AIMS Mathematics, 8(11): 25966–25989.
DOI: 10.3934/math.20231323
Received: 13 July 2023
Revised: 28 August 2023
Accepted: 03 September 2023
Published: 08 September 2023

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

Research article

A novel quaternion linear matrix equation solver through zeroing neural networks with applications to acoustic source tracking

Vladislav N. Kovalnogov[1], Ruslan V. Fedorov[1], Igor I. Shepelev[1], Vyacheslav V. Sherkunov[1], Theodore E. Simos[1,2,3,4,5*], Spyridon D. Mourtas[6,7] and Vasilios N. Katsikis[6]

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

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

Abstract: Due to its significance in science and engineering, time-varying linear matrix equation (LME) problems have received a lot of attention from scholars. It is for this reason that the issue of finding the minimum-norm least-squares solution of the time-varying quaternion LME (ML-TQ-LME) is addressed in this study. This is accomplished using the zeroing neural network (ZNN) technique, which has achieved considerable success in tackling time-varying issues. In light of that, two new ZNN models are introduced to solve the ML-TQ-LME problem for time-varying quaternion matrices of arbitrary dimension. Two simulation experiments and two practical acoustic source tracking applications show that the models function superbly.

Keywords: quaternion; linear matrix equation; minimum-norm least-squares solution; dynamical system; zeroing neural network; acoustic source tracking
Mathematics Subject Classification: 65F20, 68T05

1. Introduction

It is widely acknowledged that time-varying linear matrix equations (LMEs) solution is a key issue that is encountered in many domains [1–4], including robot manipulators [4], cell processors [3], control system architecture [2], and linear least squares regression [1]. In the past, academics have typically solved LMEs by the use of classical iterative methods [5]. However, time-invariant and real-valued LMEs are the only ones for which iterative approaches are appropriate. Due to their restricted computational capacity, only few studies have been produced to handle time-invariant or time-varying LMEs in the complex domain [6, 7]. Therefore, iterative approaches are not the best option when tackling time-varying complex computing issues in real time.

In order to deal with time-varying tasks in real time, the zeroing neural network (ZNN) technique is introduced by Zhang *et al.* in [8]. 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 [9–12]. They are now used to solve problems involving generalized inversion [13–18], linear and quadratic programming [19–21], systems of nonlinear equations [22, 23], and LMEs [7, 24], among other issues. It is important to mention that ZNNs originated from the gradient neural network (or Hopfield network), whereas gradient neural networks are recently used to solve problems involving matrix inversion [25] and systems of nonlinear equations [26].

The issue of finding the minimum-norm least-squares solution of the time-varying quaternion LME (ML-TQ-LME) is addressed in this study. Hamilton first introduced quaternions, a non-commutative number system that expands on complex numbers, in 1843 [27]. They are useful for calculations requiring three-dimensional rotations in both theoretical and applied mathematics [28]. They are particularly important in several fields, including robotics [29, 30], computer modeling [31, 32], navigation [33], electromagnetism [34], quantum mechanics [35], and mathematical physics [36, 37]. Recently, there has been increased interest in the study of time-varying quaternion (TQ) problems that involve matrices, including the inversion of TQ matrices [38], the solution of the dynamic TQ Sylvester matrix equation [39], the resolution of the TQ constrained matrix least-squares problem [40], and the resolution of the TQ linear matrix equation for square matrices [41]. Combining the quaternion and the time-varying LME can get the TQ LME. The TQ LME means that three elements in the equation are all quaternions, including two known quaternions and one unknown quaternion to be solved. By solving the ML-TQ-LME rather than the TQ LME, we are able to determine the minimum-norm least-squares solution of the TQ LMEs, which is particularly interesting when several solutions exist. Be aware that the solution provided by ML-TQ-LME always exists and is unique. Additionally, chaotic system synchronization [40], mobile manipulator control [38,42], kinematically redundant manipulator of robotic joints [30, 43] and picture restoration [41, 44] are real-world uses of TQ matrices. One thing unites all of these studies: they all use the ZNN technique to arrive at the solution.

In this paper, the ZNN technique is used to address the ML-TQ-LME problem. Particularly, two new ZNN models are developed inline with the designs presented in [45] to solve the ML-TQ-LME problem for TQ matrices of arbitrary dimension. The one model, named ZNNQ-G, is based on the gradient design and the other, named ZNNQ-D, on the direct design. Through two simulation experiments and two practical acoustic source tracking applications, the models' efficacy will be evaluated. It is significant to note that the results demonstrate the models' excellent functionality. Considering that

since the TQ LMEs may be expanded to handle different types of matrix equations, the proposed models (i.e. ZNNQ-G and ZNNQ-D) can be extended to deal with other kinds of matrix equations, such as the static quaternion LME and the TQ inverse matrix equation. Last, by doing theoretical analysis and examining the computational complexity of each model that is described, this research study adds to the body of literature.

The major results of the paper are listed below.

(1) Two new ZNN models (i.e. ZNNQ-G and ZNNQ-D) for solving the ML-TQ-LME problem are presented.

(2) The suggested ZNNQ-G and ZNNQ-D models can be used with any TQ matrix.

(3) A theoretical investigation is conducted to support the models.

(4) To support the theoretical research, simulation experiments and practical acoustic source tracking applications are carried out.

The remainder of the paper is structured as follows. Section 2 presents preliminary information and the ML-TQ-LME problem. The ZNN model in line with gradient design is introduced in Section 3, while the ZNN model in line with 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 applications to acoustic source tracking are presented in Section 5. Lastly, final thoughts and comments are provided in Section 6.

2. Preliminary information and problem formulation

This part lays out some preliminary information about TQ matrices, the ML-TQ-LME issue, ZNNs and the notation that will be utilized throughout the remainder of the study along with the major results that will be covered.

A quaternion is a division algebra or skew-field over the real number field [46]. Let $\mathbb{H}$ be the set of quaternions, $\tilde{S}(t) = S_1(t) + S_2(t)\imath + S_3(t)\jmath + S_4(t)k \in \mathbb{H}^{m\times n}$ be a TQ matrix with coefficient matrices $S_i(t) \in \mathbb{R}^{m\times n}$ for $i = 1, 2, \cdots, 4$, and $t \in [0, t_f) \subseteq [0, +\infty)$ be the time. The conjugate transpose of $\tilde{S}(t)$ is the following [47]:

$$\tilde{S}^*(t) = S_1^{\mathrm{T}}(t) - S_2^{\mathrm{T}}(t)\imath - S_3^{\mathrm{T}}(t)\jmath - S_4^{\mathrm{T}}(t)k, \tag{2.1}$$

where $()^*$ is the conjugate transpose operator and $()^{\mathrm{T}}$ is the transpose operator. Consider the TQ matrix $\tilde{B}(t) \in \mathbb{H}^{n\times g}$ with coefficient matrices $B_i(t) \in \mathbb{R}^{n\times g}$ for $i = 1, 2, \cdots, 4$. The product of $\tilde{S}(t)$ and $\tilde{B}(t)$ is as follows:

$$\tilde{S}(t)\tilde{B}(t) = \tilde{V}(t) = V_1(t) + V_2(t)\imath + V_3(t)\jmath + V_4(t)k \in \mathbb{H}^{m\times g}, \tag{2.2}$$

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

$$\begin{aligned}
V_1(t)&=S_1(t)B_1(t)-S_2(t)B_2(t)-S_3(t)B_3(t)-S_4(t)B_4(t),\\
V_2(t)&=S_1(t)B_2(t)+S_2(t)B_1(t)+S_3(t)B_4(t)-S_4(t)B_3(t),\\
V_3(t)&=S_1(t)B_3(t)+S_3(t)B_1(t)+S_4(t)B_2(t)-S_2(t)B_4(t),\\
V_4(t)&=S_1(t)B_4(t)+S_4(t)B_1(t)+S_2(t)B_3(t)-S_3(t)B_2(t).
\end{aligned} \tag{2.3}$$

Given that the quaternion foundations are covered, the ML-TQ-LME problem can be stated as

follows:

$$\begin{cases} \tilde{A}(t)\tilde{X}(t) = \tilde{B}(t), & \tilde{X}(t) \in \mathbb{H}^{n\times g}, \tilde{A}(t) \in \mathbb{H}^{m\times n}, \tilde{B}(t) \in \mathbb{H}^{m\times g},\ m \geq n \\ \tilde{X}(t)\tilde{A}(t) = \tilde{B}(t), & \tilde{X}(t) \in \mathbb{H}^{g\times m}, \tilde{A}(t) \in \mathbb{H}^{m\times n}, \tilde{B}(t) \in \mathbb{H}^{g\times n},\ m < n, \end{cases} \tag{2.4}$$

in which $\tilde{X}(t)$ is the desired solution to the ML-TQ-LME problem.

Further, the ZNN technique will be used to address the ML-TQ-LME problem. Two main 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:

$$\dot{E}(t) = -\lambda E(t), \tag{2.5}$$

where $\dot{(\cdot)}$ is the time derivative operator. On top of that, one can change the model's convergence rate by adjusting the parameter $\lambda > 0$, which is a positive real number. As an example, any ZNN model will converge even more quickly with a bigger value of λ [48–50]. 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 (2.5). As a consequence, EME can be considered a tool for monitoring ZNN model learning.

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. Moreover, the vectorization process will be denoted as vec$(\cdot)$ and the Kronecker product will be denoted as $\otimes$. Last, $\|\cdot\|_{\mathrm{F}}$ will denote the matrix Frobenius norm and $\beta \geq 0$ is the Tikhonov regularization parameter. It is important to mention that the Tikhonov regularization parameter is frequently used to address singularity issues.

3. Gradient ZNN model for solving the ML-TQ-LME

In this section we shall develop a ZNN model, named ZNNQ-G, in line with the gradient design presented in [45] to solve the ML-TQ-LME problem for TQ matrices of any dimension.

3.1. The ZNNQ-G model

We suppose that $\tilde{B}(t) \in \mathbb{H}^{m\times g}$ when $m \geq n$, or $\tilde{B}(t) \in \mathbb{H}^{g\times n}$ when $n > m$, and $\tilde{A}(t) \in \mathbb{H}^{m\times n}$ are differentiable TQ matrices. The gradient approach converts the ML-TQ-LME problem of (2.4) into the following:

$$\begin{cases} \tilde{A}^*(t)\tilde{A}(t)\tilde{X}(t) = \tilde{A}^*(t)\tilde{B}(t), & \tilde{X}(t) \in \mathbb{H}^{n\times g}, \tilde{A}(t) \in \mathbb{H}^{m\times n}, \tilde{B}(t) \in \mathbb{H}^{m\times g},\ m \geq n \\ \tilde{X}(t)\tilde{A}(t)\tilde{A}^*(t) = \tilde{B}(t)\tilde{A}^*(t), & \tilde{X}(t) \in \mathbb{H}^{g\times m}, \tilde{A}(t) \in \mathbb{H}^{m\times n}, \tilde{B}(t) \in \mathbb{H}^{g\times n},\ n > m, \end{cases} \tag{3.1}$$

or equivalent,

$$\begin{cases} \tilde{D}(t)\tilde{X}(t) = \tilde{Q}(t),\ m \geq n \\ \tilde{X}(t)\tilde{D}(t) = \tilde{Q}(t),\ n > m, \end{cases} \tag{3.2}$$

where $\tilde{D}(t) = \tilde{A}^*(t)\tilde{A}(t) \in \mathbb{H}^{n\times n}$ and $\tilde{Q}(t) = \tilde{A}^*(t)\tilde{B}(t) \in \mathbb{H}^{n\times g}$ when $m \geq n$, otherwise $\tilde{D}(t) = \tilde{A}(t)\tilde{A}^*(t) \in \mathbb{H}^{m\times m}$ and $\tilde{Q}(t) = \tilde{B}(t)\tilde{A}^*(t) \in \mathbb{H}^{g\times m}$. It is important to note that $\tilde{X}(t)$ is the unknown TQ matrix to be found.

