



An Improved Recurrent Neural Network for Complex-Valued Systems of Linear Equation and Its Application to Robotic Motion Tracking

Lei Ding, Lin Xiao*, Bolin Liao, Rongbo Lu and Hua Peng

College of Information Science and Engineering, Jishou University, Jishou, China

OPEN ACCESS

Edited by:

Shuai Li,
Hong Kong Polytechnic
University, Hong Kong

Reviewed by:

Weibing Li,
University of Leeds, United Kingdom
Yinyan Zhang,
Hong Kong Polytechnic
University, Hong Kong
Dechao Chen,
Sun Yat-sen University, China
Ke Chen,
Tampere University of
Technology, Finland

*Correspondence:

Lin Xiao
xiaolin860728@163.com

Received: 30 May 2017

Accepted: 11 August 2017

Published: 01 September 2017

Citation:

Ding L, Xiao L, Liao B, Lu R and
Peng H (2017) An Improved
Recurrent Neural Network for
Complex-Valued Systems of Linear
Equation and Its Application to
Robotic Motion Tracking.
Front. Neurobot. 11:45.
doi: 10.3389/fnbot.2017.00045

To obtain the online solution of complex-valued systems of linear equation in complex domain with higher precision and higher convergence rate, a new neural network based on Zhang neural network (ZNN) is investigated in this paper. First, this new neural network for complex-valued systems of linear equation in complex domain is proposed and theoretically proved to be convergent within finite time. Then, the illustrative results show that the new neural network model has the higher precision and the higher convergence rate, as compared with the gradient neural network (GNN) model and the ZNN model. Finally, the application for controlling the robot using the proposed method for the complex-valued systems of linear equation is realized, and the simulation results verify the effectiveness and superiority of the new neural network for the complex-valued systems of linear equation.

Keywords: complex-valued systems of linear equation, recurrent neural network, finite-time convergence, robot, gradient neural network, motion tracking

1. INTRODUCTION

Today, the complex-valued systems of linear equation has been applied into many fields (Duran-Diaz et al., 2011; Guo et al., 2011; Subramanian et al., 2014; Hezari et al., 2016; Zhang et al., 2016; Xiao et al., 2017a). In mathematics, the complex-valued systems of linear equations can be written as

$$Az(t) = b \in \mathbb{C}^n, \quad (1)$$

where $A \in \mathbb{C}^{n \times n}$ and $b \in \mathbb{C}^n$ are the complex-valued coefficients, and $z(t) \in \mathbb{C}^n$ is a complex-valued vector to be computed. Xiao et al. (2015) proposed a fully complex-valued gradient neural network (GNN) to solve such a complex-valued systems of linear equation. However, the corresponding error norm usually converges to the theoretical solution after very long time. So to increase the convergence rate, a kind of neural network called Zhang neural network (ZNN) is proposed to make the lagging error converge to 0 exponentially (Zhang and Ge, 2005; Zhang et al., 2009). However, in Xiao (2016) and Xiao et al. (2017b), Xiao pointed that the original ZNN model cannot converge to 0 within finite time, and its real-time calculation capability may be limited (Marco et al., 2006; Li et al., 2013; Li and Li, 2014; Xiao, 2015). So, Xiao (2016) presented a new design formula, which can converge to 0 within finite time for the time-varying matrix inversion.

Considering that a complex variable can be written as the combination of its real and imaginary parts, we have $A = A_{re} + jA_{im}$, $b = b_{re} + jb_{im}$, and $z(t) = z_{re}(t) + z_{im}(t)$, where the symbol $j = \sqrt{-1}$ means an imaginary unit. Therefore, the equation (1) can be presented as

$$[A_{re} + jA_{im}][z_{re}(t) + jz_{im}(t)] = b_{re} + jb_{im} \in \mathbb{C}^n, \quad (2)$$

where $A_{re} \in \mathbb{R}^{n \times n}$, $A_{im} \in \mathbb{R}^{n \times n}$, $z_{re} \in \mathbb{R}^n$, $z_{im} \in \mathbb{R}^n$, $b_{re} \in \mathbb{R}^n$, and $b_{im} \in \mathbb{R}^n$. According to the complex formula, the real (or imaginary) part of the left-side and right-side of equation is equal (Zhang et al., 2016). Then we have

$$\begin{cases} A_{re}z_{re}(t) - A_{im}z_{im}(t) = b_{re} \in \mathbb{R}^n, \\ A_{im}z_{re}(t) + A_{re}z_{im}(t) = b_{im} \in \mathbb{R}^n. \end{cases} \quad (3)$$

Thus, we can express the equation (3) in a compact matrix form as:

$$\begin{bmatrix} A_{re} & -A_{im} \\ A_{im} & A_{re} \end{bmatrix} \begin{bmatrix} z_{re}(t) \\ z_{im}(t) \end{bmatrix} = \begin{bmatrix} b_{re} \\ b_{im} \end{bmatrix} \in \mathbb{R}^{2n}. \quad (4)$$

We can write the equation (4) as

$$Cx(t) = e \in \mathbb{R}^{2n}, \quad (5)$$

where $C = \begin{bmatrix} A_{re} & -A_{im} \\ A_{im} & A_{re} \end{bmatrix}$, $x(t) = \begin{bmatrix} z_{re}(t) \\ z_{im}(t) \end{bmatrix}$, and $e = \begin{bmatrix} b_{re} \\ b_{im} \end{bmatrix}$. Now the complex-valued system of linear equation can be computed in real domain. In this situation, most methods for solving real-valued system of linear equation can be used to solve the complex-valued system of linear equation (Zhang and Ge, 2005; Zhang et al., 2009; Guo et al., 2011). For example, a gradient neural network (GNN) can be designed to solve such a real-valued system of linear equation. The GNN model can be directly presented as follows (Xiao et al., 2015):

$$\dot{x}(t) = -\gamma C^T(Cx(t) - e), \quad (6)$$

where design parameter $\gamma > 0$ is employed to adjust the convergence rate of the GNN model. Zhang et al. (Zhang et al., 2016) used the recurrent neural network to solve the complex-valued quadratic programming problems. Hezari et al. (2016) solved a class of complex symmetric system of linear equations using an iterative method. However, the above mentioned neural networks cannot converge to the desired solution within finite time. Considering that the complex-valued system of linear equation can be transformed into the real-valued system of linear equation, a new neural network can be derived from the new design formula proposed by Xiao for solving the complex-valued system of linear equation (Xiao et al., 2015). In addition, the new neural network possesses a finite-time convergence property.

