A robust noise tolerant zeroing neural network for solving time-varying linear matrix equations

Dimitrios Gerontitis[1], Ratikanta Behera[2], Yang Shi[3], Predrag S. Stanimirović[4]

[1] *Department of Information and Electronic Engineering, International Hellenic University, Thessaloniki, Greece*,
[2] *Department of Mathematics, University of Central Florida, Orlando, FL, 32816, USA*,
[3] *School of Information Engineering, Yangzhou University, Yangzhou, People's Republic of China*,
[4] *University of Niš, Faculty of Sciences and Mathematics, 18000 Niš, Serbia*
E-mail: [1] dimitrios_gerontitis@yahoo.gr, [2] ratikanta.behera@ucf.edu, [3] shiy@yzu.edu.cn, [4] pecko@pmf.ni.ac.rs

Abstract

A robust noise tolerant zeroing neural network is introduced for solving time-varying linear matrix equations. The convergence speed of designed neural dynamics is analysed theoretically and compared with convergence of neural networks which include traditional activation functions (i.e., the tunable activation function, versatile activation function and the modified sign-bi-power activation function). We investigate theoretically and verify experimentally the behaviour of the proposed robust noise tolerant zeroing neural network with the novel effective activation function. Simulation examples for demonstration purposes are presented to demonstrate superiority of the proposed activation over already existing activation functions.

Keywords: robust noise tolerant zeroing neural network; time-varying linear matrix equations

Mathematics Subject Classification (2020): 68T05, 15A09, 65F20.

1 Introduction and motivations

The design of zeroing neural networks with accelerated convergence speed in a finite time period is an important and exciting research topic in the recurrent neural networks (RNN) area. Many dynamical systems of finite time convergence have been created, such as finite-time zeroing neural networks (FTZNN) [1–5] and varying-parameter finite-time zeroing neural networks (VPFTZNN) [6–10]. In contrast, different nonlinear activation functions have been applied in the known Zhang neural network (ZNN) dynamics [11–20, 52, 55] to increase its convergence speed in a finite time. From this fact, the importance of different activation functions is observable. It is useful to mention that solutions of matrix equations play an important role in scientific research and applications, such as scattering [21–24], data analysis [25, 26] and robotics [27–33].

In view of previous comments, it will be more appropriate to establish a new nonlinear activation function. The purpose of this activation function is to accelerate the convergence speed of nonlinear ZNN dynamical system based on different parameters. This goal is crucial in the artificial intelligence because of direct applicability of dynamical systems, including future applications of the proposed activation function in the study and solution of different time-varying problems that appear in nature and in different parts of applied mathematics.

There are a lot of research articles in which the main intention is to solve the time-varying matrix equation, $A(t)X(t) = I$, and the time-variant Sylvester equations $A(t)X(t) - X(t)B(t) + C(t) = 0$, where t is the time and $X(t) \in \mathbb{R}^{\nu\times\nu}$ represents the matrix to be generated with the reference to smoothly time-variant matrices $A(t), B(t) \in \mathbb{R}^{\nu\times\nu}$ and $C(t) \in \mathbb{R}^{\nu\times\nu}$ [3, 5–8, 10]. However, the time-varying Stein matrix equations have many special properties and are widely used in engineering and scientific computation.

In this paper, the following two problems are considered.

- **Problem-1** Let $P_1(t) \in \mathbb{R}^{\nu\times\nu}$, $P_2(t) \in \mathbb{R}^{\mu\times\mu}$, $P_3(t) \in \mathbb{R}^{\nu\times\mu}$ be the the time-varying coefficient matrices for $t \geq 0$. Solve the time-varying linear matrix equation

$$P_1(t)V(t)P_2(t) = P_3(t), \tag{1.1}$$

with respect to $V(t) \in \mathbb{R}^{\nu\times\mu}$.

- **Problem-2** Let $P_1(t) \in \mathbb{R}^{\nu\times\nu}$, $P_2(t) \in \mathbb{R}^{\mu\times\mu}$, $P_3(t) \in \mathbb{R}^{\nu\times\mu}$ be the the time-varying coefficient matrices for $t \geq 0$. Solve the time-varying Stein matrix equation [48, 62–65].

$$P_1(t)V(t)P_2(t) + V(t) = P_3(t). \tag{1.2}$$

The equation (1.2) is a generalization of the Lyapunov matrix equation and the Sylvester matrix equation. i.e., the Stein matrix equation becomes the Lyapunov matrix equation if $P_1(t) = P_2^{\mathrm{T}}(t)$, and reduces to the Sylvester matrix equation if $P_2(t)$ is nonsingular.

Main contributions of this paper are summarized as follows.

(1) A novel and efficient activation function is proposed.

(2) Theoretical analysis proves a finite time convergence of the proposed dynamical models to the theoretical solutions of the linear matrix equation (1.1) and the Stein matrix equation (1.2).

(3) A novel noise resistant model based on the new activation function is defined.

(4) Simulative examples are presented in order to demonstrate superiority of the novel activation function over existing activation functions.

The paper is organized as follows. A novel activation function and corresponding robust noise tolerant and finite-time convergent zeroing neural networks (ZNNs) for solving linear matrix equation (1.1) and the Stein equation (1.2) are proposed proposed in Section 2. In Section 3, we study the convergence analysis theoretically of the proposed ZNNs without noise and with the presence of noise. A few numerical examples are discussed in Section 4 to demonstrate superiority of the new nonlinear activation functions over existing activation functions. Obtained results are summarized in the last section along with a discussion of open problems for future research.

2 A robust noise tolerant and finite-time convergent zeroing neural network

Activation functions are significantly impacted in convergence and robustness of ZNN models. At the beginning of this section, we briefly explain previous nonlinear activation functions which provide finite estimation of the convergence time are referred as follows.

- The Li or sign-bi-power (sbp) activation function [34–41, 58]:

$$\mathrm{spb}(u) = \mathrm{sgn}^p(u) + \mathrm{sgn}^{1/p}(u), \tag{2.1}$$

with

$$\mathrm{sgn}^p(u) = |u|^p\mathrm{sgn}(u), \quad p \in (0,1), \tag{2.2}$$

where

$$\mathrm{sgn}(u) = \begin{cases} 1, & u > 0 \\ 0, & u = 0 \\ -1, & u < 0. \end{cases} \tag{2.3}$$

In order to complete our presentation, some of known activation functions used in numerical testing are surveyed.

- The tunable activation function [42–46]:

$$G_{tun}(u) = \mathrm{sgn}(u)(|u|^p + |u| + |u|^{\frac{1}{p}}), \quad p \in (0,1). \tag{2.4}$$

- The versatile activation function (VAF) [47, 48, 51] given by:

$$G_{vaf}(u) = (b_1|u|^r + b_2|u|^w)\mathrm{sgn}(u) + b_3u + b_4\mathrm{sgn}(u), \tag{2.5}$$

where $b_1, b_2 > 0$, $b_3, b_4 \geq 0$, $r \in (0,1)$ and $w > 1$.

- The modified sign-bi-power activation function (MsbpAF) [48–51] given by:

$$G_{msbp}(u) = (b_1\exp(|u|^r)|u|^{1-r})\mathrm{sgn}(u)/r + b_2u + b_3\mathrm{sgn}(u), \tag{2.6}$$

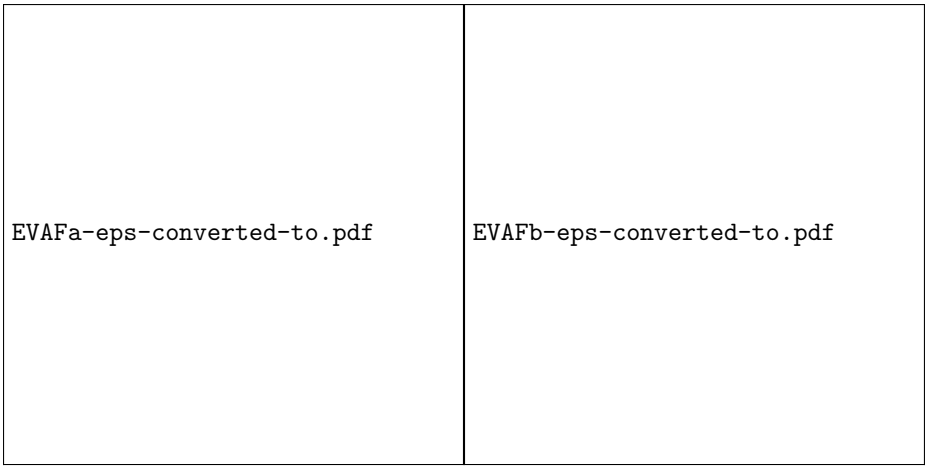
In this research, the following extended versatile activation function $\Omega_m(u)$ of the order m is proposed:

$$\Omega_m(u) = \left(b_1 \sum_{k=1}^{m} |u|^{r_k} + b_2 \sum_{k=1}^{m} |u|^{w_k} \right) \mathrm{sgn}(u) + b_3 u + b_4 \sinh(u) + b_5(\exp(|u|) - 1)\mathrm{sgn}(u), \tag{2.7}$$

where $r_k \in (0,1)$, $w_k > 1$, b_1, b_2, b_3, b_4 satisfy the same constraints as in previous activation functions, $b_5 \geq 0$ and $k = 1, 2, \ldots, m$. In order to make presentation efficient, we will choose $0 < r_1 \leq r_2 \leq \cdots \leq r_m < 1$ and $1 < w_1 \leq w_2 \leq \cdots \leq w_m$. Then the corresponding ZNN model based on activation function (2.7) is obtained by

$$\dot{E}(t) = -\lambda\, \Omega_m(E(t)). \tag{2.8}$$

The formula (2.8) is called robust noise tolerant zeroing neural network and symbolized as RNTZNN. Graphs of $\Omega_1(u)$ with $r_1 = w_1 = 1$ are plotted in Fig.1(a), while Fig.1(b) contains graphs of $\Omega_2(u)$ under $r_1 = r_2 = w_1 = w_2 = 1$. The error function appropriate for solving the Problem-1, (solving the linear matrix equation



(a) (b)

Figure 1: Behaviour of $\Omega_m(u)$ for: (a) $m = 1$; (b) $m = 2$.

(1.1)) is defined by

$$E(t) := E_L(t) = P_1(t)V(t)P_2(t) - P_3(t) \tag{2.9}$$

Further, the error function which leads to the solution of the Stein matrix equation (1.2) (the Problem-2) is defined as

$$E(t) := E_S(t) = P_1(t)V(t)P_2(t) + V(t) - P_3(t). \tag{2.10}$$

Using the general ZNN design (2.8) for establishing every element $e_{ij}(t)$ ($i = 1, 2, \ldots \nu$ and $j = 1, 2, \ldots \mu$) of $E(t)$ convergent to zero, we design a robust noise tolerant zeroing neural network for solving considered linear matrix equations (1.1) and (1.2). In particular, for solving (1.1), we obtain the following model based on (2.8) and the error function (2.9).

$$P_1(t)\dot{V}(t)P_2(t) = -\dot{P}_1(t)V(t)P_2(t) - P_1(t)V(t)\dot{P}_2(t) + \dot{P}_3(t) - \lambda\Omega_m(P_1(t)V(t)P_2(t) - P_3(t)). \tag{2.11}$$

Further, for solving Stein matrix equation (1.2), we obtain the following model by using the equation (2.8) and the error function (2.10).

$$\dot{V}(t) = \dot{P}_3(t) - \dot{P}_1(t)V(t)P_2(t) - P_1(t)\dot{V}(t)P_2(t) - P_1(t)V(t)\dot{P}_2(t) - \lambda\,\Omega_m(P_1(t)V(t)P_2(t) + V(t) - P_3(t)). \tag{2.12}$$

We end this section with the following lemma from [48, 51, 53, 54] which will be frequently used in next section. As usual, $\mathbb{R}_+$ denotes the set of positive real numbers.

