mathematics

MDPI

Article

A New Varying-Factor Finite-Time Recurrent Neural Network to Solve the Time-Varying Sylvester Equation Online

Haoming Tan [1], Junyun Wu [2,3,*], Hongjie Guan [4], Zhijun Zhang [5], Ling Tao [3,4], Qingmin Zhao [3,4] and Chunquan Li [3,4]

[1] School of Electric and Electronic Enginnering, Shanghai University of Engineering Science, Shanghai 201620, China; m320122407@sues.edu.cn
[2] School of Mathematics and Computer Sciences, Nanchang University, Nanchang 330031, China
[3] Jiangxi Provincial Key Laboratory of Intelligent Systems and Human-Machine Interaction, Nanchang 330031, China; taoling@ncu.edu.cn (L.T.); zhaoqm@ncu.edu.cn (Q.Z.); lichunquan@ncu.edu.cn (C.L.)
[4] School of Information Engineering, Nanchang University, Nanchang 330031, China; 411016620039@email.ncu.edu.cn
[5] School of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China; auzjzhang@scut.cn
* Correspondence: wujunyun@ncu.edu.cn

Abstract: This paper presents a varying-parameter finite-time recurrent neural network, called a varying-factor finite-time recurrent neural network (VFFTRNN), which is able to solve the solution of the time-varying Sylvester equation online. The proposed neural network makes the matrix coefficients vary with time and can achieve convergence in a finite time. Apart from this, the performance of the network is better than traditional networks in terms of robustness. It is theoretically proved that the proposed neural network has super-exponential convergence performance. Simulation results demonstrate that this neural network has faster convergence speed and better robustness than the return to zero neural networks and can track the theoretical solution of the time-varying Sylvester equation effectively.

Keywords: recurrent neural network (RNN); finite time; super-exponential convergence rate

MSC: 68T07; 68U01; 68U07

Citation: Tan, H.; Wu, J.; Guan, H.; Zhang, Z.; Tao, L.; Zhao, Q.; Li, C. A New Varying-Factor Finite-Time Recurrent Neural Network to Solve the Time-Varying Sylvester Equation Online. *Mathematics* **2024**, *12*, 3891. https://doi.org/10.3390/math12243891

Academic Editor: Zhanybai T. Zhusubaliyev

Received: 11 November 2024
Revised: 4 December 2024
Accepted: 5 December 2024
Published: 10 December 2024

1. Introduction

As we all know, the Sylvester equation $UP - PV + W = 0$, as a kind of linear matrix equation, plays a crucial role in many fields, especially linear algebra and matrix theory. For some mathematics, dynamic control systems, and computer vision problems, the Sylvester equation can be used as a solution, like designing a feedback control system by pole assignment optimization, linear least-squares regression, image fusion [1], clustering [2,3], and so on. In recent years, many methods have been proposed by other scholars to solve real-time linear algebra problems [4–18]. Among these methods, several are efficient and representative, like gradient-based neural network (GNN) methods and zeroing neural network (ZNN) methods. It is notable that GNN and ZNN methods both use the idea to force the time derivative of the error function decline and further to force the error function gain toward zero. The proposal of GNNs, as emblematic recurrent neural networks, is very representational for solving real-time linear algebra problems [19–21]. Generally speaking, different GNNs are designed to solve different linear algebra problems. By itself, for solving the Sylvester equation, the GNN method is roughly stated as follows:

(1) In the first step, the error function is defined as $E = ||UP - PV + W||^2/2$. In order to acquire the solution of the Sylvester equation, the error function should converge to zero.

(2) In the next step, for the purpose of acquiring the minimum point of the error function, GNN methods use a negative gradient descent direction. In addition to this, GNN methods employ a constant scalar-type parameter, which accelerates the convergence rate of the error function to solve the Sylvester equation efficiently.

However, when the task is time-varying, the traditional GNN method, which is based on gradient-based optimization, has difficulty tracking the ideal solution of the time-varying Sylvester equation. This is because the error, as defined by $E = \frac{||UP-PV+W||^2}{2}$, cannot converge to zero in such cases, resulting in residual errors [22]. Therefore, a GNN cannot solve the time-varying Sylvester function effectively. But in practice, time-varying problems are everywhere, so we need a neural network to better track locus solutions to time-varying issues in real-time. Under this background, many kinds of neural networks have been proposed to solve this kind of time-varying algebraic problem. Among these neural networks, the ZNN proposed by Zhang et al. proved to be efficient [12]. The ZNN is a kind of implicit dynamic neural network that skillfully replaces the gradient dynamic with an implicit dynamic, so that the computation error can achieve better convergence; that is, with the increase in time, the solution of the network can approach the theoretical solution in real time [23–31]. However, the ZNN also has some shortcomings, such as its convergence speed not being fast enough, convergence parameters being difficult to set, and other issues. Aiming to address the above shortcomings, Li et al. proposed a finite-time ZNN [23]. The sign-bi-power activation function is used in this network, as it effectively activates this network, makes the convergence speed much faster, and can achieve finite-time convergence. In addition, Zhang et al. proposed a variable-parameter convergence differential neural network (VP-CDNN) [32]. Different from the initial ZNN, the convergence parameter of the VP-CDNN changes with time, which makes the network hardware implementation much more reasonable than the ZNN, and the convergence speed reaches the super-exponential convergence speed. But despite this, the VP-CDNN cannot achieve finite-time convergence. Based on the above discussion, this paper proposes a recurrent neural network (VFFTRNN) for solving the time-varying Sylvester equation. The network not only has the speed of super-exponential convergence but also can achieve finite-time convergence.

The rest of this paper is divided into four parts. The second section introduces some tools needed to solve the Sylvester equation, and briefly describes the solution steps of the ZNN. In the third section, the implicit dynamic equation of the proposed VFFRTNN and the general steps of solving time-varying problems with the network are given, and the convergence and robustness of VFFTRNN under different conditions are analyzed theoretically. In the fourth section, the simulation results of MATLAB are given and compared with the simulation results of the ZNN. In the last section, the thesis is summarized.

Before concluding this section, some highlights of the paper are listed below:

(1) We propose a VFFTRNN algorithm for solving the Sylvester equation.
(2) Compared with VP-CDNN, this network can achieve finite-time convergence.
(3) We theoretically prove the convergence time of the network in the case of three activation functions.
(4) In the situation considering the existence of disturbances, the robustness of VFFTRNN is discussed.
(5) Data simulation results show that the proposed VFFTRNN has better convergence and robustness than the traditional ZNN model under the same initial conditions.

2. Problem Formulation and Knowledge Preparation

Let us observe the following Sylvester equation with smooth time-varying characteristics:

$$U(t)P(t) - P(t)V(t) + W(t) = 0, \quad \forall t \in [0, +\infty) \tag{1}$$

where time is expressed as the variable t. The time-varying smooth coefficient matrices are $W(t) \in \Re^{m\times n}$, $V(t) \in \Re^{n\times n}$, as well as $U(t) \in \Re^{m\times m}$. In addition, $\dot{W}(t) = \mathrm{d}W(t)/\mathrm{d}t$, $\dot{V}(t) = \mathrm{d}V(t)/\mathrm{d}t$, and $\dot{U}(t) = \mathrm{d}U(t)/\mathrm{d}t$ are their time derivatives, respectively, assuming these time derivatives are known or can be found exactly. Furthermore, $P(t) \in \Re^{m\times n}$ is uncharted but real, and our purpose is to find it to approach or even obtain the unique ideal solution $P^*(t)$ in finite time for the above Sylvester equation.

In order to facilitate the subsequent discussion and to solve Equation (1) more simply, (1) is best to be transformed into a vector form. The following Theorem (1) is introduced.

Theorem 1. *The Sylvester equation in matrix form (i.e., Equation (1)) can be transformed into vector form as follows:*

$$\Big(I_n \otimes U(t) - V^T(t) \otimes I_m\Big)vec(P(t)) + vec(W(t)) = 0 \tag{2}$$

where I_n and I_m represent the n-dimensional and m-dimensional identity matrices, respectively. Additionally, the symbol T represents the transpose of the matrix. The symbol $\otimes$ denotes the Kronecker product. In other words, $U \otimes V \in \Re^{mn\times mn}$ represents a large matrix, which is equivalent to transforming the ith row and $j-$column position element u_{ij} in the matrix U into a matrix $u_{ij}V$. The operation $vec(P(t)) \in \Re^{mn}$ represents a column vector obtained by stacking the columns of the matrix $P(t) \in \Re^{m\times n}$ in sequence.

Proof. See Section 2 and [12], Appendix. □

Furthermore, if we want to obtain the unique solution of Equation (1), then it must meet the regularity condition [12,33], which is described in the following Theorem (2).

Theorem 2. *If the Sylvester Equation (1) satisfies the following regularity condition, then it has a unique solution.*

$$\begin{aligned} &I_n \otimes \Big(U^T(t)U(t)\Big) - V^T(t) \otimes U^T(t) - V(t) \otimes U(t) \\ &+\Big(V(t)V^T(t)\Big) \otimes I_m \geq \alpha I_{mn}, \exists \alpha > 0, \forall t \in [0, +\infty) \end{aligned} \tag{3}$$

where I_{mn} represents the mn-dimensional identity matrix.

Proof. See Section 2 and [12], Appendix. □

With the continuous development of neural dynamics and the proposal of ZNNs, the concept of ZNNs has been applied in many engineering and scientific fields to solve time-varying problems. To clarify the details of this paper, we will next briefly introduce how to solve the Sylvester Equation (1) using a typical ZNN method.

In the first step, an error function for Equation (1) is defined as follows:

$$E(t) := U(t)P(t) - P(t)V(t) + W(t) \tag{4}$$

Next, since our goal is to obtain a unique solution to Equation (1), when we set the error function to approach 0, the unique solution of Equation (1) will be obtained naturally. In light of the design principles of neurodynamics, the following differential equations for the error function $E(t)$ can be designed:

$$\frac{dE(t)}{dt} = \dot{E}(t) = -\mu\Phi(E(t)) \tag{5}$$

where the scale-valued factor μ is a positive design constant (i.e., $\mu > 0$), which can adjust the convergence velocity of this network (5) by varying its value. $\Phi(\cdot) : \Re^{m\times n} \to \Re^{m\times n}$ is an array consisting of the same activation functions. $\Phi(\cdot)$ can represent different types of

activation functions, such as linear-type, bipolar-sigmoid-type, power-type, sign-bi-power-type, and so on. When we use the activation function of the linear type, (5) becomes a typical ZNN.

