AIMS Mathematics, 9(3): 5794–5809.
DOI: 10.3934/math.2024281
Received: 09 December 2023
Revised: 18 January 2024
Accepted: 24 January 2024
Published: 31 January 2024

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

Research article

Solving quaternion nonsymmetric algebraic Riccati equations through zeroing neural networks

Houssem Jerbi[1], Izzat Al-Darraji[2], Saleh Albadran[3], Sondess Ben Aoun[4], Theodore E. Simos[5,6,7,8,9,*], Spyridon D. Mourtas[10,11] and Vasilios N. Katsikis[10]

[1] Department of Industrial Engineering, College of Engineering, University of Hail, Ha'il 81481, Saudi Arabia

[2] Al-Khwarizmi College of Engineering, University of Baghdad, Baghdad 10081, Iraq

[3] Department of Electrical Engineering, College of Engineering, University of Hail, Ha'il 81481, Saudi Arabia

[4] Department of Computer Engineering, College of Computer Science and Engineering, University of Hail, Ha'il 81451, Saudi Arabia

[5] Center for Applied Math. and Bioinformatics, Gulf Univ. for Science and Technology, West Mishref 32093, Kuwait

[6] Department of Medical Research, China Medical Univ. Hospital, China Medical Univ., Taichung City 40402, Taiwan

[7] Laboratory of Inter-Disciplinary Problems in Energy Production, Ulyanovsk State Technical Univ., 32 Severny Venetz Street, 432027 Ulyanovsk, Russia

[8] Data Recovery Key Laboratory of Sichun Province, Neijing Normal Univ., Neijiang 641100, China

[9] Section of Mathematics, Dept. of Civil Engineering, Democritus Univ. of Thrace, Xanthi 67100, Greece

[10] Department of Economics, Division of Mathematics-Informatics and Statistics-Econometrics, National and Kapodistrian Univ. of Athens, Sofokleous 1 Street, 10559 Athens, Greece

[11] 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: Many variations of the algebraic Riccati equation (ARE) have been used to study nonlinear system stability in the control domain in great detail. Taking the quaternion nonsymmetric ARE (QNARE) as a generalized version of ARE, the time-varying QNARE (TQNARE) is introduced. This brings us to the main objective of this work: finding the TQNARE solution. The zeroing neural network (ZNN) technique, which has demonstrated a high degree of effectiveness in handling time-varying problems, is used to do this. Specifically, the TQNARE can be solved using the high order

ZNN (HZNN) design, which is a member of the family of ZNN models that correlate to hyperpower iterative techniques. As a result, a novel HZNN model, called HZ-QNARE, is presented for solving the TQNARE. The model functions fairly well, as demonstrated by two simulation tests. Additionally, the results demonstrated that, while both approaches function remarkably well, the HZNN architecture works better than the ZNN architecture.

Keywords: zeroing neural networks; nonsymmetric algebraic Riccati equations; quaternions; dynamical system
Mathematics Subject Classification: 65F20, 68T05

1. Introduction

Since Kalman demonstrated how common algebraic Riccati equations (ARE) are in the theory of filtering and optimal control [1], AREs have drawn a great deal of interest in applied mathematics and a variety of engineering tasks. Among these topics are wheeled inverted pendulums [2], multi-agent linear systems [3], and managing wind generators [4]. The linear-quadratic regulator [5], Kalman filtering [6], linear-quadratic-Gaussian and H_2/H$^\infty$ control [7,8], coprime and spectral factorizations [9, 10] all depend on ARE. Furthermore, several issues like losing controllability and numerical precision in direct and iterative methods are resolved through closed-form solutions of ARE [11]. In this paper, taking the quaternion nonsymmetric ARE (QNARE) as a generalized version of ARE, the time-varying QNARE (TQNARE) is introduced and solved.

Recently, there has been a greater focus in the study of dynamic topics using time-varying quaternion matrices (TQM). These include the constrained TQM least-squares problem [12], the inversion of TQM [13], and the pseudoinversion of TQM [14]. Moreover, practical applications of TQMs contain synchronization of chaotic systems [15], kinematic control of redundant manipulators [16], mobile manipulator control [17], and tracking acoustic sources [18]. All of these research topics have one thing in common: they use the zeroing neural network (ZNN) approach to determine the appropriate response. Remarkably, Zhang et al. presented the ZNN approach in [19] for real-time handling of time-varying problems. Particularly, ZNNs are recurrent neural networks with remarkable parallel processing performance. Among its applications are dynamical systems for computing the time-varying Moore-Penrose inverse [20, 21]. Linear/nonlinear equations systems [22–26], quadratic/linear programming [27–29], and generalized inversion [30, 31] are among the issues that they are presently employed for.

In 1843, Hamilton [32] made the first proposal regarding quaternions. They are very important in many fields, including mathematical physics [33], computer modeling [34], quantum mechanics [35], navigation [36], electromagnetism [37], and robotics [38, 39], because they are useful in computations associated with 3-dimensional rotations in applied and theoretical mathematics [40]. Keep in mind that quaternions are an extension of complex numbers into a non-commutative number system. Let

$$\mathbb{H} = \{\beta_1 + \beta_2\imath + \beta_3\jmath + \beta_4 k \mid \imath^2 = \jmath^2 = k^2 = \imath\jmath k = -1, \beta_1, \beta_2, \beta_3, \beta_4 \in \mathbb{R}\}$$

be the quaternions set and $\mathbb{H}^{m\times n}$ be the set of $m \times n$ matrices on $\mathbb{H}$ [41]. In this work, a recurrent neural

network, named high-order ZNN (HZNN), is used to solve the next TQNARE:

$$\tilde{D}(t)\tilde{X}(t) + \tilde{X}(t)\tilde{A}(t) - \tilde{X}(t)\tilde{B}(t)\tilde{X}(t) + \tilde{C}(t) = \mathbf{0}_{m\times n}, \tag{1.1}$$

where $\tilde{D}(t) \in \mathbb{H}^{m\times m}, \tilde{A}(t) \in \mathbb{H}^{n\times n}, \tilde{B}(t) \in \mathbb{H}^{n\times m}$, $\tilde{C}(t) \in \mathbb{H}^{m\times n}$ are the block coefficients, $\tilde{X}(t) \in \mathbb{H}^{m\times n}$ is the unknown matrix to be obtained, and $\mathbf{0}_{m\times n}$ represents a zero $m \times n$ matrix. It is significant to mention that (1.1) is in its standard form without assumption on the symmetry of the coefficient matrices, as indicated by the incorrect use of the word "nonsymmetric". The all ones and all zeros $u\times 1$ matrices, respectively, will be denoted by $\mathbf{1}_u$ and $\mathbf{0}_u$, while the identity $u \times u$ matrix will be denoted by I_u throughout this work. Additionally, $\otimes$ and $\odot$ will denote the Kronecker and Hadamard products, respectively, and vec$(\cdot)$ will denote the vectorization procedure. Finally, the matrix transposition, inversion, and pseudoinversion, respectively, will be denoted by the superscripts $()^{\mathrm{T}}$, $()^{-1}$, and $()^{\dagger}$, while $\|\cdot\|_{\mathrm{F}}$ will denote the matrix Frobenius norm.

From one perspective, a ZNN model is usually built in two main processes. First, the error matrix equation (EME), $E(t)$, needs to be declared. Second, the ZNN architecture that follows should be applied:

$$\dot{E}(t) = -\lambda E(t). \tag{1.2}$$

Notice that $t \in [0, t_f) \subseteq [0, +\infty)$ denotes the time, and $(\dot{\ })$ is the derivative operator in terms of time. In addition, the model's convergence speed can be adjusted by changing the parameter $\lambda \in \mathbb{R}^+$. For instance, any ZNN model converges even faster at larger values of λ [42]. The ZNN design is based on setting $E(t)$ to 0, where $t \to \infty$. This is achieved by utilizing the real-time learning rule that emerges from the building of the EME in (1.2). Thus, the EME can be thought of as a tracking tool for ZNN model learning.

