FACTA UNIVERSITATIS (NIŠ)
Ser. Math. Inform. Vol. 32, No 5 (2017), 799–809
https://doi.org/10.22190/FUMI1705799Z

MATLAB SIMULATION OF THE HYBRID OF RECURSIVE NEURAL DYNAMICS FOR ONLINE MATRIX INVERSION

Ivan S. Živković,* Predrag S. Stanimirović†

Abstract. A novel kind of a hybrid recursive neural implicit dynamics for real-time matrix inversion has been recently proposed and investigated. Our goal is to compare the hybrid recursive neural implicit dynamics on the one hand, and conventional explicit neural dynamics on the other hand. Simulation results show that the hybrid model can coincide better with systems in practice and has higher abilities in representing dynamic systems. More importantly, hybrid model can achieve superior convergence performance in comparison with the existing dynamic systems, specifically recently-proposed Zhang dynamics. This paper presents the Simulink model of a hybrid recursive neural implicit dynamics and gives a simulation and comparison to the existing Zhang dynamics for real-time matrix inversion. Simulation results confirm a superior convergence of the hybrid model compared to Zhang model.
Keywords: Zhang neural network; gradient neural network; matrix inverse; convergence.

1. Introduction

Recently, a number of nonlinear and linear recurrent neural network (RNN) models have been developed for the purpose of numerical evaluation of the matrix inverse and the pseudoinverse of full-row or full-column rank rectangular matrices (for more details, see [4, 8, 13, 14, 18]). Various recurrent neural networks for computing generalized inverses of rank-deficient matrices were designed in [15, 16]. A new type of complex-valued recurrent neural networks, called Zhang neural network (ZNN), was proposed in 2001 and has been extensively exploited in solving various time-varying complex generalized inverse problems. The design of complex ZNN

Received November 09, 2017; accepted December 28, 2017
2010 *Mathematics Subject Classification.* Primary 68T05; Secondary 68Q32, 15A09
*Author gratefully acknowledge support from the Research Project III 044006 of the Serbian Ministry of Science
†This author gratefully acknowledge support from the Research Project 174013 of the Serbian Ministry of Science

models arises from the choice of a complex matrix-valued error-monitoring function, called the Zhang function (ZF). Computation of the Moore-Penrose inverse of time-varying full-rank matrix by means of different ZNN models were investigated in [17]. Liao and Zhang in [7] proposed five different complex ZFs and, accordingly developed and investigated five different complex ZNN models for computing the time-varying complex pseudoinverse. A feedforward neural network architecture for computing the Drazin inverse A^{D} was proposed in [3]. RNN for computing the Drazin inverse of a real square matrix in real time was presented in [11]. The dynamical equation and associated artificial RNN for computing the Drazin inverse of an arbitrary square real matrix, without any restriction on its eigenvalues, were proposed in [10]. In the recent paper [9], the authors proposed two finite-time convergent complex ZNN models with for computing the Drazin inverse of a complex time-varying square matrix.

A number of problems and applications in the fields of science and engineering concern online inverse of matrix. A number of zeroing neural networks has been proposed for solving various matrix equations. See [5, 6] and included references for more details.

This paper presents a Matlab Simulink model for the implementation of the hybrid recurrent neural networks for computing inverse of a nonsingular matrix which was proposed in [1]. In addition, we present a comparisons of numerical results derived by this implementation and the Simulink implementation of the classical ZNN model for online matrix inversion of a given time invariant matrix, introduced in [13].

The following defining equation of matrix inverse $A^{-1} \in \mathbb{R}^{n\times n}$ can be given:

$$AX(t) - I = 0 \tag{1.1}$$

or

$$X(t)A - I = 0, \tag{1.2}$$

where $I \in \mathbb{R}^{n\times n}$ is the identity matrix, and $X(t) \in \mathbb{R}^{n\times n}$ denotes the unknown matrix to be inverted which corresponds to the theoretical inverse A^{-1}.

