Color restoration of images through high order zeroing neural networks

Spyridon D. Mourtas[1,2*]

[1]Department of Economics, Division of Mathematics-Informatics and Statistics-Econometrics, National and Kapodistrian University of Athens, Sofokleous 1 Street, Athens, 10559, Greece
[2]Laboratory "Hybrid Methods of Modelling and Optimization in Complex Systems", Siberian Federal University, Prosp. Svobodny 79, Krasnoyarsk, 660041, Russia

Abstract. One of the fundamental tasks in pattern recognition is color image restoration. Every color image has three channels in the RGB color space, in contrast to grayscale images. The restoration of color images is typically far more challenging than that of grayscale images because of the internal relationships among the three channels. Given that the color image restoration can be represented as a dynamic problem with quaternion matrices, a new high order zeroing neural network (HZNN) model is developed to tackle this issue. Specifically, the time-varying quaternion matrix linear equations can be solved using the HZNN design, which is a member of the family of zeroing neural network (ZNN) models that correlate to hyperpower iterative techniques. In a realistic color image restoration application, the HZNN design outperforms the ZNN design, although both approaches work amazingly well.

1 Introduction

Color picture restoration has grown in popularity as a subject of study due to the continuously expanding importance that color images play in everyday life. Particularly, in the process of storing and transmitting photos, color images tainted by noise and blur cannot be avoided. Promising attempts have been made in the literature to restore color images from their correspondingly damaged ones, such as variation, dictionary and neural network approaches. The majority of the aforementioned approaches work with color images in the real number domain [1]. It is commonly recognized that three color channels – red, green, and blue in the RGB color space – make up color images, and that these channels have internal relationships. It is unavoidable that the intrinsic relationships between the three color channels will be neglected if they are viewed as separate matrices in the real number domain. Lately, quaternion format has been frequently employed in color image processing because of its ability to well preserve channel structure [2]. Nonetheless, color image deblurring in the quaternion domain continues to be a difficult issue.

The study of dynamic problems involving time-varying quaternion matrices (TVQM) has garnered more attention recently. These problems include the pseudoinversion of TVQM [3],

* Corresponding author: spirmour@econ.uoa.gr

the constrained TVQM least-squares issue [4] and the inversion of TVQM [5]. Furthermore, real-world applications of TVQMs include chaotic system synchronization [6], kinematically redundant manipulator of robotic joints [7], mobile manipulator control [8] and acoustic source tracking [9]. There is a commonality throughout all of these studies: they employ the zeroing neural network (ZNN) method to find the response. It is noteworthy that Zhang *et al.* introduced the ZNN technique in [10] to address time-varying tasks in real time. ZNNs, in particular, are recurrent neural networks that perform exceptionally well in parallel processing. Dynamic models for computing the time-varying pseudoinverse were among their subsequent applications [11, 12].

Let $\mathbb{H} = \{\beta_1 + \beta_2\imath + \beta_3\jmath + \beta_4 k \mid \imath^2 = \jmath^2 = k^2 = \imath\jmath k = -1, \beta_1, \beta_2, \beta_3, \beta_4 \in \mathbb{R}\}$ be the quaternions set and $\mathbb{H}^{n\times n}$ be the $n \times n$ matrices set on $\mathbb{H}$ [13]. Note that quaternions are a system of non-commutative numbers that builds upon complex numbers. In this paper, a recurrent neural network, called high order ZNN (HZNN), is used to solve the next time-varying quaternion linear matrix equation (TQLME):

$$\tilde{A}(t)\tilde{X}(t) = \tilde{C}(t), \tag{1}$$

where $\tilde{A}(t), \tilde{X}(t), \tilde{C}(t) \in \mathbb{H}^{n\times n}$.