On the other hand, the family of hyperpower iterations has been the subject of extensive research and development in the last few years [43]. Nevertheless, many HZNN models were introduced and investigated in [23] because iterative approaches can be applied to discrete-time models and these methods usually require initial conditions that are approximated and sometimes can not be easily provided. An HZNN model is typically constructed through a pair of primary steps. First, we need to define the function of the high-order error matrix equation (HEME), $E_H^p(t)$, as follows:

$$E_H^p(t) = \sum_{i=1}^{p-1} E^i(t), \tag{1.3}$$

where $E^i(t) \in \mathbb{R}^{n\times n}$. Second, the HZNN architecture:

$$\dot{E}(t) \approx -\lambda E_H^p(t) \tag{1.4}$$

should be applied for finding the online solution to a time-varying issue. In addition, the model's convergence rate can be adjusted by changing the $\lambda \in \mathbb{R}^+$ parameter and the order $p \geq 2$. For example, any HZNN model converges even faster at larger values of p [23]. It is significant to note that every HZNN model reduces to a ZNN model in the case where $p = 2$ since the ZNN dynamical system of (1.2) and the HZNN dynamical system of (1.4) will match.

In this work, a new HZNN model, termed HZ-QNARE, is introduced for solving the TQNARE. Two simulation trials demonstrate the excellent performance of the model. It is important to mention

that the HZ-QNARE model has the advantage of being able to solve any equation that arises from TQNARE. Additionally, the results demonstrate that, while both approaches function remarkably well, the HZNN architecture works better than the ZNN architecture. Notice that, by theoretical examination of the HZ-QNARE model, this work adds to the corpus of literature. Finally, the list below displays the contributions made to the paper:

- A new HZNN model, named HZ-QNARE, for solving the TQNARE is introduced.
- The HZ-QNARE model is validated by a theoretical examination.
- Simulation trials are carried out to supplement the theoretical study.

The paper is formatted as follows: Section 2 presents the preliminary results of the quaternion and the TQNARE reformulation. The HZ-QNARE model is presented in Section 3 and its theoretical examination is provided in Section 4. Observe that Section 3 contains information on the computational complexity of the HZ-QNARE model. Section 5 displays simulation trials. Finally, some remarks and closing comments are included in Section 6.

2. The TQNARE reformulation

In this section, the TQNARE is reformulated, and the fundamentals of TQM are described. Observe that the TQNARE (1.1) is being reformulated to reduce the computational complexity of the HZNN method.

Employing the coefficients $D_u(t) \in \mathbb{R}^{m\times m}$ for $u = 1, \ldots, 4$, let

$$\tilde{D}(t) = D_1(t) + D_2(t)\imath + D_3(t)\jmath + D_4(t)k \in \mathbb{H}^{m\times m}$$

to be a TQM. Similarly, employing the coefficients $A_u(t) \in \mathbb{R}^{n\times n}$, $B_u(t) \in \mathbb{R}^{n\times m}$, and $C_u(t), X_u(t) \in \mathbb{R}^{m\times n}$ for $u = 1, \ldots, 4$, let $\tilde{A}(t) \in \mathbb{H}^{n\times n}$, $\tilde{B}(t) \in \mathbb{H}^{n\times m}$, and $\tilde{C}(t), \tilde{X}(t) \in \mathbb{H}^{m\times n}$. The result of multiplying $\tilde{B}(t)$ by $\tilde{X}(t)$ is:

$$\tilde{B}(t)\tilde{X}(t) = \widetilde{BX}(t) = BX_1(t) + BX_2(t)\imath + BX_3(t)\jmath + BX_4(t)k \in \mathbb{H}^{n\times n}, \tag{2.1}$$

where

$$\begin{aligned}
BX_1(t) &= -B_3(t)X_3(t) + B_1(t)X_1(t) - B_2(t)X_2(t) - B_4(t)X_4(t),\\
BX_2(t) &= B_1(t)X_2(t) + B_3(t)X_4(t) + B_2(t)X_1(t) - B_4(t)X_3(t),\\
BX_3(t) &= B_1(t)X_3(t) + B_3(t)X_1(t) - B_2(t)X_4(t) + B_4(t)X_2(t),\\
BX_4(t) &= B_4(t)X_1(t) + B_1(t)X_4(t) - B_3(t)X_2(t) + B_2(t)X_3(t).
\end{aligned} \tag{2.2}$$

Notice that $BX_u(t) \in \mathbb{R}^{n\times n}$ for $u = 1, \ldots, 4$. The products $\widetilde{DX}(t) = \tilde{D}(t)\tilde{X}(t)$, $\widetilde{XA}(t) = \tilde{X}(t)\tilde{A}(t)$ and $\widetilde{XBX}(t) = \tilde{X}(t)\widetilde{BX}(t)$ can be produced in a manner similar to that of (2.1).

Using the previously provided information, (1.1) can be rewritten as:

$$\widetilde{DX}(t) + \widetilde{XA}(t) - \widetilde{XBX}(t) + \tilde{C}(t) = \mathbf{0}_{m\times n}, \tag{2.3}$$

where

$$\begin{cases}
DX_1(t) + XA_1(t) - XBX_1(t) + C_1(t) = \mathbf{0}_{m\times n},\\
DX_2(t) + XA_2(t) - XBX_2(t) + C_2(t) = \mathbf{0}_{m\times n},\\
DX_3(t) + XA_3(t) - XBX_3(t) + C_3(t) = \mathbf{0}_{m\times n},\\
DX_4(t) + XA_4(t) - XBX_4(t) + C_4(t) = \mathbf{0}_{m\times n}.
\end{cases} \tag{2.4}$$

Then, setting

$$Q(t)=\begin{bmatrix} X_1(t) & -X_2(t) & -X_3(t) & -X_4(t) \\ X_2(t) & X_1(t) & -X_4(t) & X_3(t) \\ X_3(t) & X_4(t) & X_1(t) & -X_2(t) \\ X_4(t) & -X_3(t) & X_2(t) & X_1(t) \end{bmatrix}, \quad A(t)=\begin{bmatrix} A_1(t) \\ A_2(t) \\ A_3(t) \\ A_4(t) \end{bmatrix}, \quad C(t)=\begin{bmatrix} C_1(t) \\ C_2(t) \\ C_3(t) \\ C_4(t) \end{bmatrix},$$
$$M(t)=\begin{bmatrix} B_1(t) & -B_2(t) & -B_3(t) & -B_4(t) \\ B_2(t) & B_1(t) & -B_4(t) & B_3(t) \\ B_3(t) & B_4(t) & B_1(t) & -B_2(t) \\ B_4(t) & -B_3(t) & B_2(t) & B_1(t) \end{bmatrix}, \quad B(t)=\begin{bmatrix} B_1(t) \\ B_2(t) \\ B_3(t) \\ B_4(t) \end{bmatrix}, \quad X(t)=\begin{bmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \\ X_4(t) \end{bmatrix}, \tag{2.5}$$
$$K(t)=\begin{bmatrix} D_1(t) & -D_2(t) & -D_3(t) & -D_4(t) \\ D_2(t) & D_1(t) & -D_4(t) & D_3(t) \\ D_3(t) & D_4(t) & D_1(t) & -D_2(t) \\ D_4(t) & -D_3(t) & D_2(t) & D_1(t) \end{bmatrix}, \quad D(t)=\begin{bmatrix} D_1(t) \\ D_2(t) \\ D_3(t) \\ D_4(t) \end{bmatrix},$$

where $Q(t) \in \mathbb{R}^{4m\times 4n}, M(t) \in \mathbb{R}^{4n\times 4m}, K(t) \in \mathbb{R}^{4m\times 4m}, D(t) \in \mathbb{R}^{4m\times m}, A(t) \in \mathbb{R}^{4n\times n}, B(t) \in \mathbb{R}^{4n\times m}$, and $X(t), C(t) \in \mathbb{R}^{4m\times n}$, (2.3) might be reshuffled as below:

$$K(t)X(t) + Q(t)A(t) - Q(t)M(t)X(t) + C(t) = \mathbf{0}_{4m\times n}, \tag{2.6}$$

where the coefficients $X_u(t), u = 1, \ldots, 4$, are included both in $X(t)$ and $Q(t)$. Recall that the anticipated solution to the TQNARE (1.1) is $\tilde{X}(t)$. Furthermore, take note of the fact that (2.6) gives four TQMs, whereas (1.1) yields only one TQM.

