

Article

Towards Higher-Order Zeroing Neural Networks for Calculating Quaternion Matrix Inverse with Application to Robotic Motion Tracking

Rabeh Abbassi ¹, Housseem Jerbi ², Mourad Kchaou ¹, Theodore E. Simos ^{3,4,5,6,7,*}, Spyridon D. Mourtas ^{8,9} and Vasilios N. Katsikis ⁸

- ¹ Department of Electrical Engineering, College of Engineering, University of Hail, Hail 81451, Saudi Arabia; r.abbassi@uoh.edu.sa (R.A.); m.kchaou@uoh.edu.sa (M.K.)
- ² Department of Industrial Engineering, College of Engineering, University of Hail, Hail 81451, Saudi Arabia; h.jerbi@uoh.edu.sa
- ³ Center for Applied Mathematics and Bioinformatics, Gulf University for Science and Technology, West Mishref 32093, Kuwait
- ⁴ Laboratory of Inter-Disciplinary 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 City 40402, Taiwan
- ⁶ Data Recovery Key Laboratory of Sichun Province, Neijiang Normal University, Neijiang 641100, China
- ⁷ Section of Mathematics, Department of Civil Engineering, Democritus University of Thrace, 67100 Xanthi, Greece
- ⁸ Department of Economics, Mathematics-Informatics and Statistics-Econometrics, National and Kapodistrian University of Athens, Sofokleous 1 Street, 10559 Athens, Greece; spirmour@econ.uoa.gr (S.D.M.); vaskatsikis@econ.uoa.gr (V.N.K.)
- ⁹ Laboratory "Hybrid Methods of Modelling and Optimization in Complex Systems", Siberian Federal University, Prosp. Svobodny 79, 660041 Krasnoyarsk, Russia
- * Correspondence: simos@ulstu.ru



Citation: 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. <https://doi.org/10.3390/math11122756>

Academic Editor: Ioannis K. Argyros

Received: 16 May 2023

Revised: 13 June 2023

Accepted: 15 June 2023

Published: 18 June 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 efficient solution of the time-varying quaternion matrix inverse (TVQ-INV) is a challenging but crucial topic due to the significance of quaternions in many disciplines, including physics, engineering, and computer science. The main goal of this research is to employ the higher-order zeroing neural network (HZNN) strategy to address the TVQ-INV problem. HZNN is a family of zeroing neural network models that correlates to the hyperpower family of iterative methods with adjustable convergence order. Particularly, three novel HZNN models are created in order to solve the TVQ-INV both directly in the quaternion domain and indirectly in the complex and real domains. The noise-handling version of these models is also presented, and the performance of these models under various types of noises is theoretically and numerically tested. The effectiveness and practicality of these models are further supported by their use in robotic motion tracking. According to the principal results, each of these six models can solve the TVQ-INV effectively, and the HZNN strategy offers a faster convergence rate than the conventional zeroing neural network strategy.

Keywords: matrix inverse; quaternion; dynamical system; hyperpower iterations; zeroing neural network; robotic motion tracking

MSC: 65F20; 68T05

1. Introduction

The real-time solution to the matrix inverse [1,2], which frequently arises in robotics [3], game theory [4], nonlinear systems [5], optimal control [6,7], and neural networks [8], has attracted a lot of interest in recent times. Quaternions, on the other hand, are crucial in a wide range of domains, such as computer graphics [9], signal processing [10], human

motion modeling [11], robotics [12,13], navigation [14], quantum mechanics [15], electromagnetism [16], and mathematical physics [17,18]. Let $\mathbb{H}^{n \times n}$ present the set of all $n \times n$ matrices on the quaternion skew field $\mathbb{H} = \{\gamma_1 + \gamma_2i + \gamma_3j + \gamma_4k \mid i^2 = j^2 = k^2 = ijk = -1, \gamma_1, \gamma_2, \gamma_3, \gamma_4 \in \mathbb{R}\}$. Considering that $\tilde{A} \in \mathbb{H}^{n \times n}$, its inverse matrix is denoted by \tilde{A}^{-1} and it is the only solution \tilde{X} that satisfies the next equation [19,20]:

$$\tilde{A}\tilde{X} = I_n, \tag{1}$$

where I_n is the identity $n \times n$ matrix.

Recently, research has begun to focus on time-varying quaternion (TVQ) problems involving matrices, such as the inversion of TVQ matrices [21], solving the dynamic TVQ Sylvester matrix equation [22], addressing the TVQ constrained matrix least-squares problem [23], and solving the TVQ linear matrix equation for square matrices [24]. Furthermore, real-world applications involving TVQ matrices are employed in the kinematically redundant manipulator of robotic joints [25,26], such as the control of wearable robotic knee system [27] and control of robotic arm [13], chaotic systems synchronization [23], mobile manipulator control [21], and image restoration [24]. All of these studies have one thing in common: they all use the zeroing neural network (ZNN) approach to derive the solution.

ZNNs are a subset of recurrent neural networks that are especially good at parallel processing and are used to address time-varying issues. They were initially developed by Zhang et al. to handle the problem of time-varying matrix inversion [28], but their subsequent iterations were dynamic models used to compute the time-varying MP-inverse of full-row/column rank matrices [29–32] in the real and complex domain. Today their use has expanded to include the resolution of generalized inversion issues, including time-varying Drazin inverse [33], time-varying ML-weighted pseudoinverse [34], time-varying outer inverse [35], time-varying pseudoinverse [36], and core and core-EP inverse [37]. Their use has expanded to include the resolution of linear programming tasks [38], quadratic programming tasks [39,40], systems of nonlinear equations [41,42], and systems of linear equations [43,44]. The creation of a ZNN model typically involves two fundamental steps. First, one defines an error matrix equation (EME) function $E(t)$. Second, the next ZNN dynamical system (under the linear activation) function must be used:

$$\dot{E}(t) = -\lambda E(t), \tag{2}$$

where the operator $(\dot{\cdot})$ denotes the time derivative. Additionally, the design parameter $\lambda > 0$ is a real number that regulates the model’s convergence speed. For instance, a greater value for λ will increase the model’s convergence speed [45–47]. It is important to point out that continual learning is defined as learning continually from non-stationary data while simultaneously transferring and preserving prior knowledge. It is true that as time evolves, the architecture of ZNN relies on driving each element of the error function $E(t)$ to zero. The continuous-time learning rule, which is the consequence of the definition of the EME function (2), is used to do this. Therefore, it is possible to think of the error function as a tool for tracking the learning of ZNN models.

1.1. The Higher-Order ZNN Design

In recent years, there has been a great deal of research and development into the hyperpower iteration family [48–52]. However, various continuous-time higher-order ZNN (HZNN) models were presented and studied in Refs. [36,43,53] due to the fact that iterative approaches are realizable to discrete-time models and that these methods often require starting points that are approximated and sometimes may not be easily supplied. Beginning with the subsequent hyperpower iterations with order $p \geq 2$ [36,52]:

$$W_{k+1} = W_k \sum_{i=0}^{p-1} E_k^i, \tag{3}$$

where $E_k \in \mathbb{R}^{n \times n}$ denotes a suitable time-invariant EME, it is possible to extend the time-invariant (3) to a time-varying scenario. That is, taking into account the next EME:

$$E_H^p(t) = \sum_{i=1}^{p-1} E^i(t), \tag{4}$$

where $E^i(t) \in \mathbb{R}^{n \times n}$ and $p \geq 2$, the ZNN architecture and the hyperpower iterations approach can be combined to find the online solution to a time-varying problem. This yields the next comprehensive HZNN dynamical evolution [36,43,53] (under the linear activation function):

$$\dot{E}(t) \approx -\lambda E_H^p(t). \tag{5}$$

1.2. The Noise-Handling Higher-Order ZNN Design

Every form of noise has a significant impact on the precision of the suggested ZNN methods, and any preliminary processing for a noise reduction attaches time, sacrificing desired real-time demands. As a result, an enhanced noise-handling model for handling time-varying problems was developed in Ref. [54]. The noise-handling ZNN (NZNN) dynamical system below was introduced in particular [54]:

$$\dot{E}(t) = -\lambda E(t) - \zeta \int_0^t E(\tau) d\tau + N(t), \tag{6}$$

where ζ and λ are design parameters that track NZNN convergence, while $N(t)$ stands for the proper dimensional matrix-form noises. It should be noted that [43] introduced and examined the generalization of the NZNN architecture to the NHZNN formulation for estimating a time-varying problem. The generic NHZNN dynamical evolution may be acquired by integrating the hyperpower iterations process and the NZNN design, using the same rationale as the HZNN design in (4) and (5):

$$\dot{E}(t) \approx -\lambda E_H^p(t) - \zeta \int_0^t E_H^p(\tau) d\tau + N(t). \tag{7}$$

1.3. Problem Formulation and Key Contributions

In this paper, the TVQ inverse (TVQ-INV) problem will be addressed using the HZNN and NHZNN approaches. Particularly, the following TVQ matrix equations problem is taken into consideration for computing the TVQ-INV of any nonsingular $\tilde{A}(t) \in \mathbb{H}^{n \times n}$ [19,20]:

$$I_n - \tilde{A}(t)\tilde{X}(t) = \mathbf{0}_n, \tag{8}$$

where the TVQ matrix $\tilde{X}(t) = X_1(t) + X_2(t)i + X_3(t)j + X_4(t)k \in \mathbb{H}^{n \times n}$, with $X_i(t) \in \mathbb{R}^{n \times n}$ for $i = 1, 2, 3, 4$, is the TVQ matrix of interest, $\mathbf{0}_n$ refers to the zero $n \times n$ matrix and $t \in [0, t_f] \subseteq [0, +\infty)$ is the time. Additionally, we consider that $\tilde{A}(t)$ is a smoothly time-varying matrix and its time derivative is either given or can be accurately estimated. It is important to note that (8) is the TVQ-INV problem and it is satisfied only for $\tilde{X}(t) = \tilde{A}^{-1}(t)$. Of greater significance, we will determine whether a direct solution in the quaternion domain or an indirect solution through representation in the complex and real domains is more efficient. To do this, we will create three HZNN and three NHZNN models, one for each domain, and rigorously validate them on two numerical simulations under various types of noises and a real-world application involving robotic motion tracking. By doing theoretical analysis of all presented models, this research strengthens the existing body of literature.

The following notations are employed in the remainder of this article: $\mathbf{0}_{u \times n}$ refers to the zero $u \times n$ matrix; $\mathbf{1}_n$ refers to the $n \times 1$ matrix of ones; $\|\cdot\|_F$ is the matrix Frobenius norm; $\text{vec}(\cdot)$ denotes the vectorization process; \odot denotes the elementwise multiplication; \otimes denotes the Kronecker product; the operator $()^T$ implies transposition.

The key contributions of the paper are listed next:

- (1) For the first time, the TVQ-INV problem is addressed through the HZNN and NHZNN approaches;
- (2) With the purpose of addressing the TVQ-INV problem, three novel HZNN models and three novel NHZNN models are provided;
- (3) The models are subjected to a theoretical analysis that validates them;
- (4) Numerical simulations and applications under various types of noises are carried out to complement the theoretical concepts.

The rest of the article is divided into the following sections. Section 2 presents the three HZNN and three NHZNN models, while their theoretical analysis is presented in Section 3. Numerical simulations and applications are explored in Section 4 and, finally, Section 5 provides the concluding thoughts and comments.

2. Higher Order and Noise-Handling ZNN Models in Solving the TVQ-INV

Three HZNN models will be created in this section, each of which will operate in a distinct domain. We consider that $\tilde{A}(t) \in \mathbb{H}^{n \times n}$ is a differentiable TVQ matrix and $\tilde{X}(t) \in \mathbb{H}^{n \times n}$ is the unknown TVQ matrix to be found.

