Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 8(11): 27376–27395.
DOI: 10.3934/math.20231401
Received: 09 August 2023
Revised: 12 September 2023
Accepted: 23 September 2023
Published: 26 September 2023

Research article

A quaternion Sylvester equation solver through noise-resilient zeroing neural networks with application to control the SFM chaotic system

Sondess B. Aoun[1], **Nabil Derbel**[2], **Houssem Jerbi**[3], **Theodore E. Simos**[4,5,6,7,8,*], **Spyridon D. Mourtas**[9,10] **and Vasilios N. Katsikis**[9]

[1] Department of Computer Engineering, College of Computer Science and Engineering, Univ. of Ha'il, Ha'il City 81451, Saudi Arabia
[2] Control & Energy Management Laboratory, National Engineering School of Sfax, Univ. of Sfax, Sfax, Tunisia
[3] Department of Industrial Engineering, College of Engineering, Univ. of Ha'il, Ha'il City 81451, Saudi Arabia
[4] Laboratory of Interdisciplinary Problems in Energy Production, Ulyanovsk State Technical Univ., 32 Severny Venetz Street, 432027 Ulyanovsk, Russia
[5] Department of Medical Research, China Medical Univ. Hospital, China Medical Univ., Taichung City 40402, Taiwan
[6] Center for Applied Math. and Bioinformatics, Gulf Univ. for Science and Technology, West Mishref, 32093 Kuwait
[7] Data Recovery Key Laboratory of Sichun Province, Neijing Normal Univ., Neijiang 641100, China
[8] Section of Mathematics, Dept. of Civil Engineering, Democritus Univ. of Thrace, Xanthi 67100, Greece
[9] Department of Economics, Division of Mathematics-Informatics and Statistics-Econometrics, National and Kapodistrian Univ.of Athens, Sofokleous 1 Street, 10559 Athens, Greece
[10] Laboratory "Hybrid Methods of Modelling and Optimization in Complex Systems", Siberian Federal Univ., Prosp. Svobodny 79, 660041 Krasnoyarsk, Russia

* **Correspondence:** Email: simos@ulstu.ru.

Abstract: Dynamic Sylvester equation (DSE) problems have drawn a lot of interest from academics due to its importance in science and engineering. Due to this, the quest for the quaternion DSE (QDSE) solution is the subject of this work. This is accomplished using the zeroing neural network (ZNN) technique, which has achieved considerable success in tackling time-varying issues. Keeping in mind that the original ZNN can handle QDSE successfully in a noise-free environment, but it might not work in a noisy one, and the noise-resilient ZNN (NZNN) technique is also utilized. In light of that, one new ZNN model is introduced to solve the QDSE problem and one new NZNN model is introduced to solve the QDSE problem under different types of noises. Two simulation experiments and one application to control of the sine function memristor (SFM) chaotic system show that the models function superbly.

Keywords: Quaternion; dynamic Sylvester equation; zeroing neural network; chaos control
Mathematics Subject Classification: 65F20, 68T05

1. Introduction

It is well known that solving dynamic Sylvester equations (DSEs) is an important challenge that occurs in many fields, including robot manipulators [1, 2], cell processors [3], control system architecture [4, 5], permanent magnet synchronous motors [6], image fusion [7], fast tensor product solution [8], object detection [9] and mobile manipulators [10]. In the past, academics have typically solved DSEs using classical iterative methods [11], such as Krylov subspace techniques [12] and the Hessenberg–Schur iteration technique [13]. However, time-invariant, real-valued and small-scale matrix DSEs are the only ones for which iterative approaches are appropriate. That is, due to their restricted computational capacity, iterative approaches are not the best option when tackling time-varying complex computing issues, and particularly DSEs, in real time [14].

We address a quaternion DSE (QDSE) problem through the zeroing neural network (ZNN) and the noise-resilient ZNN (NZNN) techniques. Hamilton first introduced quaternions, a non-commutative number system that expands on complex numbers, in 1843 [15]. They are useful for calculations requiring three-dimensional rotations in both theoretical and applied mathematics [16]. They are particularly important in several fields, including robotics [17], computer modeling [18], navigation [19], electromagnetism [20], quantum mechanics [21] and mathematical physics [22]. Lately, there has been increased interest in the investigation of dynamic problems that involve time-varying quaternion matrices (TQM), including the inversion of TQM [23], the pseudoinversion of TQM [24], the resolution of the constrained TQM least-squares problem [25] and the resolution of the linear TQM equation [26]. Additionally, chaotic system synchronization [25], mobile manipulator control [23, 27], kinematically redundant manipulator of robotic joints [28] and picture restoration [26, 29] are real-world uses of TQMs. One thing unites all of these studies: they all use the ZNN technique to arrive at the solution.

In order to deal with time-varying tasks in real time, the ZNN technique is introduced by Zhang *et al.* in [30]. ZNNs are a particular kind of recurrent neural networks that excel in parallel processing and their next acceptations were dynamic models for calculating the time-varying Moore-Penrose inverse in the real and complex domains [31, 32]. They are now used to solve problems involving generalized inversion [33, 34], linear and quadratic programming [35, 36, 37], systems of nonlinear equations [38, 39], and LMEs [40, 41], among other issues. Two major processes are normally involved in the construction of a ZNN model. The function of error matrix equation (EME) $E(t)$ must first be defined. Second, the following ZNN dynamical system must be employed:

$$\dot{E}(t) = -\lambda E(t), \tag{1.1}$$

where $(\dot{\ })$ is the time derivative operator and $t \in [0, t_f) \subseteq [0, +\infty)$ denotes the time. On top of that, one can change the model's convergence rate by adjusting the parameter $\lambda \in \mathbb{R}^+$. As an example, any ZNN model will converge even more quickly with a bigger value of λ [42]. The ZNN's architecture is based on setting each element of $E(t)$ to 0, which is true as $t \to \infty$. This is accomplished using the continuous-time learning regulation that arises from the establishment of EME in (1.1). As a consequence, EME can be considered a tool for monitoring ZNN model learning.

However, the precision of the proposed ZNN techniques is significantly impacted by all types of noise, and any prior noise reduction procedure adds time and sacrifices required real-time objectives. Therefore, a noise-resilient model for handling time-varying tasks was introduced in [43]. In particular, the following NZNN dynamical system was introduced therein:

$$\dot{E}(t) = -\lambda E(t) - \zeta \int_0^t E(\tau)\mathrm{d}\tau + N(t), \tag{1.2}$$

where $N(t)$ denotes a proper dimensional matrix-form noise and $\zeta, \lambda \in \mathbb{R}^+$ are design parameters for tracking the NZNN convergence.

Let $\mathbb{H} = \{\xi_1 + \xi_2\imath + \xi_3\jmath + \xi_4 k \mid \imath^2 = \jmath^2 = k^2 = \imath\jmath k = -1, \xi_1, \xi_2, \xi_3, \xi_4 \in \mathbb{R}\}$ be the set of quaternions and $\mathbb{H}^{n\times n}$ be the set of all $n \times n$ matrices on $\mathbb{H}$ [44]. In this paper, the next QDSE problem is addressed:

$$\tilde{A}(t)\tilde{X}(t) + \tilde{X}(t)\tilde{B}(t) = \tilde{C}(t), \tag{1.3}$$

where $\tilde{A}(t) \in \mathbb{H}^{m\times m}$, $\tilde{B}(t) \in \mathbb{H}^{n\times n}$ and $\tilde{C}(t) \in \mathbb{H}^{m\times n}$ are known and $\tilde{X}(t) \in \mathbb{H}^{m\times n}$ is unknown. It is important to mention that the QDSE problem has been addressed through the ZNN technique in [45]. However, the ZNN model presented therein always assumes that $m = n = 2$ in (1.3) and it is based on the complex representation of the quaternion [46]. In light of that, a new ZNN model, termed ZNNQ-D, is introduced in this paper to solve the QDSE problem in a more direct manner and it also covers the case where $m \neq n$ with $m, n \in \mathbb{N}$ in (1.3). Computational complexity analysis proves that the ZNNQ-D model has half the computational complexity of the ZNN model presented in [45]. Additionally, a new NZNN model, termed NZNNQ-D, is introduced in this paper to solve the QDSE problem under different types of noises. Two simulation experiments and one application to control of the sine function memristor (SFM) chaotic system show that the models function superbly. Last, through theoretical analysis of each model that is described, we add to the body of literature.

The following is a list of our contributions.

- A new ZNN model, termed ZNNQ-D, for addressing the QDSE problem is presented.
- A new NZNN model, termed NZNNQ-D, for addressing the QDSE problem under different types of noises is presented.
- A theoretical investigation is conducted to support the models.
- To support the theoretical research, simulation experiments and practical application to control of the SFM chaotic system are carried out.

For the remainder of this paper, the identity $g \times g$ matrix will be referred to as I_g whereas the zero $g \times g$ and $m \times n$ matrices will be referred to as $\mathbf{0}_g$ and $\mathbf{0}_{m\times n}$, respectively, and the all ones $g \times g$ and $m \times n$ matrices will be referred to as $\mathbf{1}_g$ and $\mathbf{1}_{m\times n}$, respectively. Moreover, the vectorization process will be denoted as vec(·), the Kronecker product will be denoted as ⊗ and the Hadamard product will be denoted as ⊙. Last, $\|\cdot\|_{\mathrm{F}}$ will denote the matrix Frobenius norm, TR(·) will denote the trace of a square matrix, and $(\cdot)^{\mathrm{T}}$ will denote matrix transposition.

The paper is structured as follows. Section 2 presents quaternion preliminaries and the QDSE problem reformulation. The ZNNQ-D model, which is based on the ZNN technique, is introduced in Section 3, while the NZNNQ-D model, which is based on the NZNN technique, is introduced in Section 4. It is important to note that the theoretical analysis and the computational complexity of the models are both included in Sections 3 and 4. Simulation experiments and application to control of the SFM chaotic system are presented in Section 5. Finally, Section 6 presents closing remarks and reflections.

2. Preliminaries and reformulation of the QDSE problem

In this section, the TQM foundations are laid out, and the QDSE problem is reformulated. The purpose of reformulating the QDSE problem (1.3) is to reduce the computational complexity of the ZNN and NZNN techniques.