Lemma 2.1. *For a nonlinear dynamical system $\dot{z}(t) = \omega(z(t), t)$ if exists a radially continuous unbounded function $\mathrm{V} : \mathbb{R}^\nu \mapsto \mathbb{R}_+ \cup \{0\}$ with the capacity $\mathrm{V}(\psi) = 0$ for $\psi \in \mathrm{D}$ and any solution $\psi(t)$ satisfies*

$$\dot{V}(t) \leq -g_1 V^{\mathrm{p}}(\psi(t)) - g_2 V^{\mathrm{w}}(\psi(t)), \tag{2.13}$$

where $g_1, g_2 > 0$, $0 < \mathrm{p} < 1$, $\mathrm{w} > 1$ are constant parameters, then the dynamic inequality (2.13) is fixed-time stable and the upper bound for predefined convergence time t_p satisfies

$$t_p \leq \frac{1}{g_1(1-\mathrm{p})} + \frac{1}{g_2(\mathrm{w}-1)}. \tag{2.14}$$

3 Convergence analysis

In this section, a predefined-time convergence of proposed dynamical systems (2.11) and (2.12) are investigated without or with noise disturbance. To discuss the convergence analysis, the known notation $e_{ij}(t)$ and $\dot{v}_{ij}(t)$, with $i = 1, 2, \ldots, \nu$ and $j = 1, 2, \ldots, \mu$, will be used to denote (ij)th entries of $E(t)$ and $\dot{V}(t)$, respectively.

3.1 Theoretical convergence analysis without noise

Following estimations about the finite-time convergence are obtained for novel dynamical systems (2.11) and (2.12).

Theorem 3.1. *Consider the time-varying matrices $P_1(t), P_2(t), P_3(t)$ as in (1.1) and $E(t) := E_L(t) = [e_{ij}(t)]$, $i = 1, 2, \ldots, \nu$ and $j = 1, 2, \ldots, \mu$. If $\Omega_m(u)$ activation is used, then the neural state matrix $V(t)$ of the RNTZNN model (2.11) starting from arbitrary initial state $V(0)$ converges to theoretical solution of (1.1) in a predefined time t_f satisfying*

$$t_f \leq \frac{1}{\lambda\, b_1\, (1 - r_m)} + \frac{1}{\lambda\, b_2\, (w_m - 1)}. \tag{3.1}$$

Proof. The Lyapunov candidate is defined by $v_{ij}(t) = |e_{ij}(t)|^2$ for the (ij)th equation with the activation function $\Omega_m(u)$. The time derivative of $v_{ij}(t)$ is equal to

$$\begin{aligned}\dot{v}_{ij}(t) = -2\lambda e_{ij}(t) \left(b_1 \sum_{k=1}^{m} |e_{ij}(t)|^{r_k} + b_2 \sum_{k=1}^{m} |e_{ij}(t)|^{w_k} + b_5(\exp(|e_{ij}(t)|) - 1) \right) \operatorname{sgn}(e_{ij}(t)) \\ -2\lambda e_{ij}(t) \left(b_3 e_{ij}(t) + b_4 \sinh(e_{ij}(t)) \right).\end{aligned} \tag{3.2}$$

- If $e_{ij}(t) = 0$ then $\dot{v}_{ij}(t) = 0$.
- If $e_{ij}(t) > 0$ then $e_{ij}(t) = |e_{ij}(t)|$ and $\operatorname{sgn}(e_{ij}(t)) = 1$. Thus the equality (3.2) can be rewritten as

$$\begin{aligned}\dot{v}_{ij}(t) = -2\lambda |e_{ij}(t)| \left(b_1 \sum_{k=1}^{m} |e_{ij}(t)|^{r_k} + b_2 \sum_{k=1}^{m} |e_{ij}(t)|^{w_k} + b_5(\exp(|e_{ij}(t)|) - 1) \right) \\ -2\lambda |e_{ij}(t)| \left(b_3 |e_{ij}(t)| + b_4 \sinh(|e_{ij}(t)|) \right).\end{aligned} \tag{3.3}$$

  Since $\exp(|e_{ij}(t)|) - 1 \geq |e_{ij}(t)|$ and $\sinh(|e_{ij}(t)|) \geq |e_{ij}(t)|$, the above equation (3.3) can be simplified to

$$\dot{v}_{ij}(t) \leq -2\lambda |e_{ij}(t)| \left(b_1 \sum_{k=1}^{m} |e_{ij}(t)|^{r_k} + b_2 \sum_{k=1}^{m} |e_{ij}(t)|^{w_k} + (b_3 + b_4 + b_5)\, |e_{ij}(t)| \right), \tag{3.4}$$

  which implies

$$\dot{v}_{ij}(t) \leq -2\lambda |e_{ij}(t)| \left(b_1 \sum_{k=1}^{m} |e_{ij}(t)|^{r_k} + b_2 \sum_{k=1}^{m} |e_{ij}(t)|^{w_k} \right) \leq -2\lambda b_1 v_{ij}^{\frac{r_m+1}{2}}(t) - 2\lambda b_2 v_{ij}^{\frac{w_m+1}{2}}(t). \tag{3.5}$$

  The matrix form of the inequality (3.5) is given as

$$\dot{V}(t) \leq -2\lambda b_1 V^{\frac{r_m+1}{2}}(t) - 2\lambda b_2 V^{\frac{w_m+1}{2}}(t). \tag{3.6}$$

- If $e_{ij}(t) < 0$ then $e_{ij}(t) = -|e_{ij}(t)|$, $\operatorname{sgn}(e_{ij}(t)) = -1$, and $\sinh(-|e_{ij}(t)|) = -\sinh(|e_{ij}(t)|)$, so that dynamical system of the form (3.6) is obtained in the similar process as in case of $e_{ij}(t) > 0$.

  Using the estimation of the Lemma 2.1 it is possible to determine the predefined time of (3.6) as in (3.1). and hence complete the proof. □

Subsequent corollary about the estimation of the convergence time is obtained for the special case of $|e_{ij}(t)| \leq 1$.

Corollary 3.1. *Under the assumption of Theorem 3.1 and under the particular assumption that all the elements of $E_L(t) = [e_{ij}(t)]$ in (2.9) satisfy $|e_{ij}(t)| \leq 1$, the predefined convergence time of dynamics (2.11) is limited by*

$$t_{f_a} \leq \frac{1}{m\, \lambda\, b_1\, (1 - r_m)} + \frac{1}{m\, \lambda\, b_2\, (w_m - 1)} \tag{3.7}$$

Proof. From the inequality (3.4) the following relationship holds

$$\dot{v}_{ij}(t) \leq -2\lambda|e_{ij}(t)|\left(b_1\sum_{k=1}^{m}|e_{ij}(t)|^{r_k}+b_2\sum_{k=1}^{m}|e_{ij}(t)|^{w_k}+(b_3+b_4+b_5)\,|e_{ij}(t)|\right).$$

Since $|e_{ij}(t)| \leq 1$, it follows

$$|e_{ij}(t)|^{r_k} \geq |e_{ij}(t)|^{r_m} \quad \text{and} \quad |e_{ij}(t)|^{w_k} \geq |e_{ij}(t)|^{w_m} \quad \text{for} \quad k = 1, 2, \ldots, m$$

which implies

$$|e_{ij}(t)|^{r_1}+\cdots+|e_{ij}(t)|^{r_m} \geq |e_{ij}(t)|^{r_m}+\cdots+|e_{ij}(t)|^{r_m} = m|e_{ij}(t)|^{r_m},$$
$$|e_{ij}(t)|^{w_1}+\cdots+|e_{ij}(t)|^{w_m} \geq |e_{ij}(t)|^{w_m}+\cdots+|e_{ij}(t)|^{w_m} = m|e_{ij}(t)|^{w_m}.$$

Thus the inequality (3.4) is simplified into

$$\dot{v}_{ij}(t) \leq -2\lambda b_1 m v_{ij}^{\frac{r_m+1}{2}}(t) - 2\lambda b_2 m v_{ij}^{\frac{w_m+1}{2}}(t).$$

The matrix form of the above inequality can be written as

$$\dot{V}(t) \leq -2\lambda b_1 m V^{\frac{r_m+1}{2}}(t) - 2\lambda b_2 m V^{\frac{w_m+1}{2}}(t).$$

Then the result of the Lemma 2.1 leads to (3.7). □

Next results contains the convergence and robustness of the time-varying Stein matrix equation (1.2).

Theorem 3.2. *Assume that time-varying matrices $P_1(t), P_2(t), P_3(t)$ are from* (1.2) *and $E(t) := E_S(t) = [e_{ij}(t)]$ is the error matrix in* (2.10)*. Then, the neural state matrix $V(t)$ of the RNTZNN model* (2.12) *starting from arbitrary initial state $V(0)$ converges to theoretical solution of* (1.2) *in a finite time estimated by*

$$t_g \leq \frac{1}{\lambda\, b_1\,(1-r_m)} + \frac{1}{\lambda\, b_2\,(w_m-1)}. \tag{3.8}$$

Proof. This claim can be verified following the proof of Theorem 3.1. □

In a similar way as in the proof of the Corollary 3.1 one can verify the following corollary.

Corollary 3.2. *For the special choice that all the elements of the Zhang function $E_S(t) = [e_{ij}(t)]$ in* (2.10) *satisfy $|e_{ij}(t)| \leq 1$, the convergence time of* (2.12) *is given by*

$$t_{g_a} \leq \frac{1}{m\,\lambda\, b_1\,(1-r_m)} + \frac{1}{m\,\lambda\, b_2\,(w_m-1)}. \tag{3.9}$$

Observing the convergence bounds for t_{f_a} in (3.7), if $r_1 = r_2 = \cdots = r_m = r$, $w_1 = w_2 = \cdots = w_m = w$, $b_3 = b_4 = b_5 = 0$ and all the elements of error matrix function satisfy $|e_{ij}(t)| \leq 1$ then the convergence time of (2.11) is equal to

$$t_f = t_{f_a} \leq \frac{1}{m\,\lambda\, b_1\,(1-r)} + \frac{1}{m\,\lambda\, b_2\,(w-1)}. \tag{3.10}$$

From [47, 48, 51], the predefined convergence time of corresponding ZNN model activated by VAF is given

$$t_v \leq \frac{1}{\lambda\, b_1\,(1-r)} + \frac{1}{\lambda\, b_2\,(w-1)}. \tag{3.11}$$

So by comparing the convergence bounds (3.10) and (3.11) it can be concluded that

$$\frac{t_f}{t_v} \leq \frac{1}{m} \leq 1,$$

and it is observable that $t_f \leq t_v$.

It is important to note that the number m of parameters r_k and w_k, $k = 1, 2, \ldots, m$ in the proposed activation function (2.7) plays important role in the acceleration of convergence speed because of the estimations (3.7), (3.10). Increasing m the convergence times when $|e_{ij}(t)| \leq 1$ vanish faster to 0 and especially as $m \to \infty$ then $t_f = t_{f_a} \to 0$. From this fact it is easy to understand the advantage which provides $\Omega_m(u)$ against previous activation functions, since in each case there exists an appropriate number m which will vanish the convergence times (3.7), (3.10) to bounds lesser than in the case of other activation functions. One more parameter which plays important role in the acceleration of convergence speed in (2.7) is based on the values of b_1, b_2, b_3, b_4, b_5. Increasing values of these constants reduces the limits of convergence times.