2.1. The HZNNQ_p Model

The product of two TVQ matrices, $\tilde{A}(t) = A_1(t) + A_2(t)l + A_3(t)j + A_4(t)k \in \mathbb{H}^{n \times n}$ and $\tilde{X}(t) = X_1(t) + X_2(t)l + X_3(t)j + X_4(t)k \in \mathbb{H}^{n \times n}$, with $A_i(t), X_i(t) \in \mathbb{R}^{n \times n}$ for $i = 1, \dots, 4$, is:

$$\tilde{A}(t)\tilde{X}(t) = \tilde{Z}(t) = Z_1(t) + Z_2(t)l + Z_3(t)j + Z_4(t)k \in \mathbb{H}^{n \times n} \tag{9}$$

where

$$\begin{aligned} Z_1(t) &= A_1(t)X_1(t) - A_2(t)X_2(t) - A_3(t)X_3(t) - A_4(t)X_4(t), \\ Z_2(t) &= A_1(t)X_2(t) + A_2(t)X_1(t) + A_3(t)X_4(t) - A_4(t)X_3(t), \\ Z_3(t) &= A_1(t)X_3(t) + A_3(t)X_1(t) + A_4(t)X_2(t) - A_2(t)X_4(t), \\ Z_4(t) &= A_1(t)X_4(t) + A_4(t)X_1(t) + A_2(t)X_3(t) - A_3(t)X_2(t), \end{aligned} \tag{10}$$

with $Z_i(t) \in \mathbb{R}^{n \times n}$ for $i = 1, \dots, 4$. According to (8), setting $\tilde{Z}(t) = I$ in the case of TVQ-INV, the next system of equations is satisfied:

$$\begin{cases} A_1(t)X_1(t) - A_2(t)X_2(t) - A_3(t)X_3(t) - A_4(t)X_4(t) = I_n, \\ A_2(t)X_1(t) + A_1(t)X_2(t) - A_4(t)X_3(t) + A_3(t)X_4(t) = \mathbf{0}_n, \\ A_3(t)X_1(t) + A_4(t)X_2(t) + A_1(t)X_3(t) - A_2(t)X_4(t) = \mathbf{0}_n, \\ A_4(t)X_1(t) - A_3(t)X_2(t) + A_2(t)X_3(t) + A_1(t)X_4(t) = \mathbf{0}_n, \end{cases} \tag{11}$$

where $X_i(t), i = 1, \dots, 4$, are the unknown matrices of interest. Then, setting

$$\begin{aligned} B(t) &= \begin{bmatrix} A_1(t) & -A_2(t) & -A_3(t) & -A_4(t) \\ A_2(t) & A_1(t) & -A_4(t) & A_3(t) \\ A_3(t) & A_4(t) & A_1(t) & -A_2(t) \\ A_4(t) & -A_3(t) & A_2(t) & A_1(t) \end{bmatrix} \in \mathbb{R}^{4n \times 4n}, \\ Y(t) &= \begin{bmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \\ X_4(t) \end{bmatrix} \in \mathbb{R}^{4n \times n}, \quad \hat{I} = \begin{bmatrix} I_n \\ \mathbf{0}_n \\ \mathbf{0}_n \\ \mathbf{0}_n \end{bmatrix} \in \mathbb{R}^{4n \times n}, \end{aligned} \tag{12}$$

we have the following EME:

$$E(t) = \hat{I} - B(t)Y(t). \tag{13}$$

The fact that $E(t) \in \mathbb{R}^{4n \times n}$ is not a square EME and cannot be applied to the HZNN design in (5) is significant. Because of this, we may replace the $E(t)$ of (13) into the following equation without losing generality:

$$E(t) = (\hat{I} - B(t)Y(t))\hat{I}^T, \tag{14}$$

and its first time derivative is:

$$\dot{E}(t) = -(\dot{B}(t)Y(t) + B(t)\dot{Y}(t))\hat{I}^T. \tag{15}$$

Then, the following EME can be defined based on the HZNN design:

$$E_H^p(t) = \sum_{i=1}^{p-1} \left((\hat{I} - B(t)Y(t))\hat{I}^T \right)^i, \tag{16}$$

while its derivative is:

$$\dot{E}_H^p(t) = \sum_{i=1}^{p-1} \sum_{j=0}^{i-1} \left((\hat{I} - B(t)Y(t))\hat{I}^T \right)^j \left(-(\dot{B}(t)Y(t) + B(t)\dot{Y}(t))\hat{I}^T \right) \left((\hat{I} - B(t)Y(t))\hat{I}^T \right)^{i-1-j}, \tag{17}$$

the replacement $(\hat{I} - B(t)Y(t))\hat{I}^T = \mathbf{0}_{4n \times 4n}$ in (17) converts each of the summations into the null matrix, beside the summand referring to $j = 0, i = 1$. So, (17) is estimated as:

$$\dot{E}_H^p(t) \approx -(\dot{B}(t)Y(t) + B(t)\dot{Y}(t))\hat{I}^T = \dot{E}(t). \tag{18}$$

The next outcome is obtained by substituting $E_H^p(t)$ of (16) and $\dot{E}_H^p(t)$ of (17) into (5):

$$-(\dot{B}(t)Y(t) + B(t)\dot{Y}(t))\hat{I}^T = -\lambda \sum_{i=1}^{p-1} E^i(t), \tag{19}$$

and solving in terms of $\dot{Y}(t)$ yields:

$$-B(t)\dot{Y}(t)\hat{I}^T = -\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{B}(t)Y(t)\hat{I}^T. \tag{20}$$

The dynamic model of (20) can then be made simpler with the use of vectorization and Kronecker product:

$$-(\hat{I} \otimes B(t))\text{vec}(\dot{Y}(t)) = \text{vec}\left(-\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{B}(t)Y(t)\hat{I}^T\right). \tag{21}$$

Furthermore, after setting:

$$\begin{aligned} K_1(t) &= -(\hat{I} \otimes B(t)) \in \mathbb{R}^{16n^2 \times 4n^2}, & M_1(t) &= K_1^T(t)K_1(t) \in \mathbb{R}^{4n^2 \times 4n^2}, \\ K_2(t) &= \text{vec}\left(-\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{B}(t)Y(t)\hat{I}^T\right) \in \mathbb{R}^{16n^2}, & M_2(t) &= K_1^T(t)K_2(t) \in \mathbb{R}^{4n^2}, \\ \mathbf{y}(t) &= \text{vec}(Y(t)) \in \mathbb{R}^{4n^2}, & \dot{\mathbf{y}}(t) &= \text{vec}(\dot{Y}(t)) \in \mathbb{R}^{4n^2}, \end{aligned} \tag{22}$$

we arrive to the subsequent HZNN model:

$$M_1(t)\dot{\mathbf{y}}(t) = M_2(t). \tag{23}$$

The suggested HZNN model to be utilized when addressing the TVQ-INV of (8) is the dynamic model of (23), denoted by the notation HZNNQ_p .

2.2. The NHZNNQ_p Model

Additionally, the next outcome is obtained by substituting $E_H^p(t)$ of (16) and $\dot{E}_H^p(t)$ of (17) into (7):

$$-(\dot{B}(t)Y(t) + B(t)\dot{Y}(t))\hat{I}^T = -\lambda \sum_{i=1}^{p-1} E^i(t) - \zeta \int_0^t E_H^p(\tau) d\tau + N(t), \tag{24}$$

and solving in terms of $\dot{Y}(t)$ outputs:

$$-B(t)\dot{Y}(t)\hat{I}^T = -\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{B}(t)Y(t)\hat{I}^T - \zeta \int_0^t E_H^p(\tau) d\tau + N(t). \tag{25}$$

The dynamic model of (25) can then be made simpler with the use of vectorization and Kronecker product:

$$-(\hat{I} \otimes B(t))\text{vec}(\dot{Y}(t)) = \text{vec}(-\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{B}(t)Y(t)\hat{I}^T - \zeta \int_0^t E_H^p(\tau) d\tau + N(t)). \tag{26}$$

Furthermore, after setting:

$$\begin{aligned} r_q(t) &= \text{vec}(\int_0^t E_H^p(\tau) d\tau \hat{I}) \in \mathbb{R}^{16n^2}, \quad \dot{r}_q(t) = \text{vec}(\sum_{i=1}^{p-1} E^i(t)\hat{I}) \in \mathbb{R}^{16n^2}, \\ K_3(t) &= \begin{bmatrix} I_{4n^2} & \mathbf{0}_{4n^2 \times 4n^2} \\ \mathbf{0}_{16n^2 \times 4n^2} & K_1(t) \end{bmatrix} \in \mathbb{R}^{20n^2 \times 8n^2}, \quad M_3(t) = K_3^T(t)K_3(t) \in \mathbb{R}^{8n^2 \times 8n^2}, \\ K_4(t) &= \text{vec}(-\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{B}(t)Y(t)\hat{I}^T + N(t)) - \zeta r_q(t)\hat{I}^T \in \mathbb{R}^{16n^2}, \\ K_5(t) &= \begin{bmatrix} r_q(t) \\ K_4(t) \end{bmatrix} \in \mathbb{R}^{16n^2}, \quad M_4(t) = K_3^T(t)K_5(t) \in \mathbb{R}^{8n^2}, \\ \mathbf{y}_N(t) &= \begin{bmatrix} r_q(t) \\ \mathbf{y}(t) \end{bmatrix} \in \mathbb{R}^{8n^2}, \quad \dot{\mathbf{y}}_N(t) = \begin{bmatrix} \dot{r}_q(t) \\ \dot{\mathbf{y}}(t) \end{bmatrix} \in \mathbb{R}^{8n^2}, \end{aligned} \tag{27}$$

we arrive to the subsequent NHZNN model:

$$M_3(t)\dot{\mathbf{y}}_N(t) = M_4(t). \tag{28}$$

The suggested NHZNN model to be utilized when addressing the TVQ-INV of (8) under various types of noises is the dynamic model of (28), denoted by the notation NHZNNQ_p.

