Applied Mathematics and Computation 465 (2024) 128412

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Zeroing neural network approaches for computing time-varying minimal rank outer inverse

Predrag S. Stanimirović [a,c], Spyridon D. Mourtas [b,c], Dijana Mosić [a], Vasilios N. Katsikis [b,*], Xinwei Cao [d], Shuai Li [e]

[a] University of Niš, Faculty of Sciences and Mathematics, Višegradska 33, 18000 Niš, Serbia
[b] Department of Economics, Division of Mathematics-Informatics and Statistics-Econometrics, National and Kapodistrian University of Athens, Sofokleous 1 Street, 10559 Athens, Greece
[c] Laboratory "Hybrid Methods of Modelling and Optimization in Complex Systems", Siberian Federal University, Prosp. Svobodny 79, 660041 Krasnoyarsk, Russia
[d] School of Business, Jiangnan University, Lihu Blvd, Wuxi 214122, China
[e] Faculty of Information Technology, University of Oulu, Oulu, Finland

ARTICLE INFO

Keywords:
Matrix equation
Zeroing neural network
Generalized inverse
Dynamic system
Minimal rank outer inverse

ABSTRACT

Generalized inverses are extremely effective in many areas of mathematics and engineering. The zeroing neural network (ZNN) technique, which is currently recognized as the state-of-the-art approach for calculating the time-varying Moore-Penrose matrix inverse, is investigated in this study as a solution to the problem of calculating the time-varying minimum rank outer inverse (TV-MROI) with prescribed range and/or TV-MROI with prescribed kernel. As a result, four novel ZNN models are introduced for computing the TV-MROI, and their efficiency is examined. Numerical tests examine and validate the effectiveness of the introduced ZNN models for calculating TV-MROI with prescribed range and/or prescribed kernel.

1. Introduction and preliminaries

The real-time solution for generalized inverses [1,2], that arises very often in game theory [3], robotics [4–6], nonlinear systems [7,8], and other scientific and technical disciplines [9–11], has drawn a lot of curiosity in recent years. In particular, applications in engineering involve overseeing of attitude and orbit control systems using torque and force actuators [12], keeping track of humanoid robot motions [13], overseeing an aircraft's fault-tolerant control mode [14], as well as signal source tracking [15], navigation [16] and wireless communications [17].

The representation and computation of several generalized inverses are tightly related in the following four Penrose equations:

$$(i)\ AXA = A, \quad (ii)\ XAX = X, \quad (iii)\ (AX)^* = AX, \quad (iv)\ (XA)^* = XA.$$

The notations $A^\dagger$, A^* and $\mathrm{ra}(A)$ will represent the pseudoinverse, conjugate-transpose and rank of a matrix A, respectively, while the set of $m \times n$ matrices in the field of complex numbers $\mathbb{C}$ will be denoted as $\mathbb{C}^{m\times n}$. The symbol $A\{\rho\}$ is stated for the set of all

* Corresponding author.
E-mail addresses: pecko@pmf.ni.ac.rs (P.S. Stanimirović), spirmour@econ.uoa.gr (S.D. Mourtas), dijana@pmf.ni.ac.rs (D. Mosić), vaskatsikis@econ.uoa.gr (V.N. Katsikis), xwcao@jiangnan.edu.cn (X. Cao), lishuai8@gmail.com (S. Li).

https://doi.org/10.1016/j.amc.2023.128412
Received 25 July 2023; Received in revised form 27 September 2023; Accepted 18 October 2023

Available online 24 October 2023

matrices that satisfy equations defined by $\rho \subseteq \{1,2,3,4\}$. A ρ-inverse of A, marked with $A^{(\rho)}$, is any matrix from $A\{\rho\}$. Notice that $A\{1,2,3,4\} = \{A^{\dagger}\}$. The class consisting of outer inverses (or $\{2\}$-inverses) is defined for an arbitrary matrix $A \in \mathbb{C}^{m\times n}$ by

$$A\{2\} = \{X \in \mathbb{C}^{n\times m} \mid XAX = X\}. \tag{1.1}$$

Immediately from the definition, it can be concluded $\mathrm{ra}(A^{(2)}) \leq \mathrm{ra}(A)$. Further, it is known that an arbitrary $X \in A\{1,2\}$ satisfies $\mathrm{ra}(X) = \mathrm{ra}(A)$. The outer inverses are proved to be useful in statistics [18], in defining the iterative themes for tackling nonlinear equations [1], in stable approximations of ill-posed problems, and in linear and nonlinear issues implicating rank-deficient generalized inverses [19].

According to [20], the minimal rank outer inverse (MROI), $X \in \mathbb{C}^{n\times m}$, can be found by solving the systems of matrix equations involved in Proposition 1.1.

Proposition 1.1 *(Lemma 1, [20]). Assume $A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{n\times k}$ and $Q \in \mathbb{C}^{l\times m}$. Then it follows*

$$(\mathbf{a})\ \exists X \in \mathbb{C}^{n\times m} :\ XAB = B \Longleftrightarrow \mathrm{ra}(AB) = \mathrm{ra}(B). \tag{1.2}$$

$$(\mathbf{b})\ \exists X \in \mathbb{C}^{n\times m} :\ QAX = Q \Longleftrightarrow \mathrm{ra}(QA) = \mathrm{ra}(Q). \tag{1.3}$$

Some additional characterizations of (1.2) and (1.3) are presented in the next Proposition 1.2.

Proposition 1.2 *(Theorems 1 and 5, [20]). Assume $A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{n\times k}$, $Q \in \mathbb{C}^{l\times m}$.*

(a) *For a MROI $X \in \mathbb{C}^{n\times m}$ with prescribed range $\mathcal{R}(X) = \mathcal{R}(B)$, the subsequent statements are mutually equivalent:*

(i) $XAB = B$ *and* $\mathrm{ra}(X) = \mathrm{ra}(B)$;
(ii) $X = BB^{\dagger}X$ *and* $XAB = B$;
(iii) $XAX = X$, $X = BB^{\dagger}X$ *and* $XAB = B$.

(b) *For a MROI $X \in \mathbb{C}^{n\times m}$ with prescribed kernel $\mathcal{N}(X) = \mathcal{N}(Q)$, the subsequent statements are mutually equivalent:*

(i) $QAX = Q$ *and* $\mathrm{ra}(X) = \mathrm{ra}(Q)$;
(ii) $X = XQ^{\dagger}C$ *and* $QAX = Q$;
(iii) $XAX = X$, $X = XQ^{\dagger}Q$ *and* $QAX = Q$.

The rank of a rectangular matrix A is the maximal number of linearly independent rows or columns in A. It can be found using techniques like Gaussian elimination to reduce the matrix to its reduced row-echelon form (RREF). The number of nonzero rows in the RREF is equal to the rank of the matrix. Difficulties in computing the rank of a constant matrix and three methods for computing the matrix rank are presented in [21].

Matrices considered in the actual paper are time-varying matrices with respect to the variable t representing the time. The normal rank of a matrix $A(t)$ will be denoted by $\mathrm{nr}(A)$ and it is defined as the number of non-zero rows in its normal form. If $\mathrm{ra}(A)$ is the rank of a constant matrix A then the normal rank of a time-varying matrix $A(t)$ in considered time interval $t \in [0, t_f)$ is equal to

$$\mathrm{nr}(A(t)) = \max_{t\in[0,t_f)} \mathrm{ra}(A(t)).$$

In this research, the problems of finding time-varying MROI (TV-MROI) with prescribed range (PR) and/or TV-MROI with prescribed kernel (PK) are considered. In line with Proposition 1.1 and without loss of generality, we assume the time-varying matrices $A(t) \in \mathbb{C}^{m\times n}$, $B(t) \in \mathbb{C}^{n\times k}$ and $Q(t) \in \mathbb{C}^{l\times m}$ and unknown matrix $X(t) \in \mathbb{C}^{n\times m}$, where $t \in [0, t_f) \subseteq [0, +\infty)$ denotes the time. Therefore, the next Proposition 1.3 is the subject of our research.