The rest of the paper is organized as follows. Some known results related to the gradient-based dynamics and implicit Zhang dynamics as well as improved hybrid model are restated in Section 2. In Section 3. we present the corresponding MATLAB Simulink model of improved hybrid dynamics along with simulation examples and comparison.

2. Model formulation

We assume invertibility of the input matrix A. Then the equation (1.1) (or (1.2) in dual case) has a unique solution and the minimal eigenvalue of $A^{\mathrm{T}}A$ is greater than 0.

2.1. Gradient-based dynamics

The dynamics of the gradient neural network (GNN) models for computing inverses are based on the usage of the scalar-valued norm-based error function

$$\varepsilon(t) = \varepsilon(X(t)) = \frac{1}{2}\|E(t)\|_F^2, \tag{2.1}$$

where $E(t)$ is an appropriate error matrix and $\|A\|_F := \sqrt{\mathrm{Tr}\,(A^{\mathrm{T}}A)}$ denotes the Frobenius norm of the matrix A and $\mathrm{Tr}\,(\cdot)$ denotes the trace of a matrix. The general design formula is typically defined along the negative gradient $-\partial\varepsilon(X(t))/\partial X$ of $\varepsilon(X(t))$ until the minimum is reached. Using the above negative gradient to construct the neural dynamics, we could have the gradient-based dynamics as follows

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = -\gamma\mathcal{F}\left(\frac{\partial\varepsilon(X(t))}{\partial X}\right). \tag{2.2}$$

The scaling real parameter γ in (2.2) is used to adjust the convergence rate and could be chosen as large as possible in order to accelerate the convergence. Further, $\mathcal{F}(C)$ is an odd and monotonically increasing function array, element-wise applicable to elements of a real matrix $C = (c_{ij}) \in \mathbb{R}^{n\times n}$, i.e. $\mathcal{F}(C) = (f(c_{ij}))$, $i = 1,\ldots,m$, $j = 1,\ldots,n$, wherein $f(\cdot)$ is an odd and monotonically increasing function.

The dynamical equation of the linear recurrent neural network for the inversion of a nonsingular matrix is initiated by the error matrix $E(t) = AX(t) - I$, and it was proposed in [13]:

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = -\gamma A^{\mathrm{T}}\left(AX(t) - I\right). \tag{2.3}$$

The same principle was extended for computing the Moore–Penrose inverse of a full-column rectangular matrix $A \in \mathbb{R}_n^{m\times n}$ or a full-row rectangular matrix $A \in \mathbb{R}_m^{m\times n}$. Wang in [15] showed that the model can be used for computing the Moore-Penrose inverse of rank-deficient matrices under the zero initial condition, $V(0) = 0$.

2.2. Zhang dynamics

On the other hand, the ZNN model for online time-invariant matrix inversion is based upon the matrix-formed error function $E(t)$, instead of a scalar valued function. The time derivative of error function $E(t)$, should be chosen such that each element $e_{ij}(t)$ of $E(t)$ converges to zero, $\forall i = 1, ..., n$. A general design rule of $\dot{E(t)}$ is defined

$$\frac{\mathrm{d}E(t)}{\mathrm{d}t} = -\gamma\mathcal{F}\left(E(t)\right). \tag{2.4}$$

Substituting $E(t)$ into dynamic system (2.4) and choosing $\mathcal{F}$ to bi linear function, the following Zhang dynamics for online matrix inversion can be obtained:

$$A\dot{X} = -\gamma\left(AX(t) - I\right). \tag{2.5}$$

The implicit dynamics were originally proposed for online inversion of a time-varying matrix $A(t)$ in [19]. It was shown in [18, 19, 20] that the Zhang dynamics (2.5) globally exponentially converges to the theoretical inverse A^{-1}, starting from any initial state $X(0)$, with the exponential convergence rate γ.