Based on (5), we can obtain the following implicit dynamic equation:

$$\begin{aligned} &U(t)\dot{P}(t) - \dot{P}(t)V(t) + \dot{U}(t)P(t) - P(t)\dot{V}(t) \\ &+\dot{W}(t) = -\mu\Phi(U(t)P(t) - P(t)V(t) + W(t)) \end{aligned} \tag{6}$$

where the starting value of $P(t)$ is $P(0) = P_0 \in R^{m\times n}$. Additionally, the state matrix $P(t)$ is an approximate solution that will gradually approach the theoretical solution, and it will converge to the unique solution of (1). In light of the following lemma [12], the performance of the ZNNs in solving the Sylvester equation can be guaranteed to exhibit exponential convergence.

Lemma 1. *Considering the Sylvester Equation (1), with its time-varying coefficient matrices* $U(t) \in \Re^{m\times m}$, $V(t) \in \Re^{n\times n}$, *and* $W(t) \in \Re^{m\times n}$, *and assuming that they satisfy inequality (3), if the activation function of the implicit dynamic Equation (6) is chosen to be linear, then its state matrix* $P(t)$ *will exponentially converge from any initial state* P_0 *to the time-varying ideal solution* $P^*(t)$ *of (1).*

3. Varying-Factor Finite-Time Recurrent Neural Networks

ZNN can achieve better convergence performance by using different activation functions or setting a larger constant factor [34], but its convergence still has room for improvement. Inspired by classic ZNN design principles, a novel recurrent neural network is proposed in this paper. It can achieve a neural network with super-exponential convergence within a limited time, and because the factor before its activation function changes with time, it is called VFFTRNN. It can be expressed as

$$\begin{aligned} \dot{E}(t) &= -\Gamma(t)\Big(\Phi(E(t)) + \Phi(E^{\lambda}(t))\Big) \\ &= -(t^{\alpha} + \alpha)\Big(\Phi(E(t)) + \Phi(E^{\lambda}(t))\Big) \end{aligned} \tag{7}$$

where $\Gamma(t)$ is a function of t, and it can be set to different functions. For simplicity, we use exponential time-varying parameters $(t^{\alpha} + \alpha)$, where α is a positive scalar-valued factor (i.e., $\alpha > 0$), and the positive scalar-valued factor $\lambda = \frac{s}{q} \in (0,1)$, where s and q are odd numbers. In light of (7), we can reformulate its implicit dynamic equation as

$$\begin{aligned} &\dot{U}(t)P(t) + U(t)\dot{P}(t) - \dot{P}(t)V(t) - P(t)\dot{V}(t) + \dot{W}(t) \\ &= -(t^{\alpha} + \alpha)\Big(\Phi\big(U(t)P(t) - P(t)V(t) + W(t)\big) \\ &\quad + \Phi\big(U(t)P(t) - P(t)V(t) + W(t)\big)^{\lambda}\Big) \end{aligned} \tag{8}$$

In addition, in light of Theorem 1 and Equation (2) in Section 1, we are able to obtain the vector form of the above implicit dynamic Equation (8):

$$\begin{aligned} N(t)\dot{p}(t) = &-\dot{N}(t)p(t) - vec\big(\dot{W}(t)\big) \\ &-(t^{\alpha} + \alpha)\Big(\Phi\big(N(t)p(t) + vec(W(t))\big) \\ &+\Phi\big(N(t)p(t) + vec(W(t))\big)^{\lambda}\Big) \end{aligned} \tag{9}$$

where $N(t) := I_n \otimes U(t) - V^T(t) \otimes I_m \in \Re^{mn\times mn}$ is a matrix, $p(t) := vec(P(t)) \in \Re^{mn}$ is a vector, and $\Phi(\cdot) : \Re^{mn\times 1} \to \Re^{mn\times 1}$ is an activation function array. It is noteworthy that the transformation dimensions of $\Phi(\cdot)$ in (9) are different from those in (5). The block diagram of VFFTRNN can be drawn based on the implicit dynamic Equation (8). As shown

in Figure 1, $\int$ and $\sum$ represent the integrator and accumulator, respectively. The symbols L and R represent left and right matrix multiplication, respectively.

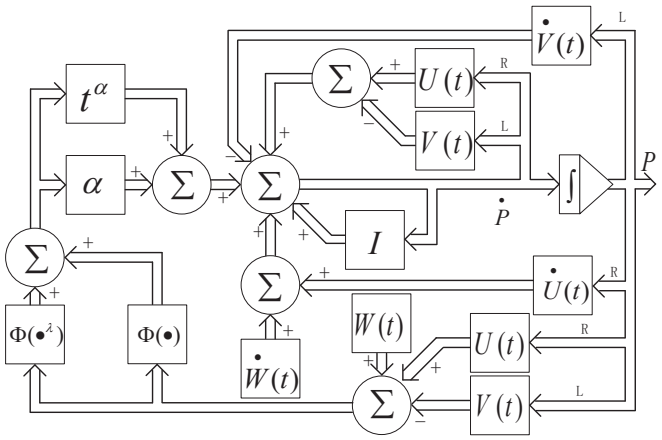

Figure 1. Block diagram of VFFTRNN model (8) for solving the Sylvester equation.

3.1. Convergence Analysis

The following theorem will lay the foundation for later reference when discussing the global convergence of VFFTRNN.

Theorem 3. *For the time-varying Sylvester Equation (1), its coefficient matrices $U(t) \in \Re^{m\times m}$, $V(t) \in \Re^{n\times n}$, and $W(t) \in \Re^{m\times n}$ are given. If Equation (1) satisfies inequality (3), and the activation function sequence is chosen to be strictly increasing, then the state matrix $P(t)$ of the VFFTRNN model (8) will globally converge from any initial state P_0 to the time-varying ideal solution $P^*(t)$ of the Sylvester Equation (1).*

Proof. First, we can convert the VFFTRNN system (7) into scalar form as follows:

$$\begin{aligned}\dot{e}_{ij}(t) = -(t^{\alpha}+\alpha)\left(\phi\left(e_{ij}(t)\right)+\phi\left(e_{ij}^{\lambda}(t)\right)\right)\\ \forall i \in \{1,2,...m\}, j \in \{1,2,...n\}\end{aligned} \tag{10}$$

where $e_{ij}(t)$ represents the ith row and $j-$ column position element of $E(t)$. In addition, $\phi(\cdot)$ represents the activation subfunction for the matrix elements.

Next, we will prove its global convergence. Let us define a Lyapunov function as shown below:

$$v_{ij}(t) = \frac{1}{2}e_{ij}^2(t), \forall i \in \{1,2,...m\}, j \in \{1,2,...n\} \tag{11}$$

According to the above, we can calculate its time derivative as follows:

$$\begin{aligned}\dot{v}_{ij}(t) &= e_{ij}(t)\dot{e}_{ij}(t)\\ &= -(t^{\alpha}+\alpha)e_{ij}\left(\phi\left(e_{ij}(t)\right)+\phi\left(e_{ij}^{\lambda}(t)\right)\right)\end{aligned}$$

It can be seen from the previous definition that $\alpha > 0$ and $\phi(\cdot)$ is monotonic and strictly increasing. Thus, for any $t \geq 0$ and $e_{ij}(t)$, we have

$$\dot{v}_{ij}(t)\begin{cases}=0 & e_{ij}(t)=0\\ \leq 0 & e_{ij}(t)\neq 0.\end{cases}$$

In light of Lyapunov stability theory, the error function (10) is globally asymptotically convergent. Therefore, for any i and j in the domain, the error function $e_{ij}(t)$ in scalar

form will asymptotically converge to zero. In other words, the online solution $P(t)$ using VFFTRNN will approach its theoretical solution $P^*(t)$ when $t \to +\infty$.

In conclusion, the proof of Theorem 3 has been completed. That is, VFFTRNN exhibits global asymptotic convergence. □

In the next step, we will demonstrate the finite-time convergence of VFFTRNN under various activation functions (i.e., linear, bipolar-sigmoid, and power functions). See the following theorem for the specific convergence time.

Theorem 4. *For the time-varying Sylvester Equation (1), the coefficient matrices $U(t) \in \Re^{m\times m}$, $V(t) \in \Re^{n\times n}$ and $W(t) \in \Re^{m\times n}$ have been given. If Equation (1) satisfies the regularity condition (3) and the activation function sequence is monotonic and strictly increasing, then the VFFTRNN system (8) can converge to the theoretical solution $P^*(t)$ of Equation (1) in finite time from any arbitrary initial state. In addition, the specific convergence time of the neural system (7) in the cases of different activation functions is as follows:*

(1) When we choose the linear activation function (i.e., $\phi(x) = x$), the convergence time of VFFTRNN $t_c < t_L$.

(2) When we choose the bipolar-sigmoid activation function (i.e., $\phi(x) = (1 - exp(-\xi x))/(1 + exp(-\xi x))$, $\xi > 0$), the convergence time of VFFTRNN $t_c < t_B$.

(3) When we choose the power activation function (i.e., $\phi(x) = x^s$, $\lambda s < 1$ and s is an odd number), the convergence time of VFFTRNN $t_c < t_P$.

The above t_L, t_B and t_P are, respectively,

$$t_L = \sqrt[\alpha+1]{\frac{\alpha+1}{1-\lambda} ln(\chi_1^{1-\lambda} + 1)}$$

$$t_B = \begin{cases} \sqrt[\alpha+1]{\frac{\alpha+1}{\omega(1-\lambda)} ln(\chi_1^{1-\lambda} + 1)} & \text{if } \chi_1 \leq 1 \\ \sqrt[\alpha+1]{\frac{\alpha+1}{\omega(1-\lambda)} ln(2) + \frac{(\alpha+1)(\chi_1-1)}{\omega}} & \text{otherwise.} \end{cases}$$

$$t_P = \begin{cases} \sqrt[\alpha+1]{\frac{\alpha+1}{2(1-\lambda s)} \chi_1^{1-\lambda s}} & \text{if } \chi_1 \leq 1 \\ \sqrt[\alpha+1]{\frac{\alpha+1}{2(1-\lambda s)} + \frac{(\alpha+1)(\chi_1^{1-s}-1)}{2(1-s)}} & \text{otherwise.} \end{cases}$$

where χ_1 represents $max(|e_{ij}(0)|)$.

Proof. In the following, we will prove the convergence time of the VFFTRNN system under three different activation functions, one by one.

