**Mathematics**

http://www.aimspress.com/journal/Math

AIMS Mathematics, 7(2): 2266–2280.
DOI: 10.3934/math.2022129
Received: 11 August 2021
Accepted: 29 October 2021
Published: 10 November 2021

*Research article*

# Zeroing neural network model for solving a generalized linear time-varying matrix equation

**Huamin Zhang**[1,*] **and Hongcai Yin**[2]

[1] College of Information and Network Engineering, Anhui Science and Technology University, Bengbu 233030, China

[2] School of Management Science and Engineering, Anhui University of Finance and Economics, Bengbu 233000, China

* **Correspondence:** Email: zhangeasymail@126.com.

**Abstract:** The time-varying solution of a class generalized linear matrix equation with the transpose of an unknown matrix is discussed. The computation model is constructed and asymptotic convergence proof is given by using the zeroing neural network method. Using an activation function, the predefined-time convergence property and noise suppression strategy are discussed. Numerical examples are offered to illustrate the efficacy of the suggested zeroing neural network models.

**Keywords:** linear time-varying matrix equation; zeroing neural network; convergence analysis
**Mathematics Subject Classification:** 15A09, 15A24

## 1. Introduction

The matrix equation are often encountered in the control and system fields [1, 2]. For example, one should solve the Sylvester matrix equation in the fields of the analysis and synthesis dynamic control systems [3, 4]. Various strategies have been suggested to solve these linear matrix equations. In the engineering and control fields, the time-variant matrix equations are often encountered [5, 6]. For example, the state equations of the linear time-variant system are time-varying. To control the robot manipulators, we need to solve the time-varying equations in the robot kinematics [7–9]. In this manuscript, we focus our energy on how to solve the following generalized linear time-varying matrix equation:

$$\boldsymbol{A}(t)\boldsymbol{X}(t)\boldsymbol{B}(t) + \boldsymbol{C}(t)\boldsymbol{X}^{\mathrm{T}}(t)\boldsymbol{D}(t) = \boldsymbol{F}(t), \tag{1.1}$$

where $\boldsymbol{A}(t), \boldsymbol{B}(t), \boldsymbol{C}(t), \boldsymbol{D}(t), \boldsymbol{F}(t) \in \mathbb{R}^{n\times n}$ are the time-varying coefficient matrices and the $\boldsymbol{X}(t) \in \mathbb{R}^{n\times n}$ is the unknown time-varying matrix to be determined.

The zeroing neural network (shorted for ZNN) method is a very active topic in the fields of control and engineering [10, 11] and automatic control field [12–14]. This method can achieve the global and exponential convergence. Due to this important convergence property of the ZNN method, based on the Lyapunov stable theory, the ordinary different equation theory and the Laplace transform theory, three different convergence approaches were established [15–18]. The ZNN method or the improved ZNN method can be used to solve a variety of matrix inversions or matrix equations. For example, the complex matrix Drazin inverse was discussed [19], the time-varying Sylvester matrix equations were solved [20–22] and the time-varying matrix square root was determined [23]. Using the ZNN method, the QR decomposition of the complex value matrix can be done and this decomposition can be used to track a robotic motion [7, 24, 25].

Due to the exponential convergence and robustness in dealing with the time-varying equation, the ZNN method has acquired some new developments. Based on the ZNN method, the varying-parameter recurrent neural-network method for solving online time-varying matrix equations has been established [26, 27] and the super-exponential convergence was proved. This varying-parameter recurrent zeroing neural network has been succeed in dealing with the complex Sylvester matrix equation [28] and quadratic programming problems [29]. In this paper, we use the ZNN method to solve Eq (1.1). First, we construct a ZNN model to solve Eq (1.1) and the convergence analysis is offered. Second, with an activation function, the predefined-time convergence property and noise suppression model is suggested and the convergence proof is presented. Third, to illustrate the efficacy of the suggested methods, two numerical experiments are offered. The main contributions of this paper can be emphasized as follows:

- Zeroing neural network model for solving linear time-varying matrix equation (1.1) is constructed and the convergence proof is offered.
- The predefined-time convergence property is discussed and the noise suppression model is suggested.
- Two numerical experiments are given to verify the effectiveness of the suggested models.

The remainder of this paper is organized as follows: Two preliminary lemmas and the analytical time-varying solution of Eq (1.1) are presented in Section 2. Section 3 offers the ZNN model for solving this time-varying linear matrix equation (1.1). Section 4 discusses the role of the activation function and the predefined-time convergence property of the ZNN model. Two numerical experiments are offered in Section 5 to illustrate the efficacy of the suggested models. Section 6 ends this note with some concluding remarks.

The following notation is used in the paper. Symbols $\dot{\phi}(t)$ and $\frac{\mathrm{d}\phi(t)}{\mathrm{d}t}$ denote the derivative of $\phi(t)$ in the argument $t$. $\boldsymbol{I}_n$ represents an identity matrix of size $n \times n$. $\boldsymbol{O}$ denotes a zero matrix with proper size. For a real square matrix $\mathrm{tr}(\boldsymbol{A})$ represents the trace function, the Fobenious norm $\|\boldsymbol{A}\|_F$ is defined by formula $\|\boldsymbol{A}\|_F^2 = \mathrm{tr}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})$. $\boldsymbol{A} \otimes \boldsymbol{B}$ stands for the Kronecker product of matrices $\boldsymbol{A} = (a_{ij})$ and $\boldsymbol{B}$ by formula

$$\boldsymbol{A} \otimes \boldsymbol{B} = (a_{ij}\boldsymbol{B}),\ i = 1, 2, \cdots, m, j = 1, 2, \cdots, n.$$

$\mathrm{col}[\boldsymbol{A}]$ denotes the vectorization operator of matrix $\boldsymbol{A}$ by defining $\mathrm{col}[\boldsymbol{A}] = [\boldsymbol{\alpha}_1^{\mathrm{T}}, \boldsymbol{\alpha}_2^{\mathrm{T}}, \cdots, \boldsymbol{\alpha}_n^{\mathrm{T}}]^{\mathrm{T}}$, where $\boldsymbol{A} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_n]$.

## 2. Preliminary work and the analytical solution

In this section, we give two preliminary lemmas [30] and the analytical solution of Eq (1.1).

**Lemma 1.** *Let* $\boldsymbol{A}(t) \in \mathbb{R}^{m\times n}$, $\boldsymbol{B}(t) \in \mathbb{R}^{n\times p}$ *and* $\boldsymbol{C}(t) \in \mathbb{R}^{p\times q}$. *Then*

$$\mathrm{col}[\boldsymbol{A}(t)\boldsymbol{B}(t)\boldsymbol{C}(t)] = [\boldsymbol{C}^{\mathrm{T}}(t) \otimes \boldsymbol{A}(t)]\mathrm{col}[\boldsymbol{B}(t)].$$

Let $\boldsymbol{e}_{in}$ denote an $n$-dimensional unit column vector which has 1 in the $i$th position and 0's elsewhere, i.e.,

$$\boldsymbol{e}_{in} := [0, 0, \cdots, 0, 1, 0, \cdots, 0]^{\mathrm{T}}.$$

Define the vec-permutation matrix

$$\boldsymbol{P}_{mn} := \begin{bmatrix} \boldsymbol{I}_m \otimes \boldsymbol{e}_{1n}^{\mathrm{T}} \\ \boldsymbol{I}_m \otimes \boldsymbol{e}_{2n}^{\mathrm{T}} \\ \vdots \\ \boldsymbol{I}_m \otimes \boldsymbol{e}_{nn}^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{mn\times mn},$$

based on the definition of the vec-permutation matrix, we obtain the conclusion $\boldsymbol{P}_{mn}^{\mathrm{T}} = \boldsymbol{P}_{nm}$ and the following conclusion.