In line with (2.2) and (2.3), the following holds in the case of (3.2):

$$\begin{cases} \begin{cases} D_1(t)X_1(t) - D_2(t)X_2(t) - D_3(t)X_3(t) - D_4(t)X_4(t) = Q_1(t), \\ D_2(t)X_1(t) + D_1(t)X_2(t) - D_4(t)X_3(t) + D_3(t)X_4(t) = Q_2(t), \\ D_3(t)X_1(t) + D_4(t)X_2(t) + D_1(t)X_3(t) - D_2(t)X_4(t) = Q_3(t), \\ D_4(t)X_1(t) - D_3(t)X_2(t) + D_2(t)X_3(t) + D_1(t)X_4(t) = Q_4(t) \end{cases}, \ m \geq n \\ \begin{cases} X_1(t)D_1(t) - X_2(t)D_2(t) - X_3(t)D_3(t) - X_4(t)D_4(t) = Q_1(t), \\ X_2(t)D_1(t) + X_1(t)D_2(t) - X_4(t)D_3(t) + X_3(t)D_4(t) = Q_2(t), \\ X_3(t)D_1(t) + X_4(t)D_2(t) + X_1(t)D_3(t) - X_2(t)D_4(t) = Q_3(t), \\ X_4(t)D_1(t) - X_3(t)D_2(t) + X_2(t)D_3(t) + X_1(t)D_4(t) = Q_4(t) \end{cases}, \ n > m, \end{cases} \tag{3.3}$$

where $X_i(t)$ and $D_i(t)$ for $i = 1, \cdots, 4$ are the coefficient real matrices of $\tilde{X}(t)$ and $\tilde{D}(t)$, respectively. Then, setting

$$Z(t) = \begin{cases} \begin{bmatrix} D_1(t) & -D_2(t) & -D_3(t) & -D_4(t) \\ D_2(t) & D_1(t) & -D_4(t) & D_3(t) \\ D_3(t) & D_4(t) & D_1(t) & -D_2(t) \\ D_4(t) & -D_3(t) & D_2(t) & D_1(t) \end{bmatrix} \in \mathbb{R}^{4n\times 4n}, & m \geq n \\ \begin{bmatrix} D_1(t) & D_2(t) & D_3(t) & D_4(t) \\ -D_2(t) & D_1(t) & -D_4(t) & D_3(t) \\ -D_3(t) & D_4(t) & D_1(t) & -D_2(t) \\ -D_4(t) & -D_3(t) & D_2(t) & D_1(t) \end{bmatrix} \in \mathbb{R}^{4m\times 4m}, & n > m, \end{cases} \tag{3.4}$$

$$Y(t) = \begin{cases} \left[X_1^{\mathrm{T}}(t), X_2^{\mathrm{T}}(t), X_3^{\mathrm{T}}(t), X_4^{\mathrm{T}}(t)\right]^{\mathrm{T}} \in \mathbb{R}^{4n\times g}, & m \geq n \\ \left[X_1(t), X_2(t), X_3(t), X_4(t)\right] \in \mathbb{R}^{g\times 4m}, & n > m, \end{cases}$$

$$W(t) = \begin{cases} \left[Q_1^{\mathrm{T}}(t), Q_2^{\mathrm{T}}(t), Q_3^{\mathrm{T}}(t), Q_4^{\mathrm{T}}(t)\right]^{\mathrm{T}} \in \mathbb{R}^{4n\times g}, & m \geq n \\ \left[Q_1(t), Q_2(t), Q_3(t), Q_4(t)\right] \in \mathbb{R}^{g\times 4m}, & n > m, \end{cases}$$

we convert the problem of (3.2) into the next problem:

$$\begin{cases} Z(t)Y(t) = W(t), \ m \geq n \\ Y(t)Z(t) = W(t), \ n > m. \end{cases} \tag{3.5}$$

Thereafter, we consider the next EME:

$$E^G(t) = \begin{cases} Z(t)Y(t) - W(t), \ m \geq n \\ Y(t)Z(t) - W(t), \ n > m, \end{cases} \tag{3.6}$$

where its first derivative is:

$$\dot{E}^G(t) = \begin{cases} \dot{Z}(t)Y(t) + Z(t)\dot{Y}(t) - \dot{W}(t), \ m \geq n \\ Y(t)\dot{Z}(t) + \dot{Y}(t)Z(t) - \dot{W}(t), \ n > m. \end{cases} \tag{3.7}$$

The following is the outcome of addressing the ZNN dynamical system in terms of $\dot{Y}(t)$ when $E(t)$ and $\dot{E}(t)$ in (2.5) are substituted with $E^G(t)$ and $\dot{E}^G(t)$ defined in (3.6) and (3.7), respectively:

$$\begin{cases} Z(t)\dot{Y}(t) = -\lambda E^G(t) + \dot{W}(t) - \dot{Z}(t)Y(t), \ m \geq n \\ \dot{Y}(t)Z(t) = -\lambda E^G(t) + \dot{W}(t) - Y(t)\dot{Z}(t), \ n > m. \end{cases} \tag{3.8}$$

Thereafter, (3.8) may be made simpler with the use of vectorization and Kronecker product:

$$\begin{cases} (I_g \otimes Z(t))\mathrm{vec}(\dot{Y}(t)) = \mathrm{vec}(-\lambda E^G(t) + \dot{W}(t) - \dot{Z}(t)Y(t)), & m \geq n \\ (Z^{\mathrm{T}}(t) \otimes I_g)\mathrm{vec}(\dot{Y}(t)) = \mathrm{vec}(-\lambda E^G(t) + \dot{W}(t) - Y(t)\dot{Z}(t)), & n > m. \end{cases} \tag{3.9}$$

Furthermore, after setting:

$$\begin{aligned} G(t) &= \begin{cases} (I_g \otimes Z(t)) \in \mathbb{R}^{4ng \times 4ng}, & m \geq n = \mathrm{rank}(\tilde{A}(t)) \\ (I_g \otimes Z(t)) + \beta I_{4ng} \in \mathbb{R}^{4ng \times 4ng}, & m \geq n > \mathrm{rank}(\tilde{A}(t)) \\ (Z^{\mathrm{T}}(t) \otimes I_g) \in \mathbb{R}^{4mg \times 4mg}, & n > m = \mathrm{rank}(\tilde{A}(t)), \\ (Z^{\mathrm{T}}(t) \otimes I_g) + \beta I_{4mg} \in \mathbb{R}^{4mg \times 4mg}, & n > m > \mathrm{rank}(\tilde{A}(t)), \end{cases} \\ Q(t) &= \begin{cases} \mathrm{vec}(-\lambda E^G(t) + \dot{W}(t) - \dot{Z}(t)Y(t)) \in \mathbb{R}^{4ng}, & m \geq n \\ \mathrm{vec}(-\lambda E^G(t) + \dot{W}(t) - Y(t)\dot{Z}(t)) \in \mathbb{R}^{4mg}, & n > m, \end{cases} \\ \mathbf{y}(t) &= \begin{cases} \mathrm{vec}(Y(t)) \in \mathbb{R}^{4ng}, & m \geq n \\ \mathrm{vec}(Y(t)) \in \mathbb{R}^{4mg}, & n > m, \end{cases} \quad \dot{\mathbf{y}}(t) = \begin{cases} \mathrm{vec}(\dot{Y}(t)) \in \mathbb{R}^{4ng}, & m \geq n \\ \mathrm{vec}(\dot{Y}(t)) \in \mathbb{R}^{4mg}, & n > m, \end{cases} \end{aligned} \tag{3.10}$$

we get at the next ZNN model:

$$G(t)\dot{\mathbf{y}}(t) = Q(t), \tag{3.11}$$

where $G(t)$ is an invertible mass matrix. The suggested ZNN model to be employed in addressing the ML-TQ-LME problem of (2.4) is the dynamic model of (3.11), referred to as ZNNQ-G.

3.2. ZNNQ-G model theoretical analysis

This section presents the ZNNQ-G (3.11) model's examination of convergence and stability.

Theorem 3.1. *Let $W(t) \in \mathbb{R}^{4n \times g}$ and $Z(t) \in \mathbb{R}^{4n \times 4n}$ when $m \geq n$, and $W(t) \in \mathbb{R}^{g \times 4m}$ and $Z(t) \in \mathbb{R}^{4m \times 4m}$ when $n > m$. Also, suppose that $W(t)$ and $Z(t)$ are differentiable. Then, the system* (3.8) *converges to the theoretical solution (TSOL) $\hat{Y}(t)$ of the LME* (3.5) *and the solution is stable, in line with Lyapunov.*

Proof. The replacement $\bar{Y}(t) := \hat{Y}(t) - Y(t)$ entails $Y(t) = \hat{Y}(t) - \bar{Y}(t)$, whereas $\hat{Y}(t)$ is the TSOL. The time-derivative of $Y(t)$ is $\dot{Y}(t) = \dot{\hat{Y}}(t) - \dot{\bar{Y}}(t)$. Note that

$$\begin{cases} Z(t)\hat{Y}(t) - W(t) = \mathbf{0}_{4n \times g}, \ m \geq n \\ \hat{Y}(t)Z(t) - W(t) = \mathbf{0}_{g \times 4m}, \ n > m, \end{cases} \tag{3.12}$$

and its first derivative

$$\begin{cases} Z(t)\dot{\hat{Y}}(t) + \dot{Z}(t)\hat{Y}(t) - \dot{W}(t) = \mathbf{0}_{4n \times g}, \ m \geq n \\ \dot{\hat{Y}}(t)Z(t) + \hat{Y}(t)\dot{Z}(t) - \dot{W}(t) = \mathbf{0}_{g \times 4m}, \ n > m. \end{cases} \tag{3.13}$$

Consequently, one can confirm the following after replacing $Y(t) = \hat{Y}(t) - \bar{Y}(t)$ with (3.6):

$$\hat{E}^G(t) = \begin{cases} Z(t)\hat{Y}(t) - Z(t)\bar{Y}(t) - W(t), \ m \geq n \\ \hat{Y}(t)Z(t) - \bar{Y}(t)Z(t) - W(t), \ n > m. \end{cases} \tag{3.14}$$

In addition, the dynamics of (2.5) yield

$$\dot{\hat{E}}^G(t) = \begin{cases} Z(t)\dot{\hat{Y}}(t) - Z(t)\dot{\bar{Y}}(t) - \dot{W}(t) + \dot{Z}(t)\hat{Y}(t) - \dot{Z}(t)\bar{Y}(t) = -\lambda\hat{E}^G(t), \ m \geq n \\ \dot{\hat{Y}}(t)Z(t) - \dot{\bar{Y}}(t)Z(t) - \dot{W}(t) + \hat{Y}(t)\dot{Z}(t) - \bar{Y}(t)\dot{Z}(t) = -\lambda\hat{E}^G(t), \ n > m. \end{cases} \tag{3.15}$$

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

$$\mathcal{L}(t) = \frac{1}{2}\left\|\hat{E}^G(t)\right\|_{\mathrm{F}}^2 = \frac{1}{2}\mathrm{Tr}\left(\hat{E}^G(t)\left(\hat{E}^G(t)\right)^{\mathrm{T}}\right). \tag{3.16}$$

