

Article

Hermitian Solutions of the Quaternion Algebraic Riccati Equations through Zeroing Neural Networks with Application to Quadrotor Control

Housseem Jerbi ¹, Obaid Alshammari ², Sondess Ben Aoun ³, Mourad Kchaou ², Theodore E. Simos ^{4,5,6,7,8,*}, Spyridon D. Mourtas ^{9,10} and Vasilios N. Katsikis ⁹

- ¹ Department of Industrial Engineering, College of Engineering, University of Hail, Hail 81481, Saudi Arabia; h.jerbi@uoh.edu.sa
 - ² Department of Electrical Engineering, College of Engineering, University of Hail, Hail 81481, Saudi Arabia; os.alshammari@uoh.edu.sa (O.A.); m.kchaou@uoh.edu.sa (M.K.)
 - ³ Department of Computer Engineering, College of Computer Science and Engineering, University of Hail, Hail 81451, Saudi Arabia; s.benaoun@uoh.edu.sa
 - ⁴ Laboratory of Interdisciplinary Problems of Energy Production, Ulyanovsk State Technical University, 32 Severny Venetz Street, 432027 Ulyanovsk, Russia
 - ⁵ Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 40402, Taiwan
 - ⁶ Center for Applied Mathematics and Bioinformatics, Gulf University for Science and Technology, West Mishref 32093, Kuwait
 - ⁷ Data Recovery Key Laboratory of Sichun Province, Neijing Normal University, Neijiang 641100, China
 - ⁸ Section of Mathematics, Department of Civil Engineering, Democritus University of Thrace, 67100 Xanthi, Greece
 - ⁹ Department of Economics, Mathematics-Informatics and Statistics-Econometrics, National and Kapodistrian University of Athens, 10559 Athens, Greece; spirmour@econ.uoa.gr (S.D.M.); vaskatsikis@econ.uoa.gr (V.N.K.)
 - ¹⁰ Laboratory "Hybrid Methods of Modelling and Optimization in Complex Systems", Siberian Federal University, 660041 Krasnoyarsk, Russia
- * Correspondence: simos@ulstu.ru



Citation: Jerbi, H.; Alshammari, O.; Aoun, S.B.; Kchaou, M.; Simos, T.E.; Mourtas, S.D.; Katsikis, V.N. Hermitian Solutions of the Quaternion Algebraic Riccati Equations through Zeroing Neural Networks with Application to Quadrotor Control. *Mathematics* **2024**, *12*, 15. <https://doi.org/10.3390/math12010015>

Academic Editor: Dimplekumar N. Chalishajar

Received: 25 November 2023

Revised: 10 December 2023

Accepted: 19 December 2023

Published: 20 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: The stability of nonlinear systems in the control domain has been extensively studied using different versions of the algebraic Riccati equation (ARE). This leads to the focus of this work: the search for the time-varying quaternion ARE (TQARE) Hermitian solution. The zeroing neural network (ZNN) method, which has shown significant success at solving time-varying problems, is used to do this. We present a novel ZNN model called 'ZQ-ARE' that effectively solves the TQARE by finding only Hermitian solutions. The model works quite effectively, as demonstrated by one application to quadrotor control and three simulation tests. Specifically, in three simulation tests, the ZQ-ARE model finds the TQARE Hermitian solution under various initial conditions, and we also demonstrate that the convergence rate of the solution can be adjusted. Furthermore, we show that adapting the ZQ-ARE solution to the state-dependent Riccati equation (SDRE) technique stabilizes a quadrotor's flight control system faster than the traditional differential-algebraic Riccati equation solution.

Keywords: zeroing neural network; quaternion; algebraic Riccati equation; quadrotor control

MSC: 65F20; 68T05

1. Introduction

Algebraic Riccati equations (AREs) have gained considerable magnitude in applied mathematics and a range of engineering issues since Kalman showed how widely used they are in filtering and optimal control theory [1]. These problems include controlling

wind generators [2], linear multi-agent systems [3], and wheeled inverted pendulums [4]. In particular, the continuous-time ARE is the following quadratic matrix equation [5]:

$$A^T X + XA - XBX + C = \mathbf{0}_{n \times n}. \quad (1)$$

Linear-quadratic regulators [6–8], Kalman filtering [5,9], linear-quadratic-Gaussian along with H_2/H^∞ control [10,11], and coprime and spectral factorizations [12,13] all depend on this matrix equation. It is significant to mention that $()^T$ signifies transposition; the zero $p \times 1$ and $p \times n$ matrices are referred to as $\mathbf{0}_p$ and $\mathbf{0}_{p \times n}$, respectively; and each matrix in (1) belongs to $\mathbb{R}^{n \times n}$, with B and C being symmetric and nonnegative definite matrices (i.e., $B = B^T \geq 0$ and $C = C^T \geq 0$). Also, there can be an infinite or a finite number of symmetric or antisymmetric solutions X with definite or indefinite signs in the solutions set for (1).

Let $\mathbb{H} = \{\beta_1 + \beta_2 i + \beta_3 j + \beta_4 k \mid i^2 = j^2 = k^2 = ijk = -1, \beta_1, \beta_2, \beta_3, \beta_4 \in \mathbb{R}\}$ be the quaternions set and $\mathbb{H}^{n \times n}$ be the $n \times n$ matrices set on \mathbb{H} [14]. Note that quaternions are a system of non-commutative numbers that builds upon complex numbers. In this paper, a recurrent neural network, called a 'zeroing neural network (ZNN)', is used to solve the next time-varying quaternion ARE (TQARE):

$$\tilde{A}^*(t)\tilde{X}(t) + \tilde{X}(t)\tilde{A}(t) - \tilde{X}(t)\tilde{B}(t)\tilde{X}(t) + \tilde{C}(t) = \mathbf{0}_{n \times n}, \quad (2)$$

where all time-varying matrices are in $\mathbb{H}^{n \times n}$, $()^*$ denotes the conjugate transpose, and $\tilde{B}(t)$ and $\tilde{C}(t)$ are assumed to be positive semidefinite (i.e., $\tilde{B}(t) = \tilde{B}^*(t) \geq 0$ and $\tilde{C}(t) = \tilde{C}^*(t) \geq 0$). According to [15], there can be an infinite or a finite number of Hermitian or non-Hermitian solutions $\tilde{X}(t)$ with definite or indefinite signs in the solutions set for (2). Notably, Hamilton proposed quaternions for the first time in 1843 [16]. Since they are helpful in computations involving three-dimensional rotations in applied and theoretical mathematics [17], they are highly significant in a number of domains, such as mathematical physics [18], computer modeling [19], quantum mechanics [20], navigation [21], electromagnetism [22], and robotics [23].

The study of dynamic problems involving time-varying quaternion matrices (TVQM) has garnered more attention recently. These problems include the linear TVQM equation [24], the pseudoinverse of TVQM [25], the constrained TVQM least-squares issue [26], and the inversion of TVQM [27]. Furthermore, real-world applications of TVQMs include chaotic system synchronization [28], robot joint kinematically redundant manipulators [29], mobile manipulator control [30], picture restoration [31], and acoustic source tracking [32]. There is a commonality throughout all of these studies: they employ the ZNN method to find the response. It is noteworthy that Zhang et al. introduced the ZNN technique in [33] to address time-varying tasks in real time. ZNNs, in particular, are recurrent neural networks that perform exceptionally well at parallel processing. Dynamical systems for computing time-varying pseudoinverses were among their subsequent applications [34,35]. Nonlinear equation systems [36,37], linear equation systems [38,39], linear/quadratic programming [40–42], and generalized inversion [43,44] are among the challenges that they are currently utilized for. A ZNN model is typically constructed via two primary steps. Firstly, we need to define the error matrix equation (EME) $E(t)$. Secondly, the dynamical system that follows has to be used:

$$\dot{E}(t) = -\lambda E(t), \quad (3)$$

where $t \in [0, t_f] \subseteq [0, +\infty)$ is the time, and the time derivative operator is $(\dot{\cdot})$. Furthermore, the convergence rate of the model can be modified by varying the parameter $\lambda \in \mathbb{R}^+$. For instance, a bigger value of λ causes any ZNN model to converge even quicker [45]. The foundation of the ZNN's architecture is to put $E(t)$ to 0, which holds as $t \rightarrow \infty$. The real-time learning rule that results from the creation of EME in (3) is used to do this. EME can therefore be regarded as a tool for tracking model learning.

In this paper, a novel ZNN model, termed 'ZQ-ARE', is presented for solving the TQARE by identifying only Hermitian solutions. One application to quadrotor control

and three simulation experiments show that the model performs admirably. Additionally, compared to the traditional differential-algebraic Riccati equation (DARE) solution, the ZQ-ARE solution adaption to the state-dependent Riccati equation (SDRE) technique demonstrates that it stabilizes a quadrotor’s flight control system more quickly. Notice that through a theoretical analysis of the ZQ-ARE model, this study contributes to the corpus of the literature. Last, the paper’s contributions are shown in the following list.

- A novel ZNN model, termed ‘ZQ-ARE’, for solving the TQARE by identifying only Hermitian solutions is introduced.
- A theoretical analysis is carried out in order to validate the ZQ-ARE model.
- Sim tests and a real-world application to quadrotor control are carried out to supplement the theoretical study.
- For the task of stabilizing the flight control system of a quadrotor, a comparison is given between the solution adaptations provided by the traditional DARE, the ZQ-ARE, and the SDRE methods.

The all-ones $p \times 1$ and $p \times n$ matrices will be indicated by $\mathbf{1}_p$ and $\mathbf{1}_{p \times n}$, respectively, while the identity $p \times p$ matrix will be indicated by I_p throughout the duration of this paper. In addition, $\otimes, \text{vec}(\cdot), \odot$ will represent the Kronecker product, the vectorization procedure, and the Hadamard product, respectively. Lastly, the matrix Frobenius norm will be represented as $\|\cdot\|_F$, and inversion and pseudoinversion will be indicated by superscripts $()^{-1}$ and $()^\dagger$, respectively.

The layout of the paper is as follows. The TQARE reformulation and quaternion preliminary results are presented in Section 2. In Section 3, the ZQ-ARE model—which is grounded in the ZNN method—is described, and in Section 4, its theoretical analysis is offered. Notice that the computational complexity of the ZQ-ARE model is described in Section 3. Simulation experiments and application to quadrotor control are shown in Section 5. Section 6 wraps up with some final thoughts and observations.

2. Preliminaries and Reformulation of the TQARE

The TQARE is reformulated and the foundations of TVQM are outlined in this part. Notice that the TQARE (2) is being reformulated in order to lower the ZNN technique’s computational complexity.

Allow $\tilde{M}(t) = M_1(t) + M_2(t)\iota + M_3(t)j + M_4(t)k \in \mathbb{H}^{n \times n}$ to represent a TVQM using the coefficient matrices $M_j(t) \in \mathbb{R}^{n \times n}$ for $j = 1, 2, 3, 4$. Next is the conjugate transpose of the TVQM $\tilde{M}(t)$ [14,46]:

$$\tilde{M}^*(t) = M_1^T(t) - M_2^T(t)\iota - M_3^T(t)j - M_4^T(t)k. \tag{4}$$

Similarly, consider the TVQMs $\tilde{C}(t), \tilde{X}(t), \tilde{B}(t), \tilde{A}(t) \in \mathbb{H}^{n \times n}$, which use the coefficient matrices $C_j(t), X_j(t), B_j(t), A_j(t) \in \mathbb{R}^{n \times n}$ for $j = 1, 2, 3, 4$. The following is the product of $\tilde{B}(t)$ and $\tilde{X}(t)$:

$$\tilde{B}(t)\tilde{X}(t) = \widetilde{BX}(t) = BX_1(t) + BX_2(t)\iota + BX_3(t)j + BX_4(t)k \in \mathbb{H}^{m \times n}, \tag{5}$$

where the following are the coefficient matrices $BX_j(t) \in \mathbb{R}^{m \times n}$:

$$\begin{aligned} BX_1(t) &= -B_3(t)X_3(t) + B_1(t)X_1(t) - B_2(t)X_2(t) - B_4(t)X_4(t), \\ BX_2(t) &= B_1(t)X_2(t) + B_3(t)X_4(t) + B_2(t)X_1(t) - B_4(t)X_3(t), \\ BX_3(t) &= B_1(t)X_3(t) + B_3(t)X_1(t) - B_2(t)X_4(t) + B_4(t)X_2(t), \\ BX_4(t) &= B_4(t)X_1(t) + B_1(t)X_4(t) - B_3(t)X_2(t) + B_2(t)X_3(t). \end{aligned} \tag{6}$$

In the same manner, we can create the products $\widetilde{AX}(t) = \tilde{A}^*(t)\tilde{X}(t)$, $\widetilde{XA}(t) = \tilde{X}(t)\tilde{A}(t)$, and $\widetilde{XBX}(t) = \tilde{X}(t)\widetilde{BX}(t)$.