**Lemma 2.** *For any matrix* $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, *it can be proved that*

$$\mathrm{col}[\boldsymbol{A}_{m\times n}^{\mathrm{T}}] = \boldsymbol{P}_{nm}\mathrm{col}[\boldsymbol{A}_{m\times n}].$$

For example, let

$$\boldsymbol{A}_{2\times 3} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \text{ and } \boldsymbol{P}_{32} = \begin{bmatrix} \boldsymbol{I}_3 \otimes \boldsymbol{e}_{12}^{\mathrm{T}} \\ \boldsymbol{I}_3 \otimes \boldsymbol{e}_{22}^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

it can be verified that $\mathrm{col}\boldsymbol{A}_{23}^{\mathrm{T}} = \boldsymbol{P}_{32}\mathrm{col}\boldsymbol{A}_{23}$ is right.

Using Lemmas 1 and 2, Eq (1.1) can be rewritten as

$$\begin{aligned} &[\boldsymbol{B}^{\mathrm{T}}(t) \otimes \boldsymbol{A}(t)]\mathrm{col}[\boldsymbol{X}(t)] + [\boldsymbol{D}^{\mathrm{T}}(t) \otimes \boldsymbol{C}(t)]\mathrm{col}[\boldsymbol{X}^{\mathrm{T}}(t)] = \mathrm{col}[\boldsymbol{F}(t)], \\ &[\boldsymbol{B}^{\mathrm{T}}(t) \otimes \boldsymbol{A}(t)]\mathrm{col}[\boldsymbol{X}(t)] + [\boldsymbol{D}^{\mathrm{T}}(t) \otimes \boldsymbol{C}(t)]\boldsymbol{P}_{nn}\mathrm{col}[\boldsymbol{X}(t)] = \mathrm{col}[\boldsymbol{F}(t)], \\ &\{\boldsymbol{B}^{\mathrm{T}}(t) \otimes \boldsymbol{A}(t) + [\boldsymbol{D}^{\mathrm{T}}(t) \otimes \boldsymbol{C}(t)]\boldsymbol{P}_{nn}\}\,\mathrm{col}[\boldsymbol{X}(t)] = \mathrm{col}[\boldsymbol{F}(t)]. \end{aligned}$$

If the coefficient matrix $\boldsymbol{B}^{\mathrm{T}}(t) \otimes \boldsymbol{A}(t) + [\boldsymbol{D}^{\mathrm{T}}(t) \otimes \boldsymbol{C}(t)]\boldsymbol{P}_{nn}$ is invertible, then the exact solution of Eq (1.1) is

$$\mathrm{col}[\boldsymbol{X}(t)] = \{\boldsymbol{B}^{\mathrm{T}}(t) \otimes \boldsymbol{A}(t) + [\boldsymbol{D}^{\mathrm{T}}(t) \otimes \boldsymbol{C}(t)]\boldsymbol{P}_{nn}\}^{-1}\,\mathrm{col}[\boldsymbol{F}(t)]. \tag{2.1}$$

This conclusion will be used in the numerical experiments section.

## 3. ZNN model for the time-varying linear matrix equation

In this section, we use the ZNN method to design a model to solve Eq (1.1). We use $\boldsymbol{X}(t)$ to denote the neural state solution and introduce the time-varying error function matrix $\boldsymbol{\Phi}(t)$ by defining

$$\boldsymbol{\Phi}(t) := \boldsymbol{A}(t)\boldsymbol{X}(t)\boldsymbol{B}(t) + \boldsymbol{C}(t)\boldsymbol{X}^{\mathrm{T}}(t)\boldsymbol{D}(t) - \boldsymbol{F}(t). \tag{3.1}$$

The ZNN method for solving Eq (1.1) is

$$\frac{\mathrm{d}\boldsymbol{\Phi}(t)}{\mathrm{d}t} = -\gamma\boldsymbol{\Phi}(t), \tag{3.2}$$

where $\gamma > 0$ is a design parameter. To reduce the convergence time and improve the convergence performance, an activation function version ZNN model is

$$\frac{\mathrm{d}\boldsymbol{\Phi}(t)}{\mathrm{d}t} = -\gamma\mathcal{F}(\boldsymbol{\Phi}(t)), \tag{3.3}$$

where $\mathcal{F}(\boldsymbol{\Phi}(t))$ represents an activation function array with $f(\phi_{ij}(t))$ as its entries. Ordinary speaking, the activation function $f(*)$ is monotonous increasing odd function. Next, we establish ZNN model for solving Eq (1.1).

Taking the derivative of $\boldsymbol{\Phi}(t)$ gives

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{\Phi}(t)}{\mathrm{d}t} =& \frac{\mathrm{d}\boldsymbol{A}(t)}{\mathrm{d}t}\boldsymbol{X}(t)\boldsymbol{B}(t) + \boldsymbol{A}(t)\frac{\mathrm{d}\boldsymbol{X}(t)}{\mathrm{d}t}\boldsymbol{B}(t) + \boldsymbol{A}(t)\boldsymbol{X}(t)\frac{\mathrm{d}\boldsymbol{B}(t)}{\mathrm{d}t} + \frac{\mathrm{d}\boldsymbol{C}(t)}{\mathrm{d}t}\boldsymbol{X}^{\mathrm{T}}(t)\boldsymbol{D}(t)\\ &+\boldsymbol{C}(t)\frac{\mathrm{d}\boldsymbol{X}^{\mathrm{T}}(t)}{\mathrm{d}t}\boldsymbol{D}(t) + \boldsymbol{C}(t)\boldsymbol{X}^{\mathrm{T}}(t)\frac{\mathrm{d}\boldsymbol{D}(t)}{\mathrm{d}t} - \frac{\mathrm{d}\boldsymbol{F}(t)}{\mathrm{d}t}.\end{aligned} \tag{3.4}$$