In recent years, the research on robot has become a hot spot (Khan et al., 2016a,b; Zanchettin et al., 2016; Guo et al., 2017), and the neural network has been successfully applied into the robot domain (He et al., 2016; Jin and Li, 2016; Woodford et al., 2016; Jin et al., 2017; Xiao, 2017). However, the application of the new design method for the complex-valued system of linear

equation in robot domain has not been reported. So this is the first time to propose a new neural network, which can convergence within finite-time for solving the complex-valued system of linear equation and its application to robot domain.

The rest of this paper is organized into four sections. Section 2 proposes a finite-time recurrent neural network (FTRNN) to deal with the complex-valued system of linear equation, and its convergence analysis is given in detail. Section 3 gives the computer-simulation results to substantiate the theoretical analysis and the superiority. Section 4 gives the results of the application for controlling the robotic motion planning. Finally, the conclusions are presented in Section 5. Before ending this section, the main contributions of the current work are presented as follows.

- The research object focuses on a complex-valued system of linear equation in complex domain, which is quite different from the previously investigated real-valued system of linear equation in real domain.
- A new finite-time recurrent neural network is proposed and investigated for solving complex-valued systems of linear equation in complex domain. In addition, it is theoretically proved to be convergent within finite time.
- Theoretical analyses and simulative results are presented to show the effectiveness of the proposed finite-time recurrent neural network. In addition, a five-link planar manipulator is used to validate the applicability of the finite-time recurrent neural network.

2. FINITE-TIME RECURRENT NEURAL NETWORK

Considering that the complex-valued system of linear equation can be computed in real domain, the error function $E(t)$ of traditional ZNN can be presented as

$$E(t) = Cx(t) - e \in \mathbb{R}^{2n}. \quad (7)$$

Then, according to the design formula $\dot{E}(t) = -\gamma\Phi(E(t))$, the original ZNN model can be presented as

$$C\dot{x}(t) = -\gamma\Phi(Cx(t) - e), \quad (8)$$

where $\Phi(\cdot)$ means an activation function array, and $\gamma > 0$ is used to adjust the convergence rate. In this paper, the new design formula in Xiao (2016) for $E(t)$ can be directly employed and written as follows:

$$\frac{dE(t)}{dt} = -\gamma\Phi(\rho_1 E(t) + \rho_2 E^{j/f}(t)), \quad (9)$$

where the parameters ρ_1 and ρ_2 satisfy $\rho_1 > 0$, $\rho_2 > 0$, and f and j mean the positive odd integer and satisfy $f > j$. Then we have

$$C\dot{x}(t) = -\gamma\Phi(\rho_1(Cx(t) - e) + \rho_2(Cx(t) - e)^{j/f}(t)). \quad (10)$$

To simplify the formula, $\Phi(\cdot)$ uses the linear activation function. Then we have

$$\frac{dE(t)}{dt} = -\gamma(\rho_1 E(t) + \rho_2 E^{j/f}(t)), \quad (11)$$

and

$$C\dot{x}(t) = -\gamma\left(\rho_1(Cx(t) - e) + \rho_2(Cx(t) - e)^{j/f}\right), \quad (12)$$

which is called the finite-time recurrent neural network (FTRNN) model to online deal with the complex-valued system of linear equation. In addition, for design formula (11) and FTRNN model (12), we have the following two theorems to ensure their finite-time convergence properties.

Theorem 1. *The error function $E(t)$ of design formula (11) converges to zero within finite-time t_u regardless of its randomly generated initial error $E(0)$:*

$$t_u = \frac{f}{\gamma\rho_1(f-j)} \ln \frac{\rho_1 h_M(0)^{(f-j)/f} + \rho_2}{\rho_2},$$

where $h_M(0)$ means the maximum element of the matrix $E(0)$.

PROOF. For design formula (11), we have

$$\frac{dE(t)}{dt} = -\left(\gamma\rho_1 E(t) + \gamma\rho_2 E^{j/f}(t)\right). \quad (13)$$

To deal with the dynamic response of the equation (13), the above differential equation can be rewritten as below:

$$E^{-j/f}(t) \diamond \frac{dE(t)}{dt} + \gamma\rho_1 E^{(f-j)/f}(t) = -\gamma\rho_2, \quad (14)$$

where the matrix-multiplication operator \diamond means the Hadamard product and can be written as

$$W \diamond S = \begin{bmatrix} W_{11}S_{11}, & W_{12}S_{12}, & \cdots, & W_{1n}S_{1n} \\ R_{21}S_{21}, & W_{21}S_{21}, & \cdots, & W_{2n}S_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ W_{m1}S_{m1}, & W_{m2}S_{m2}, & \cdots, & W_{mn}S_{mn}, \end{bmatrix} \in \mathbb{R}^{m \times n}.$$

Now let us define $Y(t) = E^{(f-j)/f}(t)$. Then, we have

$$\frac{dY(t)}{dt} = \frac{f-j}{f} E^{-j/f}(t) \diamond \frac{dE(t)}{dt}.$$

Thus, the differential equation (14) can be equivalent to the following first order differential equation:

$$\frac{dY(t)}{dt} + \frac{f-j}{f} \gamma\rho_1 Y(t) = -\frac{f-j}{f} \gamma\rho_2 I. \quad (15)$$

This is a typical first order differential equation, and we have

$$Y(t) = \left(\frac{\rho_2}{\rho_1} I + Z(0)\right) \exp\left(-\frac{f-j}{f} \gamma\rho_1 t\right) - \frac{\rho_2}{\rho_1} I. \quad (16)$$

So we have

$$E^{(f-j)/f}(t) = \left(\frac{\rho_2}{\rho_1} I + E^{(f-j)/f}(0)\right) \exp\left(-\frac{f-j}{f} \gamma\rho_1 t\right) - \frac{\rho_2}{\rho_1} I, \quad (17)$$

and

$$E(t) = \left[\left(\frac{\rho_2}{\rho_1} I + E^{(f-j)/f}(0)\right) \exp\left(-\frac{f-j}{f} \gamma\rho_1 t\right) - \frac{\rho_2}{\rho_1} I\right]^{f/(f-j)}. \quad (18)$$

From the equation (18), we can find the error matrix $E(t)$ will converge to 0 in t_u , and

$$\left(\frac{\rho_2}{\rho_1} I + E^{(f-j)/f}(0)\right) \exp\left(-\frac{f-j}{f} \gamma\rho_1 t_u\right) - \frac{\rho_2}{\rho_1} I = 0. \quad (19)$$

