^*)\|^2}{2} \leq v(y(0), 0) \exp(-\alpha \gamma t),$$

which implies that the neural state  $\mathcal{X}(t)$  converges to the theoretical solution  $\mathcal{X}^*$  with the exponential rate  $\alpha \gamma / 2$ .

(2) For the static Sylvester tensor equations with noise,

$$\begin{aligned} \dot{v}(y(t), t) &\leq -\gamma \alpha \|y(t)\|^2 + y^\top(t) \text{vec}(\theta) \\ &\leq -\sum_i |y_i(t)| (\alpha \gamma |y_i(t)| - |(\text{vec}(\theta))_i|). \end{aligned}$$

(i) If  $\alpha \gamma |y_i(t)| - |(\text{vec}(\theta))_i| \geq 0, \forall i, t$ , then according to the Lyapunov stability theory [39], the time-varying vector  $y(t)$  converges to zero as time evolves. Thus  $\mathcal{X}(t)$  converges to the theoretical solution  $\mathcal{X}^*(t)$ .

(ii) If  $\alpha \gamma |y_i(t)| - |(\text{vec}(\theta))_i| \leq 0, \exists i, t$ , then the function  $v(y(t), t)$  maybe increasing. However, from the inequality  $\alpha \gamma |y_i(t)| - |(\text{vec}(\theta))_i| \leq 0$ , it is easy to deduce that

$$|y_i(t)| \leq \frac{|(\text{vec}(\theta))_i|}{\alpha \gamma}.$$

Overall, one has

$$\|y(t)\| \leq \frac{\sqrt{IK} \|\bar{\eta}\|}{\alpha \gamma}.$$

This completes the proof. □

### 4 Numerical results

In this section, two examples are presented to show the efficiency of the proposed NTCTZNN1-3 and GNN for solving TVSTEs. All experiments are performed on a Thinkpad laptop with Intel Core 2 CPU 2.10 GHZ and RAM 4.00 GM and written in Matlab R2014a. All first-order ordinary equations encountered are solved by the built-in function ode45.

*Example 4.1* Consider the time-varying Sylvester tensor equations ( $m = n = 2, I_1 = I_2 = K_1 = K_2 = 2$ ):

$$A(t) *_n \mathcal{X} + \mathcal{X} *_m B(t) = C(t), \quad t \in [0, 10], \tag{28}$$

where

$$\begin{aligned} A(:, :, 1, 1) &= \begin{bmatrix} 2 + \cos(2t) & 0 \\ 0 & 0 \end{bmatrix}, & A(:, :, 1, 2) &= \begin{bmatrix} 0 & 1 - \sin(2t) \\ 0 & 0 \end{bmatrix}, \\ A(:, :, 2, 1) &= \begin{bmatrix} 0 & 0 \\ 2 - \cos(2t) & 0 \end{bmatrix}, & A(:, :, 2, 2) &= \begin{bmatrix} 0 & 0 \\ 0 & 1 + \sin(2t) \end{bmatrix}, \\ B(:, :, 1, 1) &= \begin{bmatrix} 2 - \sin(2t) & 0 \\ 0 & 0 \end{bmatrix}, & B(:, :, 1, 2) &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \\ B(:, :, 2, 1) &= \begin{bmatrix} 0 & 0 \\ 1 + \cos(2t) & 0 \end{bmatrix}, & B(:, :, 2, 2) &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \end{aligned}$$