On the one hand, a ZNN model is typically constructed through a pair of primary steps. Firstly, we need to define the function of the error matrix equation (EME), $E(t)$. Secondly, the ZNN dynamical system that follows has to be used:

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

where $t \in [0, t_f) \subseteq [0, +\infty)$ is the time and the time derivative operator is ( ). 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 [14]. The foundation of the ZNN's architecture is to put $E(t)$ to 0, which holds as $t \to \infty$. The continuous-time learning rule that results from the creation of EME in (2) is used to do this. EME can therefore be regarded as a tool for tracking ZNN model learning.

On the other hand, recent years have seen a significant amount of research and modification done on the family of hyperpower iterations [15]. However, many HZNN models were introduced and analyzed in [16] because iterative approaches can be applied to discrete-time models and because these methods usually require beginning conditions that are approximated and sometimes may not be easily provided. A HZNN model is typically constructed through a pair of primary steps. Firstly, we need to define the function of the high order error matrix equation (HEME), $E_H^p(t)$, as follows:

$$E_H^p(t) = \sum_{i=1}^{p-1} E^i(t), \tag{3}$$

where $E^i(t) \in \mathbb{R}^{n\times n}$. Secondly, the HZNN dynamical system that follows has to be used:

$$\dot{E}(t) \approx -\lambda E_H^p(t). \tag{4}$$

for calculating the online solution to a time-varying issue. Furthermore, the convergence rate of the model can be modified by varying the parameter $\lambda \in \mathbb{R}^+$ and the order $p \geq 2$. For instance, a bigger value of p causes any HZNN model to converge even quicker [16]. It is significant to note that every HZNN model reduces to a ZNN model in the case where $p = 2$ since the ZNN dynamical system of (2) and the HZNN dynamical system of (4) will match.

The key objective of this study is to develop a HZNN model to handle color image restoration challenges by addressing the TQLME of (1). To achieve this, a new HZNN model, termed HZQLME, is introduced. Furthermore, a comparison is made between the HZNN design and ZNN design's performance. According to the results of a color imagine restoration application, the HZNN design performs better than the ZNN design.

The primary ideas of this work can be summed up as follows: a novel HZNN model, termed HZQLME, for solving the TQLME of (1) is introduced; the performance of the HZNN design and ZNN design are compared; an application utilizing the HZNN concept for realistic color image restoration is showcased.

2 Methods

This section discusses the TQLME reformulation and quaternion preliminary results and presents the HZQLME model.

2.1 Preliminaries and reformulation of the TQLME

Allow $\tilde{X}(t) = X_1(t) + X_2(t)\imath + X_3(t)\jmath + X_4(t)k \in \mathbb{H}^{n\times n}$ to represent a TVQM using the coefficient matrices $X_j(t) \in \mathbb{R}^{n\times n}$ for $j = 1,2,3,4$. Similarly, consider the TVQMs $\tilde{C}(t), \tilde{A}(t) \in \mathbb{H}^{n\times n}$, which use the coefficient matrices $C_j(t), A_j(t) \in \mathbb{R}^{n\times n}$ for $j = 1,2,3,4$. The following is the product of $\tilde{A}(t)$ and $\tilde{X}(t)$:

$$\tilde{A}(t)\tilde{X}(t) = \widetilde{AX}(t) = AX_1(t) + AX_2(t)\imath + AX_3(t)\jmath + AX_4(t)k \in \mathbb{H}^{n\times n}, \tag{5}$$

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

$$\begin{aligned} AX_1(t) = A_1(t)X_1(t) - A_2(t)X_2(t) - A_3(t)X_3(t) - A_4(t)X_4(t), AX_2(t) = A_1(t)X_2(t) + \\ A_2(t)X_1(t) + A_3(t)X_4(t) - A_4(t)X_3(t), AX_3(t) = A_1(t)X_3(t) + A_3(t)X_1(t) + \\ A_4(t)X_2(t) - A_2(t)X_4(t), AX_4(t) = A_1(t)X_4(t) + A_4(t)X_1(t) + A_2(t)X_3(t) - \\ A_3(t)X_2(t). \end{aligned} \tag{6}$$