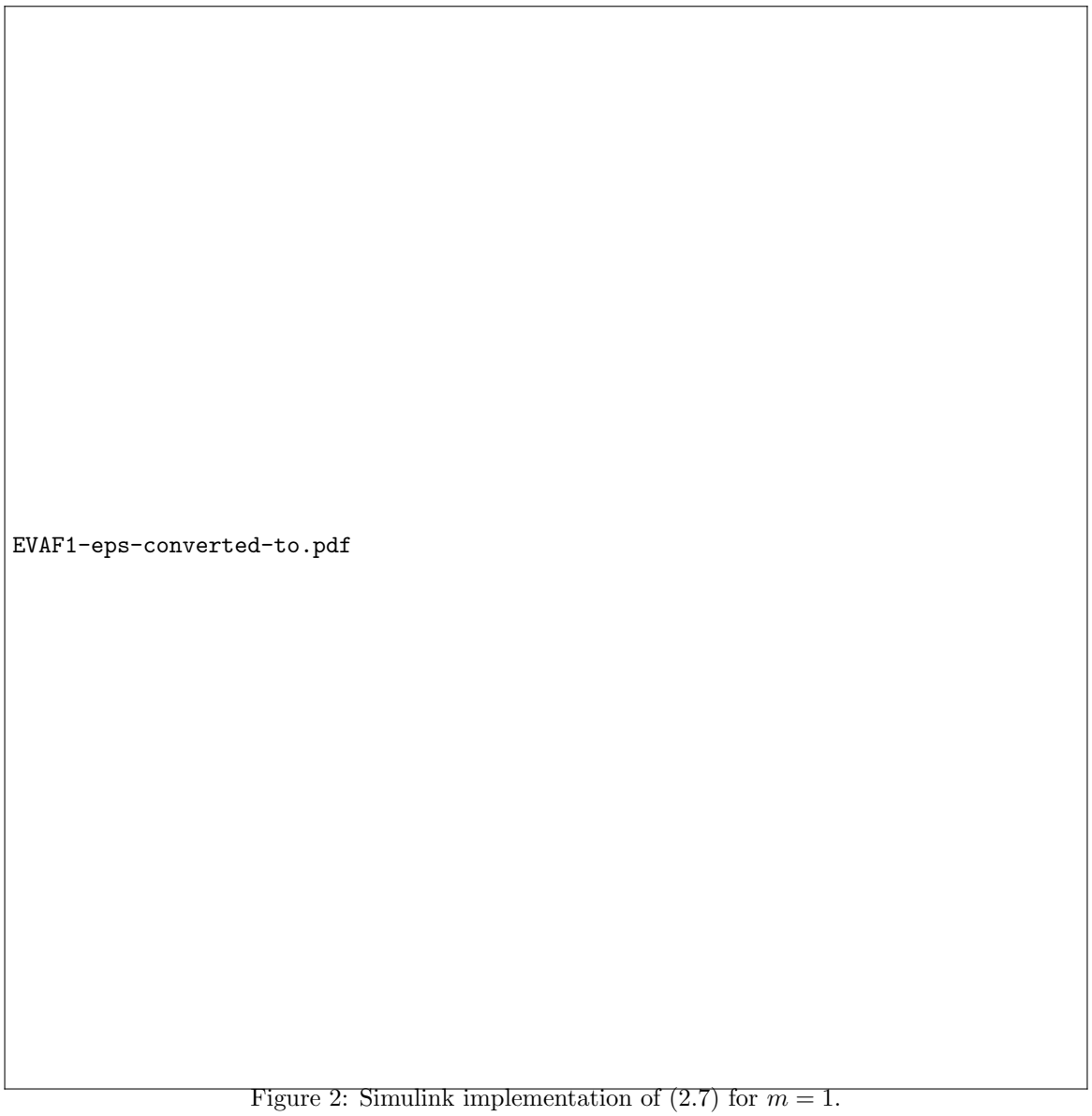


Figure 2: Simulink implementation of (2.7) for $m = 1$.

3.2 Theoretical convergence analysis in the presence of noise

The next natural problem is to apply a kind of a time-variant noise in the proposed dynamics (2.8). For this reason the following disturbed dynamical system is considered:

$$\dot{E}(t) = -\lambda\,\Omega_m(E(t)) + N(t), \tag{3.12}$$

where $N(t) = -\lambda|h(t)|E(t)$ and $h(t)$ is a bounded real function of one variable. Formula (3.12) for each element $e_{ij}(t)$ becomes

$$\dot{e}_{ij}(t) = -\lambda\,\Omega_m(e_{ij}(t)) + n_{ij}(t),$$

with the assumption $n_{ij}(t) = -\lambda\, e_{ij}(t)|h(t)|$. The following theoretical result is obtained for the convergence time of (3.12).

Theorem 3.3. *Assume that the time-varying matrices $P_1(t), P_2(t), P_3(t)$ are coefficient matrices in* (1.1) *and $E(t) := E_L(t) = [e_{ij}(t)]$ is the error matrix. Then the neural state matrix $V(t)$ of the noise perturbed RNTZNN model* (3.12) *converges to theoretical solution of* (1.1) *starting from any initial state $V(0)$ in a finite time t_n bounded by*

$$t_n \leq \frac{1}{\lambda\, b_1\,(1 - r_m)} + \frac{1}{\lambda\, b_2\,(w_m - 1)}.$$

Proof. Consider the Lyapunov candidate $\upsilon_{ij}(t) = |e_{ij}(t)|^2$ defined in (2.7) for the (ij)th system with an activa-

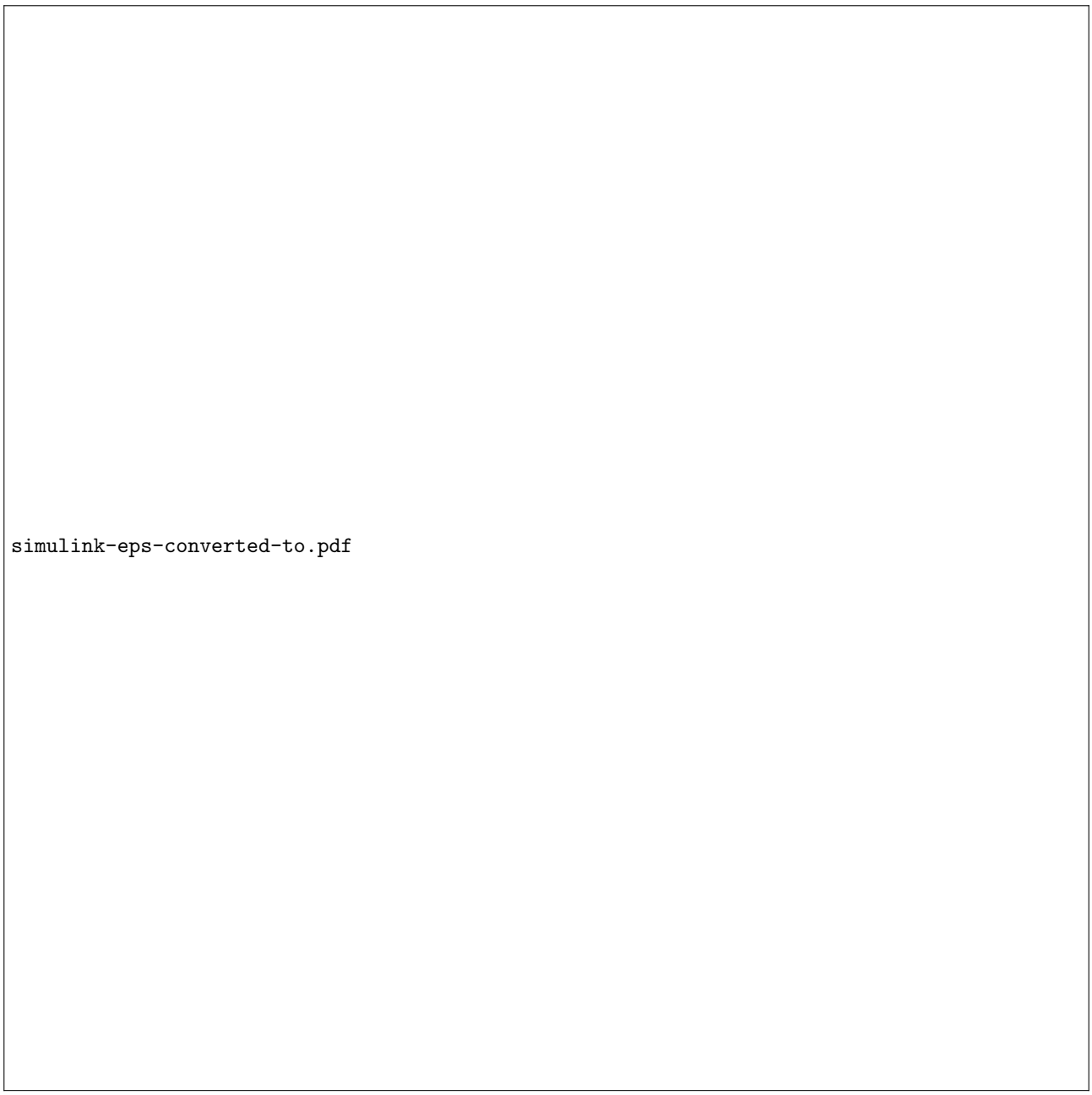


Figure 3: Simulink implementation of (2.11).

tion function $\Omega_m(u)$. Now the time derivative of $v_{ij}(t)$ is equal to

$$\begin{aligned}\dot{v}_{ij}(t) = -2\lambda e_{ij}(t)\left(b_1\sum_{k=1}^{m}|e_{ij}(t)|^{r_k} + b_2\sum_{k=1}^{m}|e_{ij}(t)|^{w_k} + b_5(\exp(|e_{ij}(t)|)-1)\right)\operatorname{sgn}(e_{ij}(t)) \\ -2\lambda e_{ij}(t)\left(b_3 e_{ij}(t) + b_4\sinh(e_{ij}(t)) + |h(t)|e_{ij}(t)\right).\end{aligned} \tag{3.13}$$

- If $e_{ij}(t) = 0$ then $\dot{v}_{ij}(t) = 0$.
- If $e_{ij}(t) > 0$ then $e_{ij}(t) = |e_{ij}(t)|$ and $\operatorname{sgn}(e_{ij}(t)) = 1$ so (3.13) can be transformed in:

$$\begin{aligned}\dot{v}_{ij}(t) = -2\lambda|e_{ij}(t)|\left(b_1\sum_{k=1}^{m}|e_{ij}(t)|^{r_k} + b_2\sum_{k=1}^{m}|e_{ij}(t)|^{w_k} + b_5(\exp(|e_{ij}(t)|)-1)\right) \\ -2\lambda|e_{ij}(t)|\left(b_3|e_{ij}(t)| + b_4\sinh(|e_{ij}(t)|) + |h(t)||e_{ij}(t)|\right).\end{aligned} \tag{3.14}$$

Since we have $\exp(|e_{ij}(t)|) - 1 \geq |e_{ij}(t)|$ and $\sinh(|e_{ij}(t)|) \geq |e_{ij}(t)|$, the above equation (3.14) can be

simplified into

$$\dot{v}_{ij}(t) \leq -2\lambda|e_{ij}(t)|\left(b_1\sum_{k=1}^{m}|e_{ij}(t)|^{r_k}+b_2\sum_{k=1}^{m}|e_{ij}(t)|^{w_k}+(b_3+b_4+b_5)\,|e_{ij}(t)|+|h(t)||e_{ij}(t)|\right). \quad (3.15)$$

Which implies

$$\dot{v}_{ij}(t) \leq -2\lambda b_1|e_{ij}(t)|\sum_{k=1}^{m}|e_{ij}(t)|^{r_k}-2\lambda b_2|e_{ij}(t)|\sum_{k=1}^{m}|e_{ij}(t)|^{w_k} \leq -2\lambda b_1 v_{ij}^{\frac{r_m+1}{2}}(t)-2\lambda b_2 v_{ij}^{\frac{w_m+1}{2}}(t),$$

This inequality is transformed to the matrix form

$$\dot{V}(t) \leq -2\lambda b_1 V^{\frac{r_m+1}{2}}(t)-2\lambda b_2 V^{\frac{w_m+1}{2}}(t). \quad (3.16)$$

We now came up to the differential inequality of the form (3.6) as in Theorem 3.1, the same results about the convergence time is obtained

- If $e_{ij}(t) < 0$ then $e_{ij}(t) = -|e_{ij}(t)|$, $\text{sgn}(e_{ij}(t)) = -1$, $\sinh(-|e_{ij}(t)|) = -\sinh(|e_{ij}(t)|)$ same equation (3.15) is obtained which implies same results as in positive $e_{ij}(t)$, and the proof is completed. □

So it is observable from the proof of Theorem 3.3 that for the special case when all elements $e_{ij}(t)$ of the error matrix $E(t)$ satisfy $|e_{ij}(t)| \leq 1$ the convergence bounds of the noise disturbed flow (3.12) is bounded by t_{n_a} where

$$t_{n_a} \leq \frac{1}{m\lambda b_1(1-r_m)}+\frac{1}{m\lambda b_2(w_m-1)},$$

The following results are obtained for the noise situation $N(t) = -\lambda E(t)|h(t)|$ in (3.12) such that $E(t)$ is defined in (2.10)

Theorem 3.4. *Assume that $P_1(t), P_2(t), P_3(t)$ are from (1.2) and the noise $N(t) = -\lambda E(t)|h(t)|$ appears in (3.12). Then the neural state matrix $V(t)$ of the RNTZNN flow (3.12), starting from the initial state $V(0)$, converges theoretical solution of (1.1) in finite time t_{g_n} estimated as*

$$t_{g_n} \leq \frac{1}{\lambda\, b_1\,(1-r_m)}+\frac{1}{\lambda\, b_2\,(w_m-1)},$$

Proof. The proof is analogous as in the case of Theorem 3.3. □

We thus conclude this section with the following corollaries, for the proof of these results, one can use the similar principles from the proof of the Corollary 3.1.

