Filomat 34:15 (2020), 5009–5018
https://doi.org/10.2298/FIL2015009D

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

# Two Novel Finite Time Convergent Recurrent Neural Networks for Tackling Complex-Valued Systems of Linear Equation

**Lei Ding[a], Lin Xiao[b,a], Kaiqing Zhou[a], Yonghong Lan[c], Yongsheng Zhang[a]**

[a]*College of Information Science and Engineering, Jishou University, Jishou 416000, China*
[b]*College of Information Science and Engineering, Hunan University, Changsha 410082, China*
[c]*College of Information Engineering, Xiangtan University, Xiangtan 411105, China*

**Abstract.** Compared to the linear activation function, a suitable nonlinear activation function can accelerate the convergence speed. Based on this finding, we propose two modified Zhang neural network (ZNN) models using different nonlinear activation functions to tackle the complex-valued systems of linear equation (CVSLE) problems in this paper. To fulfill this goal, we first propose a novel neural network called NRNN-SBP model by introducing the sign-bi-power activation function. Then, we propose another novel neural network called NRNN-IRN model by introducing the tunable activation function. Finally, simulative results demonstrate that the convergence speed of NRNN-SBP and the NRNN-IRN is faster than that of the FTRNN model. On the other hand, these results also reveal that different nonlinear activation function will have a different effect on the convergence rate for different CVSLE problems.

## 1. Introduction

With the development of science and technology, the study for the CVSLE has made great progress, such as a two-step scale-splitting iteration technique for the symmetric CVSLE [1], the blind extraction of communications sources (complex-valued sources) [2], the robot manipulators' control [3], the time-varying linear matrix equation's solution [4], the control of manipulators using image visual servoing scheme [5], the redundant manipulators' control using an adaptive projection neural network [6] and the redundant manipulators' control for tacking the periodic noise [7]. We can describe the CVSLE as

$$HC(t) = a \in \mathbb{C}^n, \tag{1}$$

where $H \in \mathbb{C}^{n\times n}$ and $a \in \mathbb{C}^n$ represent the coefficients in complex field, and $C(t) \in \mathbb{C}^n$ represents a complex-valued vector. Because a complex matrix is composed of the imaginary part and the real part, the complex

2010 *Mathematics Subject Classification*. Primary 68T07
*Keywords*. Complex-valued systems of linear equation, Zhang recurrent neural network models, NRNN-SBP model, NRNN-IRN model
Received: 10 October 2018; Revised: 31 October 2018; Accepted: 21 December 2018
Communicated by Shuai Li
Corresponding author: Lei Ding
This work is supported by the National Natural Science Foundation of China (grant nos. 61866013, 61573298, and 61503152), the Natural Science Foundation of Hunan Province, China (grant nos. 2016JJ2101 and 2017JJ3258), the Research Foundation of Education Bureau of Hunan Province, China (grant no. 15B192), the National Natural Science Foundation of China (grant nos. 61563017, 61561022, 61363073 and 61363033), and the Open Fund Project of Jishou University under grants DNGC1703.
*Email address:* yylxdinglei@126.com (Lei Ding)

matrix $H$, $C$, and $a$ can be represented as $H = H_{re} + jH_{im}$, $C = C_{re} + jC_{im}$, and $a(t) = a_{re}(t) + ja_{im}(t)$, respectively. In the above equation, the symbol $j = \sqrt{-1}$ represents an imaginary unit. So, we can rewrite the equation (1) as

$$[H_{re} + jH_{im}][C_{re}(t) + jC_{im}] = a_{re} + ja_{im} \in \mathbb{C}^n, \tag{2}$$

where $H_{re}, H_{im} \in \mathbb{R}^{n\times n}$, $C_{re}, C_{im} \in \mathbb{R}^n$, and $a_{re}, a_{im} \in \mathbb{R}^n$. Based on the complex equation's principle, we can rewrite the equation (2) as the following equation [8],

$$\begin{cases} H_{re}C_{re}(t) - H_{im}C_{im}(t) = a_{re} \in \mathbb{R}^n, \\ H_{im}C_{re}(t) + H_{re}C_{im}(t) = a_{im} \in \mathbb{R}^n, \end{cases} \tag{3}$$

Now we can represent the equation (3) as the following compact matrix form

$$\begin{bmatrix} H_{re} & -H_{im} \\ H_{im} & H_{re} \end{bmatrix} \begin{bmatrix} C_{re}(t) \\ C_{im}(t) \end{bmatrix} = \begin{bmatrix} a_{re} \\ a_{im} \end{bmatrix} \in \mathbb{R}^{2n}. \tag{4}$$

Now the equation (4) can be written as

$$Ux(t) = V \in \mathbb{R}^{2n}. \tag{5}$$

where $U = \begin{bmatrix} H_{re} & -H_{im} \\ H_{im} & H_{re} \end{bmatrix}$, $x(t) = \begin{bmatrix} C_{re}(t) \\ C_{im}(t) \end{bmatrix}$, and $V = \begin{bmatrix} a_{re} \\ a_{im} \end{bmatrix}$. Therefore, we can solve the CVSLE in the real domain. Now the methods to tackle the real-valued system of linear equation can be employed to tackle the CVSLE [9–14]. Compared with the Frobenius norm, the lagging error is more suitable to act as the performance criterion of the error matrix. So, the ZNNs using the the lagging error can obtain a desired solution at exponential convergence rate instead of infinitely long time [15–17]. Considering we cannot obtain the original ZNN models' desired solution in finite time, Xiao et al. proposed a novel design formula to improve the ZNN models, which can obtain the desired solution within finite time [18]. Using the above design formula proposed by Xiao, Ding et al. proposed a FTRNN model to solve the CVSLE [19].