3. HZNN adaptation for the TQNARE system

To address the TQNARE, we shall construct a HZNN model in this section named HZ-QNARE. Let $\tilde{D}(t) \in \mathbb{H}^{m\times m}, \tilde{A}(t) \in \mathbb{H}^{n\times n}, \tilde{B}(t) \in \mathbb{H}^{n\times m}$, and $\tilde{C}(t), \tilde{X}(t) \in \mathbb{H}^{m\times n}$ be differentiable TQMs. According to the analysis of Section 2, (2.6) is a reformulation of (1.1). Following (2.5), we formulate the matrices $Q(t) \in \mathbb{R}^{4m\times 4n}, M(t) \in \mathbb{R}^{4n\times 4m}, K(t) \in \mathbb{R}^{4m\times 4m}, D(t) \in \mathbb{R}^{4m\times m}, A(t) \in \mathbb{R}^{4n\times n}, B(t) \in \mathbb{R}^{4n\times m}$, and $C(t) \in \mathbb{R}^{4m\times n}$ to construct the next EME:

$$E(t) = Q(t)A(t) + K(t)X(t) - Q(t)M(t)X(t) + C(t), \tag{3.1}$$

where $Q(t) \in \mathbb{R}^{4m\times 4n}$ and $X(t) \in \mathbb{R}^{4m\times n}$ are unknown matrices. Because the HZNN design requires powers of $E(t)$, it is applicable only to square input matrices $M(t), A(t)$, and $C(t)$. In order to satisfy this requirement, we right multiply (3.1) with $\bar{I} = \left[I_n, \mathbf{0}_{n\times(4m-n)}\right] \in \mathbb{R}^{n\times 4m}$. Therefore, without loss of generality, we convert (3.1) as follows:

$$\begin{aligned} E_Z(t) &= (Q(t)A(t) + K(t)X(t) - Q(t)M(t)X(t) + C(t))\bar{I} \\ &= Q(t)A(t)\bar{I} + K(t)X(t)\bar{I} - Q(t)M(t)X(t)\bar{I} + C(t)\bar{I}. \end{aligned} \tag{3.2}$$

The following is (3.2)'s first derivative:

$$\begin{aligned} \dot{E}_Z(t) =& Q(t)\dot{A}(t)\bar{I} + \dot{Q}(t)A(t)\bar{I} + \dot{K}(t)X(t)\bar{I} + K(t)\dot{X}(t)\bar{I} - \dot{Q}(t)M(t)X(t)\bar{I} \\ &- Q(t)\dot{M}(t)X(t)\bar{I} - Q(t)M(t)\dot{X}(t)\bar{I} + \dot{C}(t)\bar{I}. \end{aligned} \tag{3.3}$$

Further, the following HEME is defined:

$$E_H^p(t) = \sum_{i=1}^{p-1} E_Z^i(t), \tag{3.4}$$

while its derivative is $\dot{E}_H^p(t) \approx \dot{E}_Z(t)$ according to the HZNN design.

Next, we treat the HZNN dynamical system with regard to $\dot{X}(t)$ and $\dot{Q}(t)$. This is accomplished by superseding the aforementioned $E_H^p(t)$ and $\dot{E}_H^p(t)$ in (1.4). The outcome is as follows:

$$\begin{aligned} &K(t)\dot{X}(t)\bar{I} + \dot{Q}(t)A(t)\bar{I} - \dot{Q}(t)M(t)X(t)\bar{I} - Q(t)M(t)\dot{X}(t)\bar{I} \\ &= -\lambda \sum_{i=1}^{p-1} E_Z^i(t) - \dot{K}(t)X(t)\bar{I} - Q(t)\dot{A}(t)\bar{I} + Q(t)\dot{M}(t)X(t)\bar{I} - \dot{C}(t)\bar{I}. \end{aligned} \tag{3.5}$$

Afterwards, we simplify the dynamics of (3.5) by using the Kronecker product and vectorization:

$$\begin{aligned} &(\bar{I}^{\mathrm{T}}\otimes K(t) - \bar{I}^{\mathrm{T}}\otimes(Q(t)M(t)))\mathrm{vec}(\dot{X}(t)) + ((A(t)\bar{I})^{\mathrm{T}}\otimes I_{4m} - (M(t)X(t)\bar{I})^{\mathrm{T}}\otimes I_{4m})\mathrm{vec}(\dot{Q}(t)) \\ &= \mathrm{vec}(-\lambda \sum_{i=1}^{p-1} E_Z^i(t) - \dot{K}(t)X(t)\bar{I} - Q(t)\dot{A}(t)\bar{I} + Q(t)\dot{M}(t)X(t)\bar{I} - \dot{C}(t)\bar{I}). \end{aligned} \tag{3.6}$$

It is important to note that the components of $\mathrm{vec}(\dot{Q}(t))$ and $\mathrm{vec}(\dot{X}(t))$ are the same, but they are located differently. Stated differently, (3.6) can be further reduced by stating $\mathrm{vec}(\dot{Q}(t))$ in relation to $\mathrm{vec}(\dot{X}(t))$. Because of this, $\mathrm{vec}(\dot{Q}(t))$ can be substituted in (3.6) to obtain the following equation:

$$\mathrm{vec}(\dot{Q}(t)) = R\mathrm{vec}(\dot{X}(t)). \tag{3.7}$$

Note that the algorithmic process outlined in Algorithm 1 can be used to compute the operational matrix $R \in \mathbb{R}^{16mn\times 4mn}$. Take into account that the pseudocode in Algorithm 1 adhere to standard MATLAB practices [44].

Algorithm 1 Algorithmic process for calculating the operational matrix.

Input: The matrix $X(t)$ dimensions, $m \times n$.

procedure OM_R(m, n)
  Put R =zeros($nm16, nm4$), $c = (1 : nm4)'$, Z =reshape($c, m4, n$)
  Put $V1 = Z(1 : m, :)$, $V2 = Z(1 + m : m2, :)$, $V3 = Z(1 + m2 : m3, :)$, $V4 = Z(1 + m3 : end, :)$
  Put $V = [V1, -V2, -V3, -V4; V2, V1, -V4, V3; V3, V4, V1, -V2; V4, -V3, V2, V1]$
  Set $F = reshape(V, [], 1)$
  for $u = 1 : nm16$ **do**
    $R(u,\mathrm{abs}(F(u)))$ =sign($F(u)$)
  end for
end procedure

Output: R

The (3.6) may be simplified even more by utilizing (3.7):

$$\begin{aligned} &\left(\bar{I}^{\mathrm{T}}\otimes K(t) - \bar{I}^{\mathrm{T}}\otimes(Q(t)M(t)) + ((A(t)\bar{I})^{\mathrm{T}}\otimes I_{4m} - (M(t)X(t)\bar{I})^{\mathrm{T}}\otimes I_{4m})R\right)\mathrm{vec}(\dot{X}(t)) \\ &= \mathrm{vec}(-\lambda \sum_{i=1}^{p-1} E_Z^i(t) - \dot{K}(t)X(t)\bar{I} - Q(t)\dot{A}(t)\bar{I} + Q(t)\dot{M}(t)X(t)\bar{I} - \dot{C}(t)\bar{I}). \end{aligned} \tag{3.8}$$

Additionally, upon the establishment of:

$$\begin{aligned}
&U(t)=\bar{I}^{\mathrm{T}}\otimes K(t)-\bar{I}^{\mathrm{T}}\otimes(Q(t)M(t))+\big((A(t)\bar{I})^{\mathrm{T}}\otimes I_{4m}-(M(t)X(t)\bar{I})^{\mathrm{T}}\otimes I_{4m}\big)R\in\mathbb{R}^{16m^2\times 4mn},\\
&g(t)=\mathrm{vec}(-\lambda\sum_{i=1}^{p-1}E_Z^i(t)-\dot{K}(t)X(t)\bar{I}-Q(t)\dot{A}(t)\bar{I}+Q(t)\dot{M}(t)X(t)\bar{I}-\dot{C}(t)\bar{I})\in\mathbb{R}^{16m^2},\\
&\dot{\mathbf{x}}(t)=\mathrm{vec}(\dot{X}(t))\in\mathbb{R}^{4mn},\quad \mathbf{x}(t)=\mathrm{vec}(X(t))\in\mathbb{R}^{4mn},
\end{aligned}\tag{3.9}$$

we arrive at the HZNN model below:

$$\dot{\mathbf{x}}(t)=U^{\dagger}(t)g(t).\tag{3.10}$$

The recommended HZNN model to be used in solving the TQNARE of (1.1) is Eq (3.10)'s dynamic model, also known as HZ-QNARE. Note that the algorithmic process of the HZ-QNARE model outlined in Algorithm 2 can be used for solving the TQNARE.