Proposition 1.3. *Assume the time-varying matrices $B(t) \in \mathbb{C}^{n\times k}$, $A(t) \in \mathbb{C}^{m\times n}$ and $Q(t) \in \mathbb{C}^{l\times m}$. Then it follows:*

$$(\mathbf{a})\ \exists X(t) \in \mathbb{C}^{n\times m} :\ B(t) = X(t)A(t)B(t) \Longleftrightarrow \mathrm{nr}(A(t)B(t)) = \mathrm{nr}(B(t)). \tag{1.4}$$

$$(\mathbf{b})\ \exists X(t) \in \mathbb{C}^{n\times m} :\ Q(t) = Q(t)A(t)X(t) \Longleftrightarrow \mathrm{nr}(Q(t)A(t)) = \mathrm{nr}(Q(t)). \tag{1.5}$$

Additionally, several of the basic symbols used in the paper are noteworthy. The identity $g\times g$ matrix will be referred to as I_g whereas the zero $g\times g$ and $m\times n$ matrices will be referred to as $\mathbf{0}_g$ and $\mathbf{0}_{m\times n}$, respectively, and the all-ones $g\times g$ and $m\times n$ matrices will be referred to as $\mathbf{1}_g$ and $\mathbf{1}_{m\times n}$, respectively. Moreover, the vectorization operator will be denoted as $\mathrm{vec}(\cdot)$, the Kronecker product will be denoted as $\otimes$, the Hadamard product will be denoted as $\odot$, and the time-derivative by $\dot{(\cdot)}$. Last, $()^{\mathrm{T}}$ will denote transposition, $\imath$ will denote the pure imaginary unit satisfying $\imath^2 = -1$, $\|\cdot\|_{\mathrm{F}}$ will denote the matrix Frobenius norm and $\beta \geq 0$ is the Tikhonov regularization parameter. It is important to mention that the Tikhonov regularization parameter is frequently used to address singularity issues.

In order to deal with time-varying tasks in real time, the zeroing neural network (ZNN) technique is introduced by Zhang et al. in [22]. ZNNs are a particular kind of recurrent neural networks which excel in parallel processing and their next acceptations were

dynamic models for calculating the time-varying Moore-Penrose inverse in the real and complex domains [23–26]. They are used to solve problems involving generalized inversion [27–31], linear and quadratic programming [32,33], systems of nonlinear equations [34–36], and linear matrix equations (LMEs) [37–39], among other issues. Further, the ZNN technique will be used to address the TV-MROI with PR and/or with PK problems. Two main processes are normally involved in the construction of a ZNN model. The function of error matrix equation (EME) $E(t)$ must first be defined. Second, the following ZNN dynamical system must be employed:

$$\dot{E}(t) = -\lambda E(t). \tag{1.6}$$

On top of that, one can change the model's convergence rate by adjusting the parameter $\lambda \in \mathbb{R}^+$. Any ZNN model will converge more quickly with a bigger value of λ [40–42]. The ZNN's architecture is based on setting each element of $E(t)$ to 0, which is true as $t \to \infty$. This is accomplished using the continuous-time learning regulation that arises from the establishment of EME in (1.6). As a consequence, EME can be considered a tool for monitoring ZNN model learning.

Based on the ZNN design in (1.6), four novel ZNN models are introduced for computing the TV-MROI with PR and with PR. Particularly, two ZNN models, one for a specific solution and another for a random one, are proposed for computing TV-MROI with PR, and another two ZNN models, one for a specific solution and another for a random one, are proposed for computing TV-MROI with PK. For investigating the performance of the models, six numerical examples are presented. The findings show that all models address the underlying problem with outstanding and effective performance.

The following is the paper's main highlights:

(1) The TV-MROI with PR and/or with PK problems are addressed for the first time through the ZNN approach.
(2) With the purpose of computing the TV-MROI with PR and/or PK, four novel ZNN models are introduced.
(3) A theoretical investigation is performed on the models to validate them.
(4) To support the theoretical notions, numerical experiments are conducted.

The remainder of the paper is structured as follows. The ZNN designs for computing TV-MROI with PR and/or PK are presented in sections 2 and 3, respectively. The theoretical analysis of both the models is presented in sections 2 and 3. Computational complexity is examined in Section 4. Numerical experiments are presented in Section 5. Lastly, final thoughts and comments are provided in Section 6.

2. ZNN designs for computing TV-MROI with PR

Two ZNN models are described in this section for computing TV-MROI with PR, one for a specific solution and another for a random one. Additionally, we consider that $A(t) \in \mathbb{C}^{m\times n}$ and $B(t) \in \mathbb{C}^{n\times k}$ are differentiable time-varying matrices.

According to Propositions 1.2 and 1.3, the following equation group must be solved in order to obtain a TV-MROI with PR:

$$\begin{cases} B(t)B^{\dagger}(t)X(t) = X(t), \\ X(t)A(t)B(t) = B(t). \end{cases} \tag{2.1}$$

Notice that the TV-MROI with PR in (2.1) is the unknown $X(t) \in \mathbb{C}^{n\times m}$.

2.1. ZNN model for computing a specific solution of the TV-MROI with PR

Considering the minimal-norm least-square solution $X(t) = B(t)(A(t)B(t))^{\dagger}$ to the matrix equation $X(t)A(t)B(t) = B(t)$ and utilizing the identity

$$(A(t)B(t))^{*}A(t)B(t)(A(t)B(t))^{\dagger} = (A(t)B(t))^{*}$$

arising from the property $A^{*}AA^{\dagger} = A^{*}$ [1,2], we set the unknown matrix $Y(t) = (A(t)B(t))^{\dagger} \in \mathbb{C}^{k\times m}$ so that (2.1) can be converted as

$$\begin{cases} (A(t)B(t))^{*}A(t)B(t)Y(t) = (A(t)B(t))^{*}, \\ X(t) = B(t)Y(t). \end{cases} \tag{2.2}$$

It is important to mention that (2.2) always computes the specific solution of the TV-MROI with PR problem in (2.1) that is based on the pseudoinverse of the product $A(t)B(t)$.

Theorem 2.1. *If the rank equality* $\mathrm{nr}(A(t)B(t)) = \mathrm{nr}(B(t))$ *holds, then the solution* $X(t)$ *to* (2.2) *satisfies* (2.1).