Let $\tilde{A}(t) = A_1(t) + A_2(t)\imath + A_3(t)\jmath + A_4(t)k \in \mathbb{H}^{m\times m}$ be a TQM with coefficient matrices $A_i(t) \in \mathbb{R}^{m\times m}$ for $i = 1, 2, \cdots, 4$. Similarly, consider the TQMs $\tilde{B}(t) \in \mathbb{H}^{n\times n}$, $\tilde{C}(t) \in \mathbb{H}^{m\times n}$ and $\tilde{X}(t) \in \mathbb{H}^{m\times n}$ with coefficient matrices $B_i(t) \in \mathbb{R}^{n\times n}$ and $C_i(t), X_i(t) \in \mathbb{R}^{m\times n}$ for $i = 1, 2, \cdots, 4$, respectively. On the one hand, the product of $\tilde{A}(t)$ and $\tilde{X}(t)$ is as follows:

$$\tilde{A}(t)\tilde{X}(t) = \tilde{V}(t) = V_1(t) + V_2(t)\imath + V_3(t)\jmath + V_4(t)k \in \mathbb{H}^{m\times n} \tag{2.1}$$

where the coefficient matrices $V_i(t) \in \mathbb{R}^{m\times n}$ for $i = 1, 2, \cdots, 4$ are the following:

$$\begin{aligned}
V_1(t)&=A_1(t)X_1(t)-A_2(t)X_2(t)-A_3(t)X_3(t)-A_4(t)X_4(t),\\
V_2(t)&=A_1(t)X_2(t)+A_2(t)X_1(t)+A_3(t)X_4(t)-A_4(t)X_3(t),\\
V_3(t)&=A_1(t)X_3(t)+A_3(t)X_1(t)+A_4(t)X_2(t)-A_2(t)X_4(t),\\
V_4(t)&=A_1(t)X_4(t)+A_4(t)X_1(t)+A_2(t)X_3(t)-A_3(t)X_2(t).
\end{aligned} \tag{2.2}$$

On the other hand, the product of $\tilde{X}(t)$ and $\tilde{B}(t)$ is as follows:

$$\tilde{X}(t)\tilde{B}(t) = \tilde{U}(t) = U_1(t) + U_2(t)\imath + U_3(t)\jmath + U_4(t)k \in \mathbb{H}^{m\times n} \tag{2.3}$$

where the coefficient matrices $U_i(t) \in \mathbb{R}^{m\times n}$ for $i = 1, 2, \cdots, 4$ are the following:

$$\begin{aligned}
U_1(t)&=X_1(t)B_1(t)-X_2(t)B_2(t)-X_3(t)B_3(t)-X_4(t)B_4(t),\\
U_2(t)&=X_1(t)B_2(t)+X_2(t)B_1(t)+X_3(t)B_4(t)-X_4(t)B_3(t),\\
U_3(t)&=X_1(t)B_3(t)+X_3(t)B_1(t)+X_4(t)B_2(t)-X_2(t)B_4(t),\\
U_4(t)&=X_1(t)B_4(t)+X_4(t)B_1(t)+X_2(t)B_3(t)-X_3(t)B_2(t).
\end{aligned} \tag{2.4}$$

Based on the aforementioned, (1.3) may be reformulated as below:

$$\tilde{V}(t) + \tilde{U}(t) = \tilde{C}(t), \tag{2.5}$$

where the following is true:

$$\begin{cases}
V_1(t) + U_1(t) = C_1(t),\\
V_2(t) + U_2(t) = C_2(t),\\
V_3(t) + U_3(t) = C_3(t),\\
V_4(t) + U_4(t) = C_4(t),
\end{cases} \tag{2.6}$$

Then, setting

$$
\begin{aligned}
&Y(t)=\begin{bmatrix} X_1(t) & -X_2(t) & -X_3(t) & -X_4(t)\\ X_2(t) & X_1(t) & -X_4(t) & X_3(t)\\ X_3(t) & X_4(t) & X_1(t) & -X_2(t)\\ X_4(t) & -X_3(t) & X_2(t) & X_1(t)\end{bmatrix}\in\mathbb{R}^{4m\times 4n},\quad X(t)=\begin{bmatrix} X_1(t)\\ X_2(t)\\ X_3(t)\\ X_4(t)\end{bmatrix}\in\mathbb{R}^{4m\times n},\quad B(t)=\begin{bmatrix} B_1(t)\\ B_2(t)\\ B_3(t)\\ B_4(t)\end{bmatrix}\in\mathbb{R}^{4n\times n},\\
&D(t)=\begin{bmatrix} A_1(t) & -A_2(t) & -A_3(t) & -A_4(t)\\ A_2(t) & A_1(t) & -A_4(t) & A_3(t)\\ A_3(t) & A_4(t) & A_1(t) & -A_2(t)\\ A_4(t) & -A_3(t) & A_2(t) & A_1(t)\end{bmatrix}\in\mathbb{R}^{4m\times 4m},\quad C(t)=\begin{bmatrix} C_1(t)\\ C_2(t)\\ C_3(t)\\ C_4(t)\end{bmatrix}\in\mathbb{R}^{4m\times n},
\end{aligned}
\tag{2.7}
$$

(2.5) can be reformulated as below:

$$D(t)X(t) + Y(t)B(t) = C(t), \tag{2.8}$$

in which $X(t)$ and $Y(t)$ contain the coefficient matrices $X_i(t), i = 1, \dots, 4$, of the desired solution $\tilde{X}(t)$ to the QDSE problem (1.3). In contrast to solving (1.3), which yields only one TQM, (2.8) yields four real-valued time-varying matrices.

3. ZNN adaption for QDSE solution

In this section we shall develop a ZNN model, named ZNNQ-D, to solve the QDSE problem.

3.1. *The ZNNQ-D model*

We suppose that $\tilde{A}(t) \in \mathbb{H}^{m\times m}$, $\tilde{B}(t) \in \mathbb{H}^{n\times n}$ and $\tilde{C}(t), \tilde{X}(t) \in \mathbb{H}^{m\times n}$ are differentiable TQMs. According to Section's 2 analysis, the (2.8) problem is a reformulation of the (1.3) problem. We construct the differentiable matrices $D(t) \in \mathbb{R}^{4m\times 4m}$, $B(t) \in \mathbb{R}^{4n\times n}$ and $C(t) \in \mathbb{R}^{4m\times n}$ in accordance with (2.7) and take into account the following EME:

$$E^D(t) = D(t)X(t) + Y(t)B(t) - C(t), \tag{3.1}$$

where $Y(t) \in \mathbb{R}^{4m\times 4n}$ and $X(t) \in \mathbb{R}^{4m\times n}$ are unknown matrices. Its first derivative is:

$$\dot{E}^D(t) = \dot{D}(t)X(t) + D(t)\dot{X}(t) + \dot{Y}(t)B(t) + Y(t)\dot{B}(t) - \dot{C}(t). \tag{3.2}$$

The following is the outcome of addressing the ZNN dynamical system in terms of $\dot{X}(t)$ and $\dot{Y}(t)$ when $E(t)$ and $\dot{E}(t)$ in (1.1) are substituted with $E^D(t)$ and $\dot{E}^D(t)$ determined in (3.1) and (3.2), respectively:

$$D(t)\dot{X}(t) + \dot{Y}(t)B(t) = -\lambda E^D(t) - \dot{D}(t)X(t) - Y(t)\dot{B}(t) + \dot{C}(t). \tag{3.3}$$

To simplify the dynamics of (3.3), the vectorization and Kronecker product are utilized:

$$(I_n\otimes D(t))\mathrm{vec}(\dot{X}(t))+(B^{\mathrm{T}}(t)\otimes I_{4m})\mathrm{vec}(\dot{Y}(t))=\mathrm{vec}(-\lambda E^D(t)-\dot{D}(t)X(t)-Y(t)\dot{B}(t)+\dot{C}(t)). \tag{3.4}$$

It is significant to note that identical elements, but in different positions, can be found in the vectors $\mathrm{vec}(\dot{X}(t))$ and $\mathrm{vec}(\dot{Y}(t))$. In other words, it is possible to further simplify (3.4) by rewriting the vector $\text{vec}(\dot{Y}(t))$ in terms of $\text{vec}(\dot{X}(t))$. As a consequence, it is feasible to create the next equation that substitutes $\text{vec}(\dot{Y}(t))$ in (3.4):

$$\text{vec}(\dot{Y}(t)) = J\text{vec}(\dot{X}(t)), \tag{3.5}$$

where $J \in \mathbb{R}^{16mn\times 4mn}$ is an operational matrix that may be computed utilizing the algorithmic process presented in Algorithm 1. Notice that the notations in Alg. 1 follow the usual MATLAB function theme [47].

Algorithm 1 Matrix J calculation.

```
Input: The rows $m$ and columns $n$ numbers of a matrix $A \in \mathbb{R}^{m\times n}$.
procedure OM_J($m, n$)
    Set $J$ =zeros($16mn, 4mn$), $x = (1 : 4mn)'$ and $X$ =reshape($x, 4m, n$)
    Set $X1 = X(1 : m, :)$, $X2 = X(m + 1 : 2m, :)$, $X3 = X(2m + 1 : 3m, :)$, $X4 = X(3m + 1 : end, :)$
    Set $Y = [X1, -X2, -X3, -X4; X2, X1, -X4, X3; X3, X4, X1, -X2; X4, -X3, X2, X1]$
    Set $Z = reshape(Y, [], 1)$
    for $j = 1 : 16mn$ do
        $J(i$,abs$(Z(i)))$ =sign$(Z(i))$
    end for
    return $J$
end procedure
Output: The operational matrix $J$.
```

By using (3.5), we can further simplify (3.4) as follows:

$$(I_n\otimes D(t))\text{vec}(\dot{X}(t))+(B^{\mathrm{T}}(t)\otimes I_{4m})J\text{vec}(\dot{X}(t))=\text{vec}(-\lambda E^{D}(t)-\dot{D}(t)X(t)-Y(t)\dot{B}(t)+\dot{C}(t)). \tag{3.6}$$

In addition, once the followings have been set:

$$\begin{aligned}&Q(t)=I_n\otimes D(t)+(B^{\mathrm{T}}(t)\otimes I_{4m})J\in\mathbb{R}^{4mn\times 4mn}, \quad \mathbf{r}(t)=\text{vec}(X(t))\in\mathbb{R}^{4mn},\\ &G(t)=\text{vec}(-\lambda E^{D}(t)-\dot{D}(t)X(t)-Y(t)\dot{B}(t)+\dot{C}(t))\in\mathbb{R}^{4mn}, \quad \dot{\mathbf{r}}(t)=\text{vec}(\dot{X}(t))\in\mathbb{R}^{4mn},\end{aligned} \tag{3.7}$$

we get at the next ZNN model:

$$Q(t)\dot{\mathbf{r}}(t) = G(t) \tag{3.8}$$

where $Q(t)$ is an invertible mass matrix. The recommended ZNN model to be employed in addressing the QDSE problem of (1.3) is the dynamic model of (3.8), referred to as ZNNQ-D.