Using the previously provided information, Equation (2) can be rewritten as follows:

$$\widetilde{AX}(t) + \widetilde{XA}(t) - \widetilde{XBX}(t) + \widetilde{C}(t) = \mathbf{0}_{n \times n}, \tag{7}$$

where the following is true:

$$\begin{cases} AX_1(t) + XA_1(t) - XBX_1(t) + C_1(t) = \mathbf{0}_{n \times n}, \\ AX_2(t) + XA_2(t) - XBX_2(t) + C_2(t) = \mathbf{0}_{n \times n}, \\ AX_3(t) + XA_3(t) - XBX_3(t) + C_3(t) = \mathbf{0}_{n \times n}, \\ AX_4(t) + XA_4(t) - XBX_4(t) + C_4(t) = \mathbf{0}_{n \times n}. \end{cases} \tag{8}$$

Then, setting

$$\begin{aligned} Y(t) &= \begin{bmatrix} X_1(t) & -X_2(t) & -X_3(t) & -X_4(t) \\ X_2(t) & X_1(t) & -X_4(t) & X_3(t) \\ X_3(t) & X_4(t) & X_1(t) & -X_2(t) \\ X_4(t) & -X_3(t) & X_2(t) & X_1(t) \end{bmatrix}, & A(t) &= \begin{bmatrix} A_1(t) \\ A_2(t) \\ A_3(t) \\ A_4(t) \end{bmatrix}, & C(t) &= \begin{bmatrix} C_1(t) \\ C_2(t) \\ C_3(t) \\ C_4(t) \end{bmatrix}, \\ W(t) &= \begin{bmatrix} B_1(t) & -B_2(t) & -B_3(t) & -B_4(t) \\ B_2(t) & B_1(t) & -B_4(t) & B_3(t) \\ B_3(t) & B_4(t) & B_1(t) & -B_2(t) \\ B_4(t) & -B_3(t) & B_2(t) & B_1(t) \end{bmatrix}, & B(t) &= \begin{bmatrix} B_1(t) \\ B_2(t) \\ B_3(t) \\ B_4(t) \end{bmatrix}, & X(t) &= \begin{bmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \\ X_4(t) \end{bmatrix}, \\ D(t) &= \begin{bmatrix} A_1^T(t) & A_2^T(t) & A_3^T(t) & A_4^T(t) \\ -A_2^T(t) & A_1^T(t) & A_4^T(t) & -A_3^T(t) \\ -A_3^T(t) & -A_4^T(t) & A_1^T(t) & A_2^T(t) \\ -A_4^T(t) & A_3^T(t) & -A_2^T(t) & A_1^T(t) \end{bmatrix}, \end{aligned} \tag{9}$$

where $Y(t), W(t), D(t) \in \mathbb{R}^{4n \times 4n}$ and $X(t), B(t), C(t), A(t) \in \mathbb{R}^{4n \times n}$, Equation (7) can be reconstructed as follows:

$$D(t)X(t) + Y(t)A(t) - Y(t)W(t)X(t) + C(t) = \mathbf{0}_{4n \times n}, \tag{10}$$

where the coefficient matrices of $\tilde{X}(t)$, $X_j(t)$, $j = 1, 2, 3, 4$, are contained in $X(t)$ and $Y(t)$, respectively. Remember that $\tilde{X}(t)$ is the intended solution to the TQARE (2). Notice also that (10) provides four real-valued time-varying matrices, as opposed to (2), which only yields one TVQM.

3. ZNN Modification for the TQARE System

To solve the TQARE, we will create a ZNN model called 'ZQ-ARE' in this section. Let $\tilde{A}(t), \tilde{B}(t), \tilde{C}(t), \tilde{X}(t) \in \mathbb{H}^{n \times n}$ be differentiable TVQMs. Also, let $\tilde{B}(t)$ and $\tilde{C}(t)$ be positive semidefinite and $\tilde{X}(t)$ be an unknown Hermitian solution. According to the analysis of Section 2, (10) is a reformulation of (2). Following (9), we build the matrices $D(t), W(t) \in \mathbb{R}^{4n \times 4n}$ and $A(t), B(t), C(t) \in \mathbb{R}^{4n \times n}$, accounting for the following EME:

$$E^A(t) = Y(t)A(t) + D(t)X(t) - Y(t)W(t)X(t) + C(t), \tag{11}$$

where $X(t) \in \mathbb{R}^{4n \times 4n}$ and $Y(t) \in \mathbb{R}^{4n \times n}$ are the matrices of interest to be found. The first derivative of (11) is:

$$\begin{aligned} \dot{E}^A(t) &= Y(t)\dot{A}(t) + \dot{Y}(t)A(t) + \dot{D}(t)X(t) + D(t)\dot{X}(t) \\ &\quad - \dot{Y}(t)W(t)X(t) - Y(t)\dot{W}(t)X(t) - Y(t)W(t)\dot{X}(t) + \dot{C}(t). \end{aligned} \tag{12}$$

The ZNN dynamical system is then treated in terms of $\dot{Y}(t)$ and $\dot{X}(t)$. This is achieved by substituting $E^A(t)$ and $\dot{E}^A(t)$ in (12) for $E(t)$ and $\dot{E}(t)$ in (3), respectively. The result is as follows:

$$D(t)\dot{X}(t) + \dot{Y}(t)A(t) - \dot{Y}(t)W(t)X(t) - Y(t)W(t)\dot{X}(t) = -\lambda E^A(t) - \dot{D}(t)X(t) - Y(t)\dot{A}(t) + Y(t)\dot{W}(t)X(t) - \dot{C}(t). \tag{13}$$

Next, we apply the Kronecker product and vectorization to streamline the dynamics of (13):

$$(I_n \otimes D(t) - I_n \otimes (Y(t)W(t)))\text{vec}(\dot{X}(t)) + (A^T(t) \otimes I_{4n} - (W(t)X(t))^T \otimes I_{4n})\text{vec}(\dot{Y}(t)) = \text{vec}(-\lambda E^A(t) - \dot{D}(t)X(t) - Y(t)\dot{A}(t) + Y(t)\dot{W}(t)X(t) - \dot{C}(t)). \tag{14}$$

It is crucial to notice that the vectors $\text{vec}(\dot{X}(t))$ and $\text{vec}(\dot{Y}(t))$ contain the same components but in different places. Put differently, by expressing $\text{vec}(\dot{Y}(t))$ with regard to $\text{vec}(\dot{X}(t))$, Equation (14) can be further simplified. Hence, the following equation can be written by replacing $\text{vec}(\dot{Y}(t))$ in (14):

$$\text{vec}(\dot{Y}(t)) = R\text{vec}(\dot{X}(t)), \tag{15}$$

where the operational matrix $R \in \mathbb{R}^{(4n)^2 \times (2n)^2}$ can be computed using the algorithmic procedure described in Algorithm 1. Observe that Algorithm 1’s notations adhere to the standard MATLAB function theme [47].

Algorithm 1 Calculation of matrix R

Require: The order n of the square matrix $X(t)$.

- 1: **procedure** OM_R(n)
- 2: Put $x = (1 : (2n)^2)'$, $X = \text{reshape}(x, 4n, n)$, and $R = \text{zeros}((4n)^2, (2n)^2)$
- 3: Put $F1 = X(1 : n, :)$, $F2 = X(n + 1 : 2n, :)$, $F3 = X(2n + 1 : 3n, :)$, $F4 = X(3n + 1 : 4n, :)$
- 4: Put $F = [F1, -F2, -F3, -F4; F2, F1, -F4, F3; F3, F4, F1, -F2; F4, -F3, F2, F1]$
- 5: Put $Y = \text{reshape}(F, [], 1)$
- 6: **for** $c = 1 : (4n)^2$ **do**
- 7: $R(c, \text{abs}(Y(c))) = \text{sign}(Y(c))$
- 8: **end for**
- 9: **end procedure**

Ensure: The matrix R .

Equation (14) may be simplified even more by utilizing (15) as follows:

$$\left((I_n \otimes D(t) - I_n \otimes (Y(t)W(t))) + (A^T(t) \otimes I_{4n} - (W(t)X(t))^T \otimes I_{4n})R \right) \text{vec}(\dot{X}(t)) = \text{vec}(-\lambda E^A(t) - Y(t)\dot{A}(t) - \dot{D}(t)X(t) + Y(t)\dot{W}(t)X(t) - \dot{C}(t)). \tag{16}$$

Because the aim of this study is to find explicitly Hermitian solutions, we need to locate only the components of $X_1(t)$ placed on and above its main diagonal, which are $n(n + 1)/2$ in number; and the components of $X_j(t), j = 2, 3, 4$, positioned above its main diagonal, which are $n(n - 1)/2$ in number. So it is crucial to utilize $\dot{x}(t)$ in place of $\dot{X}(t)$ by placing the aforementioned $r = n(n + 1)/2 + 3n(n - 1)/2 = 2n^2 - n$ in number components of $X_j(t), j = 1, 2, 3, 4$, into the vector $\dot{x}(t) \in \mathbb{R}^r$. The following equation may be constructed to replace $\text{vec}(\dot{X}(t)) \in \mathbb{R}^{(2n)^2}$ in (16):

$$\text{vec}(\dot{X}(t)) = Z\dot{x}(t), \tag{17}$$

where $\dot{x}(t) \in \mathbb{R}^r$, and the operational matrix $Z \in \mathbb{R}^{(2n)^2 \times r}$ can be computed using the algorithmic procedure described in Algorithm 2. This method reduces (16)’s dimension while forcing $\dot{X}(t)$ to be a Hermitian matrix. Observe that Algorithm 2’s notations adhere to the standard MATLAB function theme [47].

Algorithm 2 Calculation of matrix Z

Require: The order n of the square matrix $X(t)$.

```

1: procedure OM_Z( $n$ )
2:   Put  $r_1 = n(n + 1)/2$ ,  $r_2 = n(n - 1)/2$ ,  $Z_1 = \text{zeros}(n^2, r_1)$ , and  $Z_2 = \text{zeros}(n^2, r_2)$ 
3:   for  $j = 1 : n^2$  do
4:     Put  $h = 1 + \text{floor}(\frac{-1+j}{n})$  and  $d = 1 + \text{mod}(-1 + j, n)$ 
5:     if  $d < h$  then
6:       Put  $Z_1(j, d + h\frac{h-1}{2}) = 1$  and  $Z_2(j, d - h + 1 + h\frac{h-1}{2}) = -1$ 
7:     else if  $d == h$  then
8:       Put  $Z_1(j, d + h\frac{h-1}{2}) = 1$ 
9:     else
10:      Put  $Z_1(j, h + d\frac{d-1}{2}) = 1$  and  $Z_2(j, h - d + 1 + d\frac{d-1}{2}) = 1$ 
11:    end if
12:  end for
13:  Set  $Z = \text{blkdiag}(Z_1, Z_2, Z_2, Z_2)$  and  $k = []$ 
14:  for  $j = 1 : n$  do
15:    Put  $k = [k, j : n : (2n)^2 - n + j]$ 
16:  end for
17:  return  $Z = Z(k, :)$ 
18: end procedure

```

Ensure: The matrix Z .