Substituting Eqs (3.1) and (3.4) into Eq (3.2) gives

$$\begin{aligned}&\frac{\mathrm{d}\boldsymbol{A}(t)}{\mathrm{d}t}\boldsymbol{X}(t)\boldsymbol{B}(t) + \boldsymbol{A}(t)\frac{\mathrm{d}\boldsymbol{X}(t)}{\mathrm{d}t}\boldsymbol{B}(t) + \boldsymbol{A}(t)\boldsymbol{X}(t)\frac{\mathrm{d}\boldsymbol{B}(t)}{\mathrm{d}t} + \frac{\mathrm{d}\boldsymbol{C}(t)}{\mathrm{d}t}\boldsymbol{X}^{\mathrm{T}}(t)\boldsymbol{D}(t)\\ &+\boldsymbol{C}(t)\frac{\mathrm{d}\boldsymbol{X}^{\mathrm{T}}(t)}{\mathrm{d}t}\boldsymbol{D}(t) + \boldsymbol{C}(t)\boldsymbol{X}^{\mathrm{T}}(t)\frac{\mathrm{d}\boldsymbol{D}(t)}{\mathrm{d}t} - \frac{\mathrm{d}\boldsymbol{F}(t)}{\mathrm{d}t}\\ =&-\gamma\left[\boldsymbol{A}(t)\boldsymbol{X}(t)\boldsymbol{B}(t) + \boldsymbol{C}(t)\boldsymbol{X}^{\mathrm{T}}(t)\boldsymbol{D}(t) - \boldsymbol{F}(t)\right].\end{aligned} \tag{3.5}$$

With a proper manipulation gives

$$\boldsymbol{A}(t)\frac{\mathrm{d}\boldsymbol{X}(t)}{\mathrm{d}t}\boldsymbol{B}(t) + \boldsymbol{C}(t)\frac{\mathrm{d}\boldsymbol{X}^{\mathrm{T}}(t)}{\mathrm{d}t}\boldsymbol{D}(t) \tag{3.6}$$

$$\begin{aligned}=& -\gamma\left[\boldsymbol{A}(t)\boldsymbol{X}(t)\boldsymbol{B}(t) + \boldsymbol{C}(t)\boldsymbol{X}^{\mathrm{T}}(t)\boldsymbol{D}(t) - \boldsymbol{F}(t)\right]\\ &-\left\{\frac{\mathrm{d}\boldsymbol{A}(t)}{\mathrm{d}t}\boldsymbol{X}(t)\boldsymbol{B}(t) + \boldsymbol{A}(t)\boldsymbol{X}(t)\frac{\mathrm{d}\boldsymbol{B}(t)}{\mathrm{d}t} + \frac{\mathrm{d}\boldsymbol{C}(t)}{\mathrm{d}t}\boldsymbol{X}^{\mathrm{T}}(t)\boldsymbol{D}(t)\right.\\ &\left.+\boldsymbol{C}(t)\boldsymbol{X}^{\mathrm{T}}(t)\frac{\mathrm{d}\boldsymbol{D}(t)}{\mathrm{d}t} - \frac{\mathrm{d}\boldsymbol{F}(t)}{\mathrm{d}t}\right\}.\end{aligned} \tag{3.7}$$

Setting

$$\boldsymbol{\Psi}(t) := \frac{\mathrm{d}\boldsymbol{A}(t)}{\mathrm{d}t}\boldsymbol{X}(t)\boldsymbol{B}(t) + \boldsymbol{A}(t)\boldsymbol{X}(t)\frac{\mathrm{d}\boldsymbol{B}(t)}{\mathrm{d}t} + \frac{\mathrm{d}\boldsymbol{C}(t)}{\mathrm{d}t}\boldsymbol{X}^{\mathrm{T}}(t)\boldsymbol{D}(t) + \boldsymbol{C}(t)\boldsymbol{X}^{\mathrm{T}}(t)\frac{\mathrm{d}\boldsymbol{D}(t)}{\mathrm{d}t} - \frac{\mathrm{d}\boldsymbol{F}(t)}{\mathrm{d}t}$$

and using the formula

$$\mathrm{col}[\boldsymbol{A}(t)\boldsymbol{X}(t)\boldsymbol{B}(t)] = [\boldsymbol{B}^{\mathrm{T}}(t) \otimes \boldsymbol{A}(t)]\mathrm{col}[\boldsymbol{X}(t)],$$

Eq (3.6) can be rewritten as

$$[\boldsymbol{B}^{\mathrm{T}}(t) \otimes \boldsymbol{A}(t)]\mathrm{col}\left[\frac{\mathrm{d}\boldsymbol{X}(t)}{\mathrm{d}t}\right] + [\boldsymbol{D}^{\mathrm{T}}(t) \otimes \boldsymbol{C}(t)]\mathrm{col}\left[\frac{\mathrm{d}\boldsymbol{X}^{\mathrm{T}}(t)}{\mathrm{d}t}\right] = -\gamma\mathrm{col}[\boldsymbol{\Phi}(t)] - \mathrm{col}[\boldsymbol{\Psi}(t)]. \tag{3.8}$$

Referring to Lemma 2, Eq (3.8) can be rewritten as

$$[\boldsymbol{B}^{\mathrm{T}}(t) \otimes \boldsymbol{A}(t) + (\boldsymbol{D}^{\mathrm{T}}(t) \otimes \boldsymbol{C}(t))\, \boldsymbol{P}_{nn}]\mathrm{col}\left[\frac{\mathrm{d}\boldsymbol{X}(t)}{\mathrm{d}t}\right] = -\gamma\mathrm{col}[\boldsymbol{\Phi}(t)] - \mathrm{col}[\boldsymbol{\Psi}(t)]. \tag{3.9}$$

The activation function version ZNN model (3.9) is

$$[\boldsymbol{B}^{\mathrm{T}}(t) \otimes \boldsymbol{A}(t) + (\boldsymbol{D}^{\mathrm{T}}(t) \otimes \boldsymbol{C}(t))\, \boldsymbol{P}_{nn}]\mathrm{col}\left[\frac{\mathrm{d}\boldsymbol{X}(t)}{\mathrm{d}t}\right] = -\gamma\mathcal{F}\left\{\mathrm{col}[\boldsymbol{\Phi}(t)]\right\} - \mathrm{col}[\boldsymbol{\Psi}(t)]. \tag{3.10}$$

The convergence proof of models (3.9) and (3.10) is given by the following theorem.

**Theorem 1.** *Suppose that the design parameter $\gamma$ is a positive number, ZNN model (3.10) is global asymptotic convergent.*

*Proof.* Obviously, ZNN model equation (3.9) is a special case of Eq (3.10) when the activation function is the linear function.

We use the Lyapunov stable theory to prove the global asymptotic convergence of model (3.10). By introducing the Lyapunov energy function $\boldsymbol{\epsilon}(t) := \frac{1}{2}\|\boldsymbol{\Phi}(t)\|_F^2$. Due to the definitions of the Frobenius norm and the matrix trace, we have

$$\begin{aligned}\boldsymbol{\epsilon}(t) &= \frac{1}{2}\|\boldsymbol{\Phi}(t)\|_F^2 = \frac{1}{2}\left[\phi_{11}^2(t) + \phi_{12}^2(t) + \cdots + \phi_{1n}^2 + \cdots + \phi_{nn}^2(t)\right] \\ &= \frac{1}{2}\mathrm{tr}\left[\boldsymbol{\Phi}^{\mathrm{T}}(t)\boldsymbol{\Phi}(t)\right].\end{aligned}$$