The following identities can then be confirmed:

$$\dot{\mathcal{L}}(t) = \frac{2\mathrm{Tr}\left(\left(\hat{E}^G(t)\right)^{\mathrm{T}}\dot{\hat{E}}^G(t)\right)}{2} = \mathrm{Tr}\left(\left(\hat{E}^G(t)\right)^{\mathrm{T}}\dot{\hat{E}}^G(t)\right) = -\lambda\mathrm{Tr}\left(\left(\hat{E}^G(t)\right)^{\mathrm{T}}\hat{E}^G(t)\right). \tag{3.17}$$

As a consequence, it holds

$$\begin{aligned} &\frac{\mathrm{d}\mathcal{L}(\bar{Y}(t),t)}{\mathrm{d}t}\begin{cases} <0, & \hat{E}^G(t) \neq 0 \\ =0, & \hat{E}^G(t) = 0, \end{cases} \\ \Leftrightarrow &\dot{\mathcal{L}}(t)\begin{cases} <0, & \begin{cases} Z(t)\hat{Y}(t) - W(t) - Z(t)\bar{Y}(t) \neq 0, \ m \geq n \\ \hat{Y}(t)Z(t) - W(t) - \bar{Y}(t)Z(t) \neq 0, \ n > m. \end{cases} \\ =0, & \begin{cases} Z(t)\hat{Y}(t) - W(t) - Z(t)\bar{Y}(t) = 0, \ m \geq n \\ \hat{Y}(t)Z(t) - W(t) - \bar{Y}(t)Z(t) = 0, \ n > m. \end{cases} \end{cases} \\ \Leftrightarrow &\dot{\mathcal{L}}(t)\begin{cases} <0, & \bar{Y}(t) \neq 0 \\ =0, & \bar{Y}(t) = 0. \end{cases} \end{aligned} \tag{3.18}$$

Due to the fact that $\bar{Y}(t)$ is the equilibrium point of (3.15) and $E^G(0) = 0$, the following holds:

$$\frac{\mathrm{d}\mathcal{L}(\bar{Y}(t),t)}{\mathrm{d}t} \leq 0, \quad \forall\ \bar{Y}(t) \neq 0. \tag{3.19}$$

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

Theorem 3.2. *Let $\tilde{B}(t) \in \mathbb{H}^{m\times g}$ when $m \geq n$, or $\tilde{B}(t) \in \mathbb{H}^{g\times n}$ when $n > m$, and $\tilde{A}(t) \in \mathbb{H}^{m\times n}$ be differentiable TQ matrices. At every $t \in [0, t_f) \subseteq [0, +\infty)$, the ZNNQ-G model* (3.11) *converges exponentially to the TSOL for every initial value* $\mathbf{y}(0)$ *that one may take into consideration.*

Proof. First, the ML-TQ-LME problem of (2.4) is converted into the problem of (3.1), based on the gradient design presented in [45]. Second, using the matrices $\tilde{A}(t)$ and $\tilde{B}(t)$, we create the matrices $Z(t)$ and $W(t)$ in (3.4), where $W(t) \in \mathbb{R}^{4n\times g}$ and $Z(t) \in \mathbb{R}^{4n\times 4n}$ when $m \geq n$, and $W(t) \in \mathbb{R}^{g\times 4m}$ and $Z(t) \in \mathbb{R}^{4m\times 4m}$ when $n > m$. As a result, we convert the problem of (3.1) into the problem of (3.5). Third, to solve the problem of (3.5), the EME of (3.6) is declared. Then, for zeroing (3.6), the model (3.8) is deployed in line with the ZNN theme (2.5). According to Theorem 3.1, $Y(t) \to \hat{Y}(t)$ when $t \to \infty$ for any choice of initial value. So, the model (3.8) converges to the TSOL of the ML-TQ-LME (2.4). Fourth, the model (3.8) is simplified into the ZNNQ-G model (3.11) using the Kronecker product and vectorization. As an alternative version of (3.8), for every initial value $\mathbf{y}(0)$, the ZNNQ-G model (3.11) also converges to the TSOL $\hat{\mathbf{y}}(t)$ when $t \to \infty$. Thereafter, the proof is finished. □

3.3. ZNNQ-G model computational complexity

The complexity of creating and addressing (3.11) adds to the ZNNQ-G's total computational complexity. Particularly, the computational complexity of creating (3.11) is $O((4ng)^2)$ when $m \geq n$ operations and $O((4mg)^2)$ operations when $n > m$ because at every iteration we conduct $(4ng)^2$ multiplications and $4ng$ additions/subtractions when $m \geq n$ and $(4mg)^2$ multiplications and $4mg$ additions/subtractions when $n > m$. 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 (3.11) is $O((4ng)^3$ as it necessitates a $(4ng) \times (4ng)$ matrix when $m \geq n$, and $O((4mg)^3$ as it necessitates a $(4mg) \times (4mg)$ matrix when $n > m$. So, the ZNNQ-G model's total computational complexity is $O((4ng)^3)$ when $m \geq n$ and $O((4mg)^3)$ when $n > m$.

4. Direct ZNN model for solving the ML-TQ-LME

In this section we shall develop a ZNN model, named ZNNQ-D, in line with the direct design presented in [45] to solve the ML-TQ-LME problem for TQ matrices of any dimension.

4.1. The ZNNQ-D model

We suppose that $\tilde{B}(t) \in \mathbb{H}^{m\times g}$ when $m \geq n$, or $\tilde{B}(t) \in \mathbb{H}^{g\times n}$ when $n > m$, and $\tilde{A}(t) \in \mathbb{H}^{m\times n}$ are differentiable TQ matrices. The direct approach converts the ML-TQ-LME problem of (2.4) into the following:

$$\begin{cases} \tilde{X}(t) = \tilde{A}^{\dagger}(t)\tilde{B}(t), & \tilde{X}(t) \in \mathbb{H}^{n\times g}, \tilde{A}(t) \in \mathbb{H}^{m\times n}, \tilde{B}(t) \in \mathbb{H}^{m\times g},\ m \geq n \\ \tilde{X}(t) = \tilde{B}(t)\tilde{A}^{\dagger}(t), & \tilde{X}(t) \in \mathbb{H}^{g\times m}, \tilde{A}(t) \in \mathbb{H}^{m\times n}, \tilde{B}(t) \in \mathbb{H}^{g\times n},\ n > m, \end{cases} \tag{4.1}$$

where $\tilde{X}(t)$ is the desired solution and $\tilde{A}^{\dagger}(t)$ is the Moore-Penrose inverse of $\tilde{A}(t)$. That is, the direct approach computes the solution generated directly by the Moore-Penrose inverse of $A(t)$. If $\tilde{A}^{\dagger}(t)$ is also thought of as an unknown $\tilde{C}(t)$, (4.1) can be rewritten as follows:

$$\begin{cases} \begin{cases} \tilde{A}^{*}(t)\tilde{A}(t)\tilde{C}(t) = \tilde{A}^{*}(t), & \tilde{A}(t) \in \mathbb{H}^{m\times n}, \tilde{C}(t) \in \mathbb{H}^{n\times m}, \\ \tilde{X}(t) = \tilde{C}(t)\tilde{B}(t), & \tilde{C}(t) \in \mathbb{H}^{n\times m}, \tilde{X}(t) \in \mathbb{H}^{n\times g}, \tilde{B}(t) \in \mathbb{H}^{m\times g} \end{cases}, \ m \geq n \\ \begin{cases} \tilde{C}(t)\tilde{A}(t)\tilde{A}^{*}(t) = \tilde{A}^{*}(t), & \tilde{A}(t) \in \mathbb{H}^{m\times n}, \tilde{C}(t) \in \mathbb{H}^{n\times m}, \\ \tilde{X}(t) = \tilde{B}(t)\tilde{C}(t), & \tilde{C}(t) \in \mathbb{H}^{n\times m}, \tilde{X}(t) \in \mathbb{H}^{g\times m}, \tilde{B}(t) \in \mathbb{H}^{g\times n} \end{cases}, \ n > m, \end{cases} \tag{4.2}$$

or equivalent,

$$\begin{cases} \begin{cases} \tilde{D}(t)\tilde{C}(t) = \tilde{A}^*(t), & \tilde{A}(t) \in \mathbb{H}^{m\times n}, \tilde{C}(t) \in \mathbb{H}^{n\times m}, \\ \tilde{X}(t) = \tilde{C}(t)\tilde{B}(t), & \tilde{C}(t) \in \mathbb{H}^{n\times m}, \tilde{X}(t) \in \mathbb{H}^{n\times g}, \tilde{B}(t) \in \mathbb{H}^{m\times g} \end{cases}, \ m \geq n \\ \begin{cases} \tilde{C}(t)\tilde{D}(t) = \tilde{A}^*(t), & \tilde{A}(t) \in \mathbb{H}^{m\times n}, \tilde{C}(t) \in \mathbb{H}^{n\times m}, \\ \tilde{X}(t) = \tilde{B}(t)\tilde{C}(t), & \tilde{C}(t) \in \mathbb{H}^{n\times m}, \tilde{X}(t) \in \mathbb{H}^{g\times m}, \tilde{B}(t) \in \mathbb{H}^{g\times n} \end{cases}, \ n > m, \end{cases} \tag{4.3}$$

where $\tilde{D}(t) = \tilde{A}^*(t)\tilde{A}(t) \in \mathbb{H}^{n\times n}$ when $m \geq n$, otherwise $\tilde{D}(t) = \tilde{A}(t)\tilde{A}^*(t) \in \mathbb{H}^{m\times m}$. It is crucial to note that $\tilde{C}(t)$ and $\tilde{X}(t)$ are both the unknown TQ matrices to be found.

According to (2.2), (2.3) and (2.1), the next system is satisfied in the case of (4.3):

$$\begin{cases} \begin{cases} D_1(t)C_1(t) - D_2(t)C_2(t) - D_3(t)C_3(t) - D_4(t)C_4(t) = A_1^{\mathrm{T}}(t), \\ D_2(t)C_1(t) + D_1(t)C_2(t) - D_4(t)C_3(t) + D_3(t)C_4(t) = -A_2^{\mathrm{T}}(t), \\ D_3(t)C_1(t) + D_4(t)C_2(t) + D_1(t)C_3(t) - D_2(t)C_4(t) = -A_3^{\mathrm{T}}(t), \\ D_4(t)C_1(t) - D_3(t)C_2(t) + D_2(t)C_3(t) + D_1(t)C_4(t) = -A_4^{\mathrm{T}}(t), \\ X_1(t) = C_1(t)B_1(t) - C_2(t)B_2(t) - C_3(t)B_3(t) - C_4(t)B_4(t), \\ X_2(t) = C_2(t)B_1(t) + C_1(t)B_2(t) - C_4(t)B_3(t) + C_3(t)B_4(t), \\ X_3(t) = C_3(t)B_1(t) + C_4(t)B_2(t) + C_1(t)B_3(t) - C_2(t)B_4(t), \\ X_4(t) = C_4(t)B_1(t) - C_3(t)B_2(t) + C_2(t)B_3(t) + C_1(t)B_4(t) \end{cases}, \ m \geq n \\ \begin{cases} C_1(t)D_1(t) - C_2(t)D_2(t) - C_3(t)D_3(t) - C_4(t)D_4(t) = A_1^{\mathrm{T}}(t), \\ C_2(t)D_1(t) + C_1(t)D_2(t) - C_4(t)D_3(t) + C_3(t)D_4(t) = -A_2^{\mathrm{T}}(t), \\ C_3(t)D_1(t) + C_4(t)D_2(t) + C_1(t)D_3(t) - C_2(t)D_4(t) = -A_3^{\mathrm{T}}(t), \\ C_4(t)D_1(t) - C_3(t)D_2(t) + C_2(t)D_3(t) + C_1(t)D_4(t) = -A_4^{\mathrm{T}}(t), \\ X_1(t) = B_1(t)C_1(t) - B_2(t)C_2(t) - B_3(t)C_3(t) - B_4(t)C_4(t), \\ X_2(t) = B_2(t)C_1(t) + B_1(t)C_2(t) - B_4(t)C_3(t) + B_3(t)C_4(t), \\ X_3(t) = B_3(t)C_1(t) + B_4(t)C_2(t) + B_1(t)C_3(t) - B_2(t)C_4(t), \\ X_4(t) = B_4(t)C_1(t) - B_3(t)C_2(t) + B_2(t)C_3(t) + B_1(t)C_4(t) \end{cases}, \ n > m, \end{cases} \tag{4.4}$$

where $X_i(t), C_i(t), D_i(t), A_i(t)$ and $B_i(t)$ for $i = 1, \cdots, 4$ are the coefficient real matrices of