(1) Linear-type: For the linear-type case, the following equation can be obtained from (10).

$$\dot{e}_{ij}(t) = -(t^\alpha + \alpha)\Big(e_{ij}(t) + e_{ij}^\lambda(t)\Big) \tag{12}$$

In light of differential equation theory [34], we can obtain the solution of (12).

$$e_{ij}(t) = \Big((e_0^{1-\lambda} + 1) exp\big((\lambda - 1)(\alpha t + \frac{t^{\alpha+1}}{\alpha+1})\big) - 1\Big)^{\frac{1}{1-\lambda}} \tag{13}$$

where $e_0 = e_{ij}(0)$. Let $e_{ij}(t)$ in (13) be equal to zero, and we can obtain the following:

$$\alpha t + \frac{t^{\alpha+1}}{\alpha+1} = \frac{ln(e_0^{1-\lambda} + 1)}{1-\lambda} \geq \frac{t^{\alpha+1}}{\alpha+1}$$

then we can easily find $t \leq \sqrt[\alpha+1]{\frac{\alpha+1}{1-\lambda} ln(e_0^{1-\lambda} + 1)}$. In other words, for any i and j in the definitional domain, we have that as $t \to \sqrt[\alpha+1]{\frac{\alpha+1}{1-\lambda} ln(e_0^{1-\lambda} + 1)}$, the scalar value $e_{ij}(t) \to 0$. Note that by the definition of $e_{ij}(t)$ above, at this time, the error function

$E(t)$ in matrix form will also approach zero, which guarantees that the state matrix $P(t) \to P^*(t)$.

To sum up, we can obtain the convergence time when the VFFTRNN system uses the linear-type function as follows:

$$t_L = \sqrt[\alpha+1]{\frac{\alpha+1}{1-\lambda} ln(\chi_1^{1-\lambda} + 1)} \tag{14}$$

where $\chi_1 = max(|e_{ij}(0)|)$. This shows that in the case where the activation function is linear, the VFFTRNN method can achieve finite-time convergence in solving the Sylvester equation.

(2) Bipolar-Sigmoid-Type: In this case, we can easily see that the activation function is $\phi(x) = (1 - exp(-\xi x))/(1 + exp(-\xi x))$, where $\phi \geq 1$ and $x \in R$ are scalar factors. The following equation can be obtained from (10).

$$\begin{aligned} \dot{e}_{ij}(t) &= -(t^\alpha + \alpha)(R(t) + Q(t)) = \\ &- (t^\alpha + \alpha)\left(\frac{1 - \exp(-\xi e_{ij}(t))}{1 + \exp(-\xi e_{ij}(t))} + \frac{1 - \exp(-\xi e_{ij}^\lambda(t))}{1 + \exp(-\xi e_{ij}^\lambda(t))}\right) \end{aligned} \tag{15}$$

where $R\big(e_{ij}(t)\big) = \frac{1-\exp(-\xi e_{ij}(t))}{1+\exp(-\xi e_{ij}(t))}$, $Q\big(e_{ij}(t)\big) = \frac{1-\exp(-\xi e_{ij}^\lambda(t))}{1+\exp(-\xi e_{ij}^\lambda(t))}$. In light of the relevant theory of differential equations, we can rewrite (15) as follows:

$$\frac{dy}{dt} = -(t^\alpha + \alpha)\big(R(y) + Q(y)\big)$$

where $y = e_{ij}(t)$. According to the Lyapunov function of this system, we know that $e_{ij}(t)$ either increases or decreases monotonically as t increases. For simplicity, we will next discuss the case where $e_{ij}(0) = y_0 \geq 0$. In this case, we can derive the following:

$$\int_0^t (t^\alpha + \alpha)dt = -\int_{y_0}^0 \frac{1}{R(y) + Q(y)} dy \tag{16}$$

At this point, we consider the following two situations.

a. When $y_0 \leq 1$, noticing that $R(y)$ and $Q(y)$ are convex, solving (16) gives

$$\begin{aligned} \alpha t + \frac{t^{\alpha+1}}{\alpha+1} &= \int_0^{y_0} \frac{1}{R(y) + Q(y)} dy \\ &\leq \frac{1}{\omega}\int_0^{y_0} \frac{1}{y + y^\lambda} dy \\ &= \frac{1}{\omega(1-\lambda)} ln(y_0^{1-\lambda} + 1) \end{aligned} \tag{17}$$

where $\omega = R(1) = Q(1) = (1 - exp(-\xi))/(1 + exp(-\xi))$ is a constant, determined by the scalar factor ξ. Similar to linear-type, solving for (17), we obtain

$$t \leq \sqrt[\alpha+1]{\frac{\alpha+1}{\omega(1-\lambda)} ln(y_0^{1-\lambda} + 1)} \tag{18}$$

b. When $y_0 > 1$, based on Equations (16) and (17), Equation (19) be derived:

$$\begin{aligned} &\alpha t + \frac{t^{\alpha+1}}{\alpha+1} \\ &= \int_0^1 \frac{1}{R(y)+Q(y)}dy + \int_1^{y_0} \frac{1}{R(y)+Q(y)}dy \\ &\leq \frac{1}{\omega(1-\lambda)}ln2 + \int_1^{y_0} \frac{1}{R(y)+Q(y)}dy \end{aligned} \tag{19}$$

Notice that $R(y)$ and $Q(y)$ are monotonically increasing over the interval $y > 0$, so for $y > 1$, $R(y) > R(1) > \omega$ and $Q(y) > Q(1) > \omega$. Therefore, we have

$$\int_1^{y_0} \frac{1}{R(y)+Q(y)}dy \leq \int_1^{y_0} \frac{1}{\omega}dy = \frac{y_0-1}{\omega} \tag{20}$$

Substituting (20) into (19) and solving, we obtain

$$t \leq \sqrt[\alpha+1]{\frac{\alpha+1}{\omega(1-\lambda)}ln2 + \frac{(\alpha+1)(y_0-1)}{\omega}} \tag{21}$$

In light of the results (18) and (21) from the two situations, it can be concluded that when $t \to \sqrt[\alpha+1]{\frac{\alpha+1}{\omega(1-\lambda)}ln(y_0^{1-\lambda}+1)}$ or $t \to \sqrt[\alpha+1]{\frac{\alpha+1}{\omega(1-\lambda)}ln2 + \frac{(\alpha+1)(y_0-1)}{\omega}}$, the scalar value $e_{ij}(t) \to 0$ for any i and j in the defined domain. This means that the convergence of the state matrix $P(t) \to P^*(t)$ is guaranteed. Thus, the convergence time for the bipolar-sigmoid-type activation function can be expressed as

$$\begin{aligned} t &< \begin{cases} \sqrt[\alpha+1]{\frac{\alpha+1}{\omega(1-\lambda)}ln(\chi_1^{1-\lambda}+1)} & \text{if } \chi_1 \leq 1 \\ \sqrt[\alpha+1]{\frac{\alpha+1}{\omega(1-\lambda)}ln2 + \frac{(\alpha+1)(\chi_1-1)}{\omega}} & \text{if otherwise.} \end{cases} \\ &= t_B \end{aligned} \tag{22}$$

(3) Power-Type: For the power-type case, the activation function is defined as $\phi(x) = x^s$, where s is an odd number and $\lambda s < 1$. In this case, the following equation can be derived from (10).

$$\dot{e}_{ij}(t) = -(t^\alpha + \alpha)\left(e_{ij}^s(t) + e_{ij}^{\lambda s}(t)\right) \tag{23}$$

Similar to (16), we have

$$\frac{t^{\alpha+1}}{\alpha+1} + \alpha t = \int_0^{y_0} \frac{1}{y^s + y^{\lambda s}}dy \tag{24}$$

The discussion is divided into the following two situations:

a. When $y_0 \leq 1$, it can be found that $1 > \lambda s > s$, so $y^s > y^{\lambda s}$. In light of (24), we can obtain

$$\frac{t^{\alpha+1}}{\alpha+1} + \alpha t \leq \int_0^{y_0} \frac{1}{2y^{\lambda s}}dy = \frac{1}{2(1-\lambda s)}y_0^{1-\lambda s}$$

Then, the result (25) can be obtained:

$$t < \sqrt[\alpha+1]{\frac{\alpha+1}{2(1-\lambda s)}y_0^{1-\lambda s}} \tag{25}$$

b. When $y_0 > 1$, we know that for all $y > 1$, we have $y^p < y^{\lambda s}$. So similarly,

$$\begin{aligned}\frac{t^{\alpha+1}}{\alpha+1} + \alpha t &\leq \int_0^1 \frac{1}{2y^{\lambda s}} dy + \int_1^{y_0} \frac{1}{2y^s} dy \\ &= \frac{1}{2(1-\lambda s)} + \frac{y_0^{1-s} - 1}{2(1-s)}\end{aligned}$$

Then, the following result (26) can be obtained:

$$t < \sqrt[\alpha+1]{\frac{\alpha+1}{2(1-\lambda s)} + \frac{(\alpha+1)(y_0^{1-s}-1)}{2(1-s)}} \tag{26}$$

In conclusion, when $t \to \sqrt[\alpha+1]{\frac{\alpha+1}{2(1-\lambda s)} y_0^{1-\lambda s}}$ or $t \to \sqrt[\alpha+1]{\frac{\alpha+1}{2(1-\lambda s)} + \frac{(\alpha+1)(y_0^{1-s}-1)}{2(1-s)}}$, the scalar value $e_{(ij}(t) \to 0$ for any i and j on the definitional domain. This means that the convergence of state matrix $P(t) \to P^*(t)$ is guaranteed. The convergence time for the power-type activation function can be represented as

$$\begin{aligned} t &< \begin{cases} \sqrt[\alpha+1]{\frac{\alpha+1}{2(1-\lambda s)} \chi_1^{1-\lambda s}} & \text{if } \chi_1 \leq 1 \\ \sqrt[\alpha+1]{\frac{\alpha+1}{2(1-\lambda s)} + \frac{(\alpha+1)(\chi_1^{1-s}-1)}{2(1-s)}} & \text{if otherwise.} \end{cases} \\ &= t_P \end{aligned} \tag{27}$$

In light of the above analysis, it can be concluded that VFFTRNN achieves finite-time convergence in solving the Sylvester Equation (1), regardless of the choice of activation function. Therefore, the proof of Theorem 4 on finite-time convergence is completed. □

Remark 1. *Regardless of the chosen activation function, the convergence time of VFFTRNN is determined by the initial error $E(0)$ (i.e., $E(0) = U(0)P(0) + P(0)V(0) - W(0)$) and the design factors α and λ.*

Next, we will analyze the influence of the factors α and λ on the convergence time for different activation functions.