Proof. Using known results about the solvability and general solutions of linear matrix equations from [1,2], we conclude that the first equation in (2.2) is solvable with respect to $Y(t)$ and its general solution is given by

$$Y(t) = (A(t)B(t))^{\dagger} + Z(t) - (A(t)B(t))^{\dagger}A(t)B(t)Z(t),$$

for arbitrary $Z(t)$. The assumption $\mathrm{nr}(A(t)B(t)) = \mathrm{nr}(B(t))$ implies

$$B(t)(A(t)B(t))^{\dagger}A(t)B(t) = B(t)$$

and further

$$X(t) = B(t)Y(t) = B(t)(A(t)B(t))^{\dagger} \in A(t)\{2,3\}.$$

As a result

$$B(t)B^{\dagger}(t)X(t) = B(t)B^{\dagger}(t)B(t)(A(t)B(t))^{\dagger} = B(t)(A(t)B(t))^{\dagger},$$

which confirms that $X(t)$ from (2.2) satisfies the first equation in (2.1). □

As a result, we set the following group of EMEs (GEMEs):

$$\begin{cases} E_1(t) = (A(t)B(t))^* A(t)B(t)Y(t) - (A(t)B(t))^*, \\ E_2(t) = X(t) - B(t)Y(t). \end{cases} \tag{2.3}$$

The first time derivatives of involved EMEs are equal to

$$\begin{cases} \dot{E}_1(t) = \big(D^*(t)A(t)B(t) + (A(t)B(t))^* D(t)\big)Y(t) - D^*(t) + (A(t)B(t))^* A(t)B(t)\dot{Y}(t), \\ \dot{E}_2(t) = \dot{X}(t) - \dot{B}(t)Y(t) - B(t)\dot{Y}(t), \end{cases} \tag{2.4}$$

where $D(t) = A(t)\dot{B}(t) + \dot{A}(t)B(t)$. The next result is achieved by replacing $\dot{E}_1(t)$ and $\dot{E}_2(t)$ of (2.4) with $\dot{E}(t)$ into (1.6):

$$\begin{cases} (A(t)B(t))^* A(t)B(t)\dot{Y}(t) = -\lambda E_1(t) - \big(D^*(t)A(t)B(t) + (A(t)B(t))^* D(t)\big)Y(t) + D^*(t), \\ \dot{X}(t) - B(t)\dot{Y}(t) = -\lambda E_2(t) + \dot{B}(t)Y(t). \end{cases} \tag{2.5}$$

To simplify the dynamics of (2.5), the vectorization and Kronecker product are utilized:

$$\begin{cases} \big(I_m \otimes ((A(t)B(t))^* A(t)B(t))\big)\mathrm{vec}(\dot{Y}(t)) = \mathrm{vec}\big(-\lambda E_1(t) - \big(D^*(t)A(t)B(t) \\ \qquad\qquad +(A(t)B(t))^* D(t)\big)Y(t) + D^*(t)\big), \\ \mathrm{vec}(\dot{X}(t)) - (I_m \otimes B(t))\mathrm{vec}(\dot{Y}(t)) = \mathrm{vec}\big(-\lambda E_2(t) + \dot{B}(t)Y(t)\big). \end{cases} \tag{2.6}$$

Then, setting

$$\begin{aligned} P(t) &= \begin{bmatrix} I_m \otimes ((A(t)B(t))^* A(t)B(t)) & \mathbf{0}_{km\times mn} \\ -I_m \otimes B(t) & I_{mn} \end{bmatrix}, \\ \mathbf{P}_R(t) &= \begin{cases} P(t), & \mathrm{nr}(B(t)) = \min\{n,k\} \text{ and } \mathrm{nr}(A(t)) = \min\{n,m\} \\ P(t) + \beta I_{m(k+n)}, & \mathrm{nr}(B(t)) < \min\{n,k\} \text{ or } \mathrm{nr}(A(t)) < \min\{n,m\} \end{cases} \\ p(t) &= \mathrm{vec}\big(-\lambda E_1(t) - \big(D^*(t)A(t)B(t) + (A(t)B(t))^* D(t)\big)Y(t) + D^*(t)\big), \\ \mathbf{p}_R(t) &= \begin{bmatrix} p(t) \\ \mathrm{vec}\big(-\lambda E_2(t) + \dot{B}(t)Y(t)\big) \end{bmatrix}, \quad \mathbf{u}_R(t) = \begin{bmatrix} \mathrm{vec}(Y(t)) \\ \mathrm{vec}(X(t)) \end{bmatrix}, \quad \dot{\mathbf{u}}_R(t) = \begin{bmatrix} \mathrm{vec}(\dot{Y}(t)) \\ \mathrm{vec}(\dot{X}(t)) \end{bmatrix}, \end{aligned} \tag{2.7}$$

(2.6) can be converted into

$$\mathbf{P}_R(t)\dot{\mathbf{u}}_R(t) = \mathbf{p}_R(t), \tag{2.8}$$

where $\mathbf{P}_R(t) \in \mathbb{C}^{m(k+n)\times m(k+n)}$ is a nonsingular mass matrix and $\dot{\mathbf{u}}_R(t), \mathbf{u}_R(t), \mathbf{p}_R(t) \in \mathbb{C}^{m(k+n)}$. The ZNN model of (2.8), referred to as ZNNPR1 for convenience, can compute a specific solution of the TV-MROI with PR. Theorem 2.2 certifies global convergence with the exponential speed of the ZNNPR1 model (2.8) to the theoretical solution (TS).

Theorem 2.2. *Let $B(t) \in \mathbb{C}^{n\times k}$ and $A(t) \in \mathbb{C}^{m\times n}$ be differentiable time-varying matrices. The dynamical system (2.8) converges exponentially and globally to the theoretical solution $\mathbf{u}_R^*(t)$ starting from arbitrary initial value $\mathbf{u}_R(0)$. Additionally, the TS of TV-MROI with PR (1.4) is defined by the last mn components of $\mathbf{u}_R^*(t)$.*

Proof. First, the TV-MROI with PR (1.4) is converted into the problem of (2.2). Second, to solve the problem of (2.2), the GEME of (2.3) is declared. Then, for zeroing (2.3), the model (2.5) is deployed in line with the ZNN scheme (1.6). From [22, Theorem 1], $E_1(t)$ and $E_2(t)$ in (2.5) converge exponentially and globally to their TS, $Y^*(t)$ and $X^*(t)$, when $t \to \infty$, starting from any initial value. So, the model (2.5) converges to the TS of the TV-MROI with PR (1.4). Third, the model (2.5) is simplified into the ZNNPR1 model (2.8) using the Kronecker product and vectorization. As an alternative version of (2.5), for every initial value $\mathbf{u}(0)$, the ZNNPR1 model (2.8) converges to the TS $\mathbf{u}_R^*(t)$ when $t \to \infty$. In line with (2.7), the TS of TV-MROI with PR in (1.4) is defined by the last mn components of $\mathbf{u}_R^*(t)$. Thereafter, the proof is finished. □

2.2. ZNN model for computing a random solution of the TV-MROI with PR

Considering the identity $B^*(t)B(t)B^\dagger(t)=B^*(t)$, (2.1) can be converted as below:

$$\begin{cases} B^*(t)B(t)Z(t)=B^*(t),\\ X(t)=B(t)Z(t)X(t),\\ X(t)A(t)B(t)=B(t), \end{cases} \tag{2.9}$$

where the first matrix equation holds in the instance of the pseudoinverse of $B(t)$, i.e. $Z(t)$.