Corollary 3.3. *In the particular case when all the elements of $E(t)$ defined in (2.10), corresponding to the Stein matrix equation, satisfy $|e_{ij}(t)| \leq 1$ and under the noise disturbance $N(t) = -\lambda E(t)|h(t)|$, the convergence time of (3.12) is limited by*

$$t_{g_{n_a}} \leq \frac{1}{m\lambda b_1(1-r_m)}+\frac{1}{m\lambda b_2(w_m-1)}. \quad (3.17)$$

4 Numerical Examples

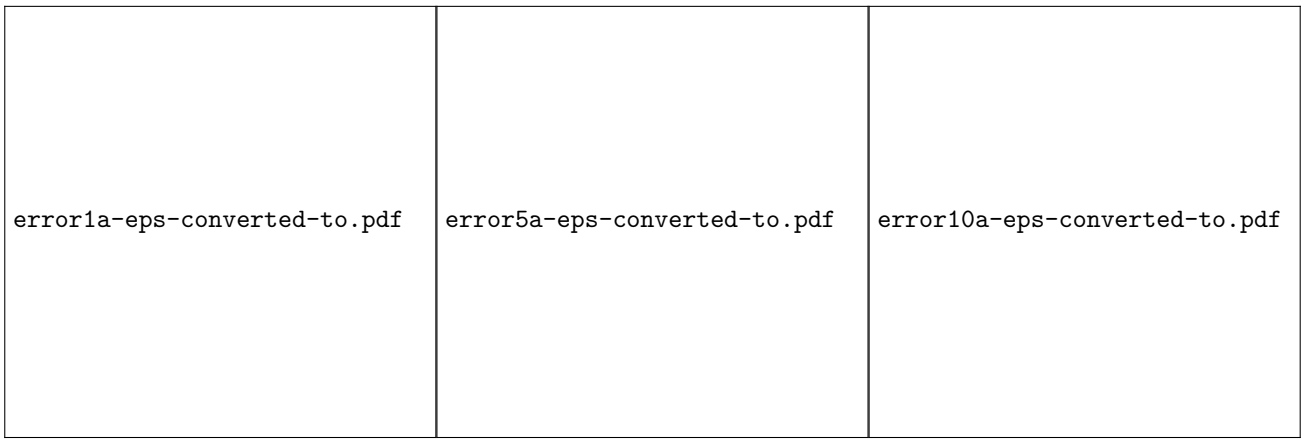
In this section the behavior of the (2.12) and (2.11) with or without the presence of noise will be examined.

4.1 Solution of linear matrix equation (1.1)

Example 4.1. The matrices from [56] are considered with entries

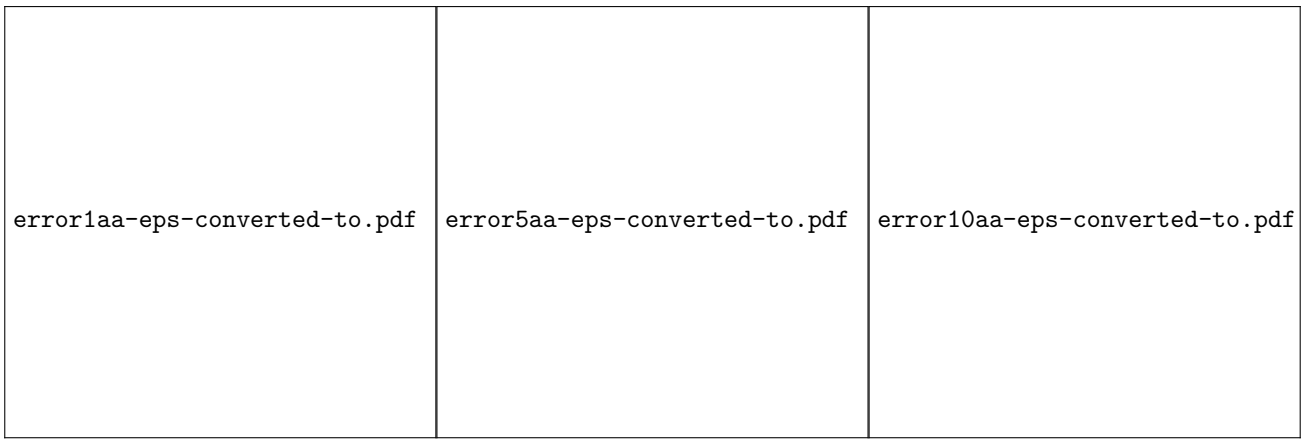
$$P_1(t) = \begin{bmatrix}\sin(0.5t) & -\cos(0.5t)\\ \cos(0.5t) & \sin(0.5t)\end{bmatrix}, P_3(t) = \begin{bmatrix}\cos(5t) & \sin(4t) & \sin(8t)\\ -\sin(5t) & (1+\cos(7t)) & \cos(6t)\end{bmatrix}$$

and $P_2(t) = I_3$, where I_3 is the 3×3 identity matrix. Under the parameters $r_1 = 0.5, r_2 = 0.6, r_3 = 0.7, r_4 = 0.8, r_5 = 0.9, r_6 = 0.95, r_7 = r_8 = r_9 = 0.99, w_1 = w_2 = \cdots = w_9 = 1.1$, $p = r_1 = 0.5$, $q = w_1 = 1.1$, $\lambda = 1, 5, 10$ and the zero initial condition $V(0) = 0$ the Simulink implementation produces the graphical results presented in Figures 4, 5, 6, and 7.



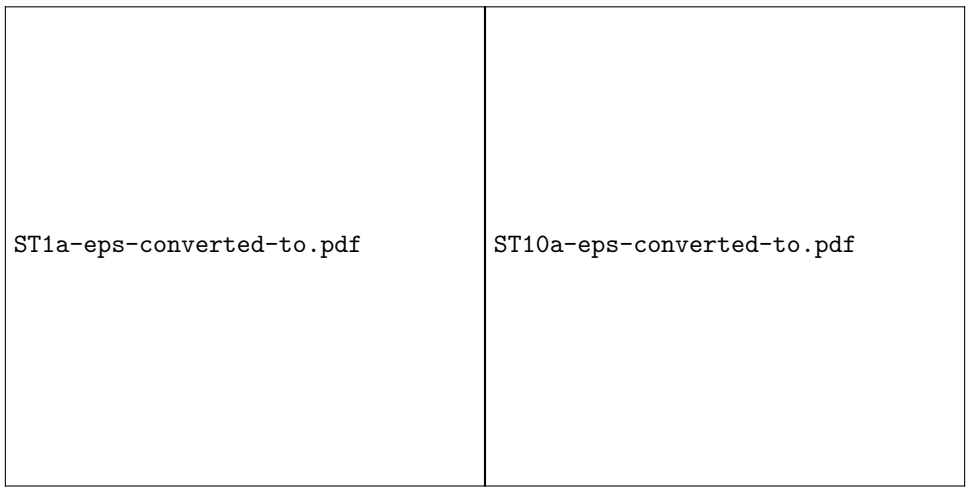
(a) $\lambda = 1$ (b) $\lambda = 5$ (c) $\lambda = 10$

Figure 4: Error comparison of $\Omega_m(u)$ with Li and tunable activation functions.



(a) $\lambda = 1$ (b) $\lambda = 5$ (c) $\lambda = 10$

Figure 5: Error comparison of $\Omega_m(u)$ with VAF and MsbpAF AFs.



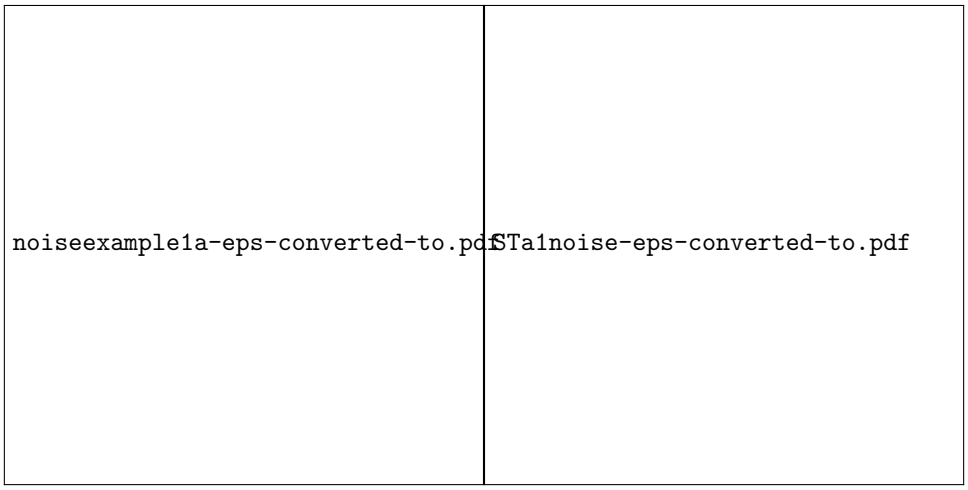
(a) $\lambda = 1$ (b) $\lambda = 10$

Figure 6: State trajectories under Ω_5 activation function.

Specifically, Fig.4 and Fig.5 represent the values of the norm $\|E_L(t)\|_F = \|P_1(t)V(t)P_2(t) - P_3(t)\|$ for different activation functions and various parameters in Ω_m. Here, in Fig.4 and Fig.5, it is observable that the

proposed activation for selected values r_k and w_k initiates faster convergence against the *sbp* activation function (2.1), tunable activation function (2.4) and VAF (2.5).

State trajectories of the state variables matrix $V(t)$ of dynamics (2.11) generated by Ω_5 are presented in Fig.6. From Fig.6 it can be observed that the solution trajectories produced by the RNTZNN formula (2.11) (red dashed lines) converge quickly to the black lines of the theoretical solution. Graphs generated with various noise perturbed models under the noise $N(t) = -\lambda E(t)|\cos(t)|$ and $\lambda = 1$ are arranged in Fig.7. In Fig.7 it is observable that under the noise $N(t) = -\lambda E(t)|\cos(t)|$ the proposed activation function initiates faster convergence rate against previous activation functions even in the noise situation.



(a) (b)

Figure 7: Results under the noise $N(t) = -\lambda E(t)|\cos(t)|$ and $\lambda = 1$: (a) Error comparison; (b) State trajectories corresponding to $\Omega_6(u)$.

Example 4.2. Consider the matrices as in the example from [57] with entries

$$P_1(t) = \begin{bmatrix} \cos(t) & \sin(t) \\ -\sin(t) & \cos(t) \end{bmatrix}, P_2(t) = \begin{bmatrix} 2 & \cos(t)+1 & \sin(t) \\ \sin(t) & 2 & \cos(t) \\ \cos(t) & \sin(t)+1 & 2 \end{bmatrix},$$

$$P_3(t) = \begin{bmatrix} 2+\sin(t)\cos(t) & \cos(t)+\sin^2(t)+\sin(t)+1 & 3\sin(t) \\ \cos^2(t)-\sin(t) & 0.5\sin(2t)+\cos(t)-2 & \cos(t) \end{bmatrix}.$$

The theoretical solution of the corresponding linear equation is

$$V^*(t) = \begin{bmatrix} \cos(t) & \sin(t) & 0 \\ \sin(t) & -\cos(t) & 1 \end{bmatrix}.$$

Graphical results generated in the simulink implementation under the parameters $r_1 = 0.3, r_2 = 0.4, r_3 = 0.5, r_4 = 0.6, r_5 = 0.7, r_6 = 0.8, r_7 = 0.9, r_8 = 0.9, w_1 = 3, w_2 = 4, w_3 = 5, w_4 = 6, w_5 = w_6 = w_7 = w_8 = 7$, $\lambda = 1, 5, 10$ and the zero initial condition $V(0) = 0$ are illustrated in Fig.8, Fig.9 and Fig.10.