Compared with a linear activation function, a suitable nonlinear activation function can accelerate the convergence rate to some extent. In [20–25] the sign-bi-power activation function is employed to improve the ZNN models, and it can be represented as

$$\Psi(g) = \mathrm{sgn}^b(g) + \mathrm{sgn}^{1/b}(g), \tag{6}$$

where $0 < b < 1$ and

$$\mathrm{sgn}^b(g) = \begin{cases} |g|^b, & \text{if } \ g > 0 \\ 0, & \text{if } \ g = 0 \\ -|g|^b, & \text{if } \ g < 0. \end{cases}$$

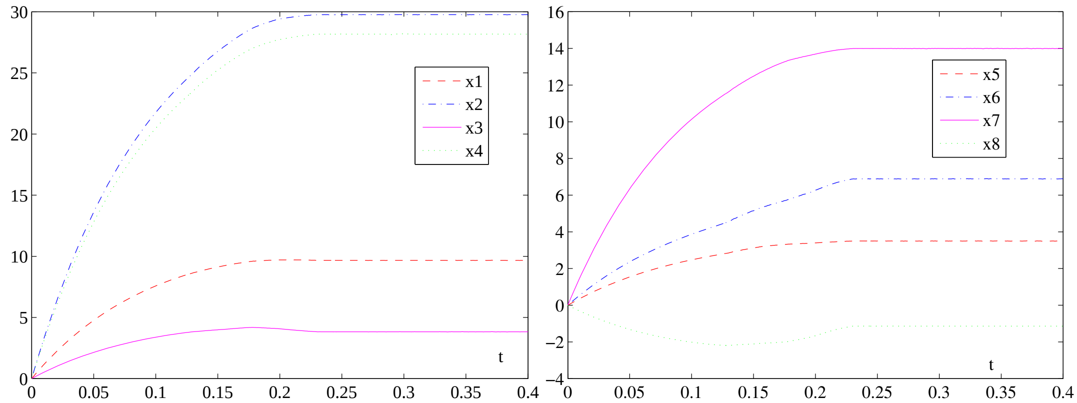
Furthermore Miao et al. proposed a tunable activation function [26], and

$$\Psi(g) = \mathrm{sign}(g)(r_1|g|^b + r_2|g| + r_3|g|^{\frac{1}{b}}), \tag{7}$$

where $0 < p < 1$, $r_1 > 0$, $r_2 > 0$, $r_3 > 0$ and

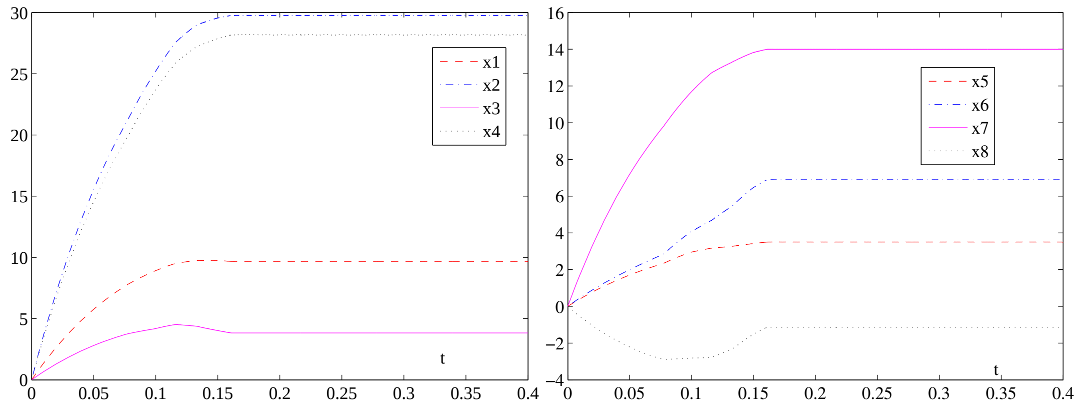
$$\mathrm{sign}(g) = \begin{cases} 1, & \text{if } \ g > 0 \\ 0, & \text{if } \ g = 0 \\ -1, & \text{if } \ g < 0. \end{cases}$$

Compared with the ZNN models using the linear activation function, the ZNN models using the suitable nonlinear activation function will have better convergent effect. As we know, different nonlinear activation function will have different convergence performance, so two different kinds of neural network models for




(a) Element of real part of $C(t)$ (b) Element of imaginary part of $C(t)$

Figure 1: Output trajectories of neural states $C(t)$ synthesized by FTRNN model (12) of example 1




(a) Element of real part of $C(t)$ (b) Element of imaginary part of $C(t)$

Figure 2: Output trajectories of neural states $C(t)$ synthesized by IRNN-SBP model (13) of example 1

tackling the CVSLE can be derived by using a sign-bi-power activation function and a tunable activation function [27–31].