As a result, we set the following GEMEs:

$$\begin{cases} E_1(t)=B^*(t)B(t)Z(t)-B^*(t),\\ E_2(t)=X(t)-B(t)Z(t)X(t),\\ E_3(t)=X(t)A(t)B(t)-B(t). \end{cases} \tag{2.10}$$

The first time derivatives of involved EMEs are equal to

$$\begin{cases} \dot{E}_1(t)=(\dot{B}^*(t)B(t)+B^*(t)\dot{B}(t))Z(t)+B^*(t)B(t)\dot{Z}(t)-\dot{B}^*(t),\\ \dot{E}_2(t)=\dot{X}(t)-\dot{B}(t)Z(t)X(t)-B(t)\dot{Z}(t)X(t)-B(t)Z(t)\dot{X}(t),\\ \dot{E}_3(t)=\dot{X}(t)A(t)B(t)+X(t)\dot{A}(t)B(t)+X(t)A(t)\dot{B}(t)-\dot{B}(t). \end{cases} \tag{2.11}$$

The result that follows is achieved by replacing $\dot{E}_1(t),\dot{E}_2(t)$ and $\dot{E}_3(t)$ of (2.11) with $\dot{E}(t)$ into (1.6):

$$\begin{cases} B^*(t)B(t)\dot{Z}(t)=-\lambda E_1(t)-(\dot{B}^*(t)B(t)+B^*(t)\dot{B}(t))Z(t)+\dot{B}^*(t),\\ \dot{X}(t)-B(t)\dot{Z}(t)X(t)-B(t)Z(t)\dot{X}(t)=-\lambda E_2(t)+\dot{B}(t)Z(t)X(t),\\ \dot{X}(t)A(t)B(t)=-\lambda E_3(t)-X(t)\dot{A}(t)B(t)-X(t)A(t)\dot{B}(t)+\dot{B}(t). \end{cases} \tag{2.12}$$

To simplify the dynamics of (2.12), the vectorization and Kronecker product are utilized:

$$\begin{cases} \big(I_n\otimes(B^*(t)B(t))\big)\mathrm{vec}(\dot{Z}(t))=\mathrm{vec}\big(-\lambda E_1(t)-(B^*(t)\dot{B}(t)+\dot{B}^*(t)B(t))Z(t)+\dot{B}^*(t)\big),\\ \big(I_{mn}-(I_m\otimes(B(t)Z(t)))\big)\mathrm{vec}(\dot{X}(t))-\big(X^*(t)\otimes B(t)\big)\mathrm{vec}(\dot{Z}(t))=\mathrm{vec}\big(-\lambda E_2(t)+\dot{B}(t)Z(t)X(t)\big),\\ ((A(t)B(t))^*\otimes I_n)\mathrm{vec}(\dot{X}(t))=\mathrm{vec}\big(-\lambda E_3(t)-X(t)A(t)\dot{B}(t)-X(t)\dot{A}(t)B(t)+\dot{B}(t)\big) \end{cases} \tag{2.13}$$

Then, setting

$$\begin{aligned} W(t)&=\begin{bmatrix} I_n\otimes(B^*(t)B(t)) & \mathbf{0}_{kn\times mn}\\ -X^*(t)\otimes B(t) & I_{mn}-(I_m\otimes(B(t)Z(t)))\\ \mathbf{0}_{kn} & (A(t)B(t))^*\otimes I_n \end{bmatrix},\\ \mathbf{W}_R(t)&=\begin{cases} W^*(t)W(t), & \mathrm{nr}(B(t))=\min\{n,k\}\ \text{and}\ \mathrm{nr}(A(t))=\min\{n,m\}\\ W^*(t)W(t)+\beta I_{n(k+m)}, & \mathrm{nr}(B(t))<\min\{n,k\}\ \text{or}\ \mathrm{nr}(A(t))<\min\{n,m\} \end{cases}\\ w(t)&=\begin{bmatrix} \mathrm{vec}\big(-\lambda E_1(t)-(B^*(t)\dot{B}(t)+\dot{B}^*(t)B(t))Z(t)+\dot{B}^*(t)\big)\\ \mathrm{vec}\big(-\lambda E_2(t)+\dot{B}(t)Z(t)X(t)\big)\\ \mathrm{vec}\big(-\lambda E_3(t)-X(t)A(t)\dot{B}(t)-X(t)\dot{A}(t)B(t)+\dot{B}(t)\big) \end{bmatrix},\\ \mathbf{w}_R(t)&=W^*(t)w(t),\quad \mathbf{v}_R(t)=\begin{bmatrix}\mathrm{vec}(Z(t))\\ \mathrm{vec}(X(t))\end{bmatrix},\quad \dot{\mathbf{v}}_R(t)=\begin{bmatrix}\mathrm{vec}(\dot{Z}(t))\\ \mathrm{vec}(\dot{X}(t))\end{bmatrix}, \end{aligned} \tag{2.14}$$

(2.13) can be converted as follows:

$$\mathbf{W}_R(t)\dot{\mathbf{v}}_R(t)=W_R^*(t)\mathbf{w}_R(t), \tag{2.15}$$

where $\mathbf{W}_R(t)\in\mathbb{C}^{n(k+m)\times n(k+m)}$ is a state-dependent nonsingular mass matrix, $W(t)\in\mathbb{C}^{n(2k+m)\times n(k+m)}$, $w(t)\in\mathbb{C}^{n(2k+m)}$, $\mathbf{w}_R(t)\in\mathbb{C}^{n(k+m)}$ and $\dot{\mathbf{v}}_R(t),\mathbf{v}_R(t)\in\mathbb{C}^{n(k+m)}$. The direct ZNN model of (2.15), referred to as ZNNPR2 for convenience, can compute a random solution of the TV-MROI with PR. Theorem 2.3 certifies universal convergence with exponential speed of the ZNNPR2 model (2.15) to the TS.

Theorem 2.3. *Let $B(t)\in\mathbb{C}^{n\times k}$ and $A(t)\in\mathbb{C}^{m\times n}$ be differentiable time-varying matrices. The dynamical system (2.15) converges exponentially and globally to the TS, $\mathbf{v}_R^*(t)$ of TV-MROI with PR (1.4), starting from arbitrary initial value $\mathbf{v}_R(0)$. Additionally, the TS of TV-MROI with PR (1.4) is the last mn elements of $\mathbf{v}_R^*(t)$.*

Proof. First, the TV-MROI with PR (1.4) is converted into the problem of (2.2). Second, to solve the problem of (2.2), the GEME of (2.3) is declared. Then, for zeroing (2.10), the model (2.12) is deployed in line with the ZNN theme (1.6). From [22, Theorem 1], $E_1(t)$ and $E_2(t)$ in (2.12) converge exponentially and globally to their TS, $Z^*(t)$ and $X^*(t)$, when $t\to\infty$, starting from any initial value.

So, the model (2.12) converges to the TS of the TV-MROI with PR (1.4). Third, the model (2.12) is simplified into the ZNNPR1 model (2.15) using the Kronecker product and vectorization. As an alternative version of (2.12), for every initial value $\mathbf{v}(0)$, the ZNNPR2 model (2.15) converges to the TS $\mathbf{v}_R^*(t)$ when $t \to \infty$. In line with (2.14), the TS of TV-MROI with PR in (1.4) is the last mn elements of $\mathbf{v}_R^*(t)$. Thereafter, the proof is finished. □

3. ZNN designs for computing TV-MROI with PK

Two ZNN models for computing TV-MROI with PK are described in this section, one for a specific solution and another for a random one. Additionally, we consider that $A(t) \in \mathbb{C}^{m\times n}$ and $Q(t) \in \mathbb{C}^{l\times m}$ are differentiable time-varying matrices.

According to Propositions 1.2 and 1.3, the following equation group must be satisfied in order to have a TV-MROI with PK:

$$\begin{cases} X(t) = X(t)Q^{\dagger}(t)Q(t), \\ Q(t)A(t)X(t) = Q(t), \end{cases} \tag{3.1}$$

where $X(t) \in \mathbb{C}^{n\times m}$ is the unknown TV-MROI with PK.