2.3. Hybrid ZNN model for matrix inversion

A gradient-based recurrent neural dynamics for real-time inverse of a time-invariant matrix was proposed by Wang [13] in the form of an explicit dynamic system

$$\dot{X}(t) = -\gamma A^{\mathrm{T}} \left(AX(t) - I\right). \tag{2.6}$$

The explicit gradient-based recursive dynamics (2.6) can be transformed into an equivalent implicit form after the multiplication with A from the left:

$$A\dot{X}(t) = -\gamma AA^{T}(AX(t) - I). \tag{2.7}$$

Recently, a novel kind of recurrent implicit dynamics for real-time matrix inversion was proposed and investigated in [1, 2]. This hybrid model can be obtained after the summation of both the left and the right hand side of the Zhang dynamics (2.5) and the modified gradient dynamics (2.7). The resulting implicit model is given by

$$A\dot{X}(t) = -\gamma(AA^{\mathrm{T}} + I)(AX(t) - I). \tag{2.8}$$

Global exponential convergence rate of the implicit dynamics (2.8) was investigated in [1, 2].

Theorem 2.1. [1] *Given a nonsingular matrix $A \in \mathbb{R}^{n\times n}$, the state matrix $X(t) \in \mathbb{R}^{n\times n}$ of the model* (2.8) *achieves global exponential convergence to the theoretical inverse $X^* = A^{-1}$ starting from any initial state $X(0) \in \mathbb{R}^{n\times n}$. In addition, the exponential convergence rate is equal to $\gamma\beta$ where $\beta > 1$ denotes the minimum eigenvalue of $A^{\mathrm{T}}A + I$.*

Now we are able to present our motivation in details. The Zhang dynamics (2.5) globally exponentially converges to the theoretical inverse A^{-1}, starting from any initial state $X(0)$, with the exponential convergence rate γ. On the other hand, the model (2.8) is globally exponentially convergent to the theoretical inverse A^{-1} with the exponential convergence rate $\gamma\beta$, where $\beta > 1$ is the minimum eigenvalue of $A^{\mathrm{T}}A + I$. Our goal is to investigate the acceleration caused by the quantity β. Since the hybrid model (2.8) requires some additional matrix multiplications with respect to the standard ZNN model (2.5), our tendency is to discover some classes of matrices which are more appropriate for the application of the hybrid model.

3. Simulation Results and its Comparison

The graphical editor and customizable block libraries available in Matlab Simulink tool are used for simulating and comparing the Zhang dynamic system (2.5) and recently proposed hybrid dunamic system (2.8). Simulink implementation of (2.5) was described in [12]. The models (2.5) and (2.8) will be termed as $ZNNNM$ and $EZNNNM$, respectively.

The Simulink implementation of the hybrid model (2.8) is based on its equivalent form given by

$$\dot{X}(t) = (I - A)\dot{X}(t) - \gamma(AA^{\mathrm{T}} + I)(AX(t) - I). \tag{3.1}$$

and it is presented in Figure 3.1. For solving differential equations in the models we used the *ode15s* solver.