To calculate the derivative of the energy function $\boldsymbol{\epsilon}(t)$, we get

$$\begin{aligned}\dot{\boldsymbol{\epsilon}}(t) &= [\phi_{11}(t)\dot{\phi}_{11}(t) + \phi_{12}(t)\dot{\phi}_{12}(t) + \cdots + \phi_{1n}(t)\dot{\phi}_{1n}(t) + \cdots + \phi_{nn}(t)\dot{\phi}_{nn}(t)] \\ &= \mathrm{tr}\left[\boldsymbol{\Phi}^{\mathrm{T}}(t)\frac{\mathrm{d}\boldsymbol{\Phi}(t)}{\mathrm{d}t}\right].\end{aligned} \tag{3.11}$$

Using ZNN model $\frac{\mathrm{d}\boldsymbol{\Phi}(t)}{\mathrm{d}t} = -\gamma\mathcal{F}(\boldsymbol{\Phi}(t))$ in Eq (3.11) gives

$$\begin{aligned}\dot{\boldsymbol{\epsilon}}(t) &= \mathrm{tr}\left[\boldsymbol{\Phi}^{\mathrm{T}}(t)\frac{\mathrm{d}\boldsymbol{\Phi}(t)}{\mathrm{d}t}\right] = -\gamma\mathrm{tr}\left[\boldsymbol{\Phi}^{\mathrm{T}}(t)\mathcal{F}(\boldsymbol{\Phi}(t))\right] \\ &= -\gamma\sum_{i=1}^{n}\sum_{j=1}^{n}\phi_{ij}(t)f[\phi_{ij}(t)].\end{aligned}$$

Because the entries $f[\phi_{ij}(t)]$, $i, j = 1, 2, \cdots, n$ are the continuous monotonous increasing odd functions, we get

$$\phi_{ij}(t)f[\phi_{ij}(t)] > 0 \ \ \forall\phi_{ij}(t) \neq 0, \ , \ i, j = 1, 2, \cdots, n.$$

Due to the Lyapunov stability theory [31–33], if $\|\boldsymbol{\Phi}(t)\|_F = 0$ then the neural state matrix $\boldsymbol{X}(t)$ is the exact solution of Eq (3.10). If $\|\boldsymbol{\Phi}(t)\|_F > 0$ then $\dot{\boldsymbol{\epsilon}}(t) < 0$ and this denotes that $\|\boldsymbol{\Phi}(t)\|_F$ monotonically decreasing and converges to the global asymptotic stable point. That is, we have $\|\boldsymbol{\Phi}(t)\|_F^2 = 0$ or the neural state matrix $\boldsymbol{X}(t)$ will converge to the exact solution of Eq (3.10). It means that ZNN model (3.10) is global asymptotic convergent. Note that model (3.9) is the special case of model (3.10). We complete the proof. □

Finding the solutions of matrix equations and coupled matrix equations are very related to deriving the identification algorithms of dynamical systems for their parameter estimation [34–36]. For some identification problems, minimizing a criterion function about the differences between the system outputs and the model outputs leads to a matrix equation. Based on the optimization techniques and the obtained matrix equations about the parameter estimation, we can derive the gradient-based identification algorithms, the least squares identification algorithms and the Newton identification algorithms. In the numerical section, we will illustrate the effectiveness of model (3.9) by numerical experiments.

## 4. Activation function and predefined-time convergence

In the practical control work, the noise should be suppressed [37, 38]. To suppress the noise, some activation functions were introduced [39, 40] and the related theoretical analysis were presented. In this section, we use an activation function to obtain the predefined-time time convergence property [8, 9] and to reduce the influence of the noise. We mainly refer to [39, 40] in reasoning the following conclusion, for more detailed discussion and more effective conclusion, one can refer to [41–44]. Referring to [39], we have the following result.

**Lemma 3.** Let $\alpha > 0, \beta > 0$, $0 < p < 1$ and $q > 1$ be some constant parameters, the following dynamic nonlinear inequality system

$$\dot{V}(t) \leqslant -\alpha V^p(\boldsymbol{x}(t)) - \beta V^q(\boldsymbol{x}(t))$$

is globally predefined-time stable and the setting time is bounded by

$$T_{\max} = \frac{1}{\alpha(1-p)} + \frac{1}{\beta(q-1)}.$$

The following universal activation function was first presented in [39].

$$f(x) = (k_1|x|^p + k_2|x|^q) \times \mathrm{sgn}(x) + k_3 x + k_4 \mathrm{sgn}(x),\ 0 < p < 1, q > 1, k_1, k_2 > 0, k_3, k_4 \geqslant 0. \tag{4.1}$$

Here, $\mathrm{sgn}(x)$ denotes the signal function. Using this activation function, the predefined-time ZNN model can be obtained.

In the practical work, some different noises can be emerged in the system, a noise version of Eq (3.10) is

$$[\boldsymbol{B}^{\mathrm{T}}(t) \otimes \boldsymbol{A}(t) + (\boldsymbol{D}^{\mathrm{T}}(t) \otimes \boldsymbol{C}(t))\,\boldsymbol{P}_{nn}]\mathrm{col}\left[\frac{\mathrm{d}\boldsymbol{X}(t)}{\mathrm{d}t}\right] = -\gamma\mathcal{F}\{\mathrm{col}[\boldsymbol{\Phi}(t)]\} - \mathrm{col}[\boldsymbol{\Psi}(t)] + \mathrm{co}[\boldsymbol{Y}(t)].$$

Here, the matrix $\boldsymbol{Y}(t)$ represents the noise matrix. Referring to [39] and using the universal activation function (4.1), one can prove that ZNN model (4.2) is effective and the predefined-time is bounded by

$$\frac{1}{\gamma k_1(1-p)} + \frac{1}{\gamma k_2(q-1)}.$$

**Theorem 2.** Suppose that the exact solution matrix $X^*(t)$ is derivable and the noise matrix $Y(t) = [y_{ij}(t)] \in \mathbb{R}^{n\times n}$ satisfies $|y_{ij}(t)| \leqslant \delta$, where $\delta \in (0, +\infty)$. If the universal activation function is used with $\gamma k_4 \geqslant \delta$ then for any initial neural state solution matrix $X(0)$ the neural state stable matrix $X(t)$ of model (4.2) will converge to the exact solution $X^*(t)$ of Eq (1.1) in predefined-time $t_c$. Here, $t_c$ satisfies

$$t_c \leqslant \frac{1}{\gamma k_1(1-p)} + \frac{1}{\gamma k_2(q-1)}.$$

*Proof.* Using error function (3.1) and activation function (4.1), the noise-perturbed ZNN model

$$\dot{\mathbf{\Phi}}(t) = -\gamma\mathcal{F}(\mathbf{\Phi}(t)) + Y(t)$$

entry-wisely consists $n^2$ sub-autonomous systems

$$\dot{\phi}_{ij}(t) = -\gamma f(\phi_{ij}(t)) + y_{ij}(t),\ i, j = 1, 2, \cdots, n, \tag{4.2}$$

where $\dot{\phi}_{ij}(t)$, $\phi_{ij}(t)$ and $y_{ij}(t)$ are the $ij$th entry of matrix $\dot{\mathbf{\Phi}}(t)$, $\mathbf{\Phi}(t)$ and $Y(t)$ respectively.