3.1. ZNN model for computing a specific solution of the TV-MROI with PK

Considering the minimal-norm least-square solution $(Q(t)A(t))^{\dagger}Q(t) = X(t)$ and the equation $(Q(t)A(t))^* = (Q(t)A(t))^{\dagger}Q(t)A(t)(Q(t)A(t))^*$ which holds, we set the unknown $Y(t) = (Q(t)A(t))^{\dagger} \in \mathbb{C}^{n\times l}$ so that (3.1) can be converted as

$$\begin{cases} Y(t)Q(t)A(t)(Q(t)A(t))^* = (Q(t)A(t))^*, \\ X(t) = Y(t)Q(t), \end{cases} \tag{3.2}$$

where the first matrix equation holds in the case $(Q(t)A(t))^{\dagger} = Y(t)$. It is significant to mention that (3.2) always computes the specific TV-MROI with PR in (3.1) that is based on the pseudoinverse of the product $Q(t)A(t)$. Details are given in Theorem 3.1.

Theorem 3.1. *If the rank equality* $\mathrm{nr}(Q(t)A(t)) = \mathrm{nr}(A(t))$ *holds, then the solution to* (3.2) *satisfies* (3.1).

Proof. Following dual principles from Theorem 2.1, it is derived

$$X(t) = (Q(t)A(t))^{\dagger}Q(t) \in A(t)\{2,4\}.$$

Therefore,

$$X(t)Q^{\dagger}(t)Q(t) = (Q(t)A(t))^{\dagger}Q(t)Q^{\dagger}(t)Q(t) = (Q(t)A(t))^{\dagger}Q(t),$$

which confirms that $X(t)$ from (3.2) satisfies the first equation in (3.1). □

Following the obtained theoretical results from Theorem 3.1, we set the following GEMEs:

$$\begin{cases} E_1(t) = Y(t)Q(t)A(t)(Q(t)A(t))^* - (Q(t)A(t))^*, \\ E_2(t) = X(t) - Y(t)Q(t), \end{cases} \tag{3.3}$$

whose first time derivatives are equal to

$$\begin{cases} \dot{E}_1(t) = Y(t)\big(D(t)(Q(t)A(t))^* + Q(t)A(t)D^*(t)\big) + \dot{Y}(t)Q(t)A(t)(Q(t)A(t))^* - D^*(t), \\ \dot{E}_2(t) = \dot{X}(t) - Y(t)\dot{Q}(t) - \dot{Y}(t)Q(t), \end{cases} \tag{3.4}$$

where $D(t) = Q(t)\dot{A}(t) + \dot{Q}(t)A(t)$. The next result is achieved by replacing $\dot{E}_1(t)$ and $\dot{E}_2(t)$ of (3.4) with $\dot{E}(t)$ into (1.6):

$$\begin{cases} \dot{Y}(t)Q(t)A(t)(Q(t)A(t))^* = -\lambda E_1(t) - Y(t)\big(D(t)(Q(t)A(t))^* + Q(t)A(t)D^*(t)\big) + D^*(t), \\ \dot{X}(t) - \dot{Y}(t)Q(t) = Y(t)\dot{Q}(t) - \lambda E_2(t). \end{cases} \tag{3.5}$$

To simplify the dynamics of (3.5), the vectorization and Kronecker product are utilized:

$$\begin{cases} \big((Q(t)A(t)(Q(t)A(t))^*) \otimes I_n\big)\mathrm{vec}(\dot{Y}(t)) = \mathrm{vec}\big(-\lambda E_1(t) - Y(t)\big(D(t)(Q(t)A(t))^* \\ \qquad\qquad +Q(t)A(t)D^*(t)\big) + D^*(t)\big), \\ \mathrm{vec}(\dot{X}(t)) - (Q^*(t) \otimes I_n)\mathrm{vec}(\dot{Y}(t)) = \mathrm{vec}\big(-\lambda E_2(t) + Y(t)\dot{Q}(t)\big). \end{cases} \tag{3.6}$$

Then, setting

$$\begin{aligned}
&P_K(t)=\begin{bmatrix}(Q(t)A(t)(Q(t)A(t))^*)\otimes I_n & \mathbf{0}_{ln\times mn}\\ -Q^*(t)\otimes I_n & I_{mn}\end{bmatrix},\\
&p(t)=\mathrm{vec}\big(-\lambda E_1(t)-Y(t)\big(D(t)(Q(t)A(t))^*+Q(t)A(t)D^*(t)\big)+D^*(t)\big),\\
&\mathbf{P}_K(t)=\begin{cases}P_K(t), & \mathrm{nr}(Q(t))=\min\{l,m\}\text{ and }\mathrm{nr}(A(t))=\min\{n,m\}\\ P_K(t)+\beta I_{n(l+m)}, & \mathrm{nr}(Q(t))<\min\{l,m\}\text{ or }\mathrm{nr}(A(t))<\min\{n,m\}\end{cases}\\
&\mathbf{u}_K(t)=\begin{bmatrix}\mathrm{vec}(Y(t))\\ \mathrm{vec}(X(t))\end{bmatrix},\quad \mathbf{p}_K(t)=\begin{bmatrix}p(t)\\ \mathrm{vec}\big(-\lambda E_2(t)+Y(t)\dot{Q}(t)\big)\end{bmatrix},\quad \dot{\mathbf{u}}_K(t)=\begin{bmatrix}\mathrm{vec}(\dot{Y}(t))\\ \mathrm{vec}(\dot{X}(t))\end{bmatrix},
\end{aligned}\tag{3.7}$$

(3.6) can be converted as follows:

$$\mathbf{P}_K(t)\dot{\mathbf{u}}_K(t)=\mathbf{p}_K(t),\tag{3.8}$$

where $\mathbf{P}_K(t)\in\mathbb{C}^{n(l+m)\times n(l+m)}$ is a nonsingular mass matrix and $\dot{\mathbf{u}}_K(t),\mathbf{u}_K(t),\mathbf{p}_K(t)\in\mathbb{C}^{n(l+m)}$. The implicit ZNN model (3.8), referred to as ZNNPK1 for convenience, can compute a specific TV-MROI with PK. Theorem 3.2 certifies global convergence with exponential speed of the ZNNPK1 model (3.8) to the TS.

Theorem 3.2. *Let $Q(t)\in\mathbb{C}^{l\times m}$ and $A(t)\in\mathbb{C}^{m\times n}$ be differentiable time-varying matrices. The dynamical system (3.8) converges exponentially and globally to the theoretical solution $u^*(t)$ starting from arbitrary initial value $\mathbf{u}_K(0)$. Additionally, the TS of TV-MROI with PK (1.5) is defined by the last mn components of $\mathbf{u}_K^*(t)$.*

Proof. First, the TV-MROI with PK (1.5) is converted into the problem of (3.2). Second, to solve the problem of (3.2), the GEME of (3.3) is declared. Then, for zeroing (3.3), the model (3.5) is deployed in line with the ZNN theme (1.6). From [22, Theorem 1], $E_1(t)$ and $E_2(t)$ in (3.5) converge exponentially and globally to their TS, $Y^*(t)$ and $X^*(t)$, when $t\to\infty$, starting from any initial value. So, the model (3.5) converges to the TS of the TV-MROI with PK (1.5). Third, the model (3.5) is simplified into the ZNNPK1 model (3.8) using the Kronecker product and vectorization. As an alternative version of (3.5), for every initial value $\mathbf{u}(0)$, the ZNNPK1 model (3.8) converges to the TS $\mathbf{u}_K^*(t)$ when $t\to\infty$. In line with (3.7), the TS of TV-MROI with PK in (1.5) is the last mn components of $\mathbf{u}_K^*(t)$. Thereafter, the proof is finished. □