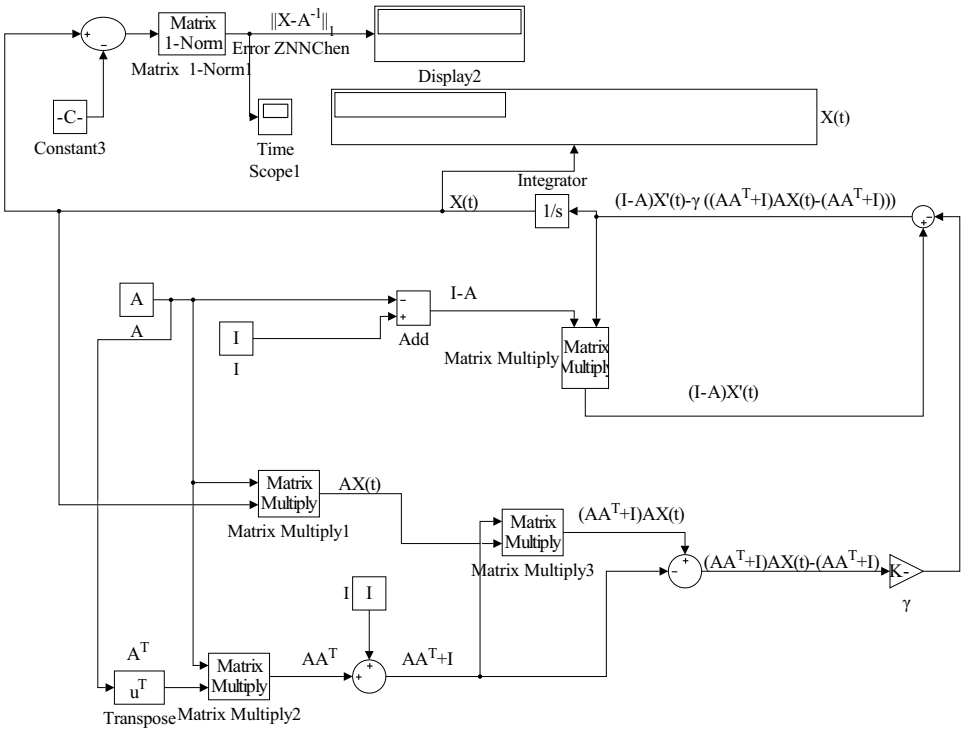

FIG. 3.1: Simulink implementation of EZNNNM model.

The next examples will show performance of both ZNNNM and EZNNNM models.

Example 3.1. **(a)** Consider the following matrix

$$A=\begin{bmatrix}
0.8147 & 0.1576 & 0.6557 & 0.7060 & 0.4387 & 0.2760 & 0.7513 & 0.8407 & 0.3517 & 0.0759\\
0.9058 & 0.9706 & 0.0357 & 0.0318 & 0.3816 & 0.6797 & 0.2551 & 0.2543 & 0.8308 & 0.0540\\
0.1270 & 0.9572 & 0.8491 & 0.2769 & 0.7655 & 0.6551 & 0.5060 & 0.8143 & 0.5853 & 0.5308\\
0.9134 & 0.4854 & 0.9340 & 0.0462 & 0.7952 & 0.1626 & 0.6991 & 0.2435 & 0.5497 & 0.7792\\
0.6324 & 0.8003 & 0.6787 & 0.0971 & 0.1869 & 0.1190 & 0.8909 & 0.9293 & 0.9172 & 0.9340\\
0.0975 & 0.1419 & 0.7577 & 0.8235 & 0.4898 & 0.4984 & 0.9593 & 0.3500 & 0.2858 & 0.1299\\
0.2785 & 0.4218 & 0.7431 & 0.6948 & 0.4456 & 0.9597 & 0.5472 & 0.1966 & 0.7572 & 0.5688\\
0.5469 & 0.9157 & 0.3922 & 0.3171 & 0.6463 & 0.3404 & 0.1386 & 0.2511 & 0.7537 & 0.4694\\
0.9575 & 0.7922 & 0.6555 & 0.9502 & 0.7094 & 0.5853 & 0.1493 & 0.6160 & 0.3804 & 0.0119\\
0.9649 & 0.9595 & 0.1712 & 0.0344 & 0.7547 & 0.2238 & 0.2575 & 0.4733 & 0.5678 & 0.3371
\end{bmatrix}.$$

This matrix has minimum eigenvalue α of $A^{\mathrm{T}}A$: $\alpha = 0.0154$. We compare the linear $ZNNNM$ and $EZNNNM$ model with the gain parameter $\gamma = 10^6$. The initial matrix is chosen by $V(0) = 0$. Figure 3.2 (right) shows the trajectories of the error norms $\|A^{-1} - X(t)\|$ in the total simulation time $t_{tot} = 10^{-5}$. Figure 3.2 (left) shows the trajectories of the error norms $\|A^{-1} - X(t)\|$ in the total simulation time $t_{tot} = 10^{-6}$.