Setting the Lyapunov energy function as $u(t) = |\phi_{ij}(t)|^2$ for the sub-autonomous system (4.2) and computing the time derivative of $u(t)$ gives

$$\begin{aligned}\dot{u}(t) &= 2\phi_{ij}(t)\dot{\phi}_{ij}(t) = 2\phi_{ij}(t)\left[-\gamma f(\phi_{ij}(t) + y_{ij}(t)\right]\\ &= -2\gamma\phi_{ij}(t)f(\phi_{ij}(t) + 2\phi_{ij}(t)y_{ij}(t).\end{aligned} \tag{4.3}$$

Substituting Eq (4.1) into Eq (4.3) and with proper operations gives

$$\begin{aligned}\dot{u}(t) &= -2\gamma\phi_{ij}(t)f[\phi_{ij}(t)] + 2\phi_{ij}(t)y_{ij}(t)\\ &= -2\gamma\phi_{ij}(t)[(k_1|\phi_{ij}(t)|^p + k_2|\phi_{ij}(t)|^q)\times \operatorname{sgn}(\phi_{ij}(t))\\ &\quad +k_3\phi_{ij}(t) + k_4\operatorname{sgn}(\phi_{ij}(t))] + 2\phi_{ij}(t)y_{ij}(t)\\ &= -2\gamma\left[k_1|\phi_{ij}(t)|^{p+1} + k_2|\phi_{ij}(t)|^{q+1}\right] - 2\gamma k_3|\phi_{ij}(t)|^2 + 2\left[\phi_{ij}(t)y_{ij}(t) - \gamma k_4|\phi_{ij}(t)|\right]\\ &\leqslant -2\gamma\left[k_1|\phi_{ij}(t)|^{p+1} + k_2|\phi_{ij}(t)|^{q+1}\right] + 2\left[|\phi_{ij}(t)||y_{ij}(t)| - \gamma k_4|\phi_{ij}(t)|\right]\\ &\leqslant -2\gamma\left[k_1|\phi_{ij}(t)|^{p+1} + k_2|\phi_{ij}(t)|^{q+1}\right] + 2\left[\delta|\phi_{ij}(t)| - \gamma k_4|\phi_{ij}(t)|\right]\\ &\leqslant -2\gamma\left[k_1|\phi_{ij}(t)|^{p+1} + k_2|\phi_{ij}(t)|^{q+1}\right] + 2(\delta - \gamma k_4)|\phi_{ij}(t)|\\ &\leqslant -2\gamma\left[k_1|\phi_{ij}(t)|^{p+1} + k_2|\phi_{ij}(t)|^{q+1}\right]\\ &= -2\gamma[k_1u^{\frac{p+1}{2}}(t) + k_2u^{\frac{q+1}{2}}(t)].\end{aligned} \tag{4.4}$$

Using Lemma 3, we get the convergence time of the $ij$th sub-system (4.2) is

$$t_{ij} \leqslant \frac{1}{2\gamma k_1\left(1 - \frac{p+1}{2}\right)} + \frac{1}{2\gamma k_2\left(\frac{q+1}{2} - 1\right)} = \frac{1}{\gamma k_1(1-p)} + \frac{1}{\gamma k_2(q-1)}. \tag{4.5}$$

Because the time $t_{ij}$ depends on the parameter $\gamma$ of the ZNN model and the parameters $k_1, k_2, p, q$ of the activation function and does not depend on any initial value $X(0)$, the convergence time of model (4.2) is

$$t_c \leqslant \frac{1}{\gamma k_1(1-p)} + \frac{1}{\gamma k_2(q-1)}. \tag{4.6}$$

Thus, we complete the proof.

Referring to [39], a similar work can prove that ZNN model (3.10) has the same convergence time upper bound, we omit it here. Theorem 2 shows that the predefined-time is independent of the initial values for a class of systems. We will verify this fact in the numerical section.

## 5. Numerical examples

In this section, we verify the effectiveness of the suggested ZNN models by giving two numerical examples. The strategy of choosing the coefficient matrices is inspired by [20, 26].

**Example 1.** We use ZNN model (3.9) to solve the time-varying matrix equation (1.1). We take

$$\begin{aligned}
&\boldsymbol{A}(t) = \begin{bmatrix} \sin(t)+3 & -\cos(t) \\ \cos(t) & \sin(t)+5 \end{bmatrix}, \quad \boldsymbol{B}(t) = \begin{bmatrix} \sin(t)+2 & \cos(t) \\ -\cos(t) & \sin(t)+3 \end{bmatrix}, \\
&\boldsymbol{C}(t) = \begin{bmatrix} \sin(t) & -\cos(t) \\ \cos(t) & \sin(t) \end{bmatrix}, \quad \boldsymbol{D}(t) = \boldsymbol{X}^*(t) = \begin{bmatrix} \sin(t) & \cos(t) \\ -\cos(t) & \sin(t) \end{bmatrix}, \\
&\boldsymbol{F}(t) := \boldsymbol{A}(t)\boldsymbol{X}(t)\boldsymbol{B}(t) + \boldsymbol{C}(t)[\boldsymbol{X}^*(t)]^{\mathrm{T}}\boldsymbol{D}(t)
\end{aligned}$$

as the coefficient matrices.

In this numerical experiment, we take the design parameter $\gamma = 10$ and the initial neural state matrix $\boldsymbol{X}(0) = i \times \boldsymbol{I}_2$, $i = 1, 2, 3$, respectively, and use model (3.9) to solve time-varying nonlinear matrix equation (1.1).

To illustrate the convergence trajectory, we use

$$\boldsymbol{X}(t) = \begin{bmatrix} x_{11}(t) & x_{12}(t) \\ x_{21}(t) & x_{22}(t) \end{bmatrix}$$

to denote the neural state solution, and the convergence trajectories of $x_{ij}(t)$ are shown in Figure 1. The red curves denote trajectories of the true solution matrix $\boldsymbol{X}^*(t)$. We use

$$\delta_1(t) := \|\boldsymbol{A}(t)\boldsymbol{X}(t)\boldsymbol{B}(t) + \boldsymbol{C}(t)\boldsymbol{X}^{\mathrm{T}}(t)\boldsymbol{D}(t) - \boldsymbol{F}(t)\|_F$$

to denote the Frobenius norm of the absolute error matrix, the trajectory of $\ln \delta_1(t)$ is shown in Figure 2. These two figures show that ZNN model (3.9) is efficacious for solving nonlinear matrix equation (1.1).

From Figure 1, we find that the trajectories of $x_{ij}(t)$ converge fast and neural state matrix $\boldsymbol{X}(t)$ converges to the time-varying solution matrix $\boldsymbol{X}^*(t)$ exactly. Figure 2 shows that $\ln \delta_1(t)$ fluctuates between $-6$ and $-8$ as the time $t$ greater than 1 second. In the numerical experiment, if we set $\gamma = 100$, then convergence precision will be improved greatly. This shows that the convergence behavior of the absolute error norm $\delta_1(t)$ will be dominated by some random factors when it converges to a certain degree. How to explain the perturbation behavior of this stage requires further study.