$\tilde{X}(t), \tilde{C}(t), \tilde{D}(t), \tilde{A}(t)$ and $\tilde{B}(t)$, respectively. Then, setting

$$Z(t) = \begin{cases} \begin{bmatrix} D_1(t) & -D_2(t) & -D_3(t) & -D_4(t) \\ D_2(t) & D_1(t) & -D_4(t) & D_3(t) \\ D_3(t) & D_4(t) & D_1(t) & -D_2(t) \\ D_4(t) & -D_3(t) & D_2(t) & D_1(t) \end{bmatrix} \in \mathbb{R}^{4n\times 4n}, & m \geq n \\ \begin{bmatrix} D_1(t) & D_2(t) & D_3(t) & D_4(t) \\ -D_2(t) & D_1(t) & -D_4(t) & D_3(t) \\ -D_3(t) & D_4(t) & D_1(t) & -D_2(t) \\ -D_4(t) & -D_3(t) & D_2(t) & D_1(t) \end{bmatrix} \in \mathbb{R}^{4m\times 4m}, & n > m, \end{cases} \quad K(t) = \begin{cases} \begin{bmatrix} B_1(t) & B_2(t) & B_3(t) & B_4(t) \\ -B_2(t) & B_1(t) & -B_4(t) & B_3(t) \\ -B_3(t) & B_4(t) & B_1(t) & -B_2(t) \\ -B_4(t) & -B_3(t) & B_2(t) & B_1(t) \end{bmatrix} \in \mathbb{R}^{4m\times 4g}, & m \geq n \\ \begin{bmatrix} B_1(t) & -B_2(t) & -B_3(t) & -B_4(t) \\ B_2(t) & B_1(t) & -B_4(t) & B_3(t) \\ B_3(t) & B_4(t) & B_1(t) & -B_2(t) \\ B_4(t) & -B_3(t) & B_2(t) & B_1(t) \end{bmatrix} \in \mathbb{R}^{4g\times 4n}, & n > m, \end{cases}$$

$$R(t) = \begin{cases} \begin{bmatrix} C_1(t), C_2(t), C_3(t), C_4(t) \end{bmatrix} \in \mathbb{R}^{n\times 4m}, & m \geq n \\ \begin{bmatrix} C_1^{\mathrm{T}}(t), C_2^{\mathrm{T}}(t), C_3^{\mathrm{T}}(t), C_4^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{4n\times m}, & n > m, \end{cases} \quad L(t) = \begin{cases} \begin{bmatrix} C_1^{\mathrm{T}}(t), C_2^{\mathrm{T}}(t), C_3^{\mathrm{T}}(t), C_4^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{4n\times m}, & m \geq n \\ \begin{bmatrix} C_1(t), C_2(t), C_3(t), C_4(t) \end{bmatrix} \in \mathbb{R}^{n\times 4m}, & n > m, \end{cases}$$

$$Y(t) = \begin{cases} \begin{bmatrix} X_1(t), X_2(t), X_3(t), X_4(t) \end{bmatrix} \in \mathbb{R}^{n\times 4g}, & m \geq n \\ \begin{bmatrix} X_1^{\mathrm{T}}(t), X_2^{\mathrm{T}}(t), X_3^{\mathrm{T}}(t), X_4^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{4g\times m}, & n > m, \end{cases} \quad W(t) = \begin{cases} \begin{bmatrix} A_1(t), -A_2(t), -A_3(t), -A_4(t) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{4n\times m}, & m \geq n \\ \begin{bmatrix} A_1^{\mathrm{T}}(t), -A_2^{\mathrm{T}}(t), -A_3^{\mathrm{T}}(t), -A_4^{\mathrm{T}}(t) \end{bmatrix} \in \mathbb{R}^{n\times 4m}, & n > m, \end{cases} \tag{4.5}$$

we convert the problem of (4.3) into the next problem:

$$\begin{cases} \begin{cases} Z(t)L(t) = W(t), \\ Y(t) = R(t)K(t) \end{cases}, & m \geq n \\ \begin{cases} L(t)Z(t) = W(t), \\ Y(t) = K(t)R(t) \end{cases}, & n > m. \end{cases} \tag{4.6}$$

Thereafter, we consider the next EME:

$$E^D(t) = \begin{cases} \begin{cases} E_1^D(t) = Z(t)L(t) - W(t), \\ E_2^D(t) = Y(t) - R(t)K(t) \end{cases}, & m \geq n \\ \begin{cases} E_1^D(t) = L(t)Z(t) - W(t), \\ E_2^D(t) = Y(t) - K(t)R(t) \end{cases}, & n > m, \end{cases} \tag{4.7}$$

where its first derivative is:

$$\dot{E}^D(t) = \begin{cases} \begin{cases} \dot{E}_1^D(t) = \dot{Z}(t)L(t) + Z(t)\dot{L}(t) - \dot{W}(t), \\ \dot{E}_2^D(t) = \dot{Y}(t) - \dot{R}(t)K(t) - R(t)\dot{K}(t) \end{cases}, & m \geq n \\ \begin{cases} \dot{E}_1^D(t) = \dot{L}(t)Z(t) + L(t)\dot{Z}(t) - \dot{W}(t), \\ \dot{E}_2^D(t) = \dot{Y}(t) - \dot{K}(t)R(t) - K(t)\dot{R}(t) \end{cases}, & n > m. \end{cases} \tag{4.8}$$

The following is the outcome of addressing the ZNN dynamical system in terms of $\dot{R}(t)$, $\dot{L}(t)$ and $\dot{Y}(t)$

when $E(t)$ and $\dot{E}(t)$ in (2.5) are substituted with $E^D(t)$ and $\dot{E}^D(t)$ defined in (4.7) and (4.8), respectively:

$$\begin{cases} \begin{cases} Z(t)\dot{L}(t) = -\lambda E_1^D(t) + \dot{W}(t) - \dot{Z}(t)L(t), \\ \dot{Y}(t) - \dot{R}(t)K(t) = -\lambda E_2^D(t) + R(t)\dot{K}(t) \end{cases}, \ m \geq n \\ \begin{cases} \dot{L}(t)Z(t) = -\lambda E_1^D(t) + \dot{W}(t) - L(t)\dot{Z}(t), \\ \dot{Y}(t) - K(t)\dot{R}(t) = -\lambda E_2^D(t) + \dot{K}(t)R(t) \end{cases}, \ n > m. \end{cases} \tag{4.9}$$

Thereafter, (4.9) may be made simpler with the use of vectorization and Kronecker product:

$$\begin{cases} \begin{cases} (I_m \otimes Z(t))\mathrm{vec}(\dot{L}(t)) = \mathrm{vec}(-\lambda E_1^D(t) + \dot{W}(t) - \dot{Z}(t)L(t)), \\ \mathrm{vec}(\dot{Y}(t)) - (K^{\mathrm{T}}(t) \otimes I_n)\mathrm{vec}(\dot{R}(t)) = \mathrm{vec}(-\lambda E_2^D(t) + R(t)\dot{K}(t)) \end{cases}, \ m \geq n \\ \begin{cases} (Z^{\mathrm{T}}(t) \otimes I_n)\mathrm{vec}(\dot{L}(t)) = \mathrm{vec}(-\lambda E_1^D(t) + \dot{W}(t) - L(t)\dot{Z}(t)), \\ \mathrm{vec}(\dot{Y}(t)) - (I_m \otimes K(t))\mathrm{vec}(\dot{R}(t)) = \mathrm{vec}(-\lambda E_2^D(t) + \dot{K}(t)R(t)) \end{cases}, \ n > m. \end{cases} \tag{4.10}$$

It is significant to note that identical elements, but in different positions, can be found in the vectors $\mathrm{vec}(\dot{L}(t))$ and $\mathrm{vec}(\dot{R}(t))$. In other words, it is possible to further simplify (4.10) by rewriting the vector $\mathrm{vec}(\dot{L}(t))$ in terms of $\mathrm{vec}(\dot{R}(t))$. As a consequence, it is feasible to create the next equation that substitutes $\mathrm{vec}(\dot{L}(t))$ in (4.10):

$$\mathrm{vec}(\dot{L}(t)) = J\mathrm{vec}(\dot{R}(t)), \tag{4.11}$$

where $J \in \mathbb{R}^{4mn \times 4mn}$ is an operational matrix that may be computed utilizing the algorithmic process presented in Algorithm 1. Notice that the notations in Algorithm 1 follow the usual MATLAB function theme [51].

Algorithm 1 Matrix J calculation.

Input: The rows m and columns n numbers of a matrix $A \in \mathbb{R}^{m\times n}$.

procedure OPE_MAT_J(m, n)
  if $m \geq n$ **then**
    $g = 4$
  else
    $g = m$
  end if
  Set $b = [\ ]$ and $r = 4mn$
  for $j = 1 : n$ **do**
    $b = [b, j : ng : r]$
  end for
  Set $b = sort(b)$, $c = length(b)$ and $J = zeros(r)$
  for $i = 1 : g$ **do**
    Set $d = b + (i - 1)n$ and $k = cr(i - 1)$
    for $j = 1 : c$ **do**
      $J(d(j) + k + (j - 1)r) = 1$
    end for
  end for
  return J
end procedure

Output: The matrix J.