3.2. ZNN model for computing a random solution of the TV-MROI with PK

Considering the identity $Q^\dagger(t)Q(t)Q^*(t)=Q^*(t)$, (3.1) can be converted as below:

$$\begin{cases}Q^*(t)=Z(t)Q(t)Q^*(t),\\ X(t)Z(t)Q(t)=X(t),\\ Q(t)A(t)X(t)=Q(t).\end{cases}\tag{3.9}$$

As a result, we set the following GEME:

$$\begin{cases}E_1(t)=Z(t)Q(t)Q^*(t)-Q^*(t),\\ E_2(t)=X(t)-X(t)Z(t)Q(t),\\ E_3(t)=Q(t)A(t)X(t)-Q(t),\end{cases}\tag{3.10}$$

where its first time derivatives are

$$\begin{cases}\dot{E}_1(t)=\ Z(t)\dot{Q}(t)Q^*(t)+\dot{Z}(t)Q(t)Q^*(t)+Z(t)Q(t)\dot{Q}^*(t)-\dot{Q}^*(t),\\ \dot{E}_2(t)=\ \dot{X}(t)-\dot{X}(t)Z(t)Q(t)-X(t)\dot{Z}(t)Q(t)-X(t)Z(t)\dot{Q}(t),\\ \dot{E}_3(t)=\ \dot{Q}(t)A(t)X(t)+Q(t)A(t)\dot{X}(t)+Q(t)\dot{A}(t)X(t)-\dot{Q}(t).\end{cases}\tag{3.11}$$

The next result is achieved by replacing $\dot{E}_1(t),\dot{E}_2(t)$ and $\dot{E}_3(t)$ of (3.11) with $\dot{E}(t)$ into (1.6):

$$\begin{cases}\dot{Z}(t)Q(t)Q^*(t)=-\lambda E_1(t)-Z(t)\dot{Q}(t)Q^*(t)-Z(t)Q(t)\dot{Q}^*(t)+\dot{Q}^*(t),\\ \dot{X}(t)-\dot{X}(t)Z(t)Q(t)-X(t)\dot{Z}(t)Q(t)=-\lambda E_2(t)+X(t)Z(t)\dot{Q}(t),\\ Q(t)A(t)\dot{X}(t)=-\lambda E_3(t)-\dot{Q}(t)A(t)X(t)-Q(t)\dot{A}(t)X(t)+\dot{Q}(t).\end{cases}\tag{3.12}$$

To simplify the dynamics of (3.12), the vectorization and Kronecker product are utilized:

$$\begin{cases}\big((Q(t)Q^*(t))\otimes I_m\big)\mathrm{vec}(\dot{Z}(t))=\mathrm{vec}\big(-\lambda E_1(t)-Z(t)\dot{Q}(t)Q^*(t)-Z(t)Q(t)\dot{Q}^*(t)+\dot{Q}^*(t)\big),\\ \big(I_{mn}-(Z(t)Q(t))^*\otimes I_n\big)\mathrm{vec}(\dot{X}(t))-\big(Q^*(t)\otimes X(t)\big)\mathrm{vec}(\dot{Z}(t))=\mathrm{vec}\big(-\lambda E_2(t)+X(t)Z(t)\dot{Q}(t)\big),\\ \big(I_m\otimes(Q(t)A(t))\big)\mathrm{vec}(\dot{X}(t))=\mathrm{vec}\big(-\lambda E_3(t)-\dot{Q}(t)A(t)X(t)-Q(t)\dot{A}(t)X(t)+\dot{Q}(t)\big).\end{cases}\tag{3.13}$$

Then, setting

$$\begin{aligned} W(t) &= \begin{bmatrix} (Q(t)Q^*(t))\otimes I_m & \mathbf{0}_{lm,mn} \\ Q^*(t)\otimes X(t) & I_{mn}-(Z(t)Q(t))^*\otimes I_n \\ \mathbf{0}_{lm} & I_m\otimes(Q(t)A(t)) \end{bmatrix}, \\ \mathbf{W}_K(t) &= \begin{cases} W^*(t)W(t), & \mathrm{nr}(Q(t))=\min\{m,l\} \text{ and } \mathrm{nr}(A(t))=\min\{n,m\} \\ W^*(t)W(t)+\beta I_{m(l+n)}, & \mathrm{nr}(Q(t))<\min\{m,l\} \text{ or } \mathrm{nr}(A(t))<\min\{n,m\} \end{cases} \\ w(t) &= \begin{bmatrix} \mathrm{vec}\big(-\lambda E_1(t)-Z(t)Q(t)\dot{Q}^*(t)-Z(t)\dot{Q}(t)Q^*(t)+\dot{Q}^*(t)\big) \\ \mathrm{vec}\big(-\lambda E_2(t)+X(t)Z(t)\dot{Q}(t)\big) \\ \mathrm{vec}\big(-\lambda E_3(t)-Q(t)\dot{A}(t)X(t)-\dot{Q}(t)A(t)X(t)+\dot{Q}(t)\big) \end{bmatrix}, \\ \mathbf{w}_K(t) &= W^*(t)w(t), \quad \mathbf{v}_K(t)=\begin{bmatrix}\mathrm{vec}(Z(t))\\ \mathrm{vec}(X(t))\end{bmatrix}, \quad \dot{\mathbf{v}}_K(t)=\begin{bmatrix}\mathrm{vec}(\dot{Z}(t))\\ \mathrm{vec}(\dot{X}(t))\end{bmatrix}, \end{aligned} \tag{3.14}$$

(3.13) can be converted as follows:

$$\mathbf{W}_K(t)\dot{\mathbf{v}}_K(t)=\mathbf{v}_K(t), \tag{3.15}$$

where $\mathbf{W}_K(t)\in\mathbb{C}^{m(l+n)\times m(l+n)}$ is a state-dependent nonsingular mass matrix, $W(t)\in\mathbb{C}^{m(2l+n)\times m(l+n)}$, $w(t)\in\mathbb{C}^{m(2l+n)}$, $\mathbf{w}_K(t)\in\mathbb{C}^{m(l+n)}$ and $\dot{\mathbf{v}}_K(t),\mathbf{v}_K(t)\in\mathbb{C}^{m(l+n)}$. The direct ZNN model of (3.15), referred to as ZNNPK2 for convenience, can compute a random solution of the TV-MROI with PK. Theorem 3.3 certifies universal convergence with exponential speed of the ZNNPK2 model (3.15) to the TS.