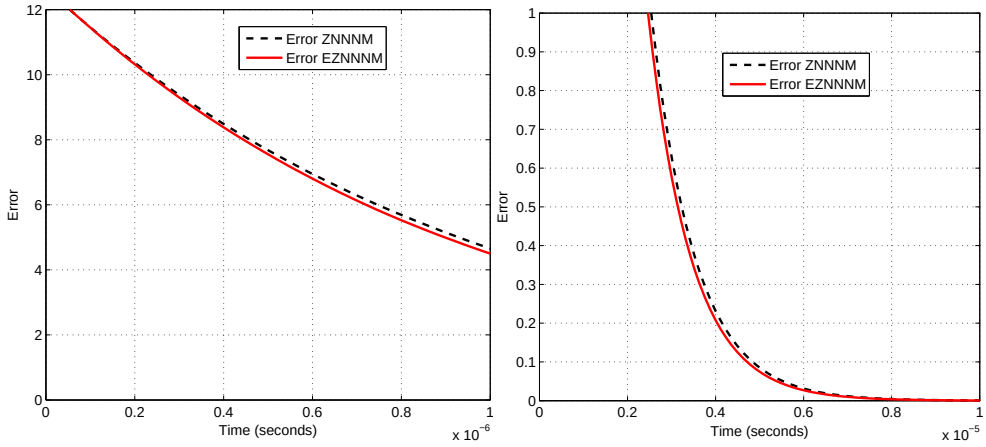

FIG. 3.2: Trajectories of the errors $\|A^{-1} - X(t)\|$ of $ZNNNM$ and $EZNNNM$ in Example 3.1.

In general, Figure 3.2 shows that $EZNNNM$ model slightly outperforms the $ZNNNM$ model. Both models generate almost identical residual norms in the initial phase and then $EZNNNM$ generates a bit smaller residual norms. According to Figure 3.2, the $EZNNNM$ model possesses a bit faster convergence.

(b) Now, consider matrix $A_1 = 5I + A$. This matrix has the smallest eigenvalue $\alpha = 16.6813$, which is quite larger than in the previous example. We apply the linear $EZNNNM$ model with the gain parameter $\gamma = 10^6$.

Figure 3.3 (right) shows the trajectories of the error norm $\|A^{-1} - X(t)\|$ in the total simulation time $t_{tot} = 10^{-5}$. Figure 3.3 (left) shows the trajectories of the error norm $\|A^{-1} - X(t)\|$ in the total simulation time $t_{tot} = 10^{-6}$.

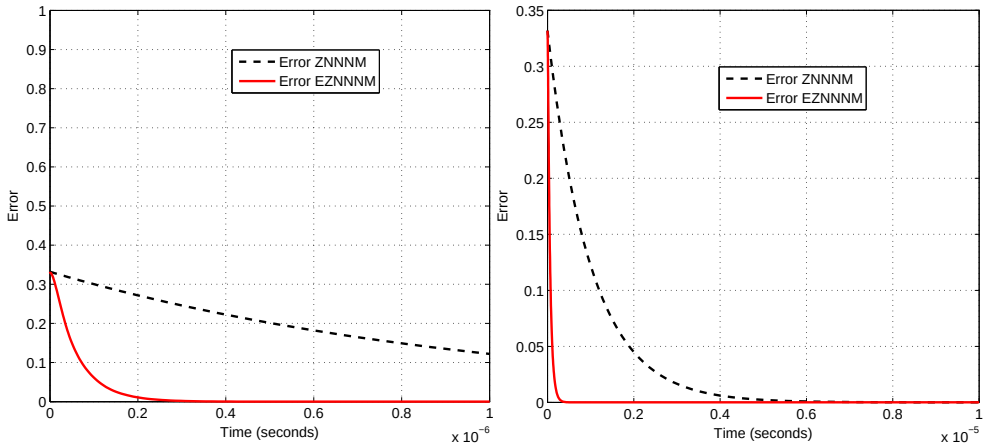

FIG. 3.3: Trajectories of the errors $\|A_1^{-1} - X(t)\|$ of $ZNNNM$ and $EZNNNM$ in Example 3.1.