3.2. ZNNQ-D model theoretical analysis

This section presents the ZNNQ-D (3.8) model's examination of convergence and stability.

Theorem 3.1. *Let $D(t) \in \mathbb{R}^{4m\times 4m}$, $B(t) \in \mathbb{R}^{4n\times n}$ and $C(t) \in \mathbb{R}^{4m\times n}$. Also, suppose that $D(t), B(t)$ and $C(t)$ are differentiable. Then, the system* (3.3) *the based on the ZNN theme* (1.1) *converges to the theoretical solution (TSOL) $\hat{X}(t)$, and the solution is stable, in line with Lyapunov.*

Proof. The replacement $\bar{X}(t) := \hat{X}(t) - X(t)$, where $\hat{X}(t)$ is the TSOL, entails $X(t) = \hat{X}(t) - \bar{X}(t)$. According to (2.7), $Y(t)$ is a rearrangement of the elements of $X(t)$. As a consequence, $Y(t) = \hat{Y}(t)-\bar{Y}(t)$

since $\hat{Y}(t)$ and $\bar{Y}(t)$ are a rearrangement of the elements of $\hat{X}(t)$ and $\bar{X}(t)$, respectively. The time-derivatives of $X(t)$ and $Y(t)$ are $\dot{X}(t) = \dot{\hat{X}}(t) - \dot{\bar{X}}(t)$ and $\dot{Y}(t) = \dot{\hat{Y}}(t) - \dot{\bar{Y}}(t)$, respectively.

Considering that

$$D(t)\hat{X}(t) + \hat{Y}(t)B(t) - C(t) = \mathbf{0}_{4m\times n}, \tag{3.9}$$

and its first derivative

$$D(t)\dot{\hat{X}}(t) + \dot{D}(t)\hat{X}(t) + \dot{\hat{Y}}(t)B(t) + \hat{Y}(t)\dot{B}(t) - \dot{C}(t) = \mathbf{0}_{4m\times n}, \tag{3.10}$$

one can confirm the following after replacing $X(t) = \hat{X}(t) - \bar{X}(t)$ and $Y(t) = \hat{Y}(t) - \bar{Y}(t)$ with (3.1):

$$\hat{E}^D(t) = D(t)\hat{X}(t) - D(t)\bar{X}(t) + \hat{Y}(t)B(t) - \bar{Y}(t)B(t) - C(t). \tag{3.11}$$

In addition, the dynamics of (1.1) yield

$$\begin{aligned}\dot{\hat{E}}^D(t) &= D(t)\dot{\hat{X}}(t) - D(t)\dot{\bar{X}}(t) + \dot{\hat{Y}}(t)B(t) - \dot{\bar{Y}}(t)B(t) - \dot{C}(t) + \dot{D}(t)\hat{X}(t) - \dot{D}(t)\bar{X}(t) + \hat{Y}(t)\dot{B}(t) - \bar{Y}(t)\dot{B}(t)\\ &= -\lambda\hat{E}^D(t).\end{aligned} \tag{3.12}$$

After that, we choose the following potential Lyapunov function to corroborate convergence:

$$\mathcal{L}(t) = \frac{1}{2}\left\|\hat{E}^D(t)\right\|_{\mathrm{F}}^2 = \frac{1}{2}\mathrm{TR}\left(\hat{E}^D(t)\left(\hat{E}^D(t)\right)^{\mathrm{T}}\right). \tag{3.13}$$

The next may then be confirmed:

$$\dot{\mathcal{L}}(t) = \frac{2\mathrm{TR}\left(\left(\hat{E}^D(t)\right)^{\mathrm{T}}\dot{\hat{E}}^D(t)\right)}{2} = \mathrm{TR}\left(\left(\hat{E}^D(t)\right)^{\mathrm{T}}\dot{\hat{E}}^D(t)\right) = -\lambda\mathrm{TR}\left(\left(\hat{E}^D(t)\right)^{\mathrm{T}}\hat{E}^D(t)\right). \tag{3.14}$$

As a consequence, it holds

$$\begin{aligned}&\dot{\mathcal{L}}(t)\begin{cases}<0, & \hat{E}^D(t)\neq 0\\ =0, & \hat{E}^D(t)=0,\end{cases} \Leftrightarrow \dot{\mathcal{L}}(t)\begin{cases}<0, & D(t)\hat{X}(t)+\hat{Y}(t)B(t)-C(t)-D(t)\bar{X}(t)-\bar{Y}(t)B(t)\neq 0\\ =0, & D(t)\hat{X}(t)+\hat{Y}(t)B(t)-C(t)-D(t)\bar{X}(t)-\bar{Y}(t)B(t)=0,\end{cases}\\ &\Leftrightarrow\dot{\mathcal{L}}(t)\begin{cases}<0, & \bar{X}(t)\neq 0\ \&\ \bar{Y}(t)\neq 0\\ =0, & \bar{X}(t)=0\ \&\ \bar{Y}(t)=0,\end{cases} \Leftrightarrow \dot{\mathcal{L}}(t)\begin{cases}<0, & \bar{X}(t)\neq 0\\ =0, & \bar{X}(t)=0.\end{cases}\end{aligned} \tag{3.15}$$

Notice that $\bar{Y}(t) \neq 0$ when $\bar{X}(t) \neq 0$ and $\bar{Y}(t) = 0$ when $\bar{X}(t) = 0$ since $Y(t)$ is a rearrangement of the $X(t)$ elements. Also, due to the fact that $\bar{X}(t)$ is the equilibrium point of (3.12) and $E^D(0) = 0$, the following holds:

$$\dot{\mathcal{L}}(t) \leq 0, \quad \forall\, \bar{X}(t) \neq 0. \tag{3.16}$$

We conclude that the equilibrium state $\bar{X}(t) = \hat{X}(t) - X(t) = 0$ is stable in line with the Lyapunov stability theory. Thereafter, $X(t) \to \hat{X}(t)$ as $t \to \infty$. Notice that $Y(t) \to \hat{Y}(t)$ as $t \to \infty$ since $Y(t)$ is a rearrangement of the $X(t)$ elements. □

Theorem 3.2. *Let $\tilde{A}(t) \in \mathbb{H}^{m\times m}$, $\tilde{B}(t) \in \mathbb{H}^{n\times n}$ and $\tilde{C}(t) \in \mathbb{H}^{m\times n}$. Also, suppose that $\tilde{A}(t), \tilde{B}(t)$ and $\tilde{C}(t)$ are differentiable. At every $t \in [0, t_f) \subseteq [0, +\infty)$, the ZNNQ-D model* (3.8) *converges exponentially to the TSOL $\hat{\mathbf{r}}(t)$ for every initial price $\mathbf{r}(0)$ that one may take into consideration.*

Proof. First, the QDSE problem of (1.3) is converted into the problem of (2.8), based on the analysis shown in Section 2. More particularly, using the matrices $\tilde{A}(t)$ and $\tilde{B}(t)$, we create the matrices $D(t) \in \mathbb{R}^{4m\times 4m}$, $B(t) \in \mathbb{R}^{4n\times n}$ and $C(t) \in \mathbb{R}^{4m\times n}$ according to (2.7). As a result, we convert the problem of (1.3) into the problem of (2.8). Second, to solve the problem of (2.8), the EME of (3.1) is declared. For zeroing (3.1), the model (3.3) is deployed in line with the ZNN theme (1.1). According to Theorem 3.1, $X(t) \to \hat{X}(t)$ and $Y(t) \to \hat{Y}(t)$ when $t \to \infty$ for any choice of initial value. So, the model (3.3) converges to the TSOL of the (1.3), since (2.8) is a reformulation of the QDSE problem of (1.3). Third, the model (3.3) is simplified into the ZNNQ-D model (3.8) using the Kronecker product and vectorization. As an alternative version of (3.3), for every initial value $\mathbf{r}(0)$, the ZNNQ-D model (3.8) also converges to the TSOL $\hat{\mathbf{r}}(t)$ when $t \to \infty$. Thereafter, the proof is finished. □

3.3. ZNNQ-D model computational complexity

The complexity of creating and addressing (3.8) adds to the ZNNQ-D's total computational complexity. Particularly, the complexity of creating (3.8) is $O((4mn)^2)$ operations because at every iteration we conduct $(4mn)^2$ multiplications and $4mn$ subtractions/additions. On top of that, the implicit MATLAB solver `ode15s` is used to address at each step the linear system of equations. The complexity of addressing (3.8) is $O((4mn)^3$ as it necessitates a $4mn \times 4mn$ matrix. So, the ZNNQ-D model's total computational complexity is $O((4mn)^3)$.

It is significant also to mention the total computational complexity of the ZNN model presented in [45]. However, this model deals with complex numbers. That is, it takes a total of 4 multiplications and 2 subtraction/addition operations to multiply two complex numbers, as shown by the formula $(q + b\imath)(c + p\imath) = qc - bp + qp\imath + bc\imath$. So, the total computational complexity of the model presented in [45] is $O((8mn)^3)$, where $m = n = 2$. Therefore, the ZNNQ-D model has half the computational complexity of the ZNN model presented in [45] when $m = n = 2$.

4. NZNN adaption for QDSE solution under different types of noises

In this section we shall develop a NZNN model, named NZNNQ-D, to solve the QDSE problem under different forms of noise.

4.1. The NZNNQ-D model

We suppose that $\tilde{A}(t) \in \mathbb{H}^{m\times m}$, $\tilde{B}(t) \in \mathbb{H}^{n\times n}$ and $\tilde{C}(t), \tilde{X}(t) \in \mathbb{H}^{m\times n}$ are differentiable TQMs. According to Section's 2 analysis, the (2.8) problem is a reformulation of the (1.3) problem. We construct the differentiable matrices $D(t) \in \mathbb{R}^{4m\times 4m}$, $B(t) \in \mathbb{R}^{4n\times n}$ and $C(t) \in \mathbb{R}^{4m\times n}$ in accordance with (2.7) and take into account the EME defined in (3.1) and its first derivative defined in (3.2).