Equation (16) may be simplified even more by utilizing (17) as follows:

$$\begin{aligned} & \left(I_n \otimes D(t) - I_n \otimes (Y(t)W(t)) + (A^T(t) \otimes I_{4n} - (W(t)X(t))^T \otimes I_{4n})R \right) Z \dot{\mathbf{x}}(t) = \\ & \text{vec}(-\lambda E^A(t) - \dot{D}(t)X(t) - Y(t)\dot{A}(t) + Y(t)\dot{W}(t)X(t) - \dot{C}(t)). \end{aligned} \tag{18}$$

Furthermore, after the following have been established:

$$\begin{aligned} K(t) &= \left(I_n \otimes D(t) - I_n \otimes (Y(t)W(t)) + (A^T(t) \otimes I_{4n} - (W(t)X(t))^T \otimes I_{4n})R \right) Z \in \mathbb{R}^{(2n)^2 \times r}, \\ L(t) &= \text{vec}(-\lambda E^A(t) - \dot{D}(t)X(t) - Y(t)\dot{A}(t) + Y(t)\dot{W}(t)X(t) - \dot{C}(t)) \in \mathbb{R}^{(2n)^2}, \end{aligned} \tag{19}$$

we arrive at the following ZNN model:

$$\dot{\mathbf{x}}(t) = K^\dagger(t)L(t). \tag{20}$$

Equation (20)'s dynamic model, additionally referred to as ZQ-ARE, is the suggested ZNN model to be used for solving the TQARE of (2).

In addition to the ZQ-ARE itself, the process of generating and solving (20) is computationally demanding. In particular, because we perform r^2 multiplications and r subtractions/additions at each iteration, the complexity of generating (20) is $\mathcal{O}(r^2)$ operations. Furthermore, the equation's linear system is solved step-by-step using the implicit MATLAB solver ode15s. Addressing (20) requires an $r \times r$ matrix, which adds $\mathcal{O}(r^3)$ in complexity. Keep in mind that $r = 2n^2 - n$. Hence, the computational complexity of the ZQ-ARE is $\mathcal{O}(r^3) = \mathcal{O}((2n^2 - n)^3)$.

4. Convergence and Stability Analysis

The ZQ-ARE (20) model's stability analysis and convergence is presented in this section. It is significant to note that theoretical support has been carefully taken into account at every step of building the ZNN model. Firstly, we construct the EME according to the time-variant problem to be solved [33], and this means it monitors the ZNN's process of solving the TQARE of (2). This EME is presented in (11) for the ZQ-ARE model. Secondly, we design an evolutionary formula in (13) based on the ZNN design in (3) to make the

EME approach zero. Finally, the relative ZQ-ARE model in (20) can be derived by these two steps. Additionally, it is worth mentioning that the Lyapunov stability theory is applied in this section to comprehensively guarantee the stability of the ZQ-ARE model.

Theorem 1. Let $W(t), D(t) \in \mathbb{R}^{4n \times 4n}$ and $A(t), B(t), C(t) \in \mathbb{R}^{4n \times n}$ be differentiable. The theoretical solution (TSOL) $X_{\mathcal{T}\mathcal{S}}(t)$ is reached by the dynamics of (13) in line with the ZNN method (3), and it is stable in accordance with Lyapunov.

Proof. $X(t) = X_{\mathcal{T}\mathcal{S}}(t) - X_{\mathcal{O}}(t)$ follows from the substitution $X_{\mathcal{O}}(t) := -X(t) + X_{\mathcal{T}\mathcal{S}}(t)$, where $X_{\mathcal{T}\mathcal{S}}(t)$ is the TSOL. The elements of $X(t)$ are rearranged in $Y(t)$, as per (9). Consequently, since $Y_{\mathcal{T}\mathcal{S}}(t)$ and $Y_{\mathcal{O}}(t)$ respectively represent a reposition of the elements of $X_{\mathcal{T}\mathcal{S}}(t)$ and $X_{\mathcal{O}}(t)$, then $Y(t) = Y_{\mathcal{T}\mathcal{S}}(t) - Y_{\mathcal{O}}(t)$. Moreover, $\dot{Y}(t) = \dot{Y}_{\mathcal{T}\mathcal{S}}(t) - \dot{Y}_{\mathcal{O}}(t)$ and $\dot{X}(t) = \dot{X}_{\mathcal{T}\mathcal{S}}(t) - \dot{X}_{\mathcal{O}}(t)$ are the time-derivatives of $Y(t)$ and $X(t)$, respectively.

Let

$$D(t)X_{\mathcal{T}\mathcal{S}}(t) + Y_{\mathcal{T}\mathcal{S}}(t)A(t) - Y_{\mathcal{T}\mathcal{S}}(t)W(t)X_{\mathcal{T}\mathcal{S}}(t) + C(t) = \mathbf{0}_{4n \times n}, \tag{21}$$

and its first derivative

$$\begin{aligned} D(t)\dot{X}_{\mathcal{T}\mathcal{S}}(t) + \dot{D}(t)X_{\mathcal{T}\mathcal{S}}(t) + \dot{Y}_{\mathcal{T}\mathcal{S}}(t)A(t) + Y_{\mathcal{T}\mathcal{S}}(t)\dot{A}(t) - \dot{Y}_{\mathcal{T}\mathcal{S}}(t)W(t)X_{\mathcal{T}\mathcal{S}}(t) \\ - Y_{\mathcal{T}\mathcal{S}}(t)\dot{W}(t)X_{\mathcal{T}\mathcal{S}}(t) - Y_{\mathcal{T}\mathcal{S}}(t)W(t)\dot{X}_{\mathcal{T}\mathcal{S}}(t) + \dot{C}(t) = \mathbf{0}_{4n \times n}. \end{aligned} \tag{22}$$

After substituting (11) for $X(t) = X_{\mathcal{T}\mathcal{S}}(t) - X_{\mathcal{O}}(t)$ and $Y(t) = Y_{\mathcal{T}\mathcal{S}}(t) - Y_{\mathcal{O}}(t)$, the following may be verified:

$$\begin{aligned} E_{\mathcal{T}\mathcal{S}}^A(t) = D(t)X_{\mathcal{T}\mathcal{S}}(t) - D(t)X_{\mathcal{O}}(t) + Y_{\mathcal{T}\mathcal{S}}(t)A(t) - Y_{\mathcal{O}}(t)A(t) - Y_{\mathcal{T}\mathcal{S}}(t)W(t)X_{\mathcal{T}\mathcal{S}}(t) \\ + Y_{\mathcal{O}}(t)W(t)X_{\mathcal{T}\mathcal{S}}(t) + Y_{\mathcal{T}\mathcal{S}}(t)W(t)X_{\mathcal{O}}(t) - Y_{\mathcal{O}}(t)W(t)X_{\mathcal{O}}(t) + C(t), \end{aligned} \tag{23}$$

and the following results from (3):

$$\begin{aligned} E_{\mathcal{T}\mathcal{S}}^A(t) = D(t)\dot{X}_{\mathcal{T}\mathcal{S}}(t) + \dot{D}(t)X_{\mathcal{T}\mathcal{S}}(t) - D(t)\dot{X}_{\mathcal{O}}(t) - \dot{D}(t)X_{\mathcal{O}}(t) + \dot{Y}_{\mathcal{T}\mathcal{S}}(t)A(t) + Y_{\mathcal{T}\mathcal{S}}(t)\dot{A}(t) \\ - \dot{Y}_{\mathcal{O}}(t)A(t) - Y_{\mathcal{O}}(t)\dot{A}(t) + \dot{C}(t) - \dot{Y}_{\mathcal{T}\mathcal{S}}(t)W(t)X_{\mathcal{T}\mathcal{S}}(t) - Y_{\mathcal{T}\mathcal{S}}(t)\dot{W}(t)X_{\mathcal{T}\mathcal{S}}(t) \\ - Y_{\mathcal{T}\mathcal{S}}(t)W(t)\dot{X}_{\mathcal{T}\mathcal{S}}(t) + \dot{Y}_{\mathcal{O}}(t)W(t)X_{\mathcal{T}\mathcal{S}}(t) + Y_{\mathcal{O}}(t)\dot{W}(t)X_{\mathcal{T}\mathcal{S}}(t) \\ + Y_{\mathcal{O}}(t)W(t)\dot{X}_{\mathcal{T}\mathcal{S}}(t) + \dot{Y}_{\mathcal{T}\mathcal{S}}(t)W(t)X_{\mathcal{O}}(t) + Y_{\mathcal{T}\mathcal{S}}(t)\dot{W}(t)X_{\mathcal{O}}(t) \\ + Y_{\mathcal{T}\mathcal{S}}(t)W(t)\dot{X}_{\mathcal{O}}(t) - \dot{Y}_{\mathcal{O}}(t)W(t)X_{\mathcal{O}}(t) - Y_{\mathcal{O}}(t)\dot{W}(t)X_{\mathcal{O}}(t) - Y_{\mathcal{O}}(t)W(t)\dot{X}_{\mathcal{O}}(t) \\ = -\lambda E_{\mathcal{T}\mathcal{S}}^A(t). \end{aligned} \tag{24}$$

Afterward, in order to confirm convergence, we select the probable Lyapunov function listed below:

$$\mathcal{P}(t) = \frac{1}{2} \left\| E_{\mathcal{T}\mathcal{S}}^A(t) \right\|_F^2 = \frac{1}{2} \text{TR} \left(E_{\mathcal{T}\mathcal{S}}^A(t) \left(E_{\mathcal{T}\mathcal{S}}^A(t) \right)^T \right). \tag{25}$$

The following can then be verified:

$$\dot{\mathcal{P}}(t) = \frac{2 \text{TR} \left(\left(E_{\mathcal{T}\mathcal{S}}^A(t) \right)^T \dot{E}_{\mathcal{T}\mathcal{S}}^A(t) \right)}{2} = \text{TR} \left(\left(E_{\mathcal{T}\mathcal{S}}^A(t) \right)^T \dot{E}_{\mathcal{T}\mathcal{S}}^A(t) \right) = -\lambda \text{TR} \left(\left(E_{\mathcal{T}\mathcal{S}}^A(t) \right)^T E_{\mathcal{T}\mathcal{S}}^A(t) \right). \tag{26}$$

Because of this, the following is valid:

$$\begin{aligned}
 \dot{\mathcal{P}}(t) & \begin{cases} = 0, & E_{\mathcal{T}\mathcal{S}}^A(t) = 0 \\ < 0, & E_{\mathcal{T}\mathcal{S}}^A(t) \neq 0, \end{cases} \\
 \Leftrightarrow \dot{\mathcal{P}}(t) & \begin{cases} = 0, & D(t)X_{\mathcal{T}\mathcal{S}}(t) - D(t)X_{\mathcal{O}}(t) + Y_{\mathcal{T}\mathcal{S}}(t)A(t) - Y_{\mathcal{O}}(t)A(t) - Y_{\mathcal{T}\mathcal{S}}(t)W(t)X_{\mathcal{T}\mathcal{S}}(t) \\ & + Y_{\mathcal{O}}(t)W(t)X_{\mathcal{T}\mathcal{S}}(t) + Y_{\mathcal{T}\mathcal{S}}(t)W(t)X_{\mathcal{O}}(t) - Y_{\mathcal{O}}(t)W(t)X_{\mathcal{O}}(t) + C(t) = 0 \\ < 0, & D(t)X_{\mathcal{T}\mathcal{S}}(t) - D(t)X_{\mathcal{O}}(t) + Y_{\mathcal{T}\mathcal{S}}(t)A(t) - Y_{\mathcal{O}}(t)A(t) - Y_{\mathcal{T}\mathcal{S}}(t)W(t)X_{\mathcal{T}\mathcal{S}}(t) \\ & + Y_{\mathcal{O}}(t)W(t)X_{\mathcal{T}\mathcal{S}}(t) + Y_{\mathcal{T}\mathcal{S}}(t)W(t)X_{\mathcal{O}}(t) - Y_{\mathcal{O}}(t)W(t)X_{\mathcal{O}}(t) + C(t) \neq 0, \end{cases} \quad (27) \\
 \Leftrightarrow \dot{\mathcal{P}}(t) & \begin{cases} = 0, & X_{\mathcal{O}}(t) = 0 \ \& \ Y_{\mathcal{O}}(t) = 0 \\ < 0, & X_{\mathcal{O}}(t) \neq 0 \ \& \ Y_{\mathcal{O}}(t) \neq 0, \end{cases} \quad \Leftrightarrow \quad \dot{\mathcal{P}}(t) \begin{cases} = 0, & X_{\mathcal{O}}(t) = 0 \\ < 0, & X_{\mathcal{O}}(t) \neq 0. \end{cases}
 \end{aligned}$$