Considering each element of the matrix $E(t)$ has the same identical dynamics, we have

$$t_{ik} = \frac{f}{\gamma\rho_1(f-j)} \ln \frac{\rho_1 h_{ik}^{(f-j)/f}(0) + \rho_2}{\rho_1}, \quad (20)$$

where h_{ik} means the ik th element of the matrix $E(0)$, and t_{ik} means the ik th finite-time convergence upper bound of the matrix $E(t)$. Let $h_M(0) = \max(h_{ik})$. Then for any ik th element of the matrix $E(t)$, we have the maximum convergence time:

$$t_u = \frac{f}{\gamma\rho_1(f-j)} \ln \frac{\rho_1 h_M(0)^{(f-j)/f} + \rho_2}{\rho_2}.$$

According to the above analysis, we can draw a conclusion that the error matrix $E(t)$ will converge to 0 within the finite time t_u regardless of its initial value $E(0)$. Now the proof is completed. \square

Theorem 2. *The state matrix $X(t)$ of FTRNN model (12) will converge to the theoretical solution of (5) in finite time t_u regardless of its randomly generated initial state $x(0)$, and*

$$t_u \in \left\{ \frac{f}{\gamma\rho_1(f-j)} \ln \frac{\rho_1 h_L(0)^{(f-j)/f} + \rho_2}{\rho_2}, \frac{f}{\gamma\rho_1(f-j)} \ln \frac{\rho_1 h_M(0)^{(f-j)/f} + \rho_2}{\rho_2} \right\},$$

where $h_M(0)$ and $h_L(0)$ mean the largest and the smallest elements of the matrix $E(0)$, respectively.

PROOF. Let $x_{(FT)}(t)$ represent the solution of the FTRNN model (12), $x_{(org)}(t)$ represent the theoretical solution of the equation (5), and $\tilde{x}(t)$ represent the difference between $x_{(FT)}(t)$ and $x_{(org)}(t)$. Then, we can obtain

$$\tilde{x}(t) = x_{(FT)}(t) - x_{(org)}(t) \in \mathbb{R}^{2n \times 2n}. \quad (21)$$

The equation (21) can be written as

$$x_{(FT)}(t) = \tilde{x}(t) + x_{(org)}(t) \in \mathbb{R}^{2n \times 2n}. \quad (22)$$

Substitutes the above equation into FTRNN model (12), we have

$$C(\dot{\tilde{x}}(t) + \dot{x}_{(org)}(t)) = -\gamma\left(\rho_1(C(\tilde{x}(t) + x_{(org)}(t)) - e) + \rho_2(C(\tilde{x}(t) + x_{(org)}(t)) - e)^{j/f}\right). \quad (23)$$

Considering $Cx_{(org)}(t) - e = 0$ and $C\dot{x}_{(org)}(t) = 0$, the above equation can be written as

$$C\dot{\tilde{x}}(t) = -\gamma(\rho_1(C\tilde{x}(t) - e) + \rho_2(C\tilde{x}(t) - e)^{j/f}).$$

Furthermore, considering $E(t) = C(\tilde{x}(t) + x_{(org)}(t)) - e$, $Cx_{(org)}(t) - e = 0$, and $E(t) = C\tilde{x}(t)$, the above differential equation can be written as

$$\frac{dE(t)}{dt} = -\gamma(\rho_1(E(t) - e) + \rho_2(E(t) - e)^{j/f}).$$

Let $\tilde{E}(t) = E(t) - e$, then we have

$$\frac{d\tilde{E}(t)}{dt} = -\gamma(\rho_1\tilde{E}(t) + \rho_2\tilde{E}^{j/f}(t)). \quad (24)$$

So according to the equation (20), we have

$$\tilde{t}_{ik} = \frac{f}{\gamma\rho_1(f-j)} \ln \frac{\rho_1\tilde{h}_{ik}^{(f-j)/f}(0) + \rho_2}{\rho_1}, \quad (25)$$

where \tilde{t}_{ik} means the time upper of ik th solution of the matrix $\tilde{E}(t)$, and \tilde{h}_{ik} means the ik th initial error value of the matrix $\tilde{E}(0)$.

Let us define $\tilde{h}_M = \max(\tilde{h}_{ik}(0))$, and $\tilde{h}_L = \min(\tilde{h}_{ik}(0))$ with $i, k = 1, 2, \dots, n$. Then for all possible i and k , we have

$$\begin{aligned} & \frac{f}{\gamma\rho_1(f-j)} \ln \frac{\rho_1\tilde{h}_L^{(f-j)/f}(0) + \rho_2}{\rho_1} \\ & \leq \tilde{t}_{ik}(t) \leq \frac{f}{\gamma\rho_1(f-j)} \ln \frac{\rho_1\tilde{h}_M^{(f-j)/f}(0) + \rho_2}{\rho_1}. \end{aligned}$$