The following is the outcome of addressing the NZNN dynamical system in terms of $\dot{X}(t)$ and $\dot{Y}(t)$ when $E(t)$ and $\dot{E}(t)$ in (1.2) are substituted with $E^D(t)$ and $\dot{E}^D(t)$ determined in (3.1) and (3.2), respectively:

$$D(t)\dot{X}(t) + \dot{Y}(t)B(t) = -\lambda E^D(t) - \zeta \int_0^t E^D(\tau)\mathrm{d}\tau + N(t) - \dot{D}(t)X(t) - Y(t)\dot{B}(t) + \dot{C}(t), \tag{4.1}$$

where $N(t) \in \mathbb{R}^{4m\times n}$ is matrix-form noise. To simplify the dynamics of (4.1), the vectorization and

Kronecker product are utilized:

$$(I_n \otimes D(t))\mathrm{vec}(\dot{X}(t)) + (B^{\mathrm{T}}(t) \otimes I_{4m})\mathrm{vec}(\dot{Y}(t)) = \mathrm{vec}(-\lambda E^D(t) - \zeta P(t) + N(t) - \dot{D}(t)X(t) - Y(t)\dot{B}(t) + \dot{C}(t)), \tag{4.2}$$

where $P(t) = \int_0^t E^D(\tau)\mathrm{d}\tau \in \mathbb{R}^{4m\times n}$. By using (3.5), we can further simplify (4.2) as follows:

$$(I_n \otimes D(t))\mathrm{vec}(\dot{X}(t)) + (B^{\mathrm{T}}(t) \otimes I_{4m})J\mathrm{vec}(\dot{X}(t)) = \mathrm{vec}(-\lambda E^D(t) - \zeta P(t) + N(t) - \dot{D}(t)X(t) - Y(t)\dot{B}(t) + \dot{C}(t)). \tag{4.3}$$

In addition, once the followings have been set:

$$\begin{aligned} R(t) &= \begin{bmatrix} I_{4mn} & \mathbf{0}_{4mn\times 4mn} \\ \mathbf{0}_{4mn\times 4mn} & I_n \otimes D(t) + (B^{\mathrm{T}}(t) \otimes I_{4m})J \end{bmatrix} \in \mathbb{R}^{8mn\times 8mn}, \quad \mathbf{h}(t) = \begin{bmatrix} \mathrm{vec}(P(t)) \\ \mathrm{vec}(X(t)) \end{bmatrix} \in \mathbb{R}^{8mn}, \\ K(t) &= \begin{bmatrix} \mathrm{vec}(E^D(t)) \\ \mathrm{vec}(-\lambda E^D(t) - \zeta P(t) + N(t) - \dot{D}(t)X(t) - Y(t)\dot{B}(t) + \dot{C}(t)) \end{bmatrix} \in \mathbb{R}^{8mn}, \quad \dot{\mathbf{h}}(t) = \begin{bmatrix} \mathrm{vec}(\dot{P}(t)) \\ \mathrm{vec}(\dot{X}(t)) \end{bmatrix} \in \mathbb{R}^{8mn}, \end{aligned} \tag{4.4}$$

we get at the next NZNN model:

$$R(t)\dot{\mathbf{h}}(t) = K(t) \tag{4.5}$$

where $R(t)$ is an invertible mass matrix. The recommended NZNN model to be employed in addressing the QDSE problem of (1.3) under different forms of noise is the dynamic model of (4.5), referred to as NZNNQ-D.

The complexity of creating and addressing (4.5) adds to the NZNNQ-D's total computational complexity. Particularly, the complexity of creating (4.5) is $O((8mn)^2)$ operations because at every iteration we conduct $(8mn)^2$ multiplications and $8mn$ subtractions/additions. On top of that, the implicit MATLAB solver `ode15s` is used to address at each step the linear system of equations. The complexity of addressing (4.5) is $O((8mn)^3$ as it necessitates a $8mn \times 8mn$ matrix. So, the NZNNQ-D model's total computational complexity is $O((8mn)^3)$.

4.2. NZNNQ-D model theoretical analysis

This section presents the NZNNQ-D (3.8) model's examination of convergence and stability. Notice that some of following theorems seek to address various forms of noise and are rehashed from [43].

Theorem 4.1. *Let $D(t) \in \mathbb{R}^{4m\times 4m}$, $B(t) \in \mathbb{R}^{4n\times n}$ and $C(t) \in \mathbb{R}^{4m\times n}$. Also, suppose that $D(t), B(t)$ and $C(t)$ are differentiable. Then, the system* (4.1)*, which is based on the NZNN theme* (1.2)*, converges to the TSOL $\hat{X}(t)$ under the ideal noise-free condition, and the solution is stable, in line with Lyapunov.*

Proof. The proof is omitted as it is analogous to the Theorem's 3.1 proof. □

Theorem 4.2. [43, Theorem 3] *Considering the Theorem's 4.1 presumptions, the system* (4.1) *contaminated with the constant noise $N(t) = N \in \mathbb{R}^{4m\times n}$ converges globally to the TSOL.*

Theorem 4.3. [43, Theorem 4] *Considering the Theorem's 4.1 presumptions, the system* (4.1) *contaminated with the linear noise $N(t) = N \cdot t \in \mathbb{R}^{4m\times n}$ converges globally to the TSOL, with the upper bound of the EME accomplishing $\lim_{t\to\infty} \|E^D(t)\|_{\mathrm{F}} = \frac{1}{\zeta}\|N\|_{\mathrm{F}}$. Also, as $\zeta \to +\infty$, $E^D(t)$ accomplishes $\lim_{t\to\infty} \|E^D(t)\|_{\mathrm{F}} \downarrow 0$.*

Theorem 4.4. [43, Theorem 5] *Considering the Theorem's 4.1 presumptions, the system* (4.1) *contaminated with the bounded random noise* $N(t) := \sigma(t) = [\sigma_{ij}(t)]_{i,j=1,\ldots,n} \in \mathbb{R}^{4m\times n}$ *preserves the bounded residual error* $\|E^D(t)\|_{\mathrm{F}}$. *Also,* $\lim_{t\to\infty} \|E^D(t)\|_{\mathrm{F}}$ *is bounded by*

$$\begin{cases} \sup\limits_{0\le\tau\le t} |\sigma_{ij}(\tau)| \frac{2\rho}{\sqrt{Q}} & , Q > 0 \\ \sup\limits_{0\le\tau\le t} |\sigma_{ij}(\tau)| \frac{4\rho\zeta}{\sqrt{-Q}} & , Q < 0 \end{cases} \tag{4.6}$$

where $Q = -4\zeta + \eta^2$ *and* $\eta, \zeta > 0$ *are parameters. Thus, the upper bound of* $\lim_{t\to\infty} \|E^D(t)\|_{\mathrm{F}}$ *is basically an inverted counterpart of* η *in the case of* $Q \neq 0$ *as well as* $\lim_{t\to\infty} \|E^D(t)\|_{\mathrm{F}}$ *arbitrarily being tiny for sufficient large* η *and proper* ζ.

Theorem 4.5. *Let* $\tilde{A}(t) \in \mathbb{H}^{m\times m}$, $\tilde{B}(t) \in \mathbb{H}^{n\times n}$ *and* $\tilde{C}(t) \in \mathbb{H}^{m\times n}$. *Also, suppose that* $\tilde{A}(t), \tilde{B}(t)$ *and* $\tilde{C}(t)$ *are differentiable. At every* $t \in [0, t_f) \subseteq [0, +\infty)$, *the NZNNQ-D model* (4.5) *converges exponentially to the TSOL* $\hat{\mathbf{h}}(t)$ *under constant, linear and bounded random noises* $N(t) \in \mathbb{R}^{4m\times n}$ *for every initial price* $\mathbf{h}(0)$ *that one may take into consideration.*

Proof. The proof is omitted as it is analogous to the Theorem's 3.2 proof once we replace Theorem 3.1 with Theorem 4.1 and also include the Theorems 4.2, 4.3 and 4.4, respectively, for the constant, linear and bounded random noises, it is omitted. □

5. Simulation experiments

This section will outline an application to control of the SFM chaotic system as well as two simulation examples (SEs). The following includes a few key justifications. In the SEs, the initial conditions (ICs) of the ZNNQ-D and NZNNQ-D models have been set as follows:

- IC1: $\mathbf{r}(0) = \mathbf{0}_{mn}$ and $\mathbf{h}(0) = \mathbf{0}_{mn}$,
- IC2: $\mathbf{r}(0) = \mathbf{1}_{mn}$ and $\mathbf{h}(0) = \mathbf{1}_{mn}$,

the parameter λ is utilized with values 10 and 10^2, and the parameter ζ is utilized with values 10, 10^2 and 10^3. For simplicity, we have set $\xi(t) = \sin(t)$ and $\psi(t) = \cos(t)$. Additionally, in the SEs and application, computations are performed using the MATLAB ode solver, `ode15s`, while its time interval has been set to $[0, 10]$ in SEs and $[0, 500]$ in application. It is crucial to note that we utilize the `ode15s` with its regular double precision arithmetic ($eps = 2.22 \cdot 10^{-16}$), which means that all of the errors in the figures of this section have a minimum value that is close to 10^{-5}.