Using the previously provided information, (1) can be rewritten as follows:

$$\widetilde{AX}(t) = \tilde{C}(t), \tag{7}$$

where the following is true:

$$\begin{cases} AX_1(t) = C_1(t), \\ AX_2(t) = C_2(t), \\ AX_3(t) = C_3(t), \\ AX_4(t) = C_4(t). \end{cases} \tag{8}$$

Then, setting

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

$$C(t) = \begin{bmatrix} C_1(t) \\ C_2(t) \\ C_3(t) \\ C_4(t) \end{bmatrix} \in \mathbb{R}^{4n\times n}, \quad X(t) = \begin{bmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \\ X_4(t) \end{bmatrix} \in \mathbb{R}^{4n\times n}, \tag{9}$$

(7) can be reconstructed as follows:

$$A(t)X(t) = C(t), \tag{10}$$

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

2.2 The HZQLME model

To solve the TQLME, we will create a HZNN model called HZQLME. Let $\tilde{A}(t), \tilde{C}(t), \tilde{X}(t) \in \mathbb{H}^{n\times n}$ be differentiable TVQMs. According to the analysis of Subsection 2.1, the (10) is a reformulation of the (1). Following (9), we build the matrices $A(t) \in \mathbb{R}^{4n\times 4n}$ and $C(t) \in \mathbb{R}^{4n\times n}$, accounting for the following EME:

$$E(t) = A(t)X(t) - C(t), \tag{11}$$

where $X(t) \in \mathbb{R}^{4n\times n}$ is the matrix of interest to be found. Since the original HZNN design requires powers of $E(t)$, it is applicable only to square input matrices $A(t)$ and $C(t)$. In order to satisfy this requirement, we right multiply (11) with $\bar{I} = [I_n, \mathbf{0}_{n\times 3n}]$, where the identity

$r \times r$ matrix is indicated by I_r and the zero $r \times r$ and $r \times p$ matrices are indicated by $\mathbf{0}_r$ and $\mathbf{0}_{r\times p}$, respectively. Therefore, without loss of generality, we convert (11) as follows:

$$E_Z(t) = (A(t)X(t) - C(t))\bar{I} = A(t)X(t)\bar{I} - C(t)\bar{I}. \tag{12}$$

The first derivative of (14) is:

$$\dot{E}_Z(t) = \dot{A}(t)X(t)\bar{I} + A(t)\dot{X}(t)\bar{I} - \dot{C}(t)\bar{I}. \tag{13}$$

Further, the following HEME is defined:

$$E_H^p(t) = \sum_{i=1}^{p-1} E_Z^i(t), \tag{14}$$

while its derivative is $\dot{E}_H^p(t) \approx \dot{E}_Z(t)$ according to the HZNN design. The HZNN dynamical system is then treated in terms of $\dot{X}(t)$. This is achieved by substituting the aforementioned $E_H^p(t)$ and $\dot{E}_H^p(t)$ in (4). The result is as follows:

$$A(t)\dot{X}(t)\bar{I} = -\lambda \sum_{i=1}^{p-1} E_Z^i(t) - \dot{A}(t)X(t)\bar{I} + \dot{C}(t)\bar{I}. \tag{15}$$

The Kronecker product, $\otimes$, and vectorization, $\mathrm{vec}(\cdot)$, are utilized to streamline the dynamics of (15):

$$(\bar{I}^{\mathrm{T}} \otimes A(t))\mathrm{vec}(\dot{X}(t)) = \mathrm{vec}(-\lambda \sum_{i=1}^{p-1} E_Z^i(t) - \dot{A}(t)X(t)\bar{I} + \dot{C}(t)\bar{I}), \tag{16}$$

where the superscript $()^{\mathrm{T}}$ signifies transposition and the mass matrix $\bar{I}^{\mathrm{T}} \otimes A(t)$ is invertible. The equation's (16) dynamic model, additionally referred to as HZQLME, is the suggested HZNN model to be used in solving the TQLME of (1). Further, the HZQLME model's stability analysis and convergence is presented in the following theorems.