$$C = A.$$

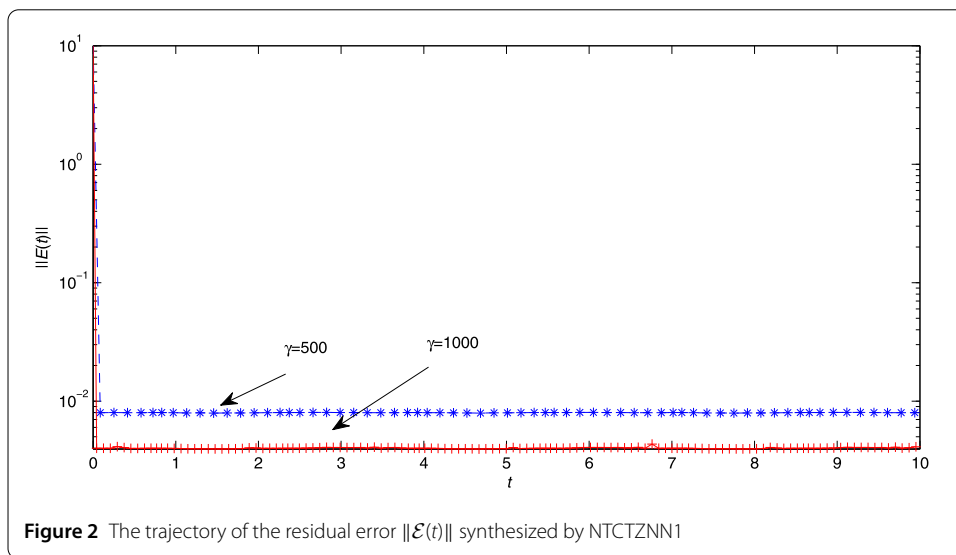
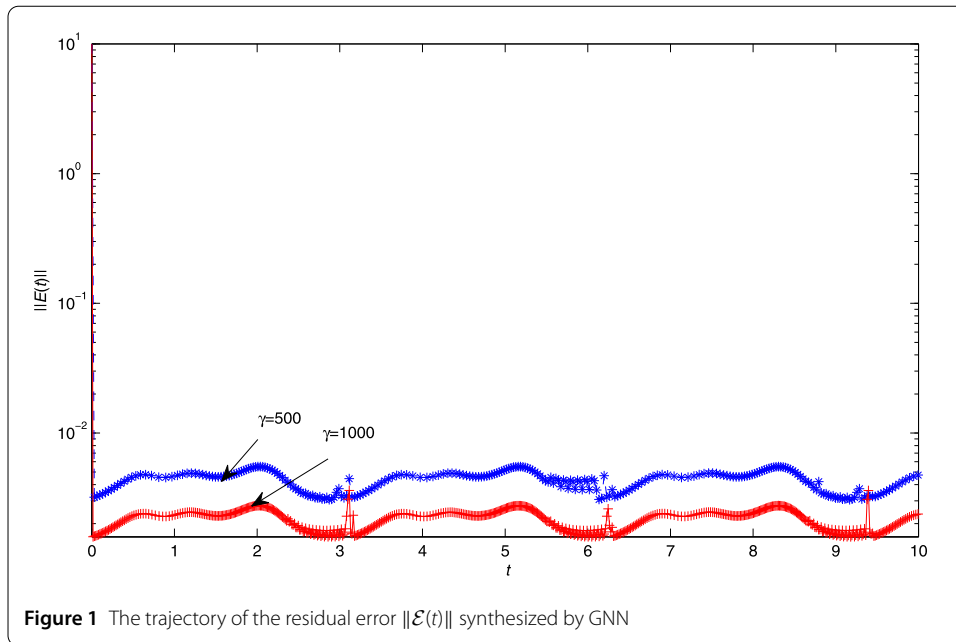
We set the noise  $\eta = 1$  and the initial state tensor  $\mathcal{X}_0 = \mathcal{I}$ .

Firstly, we use the GNN with  $\gamma = 1000$  or  $\gamma = 1000$  to solve problem (28), and the generated residual errors  $\|\mathcal{E}(t)\|$  are shown in Fig. 1. The final residual errors generated by the GNN with  $\gamma = 1000$  and the GNN with  $\gamma = 1000$  are 0.0047 and 0.0024, respectively. Furthermore, the number of iterations of the GNN with  $\gamma = 1000$  and the GNN with  $\gamma = 1000$  are 136,041 and 272,077, respectively.