Observe that because $Y(t)$ is a reorganization of the $X(t)$ entries, $Y_{\mathcal{O}}(t) \neq 0$ when $X_{\mathcal{O}}(t) \neq 0$, and $Y_{\mathcal{O}}(t) = 0$ otherwise. Furthermore, since $E^A(0) = 0$ and $X_{\mathcal{O}}(t)$ are the equilibrium points of (24), the following is true:

$$\dot{\mathcal{P}}(t) \leq 0, \quad \forall X_{\mathcal{O}}(t) \neq 0. \quad (28)$$

We find that the equilibrium state $X_{\mathcal{O}}(t) = -X(t) + X_{\mathcal{T}\mathcal{S}}(t) = 0$ is stable based on the theory of Lyapunov. After that, $t \rightarrow \infty$ and $X(t) \rightarrow X_{\mathcal{T}\mathcal{S}}(t)$. Observe that since $Y(t)$ is a reposition of the $X(t)$ elements, $t \rightarrow \infty$ and $Y(t) \rightarrow Y_{\mathcal{T}\mathcal{S}}(t)$. \square

Theorem 2. Let $\tilde{B}(t), \tilde{C}(t), \tilde{A}(t) \in \mathbb{H}^{n \times n}$ be differentiable. Also, suppose that $\tilde{C}(t)$ and $\tilde{B}(t)$ are positive semidefinite matrices. At each $t \in [0, t_f] \subseteq [0, +\infty)$, the ZQ-ARE model (20) converges to the TSOL $x_{\mathcal{T}\mathcal{S}}(t)$ exponentially for any choice of starting value $x(0)$.

Proof. First, using the analysis presented in Section 2, the TQARE of (2) is transformed into (10). More specifically, we use (9) to build the matrices $D(t), W(t) \in \mathbb{R}^{4n \times 4n}$ and $A(t), B(t), C(t) \in \mathbb{R}^{4n \times n}$ using the matrices $\tilde{A}(t), \tilde{B}(t)$ and $\tilde{C}(t)$. We therefore change (2) into (10). Secondly, the EME of (11) is declared in order to solve (10). The model (13) is used in accordance with the ZNN method (3) for zeroing (11). Theorem 1 states that when $t \rightarrow \infty$ for any choice of starting value, $Y(t) \rightarrow Y_{\mathcal{T}\mathcal{S}}(t)$ and $X(t) \rightarrow X_{\mathcal{T}\mathcal{S}}(t)$. Keep in mind that the $X(t)$ elements are rearranged into $Y(t)$. Since (10) is a reformulation of the TQARE of (2), the model (13) converges to Equation (2)’s TSOL. Third, the Kronecker product, vectorization, and the operation matrices of (15) and (17) are used to simplify (13) and form the ZQ-ARE model of (20). An alternate form of (13) is the ZQ-ARE, which converges to the TSOL $x_{\mathcal{T}\mathcal{S}}(t)$ when $t \rightarrow \infty$ for any starting value $x(0)$. This completes the proof. \square

5. Simulations and Application

This section will present an application to quadrotor control as well as three simulation examples (SEs). It is important to mention a few key justifications. In the SEs, we assigned $\eta(t) = \cos(t)$ as well as $\gamma(t) = \sin(t)$ for simplicity. Moreover, the MATLAB ode solver ode15s is used for all computations, and its time interval is set to $[0, 10]$. Notice that by using ode15s and its normal double-precision arithmetic (i.e., $\epsilon = 0.22 \times 10^{-15}$), the majority of errors in this section’s figures have minimum values that are in the neighborhood of 10^{-5} .

5.1. Walkthrough Tests

5.1.1. Example 1

The coefficients of matrix $\tilde{A}(t)$ in this SE are configured as follows:

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

the coefficients of matrix $\tilde{B}(t)$ are configured as follows:

$$B_1(t) = \begin{bmatrix} \eta(t) & \gamma(t) + 3 \\ \gamma(t) + 3 & \gamma(t) + 2 \end{bmatrix}, \quad B_2(t) = \begin{bmatrix} 0 & 6 \\ -6 & 0 \end{bmatrix},$$

$$B_3(t) = \begin{bmatrix} 0 & 5 + \eta(t) \\ -5 - \eta(t) & 0 \end{bmatrix}, \quad B_4(t) = \begin{bmatrix} 0 & \eta(t) \\ -\eta(t) & 0 \end{bmatrix},$$

and the coefficients of matrix $\tilde{C}(t)$ are configured as follows:

$$C_1(t) = \begin{bmatrix} \gamma(t) + 5 & \eta(t) \\ \eta(t) & \eta(t) + 2 \end{bmatrix}, \quad C_2(t) = \begin{bmatrix} 0 & 5 + \gamma(t) \\ -5 - \gamma(t) & 0 \end{bmatrix},$$

$$C_3(t) = \begin{bmatrix} 0 & 6 \\ -6 & 0 \end{bmatrix}, \quad C_4(t) = \begin{bmatrix} 0 & \gamma(t) \\ -\gamma(t) & 0 \end{bmatrix}.$$

Hence, $\tilde{A}(t), \tilde{B}(t), \tilde{C}(t) \in \mathbb{H}^{2 \times 2}$. Additionally, the initial condition of the ZQ-ARE model has been set to $\mathbf{x}(0) = 100 \odot \mathbf{1}_6$, and the parameter λ is employed with values of 10 and 100 in order to verify the convergence features of the ZNN technique. The ZQ-ARE model's findings are depicted in Figure 1.

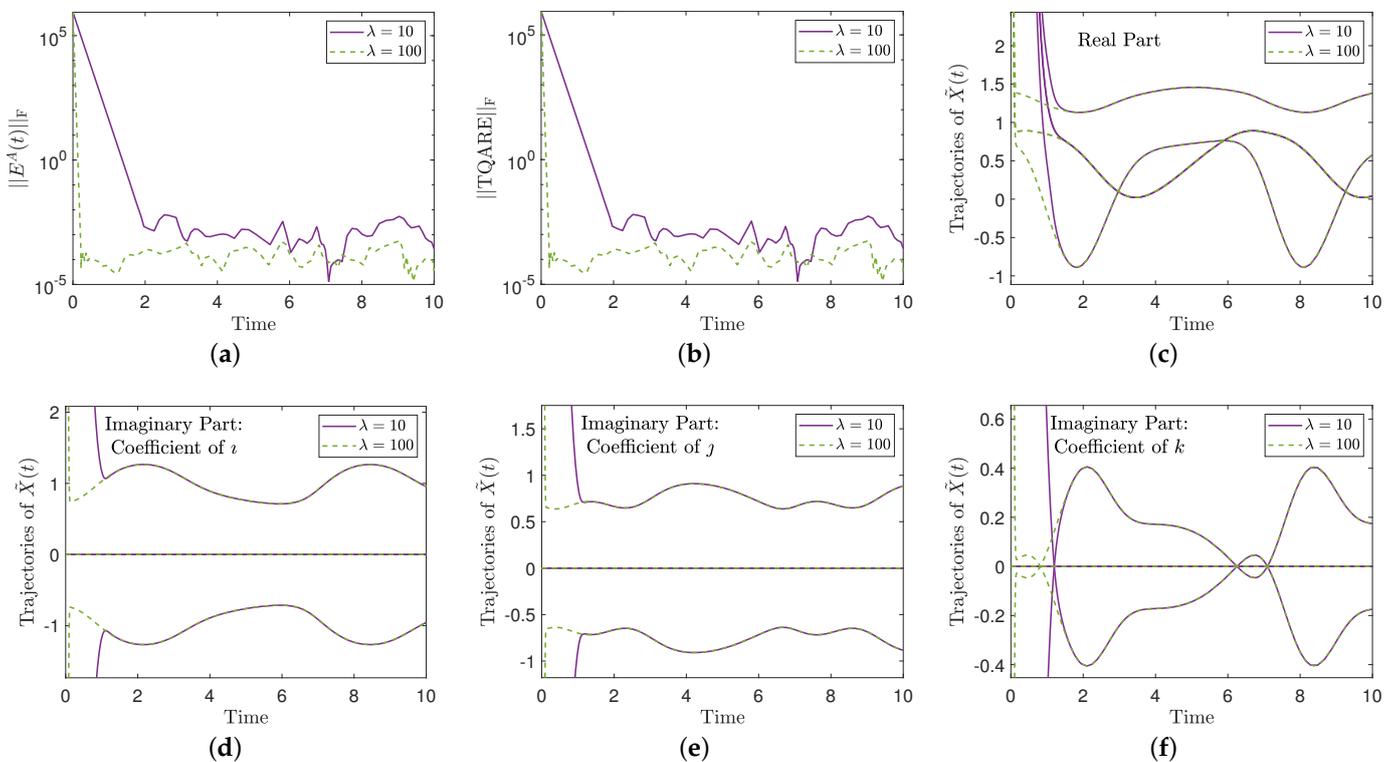


Figure 1. Equation (2) error, solution trajectories, and EMEs of ZQ-ARE from Section 5.1.1: (a) EMEs, (b) Equation (2) error, (c) real part of the trajectory, (d) imaginary part of the trajectory for i , (e) imaginary part of the trajectory for j , and (f) imaginary part of the trajectory for k .

5.1.2. Example 2

Consider the following 3×3 real matrices:

$$M_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad M_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}.$$

Then, the coefficients of matrix $\tilde{A}(t)$ in this SE are configured as follows:

$$A_1(t) = \mathbf{1}_{3 \times 3} + I_3 \odot \gamma(t), \quad A_2(t) = M_1 \odot (1 + \gamma(t)),$$

$$A_3(t) = M_2 \odot (2 + 2\gamma(t)), \quad A_4(t) = M_3 \odot (3 + \gamma(t)),$$

the coefficients of matrix $\tilde{B}(t)$ are configured as follows:

$$B_1(t) = I_3 \odot (1 + \gamma(t)), \quad B_2(t) = M_1 \odot (4 + 2\eta(t)),$$

$$B_3(t) = M_2 \odot (1 + \eta(t)), \quad B_4(t) = M_3 \odot (3 + 2\eta(t)),$$

and the coefficients of matrix $\tilde{C}(t)$ are configured as follows:

$$C_1(t) = I_3 \odot (1 + \eta(t)), \quad C_2(t) = M_1 \odot (5 + \eta(t)),$$

$$C_3(t) = M_2 \odot (7 + \eta(t)), \quad C_4(t) = M_3 \odot (1 + 3\eta(t)).$$

Therefore, $\tilde{A}(t), \tilde{B}(t), \tilde{C}(t) \in \mathbb{H}^{3 \times 3}$. Additionally, the parameter λ of the ZQ-ARE model is utilized with a value of 10, and the initial conditions (ICs) are set as follows:

- IC1: $\mathbf{x}(0) = \mathbf{0}_{15}$;
- IC2: $\mathbf{x}(0) = 100 \odot \mathbf{1}_{15}$.