Theorem 1 *Let $A(t) \in \mathbb{R}^{4n\times 4n}$ and $C(t) \in \mathbb{R}^{4n\times n}$ be differentiable. The theoretical solution (TSOL) $X_{\mathcal{TS}}(t)$ is reached by the dynamics of (15) in line with the HZNN method (4), and it is stable, in accordance with Lyapunov.*

Proof. The proof is omitted since it is similar to the proof in [16, Theorem 1]. □

Theorem 2 *Let $\tilde{A}(t), \tilde{C}(t) \in \mathbb{H}^{n\times n}$ be differentiable. At each $t \in [0, t_f) \subseteq [0, +\infty)$, the HZQLME model (16) converges to the TSOL $vec(X_{\mathcal{TS}}(t))$ exponentially for any choice of starting value $vec(X(0))$.*

Proof. First, the TQLME of (1) is transformed into the (10). More specifically, we use (9) to build the matrices $A(t) \in \mathbb{R}^{4n\times 4n}$ and $C(t) \in \mathbb{R}^{4n\times n}$ using the matrices $\tilde{A}(t)$ and $\tilde{C}(t)$. We therefore change the (1) to the (10). Secondly, the HEME of (14) is declared in order to solve the (10). The model (15) is used in accordance to the HZNN method (4) for zeroing (14). Theorem 2.2 states that when $t \to \infty$ for any choice of starting value, $X(t) \to X_{\mathcal{TS}}(t)$. Since (10) is a reformulation of the TQLME of (1), the model (15) converges to equation's (1) TSOL. Third, the Kronecker product and vectorization are used to simplify (15) and form the HZQLME model of (16). An alternate form of (15) is the HZQLME, which converges to the TSOL $\mathrm{vec}(X_{\mathcal{TS}}(t))$ when $t \to \infty$ for any starting value $\mathrm{vec}(X(0))$. After then, the proof is complete. □

3 Results and discussion

In this section we shall present an application to color restoration of images. Next are a few significant clarifications. The HZNN parameter λ is used with value 10, the order p is used with values $2,4$ and 6, while the initial values of the HZQLME model has been set to $\mathrm{vec}(X(0)) = \mathbf{0}_{4n^2}$. A MATLAB ode solver, namely ode15s, is used in the computations with the time interval being set to $[0,10]$. The matrix Frobenius norm will be represented as $\|\cdot\|_{\mathrm{F}}$, while the notation HZNN $_2$, HZNN $_4$ and HZNN $_6$ refer to the HZNN design under the orders $2,4$ and 6, respectively. It should be noted that the ZNN model is the HZNN $_2$ since, for $p = 2$, the HZNN dynamical system of (4) and the ZNN dynamical system of (2) match.

(a) Original image. (b) Contaminated image. (c) Restored image.

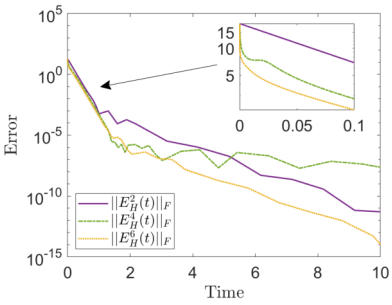
Fig. 1. Image restoration results.

The Mona Lisa thumbnail, which measures 256×256 pixels, is the picture that was used in this application (Figure 1a). Let $\tilde{V} = R_r\iota + G_gJ + B_bk$ be used to represent a colored image. Then the three imaginary parts of $\tilde{V} \in \mathbb{H}^{256\times256}$ are pixel matrices of red, green and blue channels of the color image. That is, $R_r, G_g, B_b \in \mathbb{R}^{256\times256}$ refer to red, green and blue channels, respectively. Given that $\tilde{A} = \frac{120}{100}I_{256}\iota + \frac{80}{100}I_{256}J + \frac{20}{100}I_{256}k$, the matrix $\tilde{C}$ representing the contaminated image is produced by $\tilde{C} = \tilde{A}\tilde{V}$. In order to get $\tilde{X} = \tilde{V}$, we need to solve (1). This will return the image to its initial condition. It is important to remember that the coefficient matrices of $\tilde{V}, \tilde{A}$ and $\tilde{C}$ are sparse matrices.