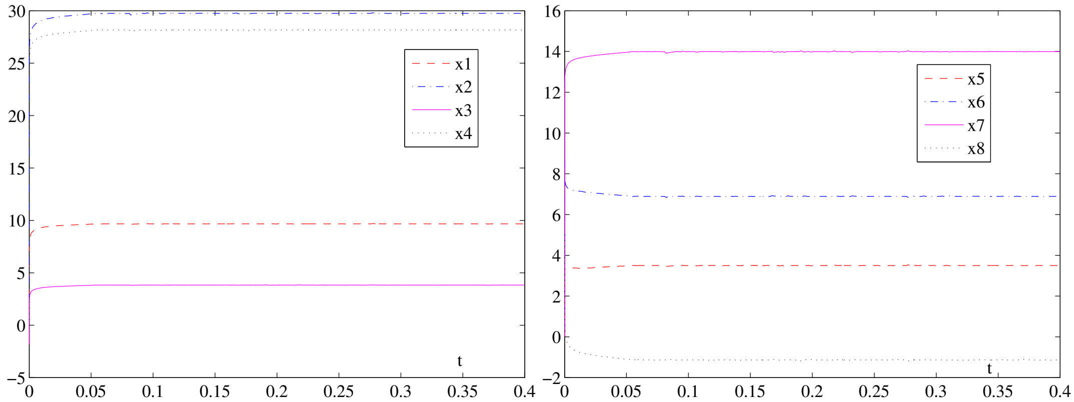
The remaining parts are divided into three sections. Section 2, two different kinds of finite time convergent neural network models for tackling CVSLE are presented. Section 3, the corresponding simulation results are given to verify the performance of the two improved neural network models. Finally, the corresponding conclusions are given in section 4.

Now we can summarize the main contributions of this paper as follows:

1) Two modified ZNN models using different nonlinear activation functions to tackle the CVSLE problems are proposed in this paper.

2) As compared to the ZNN using linear activation function, our proposed models using different nonlinear activation functions have better convergence performance. Furthermore, the corresponding theoretical proof is given to guarantee the convergence property of our proposed models.

3) The results reveal that different nonlinear activation function will have a different effect on the convergence rate for different CVSLE problems.




(a) Element of real part of $C(t)$ (b) Element of imaginary part of $C(t)$

Figure 3: Output trajectories of neural states $X(t)$ synthesized by IRNN-IRN model (14) of example 1

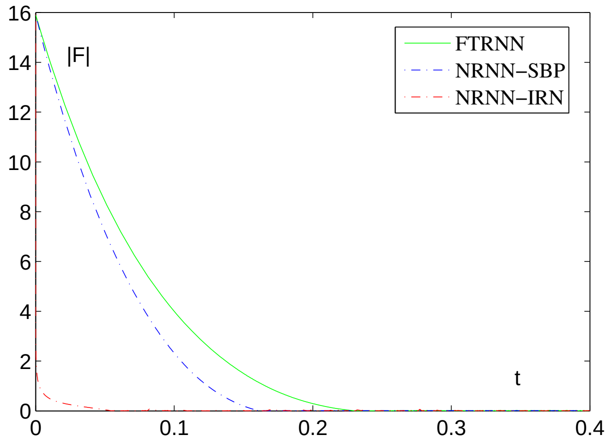



Figure 4: The evolution of the corresponding residual errors of example 1.

## 2. Two Different Finite Time Zhang Neural Network

Now according to the equation (5), we can write the error function $F(t)$ of ZNN as the following formula

$$F(t) = Ux(t) - V \in \mathbb{R}^{2n}. \tag{8}$$

So from the ZNN's design formula $\dot{F}(t) = -q\Psi(F(t))$, we can derive the following formula