Notice that we use two ICs in order to verify the results of Theorem 2. The ZQ-ARE model's findings are depicted in Figure 2.

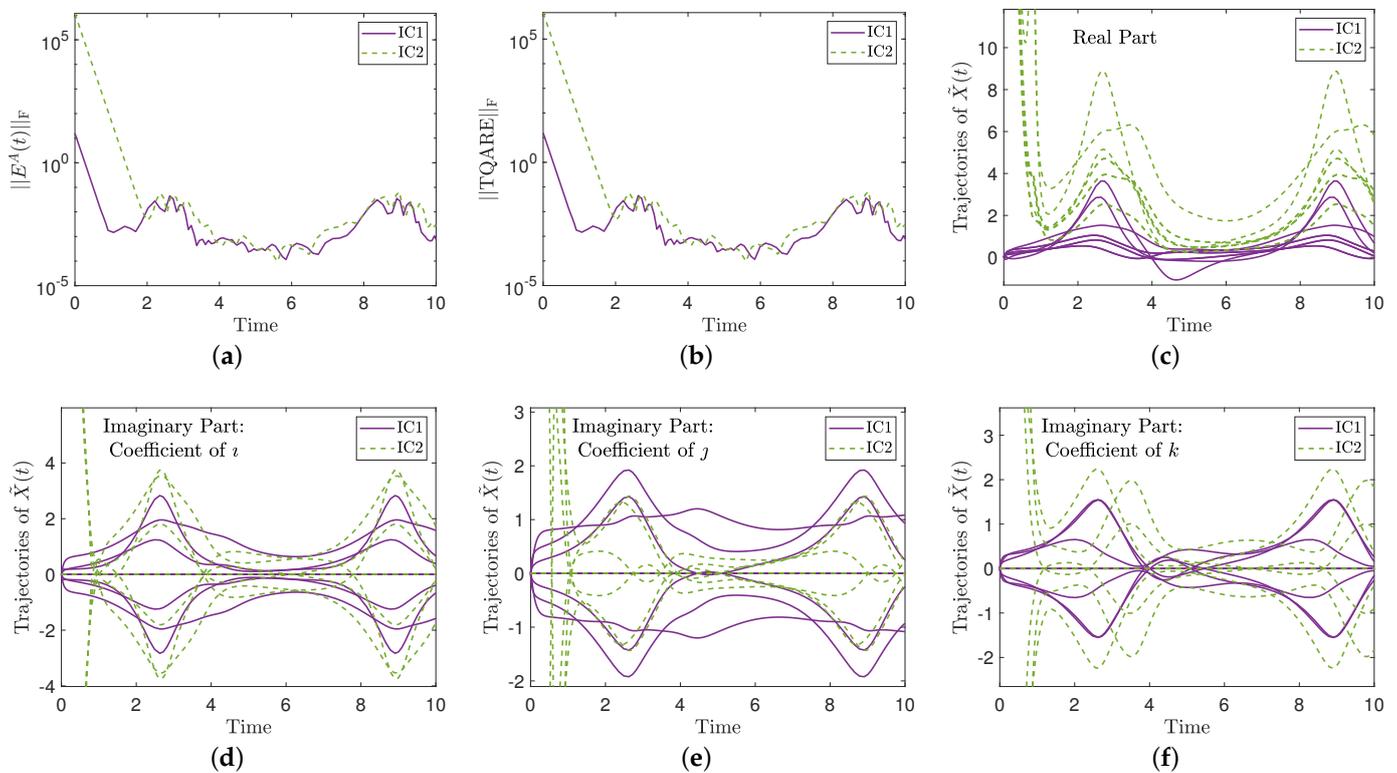


Figure 2. Equation (2) error, solution trajectories, and EMEs of ZQ-ARE from Section 5.1.2: (a) EMEs, (b) Equation (2) error, (c) real part of the trajectory, (d) imaginary part of the trajectory for i , (e) imaginary part of the trajectory for j , and (f) imaginary part of the trajectory for k .

5.1.3. Example 3

Consider the following 6×6 real matrices:

$$M_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad M_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Then, the coefficients of matrix $\tilde{A}(t)$ in this SE are configured as follows:

$$A_1(t) = \mathbf{1}_{6 \times 6} + I_6 \odot \gamma(t), \quad A_2(t) = M_1 \odot (1 + \gamma(t)),$$

$$A_3(t) = M_2 \odot (2 + 2\gamma(t)), \quad A_4(t) = M_3 \odot (3 + \gamma(t)),$$

the coefficients of matrix $\tilde{B}(t)$ are configured as follows:

$$B_1(t) = I_6 \odot (1 + \gamma(t)), \quad B_2(t) = M_1 \odot (4 + 2\eta(t)),$$

$$B_3(t) = M_2 \odot (1 + \eta(t)), \quad B_4(t) = M_3 \odot (3 + 2\eta(t)),$$

and matrix $\tilde{C}(t) = \tilde{B}(t)$. Therefore, $\tilde{A}(t), \tilde{B}(t), \tilde{C}(t) \in \mathbb{H}^{6 \times 6}$. Additionally, the parameter λ of the ZQ-ARE model is utilized with a value of 10, and the initial condition is set as $\mathbf{x}(0) = \text{round}(\mathbf{x}_{\mathcal{T}S}(0))$, where $\text{round}(\cdot)$ denotes an element-wise round function of all inputs, and $\mathbf{x}_{\mathcal{T}S}$ is the eigenvector solution [48]. It is important to note that in order for the ZQ-ARE model to find a solution that matches a certain TSOL, we utilize an initial condition that is near that TSOL. The ZQ-ARE model’s findings are depicted in Figure 3.

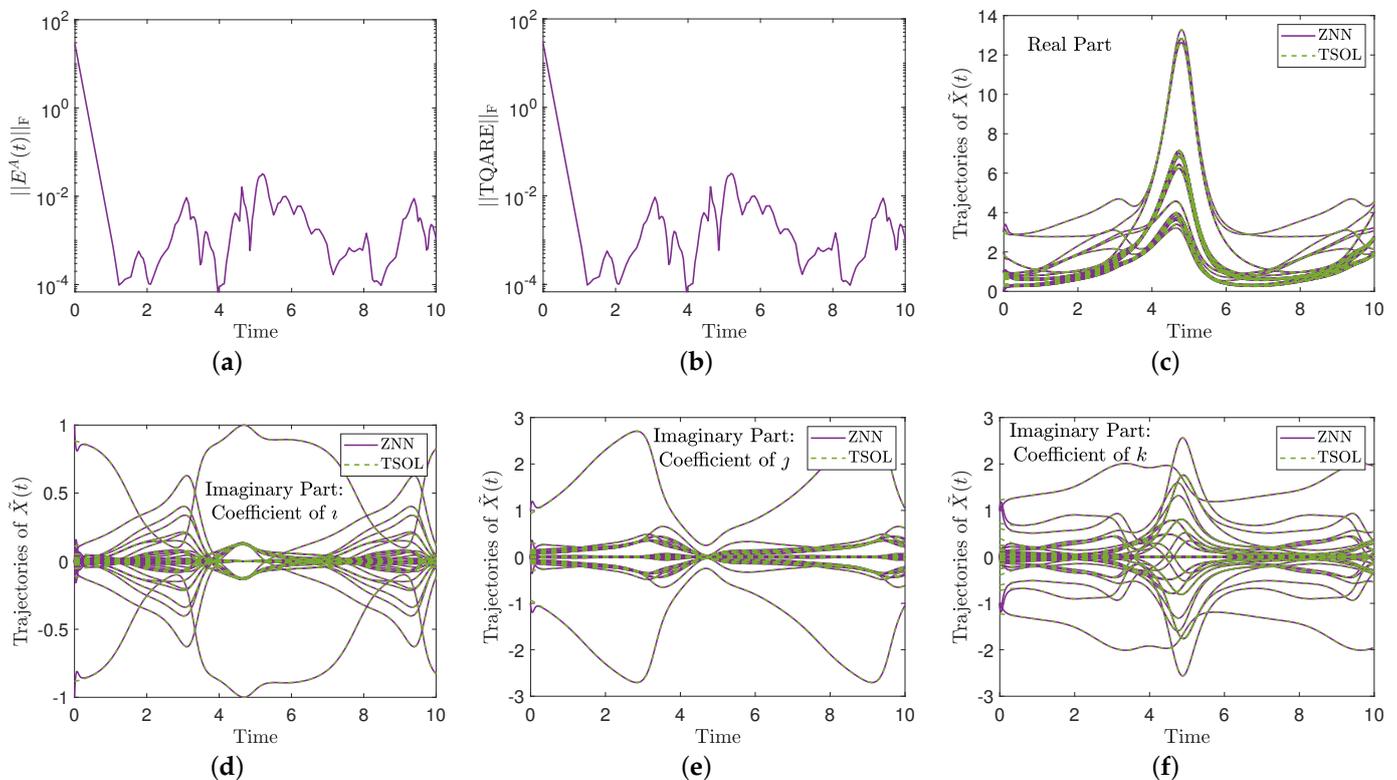


Figure 3. Equation (2) error, solution trajectories, and EME of ZQ-ARE from Section 5.1.3: (a) EME, (b) Equation (2) error, (c) real part of the trajectory, (d) imaginary part of the trajectory for i , (e) imaginary part of the trajectory for j , and (f) imaginary part of the trajectory for k .

5.2. Discussion of the SE Findings

Throughout Sections 5.1.1–5.1.3, the effectiveness of the ZQ-ARE (20) model for solving the TQARE of (2) is examined. Because of the appropriate matrices $\tilde{B}(t)$, $\tilde{C}(t)$, and $\tilde{A}(t)$, each SE has a distinct TQARE.

It can be seen that $\tilde{A}(t), \tilde{B}(t), \tilde{C}(t) \in \mathbb{H}^{2 \times 2}$ in Section 5.1.1. In other words, $n = 2$ in (2)’s TQARE. For $\lambda = 10$ and $\lambda = 100$, we have the following results for the ZQ-ARE model. Figure 1a illustrates the ZQ-ARE model’s EMEs. The two cases in this illustration begin with a big error value at $t = 0$ and end up with a tiny error value in the interval $[10^{-5}, 10^{-3}]$ at $t = 2$ when $\lambda = 10$ and at $t = 0.1$ when $\lambda = 100$. Stated differently, the convergence features of the ZNN approach are verified by the EME of the ZQ-ARE

model, which is influenced by the value of λ . The ZQ-ARE model's (2) error is displayed in Figure 1b. The results displayed there confirm those in Figure 1a, indicating that solving (10) is the same as solving (2). The solutions produced by the model are shown as trajectories in Figure 1c–f. These figures display the real part and the three imaginary parts of the solutions, respectively. It is demonstrated that the models provide the same solutions and that their convergence to the TSOL is consistent with the related EMEs' tendency towards convergence.

Furthermore, it can be seen that $\tilde{A}(t), \tilde{B}(t), \tilde{C}(t) \in \mathbb{H}^{3 \times 3}$ in Section 5.1.2. In other words, $n = 3$ in (2)'s TQARE. Under IC1 and IC2 for $\lambda = 10$, we have the following results for the ZQ-ARE model. Figure 2a illustrates the ZQ-ARE model's EMEs. The two cases in this illustration begin with a big error value at $t = 0$: with IC2's initial error value being substantially larger than IC1's. However, the EMEs reach a tiny error value in the interval $[10^{-4}, 10^{-3}]$ at $t = 1$ in the instance of IC1 and at $t = 2$ in the instance of IC2. Stated differently, Theorem 2 is validated by the ZQ-ARE model, which converges to a low value for two distinctive ICs. The ZQ-ARE model's (2) error is displayed in Figure 2b. The results displayed there confirm those in Figure 2a, indicating that solving (10) is the same as solving (2). The solutions produced by the model are shown as trajectories in Figure 2c–f. These figures display the real part and the three imaginary parts of the solutions, respectively. It is demonstrated that the models provide the same solutions and that their convergence to the TSOL is consistent with the related EMEs' tendency towards convergence.