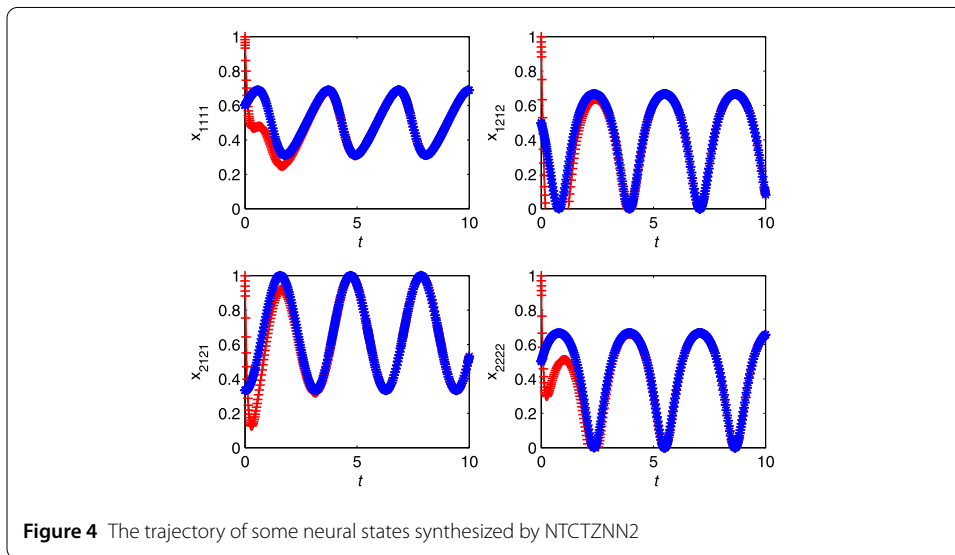
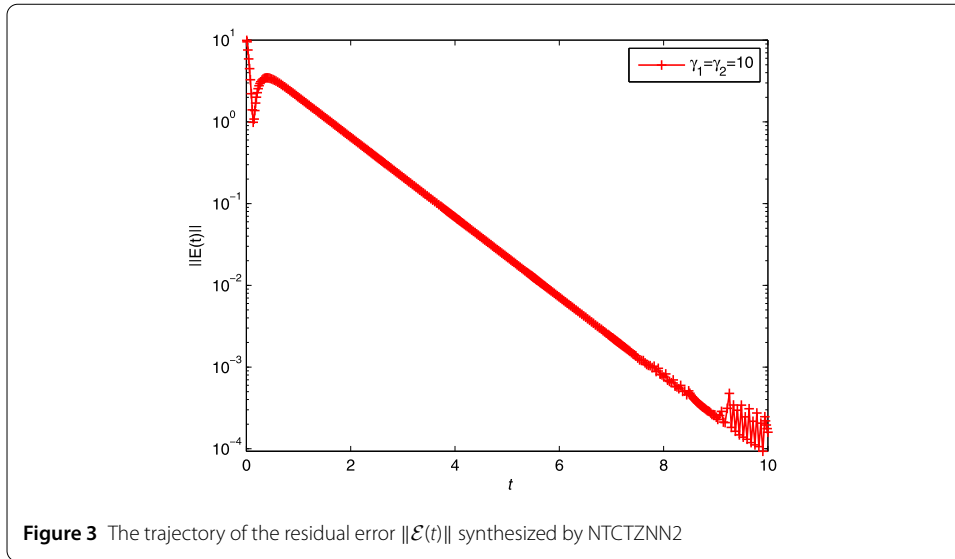
Secondly, we use NTCTZNN1 to solve problem (28). Figure 2 shows the residual error  $\|\mathcal{E}(t)\|$  generated by NTCTZNN1, and the final residual errors generated by NTCTZNN1 with  $\gamma = 500$  and NTCTZNN1 with  $\gamma = 1000$  are 0.0080 and 0.0040, respectively. From Fig. 2 we can see that the performance of NTCTZNN1 with  $\gamma = 1000$  is better than that of NTCTZNN1 with  $\gamma = 500$ , which is in accordance with Theorem 3.1. In addition, the number of iterations of the NTCTZNN1 with  $\gamma = 1000$  and the NTCTZNN1 with  $\gamma = 1000$  are 6541 and 12,561, respectively, which are only about 4% of those generated by GNN. Therefore, NTCTZNN1 computational cost is reduced greatly though it is a little less accurate than GNN.