According to Figure 3.3, the $EZNNNM$ model possesses remarkably faster convergence.

Example 3.2. Consider the matrix

$$A = \begin{bmatrix} 20 & 6 \\ -1 & 30 \end{bmatrix}$$

which satisfies $\alpha = 386.2656$. Elements of the matrix $X(t)$ generated by the model $EZNNNM$ are denoted by x_{ij}^{EZNNNM}. Similarly, elements of the matrix $X(t)$ generated by the model $ZNNNM$ are denoted by x_{ij}^{ZNNNM}. Trajectories of the elements of the matrices are presented in Figure 3.4.

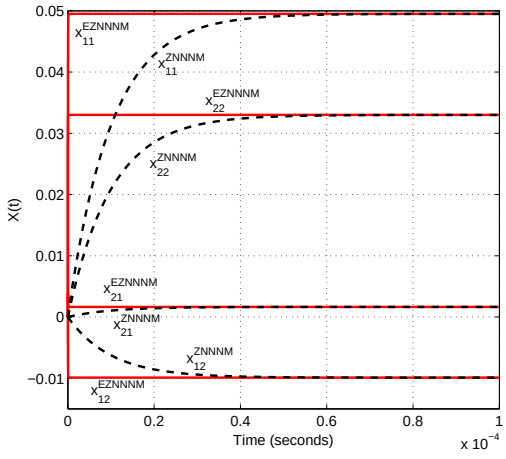

FIG. 3.4: Trajectories of $X(t)$ of $ZNNNM$ and $EZNNNM$ in Example 3.2.

Greater value α initiates significantly faster convergence of $EZNNNM$ with respect to $ZNNNM$.

Example 3.3. This example shows the influence of α on the convergence of the ZNN models. The gain parameters in the simulation is $\gamma = 10^6$ and the time period is $[0, 10^{-5}]$ s. First, consider the following randomly generated matrix B:

$$B = \begin{bmatrix} 0.9631 & 0.6241 & 0.0377 & 0.2619 & 0.1068 \\ 0.5468 & 0.6791 & 0.8852 & 0.3354 & 0.6538 \\ 0.5211 & 0.3955 & 0.9133 & 0.6797 & 0.4942 \\ 0.2316 & 0.3674 & 0.7962 & 0.1366 & 0.7791 \\ 0.4889 & 0.9880 & 0.0987 & 0.7212 & 0.7150 \end{bmatrix}$$

.

Then we are varying matrix A in the way that the value α becomes larger in every step, and testing both models on such matrix A in the order to find the error norm of the model results.

A	α	$\|X^{ZNNNM} - A^{-1}\|_1$	$\|X^{EZNNNM} - A^{-1}\|_1$
B	0.0149	0.000594	0.000494
$B + I$	0.1729	0.000143	2.7866$e-$05
$B + 2I$	1.9360	3.8564$e-$05	2.1079$e-$08
$B + 3I$	5.6776	2.2102$e-$05	1.4172$e-$09
$B + 4I$	11.4133	1.5414$e-$05	3.6347$-$10
$B + 5I$	19.1467	1.1757$-$05	1.5278$-$09
$B + 10I$	87.8023	5.2761$e-$06	1.0706$e-$11
$B + 15I$	206.4530	3.3656$e-$06	2.7424$e-$11
$B + 20I$	375.1024	2.4667$e-$06	5.7302$e-$12
$B + 50I$	2437	9.4123$e-$07	8.6352$e-$14

From the table we can see when the value of α is grater $EZNNNM$ model gives better accuracy of the solution related to $ZNNNM$ model in the same given period of time.