Algorithm 2 Algorithmic process of the HZ-QNARE model for solving the TQNARE.

Input: The coefficient matrices $\tilde{D}(t)$, $\tilde{A}(t)$, $\tilde{B}(t)$, and $\tilde{C}(t)$ of the TQNARE (1.1).

1: **procedure** HZ_QNARE($\tilde{D}(t)$, $\tilde{A}(t)$, $\tilde{B}(t)$, $\tilde{C}(t)$)
2: Create $Q(t), A(t), C(t), M(t), B(t), X(t), K(t)$, and $D(t)$ based on (2.5)
3: Create $E_Z(t)$ according to (3.2)
4: Create $E_H^p(t)$ according to (3.4)
5: Using Alg. 1, create $U(t)$ and $g(t)$ based on (3.9)
6: Set $\dot{\mathbf{x}}(t) = U^{\dagger}(t)g(t)$ and use the `ode15s` solver of MATLAB to find $\mathbf{x}(t)$
7: **end procedure**

Output: The TQNARE's solution $\mathbf{x}(t)$.

Creating and solving (3.10) is computationally demanding in addition to the HZ-QNARE itself. In this regard, the complexity of formulating (3.10) is $O((4nm)^2)$ operations since, at each iteration, we do $(4nm)^2$ multiplications and $4nm$ additions/subtractions. Besides, the `ode15s` solver of MATLAB is used to solve the linear system of equations step-by-step. A $4nm \times 4nm$ matrix is needed to address (3.10), adding $O((4nm)^3)$ to the complexity. As a result, $O((4nm)^3)$ is the HZ-QNARE's computational complexity.

4. Theoretical analysis

This section presents the stability analysis and convergence of the HZ-QNARE (3.10) model.

Theorem 4.1. *Let $M(t) \in \mathbb{R}^{4n\times 4m}, K(t) \in \mathbb{R}^{4m\times 4m}, A(t) \in \mathbb{R}^{4n\times n}$, and $C(t) \in \mathbb{R}^{4m\times n}$ be differentiable. According to Lyapunov, the dynamics of* (3.5) *in accordance with the HZNN method* (1.4) *lead to the theoretical solution (TSOL) $X_{\mathcal{TH}}(t)$, which is stable.*

Proof. The supersession $X_{\mathcal{B}}(t) := -X(t) + X_{\mathcal{TH}}(t)$ yields $X(t) = X_{\mathcal{TH}}(t) - X_{\mathcal{B}}(t)$, while the TSOL is $X_{\mathcal{TH}}(t)$. The components of $X(t)$ are reshuffled in $Q(t)$, according to (2.5). Consequently, since $Q_{\mathcal{TH}}(t)$ and $Q_{\mathcal{B}}(t)$ represent a reshuffling of the components of $X_{\mathcal{TH}}(t)$ and $X_{\mathcal{B}}(t)$, $Q(t) = Q_{\mathcal{TH}}(t) - Q_{\mathcal{B}}(t)$. Also, the 1st derivatives of $Q(t)$ and $X(t)$ are $\dot{Q}(t) = \dot{Q_{\mathcal{TH}}}(t) - \dot{Q}_{\mathcal{B}}(t)$ and $\dot{X}(t) = \dot{X_{\mathcal{TH}}}(t) - \dot{X}_{\mathcal{B}}(t)$, respectively.

Let

$$K(t)X_{\mathcal{TH}}(t)\bar{I} + Q_{\mathcal{TH}}(t)A(t)\bar{I} - Q_{\mathcal{TH}}(t)M(t)X_{\mathcal{TH}}(t)\bar{I} + C(t)\bar{I} = \mathbf{0}_{4m\times 4m}, \tag{4.1}$$

and its 1st derivative

$$\begin{aligned} &K(t)\dot{X}_{\mathcal{TH}}(t)\bar{I}+\dot{K}(t)X_{\mathcal{TH}}(t)\bar{I}+\dot{Q}_{\mathcal{TH}}(t)A(t)\bar{I}+Q_{\mathcal{TH}}(t)\dot{A}(t)\bar{I}-\dot{Q}_{\mathcal{TH}}(t)M(t)X_{\mathcal{TH}}(t)\bar{I}\\ &-Q_{\mathcal{TH}}(t)\dot{M}(t)X_{\mathcal{TH}}(t)\bar{I}-Q_{\mathcal{TH}}(t)M(t)\dot{X}_{\mathcal{TH}}(t)\bar{I}+\dot{C}(t)\bar{I}=\mathbf{0}_{4m\times 4m}. \end{aligned} \tag{4.2}$$

After substituting (3.4) for $X(t) = X_{\mathcal{TH}}(t) - X_{\mathcal{B}}(t)$ and $Q(t) = Q_{\mathcal{TH}}(t) - Q_{\mathcal{B}}(t)$, the following holds:

$$\begin{aligned} E_{\mathcal{TH}}(t)=&K(t)X_{\mathcal{TH}}(t)\bar{I}-K(t)X_{\mathcal{B}}(t)\bar{I}+Q_{\mathcal{TH}}(t)A(t)\bar{I}-Q_{\mathcal{B}}(t)A(t)\bar{I}-Q_{\mathcal{TH}}(t)M(t)X_{\mathcal{TH}}(t)\bar{I}\\ &+Q_{\mathcal{B}}(t)M(t)X_{\mathcal{TH}}(t)\bar{I}+Q_{\mathcal{TH}}(t)M(t)X_{\mathcal{B}}(t)\bar{I}-Q_{\mathcal{B}}(t)M(t)X_{\mathcal{B}}(t)\bar{I}+C(t)\bar{I}, \end{aligned} \tag{4.3}$$

and

$$E^p_{\mathcal{TH}}(t) = \sum_{i=1}^{p-1} E^i_{\mathcal{TH}}(t), \tag{4.4}$$

with the following results from (1.4):

$$\begin{aligned} \dot{E}_{\mathcal{TH}}(t)=&K(t)\dot{X}_{\mathcal{TH}}(t)\bar{I}+\dot{K}(t)X_{\mathcal{TH}}(t)\bar{I}-K(t)\dot{X}_{\mathcal{B}}(t)\bar{I}-\dot{K}(t)X_{\mathcal{B}}(t)\bar{I}+\dot{Q}_{\mathcal{TH}}(t)A(t)\bar{I}+Q_{\mathcal{TH}}(t)\dot{A}(t)\bar{I}\\ &-\dot{Q}_{\mathcal{B}}(t)A(t)\bar{I}-Q_{\mathcal{B}}(t)\dot{A}(t)\bar{I}+\dot{C}(t)\bar{I}-\dot{Q}_{\mathcal{TH}}(t)M(t)X_{\mathcal{TH}}(t)\bar{I}-Q_{\mathcal{TH}}(t)\dot{M}(t)X_{\mathcal{TH}}(t)\bar{I}\\ &-Q_{\mathcal{TH}}(t)M(t)\dot{X}_{\mathcal{TH}}(t)\bar{I}+\dot{Q}_{\mathcal{B}}(t)M(t)X_{\mathcal{TH}}(t)\bar{I}+Q_{\mathcal{B}}(t)\dot{M}(t)X_{\mathcal{TH}}(t)\bar{I}+Q_{\mathcal{B}}(t)M(t)\dot{X}_{\mathcal{TH}}(t)\bar{I}\\ &+\dot{Q}_{\mathcal{TH}}(t)M(t)X_{\mathcal{B}}(t)\bar{I}+Q_{\mathcal{TH}}(t)\dot{M}(t)X_{\mathcal{B}}(t)\bar{I}+Q_{\mathcal{TH}}(t)M(t)\dot{X}_{\mathcal{B}}(t)\bar{I}\\ &-\dot{Q}_{\mathcal{B}}(t)M(t)X_{\mathcal{B}}(t)\bar{I}-Q_{\mathcal{B}}(t)\dot{M}(t)X_{\mathcal{B}}(t)\bar{I}-Q_{\mathcal{B}}(t)M(t)\dot{X}_{\mathcal{B}}(t)\bar{I}\\ \approx&-\lambda E^p_{\mathcal{TH}}(t). \end{aligned} \tag{4.5}$$