Additionally, it can be seen that $\tilde{A}(t), \tilde{B}(t), \tilde{C}(t) \in \mathbb{H}^{6 \times 6}$ in Section 5.1.3. In other words, $n = 6$ in (2)'s TQARE. Under an initial condition that is near to the eigenvector TSOL and for $\lambda = 10$, we have the following results for the ZQ-ARE model. Figure 3a illustrates the ZQ-ARE model's EME. The EME in this illustration begins with a big error value at $t = 0$ and reaches a tiny error value in the interval $[10^{-4}, 10^{-2}]$ at $t = 1$. The ZQ-ARE model's (2) error is displayed in Figure 3b. The results displayed there confirm those in Figure 3a, indicating that solving (10) is the same as solving (2). The solutions produced by the model are shown as trajectories in Figure 3c–f. These figures display the real part and the three imaginary parts, respectively, of the ZQ-ARE model's solution and the eigenvector TSOL. It is demonstrated that the model provides the same solution and that its convergence to the eigenvector TSOL is consistent with the related EME's tendency towards convergence. Stated differently, Theorem 2 is validated by the ZQ-ARE model, which converges to the TSOL.

When everything is considered, the ZQ-ARE model solves three distinct TQAREs quite well. It is noteworthy that the discussion given above confirms the ZNN technique's convergence qualities and Theorem 2's findings. Additionally, the ZQ-ARE model's computational complexity is $\mathcal{O}((2n^2 - n)^3)$, which is similar to that of other ZNN models that approach various quaternion matrix equations (see [25,28,30–32]).

5.3. Application to Quadrotor Control

In this application, the ZNN design technique is used to stabilize the quadcopter device of Figure 4. Determining the position of a quadcopter requires the definition of coordinate systems since it is a vehicle with four separate drives and, at its center of gravity, an electric power system [49,50]. Three Euler angles and the vertical movement in the global coordinate system are the control parameters for a six-degree-of-freedom model device [51,52]. In order to ensure independent control over each drive and minimal aerodynamic effects, it is necessary to consider a fixed-frame body and a symmetrical model structure with the origin in the mass center [53,54].

Our approach uses the dynamic model derived from [55,56], which consists of the following equation system:

$$\begin{cases} \ddot{x} = \frac{u_1}{m} (\eta(\phi)\gamma(\theta)\eta(\psi) + \gamma(\phi)\gamma(\psi)) \\ \ddot{y} = \frac{u_1}{m} (\eta(\phi)\gamma(\theta)\gamma(\psi) - \gamma(\phi)\eta(\psi)) \\ \ddot{z} = \frac{u_1}{m} \eta(\phi)\eta(\psi) - g \\ \ddot{\phi} = \frac{I_y - I_z}{I_x} \dot{\theta}\dot{\psi} - \frac{J_R}{I_x} \dot{\theta}\omega_d + \frac{l}{I_x} u_2 \\ \ddot{\theta} = \frac{I_z - I_x}{I_y} \dot{\phi}\dot{\psi} - \frac{J_R}{I_y} \dot{\phi}\omega_d + \frac{l}{I_y} u_3 \\ \ddot{\psi} = \frac{I_x - I_y}{I_z} \dot{\phi}\dot{\theta} + \frac{l}{I_z} u_4, \end{cases} \quad (29)$$

where the position of the quadrotor is described by the coordinates x, y, z in the global coordinate system, and its orientation is described by the three Euler angles: i.e., the yaw angle $\psi \in [-\pi, \pi]$, pitch angle $\theta \in [-\pi/2, \pi/2]$, and roll angle $\phi \in [-\pi, \pi]$. Additionally, g is the gravitational acceleration, l is the quadrotor's arm length, d is the factor of drag, ω_d is the relative motor speed, I_x, I_y, I_z are inertias, J_R is the rotor inertia, m is the quadrotor's mass, and $u_j, j = 1, \dots, 4$ are the input variables of the system.

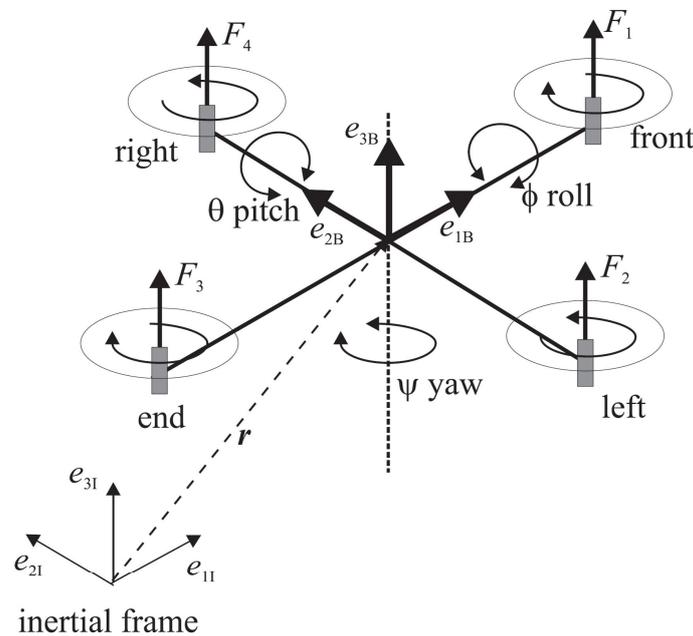


Figure 4. Configuration, inertia, and body-fixed frame of the quadrotor.

The state-space formation of (29) is the following:

$$\dot{\mathbf{x}} = \begin{bmatrix} x_2 \\ \frac{u_1}{m} (\eta(x_7)\gamma(x_9)\eta(x_{11}) + \gamma(x_7)\gamma(x_{11})) \\ x_4 \\ \frac{u_1}{m} (\eta(x_7)\gamma(x_9)\gamma(x_{11}) + \gamma(x_7)\eta(x_{11})) \\ x_6 \\ \frac{u_1}{m} \eta(x_7)\eta(x_9) - g \\ x_8 \\ \frac{I_y - I_z}{I_x} x_{12} x_{10} - \frac{J_R}{I_x} x_{10} \omega_d + \frac{l}{I_x} u_2 \\ x_{10} \\ \frac{I_z - I_x}{I_y} x_{12} x_8 - \frac{J_R}{I_y} x_8 \omega_d + \frac{l}{I_y} u_3 \\ x_{12} \\ \frac{I_x - I_y}{I_z} x_{10} x_8 + \frac{l}{I_z} u_4, \end{bmatrix} \quad (30)$$

where $\mathbf{x} = [x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T = [x_1, x_2, \dots, x_{12}]^T \in \mathbb{R}^{12}$ is the state vector and $u = [u_1, u_2, u_3, u_4]^T \in \mathbb{R}^4$ is the control vector.

In order to solve an ARE and derive the suboptimal control rule, it is crucial to note that the SDRE control approach reconstructs the nonlinear dynamics utilizing parameterization into a linear form with coefficient state-dependent matrices [57]. In the state space, the ARE equation’s coefficients vary. Therefore, it is necessary to solve a state-dependent ARE at any point in the state space. We are able to get more degrees of freedom because the state-dependent parameterization is nonunique, which can potentially improve controller performance. The structure is similar to that of a quadratic when the nonlinear performance index is minimized.

Consider the optimal control issue for the next linear time-varying system [58]:

$$\dot{\mathbf{x}}(t) = A(t)\mathbf{x}(t) + L(t)u(t), \tag{31}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$, $L(t) \in \mathbb{R}^{n \times m}$, $A(t) \in \mathbb{R}^{n \times n}$, and $u(t) \in \mathbb{R}^m$. Control gains at any state $\mathbf{x}(t)$ can be calculated using standard linear optimal control theory, i.e., choosing the control that minimizes the following cost function:

$$J = 0.5 \int_0^\infty \mathbf{x}^T(t)C(t)\mathbf{x}(t) + u^T(t)G(t)u(t)dt, \tag{32}$$

where $C(t) \in \mathbb{R}^{n \times n}$ penalizes the state and $G(t) \in \mathbb{R}^{m \times m}$ penalizes the control effort. By obtaining $X(t)$, which is the solution to the next continuous-time ARE (CARE):

$$X(t)A(t) + A^T(t)X(t) - X(t)L(t)G^{-1}(t)L^T(t)X(t) + C(t) = \mathbf{0}_{n \times n}, \tag{33}$$

the control gains that minimizes (32) become:

$$u(t) = -G^{-1}(t)L^T(t)X(t)\mathbf{x}(t). \tag{34}$$

This controller’s stability features are verified in [59,60].

One traditional approach for solving (33) is to use the forward-propagating Riccati equation (FPRE) method [61]. Particularly, the solution of (33) is obtained through the following DARE:

$$\dot{X}(t) = X(t)A(t) + A^T(t)X(t) - X(t)L(t)G^{-1}(t)L^T(t)X(t) + C(t). \tag{35}$$

On the other hand, one different approach for solving (33) is to use the ZNN method as proposed in [62]. Particularly, by using $B(t) = L(t)G^{-1}(t)L^T(t)$, it is readily understood that (33) may be turned into (2). As a result, the solution of (33) can be obtained through (20).

In our approach, the SDRE method is applied to the attitude control problem. The vector of the state variables for that problem is $\mathbf{x}_{\mathcal{I}} = [x_7, x_8, \dots, x_{12}]^T \in \mathbb{R}^6$, and the vector of the input variables is $u_{\mathcal{I}} = [u_2, u_3, u_4]^T \in \mathbb{R}^3$, while the state-dependent model derived from (30) after factorization is the following:

$$\dot{\mathbf{x}}_{\mathcal{I}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{I_y - I_z}{I_x} x_{12} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{I_z - I_x}{I_y} x_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \frac{I_x - I_y}{I_z} x_8 & 0 & 0 \end{bmatrix} \cdot \mathbf{x}_{\mathcal{I}} + \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix} \cdot u_{\mathcal{I}}, \tag{36}$$

where $A(t) \in \mathbb{R}^{6 \times 6}$ and $L(t) \in \mathbb{R}^{6 \times 3}$. Thereafter, we set $\tilde{A}(t) = A(t)$, $\tilde{C}(t) = G(t) = I_6$, and $\tilde{B}(t) = L(t)G^{-1}(t)L^T(t)$, and we use the parameter values of the object presented in

Table 1 and the ICs $\mathbf{x}_T(0) = [\pi/4, \pi/2, 3\pi/4, 0, 0, 0]^T$. Additionally, the λ parameter value is set to 100, and the ICs of the ZQ-ARE model and DARE have been set as follows:

$$\tilde{X}(0) = X(0) = \begin{bmatrix} \mathbf{1}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \mathbf{1}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \mathbf{1}_{2 \times 2} \end{bmatrix}. \tag{37}$$

The results of the ZNN and FPRE methods are presented in Figure 5. Notice that the purpose of the control procedure is stabilization at the zero point.