Thirdly, we use NTCTZNN2 to solve problem (28). The residual error  $\|\mathcal{E}(t)\|$  generated by NTCTZNN2 with  $\gamma_1 = \gamma_2 = 10$  is displayed in Fig. 3, and the final residual error is  $1.5972 \times 10^{-4}$ . In addition, the number of iterations of the NTCTZNN2 with  $\gamma = 10$  is 373, which is much less than that of NTCTZNN1.

Generally speaking, large design parameter can enhance the efficiency of NTCTZNN. However, we also observed that NTCTZNN with large design parameter often takes



more CPU time because the larger the design parameter is, the smaller the step-size of ode45 is. In fact, the CPU times of NTCTZNN2 with  $\gamma_1 = \gamma_2 = 10$  and NTCTZNN2 with  $\gamma_1 = \gamma_2 = 100$  are 1.4219 and 1.8594, respectively. Comparing Figs. 2 and 3, we find that: (1) NTCTZNN2 with  $\gamma_1 = \gamma_2 = 10$  is more efficient than NTCTZNN1 with  $\gamma = 1000$ , because the final residual error generated by NTCTZNN2 with  $\gamma_1 = \gamma_2 = 10$  is about  $10^{-4}$ , while the final residual error generated by NTCTZNN1 with  $\gamma = 1000$  is about  $10^{-3}$ ; (2) the final residual error generated by NTCTZNN1 becomes stable quickly, but the final residual error generated by NTCTZNN2 is shrinking during the tested time period  $[0, 10]$ , which is in accordance with Theorem 3.2. Overall, NTCTZNN2 can get more accurate solution but not increase the computational cost. The neural states  $\mathcal{X}(t)(1, 1, 1, 1)$ ,  $\mathcal{X}(t)(1, 2, 1, 2)$ ,  $\mathcal{X}(t)(2, 1, 2, 1)$ ,  $\mathcal{X}(t)(2, 2, 2, 2)$  computed by NTCTZNN2 are plotted in Fig. 4, which shows that the neural states converge to the corresponding entries



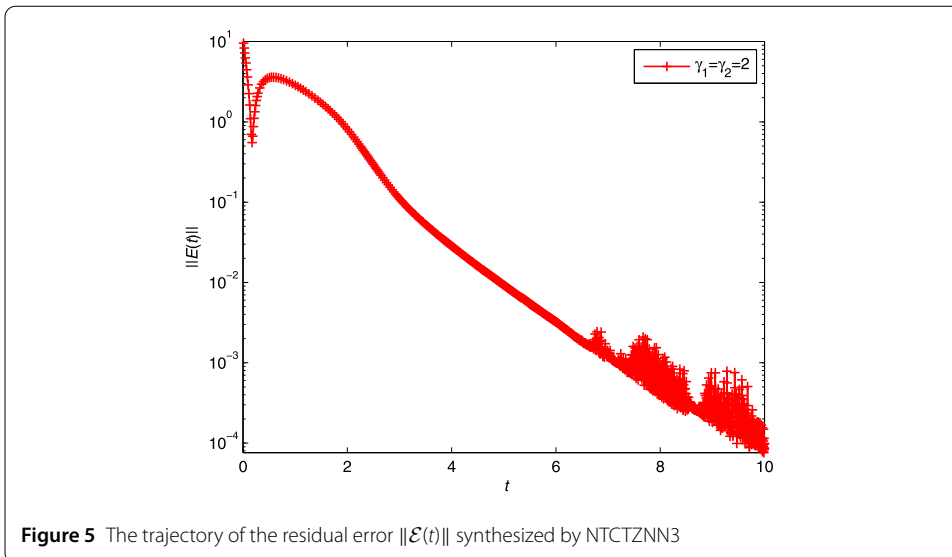
of the theoretical solution. (Here the theoretical solution  $\mathcal{X}^*(t)$  is denoted by \*-dotted blue curves, and the neural-network solutions are denoted by +-dotted red curves.)