Afterwards, we pick the Lyapunov function indicated below to verify convergence:

$$\mathcal{R}(t) = \frac{1}{2}\left\|E^p_{\mathcal{TH}}(t)\right\|^2_{\mathrm{F}} = \frac{1}{2}\mathrm{tr}\left(E^p_{\mathcal{TH}}(t)\left(E^p_{\mathcal{TH}}(t)\right)^{\mathrm{T}}\right). \tag{4.6}$$

The following can then be verified:

$$\dot{\mathcal{R}}(t) = \frac{2\mathrm{tr}\left(\left(E^p_{\mathcal{TH}}(t)\right)^{\mathrm{T}}\dot{E}^p_{\mathcal{TH}}(t)\right)}{2} = \mathrm{tr}\left(\left(E^p_{\mathcal{TH}}(t)\right)^{\mathrm{T}}\dot{E}^p_{\mathcal{TH}}(t)\right) = -\lambda\mathrm{tr}\left(\left(E^p_{\mathcal{TH}}(t)\right)^{\mathrm{T}}E^p_{\mathcal{TH}}(t)\right). \tag{4.7}$$

Consequently, the following holds true:

$$\begin{aligned} &\dot{\mathcal{R}}(t)\begin{cases} <0, & E^p_{\mathcal{TH}}(t)\neq 0\\ =0, & E^p_{\mathcal{TH}}(t)=0, \end{cases}\\ \Leftrightarrow&\dot{\mathcal{R}}(t)\begin{cases} <0, & K(t)X_{\mathcal{TH}}(t)\bar{I}-K(t)X_{\mathcal{B}}(t)\bar{I}+Q_{\mathcal{TH}}(t)A(t)\bar{I}-Q_{\mathcal{B}}(t)A(t)\bar{I}-Q_{\mathcal{TH}}(t)M(t)X_{\mathcal{TH}}(t)\bar{I}\\ & \quad +Q_{\mathcal{B}}(t)M(t)X_{\mathcal{TH}}(t)\bar{I}+Q_{\mathcal{TH}}(t)M(t)X_{\mathcal{B}}(t)\bar{I}-Q_{\mathcal{B}}(t)M(t)X_{\mathcal{B}}(t)\bar{I}+C(t)\bar{I}\neq 0\\ =0, & K(t)X_{\mathcal{TH}}(t)\bar{I}-K(t)X_{\mathcal{B}}(t)\bar{I}+Q_{\mathcal{TH}}(t)A(t)\bar{I}-Q_{\mathcal{B}}(t)A(t)\bar{I}-Q_{\mathcal{TH}}(t)M(t)X_{\mathcal{TH}}(t)\bar{I}\\ & \quad +Q_{\mathcal{B}}(t)M(t)X_{\mathcal{TH}}(t)\bar{I}+Q_{\mathcal{TH}}(t)M(t)X_{\mathcal{B}}(t)\bar{I}-Q_{\mathcal{B}}(t)M(t)X_{\mathcal{B}}(t)\bar{I}+C(t)\bar{I}=0, \end{cases}\\ \Leftrightarrow&\dot{\mathcal{R}}(t)\begin{cases} <0, & X_{\mathcal{B}}(t)\neq 0\ \&\ Q_{\mathcal{B}}(t)\neq 0\\ =0, & X_{\mathcal{B}}(t)=0\ \&\ Q_{\mathcal{B}}(t)=0, \end{cases}\quad\Leftrightarrow\quad \dot{\mathcal{R}}(t)\begin{cases} <0, & X_{\mathcal{B}}(t)\neq 0\\ =0, & X_{\mathcal{B}}(t)=0. \end{cases} \end{aligned} \tag{4.8}$$

Seeing that $Q(t)$ is a components' regrouping of $X(t)$, $Q_{\mathcal{B}}(t) \neq 0$ in the case $X_{\mathcal{B}}(t) \neq 0$ and $Q_{\mathcal{B}}(t) = 0$ in the case $X_{\mathcal{B}}(t) = 0$. Furthermore, because $E_H^p(0) = 0$ and $X_{\mathcal{B}}(t)$ are the equilibrium points of (4.5), the following holds:

$$\forall\ X_{\mathcal{B}}(t) \neq 0, \quad \dot{\mathcal{R}}(t) \leq 0. \tag{4.9}$$

It is found that the state of equilibrium $X_{\mathcal{B}}(t) = -X(t) + X_{\mathcal{TH}}(t) = 0$ is stable according to the theory of Lyapunov. Afterwards, $X(t) \to X_{\mathcal{TH}}(t)$ as $t \to \infty$. Notice that, because $Q(t)$ is a components' regrouping of $X(t)$, $Q(t) \to Q_{\mathcal{TH}}(t)$ as $t \to \infty$. □

Theorem 4.2. *Let $\tilde{D}(t) \in \mathbb{H}^{m\times m}, \tilde{A}(t) \in \mathbb{H}^{n\times n}, \tilde{B}(t) \in \mathbb{H}^{n\times m}$, and $\tilde{C}(t) \in \mathbb{H}^{m\times n}$ be differentiable. For each $t \in [0, t_f) \subseteq [0, +\infty)$, the HZ-QNARE model* (3.10) *converges exponentially to TSOL $\mathbf{x}_{\mathcal{TH}}(t)$ under any selection of initial value* $\mathbf{x}(0)$.

Proof. First, the TQNARE of (1.1) is converted into (2.6) using the analysis given in Section 2. In particular, we utilize (2.5) to construct the matrices $M(t) \in \mathbb{R}^{4n\times 4m}, K(t) \in \mathbb{R}^{4m\times 4m}, A(t) \in \mathbb{R}^{4n\times n}$, and $C(t) \in \mathbb{R}^{4m\times n}$ employing the TQMs $\tilde{D}(t), \tilde{A}(t), \tilde{B}(t)$, and $\tilde{C}(t)$. Consequently, we substitute (1.1) for (2.6). Second, in order to solve (2.6), the HEME of (3.4) is declared. Model (3.5) is applied in compliance with the HZNN technique (1.4) for zeroing (3.4). In line with Theorem 4.1, $Q(t) \to Q_{\mathcal{TH}}(t)$ and $X(t) \to X_{\mathcal{TH}}(t)$ when $t \to \infty$ for any selection of initial value. Remember that $Q(t)$ is created by rearranging the $X(t)$ parts. Model (3.5) converges to the TSOL equation (1.1) since (2.6) is a reorganization of the TQNARE of (1.1). Third, (3.5) is simplified to construct the HZ-QNARE model of (3.10) using the operation matrix of (3.7), Kronecker product, and vectorization. The HZ-QNARE is an alternative formulation of (3.5); for any beginning value $\mathbf{x}(0)$, it converges to TSOL $\mathbf{x}_{\mathcal{TH}}(t)$ when $t \to \infty$. At that point, the proof is finished. □

5. Experiments

This section exhibits two simulation trials (STs). A few significant arguments should be mentioned. For simplicity, we allocated both $\eta(t) = \cos(t)$ and $\zeta(t) = \sin(t)$ in the STs. Additionally, all calculations are performed employing the `ode15s` solver of MATLAB, with a time interval set to $[0, 10]$. Employing the `ode15s` and its regular double precision arithmetic, you will notice that most errors have minimum values that are close to 10^{-5} in this section's figures. Last, the associated TQNARE error, i.e., $\|\tilde{D}(t)\tilde{X}(t) + \tilde{X}(t)\tilde{A}(t) - \tilde{X}(t)\tilde{B}(t)\tilde{X}(t) + \tilde{C}(t)\|_{\mathrm{F}}$, is displayed as $\|\mathrm{TQNARE}\|_{\mathrm{F}}$ in the figures of this section.