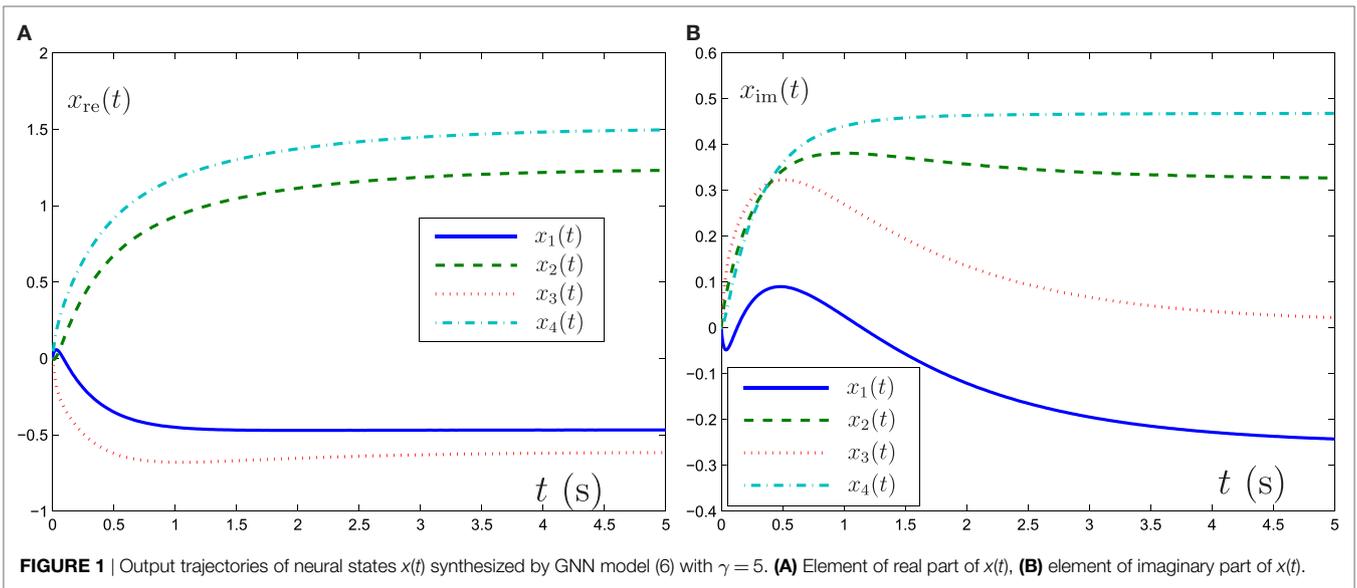


FIGURE 1 | Output trajectories of neural states $x(t)$ synthesized by GNN model (6) with $\gamma = 5$. (A) Element of real part of $x(t)$, (B) element of imaginary part of $x(t)$.

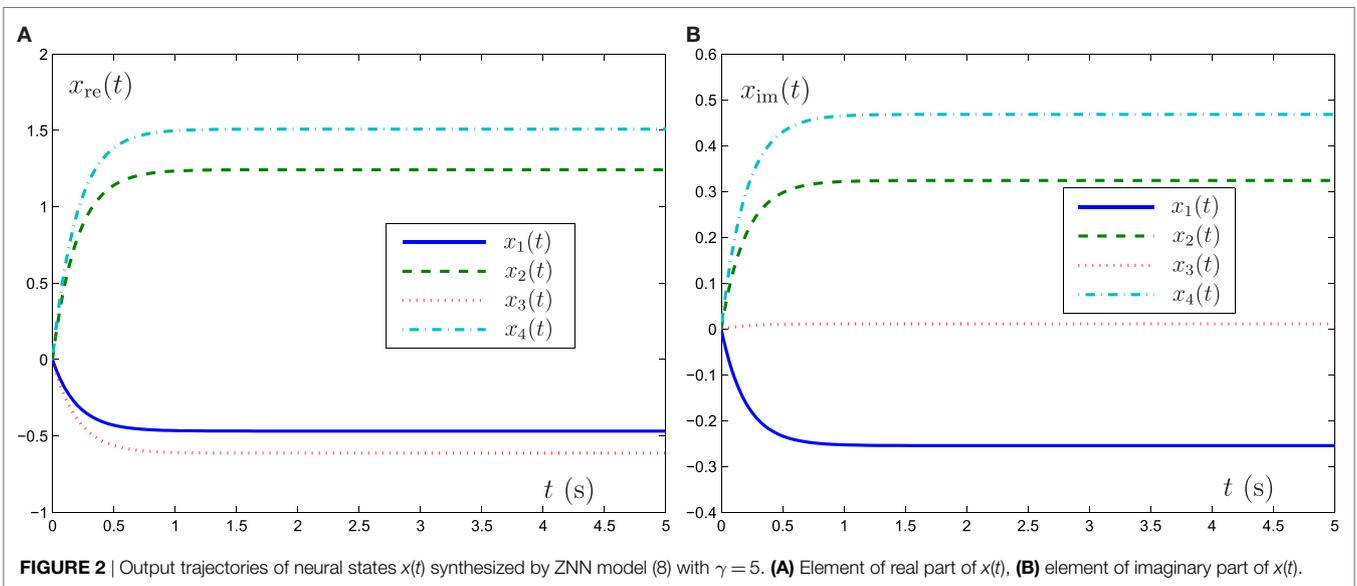


FIGURE 2 | Output trajectories of neural states $x(t)$ synthesized by ZNN model (8) with $\gamma = 5$. (A) Element of real part of $x(t)$, (B) element of imaginary part of $x(t)$.

The above equation shows that the state matrix $\tilde{x}(t) = x_{(FT)}(t) - x_{(org)}(t)$ will converges to 0 within finite time regardless of its initial error value. In another word, the matrix $x_{(FT)}(t)$ for the FTRNN model (12) will converge to the theoretical solution $x_{(org)}(t)$ for the theoretical model (5) within finite time regardless of its randomly generated initial state $x(0)$. Now the proof is completed. \square

of FTRNN model (12), we choose the same complex-valued matrix A and b as these of the paper (Xiao et al., 2015). Then we have