Theorem 3.3. *Let $Q(t)\in\mathbb{C}^{l\times m}$ and $A(t)\in\mathbb{C}^{m\times n}$ be differentiable time-varying matrices. The dynamical system (3.15) converges exponentially and globally to the TS, $v^*(t)$ of TV-MROI with PK (1.5), starting from arbitrary initial value $\mathbf{v}_K(0)$. Additionally, the TS of TV-MROI with PK (1.5) is defined by the last mn components of $\mathbf{v}_K^*(t)$.*

Proof. First, the TV-MROI with PK (1.5) is converted into the problem of (3.2). Second, to solve the problem of (3.2), the GEME of the form (3.3) is declared. Then, for zeroing (3.10), the model (3.12) is deployed in line with the ZNN theme (1.6). From [22, Theorem 1], $E_1(t)$ and $E_2(t)$ in (3.12) converge exponentially and globally to their TS, $Z^*(t)$ and $X^*(t)$, when $t\to\infty$, starting from any initial value. So, the model (3.12) converges to the TS of the TV-MROI with PK (1.5). Third, the model (3.12) is simplified into the ZNNPK1 model (3.15) using the Kronecker product and vectorization. As an alternative version of (3.12), for every initial value $\mathbf{v}(0)$, the ZNNPK2 model (3.15) converges to the TS $\mathbf{v}_K^*(t)$ when $t\to\infty$. In line with (3.14), the TS of TV-MROI with PK in (1.5) is the last mn components of $\mathbf{v}_K^*(t)$. Thereafter, the proof is finished. □

4. ZNN models computational complexity

The complexity of creating and addressing (2.8), (2.15), (3.8) and (3.15) adds total computational complexity to the ZNNPR1, ZNNPR2, ZNNPK1 and ZNNPK2 models, respectively. Particularly, the computational complexity of creating (2.8) is $\mathcal{O}((m(k+n))^2)$ operations because at every iteration we conduct $(m(k+n))^2$ multiplications and $m(k+n)$ addition/subtraction operations. For the same reasons, creating (2.15), (3.8) and (3.15) requires $\mathcal{O}((n(k+m))^2)$, $\mathcal{O}(n(l+m))^2)$ and $\mathcal{O}((m(l+n))^2)$ operations, respectively, which add to their computational complexity.

On top of that, the implicit MATLAB solver `ode15s` is used to address at each step the linear system of equations. The complexity of addressing (2.8) increases to $\mathcal{O}((m(k+n))^3$ since it involves an $(m(k+n))\times(m(k+n))$ matrix. In the same manner, the complexity of addressing (2.15), (3.8) and (3.15) with an implicit solver increases to $\mathcal{O}((n(k+m))^3)$, $\mathcal{O}((n(l+m))^3)$ and $\mathcal{O}((m(l+n))^3)$, respectively. So, the ZNNPR1, ZNNPR2, ZNNPK1 and ZNNPK2 models' total computational complexity is $\mathcal{O}((m(k+n))^3$, $\mathcal{O}((n(k+m))^3)$, $\mathcal{O}((n(l+m))^3)$ and $\mathcal{O}((m(l+n))^3)$, respectively.

5. Numerical experiments

This section investigates the performance of the ZNNPR1 (2.8), ZNNPR2 (2.15), ZNNPK1 (3.8) and ZNNPK2 (3.15) models in six numerical examples (NEs). Particularly, three NEs involve computing the TV-MROI with PR in (1.4), and three NEs involve computing the TV-MROI with PK in (1.5). The initial values, i.e. IC1, in all NEs have been set to $\mathbf{u}_R(0)=\mathbf{0}_{m(k+n)}$, $\mathbf{v}_R(0)=\mathbf{0}_{n(k+m)}$, $\mathbf{u}_K(0)=\mathbf{0}_{n(l+m)}$ and $\mathbf{v}_K(0)=\mathbf{0}_{m(l+n)}$. Notice that a second set of initial values, i.e. IC2, have also been used in NEs 5.1 and 5.2, which are $\mathbf{u}_R(0)=\mathbf{1}_{m(k+n)}$, $\mathbf{v}_R(0)=\mathbf{1}_{n(k+m)}$, $\mathbf{u}_K(0)=\mathbf{1}_{n(l+m)}$ and $\mathbf{v}_K(0)=\mathbf{1}_{m(l+n)}$. For the calculations in all NEs, the `ode45` solver of MATLAB is utilized with the time-interval $[0,10]$ under the default relative error tolerance (i.e. 10^{-5}), while the regularization parameters have been set to $\beta=10^{-8}$. Additionally, the ZNN parameter has been set to $\lambda=10$ in all NEs. However, the ZNN parameter has been set to $\lambda=10$ and $\lambda=100$ in NEs 5.3 and 5.4. It is important to mention that the additional errors considered to measure and compare the performance of the models are the following:

(MP-ii) : $\|X(t)A(t)X(t)-X(t)\|_F$ for all four models,
(PR-i) : $\|X(t)A(t)B(t)-B(t)\|_F$ for the ZNNPR1 and ZNNPR2 models,
(PR-ii) : $\left\|X(t)-B(t)B^{\dagger}(t)X(t)\right\|_F$ for the ZNNPR1 and ZNNPR2 models,

(PK-i) : $\|Q(t)-Q(t)A(t)X(t)\|_F$ for the ZNNPK1 and ZNNPK2 models,
(PK-ii) : $\left\|X(t)-X(t)Q^{\dagger}(t)Q(t)\right\|_F$ for the ZNNPK1 and ZNNPK2 models.

For convenience purposes, we have set $\alpha(t)=\sin(t)$ and $\gamma(t)=\cos(t)$. Finally, the specific TS (STS) depicted in the figures of this section corresponds to $X(t)=B(t)(A(t)B(t))^{\dagger}$ in the case of the TV-MROI with PR, and $X(t)=(Q(t)A(t))^{\dagger}Q(t)$ in the case of the TV-MROI with PK.

5.1. Numerical example 1

Consider the following matrices:

$$K_1(t)=\begin{bmatrix} 6-\gamma(\pi t) & 5-\gamma(\pi t) & 12 \\ -3-\gamma(\pi t) & -8-\gamma(\pi t) & -3 \\ 1-\gamma(\pi t) & 2-\gamma(\pi t) & 5 \\ 1-\gamma(\pi t) & 2-\gamma(\pi t) & 5 \end{bmatrix},$$

$$K_2(t)=[1+\alpha(t),4+2\alpha(t),3+\gamma(t)]^{\mathrm{T}}\odot\mathbf{1}_{3\times 2}.$$

In this example, we set the input matrices $A(t)=K_1(t)+2K_1(t)\imath\in\mathbb{C}^{4\times 3}$ with $\mathrm{nr}(A(t))=3$ and $B(t)=K_2(t)+1/2K_2(t)\imath\in\mathbb{C}^{3\times 2}$ with $\mathrm{nr}(B(t))=1$ to calculate the TV-MROI with PR. As a result, $\mathrm{nr}(A(t)B(t))=\mathrm{nr}(B(t))=1$ and the TV-MROI is $X(t)\in\mathbb{C}^{3\times 4}$ with $\mathrm{nr}(X(t))=1$. The outcomes of the ZNNPR1 and ZNNPR2 models are shown in Figs. 1 and 3.

5.2. Numerical example 2

Consider the matrix $K_1(t)$ of NE 5.1 and the following matrices:

$$L_1(t)=K_1^{\mathrm{T}}(t),$$

$$L_2(t)=[\alpha(t)+1,2+2\alpha(t),4+\gamma(t)]\odot\mathbf{1}_2.$$

In this example, we set the input matrices $A(t)=L_1(t)+2L_1(t)\imath\in\mathbb{C}^{3\times 4}$ with $\mathrm{nr}(A(t))=3$ and $Q(t)=L_2(t)+1/2L_2(t)\imath\in\mathbb{C}^{2\times 3}$ with $\mathrm{nr}(Q(t))=1$ to calculate the TV-MROI with PK. As a result, $\mathrm{nr}(Q(t)A(t))=\mathrm{nr}(Q(t))=1$ and the TV-MROI is $X(t)\in\mathbb{C}^{4\times 3}$ with $\mathrm{nr}(X(t))=1$. The outcomes of the ZNNPK1 and ZNNPK2 models are shown in Figs. 1 and 3.