$$U\dot{x}(t) = -q\Psi(Ux(t) - V), \tag{9}$$

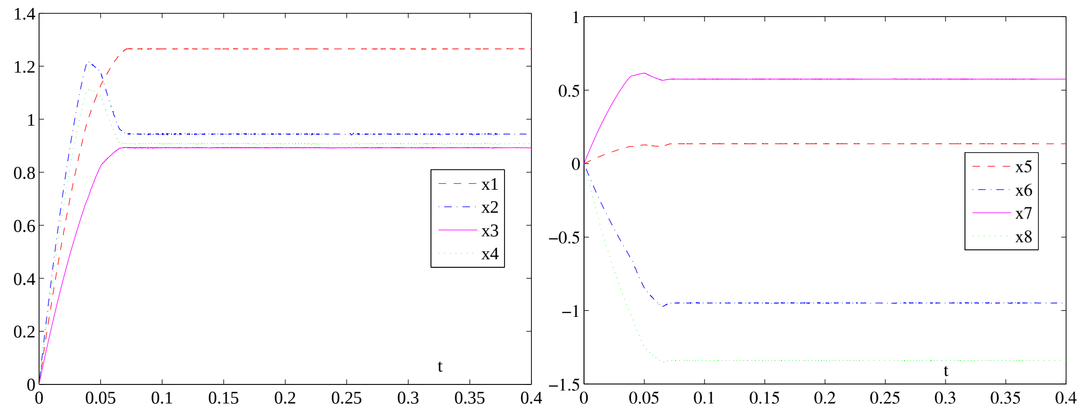
where $q > 0$ is a coefficient for adjusting the convergence rate, and $\Psi(\cdot)$ represents an activation function array. This is an original ZNN model for online tackling the CVSLE .

If we use the new design formula proposed by Xiao [18], we will have

$$\frac{\mathrm{d}F(t)}{\mathrm{d}t} = -q\Psi(s_1F(t) + s_2F^{w/z}(t)), \tag{10}$$

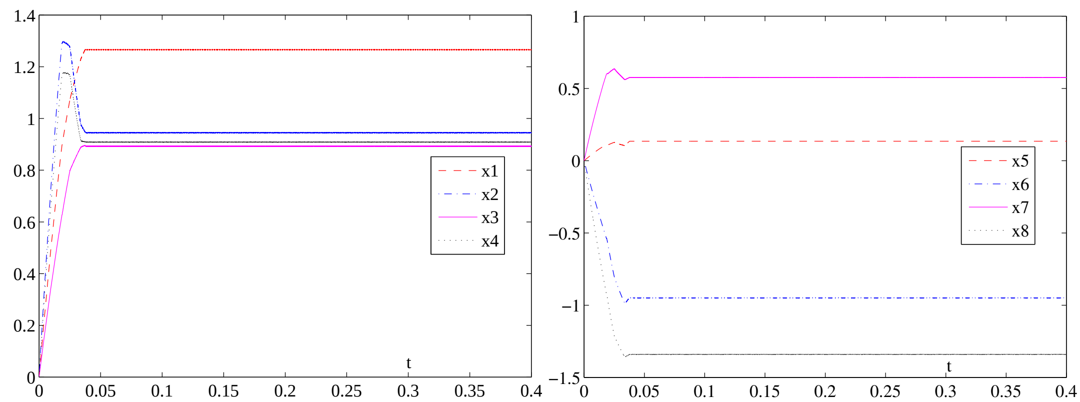
where $s_1$ and $s_2$ meet $s_1 > 0$, $s_2 > 0$, and $z$ and $w$ are the odd integer and meet $z > w > 0$. Then we have

$$U\dot{x}(t) = -q\Psi(s_1(Ux(t) - V) + s_2(Ux(t) - V)^{w/z}). \tag{11}$$




(a) Element of real part of $C(t)$ (b) Element of imaginary part of $C(t)$

Figure 5: Output trajectories of neural states $C(t)$ synthesized by FTRNN model (12) of example 2




(a) Element of real part of $C(t)$ (b) Element of imaginary part of $C(t)$

Figure 6: Output trajectories of neural states $C(t)$ synthesized by IRNN-SBP model (13) of example 2.

If we choose a linear activation function, we can have the following formula

$$U\dot{x}(t) = -q(s_1(Ux(t) - V) + s_2(Ux(t) - V)^{w/z}), \tag{12}$$

which is called the FTRNN model for online dealing with the CVSLE [19].

### 2.1. NRNN-SBP Model

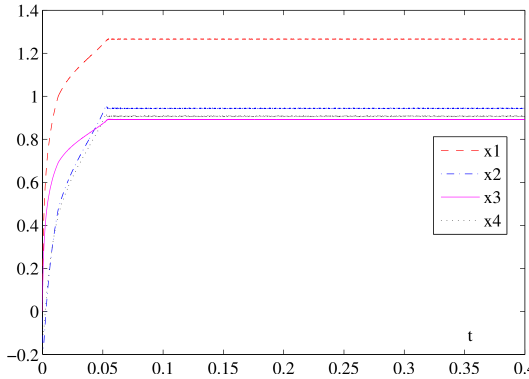
If we introduce a sign-bi-power activation function (6), we can rewrite the equation (12) as

$$\begin{aligned} U\dot{x}(t) = &-q(\mathrm{sgn}^b(s_1(Ux(t) - V) + s_2(Ux(t) - V)^{w/z}) \\ &+\mathrm{sgn}^{1/b}(s_1(Ux(t) - V) + s_2(Ux(t) - V)^{w/z})), \end{aligned} \tag{13}$$