Fourthly, we use NTCTZNN3 with  $\gamma_1 = \gamma_2 = 10$  to solve problem (28), and the activation function is set as the sign-bi-power activation function [37]:

$$f(x) = g(x) = k \left( \frac{1}{2} \text{sig}^{\tilde{\eta}}(x) + \frac{1}{2} \text{sig}^{1/\tilde{\eta}}(x) \right), \quad k > 0,$$

where  $\tilde{\eta} \in (0, 1)$  and  $\text{sig}^{\tilde{\eta}}(x)$  is defined as

$$\text{sig}^{\tilde{\eta}}(x) = \begin{cases} |x|^{\tilde{\eta}}, & \text{if } x > 0, \\ 0, & \text{if } x = 0, \\ -|x|^{\tilde{\eta}}, & \text{if } x < 0. \end{cases} = \text{sign}(x)|x|^{\tilde{\eta}}.$$



**Figure 5** The trajectory of the residual error  $\|\mathcal{E}(t)\|$  synthesized by NTCTZNN3

We set  $\tilde{\eta} = 0.5, k = 1$ . The residual error  $\|\mathcal{E}(t)\|$  generated by NTCTZNN3 with  $\gamma_1 = \gamma_2 = 2$  is displayed in Fig. 5, and the final residual error is  $9.8868 \times 10^{-5}$ . Furthermore, its number of iterations is 1097. Comparing the accuracy and the number of iterations, we can find that NTCTZNN2 is more efficient than NTCTZNN3.

Overall, taking the two criteria, i.e., accuracy and number of iterations, into consideration, we can rank the proposed four neural-networks as follows:

$$\text{NTCTZNN2} > \text{NTCTZNN3} > \text{NTCTZNN1} > \text{GNN},$$

in which  $a > b$  means that  $a$  is more efficient than  $b$ .

NTCTZNN1-3 and GNN are customized to solve TVSTEs, and they can also be exploited to solve static Sylvester tensor equations, which can be viewed as a special case of TVSTEs (i.e.,  $\dot{\mathcal{A}} = \dot{\mathcal{B}} = \dot{\mathcal{C}} = 0$ ). For simplicity, we only apply NTCTZNN3 to solve the static Sylvester tensor equations, which is given in the following example.

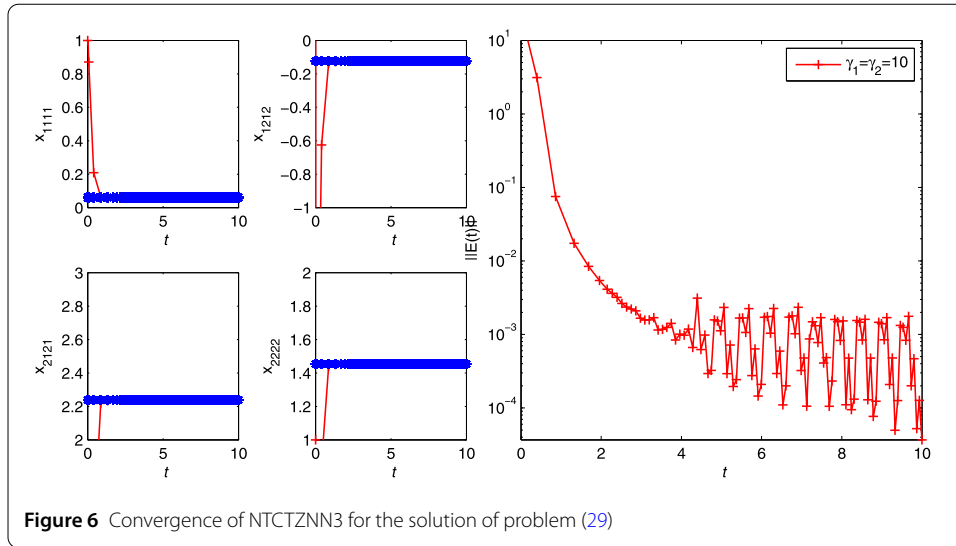
*Example 4.2* Let us consider a static Sylvester tensor equation with ( $m = n = 2, I_1 = I_2 = K_1 = K_2 = 2$ ):