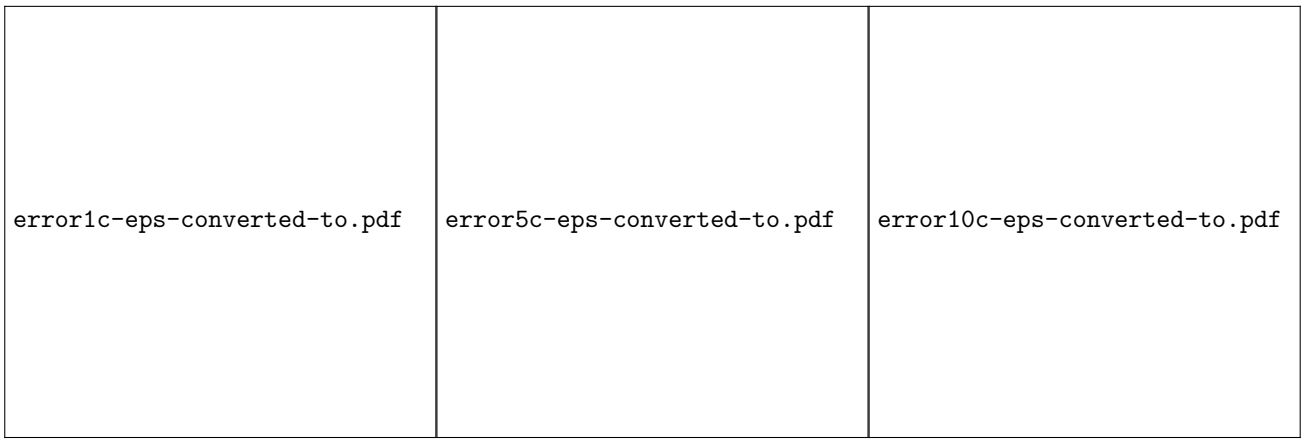
Specifically, From Fig.8 it is noticeable that Ω_m (2.7) initiates faster convergence compared to the Li, tunable, VAF, and modified sign bi power activation function (MsbpAF) (2.6). According to graphs included in Fig.8, it is visible that Ω_m vanishes to zero the norm of the error matrix $\|E_L(t)\|_F = \|P_1(t)V(t)P_2(t) - P_3(t)\|_F$ in a shorter time than other activation functions, given in (2.1), (2.4) and (2.6).

The red dashed lines of the trajectories produced by Simulink implementation Fig.3 and included in Fig.9 are in agreement with black lines of theoretical solution for $\lambda = 1, 10$.

It is interesting to inspect the behavior of the proposed model (2.11) for the non-zero initial state

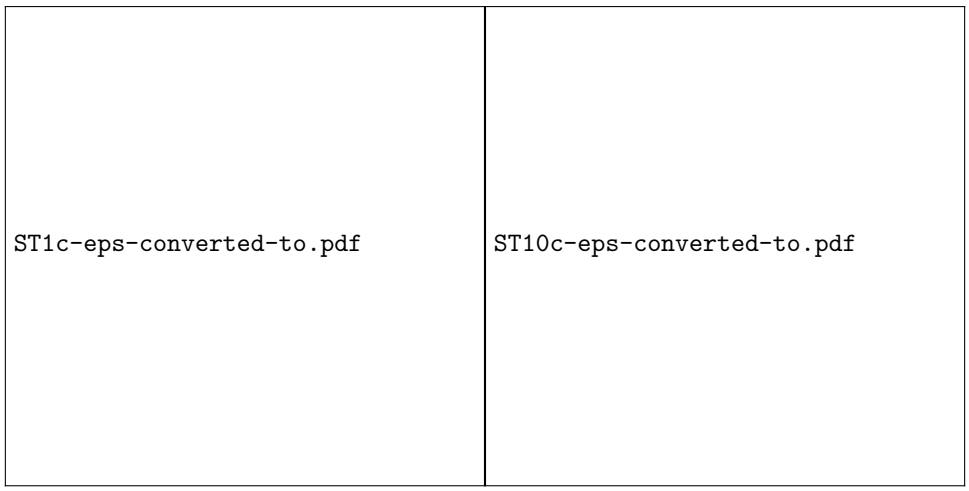
$$V_a(0) = \begin{bmatrix} -1 & -1 & -1 \\ -1 & -1 & -1 \end{bmatrix}.$$

With the same constants as in previous case, Fig.10 strengthens the fact that the proposed formula can work under non-zero initial conditions and confirms the fact that the residual error vanish faster to zero as the parameter m increases. Moreover, the green dashed line corresponding to $\Omega_8(u)$ is in excellent agreement with the theoretical solution (black line) for $\lambda = 1$.



(a) $\lambda = 1$ (b) $\lambda = 5$ (c) $\lambda = 10$

Figure 8: Error comparison of $\Omega_m(u)$ with previous activation functions.



(a) $\lambda = 1$ (b) $\lambda = 10$

Figure 9: State trajectories from $\Omega_8(u)$.

(a) (b)

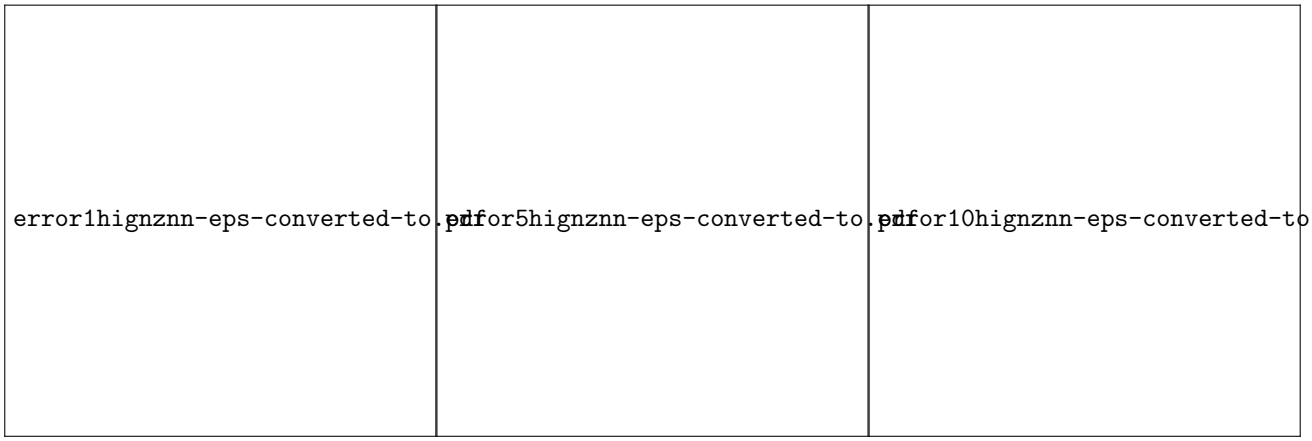
Figure 10: (a) the state trajectories under $\Omega_8(u)$; (b) residual error for $m = 6, 7, 8$ and $\lambda = 1$.

From this example we can understand the advantage of the usage of the Ω_m activation (2.7) against previous activation functions.

Example 4.3. In [59] the authors designed the high-order ZNN models for linear linear time-variant matrix equations $P_1(t)V(t) = P_3(t)$, where

$$P_1(t) = \begin{bmatrix} t+1 & t & t \\ t & t-1 & t \\ t & t & t+1 \end{bmatrix}, \quad P_3(t) = \begin{bmatrix} 0 & t+1 & 3t \\ t-1 & 0 & t \\ 2t+1 & t & 0 \end{bmatrix}. \tag{4.1}$$

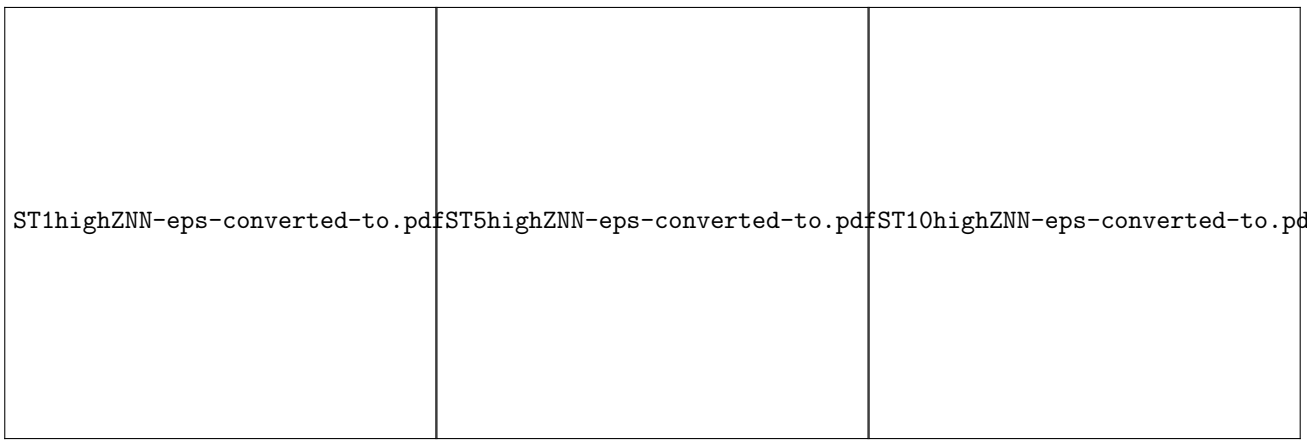
It is challenging to examine the behaviour of the proposed activation function (2.7) in the case of solving the linear matrix equation $P_1(t)V(t)P_2(t) = P_3(t)$, where $P_1(t), P_3(t)$ are the polynomial matrices defined in (4.1) and $P_2(t) = I_{3\times 3}$. For $b_1 = b_2 = b_3 = b_4 = b_5 = 1$, $r_1 = 0.3, r_2 = 0.4, r_3 = 0.5, r_4 = 0.6, r_5 = 0.7, r_6 = 0.8, r_7 = 0.9, r_8 = 0.95$, $w_1 = w_2 = w_3 = w_4 = w_5 = w_6 = w_7 = w_8 = 1.1$, $\lambda = 1, 5, 10$ under the zero initial condition. A comparison of $\Omega_m(u)$ with previous activation functions under various values λ on the Frobenius norm of the error function are presented in Fig.11.



(a) $\lambda = 1$ (b) $\lambda = 5$ (c) $\lambda = 10$

Figure 11: Error comparison of $\Omega_m(u)$ with previous activation functions.

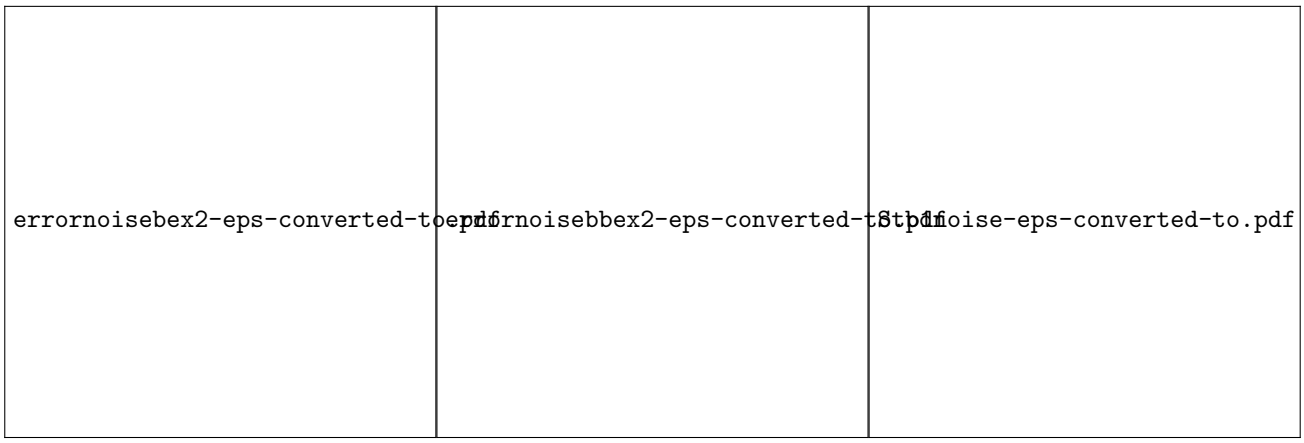
Graphs included in Fig.11 reveal that the novel activation function Ω_m defined in (2.7) initiates faster convergence against the VAF and MsbpAF for all values m. Also, greater values m initiate faster convergence. State trajectories initiated by $\Omega_6(u)$ for various λ are presented in Fig12. Fig12 confirms that the trajectories



(a) $\lambda = 1$ (b) $\lambda = 5$ (c) $\lambda = 10$

Figure 12: State trajectories initiated by $\Omega_6(u)$.

produced by the usage of Ω_6 convergs to theoretical solution in a small time instant. It must be mentioned that for $k = 1$ the residual error $||P_1(t)V(t) - P_3(t)||_F$ from RNTZNN formula converges to 0 before $t = 1$ sec,



(a) (b) (c)

Figure 13: Study under the noise $N(t) = -\lambda E(t)\exp(t)$ for: (a) $\lambda = 1$; (b) $\lambda = 10$; (c) State trajectories for $\lambda = 1$.

while the high order ZNN dynamics's error from [59] converges to 0 after this time. So the importance of Ω_m in solving time-varying linear matrix equation (1.1) is evident.