Example 3.4. In this example we ask the answer for the question: is it possible to compensate advantage of the $EZNNNM$ model using greater values γ in $ZNNNM$. For this purpose, we tested these models on the matrix

$$A = \begin{bmatrix} 5.8147 & 0.0975 & 0.1576 & 0.1419 & 0.6557 \\ 0.9058 & 5.2785 & 0.9706 & 0.4218 & 0.0357 \\ 0.1270 & 0.5469 & 5.9572 & 0.9157 & 0.8491 \\ 0.9134 & 0.9575 & 0.4854 & 5.7922 & 0.9340 \\ 0.6324 & 0.9649 & 0.8003 & 0.9595 & 5.6787 \end{bmatrix}$$

with the minimal eigenvalue of AA^{T} equal to 20.8356. The value of the gain parameter γ in Simulink model $\gamma = ZNNNM$ is 10^7 and in $EZNNNM$ is $\gamma = 10^6$. Figure 3.5 (right) shows the trajectories of the error norm $\|A^{-1} - X(t)\|$ in the total simulation time $t_{tot} = 10^{-5}$. Figure 3.5 (left) shows the trajectories of the error norm $\|A^{-1} - X(t)\|$ in the total simulation time $t_{tot} = 10^{-6}$.

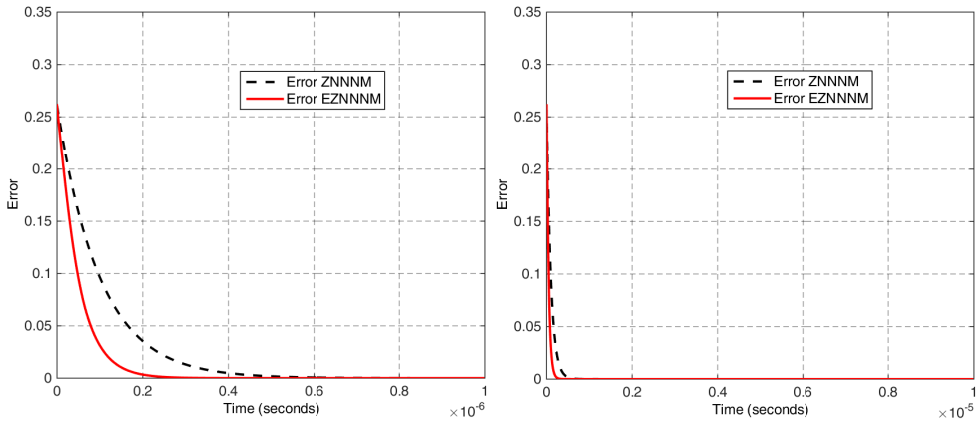

FIG. 3.5: Trajectories of the errors $\|A_1^{-1} - X(t)\|$ of $ZNNNM$ and $EZNNNM$ in Example 3.4.

According to Figure 3.3, the $EZNNNM$ model still possesses faster convergence. This means that greater values γ can not always compensate faster convergence rate of the $EZNNNM$ model with respect to the $ZNNNM$ model.

4. Conclusion

The Matlab Simulink model of a novel implicit dynamic system (2.8) for online matrix inversion is proposed in this paper. Compared to the Zhang implicit dynamic system (2.5), superior global exponential convergence to the theoretical inverse by hybrid implicit dynamic system (2.8) is confirmed and justified by several computer simulation results. Tests shows that with a greater value of α (i.e. $\beta = 1 + \alpha$), faster convergence and better accuracy of the solution can be obtain with the hybrid implicit dynamic system related to the Zhang implicit dynamic system.

REFERENCES

1. K. CHEN, *Recurrent implicit dynamics for online matrix inversion*, Appl. Math. Comput. 219(20) (2013), 10218–10224.
2. K. CHEN, C. YI, *Robustness analysis of a hybrid of recursive neural dynamics for online matrix inversion*. Appl. Math. Comput. 273 (2016), 969–975.
3. A. CICHOCKI, T. KACZOREK, A. STAJNIAK, *Computation of the Drazin inverse of a singular matrix making use of neural networks*, Bulletin of the Polish Academy of Sciences Technical Sciences, 40 (1992).
4. J.S. JANG, S.Y. LEE, S. Y. SHIN, J. S. JANG, AND S. Y. SHIN, *An optimization network for matrix inversion*, Neural Inf. Process. Ser. (1987), 397–401.