2.3. The HZNNQC_p Model

The following is a complex representation of the TVQ matrix $\check{A}(t)$ [22,55]:

$$\check{A}(t) = \begin{bmatrix} A_1(t) - A_4(t)\iota & -A_3(t) - A_2(t)\iota \\ A_3(t) - A_2(t)\iota & A_1(t) + A_4(t)\iota \end{bmatrix} \in \mathbb{C}^{2n \times 2n}. \tag{29}$$

Taking into account that the complex representation of the TVQ matrix acquired by multiplying two TVQ matrices is similar to the TVQ matrix acquired by multiplying the complex representations of two TVQ matrices (Theorem 1 in Ref. [22]), addressing (8) is equivalent to addressing the complex matrix equation:

$$\check{A}(t)\check{X}(t) = I_{2n}, \tag{30}$$

where $\check{X}(t) \in \mathbb{C}^{2n \times 2n}$, is the unknown matrix of interest, i.e., the complex representation of the TVQ matrix $\check{X}(t)$. Therefore, we set the next EME:

$$E(t) = I_{2n} - \check{A}(t)\check{X}(t), \tag{31}$$

and its first time derivative is:

$$\dot{E}(t) = -\dot{\check{A}}(t)\check{X}(t) - \check{A}(t)\dot{\check{X}}(t). \tag{32}$$

Then, the following EME can be defined based on the HZNN design:

$$E_H^p(t) = \sum_{i=1}^{p-1} (I_{2n} - \check{A}(t)\check{X}(t))^i, \tag{33}$$

while its derivative is:

$$\dot{E}_H^p(t) = \sum_{i=1}^{p-1} \sum_{j=0}^{i-1} (I_{2n} - \check{A}(t)\check{X}(t))^j (-\dot{\check{A}}(t)\check{X}(t) - \check{A}(t)\dot{\check{X}}(t)) (I_{2n} - \check{A}(t)\check{X}(t))^{i-1-j}, \tag{34}$$

the replacement $I_{2n} - \check{A}(t)\check{X}(t) = \mathbf{0}_{2n \times 2n}$ in (34) converts each of the summations into the null matrix, beside the summand referring to $j = 0, i = 1$. So, (34) is estimated as:

$$\dot{E}_H^p(t) \approx -\dot{\check{A}}(t)\check{X}(t) - \check{A}(t)\dot{\check{X}}(t) = \dot{E}(t). \tag{35}$$

The next outcome is obtained by substituting $E_H^p(t)$ of (33) and $\dot{E}_H^p(t)$ of (34) into (5):

$$-\dot{\check{A}}(t)\dot{\check{X}}(t) - \dot{\check{A}}(t)\check{X}(t) = -\lambda \sum_{i=1}^{p-1} E^i(t), \tag{36}$$

and solving in terms of $\dot{\check{X}}(t)$ outputs:

$$-\dot{\check{A}}(t)\dot{\check{X}}(t) = -\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{\check{A}}(t)\check{X}(t). \tag{37}$$

The dynamic model of (37) can then be made simpler with the use of vectorization and Kronecker product:

$$-(I_{2n} \otimes \dot{\check{A}}) \text{vec}(\dot{\check{X}}(t)) = \text{vec}(-\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{\check{A}}(t)\check{X}(t)) \tag{38}$$

Furthermore, after setting:

$$\begin{aligned} N_1(t) &= -(I_{2n} \otimes \dot{\check{A}}) \in \mathbb{C}^{4n^2 \times 4n^2}, & N_2(t) &= \text{vec}(-\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{\check{A}}(t)\check{X}(t)) \in \mathbb{C}^{4n^2}, \\ \mathbf{k}(t) &= \text{vec}(\dot{\check{X}}(t)) \in \mathbb{C}^{4n^2}, & \dot{\mathbf{k}}(t) &= \text{vec}(\dot{\check{X}}(t)) \in \mathbb{C}^{4n^2}, \end{aligned} \tag{39}$$

we arrive to the subsequent HZNN model:

$$N_1(t)\dot{\mathbf{k}}(t) = N_2(t). \tag{40}$$

The suggested HZNN model to be utilized when addressing the TVQ-INV of (8) under complex representation of the input TVQ matrix $\check{A}(t)$ is the dynamic model of (40), denoted by the notation HZNNQC_p.

2.4. The NHZNNQC_p Model

Additionally, the next outcome is obtained by substituting $E_H^p(t)$ of (33) and $\dot{E}_H^p(t)$ of (34) into (7):

$$-\ddot{A}(t)\dot{X}(t) - \dot{A}(t)\ddot{X}(t) = -\lambda \sum_{i=1}^{p-1} E^i(t) - \zeta \int_0^t E_H^p(\tau) d\tau + N(t), \tag{41}$$

and solving in terms of $\ddot{X}(t)$ outputs:

$$-\ddot{A}(t)\dot{X}(t) = -\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{A}(t)\ddot{X}(t) - \zeta \int_0^t E_H^p(\tau) d\tau + N(t). \tag{42}$$

The dynamic model of (42) can then be made simpler with the use of vectorization and Kronecker product:

$$-(I_{2n} \otimes \ddot{A}) \text{vec}(\ddot{X}(t)) = \text{vec}(-\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{A}(t)\ddot{X}(t) - \zeta \int_0^t E_H^p(\tau) d\tau + N(t)). \tag{43}$$

Furthermore, after setting:

$$\begin{aligned} r_c(t) &= \text{vec}\left(\int_0^t E_H^p(\tau) d\tau\right) \in \mathbb{C}^{4n^2}, \quad \dot{r}_c(t) = \text{vec}\left(\sum_{i=1}^{p-1} E^i(t)\right) \in \mathbb{C}^{4n^2}, \\ N_3(t) &= \begin{bmatrix} I_{4n^2} & \mathbf{0}_{4n^2 \times 4n^2} \\ \mathbf{0}_{4n^2 \times 4n^2} & N_1(t) \end{bmatrix} \in \mathbb{C}^{8n^2 \times 8n^2}, \\ N_4(t) &= \text{vec}\left(-\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{A}(t)\ddot{X}(t) + N(t)\right) - \zeta r_c(t) \in \mathbb{C}^{8n^2}, \\ \mathbf{k}_N(t) &= \begin{bmatrix} r_c(t) \\ \mathbf{k}(t) \end{bmatrix} \in \mathbb{C}^{8n^2}, \quad \dot{\mathbf{k}}_N(t) = \begin{bmatrix} \dot{r}_c(t) \\ \dot{\mathbf{k}}(t) \end{bmatrix} \in \mathbb{C}^{8n^2}, \end{aligned} \tag{44}$$

we arrive to the subsequent NHZNN model:

$$N_3(t)\dot{\mathbf{k}}_N(t) = N_4(t). \tag{45}$$

The suggested NHZNN model to be utilized when addressing the TVQ-INV of (8) under various types of noises is the dynamic model of (45), denoted by the notation NHZNNQC_p.

2.5. The HZNNQR_p Model

The following is a real representation of the TVQ matrix $\tilde{A}(t)$ [24]:

$$A(t) = \begin{bmatrix} A_1(t) & A_4(t) & -A_3(t) & A_2(t) \\ -A_4(t) & A_1(t) & -A_2(t) & -A_3(t) \\ A_3(t) & A_2(t) & A_1(t) & -A_4(t) \\ -A_2(t) & A_3(t) & A_4(t) & A_1(t) \end{bmatrix} \in \mathbb{R}^{4n \times 4n}. \tag{46}$$

Taking into account that the real representation of the TVQ matrix acquired by multiplying two TVQ matrices is similar to the TVQ matrix acquired by multiplying the real representations of two TVQ matrices (Corollary 1 in Ref. [24]), addressing (8) is equivalent to addressing the real matrix equation:

$$A(t)X(t) = I_{4n}, \tag{47}$$

where $X(t) \in \mathbb{R}^{4n \times 4n}$, is the unknown matrix of interest, i.e., the real representation of the TVQ matrix $\tilde{X}(t)$. Therefore, we set the next EME:

$$E(t) = I_{4n} - A(t)X(t), \tag{48}$$

and its first time derivative is:

$$\dot{E}(t) = -\dot{A}(t)X(t) - A(t)\dot{X}(t). \tag{49}$$

Then, the following EME can be defined based on the HZNN design:

$$E_H^p(t) = \sum_{i=1}^{p-1} (I_{4n} - A(t)X(t))^i, \tag{50}$$

while its derivative is:

$$\dot{E}_H^p(t) = \sum_{i=1}^{p-1} \sum_{j=0}^{i-1} (I_{4n} - A(t)X(t))^j (-A(t)\dot{X}(t) - \dot{A}(t)X(t)) (I_{4n} - A(t)X(t))^{i-1-j}, \tag{51}$$

the replacement $I_{4n} - A(t)X(t) = \mathbf{0}_{2n \times 2n}$ in (51) converts each of the summations into the null matrix, beside the summand referring to $j = 0, i = 1$. So, (51) is estimated as:

$$\dot{E}_H^p(t) \approx -A(t)\dot{X}(t) - \dot{A}(t)X(t) = \dot{E}(t). \tag{52}$$

The next outcome is obtained by substituting $E_H^p(t)$ of (50) and $\dot{E}_H^p(t)$ of (51) into (5):

$$-A(t)\dot{X}(t) - \dot{A}(t)X(t) = -\lambda \sum_{i=1}^{p-1} E^i(t), \tag{53}$$

and solving in terms of $\dot{X}(t)$ yields:

$$-A(t)\dot{X}(t) = -\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{A}(t)X(t). \tag{54}$$

The dynamic model of (54) can then be made simpler with the use of vectorization and Kronecker product:

$$-(I_{4n} \otimes A) \text{vec}(\dot{X}(t)) = \text{vec}(-\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{A}(t)X(t)) \tag{55}$$

Furthermore, after setting:

$$L_1(t) = -(I_{4n} \otimes A) \in \mathbb{R}^{16n^2 \times 16n^2}, \quad L_2(t) = \text{vec}(-\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{A}(t)X(t)) \in \mathbb{R}^{16n^2}, \tag{56}$$

$$\mathbf{x}(t) = \text{vec}(X(t)) \in \mathbb{R}^{16n^2}, \quad \dot{\mathbf{x}}(t) = \text{vec}(\dot{X}(t)) \in \mathbb{R}^{16n^2},$$

we arrive to the subsequent HZNN model:

$$L_1(t)\dot{\mathbf{x}}(t) = L_2(t). \tag{57}$$

The suggested HZNN model to be utilized when addressing the TVQ-INV of (8) under real representation of the input TVQ matrix $\tilde{A}(t)$ is the dynamic model of (57), denoted by the notation HZNNQR_p.

2.6. The NHZNNQR_p Model

Additionally, the next outcome is obtained by substituting $E_H^p(t)$ of (50) and $\dot{E}_H^p(t)$ of (51) into (7):

$$-A(t)\dot{X}(t) - \dot{A}(t)X(t) = -\lambda \sum_{i=1}^{p-1} E^i(t) - \zeta \int_0^t E_H^p(\tau) d\tau + N(t), \tag{58}$$

and solving in terms of $\dot{X}(t)$ outputs:

$$-A(t)\dot{X}(t) = -\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{A}(t)X(t) - \zeta \int_0^t E_H^p(\tau) d\tau + N(t). \tag{59}$$

The dynamic model of (59) can then be made simpler with the use of vectorization and Kronecker product:

$$-(I_{4n} \otimes A) \text{vec}(\dot{X}(t)) = \text{vec}\left(-\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{A}(t)X(t) - \zeta \int_0^t E_H^p(\tau) d\tau + N(t)\right). \tag{60}$$

Furthermore, after setting:

$$\begin{aligned} r_r(t) &= \text{vec}\left(\int_0^t E_H^p(\tau) d\tau\right) \in \mathbb{R}^{16n^2}, \quad \dot{r}_r(t) = \text{vec}\left(\sum_{i=1}^{p-1} E^i(t)\right) \in \mathbb{R}^{16n^2}, \\ L_3(t) &= \begin{bmatrix} I_{16n^2} & \mathbf{0}_{16n^2 \times 16n^2} \\ \mathbf{0}_{16n^2 \times 16n^2} & L_1(t) \end{bmatrix} \in \mathbb{R}^{32n^2 \times 32n^2}, \\ L_4(t) &= \text{vec}\left(-\lambda \sum_{i=1}^{p-1} E^i(t) + \dot{A}(t)X(t) + N(t)\right) - \zeta r_r(t) \in \mathbb{R}^{32n^2}, \\ \mathbf{k}_N(t) &= \begin{bmatrix} r_r(t) \\ \mathbf{k}(t) \end{bmatrix} \in \mathbb{R}^{32n^2}, \quad \dot{\mathbf{x}}_N(t) = \begin{bmatrix} \dot{r}_r(t) \\ \dot{\mathbf{x}}(t) \end{bmatrix} \in \mathbb{R}^{32n^2}, \end{aligned} \tag{61}$$

we arrive to the subsequent NHZNN model:

$$L_3(t)\dot{\mathbf{x}}_N(t) = L_4(t). \tag{62}$$

The suggested NHZNN model to be utilized when addressing the TVQ-INV of (8) under various types of noises is the dynamic model of (62), denoted by the notation NHZNNQR_p.