5.1. Simulations

Example 5.1. In this SE, the matrix $\tilde{A}(t)$ coefficients are set to

$$A_1(t) = \begin{bmatrix} 3\xi(t)+1 & 4 & 4 \\ 3\psi(t)+2 & 5 & 5 \\ 2 & \psi(t)+3 & \psi(t)+3 \end{bmatrix}, \quad A_2(t) = \begin{bmatrix} 5 & 2\xi(t)+2 & 2\xi(t)+2 \\ 3\xi(t)-2 & 3 & 3 \\ -2\psi(t)+5 & \psi(t)+5 & \psi(t)+5 \end{bmatrix},$$

$$A_3(t) = \begin{bmatrix} 3\xi(t)+2 & 5 & 5 \\ 2\xi(t)+1 & 6 & 6 \\ -\psi(t)+3 & 4 & 4 \end{bmatrix}, \quad A_4(t) = \begin{bmatrix} 1 & 2\xi(t)+3 & 2\xi(t)+3 \\ 5 & 9 & 9 \\ 3\psi(t)+2 & 5 & 5 \end{bmatrix},$$

the matrix $\tilde{B}(t)$ coefficients are set to

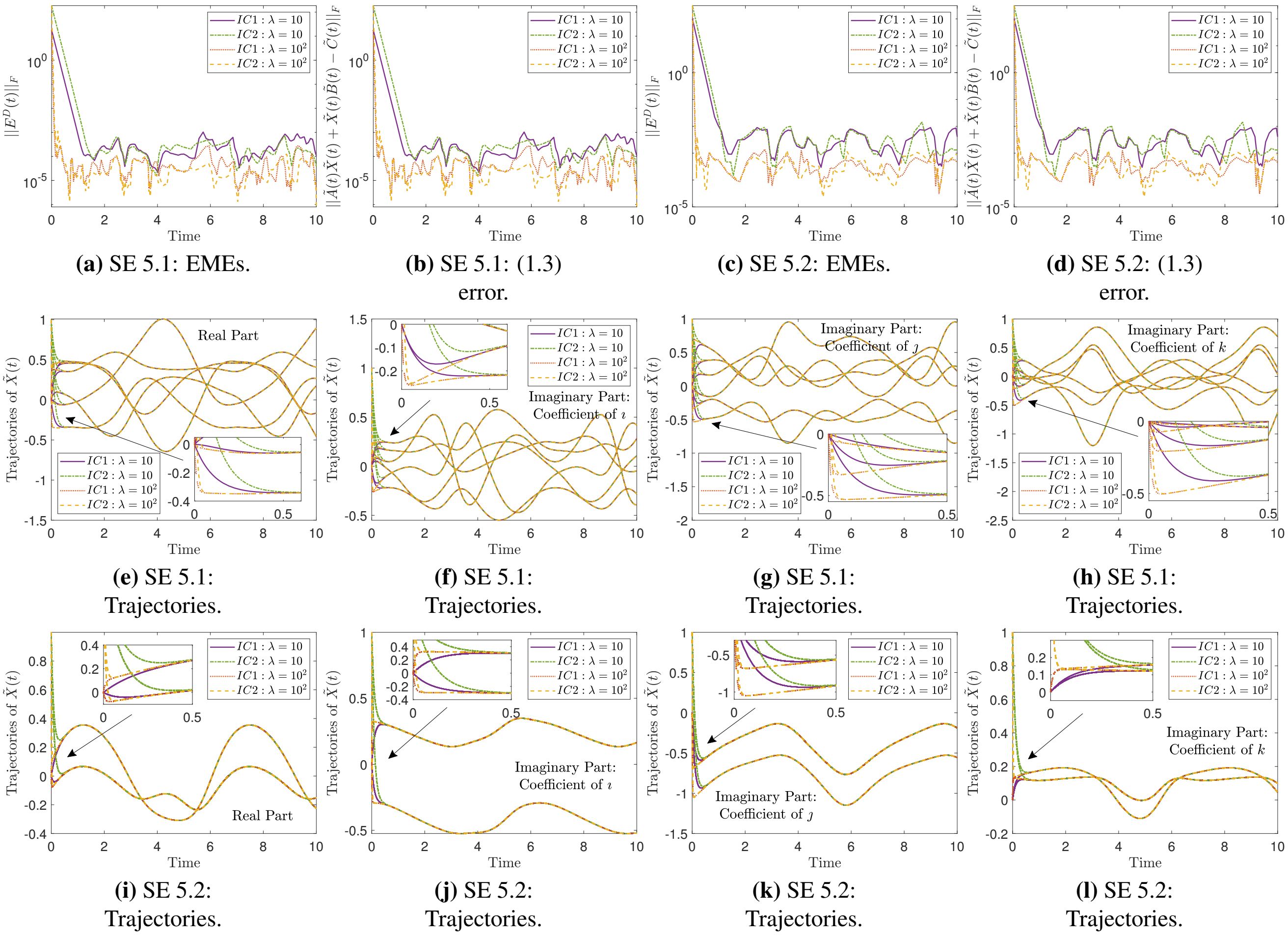
$$B_1(t) = \begin{bmatrix} \psi(t) & \xi(t)+2 \\ 5 & \xi(t)+1 \end{bmatrix}, \quad B_2(t) = \begin{bmatrix} \psi(t)+4 & 2 \\ \psi(t) & 8 \end{bmatrix}, \quad B_3(t) = \begin{bmatrix} 3\xi(t)+2 & 6 \\ 5 & 8 \end{bmatrix}, \quad B_4(t) = \begin{bmatrix} 2\xi(t)+3 & 4 \\ \psi(t) & \xi(t)+1 \end{bmatrix},$$

and the matrix $\tilde{C}(t)$ coefficients are set to

$$C_1(t) = \begin{bmatrix} \psi(t)+3 & \psi(t) \\ 3 & \psi(t)+3 \\ 1 & 1 \end{bmatrix}, \quad C_2(t) = \begin{bmatrix} \psi(t)+1 & 3 \\ \xi(t) & 4 \\ \psi(t)+3 & \xi(t) \end{bmatrix},$$

$$C_3(t) = \begin{bmatrix} 2\xi(t)+3 & 7 \\ 5 & 8 \\ 4 & \psi(t) \end{bmatrix}, \quad C_4(t) = \begin{bmatrix} 5\psi(t)+1 & 3 \\ \xi(t) & \xi(t)+1 \\ 6 & 1 \end{bmatrix}.$$

As a result, $\tilde{A}(t) \in \mathbb{H}^{3\times3}$, $\tilde{B}(t) \in \mathbb{H}^{2\times2}$ and $\tilde{B}(t) \in \mathbb{H}^{3\times2}$. The ZNNQ-D model's generated results are presented in Figure 1, whereas the NZNNQ-D model's generated results under the bounded noise $N = \mathbf{1}_{3\times2} \odot (\xi(t) + 2)$ are presented in Figure 2.



(a) SE 5.1: EMEs.

(b) SE 5.1: (1.3) error.

(c) SE 5.2: EMEs.

(d) SE 5.2: (1.3) error.

(e) SE 5.1: Trajectories.

(f) SE 5.1: Trajectories.

(g) SE 5.1: Trajectories.

(h) SE 5.1: Trajectories.

(i) SE 5.2: Trajectories.

(j) SE 5.2: Trajectories.

(k) SE 5.2: Trajectories.

(l) SE 5.2: Trajectories.

Figure 1. EMEs and (1.3) error of ZNNQ-D in SEs 5.1 and 5.2.

Example 5.2. In this SE, the matrix $\tilde{A}(t)$ coefficients are set to

$$A_1(t) = I_{10} \odot (\psi(t) + 2), \quad A_2(t) = I_{10} \odot \xi(t), \quad A_3(t) = I_{10} \odot \xi(t)/2, \quad A_4(t) = \mathbf{1}_{10} \odot \psi(t)/2,$$

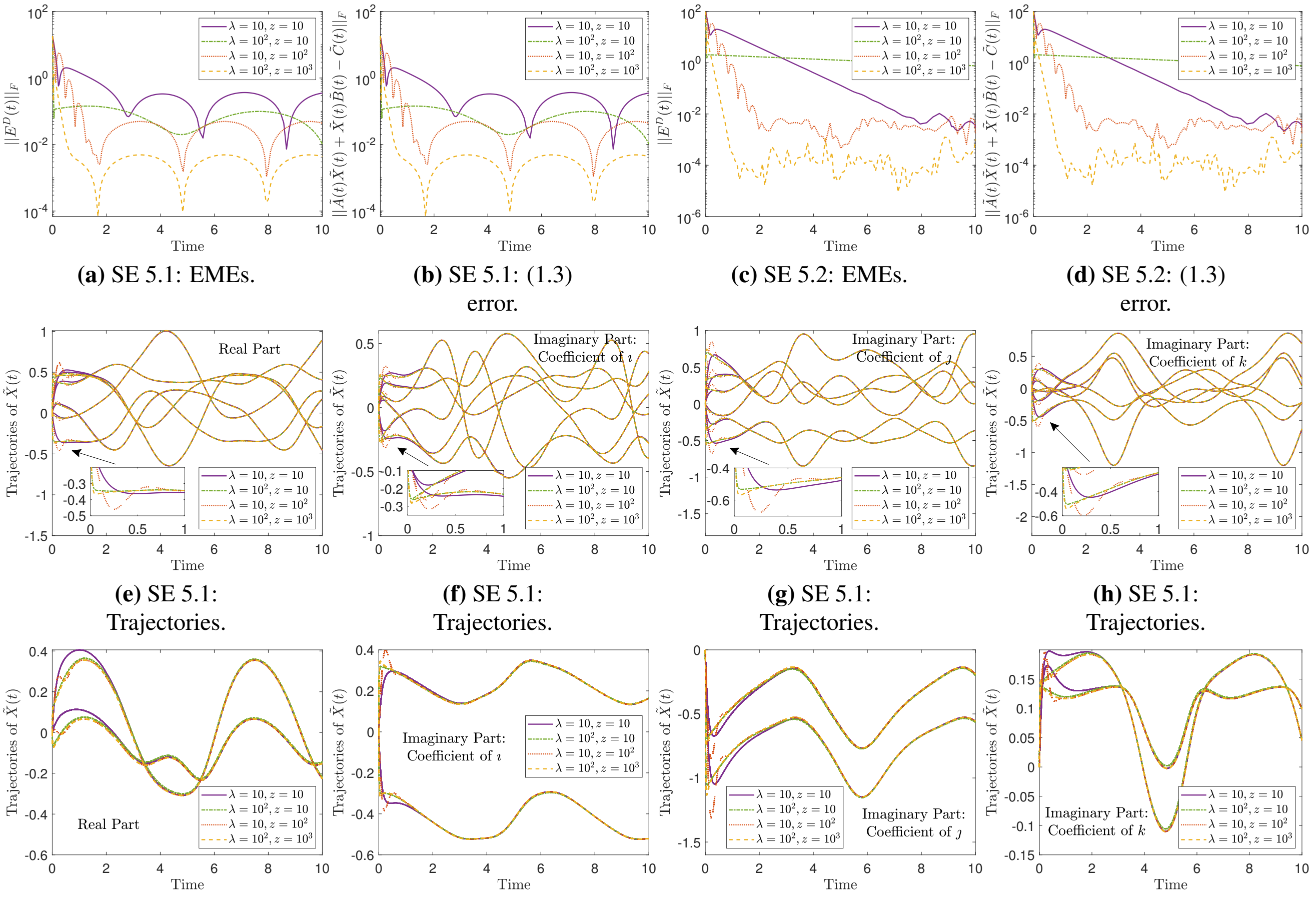
the matrix $\tilde{B}(t)$ coefficients are set to

$$B_1(t) = I_{10} \odot 2\xi(t), \quad B_2(t) = I_{10} \odot 2(\xi(t) + 5), \quad B_3(t) = I_{10} \odot 2(\xi(t) + 3), \quad B_4(t) = I_{10} \odot 2\xi(t),$$

and the matrix $\tilde{C}(t)$ coefficients are set to

$$C_1(t) = I_{10} \odot 3(\xi(t) + 3), \quad C_2(t) = \mathbf{1}_{10} \odot 3\psi(t), \quad C_3(t) = I_{10} \odot 3\xi(t), \quad C_4(t) = \mathbf{1}_{10} \odot 3(\psi(t) + 2).$$

Therefore, $\tilde{A}(t), \tilde{B}(t), \tilde{C}(t) \in \mathbb{H}^{10\times10}$. The results of the ZNNQ-D model are presented in Figure 1, while the results of the NZNNQ-D model under the constant noise $N = \mathbf{1}_{10\times10} \odot 10$ are presented in Figure 2.