In Fig.13 it is observable that the proposed activation function implies faster convergence against previous activation functions under the noise $N(t) = -\lambda|E(t)|e^t$. Moreover, the greater values m the faster the convergence speed.

Example 4.4. Observe the following matrix from [60, 61]:

$$S_5(t) = \begin{bmatrix} t+1 & t & t & t & t+1 \\ t & t-1 & t & t & t \\ t & t & t+1 & t & t \\ t & t & t & t-1 & t \\ t+1 & t & t & t & t+1 \end{bmatrix}.$$

In order to compute its Moore-Penrose inverse $S_5(t)^\dagger$, it suffices to apply considered dynamical systems for solving the matrix equation (1.1) with $P_1(t) = S_5^{\mathrm{T}}(t)S_5(t) + \epsilon I_5, P_2(t) = I_5, P_3(t) = S_5^{\mathrm{T}}(t)$, where ϵ is a positive regularization constant near to zero and I_5 represents 5×5 identity matrix. The exact Moore-Penrose inverse of $S_5(t)$ is

$$S_5(t)^\dagger = \begin{bmatrix} \frac{1-t}{4} & \frac{t}{2} & -\frac{t}{2} & \frac{t}{2} & \frac{1-t}{4} \\ \frac{t}{2} & -t-1 & t & -t & \frac{t}{2} \\ -\frac{t}{2} & t & 1-t & t & -\frac{t}{2} \\ \frac{t}{2} & -t & t & -t-1 & \frac{t}{2} \\ \frac{1-t}{4} & \frac{t}{2} & -\frac{t}{2} & \frac{t}{2} & \frac{1-t}{4} \end{bmatrix}.$$

The RNTZNN design (2.11) is applied for $m = 2, 3, 4$, $b_1 = b_2 = b_3 = b_4 = 1$, $r_1 = 0.5, r_2 = 0.6, r_3 = 0.7, r_4 = 0.8$, $w_1 = w_2 = w_3 = w_4 = 2$, $\epsilon = 10^{-7}$ with the starting value

$$V(0) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Fig.14 displays the Frobenius norm $\|P_1(t)V(t)P_2(t) - P_3(t)\|$ of the error matrix in (2.11) under the activation $\Omega_m(u)$ with $m = 5, 6$, and 7.

Fig.14 strengthens the fact that increase of the parameter m in (2.7) accelerates the convergence speed of (2.11). Moreover, in all cases Ω_m initiates faster convergence than MsbpAF. State trajectories of the error matrix $V(t)$ generated inside (2.11) under $\Omega_4(u)$ activation are presented in Fig.15.

Fig.15 confirms that the RNTZNN model can be used for computing the Moore-Penrose of a time-varying matrix.

(a) (b)

Figure 14: Error behavior initiated by $\Omega_m(u)$ activation function with $m = 5, 6$, and 7 under: (a) $\lambda = 1$; (b) $\lambda = 10$.

(a) $\lambda = 1$ (b) $\lambda = 10$

Figure 15: State trajectories by $\Omega_4(u)$ activation function.

4.2 Solution for the Stein equation

For the validation of the proposed dynamical system (2.12), the following matrices $P_1(t), P_2(t), P_3(t)$ are considered from [48]

$$P_1(t) = \begin{bmatrix} 2+\cos(2t) & 2\sin(t) \\ \sin(2t) & 2-\cos(2t) \end{bmatrix}, P_2(t) = \begin{bmatrix} 2+\cos(2t) & \sin(2t) & 0 \\ \sin(2t) & 1 & \cos(t) \\ \cos(2t) & \sin(2t) & 2 \end{bmatrix},$$

$$P_3(t) = \begin{bmatrix} -2+\cos(2t) & \sin(2t)+1 & \cos(2t) \\ \sin(2t) & -2\sin(2t) & 1+\cos(2t) \end{bmatrix}.$$

The Simulink implementation of (2.12) is presented in Fig.16 and it can be used for solving the Stein equation.

For the data set $r = 0.5, w = 2, r_1 = 0.1, r_2 = 0.2, r_3 = 0.3, r_4 = 0.4, r_5 = 0.6, r_6 = 0.6, w_1 = w_2 = w_3 = w_4 = w_5 = w_6 = 2$ and under the zero initial condition $V(0) = 0$ the Simulink implementation from Fig.16 gives the results presented in Fig.17, Fig.18, Fig.19, Fig.20 and Fig.21. Comparison of the error norm $\|P_1(t)V(t)P_2(t) - P_3(t)\}_F$ corresponding to the activation $\Omega_m(u)$ and previously proposed activation functions for the Stein matrix equation solving is presented in Fig.17.

Fig.17 strengthens the fact that the proposed activation function family implies faster convergence speed against VAF (2.5) and MsbpAF (2.6) because of the residual norm $\|E_S(t)\|_F = \|P_1(t)V(t)P_2(t)+V(t)-P_3(t)\|_F$

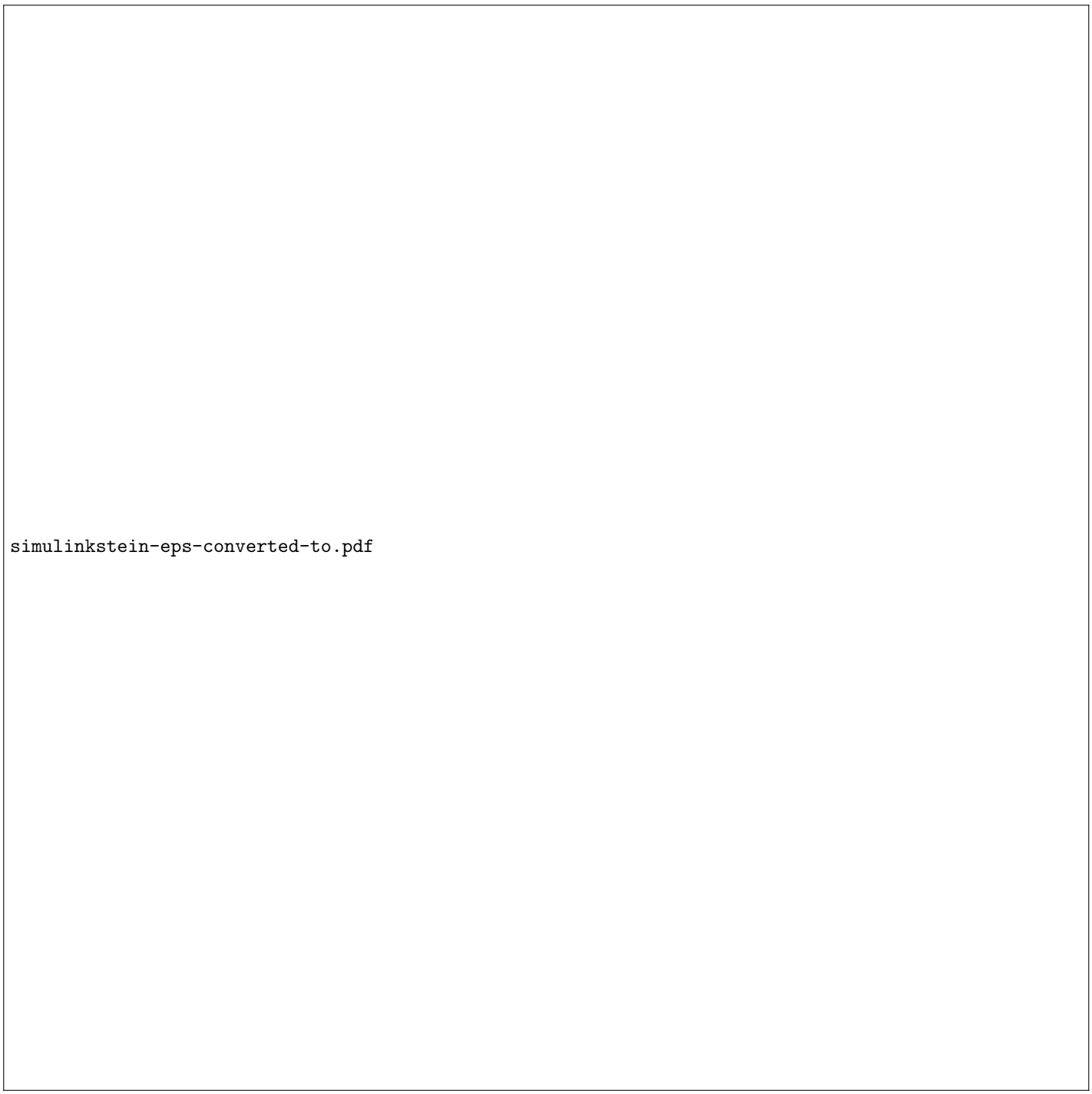
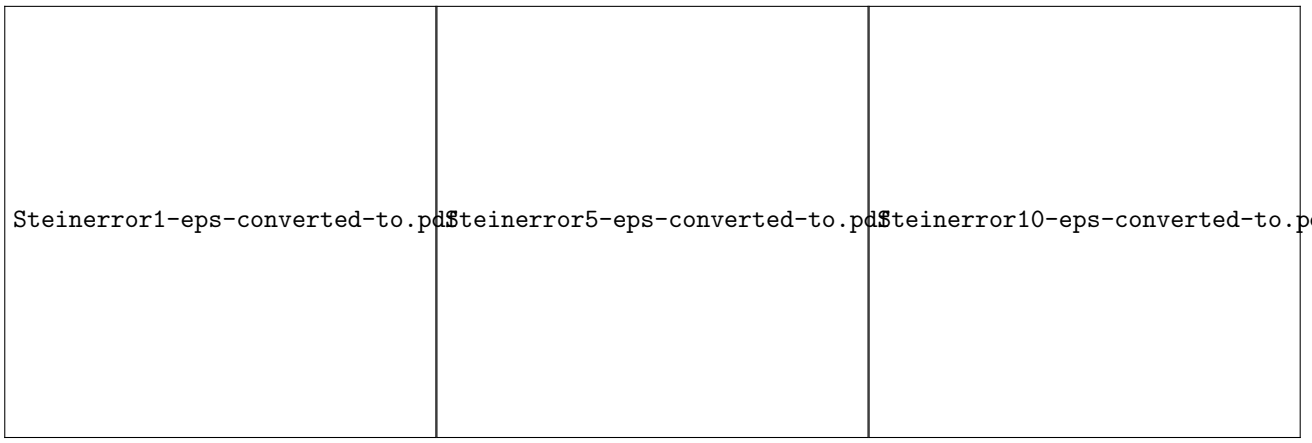


Figure 16: Simulink implementation of (2.12) for Stein equation solving.

vanishes to zero in a shorter time interval.