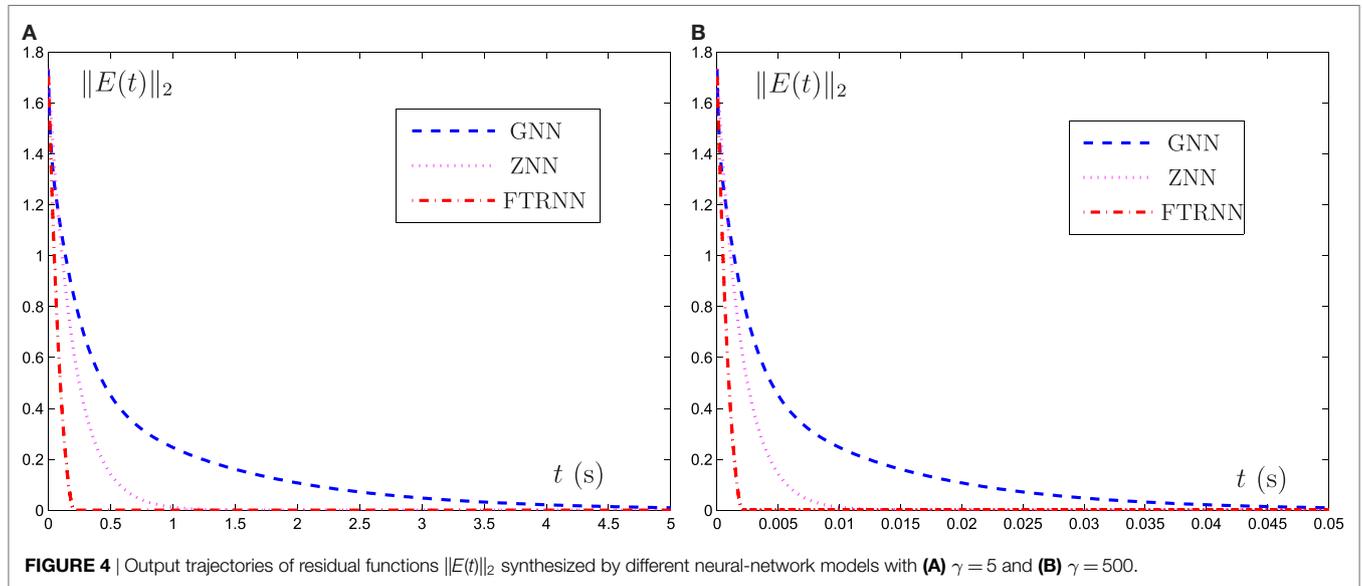
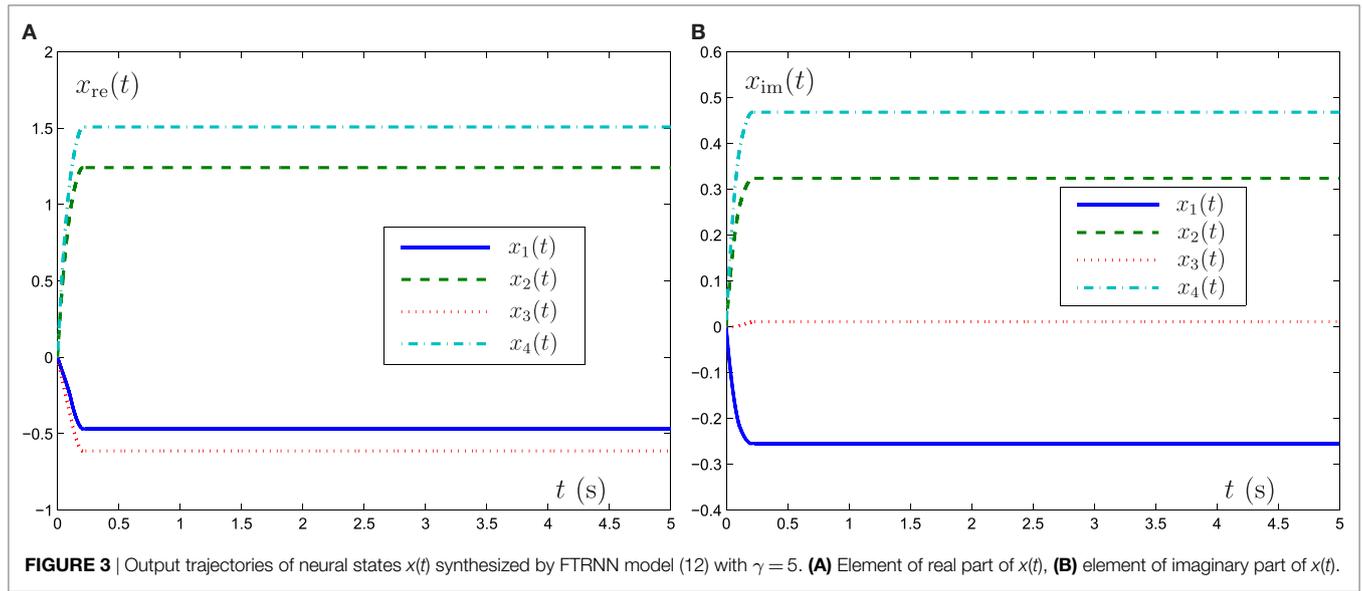
$$A = \begin{bmatrix} -0.7597 + 0.6503j & -0.8391 - 0.5440j & 0.2837 - 0.9589j & 1 \\ 0.7597 + 0.6503j & -0.8391 + 0.5440j & -0.2837 - 0.9589j & 1 \\ 0.7597 - 0.6503j & -0.8391 - 0.5440j & -0.2837 + 0.9589j & 1 \\ 0 - 1.0000j & -1.0000 & 0 + 1.0000j & 1 \end{bmatrix}$$

So we have

$$A_{re} = \begin{bmatrix} -0.7597 & -0.8391 & 0.2837 & 1 \\ 0.7597 & -0.8391 & -0.2837 & 1 \\ 0.7597 & -0.8391 & -0.2837 & 1 \\ 0 & -1.0000 & 0 & 1 \end{bmatrix}$$

3. COMPUTER SIMULATION

In this section, a digital example will be carried out to show the superiority of FTRNN model (12) to GNN model (6) and ZNN model (8). We can choose the design parameters f and j , which satisfy $f > j$. For example, we choose $f = 5$ and $j = 1$ in this paper. In addition to this, to substantiate the superiority



and

$$A_{im} = \begin{bmatrix} 0.6503 & -0.5440 & -0.9589 & 0 \\ 0.6503 & 0.5440 & -0.9589 & 0 \\ -0.6503 & -0.5440 & 0.9589 & 0 \\ -1.0000 & 0 & 1.0000 & 0 \end{bmatrix}$$

Now the randomly generated vector $b = [1.0000, 0.2837 + 0.9589j, 0.2837 - 0.9589j, 0]^T$ in Xiao et al. (2015) is employed in this paper. The theoretical solution of the complex-valued linear equation system can be written as $z_{(org)} = [-0.4683 - 0.2545j, 1.2425 + 0.3239j, -0.6126 + 0.0112j, 1.5082 + 0.4683j]$. Then according to the equation (5), we have

$$C = \begin{bmatrix} -0.7597 & -0.8391 & 0.2837 & 1 & -0.6503 & 0.5440 & 0.9589 & 0 \\ 0.7597 & -0.8391 & -0.2837 & 1 & -0.6503 & -0.5440 & 0.9589 & 0 \\ 0.7597 & -0.8391 & -0.2837 & 1 & 0.6503 & 0.5440 & -0.9589 & 0 \\ 0 & -1.0000 & 0 & 1 & 1.0000 & 0 & -1.0000 & 0 \\ 0.6503 & -0.5440 & -0.9589 & 0 & -0.7597 & -0.8391 & 0.2837 & 1 \\ 0.6503 & 0.5440 & -0.9589 & 0 & 0.7597 & -0.8391 & -0.2837 & 1 \\ -0.6503 & -0.5440 & 0.9589 & 0 & 0.7597 & -0.8391 & -0.2837 & 1 \\ -1.0000 & 0 & 1.0000 & 0 & 0 & -1.0000 & 0 & 1 \end{bmatrix}$$

and $e = [1.0000, 0.2837, 0.2837, 0, 0, 0.9589, -0.9589, 0]^T$. So the theoretical solution of the complex-valued linear equation system

can be rewritten as $x_{(org)} = [-0.4683, 1.2425, -0.6126, 1.5082, -0.2545, 0.3239, 0.0112, 0.4683]^T$.