$$\mathcal{A} *_n \mathcal{X} + \mathcal{X} *_m \mathcal{B} = \mathcal{C}, \tag{29}$$

where

$$\begin{aligned} \mathcal{A}(:, :, 1, 1) &= \begin{bmatrix} 1 & 1 \\ 2 & -4 \end{bmatrix}, & \mathcal{A}(:, :, 1, 2) &= \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \\ \mathcal{A}(:, :, 2, 1) &= \begin{bmatrix} 3 & 10 \\ -12 & -8 \end{bmatrix}, & \mathcal{A}(:, :, 2, 2) &= \begin{bmatrix} 6 & 9 \\ 3 & 10 \end{bmatrix}, \\ \mathcal{B}(:, :, 1, 1) &= \begin{bmatrix} 2 & -1 \\ -2 & 5 \end{bmatrix}, & \mathcal{B}(:, :, 1, 2) &= \begin{bmatrix} 5 & 3 \\ -1 & -3 \end{bmatrix}, \end{aligned}$$





**Figure 6** Convergence of NTCTZNN3 for the solution of problem (29)

$$B(:, :, 2, 1) = \begin{bmatrix} 10 & 3 \\ 1 & 2 \end{bmatrix}, \quad B(:, :, 2, 2) = \begin{bmatrix} 2 & 5 \\ 2 & 10 \end{bmatrix},$$

$$C = A.$$

We set the noise  $\eta = 1$  and the initial state tensor  $\mathcal{X}_0 = \mathcal{I}$ . The numerical results generated by NTCTZNN3 with  $\gamma_1 = \gamma_2 = 10$  are plotted in Fig. 6, and the final residual error is  $2.1473 \times 10^{-4}$ .

As shown in Fig. 6, the neural state computed by NTCTZNN3 converges to the theoretical solution. In addition, the sequence  $\{\|\mathcal{E}(t)\|\}$  of the residual errors converges to zero. These demonstrate the effectiveness of NTCTZNN3 for the static Sylvester tensor equations. Furthermore, the convergence property of GBB also needs to be investigated.

### 5 Conclusion

By following Zhang *et al.*'s design method, we have proposed three noise-tolerant continuous-time Zhang neural networks (NTCTZNNs) and a gradient-based neural network (GNN) to solve the time-varying Sylvester tensor equations with noise, and have established their various convergence results. These complement some existing results. Numerical results substantiate the efficacy and superiority of the proposed NTCTZNNs.

It is possible to extend the ideas in this paper for other type tensor equations, such as time-varying periodic Sylvester tensor equations, or time-varying coupled Sylvester tensor equations. Furthermore, we will apply the designed neural networks to realize the path-tracking control of different robot manipulators in the future.

#### Acknowledgements

The authors thank two anonymous reviewers for their valuable comments and suggestions, which have helped them in improving the paper.

#### Funding

This work is supported by the National Natural Science Foundation of China and Shandong Province (No. 11671228, 11601475, ZR2016AL05), the Doctoral Initiation Fund of Zaozhuang University and the Qingtan Scholar Project of Zaozhuang University.

#### Competing interests

The authors declare that there are no competing interests.

### Authors' contributions

The first author provided the problem and gave the proof of the main results, and the second author finished the numerical experiment. All authors read and approved the final manuscript.

### Author details

<sup>1</sup>School of Mathematics and Statistics, Zaozhuang University, Zaozhuang, P.R. China. <sup>2</sup>School of Data Sciences, Zhejiang University of Finance and Economics, Hangzhou, P.R. China.