5.1. Walkthrough trials

Example 5.1. In this ST, the $\tilde{D}(t)$ coefficients are set up as below:

$$D_1(t) = I_4(1 + \eta(t)), \quad D_2(t) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}, \quad D_3(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad D_4(t) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix},$$

the $\tilde{A}(t)$ coefficients are set up as below:

$$A_1(t) = \begin{bmatrix} 6\zeta(t) + 1 & 4 \\ 5\eta(t) + 2 & 6 \end{bmatrix}, \quad A_2(t) = \begin{bmatrix} 0 & 2\zeta(t) + 2 \\ 4\zeta(t) - 2 & 0 \end{bmatrix}, \quad A_3(t) = \begin{bmatrix} 0 & 5 \\ 5\zeta(t) + 1 & 0 \end{bmatrix}, \quad A_4(t) = \begin{bmatrix} 0 & 2\zeta(t) + 3 \\ 2 & 0 \end{bmatrix},$$

the $\tilde{B}(t)$ coefficients are set up as below:

$$B_1(t) = \begin{bmatrix} \eta(t) & \zeta(t)+3 & 0 & 1 \\ \zeta(t)+3 & \zeta(t)+2 & 0 & 0 \end{bmatrix}, \quad B_2(t) = \begin{bmatrix} 0 & 6 & -1 & 0 \\ -6 & 0 & 1 & 1 \end{bmatrix},$$
$$B_3(t) = \begin{bmatrix} 0 & 5+\eta(t) & 0 & 0 \\ -5-\eta(t) & 0 & 0 & 0 \end{bmatrix}, \quad B_4(t) = \begin{bmatrix} 0 & \eta(t) & 4 & 1 \\ -\eta(t) & 0 & 0 & 0 \end{bmatrix},$$

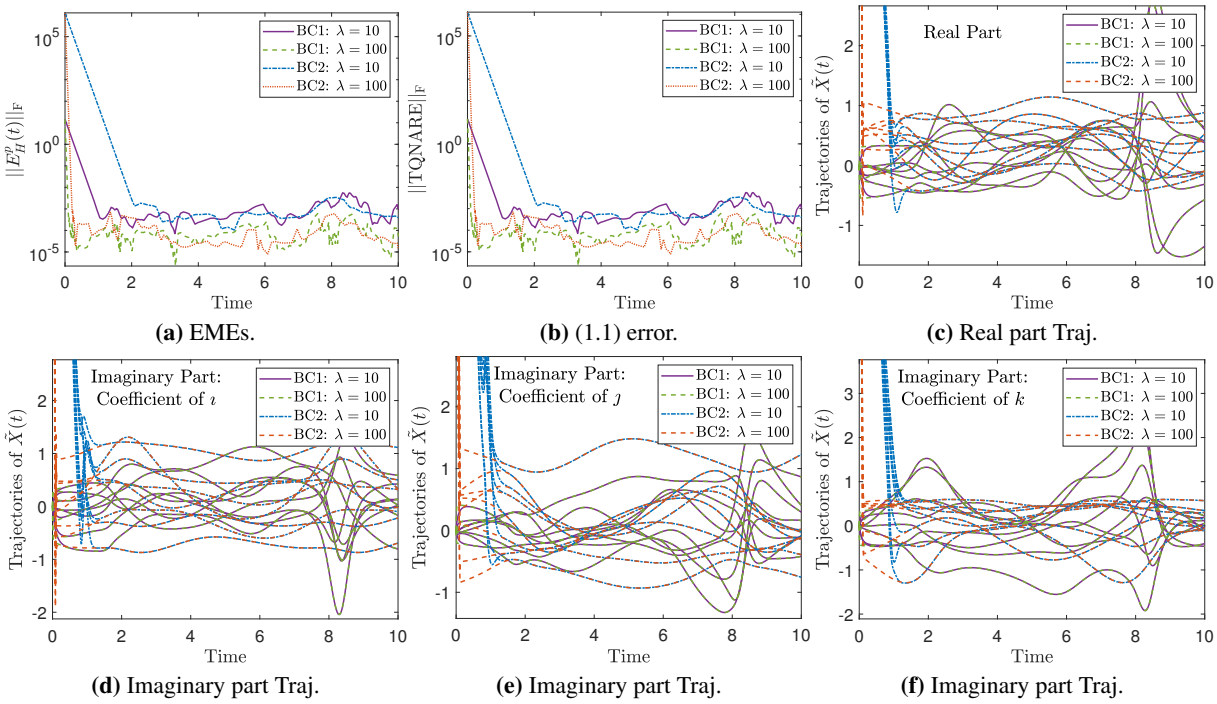
and $\tilde{C}(t)$ coefficients are set up as below:

$$C_1(t) = \begin{bmatrix} \zeta(t)+5 & \eta(t) \\ \eta(t) & \eta(t)+2 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad C_2(t) = \begin{bmatrix} 0 & 5+\zeta(t) \\ -5-\zeta(t) & 0 \\ 5 & 1 \\ 3 & 0 \end{bmatrix}, \quad C_3(t) = \begin{bmatrix} 0 & 6 \\ -6 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \quad C_4(t) = \begin{bmatrix} 0 & \zeta(t) \\ -\zeta(t) & 0 \\ 0 & 3 \\ -1 & 0 \end{bmatrix}.$$

Hence, $\tilde{D}(t) \in \mathbb{H}^{4\times4}$, $\tilde{A}(t) \in \mathbb{H}^{2\times2}$, $\tilde{B}(t) \in \mathbb{H}^{2\times4}$, and $\tilde{C}(t) \in \mathbb{H}^{4\times2}$. Furthermore, the HZ-QNARE model's λ parameter is used, with values of 10 and 100, and the beginning conditions (BCs) are established as below:

- BC1: $\mathbf{x}(0) = \mathbf{0}_{32}$,
- BC2: $\mathbf{x}(0) = 100 \odot \mathbf{1}_{32}$.

You will see that we employ two BCs to confirm Theorem 4.2's findings. The results of the HZ-QNARE model are shown in Figure 1.



(a) EMEs. **(b)** (1.1) error. **(c)** Real part Traj.

(d) Imaginary part Traj. **(e)** Imaginary part Traj. **(f)** Imaginary part Traj.

Figure 1. (1.1) error, solutions trajectories, and HEMEs of HZ-QNARE in ST 5.1.

Example 5.2. Assume the next 10×10 real matrix:

$$M = \begin{bmatrix} 3+\eta(t) & 1+\zeta(t)/2 & \dots & 1+\zeta(t)/10 \\ 1+\zeta(t)/2 & 3+\eta(t) & \dots & 1+\zeta(t)/9 \\ \vdots & \vdots & \ddots & \vdots \\ 1+\zeta(t)/10 & 1+\zeta(t)/9 & \dots & 3+\eta(t) \end{bmatrix}.$$

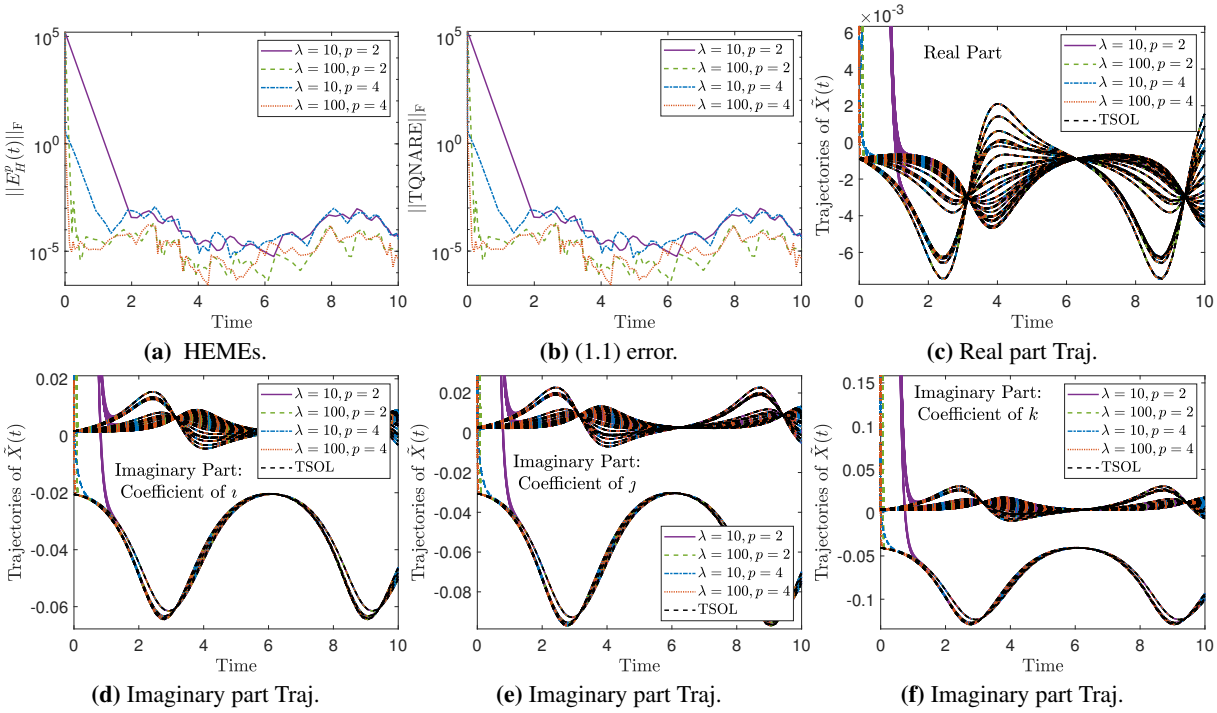
In this ST, the coefficients of $\tilde{A}(t)$ are set up as below:

$$A_1(t) = M + I_{10}\zeta(t), \quad A_2(t) = 2M + I_{10}\eta(t),$$
$$A_3(t) = 3M + I_{10}\frac{\zeta(t)}{2}, \quad A_4(t) = 4M + I_{10}\frac{\eta(t)}{2},$$

and matrices $\tilde{D}(t), \tilde{B}(t)$, and $\tilde{C}(t)$ are set up as below:

$$\tilde{D}(t) = \tilde{B}(t) = \mathbf{0}_{10\times 10}, \quad \tilde{C}(t) = -I_{10}.$$

Therefore, $\tilde{D}(t), \tilde{A}(t), \tilde{B}(t), \tilde{C}(t) \in \mathbb{H}^{10\times 10}$. In order to confirm the convergence characteristics of the HZNN method, the beginning state of the HZ-QNARE model is set to $\mathbf{x}(0) = 100 \odot \mathbf{1}_{400}$. Also, the λ parameter of the HZ-QNARE model is employed with values 10 and 100, and the order p is employed with prices 2 and 4. The results of the HZ-QNARE model are shown in Figure 2.



(a) HEMEs. **(b)** (1.1) error. **(c)** Real part Traj.

(d) Imaginary part Traj. **(e)** Imaginary part Traj. **(f)** Imaginary part Traj.

Figure 2. (1.1) error, solutions trajectories, and HEMEs of HZ-QNARE in ST 5.2.

5.2. Discussion of the ST findings

Throughout Examples 5.1 and 5.2, the effectiveness of the HZ-QNARE (3.10) model in solving the TQNARE of (1.1) is tested. Each ST has a unique TQNARE due to the suitable matrices $\tilde{D}(t), \tilde{B}(t), \tilde{C}(t)$, and $\tilde{A}(t)$.

It is noticeable that $\tilde{C}(t) \in \mathbb{H}^{4\times 2}$ in ST 5.1. In other words, $m = 4$ and $n = 2$ in (1.1)'s TQNARE. Under $\lambda = 10$, $\lambda = 100$, $p = 2$, and $p = 4$, we have gathered the following findings regarding the HZ-QNARE model: Figure 1a exhibits the HZ-QNARE model's HEMEs. The two BC's instances in this exhibition begin at $t = 0$ from a high error value. Notably, the beginning error value of BC2 is substantially higher than BC1's. Nevertheless, when $\lambda = 10$, the HEMEs attain a negligible error value within the $[10^{-4}, 10^{-2}]$ interval in the BC1 paradigm at $t = 1$ and in the BC2 paradigm at $t = 2$. Additionally, when $\lambda = 100$, the HEMEs attain a negligible error value within the $[10^{-6}, 10^{-3}]$ interval in the BC1 paradigm at $t = 0.1$ and in the BC2 paradigm at $t = 0.2$. Put another way, the HZ-QNARE model, which converges to a low value for two different BCs, validates Theorem 4.2. Figure 1b exhibits the HZ-QNARE model's (1.1) error. The outcomes shown there support those found in Figure 1a, showing that there is no difference in the solutions to (2.6) and (1.1). Trajectories representing the model's solutions are displayed in Figure 1c–1f. The three imaginary and real components of the solutions are shown in these figures. It is shown that the models offer distinct solutions for a range of BCs and that their convergence to TSOL is in line with the convergence trend of the associated HEMEs.

Further, it is noticeable that $\tilde{D}(t), \tilde{A}(t), \tilde{B}(t), \tilde{C}(t) \in \mathbb{H}^{10\times 10}$ in ST 5.2. That is, $m = n = 10$ in (1.1)'s TQNARE. It is crucial to notice that by setting $\tilde{B}(t) = \tilde{D}(t) = \mathbf{0}_{10\times 10}$ and $\tilde{C}(t) = -I_{10}$, the TQNARE is reduced into a TQM inversion problem. As a result, the TSOL of the TQNARE is the inverse of $\tilde{A}(t)$, i.e., $\tilde{A}^{-1}(t)$. Under BC1 and BC2 for $\lambda = 10$, we have gathered the next findings regarding the HZ-QNARE model. Figure 2a exhibits the HZ-QNARE model's HEMEs. The four instances in this exhibition begin at $t = 0$ from a high error value and end up with a negligible value. Notably, in the paradigm of $\lambda = 10$, they end up in the interval $[10^{-5}, 10^{-3}]$ with a negligible error value when $p = 2$ at $t = 2$, and at $t = 1.5$ when $p = 4$. In the paradigm of $\lambda = 100$, they end up in the interval $[10^{-7}, 10^{-4}]$ with a negligible error value when $p = 2$ at $t = 0.2$, and at $t = 0.15$ when $p = 4$. Put another way, the HEME of the HZ-QNARE model, which is affected by the values of λ and p, verifies the convergence qualities of the HZNN method. Figure 2b exhibits the HZ-QNARE model's (1.1) error. The outcomes shown there support those found in Figure 2a, showing that there is no difference in the solutions to (2.6) and (1.1). Trajectories representing the model's solutions are displayed in Figure 2c–2f. The three imaginary and real components of the solutions are shown in these figures. It is shown that the models offer distinct solutions for a range of BCs, and that their convergence to TSOL is in line with the convergence trend of the associated HEMEs.

After all is said and done, the HZ-QNARE model handles two different TQNAREs fairly well. Notably, the above explanation validates the convergence properties of the HZNN approach and the conclusions of Theorem 4.2. It should be noted that the ZNN model is the HZNN under $p = 2$ since the HZNN dynamical system of (1.4) and the ZNN dynamical system of (1.4) match for $p = 2$. Therefore, the HZNN architecture performs better than the ZNN architecture because, at $p = 2$, the HZ-QNARE model turns into a ZNN model, while at $p > 2$, the HZ-QNARE model achieves better convergence. Additionally, the HZ-QNARE computational complexity is $O((4nm)^3)$, which is comparable to other ZNN models that tackle different TQM equations (see [14, 15, 17, 18, 45]).

6. Conclusions

In this work, a new HZNN model, named HZ-QNARE, is introduced for solving the TQNARE. In order to validate the accuracy and effectiveness of the HZ-QNARE model, theoretical research and simulation experiments were carried out. These experiments' outcomes demonstrate that the HZNN design outperforms the ZNN architecture in terms of effectiveness.

It must be noted that the recommended HZNN model has the disadvantage of being noise-intolerant because noise of any kind has a major impact on the accuracy of the suggested HZNN approach. Thus, the primary goal of future research may be to convert this model to a noise-tolerant HZNN architecture. Other than that, the suggested HZNN model may be used for a variety of additional technology-related issues in the future, such as network and power systems [46, 47] and secure communications [48].

Use of AI tools declaration

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

Acknowledgments

This research was funded by Research Deanship of Hail University-KSA Project Number (RG-23 072).