3. Stability and Convergence Analysis

This section examines the convergence and stability of the HZNN dynamics (5) and the NHZNN dynamics (7).

3.1. The HZNNQ_p, HZNNQC_p, and HZNNQR_p Models Theoretical Analysis

The following theorems examine how effectively the HZNN dynamics (5) perform.

Theorem 1. Assuming that $B(t) \in \mathbb{R}^{4n \times 4n}$ and $Y(t) \in \mathbb{R}^{4n \times n}$ are differentiable, the dynamical system (20) converges to $\tilde{A}^{-1}(t)$, which is the theoretical solution (THESO) of the TVQ-INV (8). In light of Lyapunov, the solution is thus stable.

Proof. Let $\check{Y}(t)$ be the THESO. The replacement $\tilde{Y}(t) := \check{Y}(t) - Y(t)$ entails $Y(t) = \check{Y}(t) - \tilde{Y}(t)$ and its first time derivative is $\dot{Y}(t) = \dot{\check{Y}}(t) - \dot{\tilde{Y}}(t)$. It is important to note that

$$(\hat{I} - B(t)\check{Y}(t))\hat{I}^T = \mathbf{0}_{4n \times 4n}, \tag{63}$$

and its first time derivative is:

$$-(\dot{B}(t)\check{Y}(t) + B(t)\dot{\check{Y}}(t))\hat{I}^T = \mathbf{0}_{4n \times 4n}. \tag{64}$$

Therefore, the replacement $Y(t) = \check{Y}(t) - \bar{Y}(t)$ into (16) yields:

$$\bar{E}_H^p(t) = \sum_{i=1}^{p-1} \left((\hat{I} - B(t)(\check{Y}(t) - \bar{Y}(t)))\hat{I}^T \right)^i. \tag{65}$$

Additionally, the implicit dynamics (5) denote:

$$\dot{\bar{E}}_H^p(t) = -(\dot{B}(t)(\check{Y}(t) - \bar{Y}(t)) + B(t)(\dot{\check{Y}}(t) - \dot{\bar{Y}}(t)))\hat{I}^T = -\lambda \bar{E}_H^p(t). \tag{66}$$

The candidate Lyapunov function is subsequently identified to verify convergence:

$$\mathcal{L}(t) = \frac{1}{2} \left\| \bar{E}_H^p(t) \right\|_F^2 = \frac{1}{2} \text{Tr} \left(\bar{E}_H^p(t) \left(\bar{E}_H^p(t) \right)^T \right). \tag{67}$$

The following identities may then be confirmed:

$$\dot{\mathcal{L}}(t) = \frac{2 \text{Tr} \left(\left(\bar{E}_H^p(t) \right)^T \dot{\bar{E}}_H^p(t) \right)}{2} = \text{Tr} \left(\left(\bar{E}_H^p(t) \right)^T \dot{\bar{E}}_H^p(t) \right) = -\lambda \text{Tr} \left(\left(\bar{E}_H^p(t) \right)^T \bar{E}_H^p(t) \right). \tag{68}$$

Consequently, it holds:

$$\begin{aligned} \frac{d\mathcal{L}(t)}{dt} & \begin{cases} < 0, & \bar{E}_H^p(t) \neq 0 \\ = 0, & \bar{E}_H^p(t) = 0, \end{cases} \\ \Leftrightarrow \dot{\mathcal{L}}(t) & \begin{cases} < 0, & \sum_{i=1}^{p-1} \left((\hat{I} - B(t)(\check{Y}(t) - \bar{Y}(t)))\hat{I}^T \right)^i \neq 0 \\ = 0, & \sum_{i=1}^{p-1} \left((\hat{I} - B(t)(\check{Y}(t) - \bar{Y}(t)))\hat{I}^T \right)^i = 0, \end{cases} \\ \Leftrightarrow \dot{\mathcal{L}}(t) & \begin{cases} < 0, & \bar{Y}(t) \neq 0 \\ = 0, & \bar{Y}(t) = 0. \end{cases} \end{aligned} \tag{69}$$

We have the following when the equilibrium of the system (66) is at $\bar{Y}(t)$ and $E_H^p(0) = 0$:

$$\frac{d\mathcal{L}(t)}{dt} \leq 0, \quad \forall \bar{Y}(t) \neq 0. \tag{70}$$

The state of equilibrium:

$$\bar{Y}(t) = \check{Y}(t) - Y(t) = 0, \tag{71}$$

is deemed stable by the Lyapunov stability theory. Therefore, as $t \rightarrow \infty, Y(t) \rightarrow \check{Y}(t)$. \square

Theorem 2. Let $\tilde{A}(t) \in \mathbb{H}^{n \times n}$ be differentiable. At each time t , the HZNNQ_p model (23) exponentially converges to the THESO $\check{y}(t)$ for any possible starting point $y(0)$.

Proof. For the purpose of calculating the THESO of the TVQ-INV, the EME of (14) is declared. The model (20) is determined by utilizing the HZNN’s architecture (5) for zeroing (14). Taking into consideration Theorem 1, $Y(t) \rightarrow \check{Y}(t)$ for any starting point when $t \rightarrow \infty$. Therefore, the HZNNQ_p model (23) converges to the THESO $\check{y}(t)$ for any starting point $y(0)$ when $t \rightarrow \infty$, due to the fact that it is only a different implementation of (20). The proof is thus completed. \square

Theorem 3. Assuming that $\check{A}(t) \in \mathbb{C}^{2n \times 2n}$ is differentiable, the dynamical system (37) converges to $\check{A}^{-1}(t)$, which is the THESO of the TVQ-INV (8). In light of Lyapunov, the solution is thus stable.

Proof. Given that the proof mirrors the Theorem’s 1 proof, it is omitted. \square

Theorem 4. Let $\check{A}(t) \in \mathbb{C}^{2n \times 2n}$ be differentiable. At each time t , the HZNNQC_p model (40) exponentially converges to the THESO $\check{\mathbf{k}}(t)$ for any possible starting point $\mathbf{k}(0)$.

Proof. Given that the proof mirrors the Theorem’s 2 proof once we substitute Theorem 1 with Theorem 3, it is omitted. \square

Theorem 5. Assuming that $A(t) \in \mathbb{R}^{4n \times 4n}$ is differentiable, the dynamical system (54) converges to $A^{-1}(t)$, which is the THESO of the TVQ-INV (8). In light of Lyapunov, the solution is thus stable.

Proof. Given that the proof mirrors the Theorem’s 1 proof, it is omitted. \square

Theorem 6. Let $A(t) \in \mathbb{R}^{4n \times 4n}$ be differentiable. At each time t , the HZNNQR_p model (57) exponentially converges to the THESO $\check{\mathbf{x}}(t)$ for any possible starting point $\mathbf{x}(0)$.

Proof. Given that the proof mirrors the Theorem’s 2 proof once we substitute Theorem 1 with Theorem 5, it is omitted. \square

3.2. The NHZNNQ_p, NHZNNQC_p, and NHZNNQR_p Models Theoretical Analysis

The proficiency of the NHZNN dynamics is examined in the next theorems, which are reshaped from Ref. [43], and we will attempt to solve various types of noise.

Theorem 7 ([43]). Let $\check{A}(t) \in \mathbb{H}^{n \times n}$ be differentiable. Then the NHZNNQ_p (28), NHZNNQC_p (45), and NHZNNQR_p (62) models converge globally to the THESO, in spite of the constant noise $N(t) = N \in \mathbb{R}^{\rho \times \rho}$, where $\rho = 4n$ in the cases of NHZNNQ_p and NHZNNQR_p and $\rho = 2n$ in the case of NHZNNQC_p.

Theorem 8 ([43]). Under the suppositions of Theorem 7, the NHZNNQ_p (28), NHZNNQC_p (45), and NHZNNQR_p (62) models polluted with the linear noise $N(t) = N \cdot t \in \mathbb{R}^{\rho \times \rho}$, where $\rho = 4n$ in the cases of NHZNNQ_p and NHZNNQR_p and $\rho = 2n$ in the case of NHZNNQC_p, are convergent to the THESO, with the EME’s upper bound satisfying $\lim_{t \rightarrow \infty} \|E(t)\|_F = \frac{1}{\zeta} \|N\|_F$. In addition, as $\zeta \rightarrow +\infty$, $E(t)$ fulfills $\lim_{t \rightarrow \infty} \|E_H^p(t)\|_F \downarrow 0$.

Theorem 9 ([43]). Under the assumptions of Theorem 7, the NHZNNQ_p (28), NHZNNQC_p (45), and NHZNNQR_p (62) models when there is bounded random noise $N(t) := \sigma(t) = [\sigma_{ij}(t)]_{i,j=1,\dots,n} \in \mathbb{R}^{\rho \times \rho}$, where $\rho = 4n$ in the cases of NHZNNQ_p and NHZNNQR_p and $\rho = 2n$ in the case of NHZNNQC_p, preserve bounded residual error $\|E_H^p(t)\|_F$. In addition, $\lim_{t \rightarrow \infty} \|E_H^p(t)\|_F$ of NHZNN is bounded by

$$\begin{cases} \sup_{0 \leq \tau \leq t} |\sigma_{ij}(\tau)| \frac{2\eta}{\sqrt{Q}} & , Q > 0 \\ \sup_{0 \leq \tau \leq t} |\sigma_{ij}(\tau)| \frac{4\eta\zeta}{\sqrt{-Q}} & , Q < 0 \end{cases} \tag{72}$$

where $\eta, \zeta > 0$ are parameters and $Q = -4\zeta + \eta^2$. Therefore, in the case of $Q \neq 0$, the upper bound of $\lim_{t \rightarrow \infty} \|E_H^p(t)\|_F$ is in roughly inverse analogy to η and $\lim_{t \rightarrow \infty} \|E_H^p(t)\|_F$ being arbitrarily small for adequate large η and proper ζ .

Theorem 10. Let $\tilde{A}(t) \in \mathbb{H}^{n \times n}$ be differentiable. At each time $t \in [0, t_f) \subseteq [0, +\infty)$, the NHZNNQ_p model (28) converges to the THESO $\check{y}_N(t)$ when noise is present exponentially, for any possible starting point $y_N(0)$. For each integer $p \geq 2$ when noise is present, $\tilde{A}^{-1}(t)$ is the last $4n^2$ element of $\check{y}_N(t)$.

Proof. Given that the proof mirrors Theorem 3.1 in Ref. [56] once we substitute Theorem 1 in [57] with Theorems 7, 8 and 9, respectively, for the constant noise, the bounded random noise and the linear noise, it is omitted. □

Theorem 11. Let $\hat{A}(t) \in \mathbb{C}^{2n \times 2n}$ be differentiable. At each time $t \in [0, t_f) \subseteq [0, +\infty)$, the NHZNNQC_p model (45) converges to the THESO $\check{k}_N(t)$ when noise is present exponentially, for any possible starting point $k_N(0)$. For each integer $p \geq 2$ when noise is present, $\hat{A}^{-1}(t)$ is the last $4n^2$ element of $\check{k}_N(t)$.