(1) Linear-Type:

a. For α, in light of (14), t_L can be rewritten as

$$t_L = (\chi_2 \alpha_1)^{\frac{1}{\alpha_1}} \tag{28}$$

where $\chi_2 = \frac{\ln(\chi_1^{1-\lambda}+1)}{1-\lambda} > 0$ and $\alpha_1 = \alpha + 1 > 1$. Then, let us take the partial derivative with respect to α_1.

$$\frac{\partial t_L}{\partial \alpha_1} = (\chi_2 \alpha_1)^{\frac{1}{\alpha_1}} \frac{1 - \ln(\chi_2 \alpha_1)}{\alpha_1^2} \tag{29}$$

We know that if α_1 is large enough, $1 - ln(\chi_2 \alpha_1)$ will become less than zero. Therefore, for $\frac{\partial t_L}{\partial \alpha_1}$, there are two possible situations:

$$\begin{cases} \frac{\partial t_L}{\partial \alpha_1} < 0 & \forall \alpha_1 > 1 \\ \frac{\partial t_L}{\partial \alpha_1} > 0 & \exists \alpha_1 > 1 \end{cases}$$

In the first situation, t_L decreases as α_1 increases. In other words, as α increases, t_L decreases simultaneously. Similarly, in the second situation, t_L will first increase

and then decrease as α increases. This means that when α reaches a specific value, determined by the initial error $E(0)$, t_L will attain its maximum.

b. For λ, in light of (14), it can easily be seen that we only need to analyze $\frac{\ln(\chi_1^{1-\lambda}+1)}{1-\lambda}$. Next, take the partial derivative of this expression:

$$\frac{\partial \eta}{\partial \lambda_1} = \frac{\lambda_1 \chi_1^{\lambda_1} \ln \chi_1 - (\chi_1^{\lambda_1}+1)\ln(\chi_1^{\lambda_1}+1)}{\lambda_1^2} \tag{30}$$

where $\eta = \frac{\ln(\chi_1^{1-\lambda}+1)}{1-\lambda}$ and $\lambda_1 = 1-\lambda \in (0,1)$. Since the denominator of (30) is always positive, we only need to analyze the numerator:

$$\lambda_1 \chi_1^{\lambda_1} \ln \chi_1 - (\chi_1^{\lambda_1}+1)\ln(\chi_1^{\lambda_1}+1) \tag{31}$$

If $\chi_1 \leq 1$, then $\lambda_1 \chi_1^{\lambda_1} \ln \chi_1 \leq 0$, and $\lambda_1 \chi_1^{\lambda_1} \ln \chi_1 - (\chi_1^{\lambda_1}+1)\ln(\chi_1^{\lambda_1}+1) < 0$. Therefore, the value of (30) is always less than zero. If $\chi_1 > 1$, let $\chi_1^{\lambda_1} = \lambda_2 > 1$, then $\lambda_1 = \frac{\ln \lambda_2}{\ln \chi_1}$. The numerator can be rewritten as follows:

$$\begin{aligned} &\lambda_1 \chi_1^{\lambda_1} \ln \chi_1 - (\chi_1^{\lambda_1}+1)\ln(\chi_1^{\lambda_1}+1) \\ &= \lambda_2 \ln \lambda_2 - (\lambda_2+1)\ln(\lambda_2+1) \end{aligned} \tag{32}$$

Note that the function $f(x) = x \ln x$ is monotonically increasing when $x > 1$. Therefore, the value of Equation (32) is always less than zero. In other words, the value of Equation (30) is always less than zero.

Based on the above discussion, we can conclude that for any λ_1, it is always true that $\frac{\partial \eta}{\partial \lambda_1} < 0$. Considering the relationship between λ_1 and λ, we can conclude that when λ is larger, t_L is smaller.

(2) Bipolar-Sigmoid-Type:

a. For α, in light of (22), t_B can be rewritten as

$$t_B = (\chi_3 \alpha_1)^{\frac{1}{\alpha_1}} \tag{33}$$

where $\chi_3 = \begin{cases} \frac{\ln(\chi_1^{1-\lambda}+1)}{\omega(1-\lambda)} > 0 & , \chi_1 \leq 1 \\ \frac{\ln 2}{\omega(1-\lambda)} + \frac{\chi_1 - 1}{\omega} > 0, & \chi_1 > 1 \end{cases}$. Similar to the analysis of the linear-type above, there are two situations:

$$\begin{cases} \frac{\partial t_B}{\partial \alpha_1} < 0 & \forall \alpha_1 > 1 \\ \frac{\partial t_B}{\partial \alpha_1} > 0 & \exists \alpha_1 > 1 \end{cases}$$

For the first situation, when α increases, t_B decreases. For the second situation, there exists a specific value where t_B is maximized.

b. For λ, by analyzing the structure of Equation (22), we focus on $\frac{\ln(\chi_1^{1-\lambda}+1)}{1-\lambda}$ when $\chi_1 \leq 1$ and $\frac{\ln 2}{1-\lambda}$ when $\chi_1 > 1$. As λ increases, the value of $\frac{\ln 2}{1-\lambda}$ increases, Based on the derivation in Section 1), it is evident that for any $\lambda \in (0,1)$, as λ increases, t_B also increases.

In addition, consider the influence of the factor ξ on the convergence time t_B. We know that $\omega = \frac{1-e^{-\xi}}{1+e^{-\xi}}$, and as ξ increases, ω also increases. Furthermore, according to Equation (22), the larger ω is, the smaller t_B becomes.

(3) Power-Type:

a. For α, let $\chi_4 = \begin{cases} \frac{\chi_1^{1-\lambda s}}{2(1-\lambda s)} > 0 & ,\chi_1 \leq 1 \\ \frac{1}{2(1-\lambda s)} + \frac{1-\chi_1^{s-1}}{2(s-1)} > 0, & \chi_1 > 1 \end{cases}$, then we have

$$t_P = (\chi_4 \alpha_1)^{\frac{1}{\alpha_1}} \tag{34}$$

Similar to the analysis of the linear-type case above, there are two possible situations:

$$\begin{cases} \frac{\partial t_P}{\partial \alpha_1} < 0 & \forall \alpha_1 > 1 \\ \frac{\partial t_P}{\partial \alpha_1} > 0 & \exists \alpha_1 > 1 \end{cases}$$

For the first situation, when α increases, t_P decreases. For the second situation, there exists a certain value where t_P is maximized.

b. For λ, according to (27), we can conclude that the larger λ is, the larger t_P becomes, especially when $\chi_1 > 0$. In addition, when $\chi_1 \leq 1$, we only need to consider $\frac{\chi^{1-\lambda s}}{1-\lambda s}$. Let $\lambda_3 = 1 - \lambda s \in (0,1)$, then

$$\frac{\partial}{\partial \lambda_3}\left(\frac{\chi_1^{\lambda_3}}{\lambda_3}\right) = \frac{\chi_1^{\lambda_3}(\lambda_3 \ln \chi_1 - 1)}{\lambda_3^2} < 0 \tag{35}$$

So in conclusion, it is clear that for $\forall \lambda \in (0,1)$, as λ increases, t_P increases.

In light of the above derivation, the following conclusions can be drawn:

(1) For α, its relationship to the convergence time can always be divided into two cases. In the first case, as α increases, the convergence time decreases. In the second case, the convergence time first increases and then decreases. It is worth noting that even when $\alpha = 0$, the VFFTRNN system will still converge in finite time.

(2) For λ, the convergence time always increases with the increase in λ, regardless of the chosen activation function. Clearly, the factor λ accelerates the convergence process of the proposed VFFTRNN. If $\lambda = 1$, the proposed VFFTRNN becomes VP-CDNN. Furthermore, if the factor $\alpha = 0$, the proposed VFFTRNN reduces to ZNN.

3.2. Robustness Analysis

As we all know, in practical applications, circumstances are rarely perfect. Due to various reasons, such as errors in digital implementations or higher-order residues from circuit elements (e.g., diodes) in hardware implementations, many factors continuously interfere with the VFFTRNN model. Among these, three common negative factors include coefficient matrix disturbances, differentiation errors, and model implementation errors. Next, we will discuss the robustness of the VFFTRNN model in the presence of these negative factors.

(1) Coefficient Matrix Perturbation:

Theorem 5. *If uncharted, smooth coefficient matrix perturbations* $\Delta U(t) \in \Re^{m\times m}$, $\Delta V(t) \in \Re^{n\times n}$ *and* $\Delta W(t) \in \Re^{m\times n}$ *exist in the VFFTRNN model, and if they satisfy the following conditions:*

$$||N^{-1}(t)||_F \leq \upsilon_1, ||\hat{N}^{-1}(t)||_F \leq \upsilon_2$$
$$||W(t)||_F \leq \iota_1, ||\Delta W(t)||_F \leq \iota_2, \quad \forall t \in [0, +\infty)$$

where the mass matrices are defined as $N^{-1}(t) := \left(I_n \otimes U(t) - V(t)^T \otimes I_m\right)^{-1}$, $\hat{N}^{-1}(t) := \left(I_n \otimes (U(t) + \Delta U(t)) - (V(t) + \Delta V(t))^T \otimes I_m\right)^{-1}$. *Therefore, the calculation error* $||\hat{\tilde{P}}(t)||_F = ||\hat{P}(t) - P^*(t)||_F$ *is bounded.*

Proof. The following proof uses the linear activation function as an example. First, we define a variable $\tilde{P}(t) = P(t) - P^*(t)$ which represents the error in the state matrix $P(t) \to P^*(t)$. Then, its Frobenius norm can be written as

$$||\tilde{P}(t)||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} \tilde{p}_{ij}^2(t)} = ||\tilde{p}(t)||_2$$

where $\tilde{p}(t) := vec\left(\tilde{P}(t)\right) \in \Re^{mn\times 1}$. Furthermore, in light of (1), we can obtain an expression of W:

$$W(t) = -U(t)P^*(t) + P^*(t)V(t) \tag{36}$$

Substitute (36) into (4):

$$E(t) = U(t)\tilde{P}(t) - \tilde{P}(t)V(t)$$

In light of Theorem 1, we can acquire its vector form:

$$e(t) = N(t)\tilde{p}(t) = N(t)\big(p(t) - p^*(t)\big) \tag{37}$$

where $e(t) := vec(E(t))$ and $p^*(t) := vec(P^*(t))$. From Theorem 2 and inequality (3), it is clear that the matrix $N(t)$ is invertible. Therefore, we conclude that

$$\begin{aligned} ||\tilde{P}(t)||_F = ||\tilde{p}(t)||_2 &\le ||N^{-1}(t)||_F||e(t)||_2 \\ &\le v_1\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} e_{ij}^2(t)} \end{aligned} \tag{38}$$

Substituting (13) into (38), we can obtain

$$\begin{aligned} &||\tilde{P}(t)||_F \\ &\le v_1\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} \left((e_{ij}^{1-\lambda}(0)+1)e^{(\lambda-1)(\alpha t + \frac{t^{\alpha+1}}{\alpha+1})} - 1\right)^{\frac{2}{1-\lambda}}} \\ &\le v_1\sqrt{mn}\left((\chi_1^{1-\lambda}+1)e^{(\lambda-1)(\alpha t + \frac{t^{\alpha+1}}{\alpha+1})} - 1\right)^{\frac{1}{1-\lambda}} \\ &\le v_1\tau_1 e^{-(\alpha t + \frac{t^{\alpha+1}}{\alpha+1})} \end{aligned} \tag{39}$$

where $\tau_1 = \sqrt{mn}(\chi_1^{1-\lambda}+1)^{\frac{1}{1-\lambda}} > 0$. The results demonstrate that the VFFTRNN model exhibits super-exponential convergence in solving the Sylvester equation when employing a linear-type activation function.