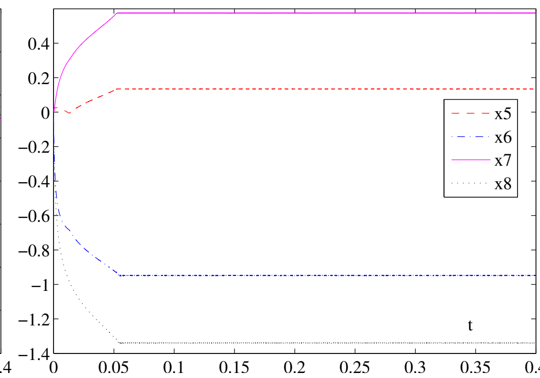
here $q > 0$, $1 > b > 0$, $s_1 > 0$, $s_2 > 0$, and

$$\mathrm{sgn}^b(g) = \begin{cases} |g|^b, & \text{if} \quad g > 0 \\ 0, & \text{if} \quad g = 0 \\ -|g|^b, & \text{if} \quad g < 0. \end{cases}$$

This model is called the NRNN-SBP model for online tackling the CVSLE.




(a) Element of real part of $C(t)$




(b) Element of imaginary part of $C(t)$

Figure 7: Output trajectories of neural states $C(t)$ synthesized by IRNN-IRN model (14) of example 2

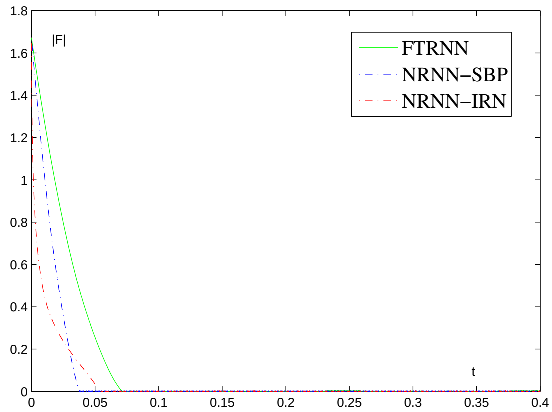



Figure 8: The evolution of the corresponding residual errors of example 2.

### 2.2. *NRNN-IRN Model*

If we introduce a tunable activation function (7), we will have the following formula

$$\begin{aligned} U\dot{x}(t) =& -q\text{sign}(s_1(Ux(t)-V)+s_2(Ux(t)-V)^{w/z})(r_1|s_1(Ux(t)-V) \\ &+s_2(Ux(t)-V)^{w/z}|^b + r_2|s_1(Ux(t)-V)+s_2(Ux(t)-V)^{w/z}| \\ &+r_3|s_1(Ux(t)-V)+s_2(Ux(t)-V)^{w/z}|^{\frac{1}{b}}), \end{aligned} \tag{14}$$

where $q>0$, $1>b>0$, $r_1>0$, $r_2>0$, $r_3>0$, $s_1>0$, $s_2>0$, $z$ and $w$ are the odd integer and satisfy $z>w>0$, and

$$\text{sign}(g)=\begin{cases} 1, & \text{if } \ g>0 \\ 0, & \text{if } \ g=0 \\ -1, & \text{if } \ g<0. \end{cases}$$

Then the equation (14) is called the NRNN-IRN model for online tackling the CVSLE.

### 2.3. *Theorem Analysis*

*Theorem:* The state matrix $x(t)$ of the model (13) (or the model (14)) will converge to its theoretical solution, whatever its initial random value $x(0)$ is.

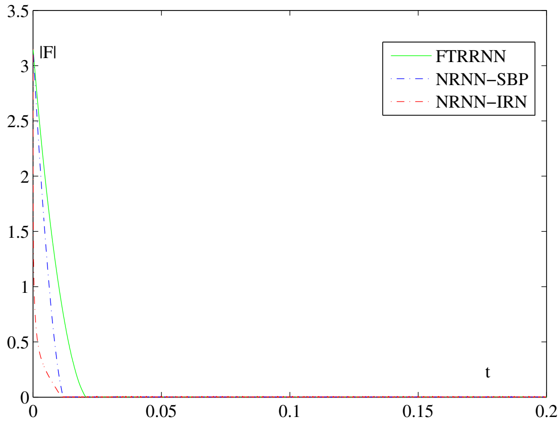



Figure 9: The evolution of the corresponding residual errors of example 3.

*Proof:* According the definition $F(t) = Ux(t) - V$, the model (13) can be written as

$$\dot{F}(t) = -q\Psi_1(s_1F(t) + s_2F(t)^{w/z}), \tag{15}$$

where $\Psi_1(\cdot) = \text{sgn}^b(\cdot)$. Now suppose $R(F(t)) = s_1F(t) + s_2F(t)^{w/z}$, we have

$$\dot{F}(t) = -q\Psi_1(R(F(t))). \tag{16}$$

As we know, the matrix $F(t)$'s each element shows the identical dynamic, then we have

$$\dot{f}_{ij}(t) = -q\Psi_1(R(f_{ij}(t))), \tag{17}$$

where $f_{ij}(t)$ represents the matrix $F(t)$'s $ij$th element. Now we define the following Lyapunov function $V_f = f_{ij}^2(t)$. Then we have

$$\dot{V}_f = -2qf_{ij}(t)\Psi_1(R(f_{ij}(t))). \tag{18}$$

As $\Psi_1(\cdot)$ and $R(\cdot)$ are all odd and monotonic increasing functions, $\Psi_1(R(f_{ij}(t)))$ must be also monotonic increasing. So we can find that $\dot{V}_f$ is negative definite and $f_{ij}(t)$ will globally converge to 0. This means that each element of matrix $x(t)$ will obtain its theoretical solution with time.