By using (4.11), we can further simplify (4.10) as follows:

$$\begin{cases} \begin{cases} (I_m\otimes Z(t))J\mathrm{vec}(\dot{R}(t))=\mathrm{vec}(-\lambda E_1^D(t)+\dot{W}(t)-\dot{Z}(t)L(t)), \\ \mathrm{vec}(\dot{Y}(t))-(K^{\mathrm{T}}(t)\otimes I_n)\mathrm{vec}(\dot{R}(t))=\mathrm{vec}(-\lambda E_2^D(t)+R(t)\dot{K}(t)) \end{cases}, & m\geq n \\ \begin{cases} (Z^{\mathrm{T}}(t)\otimes I_n)J\mathrm{vec}(\dot{R}(t))=\mathrm{vec}(-\lambda E_1^D(t)+\dot{W}(t)-L(t)\dot{Z}(t)), \\ \mathrm{vec}(\dot{Y}(t))-(I_m\otimes K(t))\mathrm{vec}(\dot{R}(t))=\mathrm{vec}(-\lambda E_2^D(t)+\dot{K}(t)R(t)) \end{cases}, & n>m. \end{cases} \tag{4.12}$$

In addition, once the following has been set:

$$U(t)=\begin{bmatrix} (I_m\otimes Z(t))J & \mathbf{0}_{4mn\times 4ng} \\ K^{\mathrm{T}}(t)\otimes I_n & I_{4ng} \end{bmatrix}\in\mathbb{R}^{4n(m+g)\times 4n(m+g)}, \quad V(t)=\begin{bmatrix} (Z^{\mathrm{T}}(t)\otimes I_n)J & \mathbf{0}_{4mn\times 4mg} \\ I_m\otimes K(t) & I_{4mg} \end{bmatrix}\in\mathbb{R}^{4m(n+g)\times 4m(n+g)}$$

$$G(t)=\begin{cases} U(t), & m\geq n=\mathrm{rank}(\tilde{A}(t)) \\ U(t)+\beta I_{4n(m+g)}, & m\geq n>\mathrm{rank}(\tilde{A}(t)) \\ V(t), & n>m=\mathrm{rank}(\tilde{A}(t)), \\ V(t)+\beta I_{4m(n+g)}, & n>m>\mathrm{rank}(\tilde{A}(t)), \end{cases}, \quad Q(t)=\begin{cases} \begin{bmatrix} \mathrm{vec}(-\lambda E_1^D(t)+\dot{W}(t)-\dot{Z}(t)L(t)) \\ \mathrm{vec}(-\lambda E_2^D(t)+R(t)\dot{K}(t)) \end{bmatrix}\in\mathbb{R}^{4n(m+g)}, & m\geq n \\ \begin{bmatrix} \mathrm{vec}(-\lambda E_1^D(t)+\dot{W}(t)-L(t)\dot{Z}(t)) \\ \mathrm{vec}(-\lambda E_2^D(t)+\dot{K}(t)R(t)) \end{bmatrix}\in\mathbb{R}^{4m(n+g)}, & n>m, \end{cases}$$

$$\mathbf{r}(t)=\begin{cases} \left[\mathrm{vec}(R(t))^{\mathrm{T}}, \mathrm{vec}(Y(t))^{\mathrm{T}}\right]^{\mathrm{T}}\in\mathbb{R}^{4n(m+g)}, & m\geq n \\ \left[\mathrm{vec}(R(t))^{\mathrm{T}}, \mathrm{vec}(Y(t))^{\mathrm{T}}\right]^{\mathrm{T}}\in\mathbb{R}^{4m(n+g)}, & n>m, \end{cases} \quad \dot{\mathbf{r}}(t)=\begin{cases} \left[\mathrm{vec}(\dot{R}(t))^{\mathrm{T}}, \mathrm{vec}(\dot{Y}(t))^{\mathrm{T}}\right]^{\mathrm{T}}\in\mathbb{R}^{4n(m+g)}, & m\geq n \\ \left[\mathrm{vec}(\dot{R}(t))^{\mathrm{T}}, \mathrm{vec}(\dot{Y}(t))^{\mathrm{T}}\right]^{\mathrm{T}}\in\mathbb{R}^{4m(n+g)}, & n>m, \end{cases} \tag{4.13}$$

we get at the next ZNN model:

$$G(t)\dot{\mathbf{r}}(t) = Q(t), \tag{4.14}$$

where $G(t)$ is an invertible mass matrix. The suggested ZNN model to be employed in addressing the ML-TQ-LME problem of (2.4) is the dynamic model of (4.14), referred to as ZNNQ-D.

4.2. ZNNQ-D model theoretical analysis

This section presents the ZNNQ-D (4.14) model's examination of convergence and stability.

Theorem 4.1. *Let $Z(t) \in \mathbb{R}^{4n\times 4n}$, $K(t) \in \mathbb{R}^{4m\times 4g}$, $L(t) \in \mathbb{R}^{4n\times m}$ and $W(t) \in \mathbb{R}^{4n\times m}$ when $m \geq n$, and $Z(t) \in \mathbb{R}^{4m\times 4m}$, $K(t) \in \mathbb{R}^{4g\times 4n}$, $L(t) \in \mathbb{R}^{n\times 4m}$ and $W(t) \in \mathbb{R}^{n\times 4m}$ when $n > m$. Also, suppose that $Z(t), K(t), W(t)$ and $L(t)$ are differentiable. Then, the system* (4.9) *converges to the TSOLs $\hat{L}(t)$ and $\hat{Y}(t)$ of the LMEs* (4.6)*, and the solutions are stable, in line with Lyapunov.*

Proof. The proof has been taken out since it resembles the proof of Theorem 3.1. □

Theorem 4.2. *Let $\tilde{B}(t) \in \mathbb{H}^{m\times g}$ when $m \geq n$, or $\tilde{B}(t) \in \mathbb{H}^{g\times n}$ when $n > m$, and $\tilde{A}(t) \in \mathbb{H}^{m\times n}$ be differentiable TQ matrices. At every $t \in [0, t_f) \subseteq [0, +\infty)$, the ZNNQ-D model* (4.14) *model converges exponentially to the TSOL for any initial value* $\mathbf{r}(0)$ *that one may take into consideration.*

Proof. The proof has been taken out since, if Theorem 3.1 is replaced with Theorem 4.1, it resembles the proof of Theorem 3.2. □

4.3. ZNNQ-D model computational complexity

The complexity of creating and addressing (4.14) adds to the ZNNQ-D's total computational complexity. Particularly, the computational complexity of creating (4.14) is $O((4n(m+g))^2)$ when $m \geq n$ operations and $O((4m(n+g))^2)$ operations when $n > m$ because at every iteration we conduct $(4n(m+g))^2$ multiplications and $4n(m+g)$ additions/subtractions when $m \geq n$ and $(4m(n+g))^2$ multiplications and $4m(n+g)$ additions/subtractions when $n > m$. 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 (4.14) is $O((4n(m+g))^3$ as it necessitates a $4n(m+g)\times 4n(m+g)$ matrix when $m \geq n$, and $O((4m(n+g))^3$ as it necessitates a $4m(n+g)\times 4m(n+g)$ matrix when $n > m$. So, the ZNNQ-D model's total computational complexity is $O((4n(m+g))^3)$ when $m \geq n$ and $O((4m(n+g))^3)$ when $n > m$.

5. Simulation experiments

This section will outline two applications for acoustic source tracking as well as two simulation examples (SEs). The following includes a few key justifications. The ZNN design parameter λ is utilized with values 10 and 100 in SEs and with value 100 in the applications. The initial conditions (ICs) of the ZNNQ-G and ZNNQ-D models have been set as follows:

- IC1: $\tilde{X}(0) = \mathbf{0}_{n\times g}$ when $m \geq n$, $\tilde{X}(0) = \mathbf{0}_{g\times m}$ when $n > m$, and $\tilde{C}(0) = \mathbf{0}_{n\times m}$.
- IC2: $\tilde{X}(0) = \tilde{A}^*(0)\tilde{B}(0)$ when $m \geq n$, and $\tilde{X}(0) = \tilde{B}(0)\tilde{A}^*(0)$ when $n > m$, and $\tilde{C}(0) = \tilde{A}^*(0)$.

The notation ML Error in the y-label of the figures corresponds to the following error:

$$\begin{cases} \left\|\tilde{X}(t) - \tilde{A}^\dagger(t)\tilde{B}(t) - (I_n - \tilde{A}^\dagger(t)\tilde{A}(t))\tilde{X}(0)\right\|_{\mathrm{F}}, & m \geq n \\ \left\|\tilde{X}(t) - \tilde{B}(t)\tilde{A}^\dagger(t) - \tilde{X}(0)(I_m - \tilde{A}(t)\tilde{A}^\dagger(t))\right\|_{\mathrm{F}}, & n > m. \end{cases} \tag{5.1}$$

For simplicity, we have set $\eta(t) = \cos(t)$ and $\zeta(t) = \sin(t)$. Lastly, in all SEs and applications, computations are performed using the MATLAB ode solver, `ode15s`, with a time interval of $[0, 10]$. It is crucial to note that we utilize the `ode15s` with its standard 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 input matrix $\tilde{A}(t)$ coefficients are set to

$$A_1(t) = \begin{bmatrix} 3\zeta(t)+1 & 4 & 4 \\ 3\eta(t)+2 & 5 & 5 \\ 2 & \eta(t)+3 & \eta(t)+3 \\ 2\eta(t)+5 & 1 & 1 \end{bmatrix}, \quad A_2(t) = \begin{bmatrix} 5 & 2+2\zeta(t) & 2+2\zeta(t) \\ 3\zeta(t)-2 & 3 & 3 \\ -2\eta(t)+5 & 5+\eta(t) & 5+\eta(t) \\ 9 & 1+2\zeta(t)+1 & 1\zeta(t) \end{bmatrix},$$

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

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

$$B_1(t) = \begin{bmatrix} \eta(t) & 2+\zeta(t) \\ 5 & 1+\zeta(t) \\ 3 & 2 \\ 7+\zeta(t) & 2 \end{bmatrix}, \quad B_2(t) = \begin{bmatrix} \eta(t)+4 & 2 \\ \eta(t) & 8 \\ \zeta(t)+3 & \zeta(t) \\ 4 & \eta(t) \end{bmatrix},$$

$$B_3(t) = \begin{bmatrix} 3\zeta(t)+2 & 6 \\ 5 & 8 \\ 4 & \zeta(t) \\ 7 & 6 \end{bmatrix}, \quad B_4(t) = \begin{bmatrix} 2\zeta(t)+3 & 4 \\ \eta(t) & sin(t)+1 \\ 9 & 4 \\ 2 & \eta(t) \end{bmatrix}.$$

As a result, $\tilde{A}(t) \in \mathbb{H}^{4\times 3}$ with $\text{rank}(\tilde{A}(t)) = 2$ and $\tilde{B}(t) \in \mathbb{H}^{4\times 2}$. Generated results are presented in Figure 1 and 2, where we have set $\beta = 10^{-6}$.

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

$$A_1(t) = \begin{bmatrix} 2\eta(t)+2 & 5 & 3\eta(t)+2 & 3 & 8 \\ 2 & \zeta(t)+3 & 2\eta(t)+5 & 8 & 3 \end{bmatrix}, \; A_2(t) = \begin{bmatrix} 4 & 2\zeta(t)+1 & 2\zeta(t)-3 & 7 & 2 \\ -3\eta(t)+4 & \eta(t)+6 & 9 & 2\zeta(t)+2 & 4 \end{bmatrix},$$