(a) SE 5.1: EMEs.

(b) SE 5.1: (1.3) error.

(c) SE 5.2: EMEs.

(d) SE 5.2: (1.3) error.

(e) SE 5.1: Trajectories.

(f) SE 5.1: Trajectories.

(g) SE 5.1: Trajectories.

(h) SE 5.1: Trajectories.

(i) SE 5.2: Trajectories.

(j) SE 5.2: Trajectories.

(k) SE 5.2: Trajectories.

(l) SE 5.2: Trajectories.

Figure 2. EMEs and (1.3) error of NZNNQ-D in SEs 5.1 and 5.2.

5.2. *Discussion on SEs results*

The performance of the ZNNQ-D (3.8) and NZNNQ-D (4.5) models for solving the QDSE problem of (1.3) is investigated throughout the SEs 5.1 and 5.2. Each SE is associated with a unique QDSE problem that is specified by the proper matrices $\tilde{A}(t)$, $\tilde{B}(t)$ and $\tilde{C}(t)$.

In the SE 5.1, we have that $\tilde{A}(t) \in \mathbb{H}^{3\times3}$, $\tilde{B}(t) \in \mathbb{H}^{2\times2}$ and $\tilde{B}(t) \in \mathbb{H}^{3\times2}$. That is, $m = 3$ and $n = 2$ in the QDSE problem of (1.3). For the ZNNQ-D model, we have the following findings under IC1 and IC2 for $\lambda = 10$ and $\lambda = 10^2$. Fig. 1a shows the EMEs of the ZNNQ-D model. In this figure, all cases start at $t = 0$ from a large error value and arrive to a small one in the interval $[10^{-6}, 10^{-3}]$ at $t = 2$ when $\lambda = 10$ and at $t = 0.2$ when $\lambda = 10^2$. In other words, the ZNNQ-D model converges to a minimum value for two different ICs, validating Theorem 3.2, whereas the value of λ influences the rate at which the EMEs converge. Fig. 1b shows the (1.3) error of the ZNNQ-D model. The findings shown there are the same as the results shown in Figure 1a, demonstrating that solving (2.8) is equivalent to solving (1.3). The trajectories of the solutions generated by the model are presented in Figure 1e–1h. The real part and the three imaginary parts of the solutions are depicted in these figures, respectively. These graphs demonstrate that the solutions generated by the models are identical and that they converge to the TSOL in a way that is compatible with the convergence tendency of the associated EMEs.

For the NZNNQ-D model in the SE 5.1, we have the following findings under IC1 and a bounded noise for $\lambda = 10$ and $\lambda = 10^2$ with $\zeta = 10$, $\zeta = 10^2$ and $\zeta = 10^3$. Figure 2a shows the EMEs of the NZNNQ-D model. In this figure, all cases start at $t = 0$ from a large error value and arrive to a small one in the interval $[10^{-4}, 10^{-1}]$ at $t = 2$ when $\lambda = 10$ and at $t = 0.2$ when $\lambda = 10^2$. Additionally, as the value of ζ rises, the overall value of EMEs is dropping. To put it another way, the ZNNQ-D model converges to a minimum value for two distinct ICs, verifying Theorem 4.5, while the values of λ and ζ affect the rate of convergence of the EMEs and the total value of the EMEs, respectively. Fig. 2b shows the (1.3) error of the NZNNQ-D model. The findings shown there are the same as the results shown in Figure 2a, demonstrating that solving (2.8) is equivalent to solving (1.3). The trajectories of the solutions generated by the model are presented in Figure 2e–2h. The real part and the three imaginary parts of the solutions are depicted in these figures, respectively. These graphs demonstrate that the solutions generated by the models are identical and that they converge to the TSOL in a way that is compatible with the convergence tendency of the associated EMEs.

In the SE 5.2, we have that $\tilde{A}(t), \tilde{B}(t), \tilde{C}(t) \in \mathbb{H}^{10\times10}$. That is, $m = n = 10$ in the QDSE problem of (1.3). For the ZNNQ-D model, we have the following findings under IC1 and IC2 for $\lambda = 10$ and $\lambda = 10^2$. Figure 1c shows the EMEs of the ZNNQ-D model. In this figure, all cases start at $t = 0$ from a large error value and arrive to a small one in the interval $[10^{-5}, 10^{-3}]$ at $t = 2$ when $\lambda = 10$ and at $t = 0.2$ when $\lambda = 10^2$. In other words, the ZNNQ-D model converges to a minimum value for two different ICs, validating Theorem 3.2, whereas the value of λ influences the rate at which the EMEs converge. Figure 1d shows the (1.3) error of the ZNNQ-D model. The findings shown there are the same as the results shown in Figure 1c, demonstrating that solving (2.8) is equivalent to solving (1.3). The trajectories of the solutions generated by the model are presented in Figure 1i–1l. The real part and the three imaginary parts of the solutions are depicted in these figures, respectively. These graphs demonstrate that the solutions generated by the models are identical and that they converge to the TSOL in a way that is compatible with the convergence tendency of the associated EMEs.

For the NZNNQ-D model in the SE 5.2, we have the following findings under IC1 and a constant noise for $\lambda = 10$ and $\lambda = 10^2$ with $\zeta = 10$, $\zeta = 10^2$ and $\zeta = 10^3$. Figure 2a shows the EMEs of the NZNNQ-D model. In this figure, all cases start at $t = 0$ from a large error value and arrive to a small one in the interval $[10^{-5}, 10^{0}]$ at $t = 2$ when $\lambda = 10$ and at $t = 0.2$ when $\lambda = 10^2$. Additionally, as the value of ζ rises, the overall value of EMEs is dropping. To put it another way, the ZNNQ-D model converges to a minimum value for two distinct ICs, verifying Theorem 4.5, while the values of λ and

ζ affect the rate of convergence of the EMEs and the total value of the EMEs, respectively. Figure 2d shows the (1.3) error of the NZNNQ-D model. The findings shown there are the same as the results shown in Figure 2c, demonstrating that solving (2.8) is equivalent to solving (1.3). The trajectories of the solutions generated by the model are presented in Figure 2i–2l. The real part and the three imaginary parts of the solutions are depicted in these figures, respectively. These graphs demonstrate that the solutions generated by the models are identical and that they converge to the TSOL in a way that is compatible with the convergence tendency of the associated EMEs.

Overall, the ZNNQ-D model works excellent in solving two different QDSE problems, and the NZNNQ-D model works excellent in solving two different QDSE problems under two different types of noises. It is important to note that the aforementioned discussion verifies the results of Theorems 3.2 and 4.5. Additionally, the total computational complexity of the ZNNQ-D and NZNNQ-D models, respectively, is $O((4mn)^3)$ and $O((8mn)^3)$. Although the ZNNQ-D model is less computationally complex overall than the NZNNQ-D model, it is unable to handle noise-polluted scenarios. Also, the computational complexity analysis presented in Section 3.3 proves that the ZNNQ-D model has half the computational complexity of the ZNN model presented in [45], while the ZNN model presented in [45] always assumes that $m = n = 2$ in (1.3). Therefore, we may conclude that the ZNNQ-D model has more advantages than the ZNN model presented in [45].

5.3. Applications to control of the SFM chaotic system

The ZNN design technique is not only useful for solving the QDSE, but it can also be applied to chaotic system control. One type of typical nonlinear system, chaotic systems [48], are employed in network and power systems [49, 50], and secure communications [51]. As a result, this section presents two controllers based on ZNNs as well as the SFM chaotic control system [52].

Following is a detailed introduction to the SFM [52]:

$$\begin{cases} \dot{x}_1(t) = \xi(x_2(t)), \\ \dot{x}_2(t) = -\xi(x_1(t))/3 + \xi(x_2(t))/2 - \eta^2\xi(x_2(t))\xi^2(x_3(t))/2, \\ \dot{x}_3(t) = -\xi(x_2(t)) - 0.6\xi(x_3(t)) + \eta\xi(x_2(t))\xi(x_3(t)). \end{cases} \tag{5.1}$$

After setting

$$\begin{aligned} X(t) &= \begin{bmatrix} x_1(t), & x_2(t), & x_3(t) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3, \quad \dot{X}(t) = \begin{bmatrix} \dot{x}_1(t) & \dot{x}_2(t) & \dot{x}_3(t) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3 \\ F(X(t)) &= \begin{bmatrix} \xi(x_2(t)) \\ -\xi(x_1(t))/3 + \xi(x_2(t))/2 - \eta^2\xi(x_2(t))\xi^2(x_3(t))/2 \\ -\xi(x_2(t)) - 0.6\xi(x_3(t)) + \eta\xi(x_2(t))\xi(x_3(t)) \end{bmatrix} \in \mathbb{R}^3 \end{aligned} \tag{5.2}$$

(5.1) may be rewritten in matrix form as follows:

$$\dot{X}(t) = F(X(t)). \tag{5.3}$$

When considering the following uncertainties $\Delta(X(t))$, noise $h(t)$ and the controller $U(t)$:

$$\begin{aligned} \Delta(X(t)) &= \begin{bmatrix} \xi(x_2(t)), & 2\psi(x_1(t)), & 3\xi(x_1(t))\psi(x_3(t)) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3, \\ h(t) &= \mathbf{1}_{3\times1} \odot (t/4 + 4) \in \mathbb{R}^3, \quad U(t) = \begin{bmatrix} u_1(t), & u_2(t), & u_3(t) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3 \end{aligned} \tag{5.4}$$