Next, we consider the case where perturbations in the coefficient matrices exist. Based on Equation (8), it is straightforward to derive the implicit dynamic equation of the VFFTRNN system under the influence of these perturbations.

$$\begin{aligned} &\dot{\hat{U}}(t)\hat{P}(t) + \hat{U}(t)\dot{\hat{P}}(t) - \dot{\hat{P}}(t)\hat{V}(t) - \hat{P}(t)\dot{\hat{V}}(t) + \dot{\hat{W}}(t) \\ &= -(t^\alpha + \alpha)(\Phi(\hat{U}(t)\hat{P}(t) - \hat{P}(t)\hat{V}(t) + \hat{W}(t)) \\ &\quad + \Phi(\hat{U}(t)\hat{P}(t) - \hat{P}(t)\hat{V}(t) + \hat{W}(t))^\lambda) \end{aligned} \tag{40}$$

where $\hat{U}(t) = U(t) + \Delta U(t)$, $\hat{V}(t) = V(t) + \Delta V(t)$ and $\hat{W}(t) = W(t) + \Delta W(t)$. $\hat{P}(t)$ denotes the solution to the VFFTRNN system with perturbation.

Furthermore, in light of (36), its vector form can be written as

$$w(t) = -N(t)p^*(t)$$

where $w(t) := vec(W(t))$. Then, we can acquire

$$p^*(t) = -N^{-1}(t)w(t)$$

Therefore, we can derive the inequality for the theoretical solution $P^*(t)$.

$$\begin{aligned} ||P^*(t)||_F = ||p^*(t)||_2 &= ||N^{-1}(t)w(t)||_2 \\ &\leq ||N^{-1}(t)||_F||w(t)||_F \leq \upsilon_1\iota_1 \end{aligned} \tag{41}$$

Let us assume that the theoretical solution of the VFFTRNN system with perturbation is $\hat{P}^*(t)$. Similar to (40), we can easily deduce that

$$\begin{aligned} ||\hat{P}^*(t)||_F &\leq ||\hat{N}^{-1}(t)||_F||\hat{W}(t)||_F \\ &\leq \upsilon_2(||W(t)||_F + ||\Delta W(t)||_F) \leq \upsilon_2(\iota_1 + \iota_2) \end{aligned} \tag{42}$$

In addition, similar to (39), we can obtain

$$||\hat{P}(t) - \hat{P}^*(t)||_F \leq \upsilon_2\tau_2 e^{-(\alpha t + \frac{t^{\alpha+1}}{\alpha+1})} \tag{43}$$

where $\tau_2 = \sqrt{mn}(\chi_5^{1-\lambda} + 1)^{\frac{1}{1-\lambda}}$ and $\chi_5 = \max|(\hat{U}(0)\hat{P}(0) - \hat{P}(0)\hat{V}(0) + \hat{W}(0))_{ij}|$. Then, based on (41), (42) and (43), we can derive an upper bound for the computation error $||\hat{\tilde{P}}(t)||_F$.

$$\begin{aligned} ||\hat{\tilde{P}}(t)||_F &= ||\hat{P}(t) - P^*(t)||_F \\ &\leq ||\hat{P}(t) - \hat{P}^*(t)||_F + ||\hat{P}^*(t) - P^*(t)||_F \\ &\leq \upsilon_2\tau_2 e^{-(\alpha t + \frac{t^{\alpha+1}}{\alpha+1})} + ||\hat{P}^*(t)||_F + ||P^*(t)||_F \\ &\leq \upsilon_2\tau_2 e^{-(\alpha t + \frac{t^{\alpha+1}}{\alpha+1})} + \upsilon_1\iota_1 + \upsilon_2(\iota_1 + \iota_2) \end{aligned} \tag{44}$$

In light of the above analysis, we conclude that when using a linear-type activation function, the computation error $||\hat{\tilde{P}}(t)||_F$ of the VFFTRNN system with perturbation is bounded. Additionally, it is worth mentioning that the solution process exhibits a super-exponential convergence rate.

Similarly, for sigmoid or power activation functions, it is not difficult to prove that the computation error of the VFFTRNN system with perturbation is bounded. Due to the limited space of this paper, the analysis for these two activation functions will not be presented here.

Thus, the proof of the robustness of Theorem 5 regarding matrix perturbation is complete. □

(2) Differentiation and Model Implementation Errors:

In the process of hardware implementation, dynamics implementation errors and differential errors related to $U(t)$, $V(t)$ and $W(t)$ are inevitable. These errors are collectively referred to as model implementation errors [35]. In this section, we analyze the robustness of the VFFTRNN system in the presence of these errors. Let us assume that the differential errors for the time derivatives of the matrices $\dot{U}(t)$ and $\dot{V}(t)$ are $\Lambda(t) \in \Re^{m\times m}$ and $\Omega(t) \in \Re^{n\times n}$, respectively, and the model implementation

error is denoted as $\Delta(t) \in R^{m\times n}$. Then, based on Equation (8), the implicit dynamic equation for the VFFTRNN system with these errors can be written as

$$\begin{aligned} &U(t)\dot{P}(t) + (\dot{U}(t) + \Lambda(t))P(t) - \dot{P}(t)V(t) \\ &- P(t)(\dot{V}(t) + \Omega(t)) + \dot{W}(t) - \Delta(t) \\ &= -(t^{\alpha} + \alpha)\Big(\Phi(U(t)P(t) - P(t)V(t) + W(t)) \\ &+ \Phi\big((U(t)P(t) - P(t)V(t) + W(t))^{\lambda}\big)\Big) \end{aligned} \tag{45}$$

Theorem 6. *If there exist unknown smooth differentiation errors $\Lambda(t)$, $\Omega(t)$ and a model implementation error $\Delta(t)$ in the VFFTRNN model, which satisfy the following conditions:*

$$\begin{aligned} ||\Lambda(t)||_F \leq \varepsilon_1 \quad ||\Omega(t)||_F \leq \varepsilon_2 \quad ||\Delta(t)||_F \leq \varepsilon_3 \\ ||N^{-1}(t)||_F \leq \upsilon_1 \quad ||W(t)||_F \leq \iota_1, \quad \forall t \in [0, +\infty) \end{aligned}$$

then its computation error $||P(t) - P^*(t)||_F$ is bounded.

Proof. Let us rewrite (45) from matrix form to vector form for easier analysis.

$$\begin{aligned} N(t)\dot{p}(t) = &-(\dot{N}(t) + \Psi(t))p(t) - \dot{w}(t) + \delta(t) \\ &-(t^{\alpha} + \alpha)\Big(\Phi(N(t)p(t) + w(t)) \\ &+ \Phi\big((N(t)p(t) + w(t))^{\lambda}\big)\Big) \end{aligned} \tag{46}$$

where $\Psi(t) := I_n \otimes \Lambda(t) - \Omega^T(t) \otimes I_m$ and $\delta(t) := vec(\Delta(t))$. In addition, we can easily observe that $e(t) = N(t)p(t) + w(t)$, and its time derivative is given by $\dot{e}(t) = \dot{N}(t)p(t) + N(t)\dot{p}(t) + \dot{w}(t)$. Therefore, Equation (46) can be rewritten as

$$\dot{e}(t) = -(t^{\alpha} + \alpha)\Big(\Phi(e(t)) + \Phi(e^{\lambda}(t))\Big) + \delta(t) - \Psi(t)p(t) \tag{47}$$

In light of (37), we can acquire

$$x(t) = N^{-1}(t)e(t) + p^*(t) \tag{48}$$

Substituting (47) into (48), we have

$$\begin{aligned} \dot{e}(t) = &-(t^{\alpha} + \alpha)\Big(\Phi(e(t)) + \Phi(e^{\lambda}(t))\Big) \\ &+ \mathrm{H}(t)e(t) + \mathrm{K}(t) \end{aligned} \tag{49}$$

where $\mathrm{H}(t) = -\Psi(t)N^{-1}(t) \in \Re^{mn\times mn}$, as well as $\mathrm{K}(t) = \delta(t) - \Psi(t)p^*(t) \in \Re^{mn\times 1}$. Next, we construct a Lyapunov function candidate in the form $v := \frac{1}{2}e^T(t)e(t)$, and its time derivative can be computed as follows:

$$\begin{aligned} \dot{v} = e^T(t)\dot{e}(t) = &-(t^{\alpha} + \alpha)e^T(t)\Big(\Phi(e(t)) + \Phi(e^{\lambda}(t))\Big) \\ &+ e^T(t)\mathrm{H}(t)e(t) + e^T(t)\mathrm{K}(t) \end{aligned} \tag{50}$$

By observing the composition of (50) and (41), we can derive the following:

$$\begin{aligned}
&e^T(t)\mathrm{H}(t)e(t)\\
&\le ||e^T(t)||_2||\mathrm{H}(t)||_F||e(t)||_2 \le e^T(t)e(t)||\mathrm{H}(t)||_F\\
&\le ||\Psi(t)||_F||M^{-1}(t)||_F e^T(t)e(t)\\
&\le v_1(||I_n \otimes \Lambda(t)||_F + ||\Omega^T(t) \otimes I_m||_F)e^T(t)e(t)\\
&\le v_1(\varepsilon_1\sqrt{n} + \varepsilon_2\sqrt{m})\sum_{i=1}^{m}\sum_{j=1}^{n} e_{ij}^2(t)
\end{aligned}$$

and

$$\begin{aligned}
&e^T(t)\mathrm{K}(t) \le ||e^T(t)||_2||\mathrm{K}(t)||_F\\
&\le ||e^T(t)||_2(||\Delta(t)||_F + ||\Psi(t)||_F||P^*(t)||_F)\\
&\le \left(\varepsilon_3 + (\varepsilon_1\sqrt{n} + \varepsilon_2\sqrt{m})v_1\iota_1\right)\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} e_{ij}^2(t)}\\
&\le \left(\varepsilon_3 + (\varepsilon_1\sqrt{n} + \varepsilon_2\sqrt{m})v_1\iota_1\right)\sum_{i=1}^{m}\sum_{j=1}^{n} |e_{ij}(t)|
\end{aligned}$$