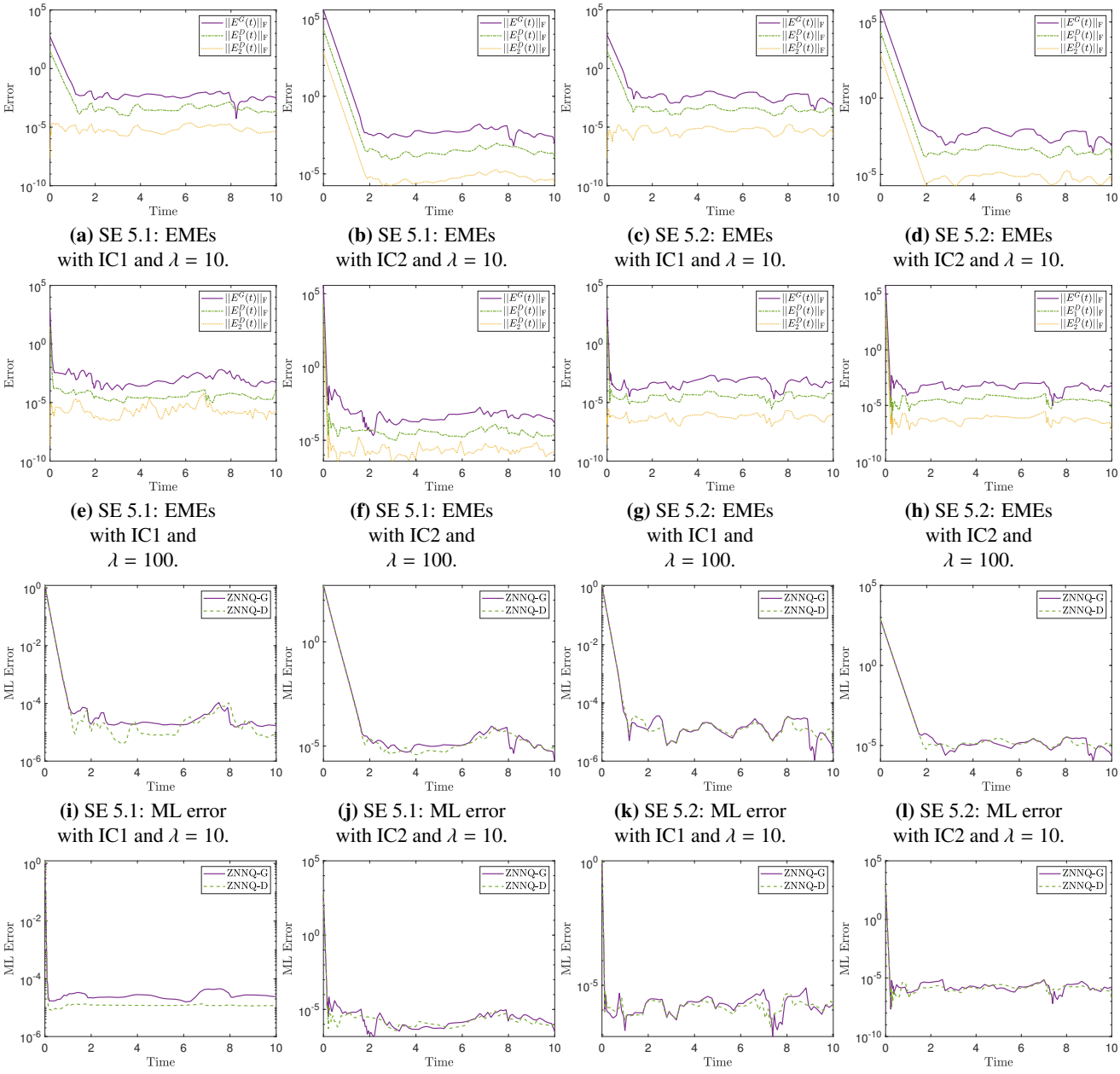
$$A_3(t) = \begin{bmatrix} 2\zeta(t)+2 & 6 & 2\zeta(t)+1 & 7 & 5 \\ -\eta(t)+2 & 3 & -\eta(t)+5 & 7 & 6 \end{bmatrix}, \quad A_4(t) = \begin{bmatrix} 5 & 2\zeta(t)+1 & 6 & 6 & 6 \\ 3\eta(t)+2 & 4 & 8 & 3\zeta(t)+1 & 1 \end{bmatrix},$$

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

$$B_1(t) = \begin{bmatrix} 3 & \zeta(t)+2 & 3 & \zeta(t)+1 & 1 \\ \eta(t) & 1 & \zeta(t)+2 & 4 & 1 \\ \eta(t) & 5 & \zeta(t)+2 & 6 & 1 \end{bmatrix}, \quad B_2(t) = \begin{bmatrix} \eta(t)+2 & 7 & \eta(t) & 3 & 3 \\ \eta(t)+3 & \zeta(t) & 3 & \eta(t) & 3 \\ \eta(t)+5 & \zeta(t) & 5 & \eta(t) & 3 \end{bmatrix},$$

$$B_3(t) = \begin{bmatrix} 2\zeta(t)+2 & 6 & 5 & 8 & 1 \\ 3 & \zeta(t) & 5 & 6 & 5 \\ 3 & \zeta(t) & 7 & 6 & 6 \end{bmatrix}, \quad B_4(t) = \begin{bmatrix} 2\zeta(t)+4 & 7 & 6 & \zeta(t)+1 & 1 \\ 7 & 4 & 3 & 2 & 1 \\ 8 & 7 & 3 & 2 & 8 \end{bmatrix}.$$

Therefore, $\tilde{A}(t) \in \mathbb{H}^{2\times 5}$ with $\text{rank}(\tilde{A}(t)) = 2$ and $\tilde{B}(t) \in \mathbb{H}^{3\times 5}$. The results are presented in Figures 1 and 2.



(a) SE 5.1: EMEs with IC1 and $\lambda = 10$.

(b) SE 5.1: EMEs with IC2 and $\lambda = 10$.

(c) SE 5.2: EMEs with IC1 and $\lambda = 10$.

(d) SE 5.2: EMEs with IC2 and $\lambda = 10$.

(e) SE 5.1: EMEs with IC1 and $\lambda = 100$.

(f) SE 5.1: EMEs with IC2 and $\lambda = 100$.

(g) SE 5.2: EMEs with IC1 and $\lambda = 100$.

(h) SE 5.2: EMEs with IC2 and $\lambda = 100$.

(i) SE 5.1: ML error with IC1 and $\lambda = 10$.

(j) SE 5.1: ML error with IC2 and $\lambda = 10$.

(k) SE 5.2: ML error with IC1 and $\lambda = 10$.

(l) SE 5.2: ML error with IC2 and $\lambda = 10$.

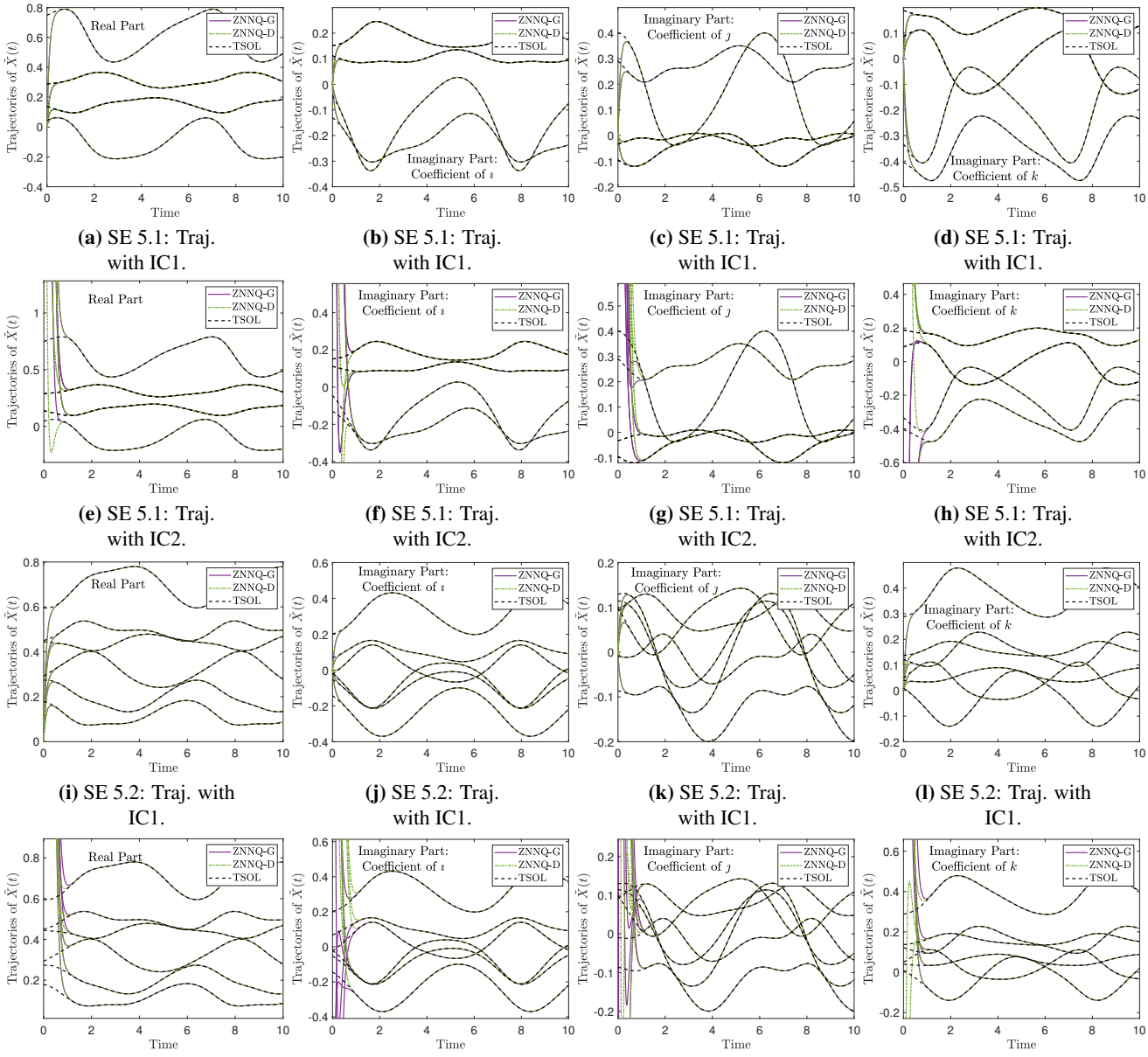
(m) SE 5.1: ML error with IC1 and $\lambda = 100$.

(n) SE 5.1: ML error with IC2 and $\lambda = 100$.

(o) SE 5.2: ML error with IC1 and $\lambda = 100$.

(p) SE 5.2: ML error with IC2 and $\lambda = 100$.

Figure 1. EMEs and ML error (5.1) in SEs 5.1-5.2.



(a) SE 5.1: Traj. with IC1. **(b)** SE 5.1: Traj. with IC1. **(c)** SE 5.1: Traj. with IC1. **(d)** SE 5.1: Traj. with IC1.

(e) SE 5.1: Traj. with IC2. **(f)** SE 5.1: Traj. with IC2. **(g)** SE 5.1: Traj. with IC2. **(h)** SE 5.1: Traj. with IC2.

(i) SE 5.2: Traj. with IC1. **(j)** SE 5.2: Traj. with IC1. **(k)** SE 5.2: Traj. with IC1. **(l)** SE 5.2: Traj. with IC1.

(m) SE 5.2: Traj. with IC2. **(n)** SE 5.2: Traj. with IC2. **(o)** SE 5.2: Traj. with IC2. **(p)** SE 5.2: Traj. with IC2.

Figure 2. $\tilde{X}(t)$ real and imaginary parts trajectories with IC1, IC2 and $\lambda = 10$ in SEs 5.1 and 5.2.

5.2. Discussion on simulation examples results

The performance of the ZNNQ-G (3.11) and ZNNQ-D (4.14) models for solving the ML-TQ-LME problem of (2.4) is investigated throughout the SEs 5.1 and 5.2. Each SE is associated with a unique ML-TQ-LME problem that is specified by the proper pair of matrices $\tilde{A}(t)$ and $\tilde{B}(t)$.