5. S. Li, S. Chen, B. Liu, *Accelerating a recurrent neural network to finite-time convergence for solving time-varying Sylvester Equation by using a sign-bi-power activation function*, Neural Process. Lett. 37 (2013), 189–205.
6. Z. Li, Y. Zhang, *Improved Zhang neural network model and its solution of time-varying generalized linear matrix equations*, Expert Syst. Appl. 37 (2010), 7213–7218.
7. B. Liao, Y. Zhang, *Different complex ZFs leading to different complex ZNN models for time-varying complex generalized inverse matrices*, IEEE Trans. Neural Netw. Learn. Syst., 25 (2014), 1621–1631.
8. F.L. Luo, Z. Bao, *Neural network approach to computing matrix inversion*, Appl. Math. Comput. 47 (1992), 109–120.
9. S. Qiao, X.-Z. Wang, Y. Wei, *Two finite-time convergent Zhang neural network models for time-varying complex matrix Drazin inverse*, Linear Algebra Appl. http://dx.doi.org/10.1016/j.laa.2017.03.014.
10. P.S. Stanimirović, I. Živković, Y. Wei, *Recurrent neural network approach based on the integral representation of the Drazin inverse*, Neural Comput. 27(10) (2015), 2107–2131.
11. P.S. Stanimirović, I. S. Živković, Y. Wei, *Recurrent neural network for computing the Drazin inverse*, IEEE Trans. Neural Netw. Learn. Syst. 26 (2015), 2830–2843.
12. I. Stojanović, P.S. Stanimirović, I. Živković, D. Gerontitis, X.-Z. Wang, *ZNN models for computing matrix inverse based on hyperpower iterative methods*, Filomat 31:10 (2017), 2999–3014.
13. J. Wang, *A recurrent neural network for real-time matrix inversion*, Appl. Math. Comput. 55 (1993), 89–100.
14. J. Wang, *Recurrent neural networks for solving linear matrix equations*, Comput. Math. Appl. 26 (1993), 23–34.
15. J. Wang, *Recurrent neural networks for computing pseudoinverses of rank-deficient matrices*, SIAM J. Sci. Comput. 18 (1997), 1479–1493.
16. Y. Wei, *Recurrent neural networks for computing weighted Moore-Penrose inverse*, Appl. Math. Comput. 116 (2000), 279–287.
17. Y. Zhang, Y. Yang, N. Tan, B. Cai, *Zhang neural network solving for time-varying full-rank matrix Moore-Penrose inverse*, Computing 92 (2011), 97–121.
18. Y. Zhang, Y. Shi, K. Chen, C. Wang, *Global exponential convergence and stability of gradient-based neural network for online matrix inversion*, Appl. Math. Comput. 215 (2009), 1301–1306.
19. Y. Zhang, *Design and analysis of a general recurrent neural network model for time-varying matrix inversion*, IEEE Trans. Neural Netw. 16(6) (2005), 1477–1490.
20. Y. Zhang, Y. Shi, K. Chen, C. Wang, *Global exponential convergence and stability of gradient-based neural network for online matrix inversion*, Appl. Math. Comput. 215 (2009), 1301–1306.

Ivan S. Živković
University of Niš, Faculty of Science and Mathematics
Department of Computer Science
Višegradska 33, 18000 Niš, Serbia
zivkovic.ivan83@gmail.com

Predrag Stanimirović
University of Niš, Faculty of Science and Mathematics
Department of Computer Science
Višegradska 33, 18000 Niš, Serbia
pecko@pmf.ni.ac.rs