(5.3) is formulated as follows:

$$\dot{X}(t) = F(X(t)) + \Delta(X(t)) + h(t) + U(t). \tag{5.5}$$

When combining (5.5) with ZNN (1.1), we have that

$$U(t) = \dot{E}(t) = -\lambda E(t) - F(X(t)), \tag{5.6}$$

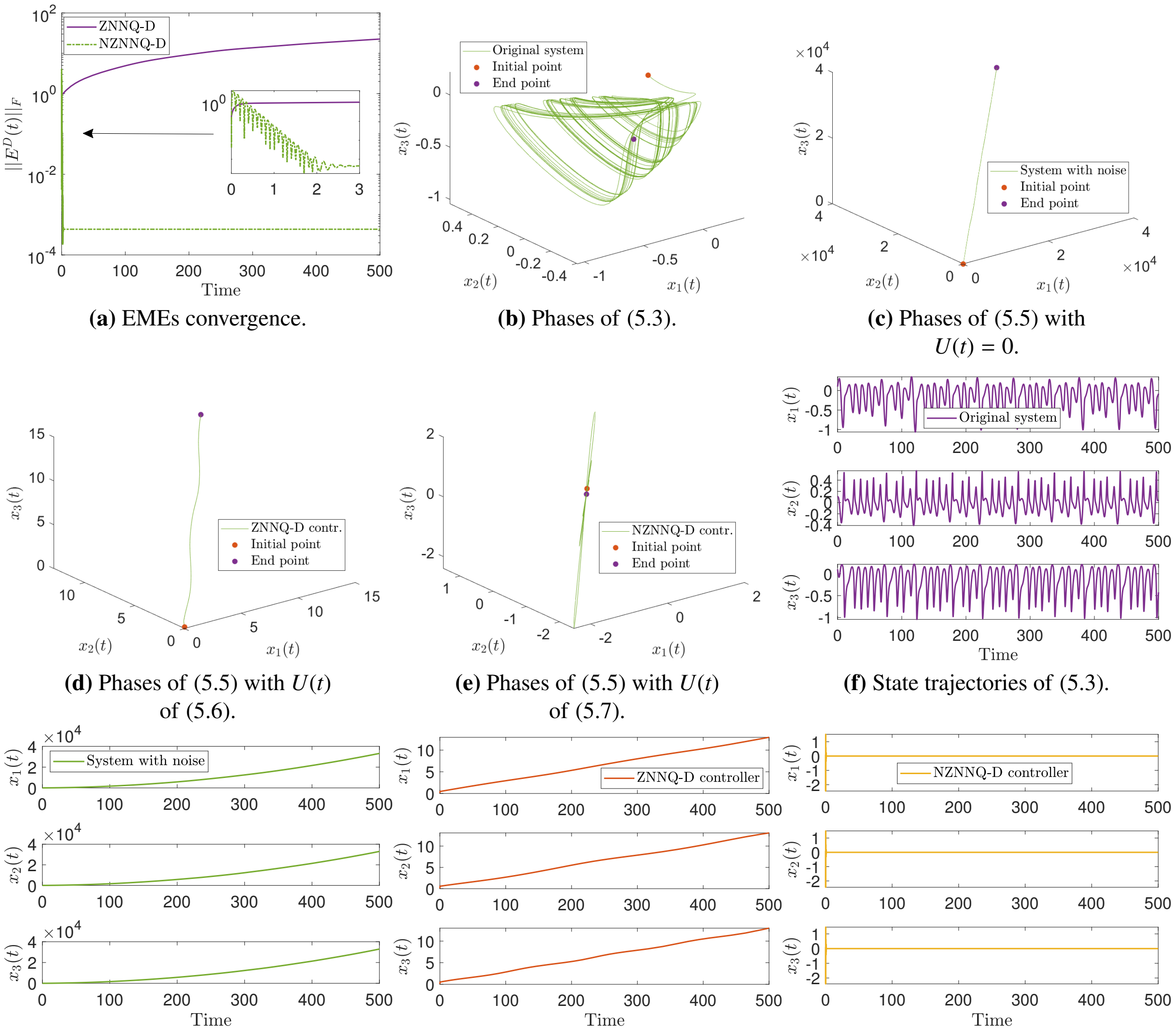
whereas, when combining (5.5) with NZNN (1.2), we have that

$$U(t) = \dot{E}(t) = -\lambda E(t) - \zeta \int_0^t E(\tau)\mathrm{d}\tau - F(X(t)). \tag{5.7}$$

It is crucial to note that we set $E(t) = X(t) - \mathbf{0}_{3\times1}$ in (5.6) and (5.7), and $\eta = 3$ in (5.2). Therefore, we set $\tilde{A}(t) = I_3$, $\tilde{B}(t) = 0$ and $\tilde{C}(t) = \mathbf{0}_3$ in the QDSE problem of (1.3). Also, the ICs have been set to $X(0) = 0.1 \odot \mathbf{1}_3$, $\mathbf{r}(0) = 0.1 \odot \mathbf{1}_3$ and $\mathbf{h}(0) = 0.1 \odot \mathbf{1}_6$, whereas the design parameters of the ZNN and NZNN have been set to $\zeta = 10^3$ and $\lambda = 10$. The results are presented in Figure 3.

Particularly, Figure 3a shows the EMEs of the ZNNQ-D and NZNNQ-D models. In this figure, both EMEs start at $t = 0$ from a high error value but only the NZNNQ-D model's EME arrives to a small one in the interval $[10^{-4}, 10^{-3}]$ at $t = 2$. In other words, the NZNNQ-D model converges to a minimum value, validating Theorem 3.2. Figure 3a shows the phases of (5.3) and Figure 3f shows their state (i.e., errors) trajectories. Figure 3b shows the phases of (5.5) with no controller, i.e. $U(t) = 0$, and Figure 3g shows their state trajectories. Fig. 3c shows the phases of (5.5) with $U(t)$ of (5.6), which is based on the ZNNQ-D model, and Figure 3h shows their state trajectories. Fig. 3d shows the phases of (5.5) with $U(t)$ of (5.7), which is based on the NZNNQ-D model, and Figure 3i shows their state trajectories.

The state of system (5.5) under controllers $U(t) = 0$ and $U(t)$ of (5.6) is unable to get close to zero in a three-dimensional space, as is seen from Figure 3g and 3h. The controller's (5.7) states and phases can both stabilize to zero at the same time. We anticipate that the the phase's end point of the controller is as close to zero as possible in order to obtain the least error possible. The preceding data show that the SFM system's phase under controller (5.7) is almost zero with a very little error. The experimental findings support the controller's (5.7) viability and efficacy. In this regard, while using SFM chaotic system control, a controller based on NZNNQ-D may also successfully eliminate any existing linear noise as well as any extra interferences.



(a) EMEs convergence.

(b) Phases of (5.3).

(c) Phases of (5.5) with $U(t) = 0$.

(d) Phases of (5.5) with $U(t)$ of (5.6).

(e) Phases of (5.5) with $U(t)$ of (5.7).

(f) State trajectories of (5.3).

(g) State trajectories of (5.5) with $U(t) = 0$.

(h) State trajectories of (5.5) with $U(t)$ of (5.6).

(i) State trajectories of (5.5) with $U(t)$ of (5.7).

Figure 3. EMEs, phases and state trajectories of the SFM systems (5.3) and (5.5).

6. Conclusions

One new ZNN model, termed ZNNQ-D, is introduced in (3.8) to solve the QDSE problem and one new NZNN model, termed NZNNQ-D, is introduced in (4.5) to solve the QDSE problem under different types of noises. The creation of such models has been backed by theoretical research and an examination of their computational complexity, in addition to simulation examples and application to control of the SFM chaotic system. The findings of the simulation examples and the application show that the models function superbly.

The established results open the door for future interesting study efforts in light of this. Here are a few topics to contemplate:

- The use of nonlinear ZNNs in dynamic quaternion issues may be investigated.

- It is doable to examine applying the finite-time ZNN theme to dynamic quaternion problems.
- Another area of research is using carefully selected design parameters stated in fuzzy settings to quicken the ZNN models' convergence.
- A potential research direction is to investigate the application and performance of ZNN controllers in chaotic systems other than SFM.

Use of AI tools declaration

The authors declare they have not used artificial intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by a Mega Grant from the Government of the Russian Federation within the framework of federal project No. 075-15-2021-584.

Conflict of interest

The authors declare no conflict of interest. Vasilios N. Katsikis is an editorial board member for AIMS Mathematics and was not involved in the editorial review or the decision to publish this article. All authors declare that there are no competing interests.

References

1. W. Li, L. Han, X. Xiao, B. Liao and C. Peng, A gradient-based neural network accelerated for vision-based control of an RCM-constrained surgical endoscope robot, *Neural Comput. Appl.*, **34** (2022), 1329–1343. https://doi.org/10.1007/s00521-021-06465-x

2. Z. Li, B. Liao, F. Xu, D. Guo, A new repetitive motion planning scheme with noise suppression capability for redundant robot manipulators, *IEEE Trans. Syst. Man Cybern. Syst.*, **50** (2020), 5244–5254.

3. J. Kurzak, A. Buttari, J. J. Dongarra, Solving systems of linear equations on the CELL processor using Cholesky factorization, *IEEE Trans. Parallel Distributed Syst.*, **19** (2008), 1175–1186.

4. H. R. Shaker, M. Tahavori, Control configuration selection for bilinear systems via generalised Hankel interaction index array, *Int. J. Control*, **88** (2015), 30–37. https://doi.org/10.1007/s00521-021-06465-x

5. S. D. Mourtas, V. N. Katsikis, C. Kasimis, Feedback control systems stabilization using a bio-inspired neural network, *EAI Endorsed Trans. AI Robotics*, **1** (2022), 1–13.

6. B. Liao, L. Han, X. Cao, S. Li, J. Li, Double integral-enhanced zeroing neural network with linear noise rejection for time-varying matrix inverse, *CAAI Trans. Intell. Technol.*, 1–14.

7. Q. Wei, N. Dobigeon, J. Tourneret, J. M. Bioucas-Dias, S. J. Godsill, R-FUSE: Robust fast fusion of multiband images based on solving a Sylvester equation, *IEEE Signal Process. Lett.*, **23** (2016), 1632–1636. https://doi.org/10.1109/LSP.2016.2608858

8. S. Dolgov, J. W. Pearson, D. V. Savostyanov, M. Stoll, Fast tensor product solvers for optimization problems with fractional differential equations as constraints, *Appl. Math. Comput.*, **273** (2016), 604–623. https://doi.org/10.1016/j.amc.2015.09.042

9. L. Huo, S. Yang, L. Jiao, S. Wang, J. Shi, Local graph regularized coding for salient object detection, *Infrared Phys. Technol.*, **77** (2016), 124–131. https://doi.org/10.1016/j.infrared.2016.05.002

10. X. Yan, M. Liu, L. Jin, S. Li, B. Hu, X. Zhang, et al., New zeroing neural network models for solving nonstationary Sylvester equation with verifications on mobile manipulators, *IEEE T. Ind. Inform.*, **15** (2019), 5011–5022.