Similarly, we can use the above method to prove the convergence property of the model (14).

This proof is successful. ■

## 3. Computer Simulation

Now the following three illustrative examples will be carried to compare the NRNN-SBP model (13) and the NRNN-IRN model (14) with the FTRNN model (12). To display the different convergent rate, the trail of each neural-state and the residual error norm $\|F(t)\|_2$ are shown in the corresponding figures. For comparison, we choose a same group of parameters $q = 10$, $w = 1$, $b = \frac{1}{8}$, $z = 5$, $w = 1$, $r_1 = r_2 = r_3 = 1$, $s_1 = s_2 = 1$ in the example 1, the example 2 and the example 3.

Example 1:

$$H_1C_1(t) = a_1 \in \mathbb{C}^n,$$

where

$$H_1 = \begin{bmatrix} -0.5444+0.6754j & -0.8967-0.5434j & 0.2352-0.7875j & 0.7854+0.9544j \\ 0.7724+0.8944j & -0.8244+0.8478j & -0.8525-0.5486j & 0.8651-0.3564j \\ 0.8544-0.3521j & -0.8215-0.6234j & -0.7445+0.3647j & 0.7443+0.9634j \\ 0.7455-0.3647j & -0.7542+0.3645j & 0.7411+0.3644j & 0.8564-0.7565j \end{bmatrix}$$

and

$$a_1 = \begin{bmatrix} 4.5634 + 8.5445j \\ 2.3432 + 5.8452j \\ 3.4533 - 7.3564j \\ 4.5323 - 5.7842j \end{bmatrix}.$$

According to equation (5) we have

$$U_1 = \begin{bmatrix} -0.5444 & -0.8967 & 0.2352 & 0.7854 & -0.6754 & 0.5434 & 0.7875 & -0.9544 \\ 0.7724 & -0.8244 & -0.8525 & 0.8651 & -0.8944 & -0.8478 & 0.5486 & 0.3564 \\ 0.8544 & -0.8215 & -0.7445 & 0.7443 & 0.3521 & 0.6234 & -0.3647 & -0.9634 \\ 0.7455 & -0.7542 & 0.7411 & 0.8564 & 0.3647 & -0.3645 & -0.3644 & 0.7565 \\ 0.6754 & -0.5434 & -0.7875 & 0.9544 & -0.5444 & -0.8967 & 0.2352 & 0.7854 \\ 0.8944 & 0.8478 & -0.5486 & -0.3564 & 0.7724 & -0.8244 & -0.8525 & 0.8651 \\ -0.3521 & -0.6234 & 0.3647 & 0.9634 & 0.8544 & -0.8215 & -0.7445 & 0.7443 \\ -0.3647 & 0.3645 & 0.3644 & -0.7565 & 0.7455 & -0.7542 & 0.7411 & 0.8564 \end{bmatrix}$$

and $V_1 = [4.5634 \quad 2.3432 \quad 3.4533 \quad 4.5323 \quad 8.5445 \quad 5.8452 \quad -7.3564 \quad -5.7842]^{\mathrm{T}}$, where T represents the matrix $V_1$'s transpose.

Example 2:

$$H_2C_2(t) = a_2 \in \mathbb{C}^n.$$

Suppose $H_2 = H_1$, and

$$a_2 = \begin{bmatrix} 0.5121 + 0.3654j \\ 0.7451 + 0.3564j \\ 0.8542 - 0.3647j \\ 0.8422 - 0.3843j \end{bmatrix}.$$

Similarly, according to equation (5) we have

$$U_2 = \begin{bmatrix} -0.5444 & -0.8967 & 0.2352 & 0.7854 & -0.6754 & 0.5434 & 0.7875 & -0.9544 \\ 0.7724 & -0.8244 & -0.8525 & 0.8651 & -0.8944 & -0.8478 & 0.5486 & 0.3564 \\ 0.8544 & -0.8215 & -0.7445 & 0.7443 & 0.3521 & 0.6234 & -0.3647 & -0.9634 \\ 0.7455 & -0.7542 & 0.7411 & 0.8564 & 0.3647 & -0.3645 & -0.3644 & 0.7565 \\ 0.6754 & -0.5434 & -0.7875 & 0.9544 & -0.5444 & -0.8967 & 0.2352 & 0.7854 \\ 0.8944 & 0.8478 & -0.5486 & -0.3564 & 0.7724 & -0.8244 & -0.8525 & 0.8651 \\ -0.3521 & -0.6234 & 0.3647 & 0.9634 & 0.8544 & -0.8215 & -0.7445 & 0.7443 \\ -0.3647 & 0.3645 & 0.3644 & -0.7565 & 0.7455 & -0.7542 & 0.7411 & 0.8564 \end{bmatrix}$$

and $V_2 = [0.5121 \quad 0.7451 \quad 0.8542 \quad 0.8422 \quad 0.3654 \quad 0.3564 \quad -0.3647 \quad -0.3843]^{\mathrm{T}}$, where T represents the matrix $V_2$'s transpose.