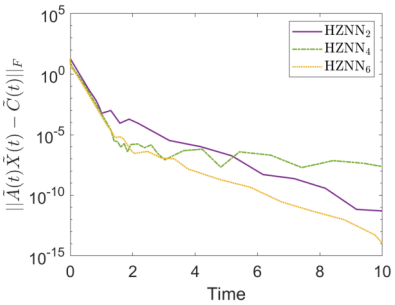
The HZNN model's findings are presented in Figures 1 and 2. Particularly, the pathways of each model's HEMEs are shown in Figure 2a, from which we can infer the models' convergence. The values of the HEMEs of all three HZNNs decrease as t increases from 0 to 10. Specifically, we notice that the HEME of the HOZNN models with values in the range $[10,10^2]$ starts to converge at $t = 0$. The HOZNN models exhibit quicker convergence prior to $t = 2$ when p is larger, and the HEMEs' lowest values fall between $[10^{-7}, 10^{-3}]$. The HOZNN models converge to the zero matrix at a slower rate than before $t = 2$. Stated differently, the HZNN models' HEME starts with an unfavorable initial condition and gradually approaches the zero matrix. The models' HEME ending values fall within the range of 10^{-10}. As a result, the three HZNNs have converged. The TS trajectories of the real and imaginary parts of $\tilde{X}(t)$ are contrasted to the trajectories obtained by the three models in Figures 2c-2f, which emphasize this further.

On the other hand, Figure 2b provides a better evaluation of each model's performance on the task of image color restoration. There, we have shown the associated $||\tilde{A}(t)\tilde{X}(t) - \tilde{B}(t)||_{\mathrm{F}}$ curves for each model after transforming all involved matrices back into quaternion form. The HZNN $_2$, HZNN $_4$, and HZNN $_6$ models' corresponding curves have the same routes and are positioned lower as p increases. Also, the results displayed there confirm those in Figure 2a, indicating that solving (10) is the same as solving (1).

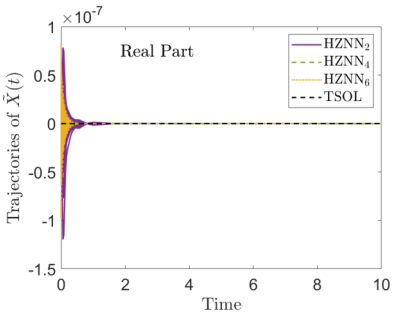
It is clear that the color restoration procedure was successful after examining the original image (Figure 1a), the polluted image (Figure 1b), and finally the restored image (Figure 1c). As a result, tainted photos can be successfully restored to their natural color using the created HZQLME model. Also, it is crucial to mention that the ZNN model performs better than the HZNN model because at $p = 2$, the HZNN model turns into a ZNN model, while at $p > 2$, the HZNN model achieves better convergence. In conclusion, the HZNN approach can be used to restore the color of images in addition to being effective at solving the TQLME.



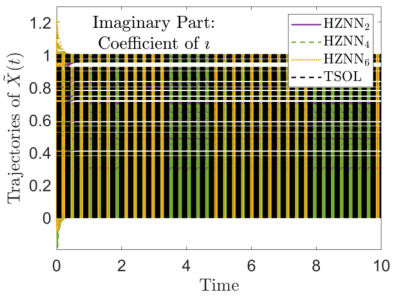
(a) HEMEs values.