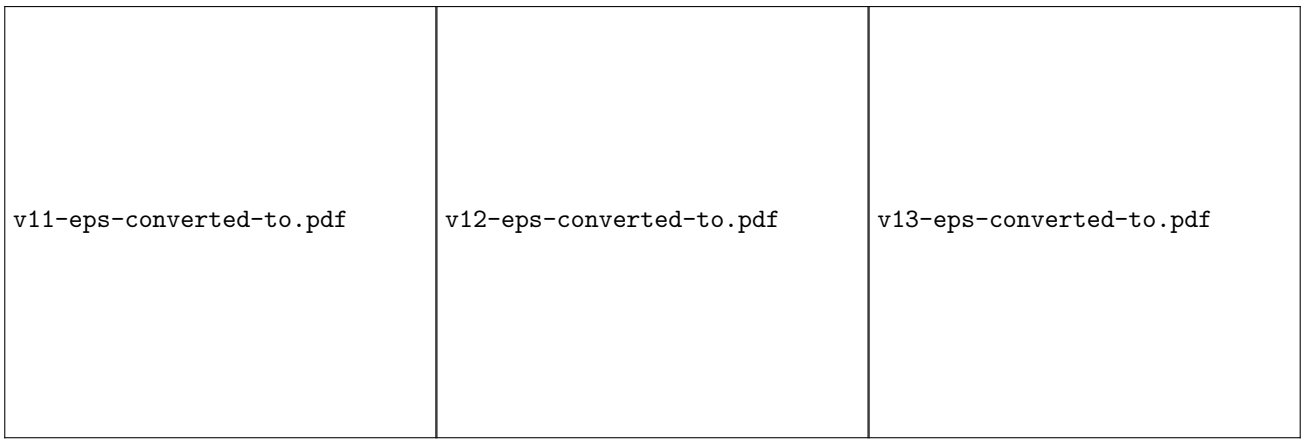
In addition, state trajectories produced by Ω_6(u) for $\lambda = 1$ in Fig.18 and Fig.19 (blue lines) are in excellent agreement with the theoretical solution from [48]. If the initial state is changed from zero to $V(0) = I_{2\times 3}$ it is observable from Fig.20 that the new method with the activation Ω_m is efficient for different initial states and initiates faster convergence gainst previous VAF and MsbpAF.

It is interesting to examine the behaviour of the implicit dynamics (2.12) with the presence of a noise in the form $N(t) = -\lambda E(t)|h(t)|$. For the purposes of this numerical experiment the time-varying function $h(t) = 5\exp(\sin(t))$ inside the noise $N(t)$ is considered. It is noticeable that Ω_m, defined in (2.7), can work and under the noise situation with better results than previous activation function VAF (2.5) and MsbpAF (2.6) respectively.



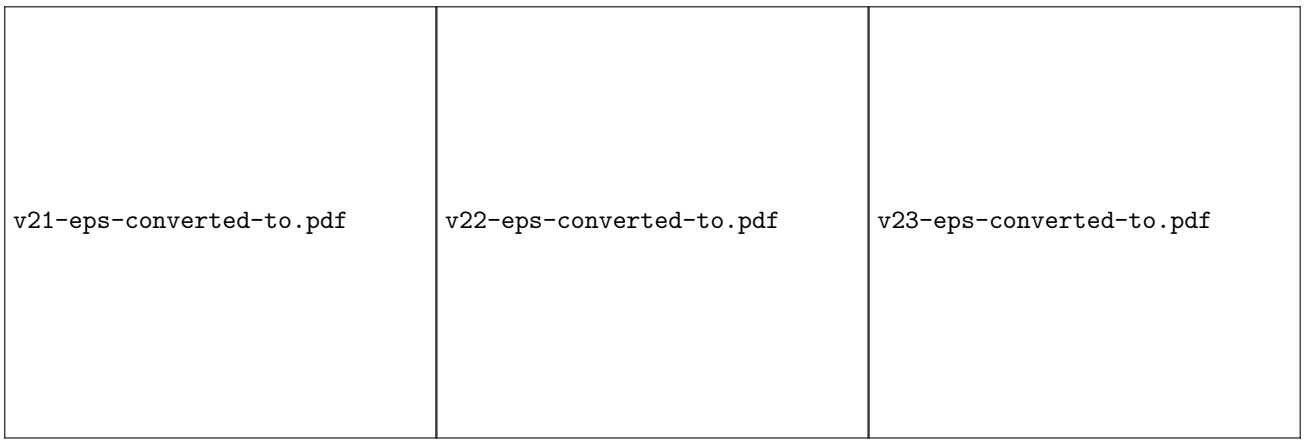
(a) $\lambda = 1$ (b) $\lambda = 5$ (c) $\lambda = 10$

Figure 17: Error comparison of $\Omega_m(u)$ with previous activation functions for the Stein matrix equation solving.



(a) $v_{11}(t)$ (b) $v_{12}(t)$ (c) $v_{13}(t)$

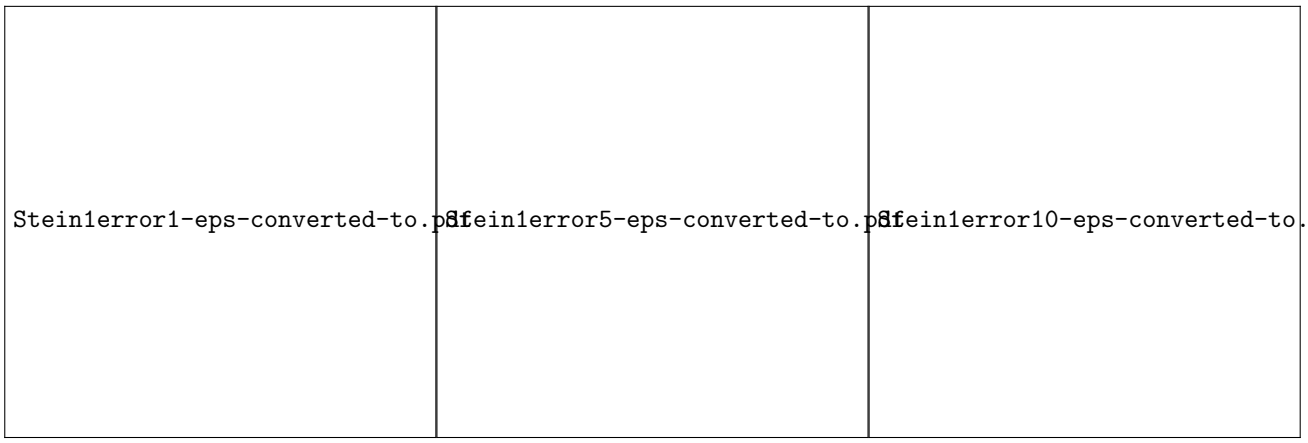
Figure 18: (a) $v_{11}(t)$ (b) $v_{12}(t)$ (c) $v_{13}(t)$ under $\Omega_6(u)$ under the zero initial condition and $\lambda = 1$.



(a) $v_{21}(t)$ (b) $v_{22}(t)$ (c) $v_{23}(t)$

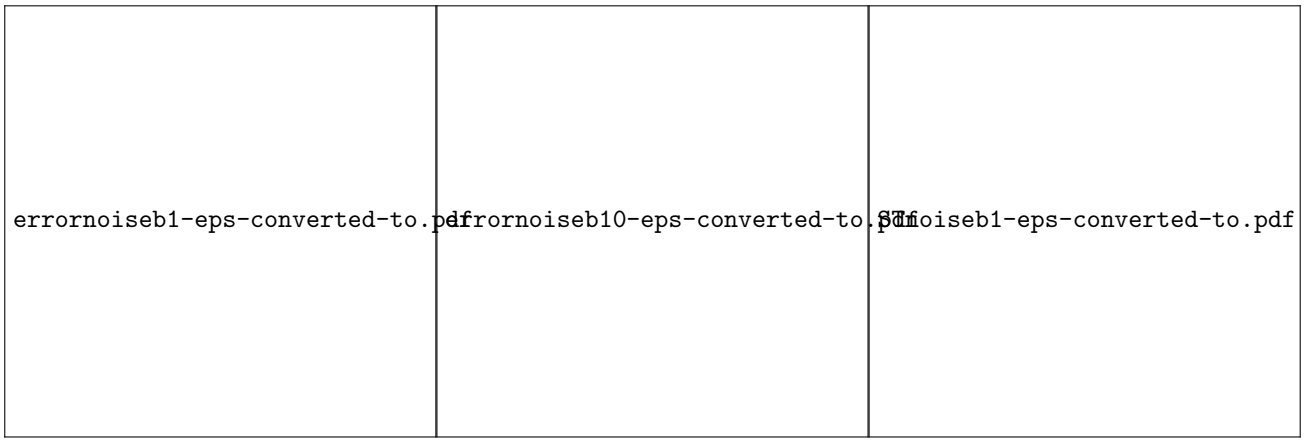
Figure 19: (a) $v_{21}(t)$; (b) $v_{22}(t)$; (c) $v_{23}(t)$ under $\Omega_6(u)$ under the zero initial condition and $\lambda = 1$.

This numerical example shows applicability of the proposed Ω_m activation function in solving linear matrix equation (1.1) and the Stein matrix equation as as well as in solving related matrix equations.



(a) $\lambda = 1$ (b) $\lambda = 5$ (c) $\lambda = 10$

Figure 20: Error comparison of $\Omega_m(u)$ with previous activation functions for Stein matrix equation solving under the initial condition $V(0) = I$.



(a) (b) (c)

Figure 21: Study under the noise $n(t) = -5\lambda e_{ij}(t)\exp(\sin(t))$ for: (a) $\lambda = 1$; (b) $\lambda = 10$; (c) State trajectories for $\lambda = 1$.

5 Conclusion

We introduced a robust noise-tolerant zeroing neural network to solve the time-varying linear matrix equations. It is verified theoretically and numerically that the neural state matrix $V(t)$ converges to the theoretical solution of (1.1) in a predefined convergence time. The convergence rate can increase effectively by increasing the number of the parameters $b_1, b_2, b_3, b_4, b_5, m$ used in the novel activation function (2.7). Numerical experiments generated by the simulink model in Fig.3, strengthens the fact that the RNTZNN evolution dynamics give better results in finite time estimation, and the extended versatile activation function order m implies faster convergence property against previous VAF and MsbpAF activation functions. Future directions of this manuscript can be considered the solution of more complicated time-varying matrix formulas and the construction of a novel family of activation functions based on $\Omega_m(u)$. Some of possibilities for further research based on the RNTZNN formula include:

- Extension of the proposed RNTZNN in computation of generalized inverses of tensors.
- Application to solving algebraic Riccati equations.
- Application to solving Yang-Baxter matrix equations.
- Computation of generalized tensor eigenvalue problems.
- Application for real-time computation of inverse kinematics of redundant manipulators.
- Tracking of nonlinear mass-spring-damper mechanical system under division-by-zero problem.

- Discretization of RNTZNN formula to improve accuracy for the existing results.

Acknowledgements. Ratikanta Behera is grateful to the Mohapatra Family Foundation and the College of Graduate Studies, University of Central Florida, Orlando, Florida, USA for their financial support for this research.
Predrag Stanimirović acknowledges support from the Ministry of Education, Science and Technological Development, Republic of Serbia, Grant No. 174013/451-03-68/2020-14.

Conflict of interest. The authors declare that they have no conflict of interest.

References

[1] L. Xiao, *A new design formula exploited for accelerating Zhang neural network and its application to time-varying matrix inversion*, Theor. Comput. Sci. **647** (2016), 50–58.

[2] Y. Kong, H. Lu, Y. Xue, and H. Xia, *Terminal neural computing: finite convergence and its applications*, Neurocomputing **217** (2016), 133–141.

[3] L. Xiao, Y. Zhang, K. Li, B. Liao, and Z. Tan, *A novel recurrent neural network and its finite-time solution to time-varying complex matrix inversion*, Neurocomputing **331** (2019), 483–492.

[4] L. Xiao, B. Liao, S. Li, Z. Zhang, L. Ding, and L. Jin, *Design and analysis of FTZNN applied to the real-time solution of a nonstationary Lyapunov equation and tracking control of a wheeled mobile manipulator*, IEEE Transactions on Industrial Informatics **14** (2017), 98–105.

[5] J. Jin, L. Xiao, M. Lu, and J. Li, *Design and analysis of two FTRNN models with application to time-varying Sylvester equation*, IEEE Access **7** (2019), 58945–58950.

[6] D. Gerontitis, L. Moysis, P. Stanimirović, V. N. Katsikis, and C. Volos, *Varying-parameter finite-time zeroing neural network for solving linear algebraic systems*, Electronics Letters **56** (2020), 810–813.

[7] D. Gerontitis, R. Behera, J. K. Sahoo, and P. Stanimirović, *Improved finite-time zeroing neural network for time-varying division*, Studies in Applied Mathematics **146** (2021), 526–549.

[8] P. Stanimirović, D. Gerontitis, P. Tzekis, R. Behera, and J. K. Sahoo, *Simulation of Varying Parameter Recurrent Neural Network with application to matrix inversion*, Mathematics and Computers in Simulation **185** (2021), 614–628.