Proof. Given that the proof mirrors the proof of Theorem 10, it is omitted. □

Theorem 12. Let $A(t) \in \mathbb{R}^{4n \times 4n}$ be differentiable. At each time $t \in [0, t_f) \subseteq [0, +\infty)$, the NHZNNQR_p model (62) converges to the THESO $\check{x}_N(t)$ when noise is present exponentially, for any possible starting point $x_N(0)$. For each integer $p \geq 2$ when noise is present, $A^{-1}(t)$ is the last $16n^2$ elements of $\check{x}_N(t)$.

Proof. Given that the proof mirrors the proof of Theorem 10, it is omitted. □

4. Computational Simulations

We will present two simulation examples (SEs) and one application to robotic motion tracking in this section. What follows are a few crucial explanations. The HZNN design parameter λ is applied with value 10 in all SEs and with value 100 in the application. The starting points of the HZNNQ_p, HZNNQC_p, and HZNNQR_p models have been set to $y(0) = \text{vec}([A_1^T(0), A_2^T(0), A_3^T(0), A_4^T(0)]^T)$, $k(0) = \text{vec}(\check{A})$ and $x(0) = \text{vec}(A)$, respectively, and the starting points of the NHZNNQ_p, NHZNNQC_p, and NHZNNQR_p models have been set to $y_N(0) = \text{vec}([y^T(0), y^T(0)]^T)$, $k_N(0) = \text{vec}([k^T(0), k^T(0)]^T)$ and $x_N(0) = \text{vec}([x^T(0), x^T(0)]^T)$, respectively. For convenience purposes, we have set $\beta(t) = \cos(t)$ and $\alpha(t) = \sin(t)$. Further, the noises used are the next:

- $N(t) = 10 \cdot \mathbf{1}_\rho$ represents the constant noise;
- $N(t) = (2 + t/4) \cdot \mathbf{1}_\rho$ represents the linear noise;
- $N(t) = 2 + \alpha(t) \cdot \mathbf{1}_\rho/4$ represents the bounded noise.

Finally, a MATLAB ode solver, to be specific ode15s, is used with the time interval being set to $[0, 10]$ and $[0, 20]$, respectively, in all SEs and the application. For this ode solver, the default double precision arithmetic ($eps = 2.22 \times 10^{-16}$) is applied, causing the minimum value in each of the figures in this section to be primarily of the form 10^{-5} .

4.1. Simulation Examples

4.1.1. Example 1

The following are the input matrix $\tilde{A}(t)$ coefficients:

$$A_1(t) = \begin{bmatrix} 2\alpha(t) + 2 & -4 & 4 \\ 2\alpha(t) + 6 & -2 & 6 \\ 2\alpha(t) + 7 & -2 & 4 \end{bmatrix}, \quad A_2(t) = \begin{bmatrix} 6 & 2\alpha(t) + 1 & 4 \\ 5 & 3\alpha(t) + 1 & 3 \\ 5 & 2\alpha(t) + 2 & 7 \end{bmatrix},$$

$$A_3(t) = \begin{bmatrix} 3\alpha(t) + 2 & 9 & 5 \\ 2\alpha(t) + 3 & 12 & 2 \\ 3\alpha(t) + 4 & 3 & 5 \end{bmatrix}, \quad A_4(t) = \begin{bmatrix} -2 & \alpha(t) + 1 & 7 \\ -4 & 2\alpha(t) + 4 & 8 \\ -2 & 3\alpha(t) + 1 & 9 \end{bmatrix}.$$

As a consequence, $\tilde{A}(t) \in \mathbb{H}^{3 \times 3}$. The performance of the HZNN and NHZNN models is shown in Figures 1 and 2.

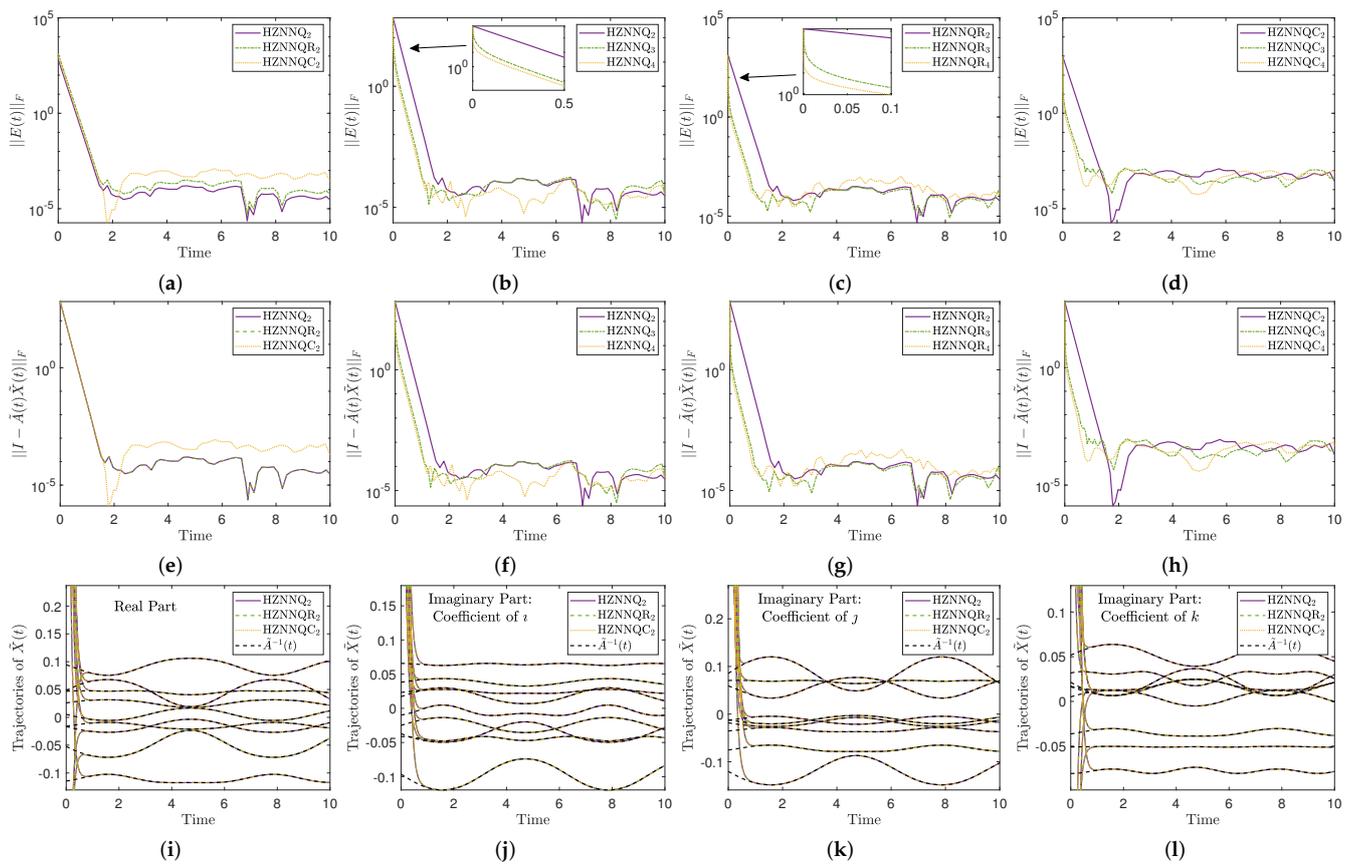


Figure 1. EMEs, error of (8), and the trajectories of $\tilde{X}(t)$ in Section 4.1.1. (a) EMEs for $p = 2$. (b) EMEs of HZNNQ $_p$. (c) EMEs of HZNNQR $_p$. (d) EMEs of HZNNQC $_p$. (e) Error of (8) for $p = 2$. (f) HZNNQ $_p$ error of (8). (g) HZNNQR $_p$ error of (8). (h) HZNNQC $_p$ error of (8). (i) Solutions traj. (j) Solutions traj. (k) Solutions traj. (l) Solutions traj.

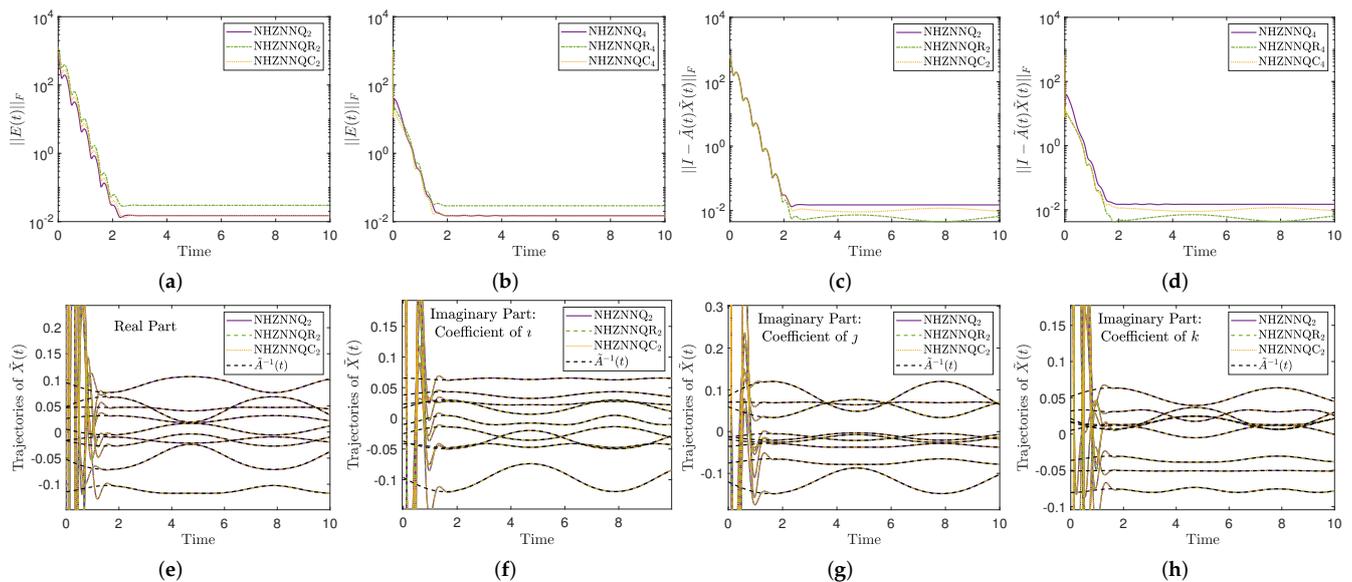


Figure 2. EMEs, error of (8), and the trajectories of $\tilde{X}(t)$ in Section 4.1.1 under linear noise with $z = 100$. (a) EMEs for $p = 2$. (b) EMEs for $p = 4$. (c) Error of (8) for $p = 2$. (d) Error of (8) for $p = 4$. (e) Solutions traj. (f) Solutions traj. (g) Solutions traj. (h) Solutions traj.

4.1.2. Example 2

Considering the following matrix

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

the following are the input matrix $\tilde{A}(t)$ coefficients:

$$A_1(t) = K \odot (1 + \alpha(t)), \quad A_2(t) = K^T \odot (1 + 2\alpha(t)), \\ A_3(t) = K \odot (1 + 3\beta(t)), \quad A_4(t) = K^T \odot (1 + 4\beta(t)).$$

As a consequence, $\tilde{A}(t) \in \mathbb{H}^{5 \times 5}$. The performance of the HZNN and NHZNN models is shown in Figures 3 and 4.