### Publisher's Note

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

Received: 1 July 2019 Accepted: 5 November 2019 Published online: 12 November 2019

### References

1. Einstein, A.: The foundation of the general theory of relativity. In: Kox, A.J., Klein, M.J., Schulmann, R. (eds.) *The Collected Papers of Albert Einstein*, vol. 6. Princeton University Press, Princeton (2007)
2. Haussühl, S.: *Physical Properties of Crystals*. Wiley, Weinheim (2007)
3. Brazell, M., Li, N., Navasca, C., Tamon, C.: Solving multilinear systems via tensor inversion. *SIAM J. Matrix Anal. Appl.* **34**, 542–570 (2013)
4. Zhang, Y.N., Jiang, D.C., Wang, J.: A recurrent neural network for solving Sylvester equation with time-varying coefficients. *IEEE Trans. Neural Netw.* **13**(5), 1053–1063 (2002)
5. Zhou, B., Duam, G.R., Lin, Z.: A parametric periodic Lyapunov equation with application in semi-global stabilization of discrete-time periodic systems subject to actuator saturation. *Automatica* **47**, 316–325 (2011)
6. Wang, Q.W., He, Z.H.: Systems of coupled generalized Sylvester matrix equations. *Automatica* **50**(11), 2840–2844 (2014)
7. Sun, M., Wang, Y.J.: The conjugate gradient methods for solving the generalized periodic Sylvester matrix equations. *J. Appl. Math. Comput.* **60**, 413–434 (2019)
8. Sun, L.Z., Zheng, B.D., Bu, C.J., Wei, Y.M.: Moore–Penrose inverse of tensors via Einstein product. *Linear Multilinear Algebra* **64**, 686–698 (2016)
9. Li, B.W., Sun, Y.S., Zhang, D.W.: Chebyshev collocation spectral methods for coupled radiation and conduction in a concentric spherical participating medium. *J. Heat Transf.* **131**, 062701 (2009)
10. Li, B.W., Tian, S., Sun, Y.S., Mao, Z.: Schur-decomposition for 3D matrix equations and its application in solving radiative discrete ordinates equations discretized by Chebyshev collocation spectral method. *J. Comput. Phys.* **229**, 1198–1212 (2010)
11. Ding, F., Chen, T.W.: Gradient based iterative algorithms for solving a class of matrix equations. *IEEE Trans. Autom. Control* **50**(8), 1216–1221 (2005)
12. Wang, Q.W., Xu, X.J.: Iterative algorithms for solving some tensor equations. *Linear Multilinear Algebra* **67**, 1325–1349 (2019)
13. Huang, B.H., Ma, C.F.: An iterative algorithm to solve the generalized Sylvester tensor equations. *Linear Multilinear Algebra* (2018). <https://doi.org/10.1080/03081087.2018.1536732>
14. Lv, L.L., Zhang, Z., Zhang, L., Wang, W.S.: An iterative algorithm for periodic Sylvester matrix equations. *J. Ind. Manag. Optim.* **14**(1), 413–425 (2018)
15. Hajarjan, M.: Generalized conjugate direction algorithm for solving the general coupled matrix equations over symmetric matrices. *Numer. Algorithms* **73**(3), 591–609 (2016)
16. Hajarjan, M.: New finite algorithm for solving the generalized nonhomogeneous Yakubovich-transpose matrix equation. *Asian J. Control* **19**(1), 164–172 (2017)
17. Hajarjan, M.: Computing symmetric solutions of general Sylvester matrix equations via Lanczos version of biconjugate residual algorithm. *Comput. Math. Appl.* **76**(4), 686–700 (2018)
18. Lv, L.L., Zhang, Z.: On the periodic Sylvester equations and their applications in periodic Luenberger observers design. *J. Franklin Inst.* **353**(5), 1005–1018 (2016)
19. Lv, L.L., Zhang, Z.: Finite iterative solutions to periodic Sylvester matrix equations. *J. Franklin Inst.* **354**(5), 2358–2370 (2017)
20. Lv, L.L., Zhang, Z.: A parametric poles assignment algorithm for second-order linear periodic systems. *J. Franklin Inst.* **354**(8), 8057–8071 (2017)
21. Lv, L.L., Zhang, Z., Zhang, L.: A periodic observers synthesis approach for LDP systems based on iteration. *IEEE Access* **6**, 8539–8546 (2018)
22. Lv, L.L., Zhang, Z., Zhang, L., Liu, X.X.: Gradient based approach for generalized discrete-time periodic coupled Sylvester matrix equations. *J. Franklin Inst.* **355**(15), 7691–7705 (2018)
23. Hajarjan, M.: Matrix iterative methods for solving the Sylvester-transpose and periodic Sylvester matrix equations. *J. Franklin Inst.* **350**(10), 3328–3341 (2013)
24. Hajarjan, M.: Generalized reflexive and anti-reflexive solutions of the coupled Sylvester matrix equations via CD algorithm. *J. Vib. Control* **24**(2), 343–356 (2016)
25. Hajarjan, M.: Least squares solution of the linear operator equation. *J. Optim. Theory Appl.* **170**(1), 205–219 (2016)
26. Hajarjan, M.: Solving the general Sylvester discrete-time periodic matrix equations via the gradient based iterative method. *Appl. Math. Lett.* **52**, 87–95 (2016)
27. Sun, M., Wang, Y.J., Liu, J.: Two modified least-squares iterative algorithms for the Lyapunov matrix equations. *Adv. Differ. Equ.* **2019**, 305 (2019)
28. Guo, D.S., Lin, X.J., Su, Z.Z., Sun, S.B., Huang, Z.J.: Design and analysis of two discrete-time ZD algorithms for time-varying nonlinear minimization. *Numer. Algorithms* **77**(1), 23–36 (2018)
29. Sun, M., Tian, M.Y., Wang, Y.J.: Discrete-time Zhang neural networks for time-varying nonlinear optimization. *Discrete Dyn. Nat. Soc.* **4745759**, 1–14 (2019)