First, a zero initial complex-valued state $z(0) \in \mathbb{C}^4$ is generated, which can be transformed into the real-valued state $x(0) \in \mathbb{R}^8$ in real domain. To help facilitate the contrast, we choose the design parameter $\gamma = 5$ and $\gamma = 500$, respectively.

Now GNN model (6), ZNN model (8), and FTRNN model (12) are applied to solve this complex-valued systems of linear equation problem. The output trajectories of the corresponding neural-state solutions are displayed in **Figures 1–3**. As seen from such three figures, we can conclude that the output trajectories of the neural-state solutions can converge to the theoretical solutions, but the convergence rates are different. By comparison, we can easily find that FTRNN model (12) has a fastest convergence property.

To directly show the solution process of such three neural-network models, the evolution of the corresponding residual errors, measured by the norm $\|E(t)\|_2$, is plotted in **Figure 4** under the conditions of $\gamma = 5$ and $\gamma = 500$. From **Figure 4A**, the results are consistent with those of **Figures 1–3**. In addition, from **Figure 4B**, the convergence speeds of GNN model (6), ZNN model (8), and FTRNN model (12) can be improved as the value of γ increases.

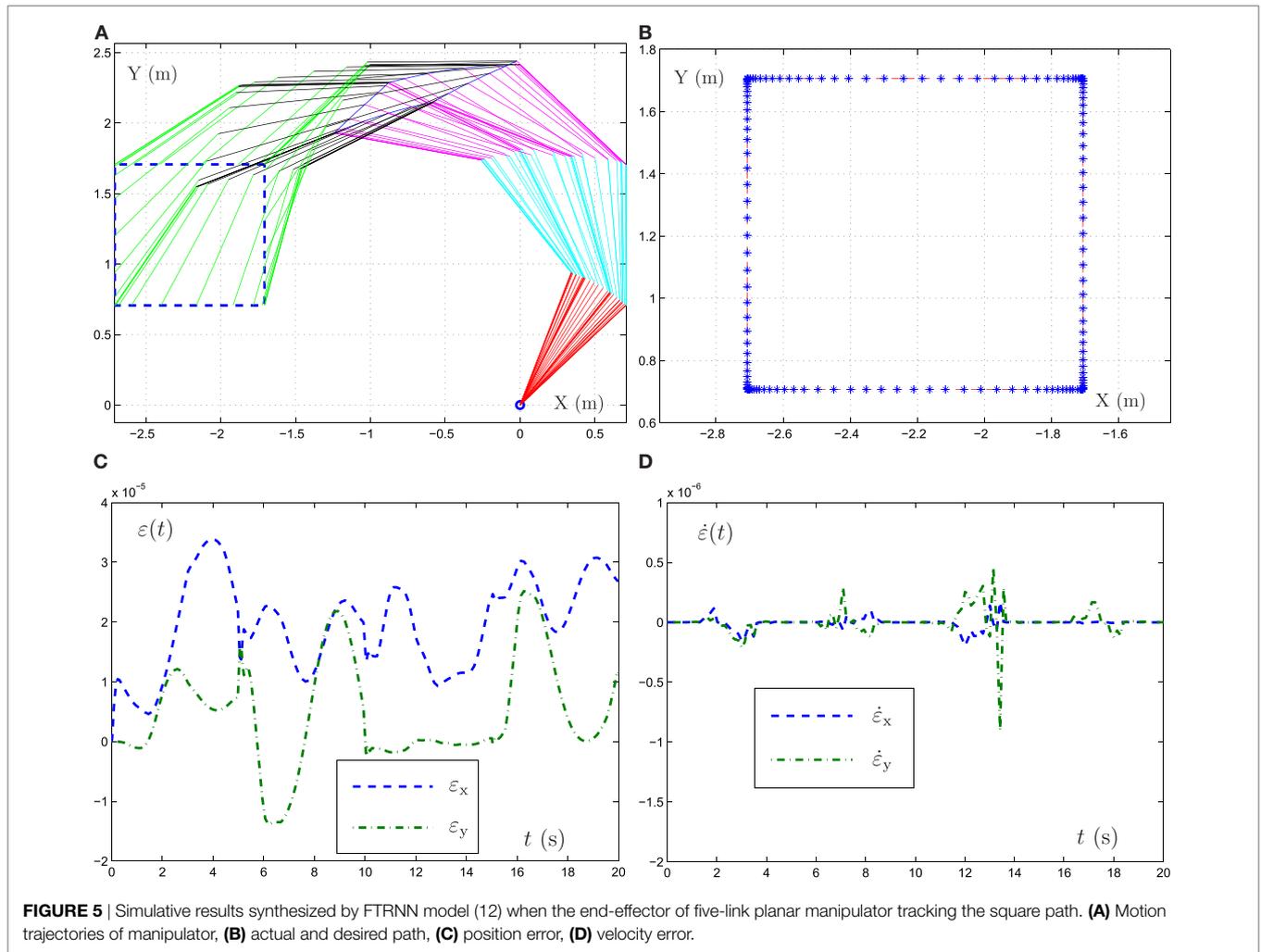
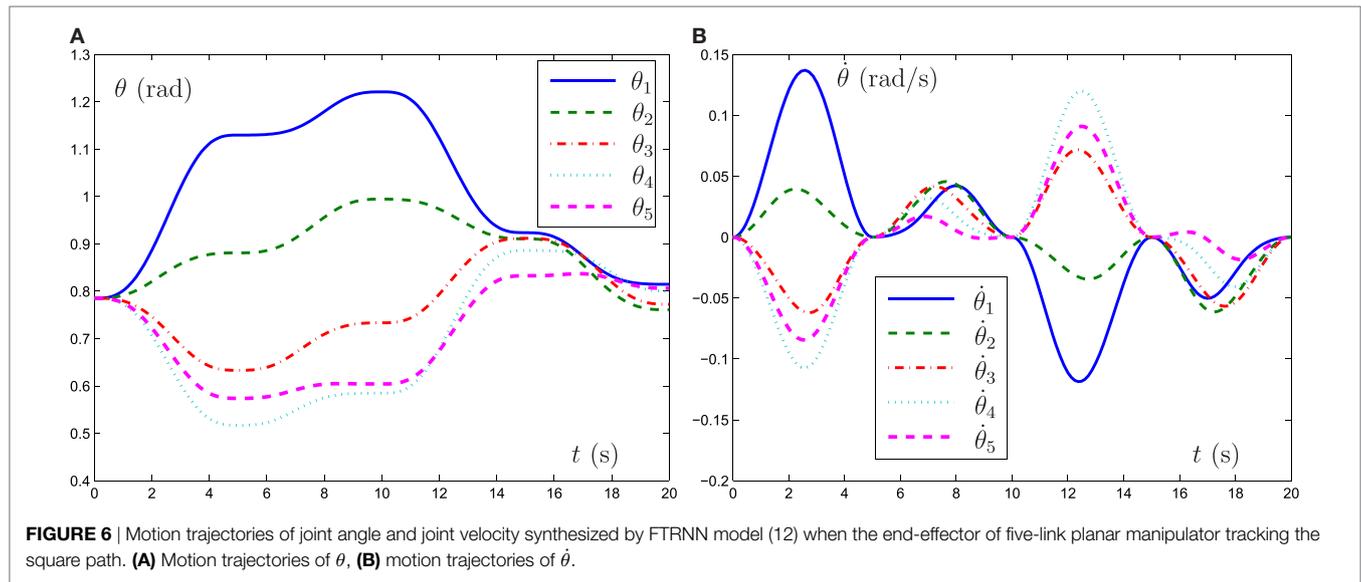