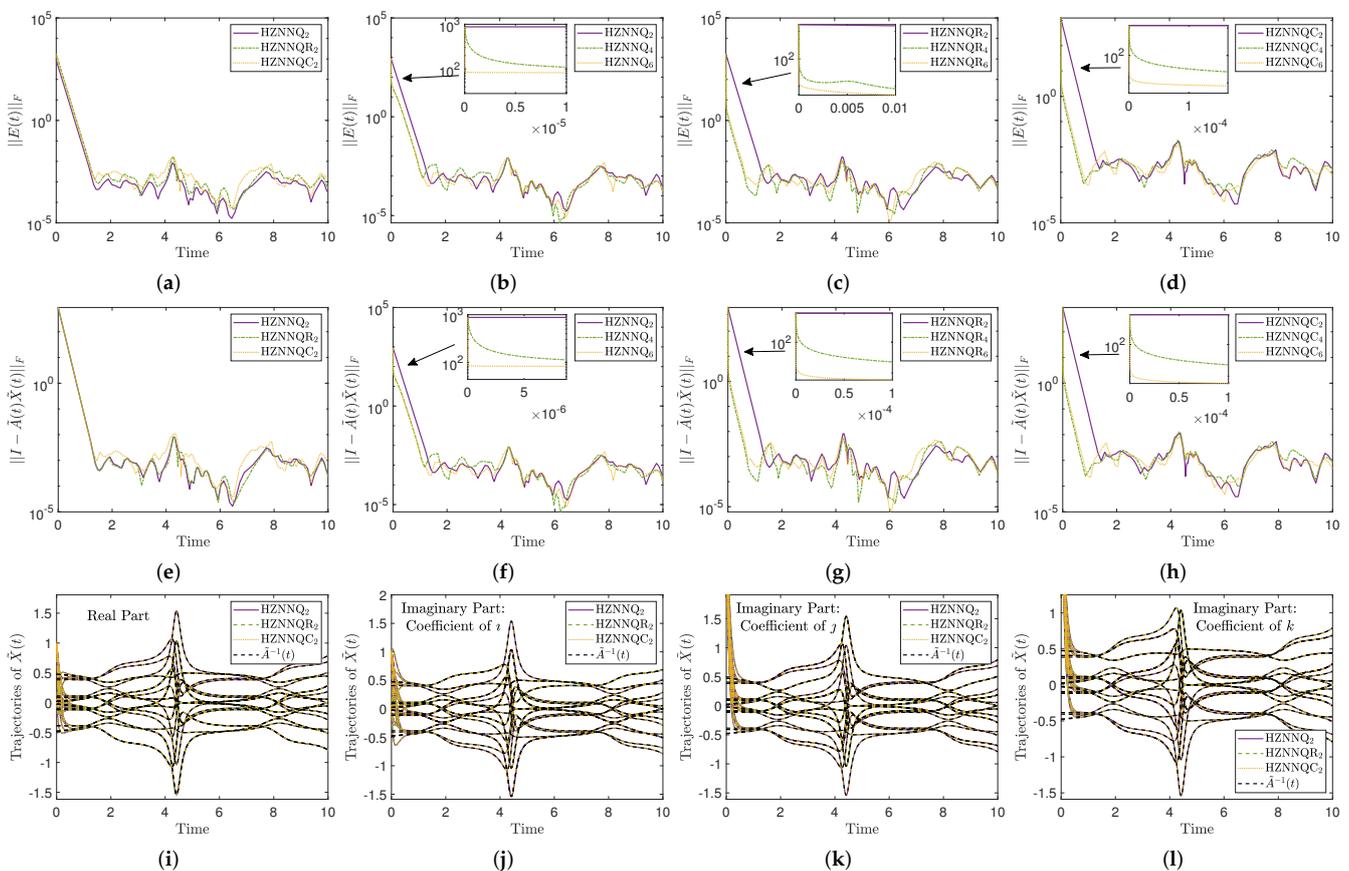


Figure 3. EMEs, error of (8), and the trajectories of $\tilde{X}(t)$ in Section 4.1.2. (a) EMEs for $p = 2$. (b) EMEs of HZNNQ $_p$. (c) EMEs of HZNNQR $_p$. (d) EMEs of HZNNQC $_p$. (e) Error of (8) for $p = 2$. (f) HZNNQ $_p$ error of (8). (g) HZNNQR $_p$ error of (8). (h) HZNNQC $_p$ error of (8). (i) Solutions traj. (j) Solutions traj. (k) Solutions traj. (l) Solutions traj.

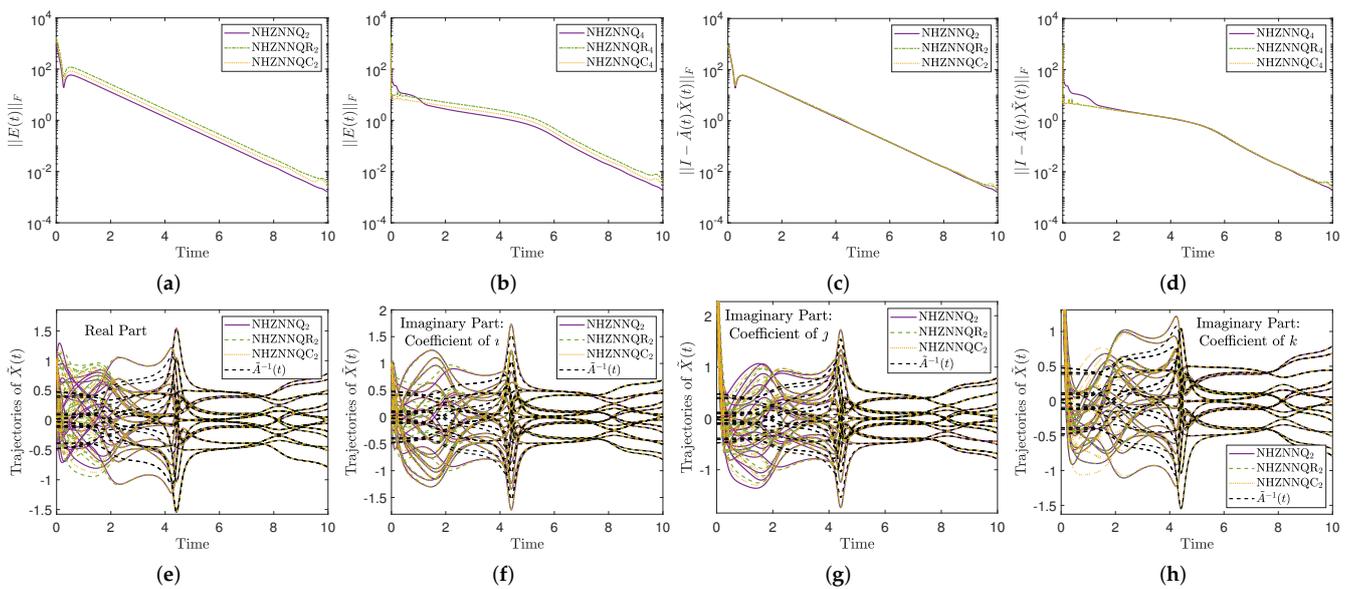


Figure 4. EMEs, error of (8), and the trajectories of $\tilde{X}(t)$ in Section 4.1.2 under constant noise with $z = 10$. (a) EMEs for $p = 2$. (b) EMEs for $p = 4$. (c) Error of (8) for $p = 2$. (d) Error of (8) for $p = 4$. (e) Solutions traj. (f) Solutions traj. (g) Solutions traj. (h) Solutions traj.

4.2. Application to Robotic Motion Tracking

The applicability of the NHZNNQ_p, NHZNNQC_p and NHZNNQR_p models is validated in this experiment using a 3-link planar manipulator (PM), as shown in Figure 5a. It is important to mention that the 3-link PM’s kinematics equations at the position level $r(t) \in \mathbb{R}^n$ and the velocity level $\dot{r}(t) \in \mathbb{R}^n$ are expressed as follows:

$$r(t) = f(\theta(t)), \quad \dot{r}(t) = J(\theta)\dot{\theta}(t), \tag{73}$$

where $\theta \in \mathbb{R}^n$ is the angle of the 3-link PM, $J(\theta) = \partial f(\theta) / \partial \theta \in \mathbb{R}^{n \times n}$, and $f(\cdot)$ is a nonlinear smooth mapping function, $r(t)$ is the end effector’s position.

To comprehend how this 3-link PM tracked motion, the inverse kinematic equation is addressed. The equation of velocity can be thought of as a linear equations system when the end effector motion tracking task is assigned with $\dot{r}(t)$ known and $\dot{\theta}(t)$ unknown. To put it another way, by setting $\tilde{A}(t) = J(\theta)$, we find $\tilde{X}(t) = A^{-1}(t)$ to solve $\dot{\theta}(t) = \tilde{X}(t)\dot{r}(t)$. Therefore, we may track control of the 3-link PM by using the ZNN models to resolve the underlying linear equation system.

The 3-link PM’s end-effector is anticipated to follow a infinity-shaped path in the simulation experiment; Ref. [58] contains the X and Y-axis velocity functions of this path along with the specifications of 3-link PM. Additionally, the link length is $\alpha = [1, 2/3, 5/4]^T$ and the initial value of the joints is $\theta(0) = [\pi/8, \pi/8, \pi/8]^T$. The performance of the NHZNN models under the bounded noise is shown in Figure 5.

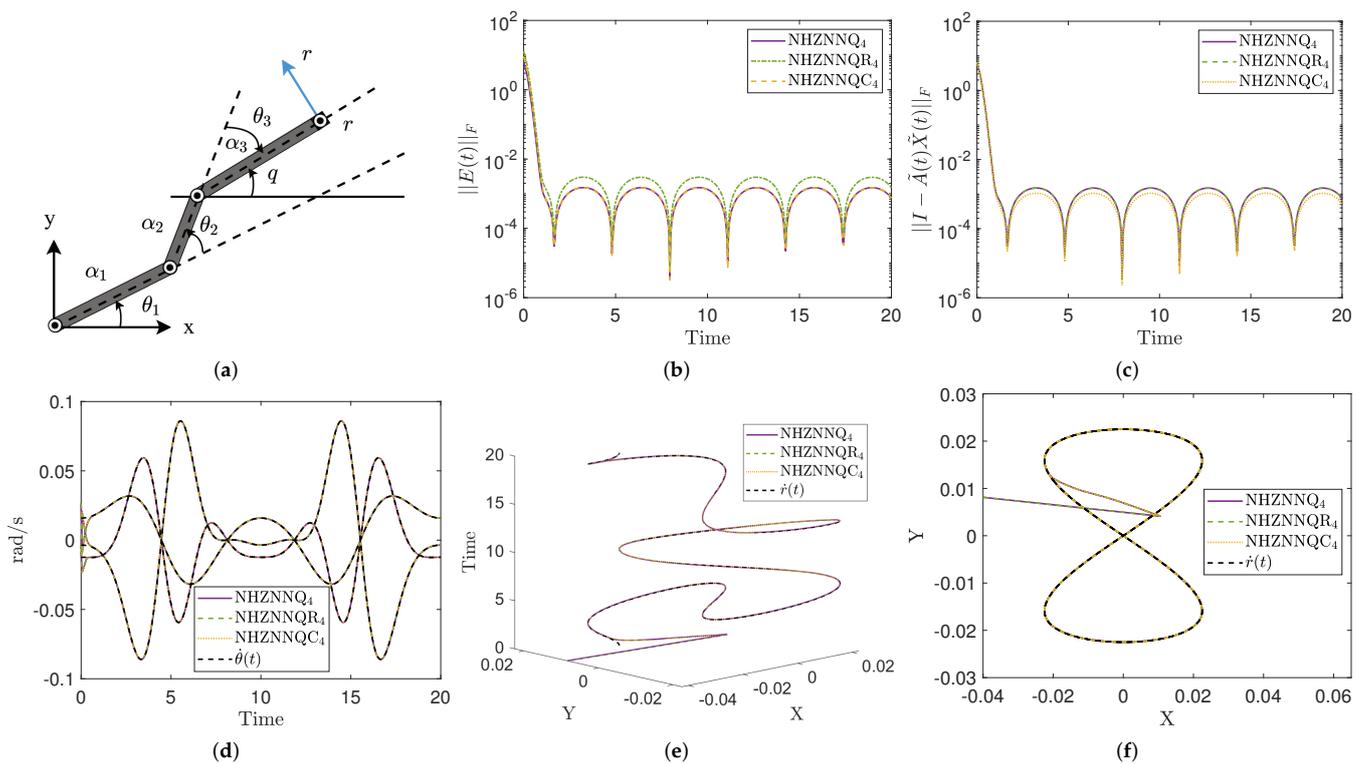


Figure 5. Robotic motion tracking application results under bounded noise with $z = 1000$. (a) 3-link PM. (b) EMEs for $p = 4$. (c) Error of (8). (d) Velocity. (e) Path tracking 3D. (f) Path tracking 2D.