By substituting the above two equations into (50), we can obtain the following:

$$\begin{aligned}
\dot{v} \le -\sum_{i=1}^{m}\sum_{j=1}^{n} |e_{ij}(t)|\Big((t^\alpha + \alpha)\left(\phi(|e_{ij}(t)|) + \phi(|e_{ij}^\lambda(t)|)\right)\\
- \zeta_1 v_1|e_{ij}(t)| - \zeta_2\Big) = \Upsilon(t)
\end{aligned} \tag{51}$$

where $\zeta_1 = \varepsilon_1\sqrt{n} + \varepsilon_2\sqrt{m}$ and $\zeta_2 = \varepsilon_3 + \zeta_1 v_1 \iota_1$.
We know that $\phi(|e_{ij}(t)|) \ge |e_{ij}(t)|$ for any t, and that $t^\alpha + \alpha$ increases as t increases. Therefore, there always exists a value t_0, such that for all $t > t_0$, $(t^\alpha + \alpha)\left(\phi(|e_{ij}(t)|) + \phi(|e_{ij}^\lambda(t)|)\right) - \zeta_1 v_1|e_{ij}(t)| - \zeta_2\Big) > 0$. Thus, for Equation (51), we need to consider the following two cases:

$$\begin{cases}
\alpha(\phi(|e_{ij}(0)|) + \phi(|e_{ij}^\lambda(0)|)) - \zeta_1 v_1|e_{ij}(0)| - \zeta_2 \ge 0\\
\qquad\qquad \forall i \in \{1, ..., m\}, j \in \{1, ..., n\}\\
\alpha(\phi(|e_{ij}(0)|) + \phi(|e_{ij}^\lambda(0)|)) - \zeta_1 v_1|e_{ij}(0)| - \zeta_2 < 0\\
\qquad\qquad \exists i \in \{1, ..., m\}, j \in \{1, ..., n\}
\end{cases} \tag{52}$$

(1) For the first situation, it is easy to see that $\Upsilon(t) < 0$, i.e., $\dot{v} \le 0$. Therefore, the error $e(t)$ will decrease monotonically as time t increases, which indicates that $P(t)$ will eventually converge to $P^*(t)$ as time increases, i.e., $||P(t) - P^*(t)||_F$ will approach 0.

(2) For the second situation, due to the uncertainty of the sign of $\Upsilon(0)$, we need to further subdivide it into two cases. If $\Upsilon(0) \le 0$, then the analysis follows the same reasoning as in case 1. Moreover, if $\Upsilon(0) > 0$, there exists a time t_0 such that $\Upsilon(t) \le 0$ for all $t > t_0$. This indicates that the error $e(t)$ will initially increase and then decrease. Based on the results in [36], we can obtain the upper bound of $||\widetilde{P}(t)||_F = ||P(t) - P^*(t)||_F$ when using a linear activation function:

$$||\widetilde{P}(t)||_F \le \frac{v_1\zeta_2(mn + \sqrt{mn})}{2(\rho(t^\alpha + \alpha) - v_1\zeta_1)} \tag{53}$$

Since $\zeta_2 > 0$ and $(mn + \sqrt{mn}) > 0$, it is guaranteed that (52) holds true. The design factor ρ should be set greater than $\frac{\upsilon_1\zeta_2}{\alpha}$ in order for the denominator of the fraction to be positive. Therefore, as $t \to +\infty$, the computation error $||\widetilde{P}(t)||_F$ will approach 0.

Summing up the above, as time increases, the computation error $||\widetilde{P}(t)||_F$ tends to zero in both cases. The difference is that in the first situation, the computation error $||\widetilde{P}(t)||_F$ continuously decreases, whereas in the second situation, the computation error $||\widetilde{P}(t)||_F$ first increases and then decreases. Additionally, there exists an upper bound given by $\frac{\upsilon_1\zeta_2(mn+\sqrt{mn})}{2(\rho(t^\alpha+\alpha)-\upsilon_1\zeta_1)}$. □

4. Illustrative Example

This section presents simulations using MATLAB 2021b for solving the Sylvester equation with the proposed VFFTRNN. All experiments are conducted on a personal computer equipped with a 64-bit operating system, 16.00 GB RAM, and an Intel(R) Core(TM) i5-12600KF Processor (20M Cache, up to 4.90 GHz).

We will verify the performance of the proposed VFFTRNN and compare it with ZNN by providing a specific example. The coefficients of the time-varying Sylvester Equation (1) are set as follows:

$$U(t) = \begin{bmatrix} \frac{\sin t}{10} & \frac{\cos t}{10} \\ \frac{-\cos t}{10} & \frac{\sin t}{10} \end{bmatrix}, V(t) = \begin{bmatrix} \frac{\sin t}{100} & 0 \\ 0 & \frac{\cos t}{50} \end{bmatrix}$$
$$W(t) = \begin{bmatrix} \frac{\sin^2 t}{10} - 1 & -\frac{\cos^2 t}{5} \\ \frac{\sin t \cos t}{10} & \frac{\sin t \cos t}{5} - 1 \end{bmatrix}.$$

Then, its theoretical solution $P^*(t)$ can be easily obtained:

$$P^*(t) = \begin{bmatrix} 10\sin t & -10\cos t \\ 10\cos t & 10\sin t \end{bmatrix}.$$

In addition, we write $P(t)$ solved by the VFFTRNN in the following form:

$$P(t) = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$$

where p_{ij} denotes the ith row and $j-$ column element of $P(t)$. For the sake of discussion, let us set the initial state P_0 as follows:

$$P_0 = \begin{bmatrix} p_a & p_b \\ p_c & p_d \end{bmatrix}$$

where p_a, p_b, p_c and p_d are numbers generated by MATLAB which lie within the range $[-25, 25]$.

Generally, the system is considered to have converged in solving the Sylvester Equation (1) when the computation error $||P(t) - P^*(t)||_F$ is less than 0.1% of the range (i.e., $||P(t) - P^*(t)|| \leq 50 \times 0.1\% = 0.05$). In other words, we only need to measure the time when the calculation error $||P(t) - P^*(t)||_F$ falls below 0.05, which allows us to determine the convergence time.

4.1. Convergence Discussion

In this section, we will discuss the convergence performance of the VFFTRNN and ZNN models in solving the Sylvester equation under various conditions.

First, we observe the effects of different factors on the convergence performance of ZNN and VFFTRNN with the same activation function. Table 1 presents a comparison of the convergence times of ZNN and VFFTRNN under various design factors.

Figures 2 and 3 show detailed simulation results of both models under different conditions for $\alpha = \mu = 0.1$, $\alpha = \mu = 1$, and $\alpha = \mu = 5$. As seen, when μ is sufficiently small (e.g., $\mu = 0.01$), ZNN fails to converge. However, with $\mu = 0.1$, the convergence time of ZNN is 67.55 s. As the design factor μ increases, the convergence performance of ZNN improves gradually. From Figure 2, we observe that when $\mu = 1$, the computation error $||P(t) - P^*(t)||_F$ converges to zero in 6.98 s, nearly ten times faster than with $\mu = 0.1$. When $\mu = 5$, the convergence time further decreases to just 1.37 s.

Next, we examine the convergence time of the computation error using VFFTRNN. With $\alpha = 0.01$ or $\alpha = 0.1$, the computation error $||P(t) - P^*(t)||_F$ converges to zero in just 2.39 or 2.35 s, which is significantly faster than ZNN.

Furthermore, the comparison results in Table 1 and Figures 2 and 3 clearly show that the convergence performance of VFFTRNN is significantly better than that of ZNN. It is worth noting that these simulation results use a linear-type activation function.

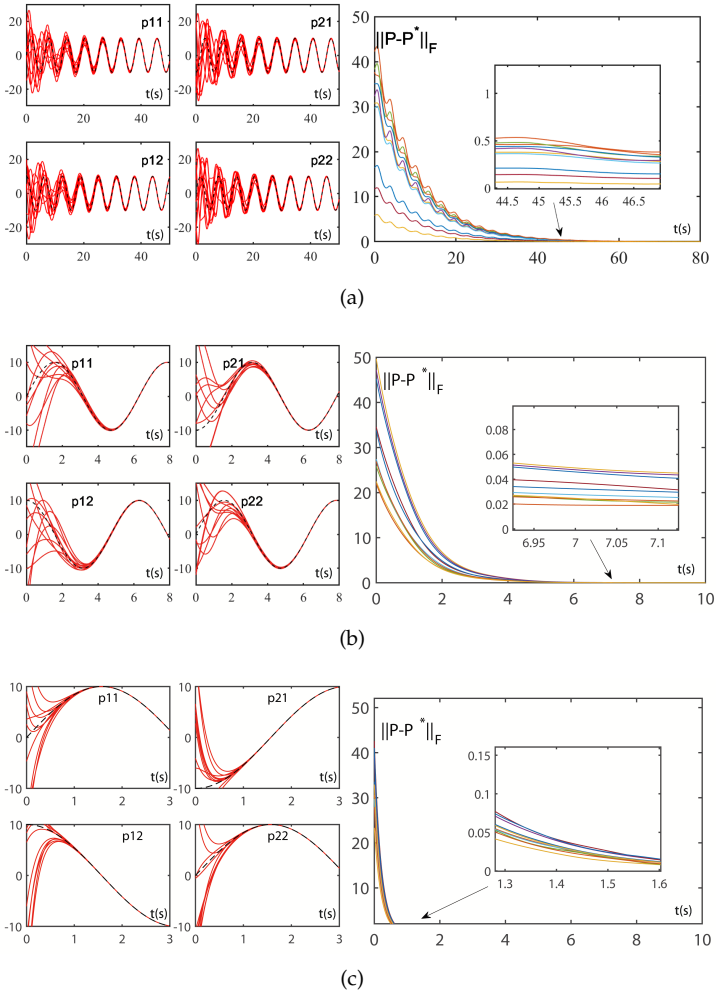

Figure 2. Solving Sylvester Equation (1) with given coefficients online with ZNN and its computational error $||P(t) - P^*(t)||_F$. Several red solid curves represent different initial states while the unique black dotted curve represents the theoretical solution. (**a**) Solution to ZNN with $\mu = 0.1$ and error $||P(t) - P^*(t)||_F$ of ZNN with $\mu = 0.1$. (**b**) Solution to ZNN with $\mu = 1$ and error $||P(t) - P^*(t)||_F$ of ZNN with $\mu = 1$. (**c**) Solution to ZNN with $\mu = 5$ and error $||P(t) - P^*(t)||_F$ of ZNN with $\mu = 5$.