[9] C. Mo, D. Gerontitis, and P. Stanimirović, *Solving the time-varying tensor square root equation by varying-parameters finite-time Zhang neural network*, Neurocomputing **445** (2021), 309–325.

[10] D. Gerontitis, R. Behera, P. Tzekis, and P. Stanimirović, *A family of varying-parameter finite-time zeroing neural networks for solving time-varying Sylvester equation and its application*, Journal of Computational and Applied Mathematics (2021), 113826.

[11] Y. Zhang, D. Jiang, and J. Wang, *A recurrent neural network for solving Sylvester equation with time-varying coefficients*, IEEE Transactions on Neural Networks **13**, 1053–1063, (2002).

[12] Y. Zhang, and S. S. Ge, *Design and analysis of a general recurrent neural network model for time-varying matrix inversion*, IEEE Transactions on Neural Networks **16**, 1477–1490, (2005).

[13] Y. Zhang, Z. Fan, and Z. Li, *Zhang neural network for online solution of time-varying Sylvester equation*, International Symposium on Intelligence Computation and Applications Springer, Berlin, Heidelberg., 276–285, (2007).

[14] K. Chen, S. Yue, and Y. Zhang, *MATLAB simulation and comparison of Zhang neural network and gradient neural network for online solution of linear time-varying matrix equation* $AXB - C = 0$, International Conference on Intelligent Computing Springer, Berlin, Heidelberg., 68–75, (2008).

[15] Y. Zhang, S. Yue, K. Chen, and C. Yi, *Matlab simulation and comparison of Zhang neural network and gradient neural network for time-varying Lyapunov equation solving*, International Symposium on Neural Networks. Springer, Berlin, Heidelberg., 117–127, (2008).

[16] Y. Zhang, X. Guo, W. Ma, K. Chen, and B. Cai, *MATLAB Simulink modeling and simulation of Zhang neural network for online time-varying matrix inversion*, IEEE International Conference on Networking, Sensing and Control. Springer, Berlin, Heidelberg., 1480–1485, (2008).

[17] Y. Zhang, Z. Chen, K. Chen, and B. Cai, *Zhang neural network without using time-derivative information for constant and time-varying matrix inversion*, 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 142–146, (2008).

[18] Y. Zhang, Y. Yang, and N. Tan, *Time-varying matrix square roots solving via Zhang neural network and gradient neural network: modeling, verification and comparison*, International Symposium on Neural Networks, Springer, Berlin, Heidelberg, 11–20, (2009).

[19] Y. Zhang, X. Li, and Z. Li, *Modeling and verification of Zhang neural networks for online solution of time-varying quadratic minimization and programming*, International Symposium on Intelligence Computation and Applications, Berlin, Heidelberg, 101–110, (2009).

[20] Y. Zhang, W. Ma, and B. Cai, *From Zhang neural network to Newton iteration for matrix inversion*, IEEE Transactions on Circuits and Systems I: Regular Papers **56**, 1405–1415, (2008).

[21] N. L. Tsitsas, D. I. Kaklamani, and N. K. Uzunoglu, *Rigorous integral equation analysis of nonsymmetric coupled grating slab waveguides*, Journal of the Optical Society of America A **23**, 2888–2905, (2006).

[22] N. L. Tsitsas, D. I. Kaklamani, and N. K. Uzunoglu, *Integral equation analysis of coupling in symmetric grating assisted optical waveguides*, Journal of the Optical Society of America A **23**, 1488–1502, (2006).

[23] N. L. Tsitsas, and N. K. Uzunoglu, *Scattering by a grating slab waveguide with regular plane regions grooves: Integral Equation Modeling*, IEEE Transactions on Magnetics **45**, 1080–1083, (2009).

[24] D. K. Gerontitis, and N. L. Tsitsas, *Scattering by an all-dielectric metasurface including a periodic arrangement of arbitrary scatterers*, IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS), 1–4, (2019).

[25] B. G. Greenberg, and A. E. Sarhan, *Matrix inversion, its interest and application in analysis of data*, Journal of the American Statistical Association **54**, 755–766, (1959).

[26] M. Wu, B. Yin, A. Vosoughi, C. Studer, J. R. Cavallaro, and C. Dick, *Approximate matrix inversion for high-throughput data detection in the large-scale MIMO uplink*, IEEE international symposium on circuits and systems (ISCAS) 2155–2158, (2013).

[27] R. H. Sturges, *Analog matrix inversion (robot kinematics)*, IEEE Journal of Robotics and Automation **4**, 157–162, (1988).

[28] Y. Zhang, *Towards piecewise-linear primal neural networks for optimization and redundant robotics*, Proceedings of IEEE International Conference on Networking, Sensing and Control, 374–379, (2006).

[29] Z. Zhang, X. Deng, L. Kong, and S. Li, *A circadian rhythms learning network for resisting cognitive periodic noises of time-varying dynamic system and applications to robots*,IEEE Transactions on Cognitive and Developmental Systems **12**, 575–587, (2019).

[30] Z. Zhang, S. Yang, Y. Chen, Y. Luo, H. Yang, and Y. Liu, *A vector-based constrained obstacle avoidance scheme for wheeled mobile redundant robot manipulator*, IEEE Transactions on Cognitive and Developmental Systems **13**, 465–474, (2020).

[31] Z. Li, W. Yuan, S. Zhao, Z. Yu, Y. Kang, and C. L. P. Chen, *Brain-actuated control of dual-arm robot manipulation with relative motion*, IEEE Transactions on Cognitive and Developmental Systems **11**, 51–62, (2019).

[32] J. Li, Z. Li, X. Li, Y. Feng, Y. Hu, and B. Xu, *Skill learning strategy based on dynamic motion primitives for human–robot cooperative manipulation*, IEEE Transactions on Cognitive and Developmental Systems **13**, 105–117, (2020).

[33] Z. Li, and S. Li, *Kinematic control of manipulator with remote center of motion constraints synthesised by a simplified recurrent neural network*, Neural Processing Letters, 1–20, (2021).

[34] S. Li, S. Chen, and 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 Processing Letters **37**, 189–205, (2013).

[35] D. Guo, and Y. Zhang, *Li-function activated ZNN with finite-time convergence applied to redundant-manipulator kinematic control via time-varying Jacobian matrix pseudoinversion*, Applied Soft Computing **24**, 158–168, (2014).

[36] S. Li, and Y. Li, *Nonlinearly activated neural network for solving time-varying complex Sylvester equation*, IEEE Transactions on Systems, Man, and Cybernetics **44**, 1397–1407, (2013).

[37] L. Xiao, K. Li, Z. Tan, Z. Zhang, B. Liao, K. Chen, L. Jin and S. Li, *Nonlinear gradient neural network for solving system of linear equations*, Information Processing Letters **142**, 35–40, (2019).

[38] L. Xiao, S. Li, K. Li, L. Jin, and B. Liao, *Co-design of finite-time convergence and noise suppression: a unified neural model for time varying linear equations with robotic applications*, IEEE Transactions on Systems, Man, and Cybernetics: Systems **50**, 5233–5243, (2018).

[39] L. Xiao, Z. Zhang, Z. Zhang, W. Li, and S. Li, *Design, verification and robotic application of a novel recurrent neural network for computing dynamic Sylvester equation*, Neural Networks **105**, 185–196, (2018).

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

[41] L. Xiao, Q. Yi, J. Dai, K. Li, and Z. Hu, *Design and analysis of new complex zeroing neural network for a set of dynamic complex linear equations*, Neurocomputing **363**, 171–181, (2019).

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

[43] L. Ding, L. Xiao, K. Zhou, Y. Lan, and Y. Zhang, *Two novel finite time convergent recurrent neural networks for tackling complex-valued systems of linear equation*, Filomat **34**, 5009–5018, (2020).

[44] Z. Jian, L. Xiao, J. Dai, Z. Tang, and C. Liu, *Design and analysis of new zeroing neural network models with improved finite-time convergence for time-varying reciprocal of complex matrix*, IEEE Transactions on Industrial Informatics **16**, 3838–3848, (2019).

[45] L. Xiao, Y. Zhang, J. Dai, Q. Zuo, and S. Wang, *Comprehensive analysis of a new varying parameter zeroing neural network for time varying matrix inversion*, IEEE Transactions on Industrial Informatics **17**, 1604–1613, (2020).

[46] L. Xiao, J. Tao, J. Dai, Y. Wang, L. Jia, and Y. He, *A Parameter-changing and complex-valued zeroing neural-network for finding solution of time-varying complex linear matrix equations in finite-time*, IEEE Transactions on Industrial Informatics **17**, 6634–6643, (2021).

[47] L. Xiao, Y. Zhang, J. Dai, K. Chen, S. Yang, W. Li, B. Liao, L. Ding, and J. Li, *A new noise-tolerant and predefined-time ZNN model for time-dependent matrix inversion*, Neural Networks **117**, 124–134, (2019).

[48] J. Dai, L. Jia, and L. Xiao, *Design and analysis of two prescribed-time and robust ZNN models with application to time-variant Stein matrix equation*, IEEE Transactions on Neural Networks and Learning Systems **32**, 1668–1677, (2020).

[49] H. Zhang, and L. Wan, *Zeroing neural network methods for solving the Yang-Baxter-like matrix equation*, Neurocomputing **383**, 1409–418, (2020).

[50] J. Gong, and J. Jin, *A better robustness and fast convergence zeroing neural network for solving dynamic nonlinear equations*, Neural Computing and Applications, 1–11, (2021).

[51] L. Xiao, Y. Zhang, J. Dai, J. Li, and W. Li, *New noise-tolerant ZNN models with predefined-time convergence for time-variant Sylvester equation solving*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 3629–3640 (2019).

[52] J. Dai, Y. Li, L. Xiao, L. Jia, Q. Liao, and J. Li, *Comprehensive study on complex-valued ZNN models activated by novel nonlinear functions for dynamic complex linear equations*, Information Sciences **561**, 101–114 (2021).

[53] A. Polyakov, *Nonlinear feedback design for fixed-time stabilization of linear control systems*, IEEE Transactions on Automatic Control **57**, 2106–2110 (2011).

[54] C. Aouiti, and F. Miaadi, *A new fixed-time stabilization approach for neural networks with time-varying delays*, Neural Computing and Applications **32**, 3295–3309 (2020).

[55] J. Jin, and J. Gong, *A noise-tolerant fast convergence ZNN for dynamic matrix inversion*, International Journal of Computer Mathematics **98**, 2202–2219 (2021).

[56] J. Dai, X. Yang, L. Xiao, L. Jia, and Y. Li, *ZNN with fuzzy adaptive activation functions and Its application to time-varying linear matrix equation*, IEEE Transactions on Industrial Informatics, (2021).

[57] Z. Zhang, X. Deng, X. Qu, B. Liao, L. D. Kong, and L. Li, *A varying-gain recurrent neural network and its application to solving online time-varying matrix equation*, IEEE Access **6**, 77940–77952, (2018).

[58] J. Dai, Y. Chen, L. Xiao, L. Jia, and Y. He, *Design and analysis of a hybrid GNN-ZNN model with a fuzzy adaptive factor for matrix inversion*, IEEE Transactions on Industrial Informatics, (2021).

[59] L. Xiao, H. Tan, J. Dai, L. Jia, and W. Tang, *High-order error function designs to compute time-varying linear matrix equations*, Information Sciences **576** 173–186, (2021).

[60] P. Stanimirović, V. N. Katsikis, and D. Gerontitis, *A new varying-parameter design formula for Solving time-varying problems*, Neural Processing Letters **53**, 107–129, (2021).

[61] G. Zielke, *Report on test matrices for generalized inverses*, Computing **36**, 105–162, (1986).

[62] K. Jbilou, and A. Messaoudi, *A computational method for symmetric Stein matrix equations*, Numerical Linear Algebra in Signals, Systems and Control, Springer, Dordrecht, 295–311, (2011).

[63] C. Y. Chiang, *A note on the $\top$-Stein matrix equation*, Abstract and Applied Analysis, Hindawi, (2013).

[64] G. Chen, X. Zeng, and Y. Hong, *Distributed optimisation design for solving the Stein equation with constraints*, IET Control Theory & Applications **13**, 2492–2499, (2019).

[65] Y. Chai, J. Feng, C. Xu, and S. Qin, *Distributed network flow to solve constrained linear matrix equation*, In 13th International Conference on Advanced Computational Intelligence (ICACI), IEEE, 146–153, (2021).