30. Sun, M., Wang, Y.J.: General five-step discrete-time Zhang neural network for time-varying nonlinear optimization. *Bull. Malays. Math. Soc.* (2019). <https://doi.org/10.1007/s40840-019-00770-4>
31. Zhang, Y.N., Li, Z.: Zhang neural network for online solution of time-varying convex quadratic program subject to time-varying linear-equality constraints. *Phys. Lett. A* **373**, 1639–1643 (2009)
32. Jin, L., Zhang, Y.N.: Discrete-time Zhang neural network of  $\mathcal{O}(\tau^3)$  pattern for time-varying matrix pseudoinversion with application to manipulator motion generation. *Neurocomputing* **142**, 165–173 (2014)
33. Sun, M., Liu, J.: General six-step discrete-time Zhang neural network for time-varying tensor absolute value equations. *Discrete Dyn. Nat. Soc.* (2019). Accepted
34. Brazell, M., Li, N., Navasca, C., Tamon, C.: Solving multilinear systems via tensor inversion. *SIAM J. Matrix Anal. Appl.* **34**, 542–570 (2013)
35. Guo, D.S., Zhang, Y.N.: Zhang neural network versus gradient-based neural network for time-varying linear matrix equation solving. *Neurocomputing* **74**, 3708–3712 (2011)
36. Jin, L., Zhang, Y.N.: Integration-enhanced Zhang neural network for real-time-varying matrix inversion in the presence of various kinds of noises. *IEEE Trans. Neural Netw. Learn. Syst.* **27**(12), 2615–2627 (2016)
37. Li, S., Chen, S., Liu, B.: 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**, 1–17 (2015)
38. Yi, C.F., Chen, Y.H., Lu, Z.H.: Improved gradient-based neural networks for online solution of Lyapunov matrix equation. *Inf. Process. Lett.* **111**, 780–786 (2011)
39. Ge, S.S., Lee, T.H., Harris, C.J.: *Adaptive Neural Network Control of Robotic Manipulators*. World Scientific, London (1998)

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)

---