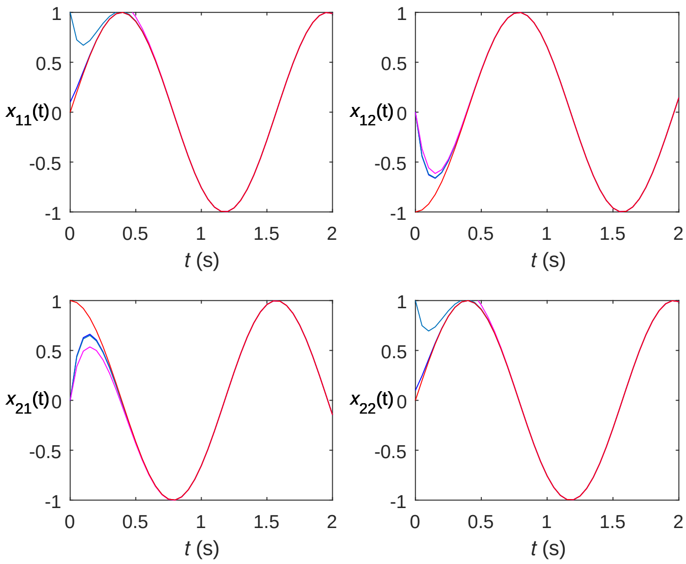


**Figure 1.** The convergence of model (3.8).

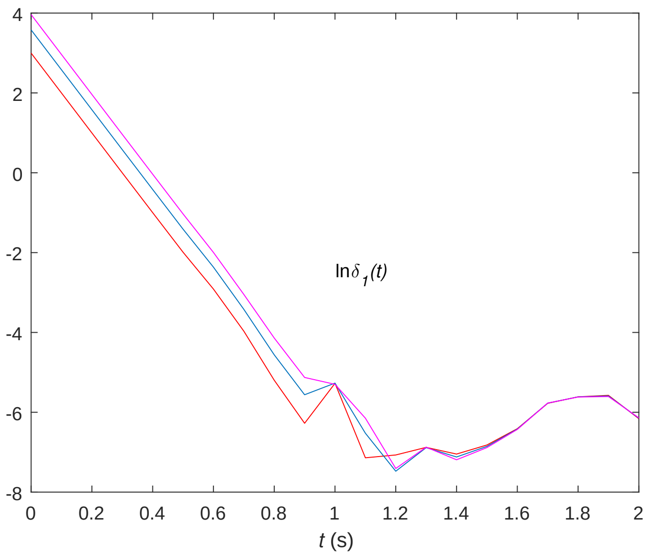


**Figure 2.** $\ln\delta_1(t)$ versus time $t$.

To illustrate the effectiveness of the noise-perturbed ZNN model (4.2), we set $\boldsymbol{Y}(t) = \sin(t)\boldsymbol{I}_2$ as the noise matrix and use model (4.2) to solve Eq (1.1). Here, we set the related parameters as

$$k_1 = k_2 = k_3 = k_4 = 1, \;\; p = 0.5, \;\; q = 2.$$

Using these parameters, the upper bound of the predefined-time convergence time is

$$T_{\max} = \frac{1}{\gamma k_1(1-p)} + \frac{1}{\gamma k_2(q-1)} = \frac{1}{10(1-0.5)} + \frac{1}{10(2-1)} = 0.3.$$

The convergence index $\delta_1(t)$ is shown in Figure 3. Figure 3 shows that the theoretical predefined convergence time conforms to the practical convergence time. Although the predefined convergence time is 0.3 second, on the computer testing, we find that there much more time are consumed due to the computation of the activation function.

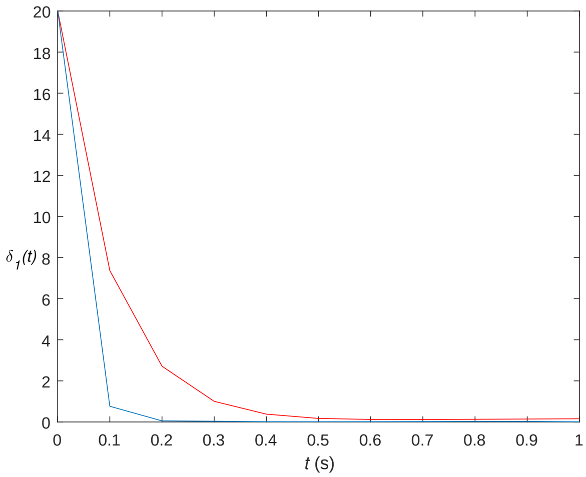


**Figure 3.** The convergence of models (4.2) and (3.9).

**Example 2.** Set $\boldsymbol{F}(t) = \begin{bmatrix} \sin(t)+9 & 0 \\ 0 & \sin(t)+10 \end{bmatrix}$. The other initial conditions are the same as Example 1, we use ZNN model (3.9) to solve Eq (1.1). The convergence performance of the neural state matrix $\boldsymbol{X}(t)$ is shown in Figure 4. We use $\delta_2(t)$ to denote the absolute error Frobenius matrix norm of this example and $\delta_2(t)$ shares the same definition of $\delta_1(t)$, the natural logarithm of the absolute error Frobenius matrix norm $\ln \delta_2(t)$ versus the time $t$ is shown in Figure 5.

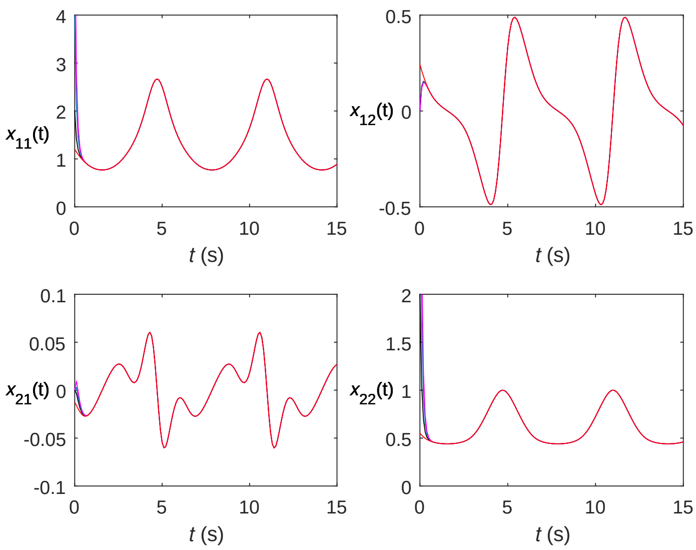


**Figure 4.** The convergence of model (3.9).

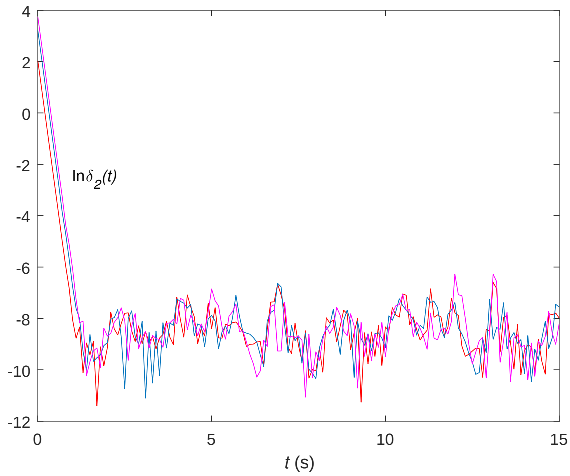


**Figure 5.** $\ln \delta_2(t)$ versus time $t$.

From Figure 4, we find that every entries of the neural state matrix $\boldsymbol{X}(t)$ converge to some fixed time-varying functions fast. This means the neural state matrix $\boldsymbol{X}(t)$ converge to the solution matrix $\boldsymbol{X}^*(t)$. From Figure 5, we find that the absolute error logarithm goes to $-8$ fast about 2 seconds. These shows that the ZNN model (3.9) is effective in solving the time-varying linear matrix equation (1.1).