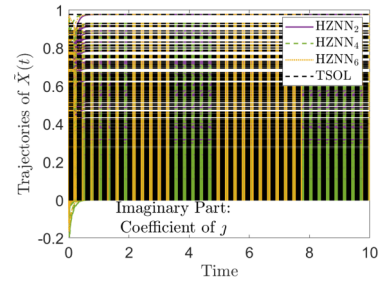
(b) Error of TQLME.



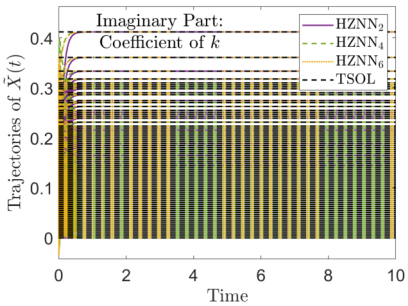
(c) Traj. of $\tilde{X}(t)$.



(d) Traj. of $\tilde{X}(t)$.



(e) Traj. of $\tilde{X}(t)$.



(f) Traj. of $\tilde{X}(t)$.

Fig. 2. Errors, and solutions trajectories.

4 Conclusion

In this paper, a novel HZNN model, termed HZQLME, is presented for solving the TQLME. An application to color restoration of images is carried out to support the HZQLME model practical applicability and accuracy. The results of this application show that the HZNN method is more effective than the ZNN method. So, we can conclude that the HZNN method is not only fruitful for solving the TQLME, but it may also be employed to color restoration of images.

This work was supported by the Ministry of Science and Higher Education of the Russian Federation (Grant No. 075-15-2022-1121).

References

1. Q. Dai, F. Fang, J. Li, G. Zhang, Pattern Recognit **118**, 108019 (2021)
2. C. Huang, Z. Li, Y. Liu, T. Wu, T. Zeng, Pattern Recognit **128**, 108665 (2022)
3. V.N. Kovalnogov, R.V. Fedorov, D.A. Demidov, M.A. Malyoshina, T.E. Simos, S.D. Mourtas, V.N. Katsikis, AIMS Math **8**, 22875-22895 (2023)
4. L. Xiao, P. Cao, W. Song, L. Luo, W. Tang, IEEE Trans. Neural Netw. Learn. Syst., 1-10 (2023)
5. L. Xiao, S. Liu, X. Wang, Y. He, L. Jia, Y. Xu, IEEE Trans. Ind. Informatics **18**, 1562-1571 (2022)
6. S.B. Aoun, N. Derbel, H. Jerbi, T. E. Simos, S. D. Mourtas, V. N. Katsikis, AIMS Math. **8**, 27376-27395 (2023)
7. N. Tan, P. Yu, F. Ni, IEEE Trans. Instrum. Meas. **71**, 1-14 (2022)
8. R. Abbassi, H. Jerbi, M. Kchaou, T.E. Simos, S.D. Mourtas, V.N. Katsikis, Mathematics **11**, 2756 (2023)
9. V.N. Kovalnogov, R.V. Fedorov, I.I. Shepelev, V.V. Sherkunov, T.E. Simos, S.D. Mourtas, V.N. Katsikis, AIMS Math **8,** 25966-25989 (2023)
10. Y. Zhang, S.S. Ge, IEEE Trans. Neural Netw. **16**, 1477-1490 (2005)
11. Y. Chai, H. Li, D. Qiao, S. Qin, J. Feng, Int. J. Comput. Intell. Syst. **13**, 663-671 (2020)
12. W. Wu, B. Zheng, Neurocomputing **418**, 221-231 (2020)
13. F. Zhang, Linear algebra Appl. **251**, 21-57 (1997)
14. J. Dai, P. Tan, X. Yang, L. Xiao, L. Jia, Y. He, Knowledge-Based Syst. **242**, 108405 (2022)
15. P.S. Stanimirović, S. Srivastava, D.K. Gupta, J. Comput. Appl. Math. **331**, 133-155 (2018)
16. H. Jerbi, H. Alharbi, M. Omri, L. Ladhar, T.E. Simos, S.D. Mourtas, V.N. Katsikis, Mathematics **10**, 4490 (2022)