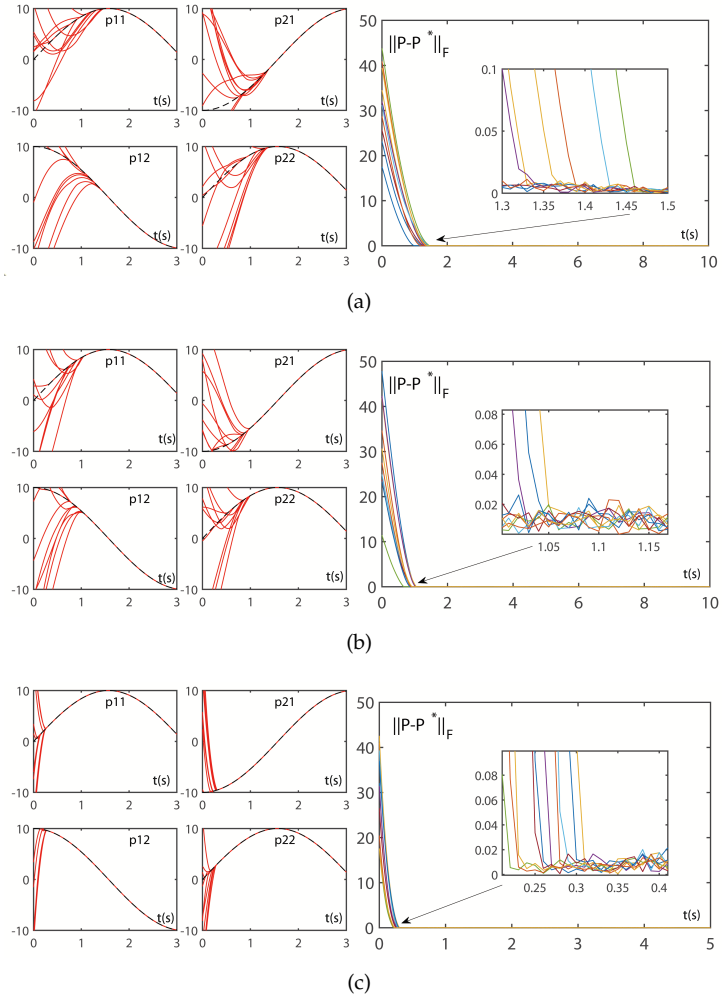

Figure 3. Solving Sylvester Equation (1) with given coefficients online with VFFTRNN and its computational error $||P(t) - P^*(t)||_F$ ($\lambda = \frac{5}{7}$). Several red solid curves represent different initial states while the unique black dotted curve represents the theoretical solution. (**a**) Solution to VFFTRNN with $\alpha = 0.1$ and error $||P(t) - P^*(t)||_F$ of VFFTRNN with $\alpha = 0.1$. (**b**) Solution to VFFTRNN with $\alpha = 1$ and error $||P(t) - P^*(t)||_F$ of VFFTRNN with $\alpha = 1$. (**c**) Solution to VFFTRNN with $\alpha = 5$ and error $||P(t) - P^*(t)||_F$ of VFFTRNN with $\alpha = 5$.

Table 1. Comparisons of convergence time of ZNN and VFFTRNN when solving Sylvester equation.

Parameter ($\alpha = \mu$)	$T_{ZNN(s)}$	$T_{VFFTRNN(s)}$
0.01	Cannot converge	2.39
0.1	67.55	2.25
0.2	34.20	1.99
0.5	13.88	1.69
1	6.98	1.41
2	3.48	1.02
5	1.37	0.49

Second, we observed the effects of different activation functions on the convergence performance of ZNN and VFFTRNN when the same factors were set.

Figures 4 and 5 show the computation error $||P(t) - P^*(t)||_F$ curves for ZNN and VFFTRNN under three activation functions (i.e., linear, bipolar-sigmoid, and power-type).

When using the linear activation function, it can be observed that ZNN takes 6.98 s to converge to zero, whereas VFFTRNN converges in just 1.05 s. Similarly, when the

bipolar-sigmoid or power activation functions are used, the convergence time of VFFTRNN is significantly shorter than that of ZNN.

This demonstrates the superior performance of the VFFTRNN system compared to ZNN and also verifies its finite-time convergence property for solving the Sylvester equation under different activation functions. It is worth noting that the factors were set to $\mu = \alpha = 1$ and $\lambda = \frac{1}{5}$ in the above simulation results.

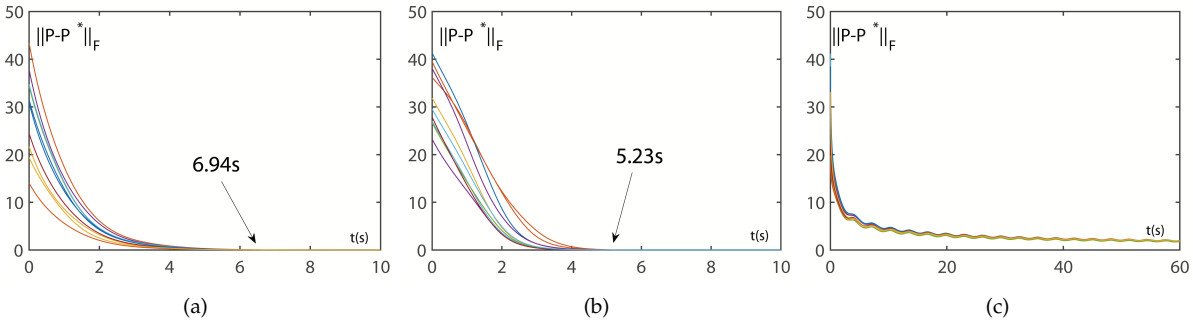

Figure 4. The curves in the figure represent the convergence errors under different initial conditions. When ZNN uses different activation functions without perturbation to solve Sylvester Equation (1), its convergence error $||P(t) - P^*(t)||_F$ is as shown above ($\mu = 1$). (**a**) Linear. (**b**) Sigmoid. (**c**) Power ($s = 3$).

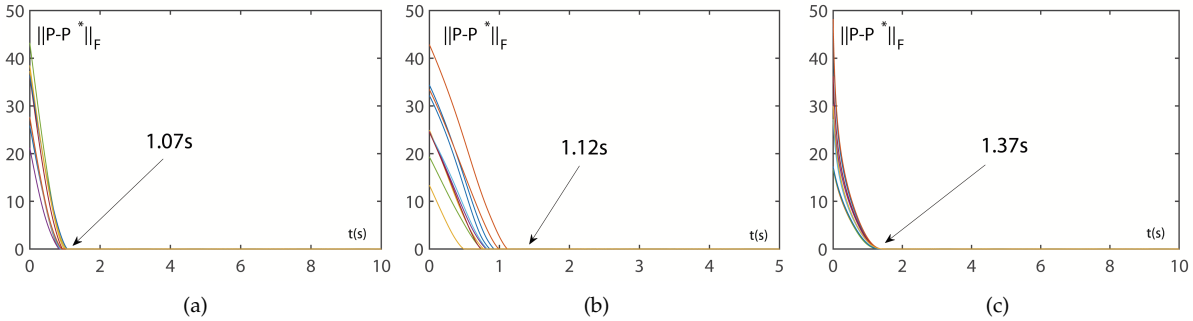

Figure 5. The curves in the figure represent the convergence errors under different initial conditions. WhenVFFTRNN uses different activation functions without perturbation to solve Sylvester Equation (1), its convergence error $||P(t) - P^*(t)||_F$ is as shown above ($\alpha = 1$ and $\lambda = \frac{1}{5}$). (**a**) Linear. (**b**) Sigmoid. (**c**) Power ($s = 3$).

4.2. Robustness Discussion

In this section, we will verify the robustness of the VFFTRNN and compare it with the ZNN. To ensure both generality and simplicity, we consider the effects of differentiation and model implementation errors. For the simulation, we set the differentiation and model implementation errors as follows:

$$\Lambda(t) = \beta_1 \begin{bmatrix} \cos 2t & 0 \\ 0 & \sin 2t \end{bmatrix}, \Omega(t) = \beta_2 \begin{bmatrix} 0 & \cos 2t \\ \sin 2t & 0 \end{bmatrix}$$
$$\Delta(t) = \beta_3 \begin{bmatrix} \sin 2t & 0 \\ 0 & \cos 2t \end{bmatrix}.$$

where β_1, β_2 and β_3 are adjustable factors that control the magnitude of these disturbances.

In this section, we will compare the robustness of VFFTRNN and ZNN using the four cases shown in Figure 6. From Figure 6, it can be observed that when $\beta_1 = \beta_2 = \beta_3$, the disturbances are small, but the computation error $||P(t) - P^*(t)||_F$ of ZNN does not converge to zero. In contrast, for VFFTRNN, we can see that the computation error converges within approximately 4 s.

Examining the other three cases, it is evident that the computation error of VFFTRNN still converges to zero, even with much larger disturbances. Notably, the convergence time of VFFTRNN decreases as the design factor α increases or λ decreases.

In conclusion, although the presence of disturbances can affect the convergence performance of VFFTRNN, it still converges to zero in finite time, regardless of the size of the disturbance, confirming the robustness of the model.

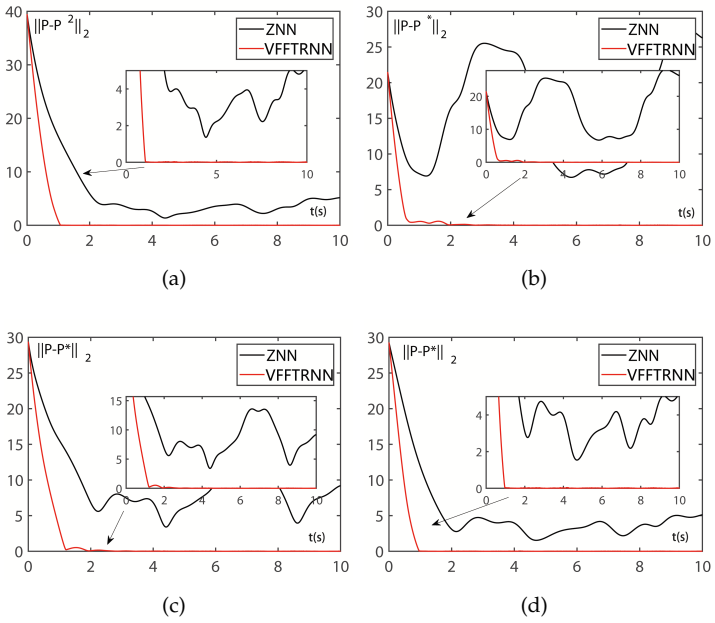

Figure 6. Robustness of ZNN and VFFTRNN with varying degrees of differential error and model implementation error. The arrows point to the original location of the magnified detail in each figure. (**a**) $\beta_1 = \beta_2 = \beta_3 = 0.05$. (**b**) $\beta_1 = \beta_2 = 0.05$ and $\beta_3 = 0.15$. (**c**) $\beta_1 = \beta_3 = 0.05$. and $\beta_2 = 0.15$. (**d**) $\beta_1 = 0.15$ and $\beta_2 = \beta_3 = 0.05$.