Conflict of interest

The authors declare no conflicts of interest.

References

1. R. E. Kalman, Contributions to the theory of optimal control, *Bol. Soc. Mat. Mexicana*, **5** (1960), 102–119.
2. G. Rigatos, K. Busawon, J. Pomares, M. Abbaszadeh, Nonlinear optimal control for the wheeled inverted pendulum system, *Robotica*, **38** (2020), 29–47. https://doi.org/10.1017/S0263574719000456
3. X. Dong, G. Hu, Time-varying formation tracking for linear multiagent systems with multiple leaders, *IEEE Trans. Autom. Control*, **62** (2017), 3658–3664. https://doi.org/10.1109/TAC.2017.2673411
4. B. Qin, H. Sun, J. Ma, W. Li, T. Ding, Z. Wang, et al., Robust H_∞ control of doubly fed wind generator via state-dependent Riccati equation technique, *IEEE Trans. Power Syst.*, **34** (2019), 2390–2400. https://doi.org/10.1109/TPWRS.2018.2881687
5. L. T. Aguilar, Y. Orlov, L. Acho, Nonlinear H_∞-control of nonsmooth time-varying systems with application to friction mechanical manipulators, *Automatica*, **39** (2003), 1531–1542. https://doi.org/10.1016/S0005-1098(03)00148-1
6. A. J. Laub, A schur method for solving algebraic Riccati equations, *IEEE Trans. Autom. Control*, **24** (1979), 913–921. https://doi.org/10.1109/TAC.1979.1102178

7. A. Ferrante, L. Ntogramatzidis, The generalized continuous algebraic Riccati equation and impulse-free continuous-time LQ optimal control, *Automatica*, **50** (2014), 1176–1180. https://doi.org/10.1016/j.automatica.2014.02.014
8. T. Ohtsuka, A recursive elimination method for finite-horizon optimal control problems of discrete-time rational systems, *IEEE Trans. Autom. Control*, **59** (2014), 3081–3086. https://doi.org/10.1109/TAC.2014.2321231
9. Y. Oshman, I. Bar-Itzhack, Eigenfactor solution of the matrix Riccati equation–a continuous square root algorithm, *IEEE Trans. Autom. Control*, **30** (1985), 971–978. https://doi.org/10.1109/TAC.1985.1103823
10. P. Van Dooren, A generalized eigenvalue approach for solving Riccati equations, *SIAM J. Sci. Stat. Comput.*, **2** (1981), 121–135. https://doi.org/10.1137/0902010
11. R. C. Li, W. Kahan, A family of anadromic numerical methods for matrix Riccati differential equations, *Math. Comp.*, **81** (2012), 233–265. https://doi.org/10.1090/S0025-5718-2011-02498-1
12. 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 Trans. Neur. Net. Lear. Syst.*, 2023, 1–10. https://doi.org/10.1109/TNNLS.2023.3242313
13. L. Xiao, S. Liu, X. Wang, Y. He, L. Jia, Y. Xu, Zeroing neural networks for dynamic quaternion-valued matrix inversion, *IEEE Trans. Ind. Inf.*, **18** (2022), 1562–1571. https://doi.org/10.1109/TII.2021.3090063
14. 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
15. S. B. Aoun, N. Derbel, H. Jerbi, T. E. Simos, S. D. Mourtas, V. N. Katsikis, A quaternion Sylvester equation solver through noise-resilient zeroing neural networks with application to control the SFM chaotic system, *AIMS Math.*, **8** (2023), 27376–27395. https://doi.org/10.3934/math.20231401
16. N. Tan, P. Yu, F. Ni, New varying-parameter recursive neural networks for model-free kinematic control of redundant manipulators with limited measurements, *IEEE Trans. Instrum. Meas.*, **71** (2022), 1–14. https://doi.org/10.1109/TIM.2022.3161713
17. 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
18. V. N. Kovalnogov, R. V. Fedorov, I. I. Shepelev, V. V. Sherkunov, T. E. Simos, S. D. Mourtas, et al., A novel quaternion linear matrix equation solver through zeroing neural networks with applications to acoustic source tracking, *AIMS Math.*, **8** (2023), 25966–25989. https://doi.org/10.3934/math.20231323
19. Y. Zhang, S. S. Ge, Design and analysis of a general recurrent neural network model for time-varying matrix inversion, *IEEE Trans. Neur. Net.*, **16** (2005), 1477–1490. https://doi.org/10.1109/TNN.2005.857946
20. 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

21. 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
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, et al., 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 Trans. Fuzzy Syst.*, **30** (2022), 3426–3435. https://doi.org/10.1109/TFUZZ.2021.3115969
25. 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. https://doi.org/10.3390/math11030600
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 Trans. Neural Networks Learn. Syst.*, 2022, 1–10. https://doi.org/10.1109/TNNLS.2022.3225309
27. 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
28. 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. https://doi.org/10.3390/math10183335
29. 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
30. S. Qiao, Y. Wei, X. Zhang, Computing time-varying ML-weighted pseudoinverse by the Zhang neural networks, *Numer. Funct. Anal. Optim.*, **41** (2020), 1672–1693. https://doi.org/10.1080/01630563.2020.1740887
31. 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
32. W. R. Hamilton, On a new species of imaginary quantities, connected with the theory of quaternions, *Proceedings of the Royal Irish Academy (1836–1869)*, **2** (1840), 424–434.
33. A. Szynal-Liana, I. Włoch, Generalized commutative quaternions of the Fibonacci type, *Bol. Soc. Mat. Mex.*, **28** (2022), 1. https://doi.org/10.1007/s40590-021-00386-4
34. D. Pavllo, C. Feichtenhofer, M. Auli, D. Grangier, Modeling human motion with quaternion-based neural networks, *Int. J. Comput. Vis.*, **128** (2020), 855–872. https://doi.org/10.1007/s11263-019-01245-6
35. S. Giardino, Quaternionic quantum mechanics in real Hilbert space, *J. Geom. Phys.*, **158** (2020), 103956. https://doi.org/10.1016/j.geomphys.2020.103956
36. A. M. S. Goodyear, P. Singla, D. B. Spencer, Analytical state transition matrix for dual-quaternions for spacecraft pose estimation, *AAS/AIAA Astrodynamics Specialist Conference, 2019*, Univelt Inc., 2020, 393–411.

37. M. E. Kansu, Quaternionic representation of electromagnetism for material media, *Int. J. Geom. Meth. Mod. Phys.*, **16** (2019), 1950105. https://doi.org/10.1142/S0219887819501056
38. E. Özgür, Y. Mezouar, Kinematic modeling and control of a robot arm using unit dual quaternions, *Rob. Auton. Syst.*, **77** (2016), 66–73. https://doi.org/10.1016/j.robot.2015.12.005
39. S. Stepień, P. Superczyńska, Modified infinite-time state-dependent Riccati equation method for nonlinear affine systems: quadrotor control, *Appl. Sci.*, **11** (2021), 10714. https://doi.org/10.3390/app112210714
40. M. Joldeş, J. M. Muller, Algorithms for manipulating quaternions in floating-point arithmetic, *2020 IEEE 27th Symposium on Computer Arithmetic (ARITH)*, 2020, 48–55. https://doi.org/10.1109/ARITH48897.2020.00016
41. F. Zhang, Quaternions and matrices of quaternions, *Linear Algebra Appl.*, **251** (1997), 21–57. https://doi.org/10.1016/0024-3795(95)00543-9
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, *Knowl.-Based Syst.*, **242** (2022), 108405. https://doi.org/10.1016/j.knosys.2022.108405
43. P. S. Stanimirović, S. Srivastava, D. K. Gupta, From Zhang Neural Network to scaled hyperpower iterations, *J. Comput. Appl. Math.*, **331** (2018), 133–155. https://doi.org/10.1016/j.cam.2017.09.048
44. A. K. Gupta, *Numerical methods using MATLAB*, CA: Apress Berkeley, 2014. https://doi.org/10.1007/978-1-4842-0154-1
45. 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
46. 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
47. 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
48. 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 Trans. Neural Networks Learn. Syst.*, **31** (2020), 3334–3345. https://doi.org/10.1109/TNNLS.2019.2943548

AIMS Press