11. T. Sarkar, K. Siarkiewicz, R. Stratton, Survey of numerical methods for solution of large systems of linear equations for electromagnetic field problems, *IEEE T. Antenn. Propag.*, **29** (1981), 847–856.

12. F. P. A. Beik, F. Saberi Movahed, S. Ahmadi-Asl, On the Krylov subspace methods based on tensor format for positive definite Sylvester tensor equations., *Numer. Linear Algebr.*, **23** (2016), 444–466.

13. C. Song, J. e. Feng, X. Wang, J. Zhao, Finite iterative method for solving coupled Sylvester-transpose matrix equations, *J. Appl. Math. Comput.*, **46** (2014), 351–372.

14. L. Xiao, J. Tao, W. Li, An arctan-type varying-parameter ZNN for solving time-varying complex Sylvester equations in finite time, *IEEE T. Ind. Inform.*, **18** (2022), 3651–3660.

15. W. R. Hamilton, On a new species of imaginary quantities, connected with the theory of quaternions, *P. Royal Irish Acad.*, **2** (1840), 424–434.

16. M. Joldeş, J. M. Muller, Algorithms for manipulating quaternions in floating-point arithmetic, in *2020 IEEE 27th Symposium on Computer Arithmetic (ARITH)*, IEEE, 2020, 48–55.

17. E. Özgür, Y. Mezouar, Kinematic modeling and control of a robot arm using unit dual quaternions, *Robot. Auton. Syst.*, **77** (2016), 66–73. https://doi.org/10.1016/j.robot.2015.12.005

18. D. Pavllo, C. Feichtenhofer, M. Auli, D. Grangier, Modeling human motion with quaternion-based neural networks, *Int. J. Comput. Vision*, **128** (2020), 855–872. https://doi.org/10.1007/s11263-019-01207-y

19. A. M. S. Goodyear, P. Singla, D. B. Spencer, Analytical state transition matrix for dual-quaternions for spacecraft pose estimation, in *AAS/AIAA Astrodynamics Specialist Conference, 2019*, Univelt Inc., 2020, 393–411.

20. M. E. Kansu, Quaternionic representation of electromagnetism for material media, *Int. J. Geom. Methods M.*, **16** (2019), 1950105. https://doi.org/10.1142/S0219887819501056

21. S. Giardino, Quaternionic quantum mechanics in real Hilbert space, *J. Geom. Phys.*, **158** (2020), 103956.

22. A. Szynal-Liana, I. Włoch, Generalized commutative quaternions of the Fibonacci type, *Boletín de la Sociedad Matemática Mexicana*, **28** (2022), 1.

23. L. Xiao, S. Liu, X. Wang, Y. He, L. Jia, Y. Xu, Zeroing neural networks for dynamic quaternion-valued matrix inversion, *IEEE Trans. Ind. Informatics*, **18** (2022), 1562–1571.

24. V. N. Kovalnogov, R. V. Fedorov, D. A. Demidov, M. A. Malyoshina, T. E. Simos, S. D. Mourtas, et al., Computing quaternion matrix pseudoinverse with zeroing neural networks, *AIMS Math.*, **8** (2023), 22875–22895. https://doi.org/10.3934/math.20231164

25. L. Xiao, P. Cao, W. Song, L. Luo, W. Tang, A fixed-time noise-tolerance ZNN model for time-variant inequality-constrained quaternion matrix least-squares problem, *IEEE T. Neur. Net. Lear.*, 1–10.

26. L. Xiao, Y. Zhang, W. Huang, L. Jia, X. Gao, A dynamic parameter noise-tolerant zeroing neural network for time-varying quaternion matrix equation with applications, *IEEE T. Neur. Net. Lear.*, 1–10.

27. R. Abbassi, H. Jerbi, M. Kchaou, T. E. Simos, S. D. Mourtas, V. N. Katsikis, Towards higher-order zeroing neural networks for calculating quaternion matrix inverse with application to robotic motion tracking, *Mathematics*, **11** (2023), 2756.

28. N. Tan, P. Yu, F. Ni, New varying-parameter recursive neural networks for model-free kinematic control of redundant manipulators with limited measurements, *IEEE T. Instrum. Meas.*, **71** (2022), 1–14.

29. V. N. Kovalnogov, R. V. Fedorov, D. A. Demidov, M. A. Malyoshina, T. E. Simos, V. N. Katsikis, et al., Zeroing neural networks for computing quaternion linear matrix equation with application to color restoration of images, *AIMS Math.*, **8** (2023), 14321–14339. https://doi.org/10.3934/math.2023733

30. Y. Zhang, S. S. Ge, Design and analysis of a general recurrent neural network model for time-varying matrix inversion, *IEEE T. Neur. Net.*, **16** (2005), 1477–1490.

31. Y. Chai, H. Li, D. Qiao, S. Qin, J. Feng, A neural network for Moore-Penrose inverse of time-varying complex-valued matrices, *Int. J. Comput. Intell. Syst.*, **13** (2020), 663–671.

32. W. Wu, B. Zheng, Improved recurrent neural networks for solving Moore-Penrose inverse of real-time full-rank matrix, *Neurocomputing*, **418** (2020), 221–231. https://doi.org/10.1016/j.neucom.2020.08.026

33. S. Qiao, Y. Wei, X. Zhang, Computing time-varying ML-weighted pseudoinverse by the Zhang neural networks, *Numer. Func. Anal. Opt.*, **41** (2020), 1672–1693.

34. X. Wang, P. S. Stanimirovic, Y. Wei, Complex ZFs for computing time-varying complex outer inverses, *Neurocomputing*, **275** (2018), 983–1001. https://doi.org/10.1016/j.neucom.2017.09.034

35. S. D. Mourtas, V. N. Katsikis, Exploiting the Black-Litterman framework through error-correction neural networks, *Neurocomputing*, **498** (2022), 43–58. https://doi.org/10.1016/j.neucom.2022.05.036

36. V. N. Kovalnogov, R. V. Fedorov, D. A. Generalov, A. V. Chukalin, V. N. Katsikis, S. D. Mourtas, et al., Portfolio insurance through error-correction neural networks, *Mathematics*, **10** (2022), 3335.

37. S. D. Mourtas, C. Kasimis, Exploiting mean-variance portfolio optimization problems through zeroing neural networks, *Mathematics*, **10** (2022), 3079. https://doi.org/10.3390/math10173079 .

38. W. Jiang, C. L. Lin, V. N. Katsikis, S. D. Mourtas, P. S. Stanimirović, T. E. Simos, Zeroing neural network approaches based on direct and indirect methods for solving the Yang–Baxter-like matrix equation, *Mathematics*, **10** (2022), 1950.

39. H. Jerbi, H. Alharbi, M. Omri, L. Ladhar, T. E. Simos, S. D. Mourtas, V. N. Katsikis, Towards higher-order zeroing neural network dynamics for solving time-varying algebraic Riccati equations, *Mathematics*, **10** (2022), 4490. https://doi.org/10.3390/math10234490

40. V. N. Katsikis, P. S. Stanimirović, S. D. Mourtas, L. Xiao, D. Karabasević, D. Stanujkić, Zeroing neural network with fuzzy parameter for computing pseudoinverse of arbitrary matrix, *IEEE T. Fuzzy Syst.*, **30** (2022), 3426–3435.

41. H. Alharbi, H. Jerbi, M. Kchaou, R. Abbassi, T. E. Simos, S. D. Mourtas, et al., Time-varying pseudoinversion based on full-rank decomposition and zeroing neural networks, *Mathematics*, **11** (2023), 600.

42. J. Dai, P. Tan, X. Yang, L. Xiao, L. Jia, Y. He, A fuzzy adaptive zeroing neural network with superior finite-time convergence for solving time-variant linear matrix equations, *Knowledge-Based Syst.*, **242** (2022), 108405. https://doi.org/10.1016/j.knosys.2022.108405

43. L. Jin, Y. Zhang, S. Li, Integration-enhanced Zhang neural network for real-time-varying matrix inversion in the presence of various kinds of noises, *IEEE T. Neur. Net. Lear.*, **27** (2016), 2615–2627. https://doi.org/10.1109/TNNLS.2015.2497715

44. F. Zhang, Quaternions and matrices of quaternions, *Linear Algebra Appl.*, **251** (1997), 21–57.

45. L. Xiao, W. Huang, X. Li, F. Sun, Q. Liao, L. Jia, et al., ZNNs with a varying-parameter design formula for dynamic Sylvester quaternion matrix equation, *IEEE T. Neur. Net. Lear.*, 1–11.

46. L. Huang, Q. W. Wang, Y. Zhang, The Moore-Penrose inverses of matrices over quaternion polynomial rings, *Linear Algebra Appl.*, **475** (2015), 45–61. https://doi.org/10.1016/j.laa.2015.02.033

47. A. K. Gupta, *Numerical methods using MATLAB*, MATLAB solutions series, Apress: Berkeley, CA, USA, New York, NY, 2014.

48. R. Zhang, X. Xi, H. Tian, Z. Wang, Dynamical analysis and finite-time synchronization for a chaotic system with hidden attractor and surface equilibrium, *Axioms*, **11** (2022), 579.

49. H. Su, R. Luo, M. Huang, J. Fu, Robust fixed time control of a class of chaotic systems with bounded uncertainties and disturbances, *Int. J. Control Autom. Syst.*, **20** (2022), 813–822. https://doi.org/10.1007/s12555-020-0782-1

50. J. Singer, Y. Wang, H. H. Bau, Controlling a chaotic system, *Phys. Rev. Lett.*, **66** (1991), 1123. https://doi.org/10.1103/PhysRevLett.66.1123

51. W. He, T. Luo, Y. Tang, W. Du, Y. Tian, F. Qian, Secure communication based on quantized synchronization of chaotic neural networks under an event-triggered strategy, *IEEE T. Neur. Net. Lear.*, **31** (2020), 3334–3345.

52. J. Sun, X. Zhao, J. Fang, Y. Wang, Autonomous memristor chaotic systems of infinite chaotic attractors and circuitry realization, *Nonlinear Dynam.*, **94** (2018), 2879–2887. https://doi.org/10.1007/s11071-018-4531-4