Example 3:

$$H_3C_3(t) = a_3 \in \mathbb{C}^n.$$

Suppose $H_3 = H_2$, and

$$a_3 = \begin{bmatrix} 1.5321 + 0.2444j \\ 1.2845 + 0.9589j \\ 1.2845 - 0.9589j \\ 1.5436 \end{bmatrix}$$

Then ccording to equation (5) we have

$$U_3 = \begin{bmatrix} -0.5444 & -0.8967 & 0.2352 & 0.7854 & -0.6754 & 0.5434 & 0.7875 & -0.9544 \\ 0.7724 & -0.8244 & -0.8525 & 0.8651 & -0.8944 & -0.8478 & 0.5486 & 0.3564 \\ 0.8544 & -0.8215 & -0.7445 & 0.7443 & 0.3521 & 0.6234 & -0.3647 & -0.9634 \\ 0.7455 & -0.7542 & 0.7411 & 0.8564 & 0.3647 & -0.3645 & -0.3644 & 0.7565 \\ 0.6754 & -0.5434 & -0.7875 & 0.9544 & -0.5444 & -0.8967 & 0.2352 & 0.7854 \\ 0.8944 & 0.8478 & -0.5486 & -0.3564 & 0.7724 & -0.8244 & -0.8525 & 0.8651 \\ -0.3521 & -0.6234 & 0.3647 & 0.9634 & 0.8544 & -0.8215 & -0.7445 & 0.7443 \\ -0.3647 & 0.3645 & 0.3644 & -0.7565 & 0.7455 & -0.7542 & 0.7411 & 0.8564 \end{bmatrix}$$

and $V_3 = [1.5321 \quad 1.2845 \quad 1.2845 \quad 1.5436 \quad 0.2444 \quad 0.9589 \quad -0.9589 \quad 0]^{\mathrm{T}}$, where T represents the matrix $V_3$'s transpose.

From the Figs.1-9, we can find the NRNN-SBP model (13) and the NRNN-IRN model (14) have higher convergence rate than the FTRNN model (12). Furthermore from the example 1 and the example 3, we can find that the NRNN-IRN model has the fastest convergence rate. However, from the example 2, we can find that the NRNN-SBP model has the fastest convergence rate. This phenomenon reveals that different nonlinear activation function will have a different effect on the convergence rate for different CVSLE problems.

## 4. Conclusions

It is the first time to build two different finite-time convergent ZNN models to tackle the CVSLE in this paper. The results show that the convergence speed of NRNN-SBP and the NRNN-IRN is faster than that of the FTRNN model, and also reveal that different nonlinear activation function will have a different effect to accelerate the convergence rate for different CVSLE problems. So how to choose a suitable nonlinear activation function for a different CVSLE problem will be our next research.

## References


[1] D. K. Salkuyeh, T. S. Siahkolaei, Two-parameter TSCSP method for solving complex symmetric system of linear equations, Neurocomputing 55 (2018) 8–28.

[2] K. Subramanian, R. Savitha, S. Suresh, A complex-valued neuro-fuzzy inference system and its learning mechanism, Neurocomputing 123 (2014) 110–120.

[3] R. Venkatesh Babu, S. Suresh, R. Savitha, Human action recognition using a fast learning fully complex-valued calssifier, Neurocomputing 89 (2012) 202–212.

[4] I. Duran-Diaz, S. Cruces, M. A. Sarmiento-Vega, P. Aguilera-Bonet, Cyclic maximization of non-gaussianity for blind signal extraction of complex-valued sources, Neurocomputing 74 (2011) 2867–2873.

[5] Y. Zhang, S. Li, A Neural Controller for Image-Based Visual Servoing of Manipulators With Physical Constraints, IEEE Transactions on Neural Networks and Learning Systems, 29 (2018) 5419–5429.

[6] Y. Zhang, S. Chen, S. Li, Z. Zhang, Adaptive Projection Neural Network for Kinematic Control of Redundant Manipulators With Unknown Physical Parameters, IEEE Transactions on Industrial Electronics, 65 (2018) 4909–4920.

[7] Y. Zhang, S. Li, S. Kadry, B. Liao, Recurrent Neural Network for Kinematic Control of Redundant Manipulators With Periodic Input Disturbance and Physical Constraints, IEEE Transactions on Cybernetics, (2018) 1–12.