5. Conclusions

In this paper, we propose a novel neural network called VFFTRNN and investigate its finite-time convergence and robustness for solving the Sylvester equation under three different activation functions. Compared to the conventional ZNN, the proposed VFFTRNN introduces a time-varying coefficient and a fractional power term, which accelerate the convergence speed. More importantly, VFFTRNN not only guarantees finite-time convergence but also achieves a super-exponential convergence rate. Simulation results demonstrate the superior performance of the proposed neural network in solving the Sylvester equation. Therefore, VFFTRNN offers significant advantages for real hardware implementation.

In the future, for hardware implementation, we will consider using FPGA to implement VFFTRNN networks. The flexibility and parallel processing capabilities of FPGA make it highly suitable for real-time applications, enabling efficient computation of network operations. This method will help achieve high performance and low power consumption, making it a feasible solution for the practical deployment of VFFTRNN networks.

Author Contributions: Methodology, H.T.; conceptualization, J.W.; software, H.G.; validation, Z.Z.; formal analysis, L.T.; investigation, Q.Z.; project administration, C.L. All authors have read and agreed to published version of this manuscript.

Funding: This work was supported in part by the Jiangxi Provincial Key Laboratory of Intelligent Systems and Human–Machine Interaction under grant 2024SSY03121, and in part by the Science and Technology Department of Jiangxi Province of China under grants 20204ABC03A39, 20161ACB21007, 20171BBE50071, and 20171BAB202033.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wei, Q.; Dobigeon, N.; Tourneret, J.Y.; Bioucas-Dias, J.; Godsill, S. R-FUSE: Robust fast fusion of multiband images based on solving a Sylvester equation. *IEEE Signal Process. Lett.* **2016**, *23*, 1632–1636. [CrossRef]
2. Wang, L.; Li, D.; He, T.; Xue, Z. Manifold regularized multi-view subspace clustering for image representation. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 283–288.
3. Hu, H.; Lin, Z.; Feng, J.; Zhou, J. Smooth representation clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 3834–3841.
4. Jang, J.S.; Lee, S.Y.; Shin, S.Y. An optimization network for matrix inversion. In *Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 1987.
5. Fa-Long, L.; Zheng, B. Neural network approach to computing matrix inversion. *Appl. Math. Comput.* **1992**, *47*, 109–120. [CrossRef]
6. Cichocki, A.; Unbehauen, R. Neural networks for solving systems of linear equations and related problems. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **1992**, *39*, 124–138. [CrossRef]
7. Xia, Y.; Wang, J. A recurrent neural network for solving linear projection equations. *Neural Netw.* **2000**, *13*, 337–350. [CrossRef]
8. Zhang, Y.; Wang, J. Recurrent neural networks for nonlinear output regulation. *Automatica* **2001**, *37*, 1161–1173. [CrossRef]
9. Li, S.; Li, Y. Nonlinearly activated neural network for solving time-varying complex Sylvester equation. *IEEE Trans. Cybern.* **2013**, *44*, 1397–1407. [CrossRef] [PubMed]
10. Xiao, L.; Liao, B.; Luo, J.; Ding, L. A convergence-enhanced gradient neural network for solving Sylvester equation. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 3910–3913.
11. Zhang, Z.; Zheng, L.; Weng, J.; Mao, Y.; Lu, W.; Xiao, L. A new varying-parameter recurrent neural-network for online solution of time-varying Sylvester equation. *IEEE Trans. Cybern.* **2018**, *48*, 3135–3148. [CrossRef]
12. Zhang, Y.; Jiang, D.; Wang, J. A recurrent neural network for solving Sylvester equation with time-varying coefficients. *IEEE Trans. Neural Netw.* **2002**, *13*, 1053–1063. [CrossRef]
13. Yan, X.; Liu, M.; Jin, L.; Li, S.; Hu, B.; Zhang, X.; Huang, Z. New zeroing neural network models for solving nonstationary Sylvester equation with verifications on mobile manipulators. *IEEE Trans. Ind. Inform.* **2019**, *15*, 5011–5022. [CrossRef]
14. Zhang, Z.; Zheng, L. A complex varying-parameter convergent-differential neural-network for solving online time-varying complex Sylvester equation. *IEEE Trans. Cybern.* **2018**, *49*, 3627–3639. [CrossRef]
15. Deng, J.; Li, C.; Chen, R.; Zheng, B.; Zhang, Z.; Yu, J.; Liu, P.X. A Novel Variable-Parameter Variable-Activation-Function Finite-Time Neural Network for Solving Joint-Angle Drift Issues of Redundant-Robot Manipulators. *IEEE/ASME Trans. Mechatron.* **2024**, 1–12. [CrossRef]
16. Xu, X.; Sun, J.; Endo, S.; Li, Y.; Benjamin, S.C.; Yuan, X. Variational algorithms for linear algebra. *Sci. Bull.* **2021**, *66*, 2181–2188. [CrossRef] [PubMed]
17. Li, Y.; Chen, W.; Yang, L. Multistage linear gauss pseudospectral method for piecewise continuous nonlinear optimal control problems. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 2298–2310. [CrossRef]
18. Wang, X.; Liu, J.; Qiu, T.; Mu, C.; Chen, C.; Zhou, P. A real-time collision prediction mechanism with deep learning for intelligent transportation system. *IEEE Trans. Veh. Technol.* **2020**, *69*, 9497–9508. [CrossRef]
19. Zhang, Z.; Li, S.; Zhang, X. Simulink comparison of varying-parameter convergent-differential neural-network and gradient neural network for solving online linear time-varying equations. In Proceedings of the 2016 12th World Congress on Intelligent Control and Automation (WCICA), Guilin, China, 12–15 June 2016; pp. 887–894
20. Zhang, Y.; Chen, D.; Guo, D.; Liao, B.; Wang, Y. On exponential convergence of nonlinear gradient dynamics system with application to square root finding. *Nonlinear Dyn.* **2015**, *79*, 983–1003. [CrossRef]
21. Wang, S.; Dai, S.; Wang, K. Gradient-based neural network for online solution of lyapunov matrix equation with li activation function. In Proceedings of the 4th International Conference on Information Technology and Management Innovation, Shenzhen, China, 12–13 September 2015; Atlantis Press: Amsterdam, The Netherlands, 2015; pp. 955–959.
22. Ma, W.; Zhang, Y.; Wang, J. Matlab simulink modeling and simulation of zhang neural networks for online time-varying sylvester equation solving. *Comput. Math. Math. Phys.* **2008**, *47*, 285–289.
23. Li, S.; Chen, S.; Liu, B. Accelerating a recurrent neural network to finite-time convergence for solving time-varying sylvester equation by using a sign-bi-power activation function. *Neural Process. Lett.* **2013**, *37*, 189–205. [CrossRef]
24. Shen, Y.; Miao, P.; Huang, Y.; Shen, Y. Finite-time stability and its application for solving time-varying sylvester equation by recurrent neural network. *Neural Process. Lett.* **2015**, *42*, 763–784. [CrossRef]
25. Jin, L.; Zhang, Y.; Li, S.; Zhang, Y. Modified znn for time-varying quadratic programming with inherent tolerance to noises and its application to kinematic redundancy resolution of robot manipulators. *IEEE Trans. Ind. Electron.* **2016**, 63, 6978–6988. [CrossRef]

26. Mao, M.; Li, J.; Jin, L.; Li, S.; Zhang, Y. Enhanced discrete-time zhang neural network for time-variant matrix inversion in the presence of bias noises. *Neurocomputing* **2016**, *207*, 220–230. [CrossRef]
27. Xiao, L.; Liao, B. A convergence-accelerated zhang neural network and its solution application to lyapunov equation. *Neurocomputing* **2016**, 193, 213–218. [CrossRef]
28. Liao, B.; Zhang, Y.; Jin, L. Taylor discretization of znn models for dynamic equality-constrained quadratic programming with application to manipulators. *Neural Netw. Learn. Syst. IEEE Trans.* **2016**, *27*, 225–237. [CrossRef] [PubMed]
29. Xiao, L. A finite-time convergent zhang neural network and its application to real-time matrix square root finding. *Neural Comput. Appl.* **2017**, *31*, 793–800. [CrossRef]
30. Zhang, Y.; Mu, B.; Zheng, H. Link between and comparison and combination of zhang neural network and quasi-newton bfgs method for time-varying quadratic minimization. *IEEE Trans. Cybern.* **2013**, *43*, 490–503. [CrossRef]
31. Yan, D.; Li, C.; Wu, J.; Deng, J.; Zhang, Z.; Yu, J.; Liu, P.X. A novel error-based adaptive feedback zeroing neural network for solving time-varying quadratic programming problems. *Mathematics* **2024**, *12*, 2090. [CrossRef]
32. Zhang, Z.; Lu, Y.; Zheng, L.; Li, S.; Yu, Z.; Li, Y. A new varying-parameter convergent-differential neural-network for solvingtime-varying convex QP problem constrained by linear-equality. *IEEE Trans. Autom. Control* **2018**, *63*, 4110–4125. [CrossRef]
33. Horn, R.A.; Johnson, C.R. *Topics in Matrix Analysis, 1991*; Cambridge University Presss: Cambridge, UK, 1991; Volume 37, p. 39.
34. Strang, G.; Freund, L. Introduction to applied mathematics. *J. Appl. Mech.* **1986**, *53*, 480. [CrossRef]
35. Mead, C.; Ismail, M. *Analog VLSI Implementation of Neural Systems*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 80.
36. Zhang, Y.; Ruan, G.; Li, K.; Yang, Y. Robustness analysis of the zhang neural network for online time-varying quadratic optimization. *J. Phys. A Math. Theor.* **2010**, *43*, 245202. [CrossRef]