FIGURE 5 | Simulative results synthesized by FTRNN model (12) when the end-effector of five-link planar manipulator tracking the square path. **(A)** Motion trajectories of manipulator, **(B)** actual and desired path, **(C)** position error, **(D)** velocity error.



Now we can draw a conclusion that, as compared with GNN model (6) and ZNN model (8), FTRNN model (12) has the most superiority for solving the complex-valued system of linear equation problem.

4. APPLICATION TO ROBOTIC MOTION TRACKING

In this section, a five-link planar manipulator is used to validate the applicability of the finite-time recurrent neural network (FTRNN) (Zhang et al., 2011). It is well known that the kinematics equations of the five-link planar manipulator at the position level and at the velocity level are, respectively, written as follows (Xiao and Zhang, 2013, 2014a,b, 2016; Xiao et al., 2017c):

$$r(t) = f(\theta(t)) \quad (26)$$

$$\dot{r}(t) = J(\theta)\dot{\theta}(t) \quad (27)$$

where θ denotes the angle vector of the five-link planar manipulator, $r(t)$ denotes the end-effector position vector, $f(\cdot)$ stands for a smooth non-linear mapping function, and $J(\theta) = \partial f(\theta)/\partial \theta \in \mathbb{R}^{m \times n}$.

To realize the motion tracking of this five-link planar manipulator, the inverse kinematic equation has been solved. Especially, equation (27) can be seen as a system of linear equations when the end-effector motion tracking task is allocated [i.e., $\dot{r}(t)$ is known and $\dot{\theta}(t)$ needs to be solved]. Thus, we can use the proposed FTRNN model (12) to solve this system of linear equations. Then, based on the design process of FTRNN model (12), we can obtain the following dynamic model to track control of the five-link planar manipulator [based on the formulation of equation (27)]:

$$C\dot{x}(t) = -\gamma \left(\rho_1(Cx(t) - e) + \rho_2(Cx(t) - e)^{j/f} \right),$$

where $C = J$, $x = \dot{\theta}$ and $e = \dot{r}(t)$.

In the simulation experiment, a square path (with the radius being 1 m) is allocated for the five-link planar manipulator to

track. Besides, initial state of the mobile manipulator is set as $\theta(0) = [\pi/4, \pi/4, \pi/4, \pi/4, \pi/4]^T$, $\gamma = 10^3$ and task duration is 20 s. The experiment results are shown in **Figures 5 and 6**. From the results shown in such two figures, we can obtain that the five-link planar manipulator completes the square path tracking task successfully.

5. CONCLUSION

In this paper, a finite-time recurrent neural network (FTRNN) for the complex-valued system of linear equation in complex domain is proposed and investigated. This is the first time to propose such a neural network model, which can convergence within finite time to online deal with the complex-valued system of linear equation in complex domain, and the first time to apply this FTRNN model for robotic path tracking by solving the system of linear equation. The simulation experiments show that the proposed FTRNN model has better effectiveness, as compared to the GNN model and the ZNN model for the complex-valued system of linear equation in complex domain.

AUTHOR CONTRIBUTIONS

LD: experiment preparation, publication writing; LX: experiment preparation, data processing, publication writing; BL: technology support, data acquisition, publication review; RL: supervision of data processing, publication review; HP revised the manuscript.

FUNDING

This work is supported by the National Natural Science Foundation of China under grants 61503152 and 61363073, the Natural Science Foundation of Hunan Province, China under grants 2016JJ2101 and 2017JJ3258, the National Natural Science Foundation of China under grants, 61563017, 61662025, and 61561022, and the Research Foundation of Jishou University, China under grants 2017JSUJD031, 2015SYJG034, JGY201643, and JG201615.