Particularly, it can be seen that $\tilde{A}(t), \tilde{B}(t), \tilde{C}(t) \in \mathbb{H}^{6 \times 6}$. In other words, $n = 6$ in (2)'s TQARE. For $\lambda = 100$, we have the following results for the ZQ-ARE model. Figure 5a illustrates the ZQ-ARE model's EMEs. The EME in this image starts with a big value at $t = 0$ and ends up with a small value in the interval $[10^{-16}, 10^{-15}]$ at $t = 5$. Figure 5b depicts the (2) error of the ZQ-ARE solution of the ZNN method and the DARE solution of the FPRE method. The findings show that the ZQ-ARE solution is more accurate than the DARE solution. Figure 5c displays the trajectories of the real parts of the solutions generated by the models; the solutions' imaginary parts are left out because they are zeros. This graph shows that although the models produce identical solutions, the ZNN approach converges more quickly than the FPRE method. Figure 5d shows the time-plot of the quadrotor's angles, and Figure 5e shows the time-plot of the quadrotor's velocities. These graphs demonstrate that stabilization at the "zero" point for the angles and velocities is successful by both models, but the ZNN method achieve faster stabilization than the FPRE method. Figure 5f shows the quadrotor's position during attitude control, where we can observe that the ZNN and FPRE methods produce identical results. Based on the aforementioned observations, the results of the application to quadrotor control show that the ZNN method is more effective than the FPRE method. To conclude, the ZNN method is not only fruitful for solving the TQARE, but it may also be employed to quadrotor control with high efficiency.

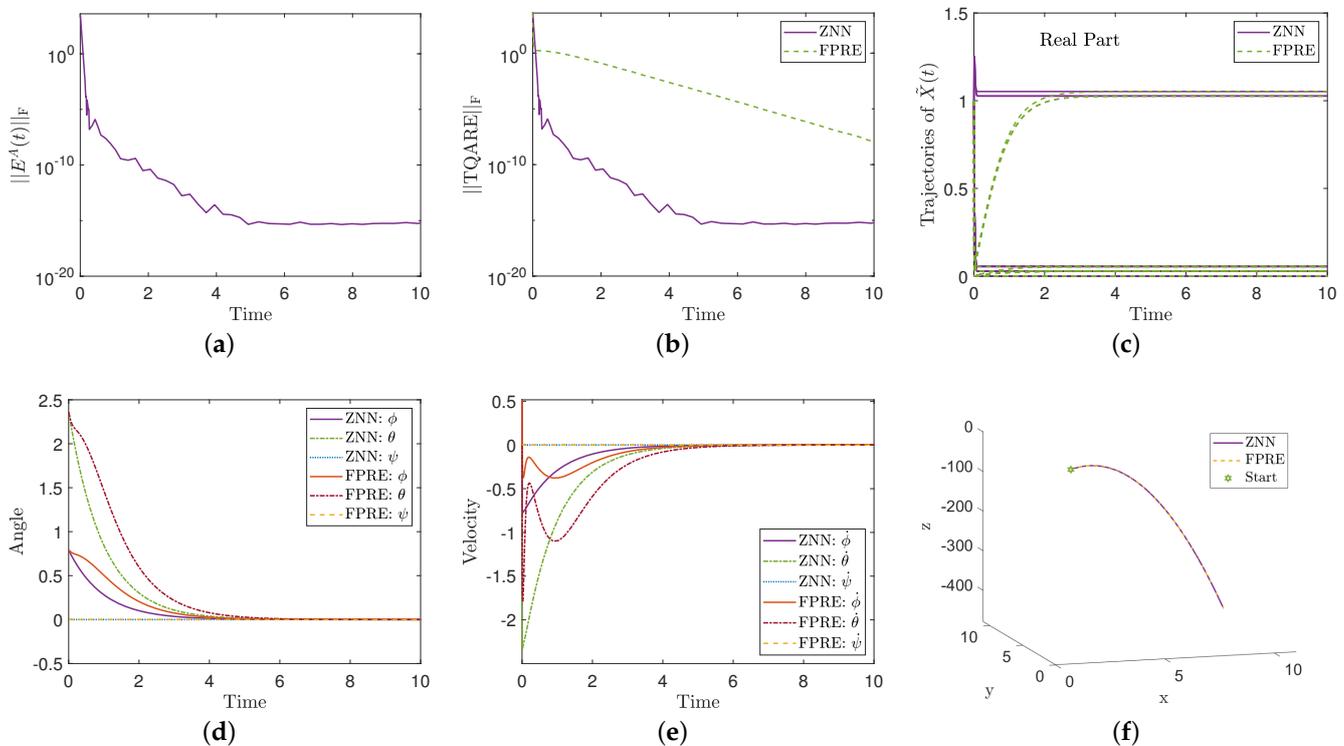


Figure 5. EME, solution trajectories, Equation (2) errors, time-plot of angles, and velocities, and plot of the position of the quadrotor during attitude control: (a) EME convergence, (b) Equation (2) error, (c) solution trajectories, (d) time-plot of angles, (e) time-plot of velocities, and (f) plot of position during attitude control.

Table 1. The parameters utilized in the quadrotor simulation.

Parameter	Value	Unit
m	0.5	kg
g	9.81	m/s ²
l	0.3	m
J_R	0.01	N
\mathcal{I}_x	0.0081	kg·m ²
\mathcal{I}_y	0.0081	kg·m ²
\mathcal{I}_z	0.0162	kg·m ²

6. Conclusions

In this paper, a novel ZNN model, termed 'ZQ-ARE', is presented for solving the TQARE by identifying only Hermitian solutions. Theoretical investigation and simulation experiments are conducted to support the ZQ-ARE model's accuracy and efficiency, while an application to quadrotor control is carried out to support the ZQ-ARE model's practical applicability. Specifically, in three SEs, the ZQ-ARE model finds the TQARE Hermitian solutions under various initial conditions, and we also demonstrate that the convergence rate of the solution increases as the value of the parameter λ rises. For the purpose of stabilizing the flight control system of a quadrotor, a comparison is given between the solution adaptations provided by the ZNN and FPRES methods. The results of this application show that the ZNN method is more effective than the FPRES method. So we can conclude that the ZNN method is not only fruitful for solving the TQARE, but it may also be employed for quadrotor control with high efficiency.

It is significant to mention that the suggested ZNN model has the drawback of being noise-intolerant since all kinds of noise significantly affect the suggested ZNN approach's accuracy. Therefore, the adaptation of this model to a noise-tolerant ZNN design could be the main focus of future study. To be more precise, the suggested model might be made noise-tolerant by substituting the original ZNN design with a noise-tolerant ZNN architecture, such as the one used in [37]. Moreover, future research may involve applying the proposed ZNN model to a range of other technical issues, including secure communications with application to acoustic source tracking [63] and network and power systems with application to chaotic system stabilization [64,65].

Author Contributions: Conceptualization, H.J., O.A., S.B.A., M.K., T.E.S., S.D.M. and V.N.K.; methodology, M.K., T.E.S., S.D.M. and V.N.K.; software, H.J., O.A., S.B.A., M.K., T.E.S., S.D.M. and V.N.K.; validation, H.J., O.A., S.B.A., M.K., T.E.S., S.D.M. and V.N.K.; formal analysis, T.E.S. and V.N.K.; Investigation, T.E.S., S.D.M. and V.N.K.; resources, H.J., O.A., S.B.A., M.K., T.E.S., S.D.M. and V.N.K.; data curation, H.J., O.A., S.B.A., M.K., T.E.S., S.D.M. and V.N.K.; writing—original draft, S.D.M. and V.N.K.; writing—review & editing, T.E.S.; visualization, H.J., O.A., S.B.A., M.K., T.E.S., S.D.M. and V.N.K.; supervision, T.E.S.; project administration, H.J.; funding acquisition, H.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Research Deanship of Ha'il University, Kingdom of Saudi Arabia, project number (RG-23 072).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The authors acknowledge the Research Deanship of Ha'il University, Kingdom of Saudi Arabia, for administrative, financial, and technical support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kalman, R.E. Contributions to the theory of optimal control. *Bol. Soc. Mat. Mex.* **1960**, *5*, 102–119.
2. Qin, B.; Sun, H.; Ma, J.; Li, W.; Ding, T.; Wang, Z.; Zomaya, A.Y. Robust H_∞ control of doubly fed wind generator via state-dependent Riccati equation technique. *IEEE Trans. Power Syst.* **2019**, *34*, 2390–2400. [[CrossRef](#)]
3. Dong, X.; Hu, G. Time-varying formation tracking for linear multiagent systems with multiple leaders. *IEEE Trans. Autom. Control* **2017**, *62*, 3658–3664. [[CrossRef](#)]
4. Rigatos, G.; Busawon, K.; Pomares, J.; Abbaszadeh, M. Nonlinear optimal control for the wheeled inverted pendulum system. *Robotica* **2020**, *38*, 29–47. [[CrossRef](#)]
5. Laub, A. A Schur method for solving algebraic Riccati equations. *IEEE Trans. Autom. Control.* **1979**, *24*, 913–921. [[CrossRef](#)]
6. Aguilar, L.T.; Orlov, Y.; Aho, L. Nonlinear H_∞ -control of nonsmooth time-varying systems with application to friction mechanical manipulators. *Automatica* **2003**, *39*, 1531–1542. [[CrossRef](#)]
7. Mohammadi, H.; Zare, A.; Soltanolkotabi, M.; Jovanović, M.R. Convergence and sample complexity of gradient methods for the model-free linear-quadratic regulator problem. *IEEE Trans. Autom. Control.* **2022**, *67*, 2435–2450. [[CrossRef](#)]
8. Hambly, B.; Xu, R.; Yang, H. Policy gradient methods for the noisy linear quadratic regulator over a finite horizon. *SIAM J. Control Optim.* **2021**, *59*, 3359–3391. [[CrossRef](#)]
9. Hossain, M.; Haque, M.E.; Arif, M.T. Kalman filtering techniques for the online model parameters and state of charge estimation of the Li-ion batteries: A comparative analysis. *J. Energy Storage* **2022**, *51*, 104174. [[CrossRef](#)]
10. Hamidi, F.; Jerbi, H.; Alharbi, H.; Leiva, V.; Popescu, D.; Rajhi, W. Metaheuristic solution for stability analysis of nonlinear systems using an intelligent algorithm with potential applications. *Fractal Fract.* **2023**, *7*, 78. [[CrossRef](#)]
11. Kchaou, M.; Regaieg, M.A.; Jerbi, H.; Abbassi, R.; Stefanou, D.; Popescu, D. Admissible control for non-linear singular systems subject to time-varying delay and actuator saturation: An interval type-2 fuzzy approach. *Actuators* **2023**, *12*, 30. [[CrossRef](#)]
12. Oshman, Y.; Bar-Itzhack, I. Eigenfactor solution of the matrix Riccati equation—A continuous square root algorithm. *IEEE Trans. Autom. Control.* **1985**, *30*, 971–978. [[CrossRef](#)]
13. Dooren, P.V. A generalized eigenvalue approach for solving Riccati equations. *SIAM J. Sci. Stat. Comput.* **1981**, *2*, 121–135. [[CrossRef](#)]
14. Zhang, F. Quaternions and matrices of quaternions. *Linear Algebra Its Appl.* **1997**, *251*, 21–57. [[CrossRef](#)]
15. Rodman, L. *Topics in Quaternion Linear Algebra*; Princeton Series in Applied Mathematics; Princeton University Press: Princeton, NJ, USA, 2014. [[CrossRef](#)]
16. Hamilton, W.R. On a new species of imaginary quantities, connected with the theory of quaternions. *Proc. R. Ir. Acad.* **1840**, *2*, 424–434.
17. Joldeş, M.; Muller, J.M. Algorithms for manipulating quaternions in floating-point arithmetic. In Proceedings of the 2020 IEEE 27th Symposium on Computer Arithmetic (ARITH), Portland, OR, USA, 7–10 June 2020; pp. 48–55.
18. Szynal-Liana, A.; Włoch, I. Generalized commutative quaternions of the Fibonacci type. *Bol. Soc. Mat. Mex.* **2022**, *28*, 1. [[CrossRef](#)]
19. Pavillo, D.; Feichtenhofer, C.; Auli, M.; Grangier, D. Modeling human motion with quaternion-based neural networks. *Int. J. Comput. Vis.* **2020**, *128*, 855–872. [[CrossRef](#)]
20. Giardino, S. Quaternionic quantum mechanics in real Hilbert space. *J. Geom. Phys.* **2020**, *158*, 103956. [[CrossRef](#)]
21. Goodyear, A.M.S.; Singla, P.; Spencer, D.B. Analytical state transition matrix for dual-quaternions for spacecraft pose estimation. In Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, Maui, HI, USA, 13–17 January 2019; Univelt Inc.: Escondido, CA, USA, 2020; pp. 393–411.
22. Kansu, M.E. Quaternionic representation of electromagnetism for material media. *Int. J. Geom. Methods Mod. Phys.* **2019**, *16*, 1950105. [[CrossRef](#)]
23. Özgür, E.; Mezouar, Y. Kinematic modeling and control of a robot arm using unit dual quaternions. *Robot. Auton. Syst.* **2016**, *77*, 66–73. [[CrossRef](#)]
24. Xiao, L.; Zhang, Y.; Huang, W.; Jia, L.; Gao, X. A dynamic parameter noise-tolerant zeroing neural network for time-varying quaternion matrix equation with applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–10. [[CrossRef](#)]
25. Kovalnogov, V.N.; Fedorov, R.V.; Demidov, D.A.; Malyoshina, M.A.; Simos, T.E.; Mourtas, S.D.; Katsikis, V.N. Computing quaternion matrix pseudoinverse with zeroing neural networks. *AIMS Math.* **2023**, *8*, 22875–22895. [[CrossRef](#)]
26. Xiao, L.; Cao, P.; Song, W.; Luo, L.; Tang, W. A fixed-time noise-tolerance ZNN model for time-variant inequality-constrained quaternion matrix least-squares problem. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–10. [[CrossRef](#)]
27. Xiao, L.; Liu, S.; Wang, X.; He, Y.; Jia, L.; Xu, Y. Zeroing neural networks for dynamic quaternion-valued matrix inversion. *IEEE Trans. Ind. Inform.* **2022**, *18*, 1562–1571. [[CrossRef](#)]
28. Aoun, S.B.; Derbel, N.; Jerbi, H.; Simos, T.E.; Mourtas, S.D.; Katsikis, V.N. A quaternion Sylvester equation solver through noise-resilient zeroing neural networks with application to control the SFM chaotic system. *AIMS Math.* **2023**, *8*, 27376–27395. [[CrossRef](#)]
29. Tan, N.; Yu, P.; Ni, F. New varying-parameter recursive neural networks for model-free kinematic control of redundant manipulators with limited measurements. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–14. [[CrossRef](#)]
30. Abbassi, R.; Jerbi, H.; Kchaou, M.; Simos, T.E.; Mourtas, S.D.; Katsikis, V.N. Towards higher-order zeroing neural networks for calculating quaternion matrix inverse with application to robotic motion tracking. *Mathematics* **2023**, *11*, 2756. [[CrossRef](#)]