4.3. Results and Discussion

The performance of the HZNNQ_p (23), HZNNQC_p (40), HZNNQR_p (57), NHZNNQ_p (28), NHZNNQC_p (45) and NHZNNQR_p (62) models for solving the TVQ-INV (8) is examined by the SEs in Sections 4.1.1 and 4.1.2. A unique TVQ-INV problem, described by the proper TVQ matrix $\tilde{A}(t)$, is assigned to each section.

For the SE in Section 4.1.1, the results of the HZNNQ_p, HZNNQC_p, and HZNNQR_p models with input TVQ matrix $\tilde{A}(t) \in \mathbb{H}^{3 \times 3}$ are presented in Figure 1. Particularly, Figure 1a–d depict the Frobenius norm of the models’ EMEs under $p = 2, 3, 4$. In the case of $p = 2$, we observe that, by the time-mark of $t \approx 2$, the models’ EMEs converge to the range $[10^{-5}, 10^{-3}]$. The time-mark, however, gets shorter as p ’s value grows. The Frobenius norm of the model’s EMEs values, shown in Table 1, likewise supports the aforementioned finding. The Frobenius norm of (8) in Figure 1e–h further supports this tendency. A higher price for λ will typically push the HZNN models to converge even more quickly. The fact that all models successfully converged is emphasized once more in Figure 1i–l, where the theoretical trajectories of the real and imaginary parts of the TVQ matrix $\tilde{A}^{-1}(t)$ are contrasted with the $\tilde{X}(t)$ trajectories retrieved by the three models. On the other hand, the results of the NHZNNQ_p, NHZNNQC_p, and NHZNNQR_p models under the linear noise with $z = 100$ are presented in Figure 2. Particularly, Figure 2a,b depict the Frobenius norm of the models’ EMEs under $p = 2$ and $p = 4$, respectively, and Figure 2c,d depict the Frobenius norm of (8) under $p = 2$ and $p = 4$, respectively. We note that the errors of the models converge to the range $[10^{-2}, 10^{-1}]$ by the time-mark of $t \approx 2$ in the case of $p = 2$ and the time-mark of $t \approx 1.5$ in the case of $p = 4$. That is, the time-mark gets shorter as p ’s value grows. The fact that all models successfully converged is emphasized once more in Figure 2e–h, where the theoretical trajectories of the real and imaginary parts of the TVQ matrix $\tilde{A}^{-1}(t)$ are contrasted with the $\tilde{X}(t)$ trajectories retrieved by the three models.

For the SE in Section 4.1.2, the results of the HZNNQ_p, HZNNQC_p, and HZNNQR_p models with input TVQ matrix $\tilde{A}(t) \in \mathbb{H}^{5 \times 5}$ are presented in Figure 3. Particularly, Figure 3a–d depict the Frobenius norm of the models’ EMEs under $p = 2, 4, 6$. In the case

of $p = 2$, we observe that, by the time-mark of $t \approx 1.8$, the models' EMEs converge to the range $[10^{-5}, 10^{-3}]$. The time-mark, however, gets shorter as p 's value grows. The EMEs' Frobenius norm values of the models, as shown in Table 1, can be used to confirm the aforementioned result. The Frobenius norm of (8) in Figure 3e–h further supports this tendency. The fact that all models successfully converged is emphasized once more in Figure 3i–l, where the theoretical trajectories of the real and imaginary parts of the TVQ matrix $\tilde{A}^{-1}(t)$ are contrasted with the $\tilde{X}(t)$ trajectories retrieved by the three models. On the other hand, the results of the NHZNNQ $_p$, NHZNNQC $_p$, and NHZNNQR $_p$ models under the constant noise with $z = 10$ are presented in Figure 4. Particularly, Figure 4a,b depict the Frobenius norm of the models' EMEs under $p = 2$ and $p = 4$, respectively, and Figure 4c,d depict the Frobenius norm of (8) under $p = 2$ and $p = 4$, respectively. We note that the errors of the models converge to the range $[10^{-3}, 10^{-2}]$ by the time-mark of $t \approx 10$ in both cases of $p = 2$ and $p = 4$. In contrast to $p = 2$, the errors of the models converge slightly more quickly when $p = 4$ in the time-range $[0, 5]$. In other words, when p 's value increases, the pace of convergence increases in the time-range $[0, 5]$. The fact that all models successfully converged is emphasized once more in Figure 4e–h, where the theoretical trajectories of the real and imaginary parts of the TVQ matrix $\tilde{A}^{-1}(t)$ are contrasted with the $\tilde{X}(t)$ trajectories retrieved by the three models.

For the application in Section 4.2, the results of the NHZNNQ $_p$, NHZNNQC $_p$, and NHZNNQR $_p$ models under the bounded noise with $z = 1000$ are presented in Figure 5. Particularly, Figure 5b,c depict the Frobenius norms of the models' EMEs and (8) under $p = 4$, respectively. We note that the errors of the models converge to the range $[10^{-5}, 10^{-2}]$ by the time-mark of $t \approx 1.5$. The fact that all models successfully converged is emphasized once more in Figure 5d–f, which depict the trajectories of the velocity and the infinity-shaped path tracking. As seen in these figures, all NHZNN model solutions match the actual velocity $\hat{\theta}(t)$, and the 3-link PM successfully completes the infinity-shaped path tracking task, where $\hat{r}(t)$ is the actual infinity-shaped path.

Lastly, once we take into account the complexity of each model, the results above can be placed into better context. Particularly, the HZNNQ $_p$ model performs $4n^2$ additions/subtractions and $(4n^2)^2$ multiplications in each iteration of (23), which results in a computational complexity of $\mathcal{O}((4n^2)^3)$ when an ode MATLAB solver is used. In the same manner, the HZNNQC $_p$ model performs $4n^2$ additions/subtractions and $(4n^2)^2$ multiplications in each iteration of (40). By converting these measurements from the complex domain into the real domain, the HZNNQC $_p$ model performs $8n^2$ additions/subtractions and $(8n^2)^2$ multiplications, which results in a computational complexity of $\mathcal{O}((8n^2)^3)$. The HZNNQR $_p$ model performs $16n^2$ additions/subtractions and $(16n^2)^2$ multiplications in each iteration of (57), which results in a computational complexity of $\mathcal{O}((16n^2)^3)$. The NHZNNQ $_p$ model performs $8n^2$ additions/subtractions and $(8n^2)^2$ multiplications in each iteration of (28), which results in a computational complexity of $\mathcal{O}((8n^2)^3)$. The NHZNNQC $_p$ model performs $8n^2$ additions/subtractions and $(8n^2)^2$ multiplications in each iteration of (45). Converting these measurements from the complex domain into the real domain, the NHZNNQC $_p$ model performs $16n^2$ additions/subtractions and $(16n^2)^2$ multiplications which results in a computational complexity of $\mathcal{O}((16n^2)^3)$. The NHZNNQR $_p$ model performs $32n^2$ additions/subtractions and $(32n^2)^2$ multiplications in each iteration of (62), which results in a computational complexity of $\mathcal{O}((32n^2)^3)$. Because the dimensions of the associated real valued matrix $A(t)$ are two times larger than those of the complex valued matrix $\tilde{A}(t)$ and four times larger than those of the quaternion valued matrix $\tilde{A}(t)$, the HZNNQR $_p$ and NHZNNQR $_p$ are, by far, the most complex models. Because of this, choosing to address the TVQ-INV problem in the real domain has a significant memory penalty, with RAM fast being a limiting factor as $\tilde{A}(t)$ grows in size. All six ZNN models can solve the TVQ-INV problem when all factors are considered, however the HZNNQ $_p$ seems to have the most potential in the absence of noise, and the NHZNNQ $_p$ seems to have the most promise in the presence of noise.

Table 1. Frobenius norm of the HZNN and NHZNN models' EMEs in SEs of Sections 4.1.1 and 4.1.2.

Model	SE of Section 4.1.1					SE of Section 4.1.2				
	p	$t = 0$	$t = 10^{-6}$	$t = 10^{-3}$	$t = 10^{-1}$	p	$t = 0$	$t = 10^{-6}$	$t = 10^{-3}$	$t = 10^{-1}$
HZNNQ _p	2	663.7	663.7	657.3	244.2	2	876.2	876.2	867.5	322.8
	3	663.7	661.8	288.1	6.6	4	876.2	236.3	79.1	22.1
	4	663.7	461.7	78.6	3.1	6	876.2	83.5	68.4	20.1
HZNNQC _p	2	1327.4	1327.4	1314.5	488.5	2	1752.5	1752.5	1735.1	645.1
	3	1327.4	1321.5	212.3	1.9	4	1752.5	439.2	16.8	1.7
	4	1327.4	512.4	23.4	0.9	6	1752.5	26.7	8.6	1.5
HZNNQR _p	2	938.6	938.6	929.5	345.1	2	1239.2	1239.2	1227.1	455.6
	3	938.6	934.5	150.2	1.3	4	1239.2	310.7	11.8	1.2
	4	938.6	362.7	16.5	0.6	6	1239.2	18.8	6.1	1.1
NHZNNQ _p	2	663.7	663.7	657.9	199.9	2	876.2	876.2	867.6	304.2
	4	663.7	461.6	76.9	34.9	4	876.2	236.3	79.1	24.5
NHZNNQR _p	2	1327.4	1327.4	1315.8	400.7	2	1752.5	1752.5	1735.2	610.1
	4	1327.4	512.5	21.3	18.5	4	1752.5	439.2	16.3	9.3
NHZNNQR _p	2	938.6	938.6	930.4	285.4	2	1239.2	1239.2	1227.1	432.8
	4	938.6	362.3	15.1	13.1	4	1239.2	310.8	11.5	6.6

5. Conclusions

In view of handling the TVQ-INV problem, three models based on the HZNN design, namely HZNNQ_p, HZNNQC_p, and HZNNQR_p, and three models based on the NHZNN design, namely NHZNNQ_p, NHZNNQC_p, and NHZNNQR_p, have been proposed. Along with SEs and a practical application to Robotic motion tracking, the theoretical investigation has aided the creation of those models. Both direct and indirect approaches to solving the TVQ-INV problem—representing the results in the complex and real domains before converting the results back to the quaternion domain—have proved effective. Of the two approaches, the direct method, used by the HZNNQ_p and NHZNNQ_p models, has been suggested as the most effective and efficient. That is, according to the principal results, each of these six models can solve the TVQ-INV effectively, and the HZNN strategy offers a faster convergence rate than the conventional ZNN strategy. In light of this, the established findings pave the path for more engaging research projects. Here are a few topics to consider for future studies:

- One may look at using nonlinear ZNNs for time-varying quaternion issues;
- It is possible to investigate using the finite-time ZNN framework to time-varying quaternion problems.