REFERENCES

- Duran-Diaz, I., Cruces, S., Sarmiento-Vega, M. A., and Aguilera-Bonet, P. (2011). Cyclic maximization of non-gaussianity for blind signal extraction of complex-valued sources. *Neurocomputing* 74, 2867–2873. doi:10.1016/j.neucom.2011.03.031
- Guo, D., Nie, Z., and Yan, L. (2017). The application of noise-tolerant ZD design formula to robots' kinematic control via time-varying nonlinear equations solving. *IEEE Trans. Syst. Man Cybern. Syst.* doi:10.1109/TSMC.2017.2705160
- Guo, D., Yi, C., and Zhang, Y. (2011). Zhang neural network versus gradient-based neural network for time-varying linear matrix equation solving. *Neurocomputing* 74, 3708–3712. doi:10.1016/j.neucom.2011.05.021
- He, W., Chen, Y., and Yin, Z. (2016). Adaptive neural network control of an uncertain robot with full-state constraints. *IEEE Trans. Cybern.* 46, 620–629. doi:10.1109/TCYB.2015.2411285
- Hezari, D., Salkuyeh, D. K., and Edalatpour, V. (2016). A new iterative method for solving a class of complex symmetric system of linear equations. *Numer. Algorithms* 73, 1–29. doi:10.1007/s11075-016-0123-x
- Jin, L., and Li, S. (2016). Distributed task allocation of multiple robots: a control perspective. *IEEE Trans. Syst. Man Cybern. Syst.* doi:10.1109/TSMC.2016.2627579
- Jin, L., Li, S., Xiao, L., Lu, R., and Liao, B. (2017). Cooperative motion generation in a distributed network of redundant robot manipulators with noises. *IEEE Trans. Syst. Man Cybern. Syst.* doi:10.1109/TSMC.2017.2693400
- Khan, M., Li, S., Wang, Q., and Shao, Z. (2016a). Formation control and tracking for co-operative robots with non-holonomic constraints. *J. Intell. Robot. Syst.* 82, 163–174. doi:10.1007/s10846-015-0287-y
- Khan, M., Li, S., Wang, Q., and Shao, Z. (2016b). CPS oriented control design for networked surveillance robots with multiple physical constraints. *IEEE Trans. Comput. Aided Des. Integr. Circuit Syst.* 35, 778–791. doi:10.1109/TCAD.2016.2524653
- Li, S., Chen, S., and Liu, B. (2013). 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, 189–205. doi:10.1007/s11063-012-9241-1
- Li, S., and Li, Y. (2014). Nonlinearly activated neural network for solving time-varying complex Sylvester equation. *IEEE Trans. Cybern.* 44, 1397–1407. doi:10.1109/TCYB.2013.2285166
- Marco, M., Forti, M., and Grazzini, M. (2006). Robustness of convergence in finite time for linear programming neural networks. *Int. J. Circuit Theory Appl.* 34, 307–316. doi:10.1002/cta.352
- Subramanian, K., Savitha, R., and Suresh, S. (2014). A complex-valued neuro-fuzzy inference system and its learning mechanism. *Neurocomputing* 123, 110–120. doi:10.1016/j.neucom.2013.06.009
- Woodford, G. W., Pretorius, C. J., and Plessis, M. C. D. (2016). Concurrent controller and simulator neural network development for a differentially-steered robot in evolutionary robotics. *Rob. Auton. Syst.* 76, 80–92. doi:10.1016/j.robot.2015.10.011
- Xiao, L. (2015). A finite-time convergent neural dynamics for online solution of time-varying linear complex matrix equation. *Neurocomputing* 167, 254–259. doi:10.1016/j.neucom.2015.04.070
- Xiao, L. (2016). A new design formula exploited for accelerating Zhang neural network and its application to time-varying matrix inversion. *Theor. Comput. Sci.* 647, 50–58. doi:10.1016/j.tcs.2016.07.024
- Xiao, L. (2017). Accelerating a recurrent neural network to finite-time convergence using a new design formula and its application to time-varying matrix square root. *J. Franklin Inst.* 354, 5667–5677. doi:10.1016/j.jfranklin.2017.06.012
- Xiao, L., Liao, B., Zeng, Q., Ding, L., and Lu, R. (2017a). “A complex gradient neural dynamics for fast complex matrix inversion,” in *International Symposium on Neural Networks* (Springer), 521–528.
- Xiao, L., Liao, B., Jin, J., Lu, R., Yang, X., and Ding, L. (2017b). A finite-time convergent dynamic system for solving online simultaneous linear equations. *Int. J. Comput. Math.* 94, 1778–1786. doi:10.1080/00207160.2016.1247436
- Xiao, L., Liao, B., Li, S., Zhang, Z., Ding, L., and Jin, L. (2017c). Design and analysis of FTZNN applied to real-time solution of nonstationary Lyapunov equation and tracking control of wheeled mobile manipulator. *IEEE Trans. Ind. Inf.* doi:10.1109/TII.2017.2717020
- Xiao, L., Meng, W. W., Lu, R. B., Yang, X., Liao, B., and Ding, L. (2015). “A fully complex-valued neural network for rapid solution of complex-valued systems of linear equations,” in *International Symposium on Neural Networks 2015, Lecture Notes in Computer Science*, Vol. 9377, 444–451.
- Xiao, L., and Zhang, Y. (2013). Acceleration-level repetitive motion planning and its experimental verification on a six-link planar robot manipulator. *IEEE Trans. Control Syst. Technol.* 21, 906–914. doi:10.1109/TCST.2012.2190142
- Xiao, L., and Zhang, Y. (2014a). Solving time-varying inverse kinematics problem of wheeled mobile manipulators using Zhang neural network with exponential convergence. *Nonlinear Dyn.* 76, 1543–1559. doi:10.1007/s11071-013-1227-7
- Xiao, L., and Zhang, Y. (2014b). A new performance index for the repetitive motion of mobile manipulators. *IEEE Trans. Cybern.* 44, 280–292. doi:10.1109/TCYB.2013.2253461
- Xiao, L., and Zhang, Y. (2016). Dynamic design, numerical solution and effective verification of acceleration-level obstacle-avoidance scheme for robot manipulators. *Int. J. Syst. Sci.* 47, 932–945. doi:10.1080/00207721.2014.909971
- Zanchettin, A. M., Ceriani, N. M., Rocco, P., Ding, H., and Matthias, B. (2016). Safety in human-robot collaborative manufacturing environments: metrics and control. *IEEE Trans. Autom. Sci. Eng.* 13, 882–893. doi:10.1109/TASE.2015.2412256
- Zhang, Y., and Ge, S. S. (2005). Design and analysis of a general recurrent neural network model for time-varying matrix inversion. *IEEE Trans. Neural Netw.* 16, 1447–1490. doi:10.1109/TNN.2005.857946
- Zhang, Y., Shi, Y., Chen, K., and Wang, C. (2009). Global exponential convergence and stability of gradient-based neural network for online matrix inversion. *Appl. Math. Comput.* 215, 1301–1306. doi:10.1016/j.amc.2009.06.048
- Zhang, Y., Xiao, L., Xiao, Z., and Mao, M. (2016). *Zeroing Dynamics, Gradient Dynamics, and Newton Iterations*. Boca Raton: CRC Press.
- Zhang, Y., Yang, Y., Tan, N., and Cai, B. (2011). Zhang neural network solving for time-varying full-rank matrix Moore-Penrose inverse. *Computing* 92, 97–121. doi:10.1007/s00607-010-0133-9

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The reviewer, YZ, and handling editor declared their shared affiliation.

Copyright © 2017 Ding, Xiao, Liao, Lu and Peng. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.