[8] W. liao, J. Wang, J. Wang, A recurrent neural network for solvimng complex-valued quadratic programming problems with equality constraints, In: Tan, Y., Shi Y., Tan K.C.(eds) ICSI 2010, vol. 6146, Part II. LNCS,. Heidelberg: Springer, (2010) 321–326.

[9] D. Guo, C. Yi, Y. Zhang, Zhang neural network versus gradient-based neural network for time-varying linear matrix equation solving, Neurocomputing 74 (2011) 3708–3712.

[10] L. Xiao, A nonlinearly-activated neurodynamic model and its finite-time solution to equality-constrained quadratic optimization with nonstationary coefficients, Appl. Soft Comput., 40 (2016) 252–259.

[11] L. Xiao, R. Lu, Finite-time solution to nonlinear equation using recurrent neural dynamics with a specially-constructed activation function, Neurocomputing, 151, (2015) 246–251.

[12] L. Xiao, Y. Zhang, Solving time-varying inverse kinematics problem of wheeled mobile manipulators using Zhang neural network with exponential convergence, Nonlinear Dyn. 76(2) (2014) 1543-1559.

[13] S. Srivastava, P. S. Stanimirovi, V. N. Katsikis, and D. K. Gupta, A Family of Iterative Methods with Accelerated Convergence for Restricted Linear System of Equations, Mediterranean Journal of Mathematics, 14 (2017) 222.

[14] L. Xiao, W. Meng, R. Lu, X. Yang, B. Liao, and L. Ding, 'A fully complex-valued neural network for rapid solution of complex-valued systems of linear equations, ISNN 2015, LNCS 9377, (2015) 444–451.

[15] D. Guo, Z. Nie, L. Yan, Novel Discrete-Time Zhang Neural Network for Time-Varying Matrix Inversion, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47 (2017) 2301-2310.

[16] L. Xiao, A finite-time convergent neural dynamics for online solution of time-varying linear complex matrix equation, Neurocomputing, 167, (2015) 254–259.

[17] Q. Xiang, B. Liao, L. Xiao, L. Lin, S. Li, Discrete-time noise-tolerant Zhang neural network for dynamic matrix pseudoinversion, Soft Computing, (2018) 1–12.

[18] L. Xiao, A New Design Formula Exploited for Accelerating Zhang Neural Network and Its Application to Time-Varying Matrix Inversion, Theoretical Computer Science, 647 (2016) 50–58.

[19] L. Ding, L. Xiao, B. Liao, R. Lu, H. Peng, An Improved Recurrent Neural Network for Complex-Valued Systems of Linear Equation and Its Application to Robotic Motion Tracking, Frontiers in Neurobotics, 11 (2017) 1–8.

[20] S. Li, S. 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.

[21] L. Xiao, S. Li, *et al.*, Zeroing neural dynamics for control design: comprehensive analysis on stability, robustness, and convergence speed, IEEE Trans. Ind. Inform., to be published, 2018.

[22] L. Xiao, S. Li, *et al.*, Co-design of finite-time convergence and noise suppression: a unified neural model for time varying linear equations with robotic applications, IEEE Trans. Syst., Man, Cybern., Syst., to be published, 2018.

[23] L. Xiao, Z. Zhang, *et al.*, Design, verification and robotic application of a novel recurrent neural network for computing dynamic Sylvester equation, Neural Netw. 105 (2018) 185-196.

[24] S. Li, Y. Li, Nonlinearly Activated Neural Network for Solving Time-Varying Complex Sylvester Equation, IEEE Trans Cyber, 44 (2014) 1397-1407.

[25] L. Xiao, B. Liao, A convergence-accelerated Zhang neural network and its solution application to Lyapunov equation, Neurocomputing, 193 (2016) 213–218.

[26] P. Miao, Y. Shen, Y. Huang, Y. Wang, Solving time-varying quadratic programs based on finite-time Zhang neural networks and their application to robot tracking, Neural Comput and Applic, 26 (2015) 693–703.

[27] L. Xiao, A finite-time convergent Zhang neural network and its application to real-time matrix square root finding, Neural Comput. Appl., to be published, 2018.

[28] L. Xiao, Z. Zhang, and S. Li, "Solving Time-varying system of nonlinear equations by finite-time recurrent neural networks with application to motion tracking of robot manipulators," IEEE Trans. Syst., Man, Cybern., Syst., to be published, 2018.

[29] L. Jin, S. Li, L. Xiao, R. Lu, and B. Liao, Cooperative motion generation in a distributed network of redundant robot manipulators with noises, IEEE Trans. Syst. Man Cybern. Syst., 48(10), (2017) 1715-1724.

[30] L. Xiao, B. Liao, S. Li, and K. Chen, "Nonlinear recurrent neural networks for finite-time solution of general time-varying linear matrix equations," *Neural Netw.*, vol. 98, pp. 102–113, 2018.

[31] L. Xiao, B. Liao, S. Li, *et al.*, Design and analysis of FTZNN applied to real-time solution of nonstationary Lyapunov equation and tracking control of wheeled mobile manipulator, IEEE Trans. Industr. Inform. 14(1) (2018) 98-105.