In the case of SE 5.1, we have that $\tilde{A}(t) \in \mathbb{H}^{4\times 3}$ with $\text{rank}(\tilde{A}(t)) = 2$ and $\tilde{B}(t) \in \mathbb{H}^{4\times 2}$. That is, $m \geq n > \text{rank}(\tilde{A}(t))$. Figure 1a and 1e, respectively, show the EMEs of the ZNNQ-G and ZNNQ-D models under IC1 for $\lambda = 10$ and $\lambda = 100$, while Figure 1b and 1f show the EMEs of the ZNNQ-G

and ZNNQ-D models under IC2 for $\lambda = 10$ and $\lambda = 100$, respectively. In these figures, all models start at $t = 0$ from a high error value and converge to a low error value in the interval $[10^{-5}, 10^{-3}]$ at $t = 2$ when $\lambda = 10$ and at $t = 0.2$ $\lambda = 100$. The ZNNQ-D model, however, converges to its minimum value more quickly than the ZNNQ-G model and has a lower overall error value. Notice that the minimum value close to 10^{-5} is expected because of the MATLAB `ode15s` default double precision arithmetic. We can further confirm that the ZNN models converge to their minimal value regardless of the ICs, and that they do so even more swiftly the larger the value of λ is. The ML errors in Figure 1i, 1m, 1j and 1n follow the convergence tendency of their corresponding EME in Figure 1a, 1e, 1b and 1f, respectively. However, unlike the comparable EMEs, the models' ML errors have a similar lower overall error value and converge to their minimum value at the same rate. For $\lambda = 10$, the trajectories of the solutions generated by the models under IC1 are presented in Figure 2a–2d and under IC2 are presented in Figure 2e–2h. The real part and the three imaginary parts of the solutions are depicted in these figures, respectively, and their relation to the TSOL is shown. These figures show that the solutions produced by the models match up to the TSOL and that they converge to the TSOL in a manner consistent with the corresponding ML error's convergence tendency.

In the case of SE 5.2, we have that $\tilde{A}(t) \in \mathbb{H}^{2\times 5}$ with $\text{rank}(\tilde{A}(t)) = 2$ and $\tilde{B}(t) \in \mathbb{H}^{3\times 5}$. That is, $n > m = \text{rank}(\tilde{A}(t))$. Figure 1c and 1g, respectively, show the EMEs of the ZNNQ-G and ZNNQ-D models under IC1 for $\lambda = 10$ and $\lambda = 100$, while Figure 1d and 1h show the EMEs of the ZNNQ-G and ZNNQ-D models under IC2 for $\lambda = 10$ and $\lambda = 100$, respectively. In these figures, all models start at $t = 0$ from a high error value and converge to a low error value in the interval $[10^{-5}, 10^{-3}]$ at $t = 2$ when $\lambda = 10$ and at $t = 0.2$ $\lambda = 100$. The ZNNQ-D model, however, converges to its minimum value more quickly than the ZNNQ-G model and has a lower overall error value. We can further confirm that the ZNN models converge to their minimal value regardless of the ICs, and that they do so even more swiftly the larger the value of λ is. The ML errors in Figure 1k, 1o, 1l and 1p follow the convergence tendency of their corresponding EME in Figure 1c, 1g, 1d and 1h, respectively. However, unlike the comparable EMEs, the models' ML errors have a similar lower overall error value and converge to their minimum value at the same rate. For $\lambda = 10$, the trajectories of the solutions generated by the models under IC1 are presented in Figure 2i–2l and under IC2 are presented in Figure 2m–2p. The real part and the three imaginary parts of the solutions are depicted in these figures, respectively, and their relation to the TSOL is shown. These figures show that the solutions produced by the models match up to the TSOL and that they converge to the TSOL in a manner consistent with the corresponding ML error's convergence tendency.

Overall, the ZNNQ-G and ZNNQ-D models work excellent in solving two different ML-TQ-LME problems, while the aforementioned discussion verifies the results of Theorems 3.1–4.2. Additionally, the total computational complexity of the ZNNQ-G and ZNNQ-D models, respectively, is $O((4ng)^3)$ and $O((4n(m+g))^3)$ when $m \geq n$, and $O((4mg)^3)$ and $O((4m(n+g))^3)$ when $n > m$. That is, the ZNNQ-G model has lower total computational complexity than the ZNNQ-D model. Therefore, even though both models have comparable accuracy, we may conclude that the ZNNQ-G model has more advantages than the ZNNQ-D model.

5.3. Applications to acoustic source tracking

The ZNNQ-G and ZNNQ-D models are utilized in this subsection to simulate acoustic source tracking based on the time delay of arrival (TDOA). TDOA is a technique for estimating the position

of the signal source based on the difference in signal arrival times at receivers placed in various locations [52]. TDOA has been widely used in a variety of fields, including videoconferencing [53], indoor positioning [54], navigation [55] and a ultra-wideband localization system [56], as a form of passive source localization. The localization of an acoustic source could be transformed into a task of addressing LMEs based on TDOA.

The signal source for the acoustic source localization is an acoustic source, and the receivers are microphones. These applications examine the issue of localizing moving acoustic sources, where the position of the acoustic sound is subject to time. To keep things simple, we look into the 2-dimensional localization, from which we can expand to the 3-dimensional localization. First, we define the coordinates of the acoustic source, $w(t)$, and the coordinates of the k microphones, C, as follows:

$$w(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \in \mathbb{R}^2, \quad C = \begin{bmatrix} x_1 & x_2 & \dots & x_k \\ y_1 & y_2 & \dots & y_k \end{bmatrix} \in \mathbb{R}^{2\times k}, \tag{5.2}$$

where the k microphones are incidentally positioned and fixed. The acoustic source moves in the first application along a circular path trajectory as time passes, and in the second application along an infinity-shaped path trajectory as time passes. Particularly, taken from [57], the following circular path trajectory is used in the first application (App. 1):

$$\begin{aligned} p_x(t) &= \psi\eta(2\pi\zeta(\pi t/(2T))^2 + \pi/6)/(2T), \\ p_y(t) &= \psi\zeta(2\pi\zeta(\pi t/(2T))^2 + \pi/6)/(2T), \end{aligned} \tag{5.3}$$

where we have set the task duration $T = 5$ and the design parameter $\psi = 5$. In the second application (App. 2), the following infinity-shaped path trajectory is used [57]:

$$\begin{aligned} p_x(t) &= -\psi\zeta(4\pi\zeta(\pi t/(2T))^2 + \pi/3)/(2T), \\ p_y(t) &= \psi\zeta(2\pi\zeta(\pi t/(2T))^2 + \pi/6)/(2T), \end{aligned} \tag{5.4}$$

where we have set the task duration $T = 10$ and the design parameter $\psi = 10$.

The following equations are then provided, taken from [52]:

$$\begin{aligned} s_i(t) = uT_i(t) &= \sqrt{(x_i - p_x(t))^2 + (y_i - p_y(t))^2}, \\ u\Delta T_i(t) = u(T_i(t) - T_1(t)) &= s_i(t) - s_1(t), \end{aligned} \tag{5.5}$$

where $i = 1, 2, \dots, k$, the speed of sound is $u = 340.29$ m/s, $s_i(t)$ is the distance between the acoustic source and the ith microphone, the time at which sound arrives at the ith microphone is $T_i(t)$, and $\Delta T_i(t)$ is the difference in arrival time among the 1th and ith microphones. After the derivation process presented in [52], we set:

$$A(t) = \begin{bmatrix} h_{31}(t) & h_{32}(t) \\ h_{41}(t) & h_{42}(t) \\ \vdots & \vdots \\ h_{k1}(t) & h_{k2}(t) \end{bmatrix}, \quad B(t) = \begin{bmatrix} -b_3(t) \\ -b_4(t) \\ \vdots \\ -b_k(t) \end{bmatrix}, \tag{5.6}$$

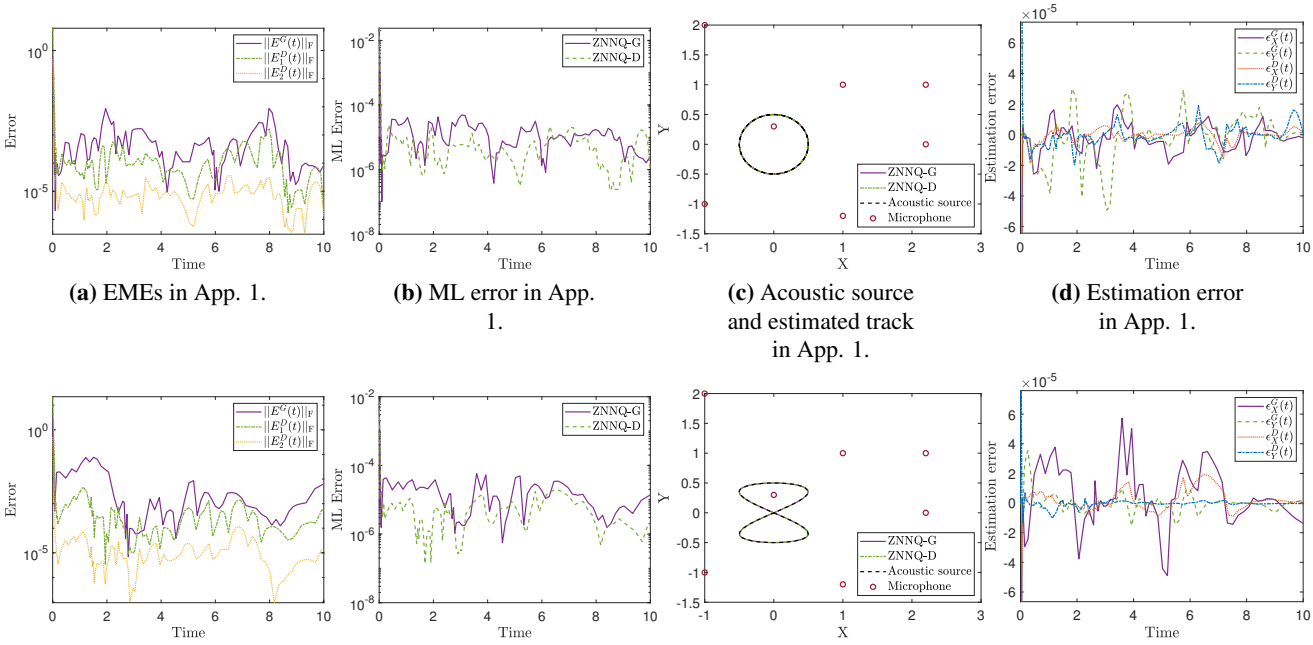
where

$$h_{i1}(t) = \frac{2(x_i - x_1)}{u\Delta T_i(t)} - \frac{2(x_2 - x_1)}{u\Delta T_2(t)}, \quad h_{i2}(t) = \frac{2(y_i - y_1)}{u\Delta T_i(t)} - \frac{2(y_2 - y_1)}{u\Delta T_2(t)},$$
$$b_i(t) = u\Delta T_i(t) - u\Delta T_2(t) + \frac{x_1^2 - x_i^2 + y_1^2 - y_i^2}{u\Delta T_i(t)} - \frac{x_1^2 - x_2^2 + y_1^2 - y_2^2}{u\Delta T_2(t)}, \tag{5.7}$$

for $i = 3, 4, \ldots, k$ with $k \geq 4$, to create the following equation:

$$A(t)w(t) = B(t), \tag{5.8}$$

where $w(t)$ is unknown. We chose the number of microphones to be 7 in both applications with $C = \begin{bmatrix} 1 & 2.2 & -1 & 2.2 & -1 & 1 & 0 \\ 1 & 1 & 2 & 0 & -1 & -1.2 & 0.3 \end{bmatrix} \in \mathbb{R}^{2\times 7}$, whereas we have set the initial state $w(0) = [0.42, 0.27]^{\mathrm{T}}$ in the first application and $w(0) = [-0.44, 0.25]^{\mathrm{T}}$ in the second application. Therefore, we set the ICs $\tilde{X}(0) = w(0)$ and $\tilde{C}(0) = \mathbf{0}_{2\times 5}$ in the ZNNQ-G and ZNNQ-D models. The results are presented in Figure 3.