## 6. Conclusions

To solve the linear time-varying matrix equation $\boldsymbol{A}(t)\boldsymbol{X}(t)\boldsymbol{B}(t) + \boldsymbol{C}(t)\boldsymbol{X}^{\mathrm{T}}(t)\boldsymbol{D}(t) = \boldsymbol{F}(t)$, the zeroing neural network model is constructed. The convergence proof is given and the predefined-time convergence property and noise suppression technique are discussed. Two numerical examples are offered to illustrate the efficacy of the suggested models. Although we have proven that the Frobenius norm of the absolute error matrix is monotonic decreasing, the numerical experiments show that the convergence behavior will be dominated by some random factors when the absolute error norm decreasing to certain degree. We will study this fluctuation in the future. The proposed algorithms in this paper can combine other estimation methods to explore new iterative solution algorithms of some linear and nonlinear matrix equations and of the parameter estimation for dynamical systems and can be applied to other literatures such as signal processing and system modeling.

## Acknowledgments

This work was supported by Natural Science Foundation of Anhui Provincial Education Department (KJ2020A0063) and Talent Introduction Projection of Anhui Science and Technology University (XWYJ201809).

## Conflict of interest

The authors declare there is no conflict of interest.

## References


1. J. Q. Gong, J. Jin, A better robustness and fast convergence zeroing neural network for solving dynamic nonlinear equations, *Neural Comput. Appl.*, 2021, 1–11. doi:10.1007/s00521-020-05617-9.
2. L. Xie, J. Ding, F. Ding, Gradient based iterative solutions for general linear matrix equations, *Comput. Math. Appl.*, **58** (2009), 1441–1448. doi: 10.1016/j.camwa.2009.06.047.
3. Z. N. Zhang, F. Ding, X. G. Liu, Hierarchical gradient based iterative parameter estimation algorithm for multivariable output error moving average systems, *Comput. Math. Appl.*, **61** (2011), 672–682. doi: 10.1016/j.camwa.2010.12.014.
4. F. Ding, Combined state and least squares parameter estimation algorithms for dynamic systems, *Appl. Math. Modell.*, **38** (2014), 403–412. doi: 10.1016/j.apm.2013.06.007.
5. H. M. Zhang, Quasi gradient-based inversion-free iterative algorithm for solving a class of the nonlinear matrix equations, *Comput. Math. Appl.*, **77** (2019), 1233–1244. doi: 10.1016/j.camwa.2018.11.006.
6. H. M. Zhang, L. J. Wan, Zeroing neural network methods for solving the Yang-Baxter-like matrix equation, *Neurocomputing*, **383** (2020), 409–418. doi: 10.1016/j.neucom.2019.11.101.
7. V. N. Katsikis, S. D. Mourtas, P. S. Stanimirović, Y. N. Zhang, Solving complex-valued time-varying linear matrix equations via QR decomposition with applications to robotic motion tracking and on angle-of-arrival localization, *IEEE T. Neur. Net. Lear.*, 2021, 1–10. doi: 10.1109/TNNLS.2021.3052896.
8. L. Xiao, Z. J. Zhang, S. Li, Solving time-varying system of nonlinear equations by finite-time recurrent neural networks with application to motion tracking of robot manipulators, *IEEE T. Syst. Man Cy.-S.*, **49** (2019), 2210–2220. doi: 10.1109/TSMC.2018.2836968.
9. L. Xiao, S. Li, K. L. Li, L. Jin, B. L. Liao, Co-design of finite-time convergence and noise suppression: A unified neural model for time varying linear equations with robotic applications, *IEEE T. Syst. Man Cy.-S.*, **50** (2020), 5233–5243. doi: 10.1109/TSMC.2018.2870489.
10. L. Jin, S. Li, B. L. Liao, Z. J. Zhang, Zeroing neural networks: A survey, *Neurocomputing*, **267** (2017), 597–604. doi: 10.1016/j.neucom.2017.06.030.
11. Z. B. Sun, T. Shi, L. Jin, B. C. Zhang, Z. X. Pang, J. Z. Yu, Discrete-time zeroing neural network of $O(\tau^4)$ pattern for online time-varying nonlinear optimization problem: Application to manipulator motion generation, *J. Frankl. I.*, **358** (2021), 7203–7220. doi: 10.1016/j.jfranklin.2021.07.006.
12. L. Jin, Y. N. Zhang, Discrete-time Zhang neural network for online time-varying nonlinear optimization with application to manipulattor motion generation, *IEEE T. Neur. Net. Lear.*, **26** (2015), 1525–1531. doi: 10.1109/TNNLS.2014.2342260.
13. L. Jin, S. Li, Distributed task allocation of multiple robots: A control perspective, *IEEE T. Syst. Man Cy.-S.*, **48** (2018), 693–701. doi: 10.1109/TSMC.2016.2627579.
14. Y. M. Qi, L. Jin, Y. N. Wang, L. Xiao, J. L. Zhang, Complex-valued discrete-time neural dynamics for perturbed time-dependent complex quadratic programming with applications, *IEEE T. Neur. Net. Lear.*, **31** (2020), 3555–3569. doi: 10.1109/TNNLS.2019.2944992.

15. L. Jin, Y. N. Zhang, Continuous and discrete Zhang dynamics for real-time varying nonlinear optimization, *Numer. Algorithms*, **73** (2016), 115–140. doi: 10.1007/s11075-015-0088-1.

16. L. Jin, Y. N. Zhang, S. Li, Y. Y. Zhang, Noise-tolerant ZNN models for solving time-varying zero-finding problems: A control-theoretic approach, *IEEE T. Automat. Contr.*, **62** (2017), 992–997. doi: 10.1109/TAC.2016.2566880.

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

18. Z. B. Sun, F. Li, B. C. Zhang, Y. Y. Sun, L. Jin, Different modified zeroing neural dynamics with inherent tolerance to noises for time-varying reciprocal problems: A control-theoretic approach, *Neurocomputing*, **337** (2019), 165–179. doi: 10.1016/j.neucom.2019.01.064.

19. S. Z. Qiao, X. Z. Wang, Y. M. Wei, Two finite-time convergent Zhang neural network models for time-varying complex matrix Drazin inverse, *Linear Algebra Appl.*, **542** (2018), 101–117. doi: 10.1016/j.laa.2017.03.014.