Author Contributions: All authors (R.A., M.K., H.J., T.E.S., S.D.M. and V.N.K.) contributed equally. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been funded by Research Deanship at University of Hail, Saudi Arabia, through Project Number # RG-21-139.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Ben-Israel, A.; Greville, T.N.E. *Generalized Inverses: Theory and Applications*, 2nd ed.; CMS Books in Mathematics; Springer: New York, NY, USA, 2003. [[CrossRef](#)]
2. Wang, G.; Wei, Y.; Qiao, S.; Lin, P.; Chen, Y. *Generalized Inverses: Theory and Computations*; Springer: Singapore, 2018; Volume 53.

3. Zhang, S.; Dong, Y.; Ouyang, Y.; Yin, Z.; Peng, K. Adaptive neural control for robotic manipulators with output constraints and uncertainties. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 5554–5564. [[CrossRef](#)] [[PubMed](#)]
4. Yuan, Y.; Wang, Z.; Guo, L. Event-triggered strategy design for discrete-time nonlinear quadratic games with disturbance compensations: The noncooperative case. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 1885–1896. [[CrossRef](#)]
5. Mourtas, S.D.; Katsikis, V.N.; Kasimis, C. Feedback control systems stabilization using a bio-inspired neural network. *EAI Endorsed Trans. AI Robots* **2022**, *1*, 1–13. [[CrossRef](#)]
6. Yang, X.; He, H. Self-learning robust optimal control for continuous-time nonlinear systems with mismatched disturbances. *Neural Netw.* **2018**, *99*, 19–30. [[CrossRef](#)] [[PubMed](#)]
7. Li, S.; He, J.; Li, Y.; Rafique, M.U. Distributed recurrent neural networks for cooperative control of manipulators: A game-theoretic perspective. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 415–426. [[CrossRef](#)] [[PubMed](#)]
8. Mourtas, S.D. A weights direct determination neuronet for time-series with applications in the industrial indices of the federal reserve bank of St. Louis. *J. Forecast.* **2022**, *14*, 1512–1524. [[CrossRef](#)]
9. 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.
10. Szyal-Liana, A.; Włoch, I. Generalized commutative quaternions of the Fibonacci type. *Boletín Soc. Mat. Mex.* **2022**, *28*, 1. [[CrossRef](#)]
11. Pavllo, D.; Feichtenhofer, C.; Auli, M.; Grangier, D. Modeling human motion with quaternion-based neural networks. *Int. J. Comput. Vis.* **2020**, *128*, 855–872. [[CrossRef](#)]
12. Özgür, E.; Mezouar, Y. Kinematic modeling and control of a robot arm using unit dual quaternions. *Robot. Auton. Syst.* **2016**, *77*, 66–73. [[CrossRef](#)]
13. Du, G.; Liang, Y.; Gao, B.; Otaibi, S.A.; Li, D. A cognitive joint angle compensation system based on self-feedback fuzzy neural network with incremental learning. *IEEE Trans. Ind. Inform.* **2021**, *17*, 2928–2937. [[CrossRef](#)]
14. 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, Portland, ME, USA, 11–15 August 2019; Univelt Inc.: Escondido, CA, USA, 2020; pp. 393–411.
15. Giardino, S. Quaternionic quantum mechanics in real Hilbert space. *J. Geom. Phys.* **2020**, *158*, 103956. [[CrossRef](#)]
16. Kansu, M.E. Quaternionic representation of electromagnetism for material media. *Int. J. Geom. Methods Mod. Phys.* **2019**, *16*, 1950105. [[CrossRef](#)]
17. Weng, Z.H. Field equations in the complex quaternion spaces. *Adv. Math. Phys.* **2014**, *2014*, 450262. [[CrossRef](#)]
18. Ghiloni, R.; Moretti, V.; Perotti, A. Continuous slice functional calculus in quaternionic Hilbert spaces. *Rev. Math. Phys.* **2013**, *25*, 1350006. [[CrossRef](#)]
19. Kyrchei, I.I.; Mosić, D.; Stanimirović, P.S. MPCEP- \ast CEPMP-solutions of some restricted quaternion matrix equations. *Adv. Appl. Clifford Algebr.* **2022**, *32*, 22. [[CrossRef](#)]
20. Huang, L.; Wang, Q.W.; Zhang, Y. The Moore-Penrose inverses of matrices over quaternion polynomial rings. *Linear Algebra Its Appl.* **2015**, *475*, 45–61. [[CrossRef](#)]
21. 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](#)]
22. Xiao, L.; Huang, W.; Li, X.; Sun, F.; Liao, Q.; Jia, L.; Li, J.; Liu, S. ZNNs with a varying-parameter design formula for dynamic Sylvester quaternion matrix equation. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–11. [[CrossRef](#)]
23. 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](#)]
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. 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](#)]
26. Dachang, Z.; Aodong, C.; Baolin, D.; Puchen, Z. Dual-mode synchronization predictive control of robotic manipulator. *J. Dyn. Syst. Meas. Control* **2022**, *144*, 111002. [[CrossRef](#)]
27. Jerbi, H.; Al-Darraj, I.; Tsaramirsis, G.; Ladhar, L.; Omri, M. Hamilton-Jacobi inequality adaptive robust learning tracking controller of wearable robotic knee system. *Mathematics* **2023**, *11*, 1351. [[CrossRef](#)]
28. 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](#)]
29. 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](#)]
30. Sun, Z.; Li, F.; Jin, L.; Shi, T.; Liu, K. Noise-tolerant neural algorithm for online solving time-varying full-rank matrix Moore-Penrose inverse problems: A control-theoretic approach. *Neurocomputing* **2020**, *413*, 158–172. [[CrossRef](#)]
31. 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](#)]
32. Zhang, Y.; Yang, Y.; Tan, N.; Cai, B. Zhang neural network solving for time-varying full-rank matrix Moore-Penrose inverse. *Computing* **2011**, *92*, 97–121. [[CrossRef](#)]

33. Qiao, S.; Wang, X.Z.; Wei, Y. Two finite-time convergent Zhang neural network models for time-varying complex matrix Drazin inverse. *Linear Algebra Its Appl.* **2018**, *542*, 101–117. [[CrossRef](#)]
34. 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](#)]
35. Wang, X.; Stanimirovic, P.S.; Wei, Y. Complex ZFs for computing time-varying complex outer inverses. *Neurocomputing* **2018**, *275*, 983–1001. [[CrossRef](#)]
36. Simos, T.E.; Katsikis, V.N.; Mourtas, S.D.; Stanimirović, P.S.; Gerontitis, D. A higher-order zeroing neural network for pseudoinversion of an arbitrary time-varying matrix with applications to mobile object localization. *Inf. Sci.* **2022**, *600*, 226–238. [[CrossRef](#)]
37. Zhou, M.; Chen, J.; Stanimirovic, P.S.; Katsikis, V.N.; Ma, H. Complex varying-parameter Zhang neural networks for computing core and core-EP inverse. *Neural Process. Lett.* **2020**, *51*, 1299–1329. [[CrossRef](#)]
38. 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](#)]
39. Mourtas, S.D.; Katsikis, V.N. Exploiting the Black-Litterman framework through error-correction neural networks. *Neurocomputing* **2022**, *498*, 43–58. [[CrossRef](#)]
40. Mourtas, S.D.; Kasimis, C. Exploiting mean-variance portfolio optimization problems through zeroing neural networks. *Mathematics* **2022**, *10*, 3079. [[CrossRef](#)]
41. 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](#)]
42. Katsikis, V.N.; Mourtas, S.D.; Stanimirović, P.S.; Zhang, Y. Continuous-time varying complex QR decomposition via zeroing neural dynamics. *Neural Process. Lett.* **2021**, *53*, 3573–3590. [[CrossRef](#)]
43. Stanimirović, P.S.; Katsikis, V.N.; Li, S. Higher-order ZNN dynamics. *Neural Process. Lett.* **2019**, *51*, 697–721. [[CrossRef](#)]
44. Kornilova, M.; Kovalnogov, V.; Fedorov, R.; Zamaleev, M.; Katsikis, V.N.; Mourtas, S.D.; Simos, T.E. Zeroing neural network for pseudoinversion of an arbitrary time-varying matrix based on singular value decomposition. *Mathematics* **2022**, *10*, 1208. [[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. Xiao, L.; Tan, H.; Dai, J.; Jia, L.; Tang, W. High-order error function designs to compute time-varying linear matrix equations. *Inf. Sci.* **2021**, *576*, 173–186. [[CrossRef](#)]
47. Zhong, N.; Huang, Q.; Yang, S.; Ouyang, F.; Zhang, Z. A varying-parameter recurrent neural network combined with penalty function for solving constrained multi-criteria optimization scheme for redundant robot manipulators. *IEEE Access* **2021**, *9*, 50810–50818. [[CrossRef](#)]
48. Climent, J.; Thome, N.; Wei, Y. A geometrical approach on generalized inverses by Neumann-type series. *Linear Algebra Appl.* **2001**, *332–334*, 533–540. [[CrossRef](#)]
49. Li, W.; Li, Z. A family of iterative methods for computing the approximate inverse of a square matrix and inner inverse of a non-square matrix. *Appl. Math. Comput.* **2010**, *215*, 3433–3442. [[CrossRef](#)]
50. Liu, X.; Jin, H.; Yu, Y. Higher-order convergent iterative method for computing the generalized inverse and its application to Toeplitz matrices. *Linear Algebra Appl.* **2013**, *439*, 1635–1650. [[CrossRef](#)]
51. Weiguo, L.; Juan, L.; Tiantian, Q. A family of iterative methods for computing Moore-Penrose inverse of a matrix. *Linear Algebra Appl.* **2013**, *438*, 47–56. [[CrossRef](#)]
52. Stanimirović, P.S.; Srivastava, S.; Gupta, D.K. From Zhang Neural Network to scaled hyperpower iterations. *J. Comput. Appl. Math.* **2018**, *331*, 133–155. [[CrossRef](#)]
53. Katsikis, V.N.; Stanimirović, P.S.; Mourtas, S.D.; Li, S.; Cao, X. Chapter Towards higher order dynamical systems. In *Generalized Inverses: Algorithms and Applications*; Mathematics Research Developments; Nova Science Publishers, Inc.: New York, NY, USA, 2021; pp. 207–239.
54. Jin, L.; Zhang, Y.; Li, S. Integration-Enhanced Zhang Neural Network for Real-Time-Varying Matrix Inversion in the Presence of Various Kinds of Noises. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 2615–2627. [[CrossRef](#)]
55. Farebrother, R.W.; Groß, J.; Troschke, S.O. Matrix representation of quaternions. *Linear Algebra Its Appl.* **2003**, *362*, 251–255. [[CrossRef](#)]
56. Simos, T.E.; Katsikis, V.N.; Mourtas, S.D.; Stanimirović, P.S. Finite-time convergent zeroing neural network for solving time-varying algebraic Riccati equations. *J. Frankl. Inst.* **2022**, *359*, 10867–10883. [[CrossRef](#)]
57. Liu, H.; Wang, T.; Guo, D. Design and validation of zeroing neural network to solve time-varying algebraic Riccati equation. *IEEE Access* **2020**, *8*, 211315–211323. [[CrossRef](#)]
58. Zhang, Y.; Jin, L. *Robot Manipulator Redundancy Resolution*; John Wiley & Sons: Hoboken, NJ, USA, 2017. [[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.