(a) EMEs in App. 1. **(b)** ML error in App. 1. **(c)** Acoustic source and estimated track in App. 1. **(d)** Estimation error in App. 1.

(e) EMEs in App. 2. **(f)** ML error in App. 2. **(g)** Acoustic source and estimated track in App. 2. **(h)** Estimation error in App. 2.

Figure 3. EMEs, ML error (5.1), acoustic source path and the estimated track in applications 1 and 2.

In App. 1, where the acoustic source moves in a circular path trajectory, Figure 3a shows the EMEs of the ZNNQ-G and ZNNQ-D models for $\lambda = 100$, and in App. 2, where the acoustic source moves in an infinity-shaped path trajectory, Figure 3e shows the EMEs of the ZNNQ-G and ZNNQ-D models for $\lambda = 100$. In these figures, all models start at $t = 0$ from a high error value and converge to a low error value in the interval $[10^{-6}, 10^{-3}]$ at $t = 0.1$. The ZNNQ-D model, however, has a lower overall error value than the ZNNQ-G model. The ML errors in Figure 3b and 3f follow the convergence tendency

of their corresponding EMEs in Figure 3a and 3e. But unlike the comparable EMEs, the models' ML errors have a similar lower overall error value and converge to their minimum value at the same rate. The actual acoustic source and the estimated track produced by the models are presented in Figure 3c and 3g. In particular, we can see that the estimated track matches the actual acoustic source in Figure 3c, which moves in a circular path trajectory, and Figure 3d, which moves in an infinity-shaped path trajectory. The errors in Figure 3d further support this, where the error is below 5×10^{-5} in both cases. It is important to note that the errors $\epsilon_X = x(t) - p_x(t)$ and $\epsilon_Y = y(t) - p_y(t)$, while the estimated tracks provided by the ZNNQ-G and ZNNQ-D models, respectively, are denoted by the superscripts G and D in these errors. In other words, the ZNNQ-G and ZNNQ-D models were successful in locating the acoustic source.

6. Conclusions

Two models, ZNNQ-G and ZNNQ-D, have been presented in order to address the ML-TQ-LME problem for TQ input matrices of any dimension. The creation of such models has been backed by theoretical research and an examination of their computational complexity, in addition to simulation examples and practical acoustic source tracking applications. The gradient design, utilized by the ZNNQ-G model, has been suggested as being more efficient than the direct design, represented by the ZNNQ-D model, since the ML-TQ-LME problem has been successfully addressed.

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

- The use of nonlinear ZNNs in TQ issues may be investigated.
- It is possible to examine the application of the finite-time ZNN theme to TQ problems.
- Another area of research is using carefully selected design parameters stated in fuzzy settings to quicken ZNN model convergence.

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. G. X. Huang, F. Yin, K. Guo, An iterative method for the skew-symmetric solution and the optimal approximate solution of the matrix equation AXB=C, *J. Comput. Appl. Math*, **212** (2008), 231–244. https://doi:10.1016/j.cam.2006.12.005
2. 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.
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. Z. Zhang, Z. Yan, An adaptive fuzzy recurrent neural network for solving non-repetitive motion problem of redundant robot manipulators, *IEEE Trans. Fuzzy Syst.*, **28** (2020), 684–691. https://doi.org/10.1109/TFUZZ.2019.2914618
5. T. Sarkar, K. Siarkiewicz, R. Stratton, Survey of numerical methods for solution of large systems of linear equations for electromagnetic field problems, *IEEE Trans. Antennas Propag.*, **29** (1981), 847–856. https://doi.org/10.1109/TAP.1981.1142695
6. L. Xiao, J. Tao, J. Dai, Y. Wang, L. Jia, 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 T. Ind. Inform.*, **17** (2021), 6634–6643. https://doi.org/10.1109/TII.2021.3049413
7. 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.
8. Y. Zhang, S. S. Ge, Design and analysis of a general recurrent neural network model for time-varying matrix inversion, *IEEE T. Neur. Network.*, **16** (2005), 1477–1490. https://doi.org/10.1109/TNN.2005.857946
9. 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. https://doi.org/10.2991/ijcis.d.200527.001
10. Z. Sun, F. Li, L. Jin, T. Shi, K. Liu, Noise-tolerant neural algorithm for online solving time-varying full-rank matrix Moore-Penrose inverse problems: A control-theoretic approach, *Neurocomputing*, **413** (2020), 158–172. https://doi.org/10.1016/j.neucom.2020.06.050
11. 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
12. 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. https://doi.org/10.1007/s00607-010-0133-9
13. 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.*, **542** (2018), 101–117. https://doi.org/10.1016/j.laa.2017.03.014
14. 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. https://doi.org/10.1080/01630563.2020.1740887

15. 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
16. T. E. Simos, V. N. Katsikis, S. D. Mourtas, P. S. Stanimirović, D. Gerontitis, A higher-order zeroing neural network for pseudoinversion of an arbitrary time-varying matrix with applications to mobile object localization, *Inf. Sci.*, **600** (2022), 226–238. https://doi.org/10.1016/j.ins.2022.03.094
17. M. Zhou, J. Chen, P. S. Stanimirovic, V. N. Katsikis, H. Ma, Complex varying-parameter Zhang neural networks for computing core and core-EP inverse, *Neural Process. Lett.*, **51** (2020), 1299–1329. https://doi.org/10.1007/s11063-019-10141-6
18. J. Liu, H. Cai, C. Jiang, X. Han, Z. Zhang, An interval inverse method based on high dimensional model representation and affine arithmetic, *Appl. Math. Model.*, **63** (2018), 732–743. https://doi.org/10.1016/j.apm.2018.07.009
19. 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
20. 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.
21. 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
22. 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. https://doi.org/10.3390/math10111950
23. 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
24. 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. https://doi.org/10.1109/TFUZZ.2021.3115969
25. Y. Zhang, S. Li, J. Weng, B. Liao, GNN model for time-varying matrix inversion with robust finite-time convergence, *IEEE T. Neur. Net. Lear.*, (2022), 1–11. https://doi.org/10.1109/TNNLS.2022.3175899
26. Y. Zhang, Improved GNN method with finite-time convergence for time-varying Lyapunov equation, *Inf. Sci.*, **611** (2022), 494–503. https://doi.org/10.1016/j.ins.2022.08.061
27. W. R. Hamilton, On a new species of imaginary quantities, connected with the theory of quaternions, *P. Royal Irish Acad.*, **2** (1840), 424–434. https://www.jstor.org/stable/20520177
28. 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.
29. 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
30. G. Du, Y. Liang, B. Gao, S. A. Otaibi, D. Li, A cognitive joint angle compensation system based on self-feedback fuzzy neural network with incremental learning, *IEEE T. Ind. Inform.*, **17** (2021), 2928–2937. https://doi.org/10.1109/TII.2020.3003940

31. A. Szynal-Liana, I. Włoch, Generalized commutative quaternions of the Fibonacci type, *Boletín de la Sociedad Matemática Mexicana*, **28** (2022), 1. https://doi.org/10.1007/s40590-021-00386-4
32. 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
33. 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.
34. M. E. Kansu, Quaternionic representation of electromagnetism for material media, *Int. J. Geom. Methods M.*, **16** (2019), 1950105. https://doi.org/10.1142/S0219887819501056
35. S. Giardino, Quaternionic quantum mechanics in real Hilbert space, *J. Geom. Phys.*, **158** (2020), 103956. https://doi.org/10.1142/S0219887819501056
36. Z. H. Weng, Field equations in the complex quaternion spaces, *Adv. Math. Phys.*, **2014**.
37. R. Ghiloni, V. Moretti, A. Perotti, Continuous slice functional calculus in quaternionic Hilbert spaces, *Rev. Math. Phys.*, **25** (2013), 1350006. https://doi.org/10.1142/S0219887819501056
38. L. Xiao, S. Liu, X. Wang, Y. He, L. Jia, Y. Xu, Zeroing neural networks for dynamic quaternion-valued matrix inversion, *IEEE Trans. Ind. Inform.*, **18** (2022), 1562–1571.
39. 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. Network. Lear.*, (2022), 1–11.
40. 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. Network. Lear.*, (2023), 1–10. https://doi.org/10.1109/TNNLS.2023.3242313
41. 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. Network. Lear.*, (2022), 1–10.
42. 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. https://doi.org/10.3390/math11122756
43. 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. Instru. Meas.*, **71** (2022), 1–14. https://doi.org/10.1109/TIM.2022.3161713
44. 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
45. P. S. Stanimirović, S. D. Mourtas, V. N. Katsikis, L. A. Kazakovtsev, V. N. Krutikov, Recurrent neural network models based on optimization methods, *Mathematics*, **10** (2022), 4292. https://doi.org/10.3390/math10224292
46. F. Zhang, Quaternions and matrices of quaternions, *Linear Algebra Appl.*, **251** (1997), 21–57. https://doi.org/10.1016/0024-3795(95)00543-9

47. J. Groß, G. Trenkler, S. O. Troschke, Quaternions: Further contributions to a matrix oriented approach, *Linear Algebra Appl.*, **326** (2001), 205–213. https://doi.org/10.1016/S0024-3795(00)00283-4
48. 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, *Knowl. Based Syst.*, **242** (2022), 108405. https://doi.org/10.1016/j.knosys.2022.108405
49. L. Xiao, H. Tan, J. Dai, L. Jia, W. Tang, High-order error function designs to compute time-varying linear matrix equations, *Inform. Sciences*, **576** (2021), 173–186. https://doi.org/10.1016/j.ins.2021.06.038
50. N. Zhong, Q. Huang, S. Yang, F. Ouyang, Z. Zhang, A varying-parameter recurrent neural network combined with penalty function for solving constrained multi-criteria optimization scheme for redundant robot manipulators, *IEEE Access*, **9** (2021), 50810–50818. https://doi.org/10.1109/ACCESS.2021.3068731
51. A. K. Gupta, *Numerical methods using MATLAB*, MATLAB solutions series, Apress: Berkeley, CA, USA, New York, NY, 2014.
52. L. Jin, J. Yan, X. Du, X. Xiao, D. Fu, RNN for solving time-variant generalized Sylvester equation with applications to robots and acoustic source localization, *IEEE Trans. Ind. Inform.*, **16** (2020), 6359–6369. https://doi.org/10.1109/TII.2020.2964817
53. K. Kim, S. Wang, H. Ryu, S. Q. Lee, Acoustic-based position estimation of an object and a person using active localization and sound field analysis, *Appl. Sci.*, **10** (2020), 9090. https://doi.org/10.3390/app10249090
54. P. Du, S. Zhang, C. Chen, A. Alphones, W. D. Zhong, Demonstration of a low-complexity indoor visible light positioning system using an enhanced TDOA scheme, *IEEE Photon. J.*, **10** (2018), 1–10. https://doi.org/10.1109/JPHOT.2018.2840681
55. A. G. Dempster, E. Cetin, Interference localization for satellite navigation systems, *Proc. IEEE*, **104** (2016), 1318–1326. https://doi.org/10.1109/JPROC.2016.2530814
56. J. Tiemann, F. Eckermann, C. Wietfeld, ATLAS - an open-source TDOA-based ultra-wideband localization system, In: *Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)* (ed. A. de Henares), Spain, 2016.
57. Y. Zhang, L. Jin, *Robot Manipulator Redundancy Resolution*, John Wiley Sons: Hoboken, NJ, USA, 2017.

AIMS Press