20. Z. Li, Y. N. Zhang, Improved Zhang neural network model and its solution of time-varying generalized linear matrix equations, *Expert Syst. Appl.*, **37** (2010), 7213–7218. doi: 10.1016/j.eswa.2010.04.007.

21. S. Li, S. F. Chen, B. Liu, 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** (2013), 189–205. doi: 10.1007/s11063-012-9241-1.

22. Y. J. Shen, P. Miao, Y. H. Huang, Y. Shen, Finite-time stability and its application for solving time-varying Sylvester equation by recurrent neural network, *Neural Process. Lett.*, **42** (2015), 763–784. doi: 10.1007/s11063-014-9397-y.

23. L. Xiao, Accelerating a recurrent neural network to finite-time convergence using a new design formula and its application to time-varying matrix square root, *J. Frankl. I.*, **354** (2017), 5667–5677. doi: 10.1016/j.jfranklin.2017.06.012.

24. L. Jia, L. Xiao, J. H. Dai, Z. H. Qi, Z. J. Zhang, Y. S. Zhang, Design and application of an adaptive fuzzy control strategy to zeroing neural network for solving time-variant QP problem, *IEEE T. Fuzzy Syst.*, **29** (2021), 1544–1555. doi: 10.1109/TFUZZ.2020.2981001.

25. V. N. Katsikis, S. D. Mourtas, P. S. Stanimirović, Y. N. Zhang, Continuous-time varying complex QR decomposition via zeroing neural dynamics, *Neural Process. Lett.*, **53** (2021), 3573–3590. doi: 10.1007/s11063-021-10566-y.

26. Z. J. Zhang, L. N. Zheng, J. Weng, Y. J. Mao, W. Lu, L. Xiao, A new varying-parameter recurrent neural-network for online solution of time-varying Sylvester equation, *IEEE T. Cybernetics*, **48** (2018), 3135–3148. doi: 10.1109/TCYB.2017.2760883.

27. Z. J. Zhang, L. N. Zheng, T. R. Qiu, F. Q. Deng, Varying-parameter convergent-differential neural solution to time-varying overdetermined system of linear equations, *IEEE T. Automat. Contr.*, **65** (2020), 874–881. doi: 10.1109/TAC.2019.2921681.

28. Z. J. Zhang, L. N. Zheng, A complex varying-parameter convergent-differential neural-network for solving online time-varying complex Sylvester equation, *IEEE T. Cybernetics*, **49** (2019), 3627–3639. doi: 10.1109/TCYB.2018.2841970.

29. Z. J. Zhang, L. D. Kong, L. N. Zheng, P. C. Zhang, X. L. Qu, B. L. Liao, et al., Robustness analysis of a power-type varying-parameter recurrent neural network for solving time-varying QM and QP problems and applications, *IEEE T. Syst. Man Cy.-S.*, **50** (2020), 5106–5118. doi: 10.1109/TSMC.2018.2866843.

30. H. M. Zhang, F. Ding, On the Kronecker products and their applications, *J. Appl. Math.*, **2013** (2013), 1–8. doi: 10.1155/2013/296185.

31. Y. N. Zhang, D. C. Jiang, J. Wang, A recurrent neural network for solving Sylvester equation with time-varying coefficients, *IEEE T. Neural Networ.*, **13** (2002), 1053–1063 doi: 10.1109/TNN.2002.1031938.

32. K. Chen, Recurrent implicit dynamics for online matrix inversion, *Appl. Math. Comput.*, **219** (2013), 10218–10224. doi: 10.1016/j.amc.2013.03.117.

33. Y. N. Zhang, K. Chen, H. Z. Tan, Performance analysis of gradient neural network exploited for online time-varying matrix inversion, *IEEE T. Automat. Contr.*, **54** (2009), 1940–1945. doi: 10.1109/TAC.2009.2023779.

34. F. Ding, G. J. Liu, X. P. Liu, Parameter estimation with scarce measurements, *Automatica*, **47** (2011), 1646–1655. doi: 10.1016/j.automatica.2011.05.007.

35. F. Ding, Y. J. Liu, B. Bao, Gradient based and least squares based iterative estimation algorithms for multi-input multi-output systems, *P. I. Mech. Eng. I-J. Sys.*, **226** (2012), 43–55. doi: 10.1177/0959651811409491.

36. F. Ding, G. J. Liu, X. P. Liu, Partially coupled stochastic gradient identification methods for non-uniformly sampled systems, *IEEE T. Automat. Contr.*, **55** (2010), 1976–1981. doi: 10.1109/TAC.2010.2050713.

37. L. Xiao, J. H. Dai, L. Jin, W. B. Li, S. Li, J. Hou, A noise-enduring and finite-time zeroing neural network for equality-constrained time-varying nonlinear optimization, *IEEE T. Syst. Man Cy.-S.*, **51** (2021), 4729–4740. doi: 10.1109/TSMC.2019.2944152.

38. L. Xiao, K. L. Li, M. X. Duan, Computing time-varying quadratic optimization with finite-time convergence and noise tolerance: A unified framework for zeroing neural network, *IEEE Trans. Neural Netw. Lear. Syst.*, **30** (2019), 3360–3369. doi: 10.1109/TNNLS.2019.2891252.

39. L. Xiao, Y. S. Zhang, J. H. Dai, K. Chen, S. Yang, W. B. Li, et al., A new noise-tolerant and predefined-time ZNN model for time-dependent matrix inversion, *Neural Networks*, **117** (2019), 124–134. doi: 10.1016/j.neunet.2019.05.005.

40. F. Yu, L. Liu, L. Xiao, K. L. Li, S. Cai, A robust and fixed-time zeroing neural dynamics for computing time-variant nonlinear equation using a novel nonliear activation function, *Neurocomputing*, **350** (2019), 108–116. doi: 10.1016/j.neucom.2019.03.053.

41. L. Xiao, Y. K. Cao, J. H. Dai, L. Jia, H. Y. Tan, Finite-time and predefined-time convergence design for zeroing neural network: Theorem, method, and verification, *IEEE T. Ind. Inform.*, **17** (2021), 4724–4732. doi: 10.1109/TII.2020.3021438.

42. L. Xiao, J. H. Dai, R. B. Lu, S. Li, J. C. Li, S. J. Wang, Design and comprehensive analysis of a noise-tolerant ZNN model with limited-time convergence for time-dependent nonlinear minimization, *IEEE T. Neur. Net. Lear.*, **31** (2020), 5339–5348. doi: 10.1109/TNNLS.2020.2966294.

43. L. Xiao, Y. S. Zhang, Q. Y. Zuo, J. H. Dai, J. C. Li, W. S. Tang, A noise-tolerant zeroing neural network for time-dependent complex matrix inversion under various kinds of noises, *IEEE T. Ind. Inform.*, **16** (2020), 3757–3766. doi: 10.1109/TII.2019.2936877.

44. M. Liu, L. M. Chen, X. H. Du, L. Jin, M. S. Shang, Activated gradients for deep neural networks, *IEEE T. Neur. Net. Lear.*, 2021, 1–13. doi: 10.1109/TNNLS.2021.3106044.

AIMS Press

© 2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)