31. Kovalnogov, V.N.; Fedorov, R.V.; Demidov, D.A.; Malyoshina, M.A.; Simos, T.E.; Katsikis, V.N.; Mourtas, S.D.; Sahas, R.D. Zeroing neural networks for computing quaternion linear matrix equation with application to color restoration of images. *AIMS Math.* **2023**, *8*, 14321–14339. [[CrossRef](#)]
32. Kovalnogov, V.N.; Fedorov, R.V.; Shepelev, I.I.; Sherkunov, V.V.; Simos, T.E.; Mourtas, S.D.; Katsikis, V.N. A novel quaternion linear matrix equation solver through zeroing neural networks with applications to acoustic source tracking. *AIMS Math.* **2023**, *8*, 25966–25989. [[CrossRef](#)]
33. Zhang, Y.; Ge, S.S. Design and analysis of a general recurrent neural network model for time-varying matrix inversion. *IEEE Trans. Neural Netw.* **2005**, *16*, 1477–1490. [[CrossRef](#)]
34. Chai, Y.; Li, H.; Qiao, D.; Qin, S.; Feng, J. A neural network for Moore-Penrose inverse of time-varying complex-valued matrices. *Int. J. Comput. Intell. Syst.* **2020**, *13*, 663–671. [[CrossRef](#)]
35. Wu, W.; Zheng, B. Improved recurrent neural networks for solving Moore-Penrose inverse of real-time full-rank matrix. *Neurocomputing* **2020**, *418*, 221–231. [[CrossRef](#)]
36. Jiang, W.; Lin, C.L.; Katsikis, V.N.; Mourtas, S.D.; Stanimirović, P.S.; Simos, T.E. Zeroing neural network approaches based on direct and indirect methods for solving the Yang–Baxter-like matrix equation. *Mathematics* **2022**, *10*, 1950. [[CrossRef](#)]
37. Jerbi, H.; Alharbi, H.; Omri, M.; Ladhar, L.; Simos, T.E.; Mourtas, S.D.; Katsikis, V.N. Towards higher-order zeroing neural network dynamics for solving time-varying algebraic Riccati equations. *Mathematics* **2022**, *10*, 4490. [[CrossRef](#)]
38. Katsikis, V.N.; Stanimirović, P.S.; Mourtas, S.D.; Xiao, L.; Karabasević, D.; Stanujkić, D. Zeroing neural network with fuzzy parameter for computing pseudoinverse of arbitrary matrix. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 3426–3435. [[CrossRef](#)]
39. Alharbi, H.; Jerbi, H.; Kchaou, M.; Abbassi, R.; Simos, T.E.; Mourtas, S.D.; Katsikis, V.N. Time-varying pseudoinversion based on full-rank decomposition and zeroing neural networks. *Mathematics* **2023**, *11*, 600. [[CrossRef](#)]
40. Mourtas, S.D.; Katsikis, V.N. Exploiting the Black-Litterman framework through error-correction neural networks. *Neurocomputing* **2022**, *498*, 43–58. [[CrossRef](#)]
41. Kovalnogov, V.N.; Fedorov, R.V.; Generalov, D.A.; Chukalin, A.V.; Katsikis, V.N.; Mourtas, S.D.; Simos, T.E. Portfolio insurance through error-correction neural networks. *Mathematics* **2022**, *10*, 3335. [[CrossRef](#)]
42. Mourtas, S.D.; Kasimis, C. Exploiting mean-variance portfolio optimization problems through zeroing neural networks. *Mathematics* **2022**, *10*, 3079. [[CrossRef](#)]
43. Qiao, S.; Wei, Y.; Zhang, X. Computing time-varying ML-weighted pseudoinverse by the Zhang neural networks. *Numer. Funct. Anal. Optim.* **2020**, *41*, 1672–1693. [[CrossRef](#)]
44. Wang, X.; Stanimirovic, P.S.; Wei, Y. Complex ZFs for computing time-varying complex outer inverses. *Neurocomputing* **2018**, *275*, 983–1001. [[CrossRef](#)]
45. Dai, J.; Tan, P.; Yang, X.; Xiao, L.; Jia, L.; He, Y. A fuzzy adaptive zeroing neural network with superior finite-time convergence for solving time-variant linear matrix equations. *Knowl.-Based Syst.* **2022**, *242*, 108405. [[CrossRef](#)]
46. Groß, J.; Trenkler, G.; Troschke, S.O. Quaternions: Further contributions to a matrix oriented approach. *Linear Algebra Its Appl.* **2001**, *326*, 205–213. [[CrossRef](#)]
47. Gupta, A.K. *Numerical Methods Using MATLAB*; MATLAB Solutions Series; Apress: Berkeley, CA, USA; New York, NY, USA, 2014.
48. Simos, T.E.; Katsikis, V.N.; Mourtas, S.D.; Stanimirović, P.S. Unique non-negative definite solution of the time-varying algebraic Riccati equations with applications to stabilization of LTV systems. *Math. Comput. Simul.* **2022**, *202*, 164–180. [[CrossRef](#)]
49. Carino, J.; Abaunza, H.; Castillo, P. Quadrotor quaternion control. In Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015; IEEE: Denver, CO, USA, 2015. [[CrossRef](#)]
50. Fresk, E.; Nikolakopoulos, G. Experimental evaluation of a full quaternion based attitude quadrotor controller. In Proceedings of the 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), Luxembourg, 8–11 September 2015; IEEE: Luxembourg, 2015. [[CrossRef](#)]
51. Chen, Y.; Perez-Arancibia, N.O. Adaptive control of aerobatic quadrotor maneuvers in the presence of propeller-aerodynamic-coefficient and torque-latency time-variations. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; IEEE: Montreal, QC, Canada, 2019. [[CrossRef](#)]
52. Nekoo, S.R.; Acosta, J.Á.; Ollero, A. Quaternion-based state-dependent differential Riccati equation for quadrotor drones: Regulation control problem in aerobatic flight. *Robotica* **2022**, *40*, 3120–3135. [[CrossRef](#)]
53. Liang, T.; Yang, K.; Han, Q.; Li, C.; Li, J.; Deng, Q.; Chen, S.; Tuo, X. Attitude estimation of quadrotor UAV based on QUKF. *IEEE Access* **2023**, *11*, 111133–111141. [[CrossRef](#)]
54. Li, J.; Chen, P.; Chang, Z.; Zhang, G.; Guo, L.; Zhao, C. Trajectory tracking control of quadrotor based on fractional-order s-plane model. *Machines* **2023**, *11*, 672. [[CrossRef](#)]
55. Stepień, S.; Superczyńska, P. Modified infinite-time state-dependent Riccati equation method for nonlinear affine systems: Quadrotor control. *Appl. Sci.* **2021**, *11*, 10714. [[CrossRef](#)]
56. Voos, H. Nonlinear state-dependent Riccati equation control of a quadrotor UAV. In Proceedings of the 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, Munich, Germany, 4–6 October 2006; IEEE: Munich, Germany, 2006. [[CrossRef](#)]
57. Tan, S.; Zhong, W. Numerical solutions of linear quadratic control for time-varying systems via symplectic conservative perturbation. *Appl. Math. Mech.* **2007**, *28*, 277–287. [[CrossRef](#)]

58. Weiss, A.; Kolmanovsky, I.; Bernstein, D.S. Forward-integration Riccati-based output-feedback control of linear time-varying systems. In Proceedings of the 2012 American Control Conference (ACC), Montreal, QC, Canada, 27–29 June 2012; pp. 6708–6714.
59. Mracek, C.P.; Cloutier, J.R. Control designs for the nonlinear benchmark problem via the state-dependent Riccati equation method. *Int. J. Robust Nonlinear Control* **1998**, *8*, 401–433. [[CrossRef](#)]
60. Erdem, E.B.; Alleyne, A.G. Design of a class of nonlinear controllers via state dependent Riccati equations. *IEEE Trans. Control. Syst. Technol.* **2004**, *12*, 133–137. [[CrossRef](#)]
61. Prach, A.; Tekinalp, O.; Bernstein, D.S. A numerical comparison of frozen-time and forward-propagating Riccati equations for stabilization of periodically time-varying systems. In Proceedings of the American Control Conference, Portland, OR, USA, 4–6 June 2014; pp. 5633–5638.
62. Simos, T.E.; Katsikis, V.N.; Mourtas, S.D.; Stanimirović, P.S. Solving time-varying nonsymmetric algebraic Riccati equations with zeroing neural dynamics. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 6575–6587. [[CrossRef](#)]
63. He, W.; Luo, T.; Tang, Y.; Du, W.; Tian, Y.; Qian, F. Secure communication based on quantized synchronization of chaotic neural networks under an event-triggered strategy. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 3334–3345. [[CrossRef](#)] [[PubMed](#)]
64. Su, H.; Luo, R.; Huang, M.; Fu, J. Robust fixed time control of a class of chaotic systems with bounded uncertainties and disturbances. *Int. J. Control Autom. Syst.* **2022**, *20*, 813–822. [[CrossRef](#)]
65. Singer, J.; Wang, Y.; Bau, H.H. Controlling a chaotic system. *Phys. Rev. Lett.* **1991**, *66*, 1123. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.