5.3. Numerical example 3

Consider the following matrices:

$$K_1(t)=\begin{bmatrix} 6-\gamma(\pi t) & 5-\gamma(\pi t) & 12 & 1 & 5 \\ -3-\gamma(\pi t) & -8-\gamma(\pi t) & -3 & 0 & 5 \\ 1-\gamma(\pi t) & 2-\gamma(\pi t) & 5 & 1 & 7 \\ 1-\gamma(\pi t) & 2-\gamma(\pi t) & 5 & 1 & 9 \end{bmatrix},$$

$$K_2(t)=[2K_1(:,1)(t),5K_1(:,2)(t),4K_1(:,3)(t),3K_1(:,2)(t),6K_1(:,1)(t)]^{\mathrm{T}}.$$

In this example, we set the input matrices $A(t)=K_1(t)+2K_1(t)\imath\in\mathbb{C}^{4\times 5}$ with $\mathrm{nr}(A(t))=4$ and $B(t)=K_2(t)+1/2K_2(t)\imath\in\mathbb{C}^{5\times 4}$ with $\mathrm{nr}(B(t))=3$ to calculate the TV-MROI with PR. As a result, $\mathrm{nr}(A(t)B(t))=\mathrm{nr}(B(t))=3$ and the TV-MROI is $X(t)\in\mathbb{C}^{5\times 4}$ with $\mathrm{nr}(X(t))=3$. The outcomes of the ZNNPR1 and ZNNPR2 models are shown in Figs. 2 and 3.

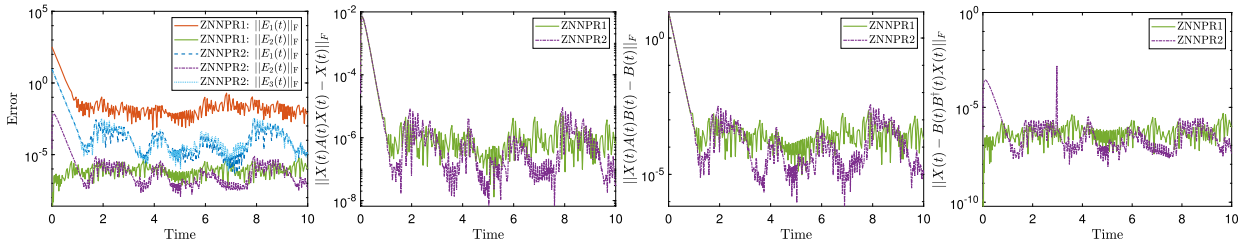
5.4. Numerical example 4

Consider the matrix $K_1(t)$ of NE 5.3 and the following matrices:

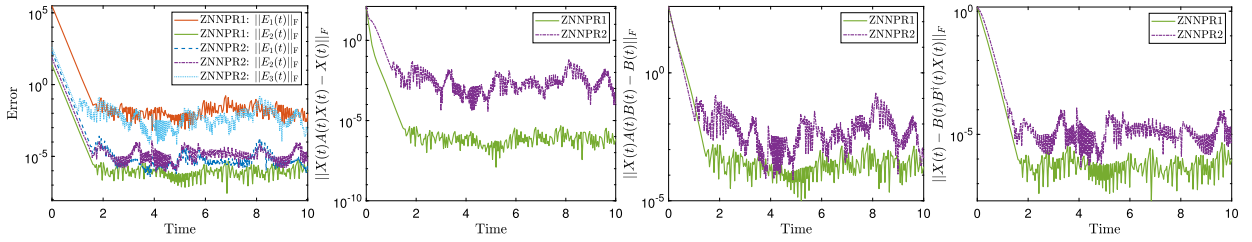
$$L_1(t)=K_1^{\mathrm{T}}(t),$$

$$L_2(t)=[(2L_1(1,:)(t))^{\mathrm{T}},(3L_1(2,:)(t))^{\mathrm{T}},(4L_1(3,:)(t))^{\mathrm{T}},(5L_1(2,:)(t))^{\mathrm{T}},(6L_1(1,:)(t))^{\mathrm{T}}].$$

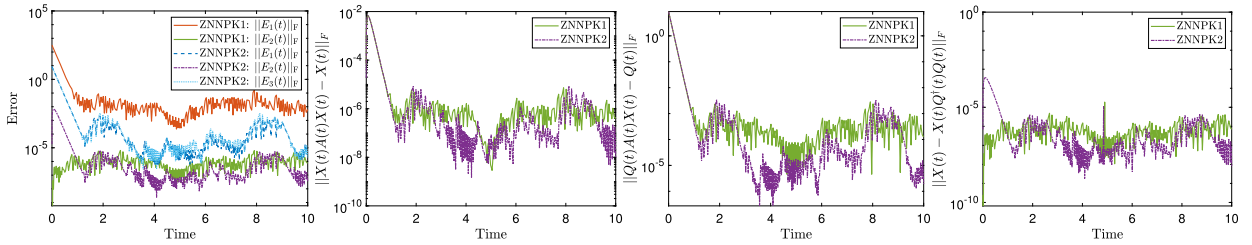
In this example, we set the input matrices $A(t)=L_1(t)+2L_1(t)\imath\in\mathbb{C}^{5\times 4}$ with $\mathrm{nr}(A(t))=4$ and $Q(t)=L_2(t)+1/2L_2(t)\imath\in\mathbb{C}^{4\times 5}$ with $\mathrm{nr}(Q(t))=3$ to calculate the TV-MROI with PK. As a result, $\mathrm{nr}(Q(t)A(t))=\mathrm{nr}(Q(t))=3$ and the TV-MROI is $X(t)\in\mathbb{C}^{4\times 5}$ with $\mathrm{nr}(X(t))=3$. The outcomes of the ZNNPK1 and ZNNPK2 models are shown in Figs. 2 and 3.



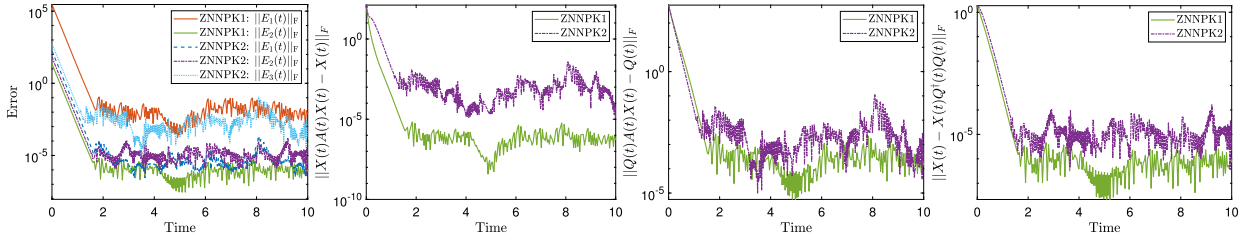
(a) NE 5.1: GEMEs error with IC1. (b) NE 5.1: (MP-ii) error with IC1. (c) NE 5.1: (PR-i) error with IC1. (d) NE 5.1: (PR-ii) error with IC1.



(e) NE 5.1: GEMEs error with IC2. (f) NE 5.1: (MP-ii) error with IC2. (g) NE 5.1: (PR-i) error with IC2. (h) NE 5.1: (PR-ii) error with IC2.



(i) NE 5.2: GEMEs error with IC1. (j) NE 5.2: (MP-ii) error with IC1. (k) NE 5.2: (PK-i) error with IC1. (l) NE 5.2: (PK-ii) error with IC1.



(m) NE 5.2: GEMEs error with IC2. (n) NE 5.2: (MP-ii) error with IC2. (o) NE 5.2: (PK-i) error with IC2. (p) NE 5.2: (PK-ii) error with IC2.

Fig. 1. Errors of the ZNNPR1, ZNNPR2, ZNNPK1 and ZNNPK2 models in NEs 5.1 and 5.2.

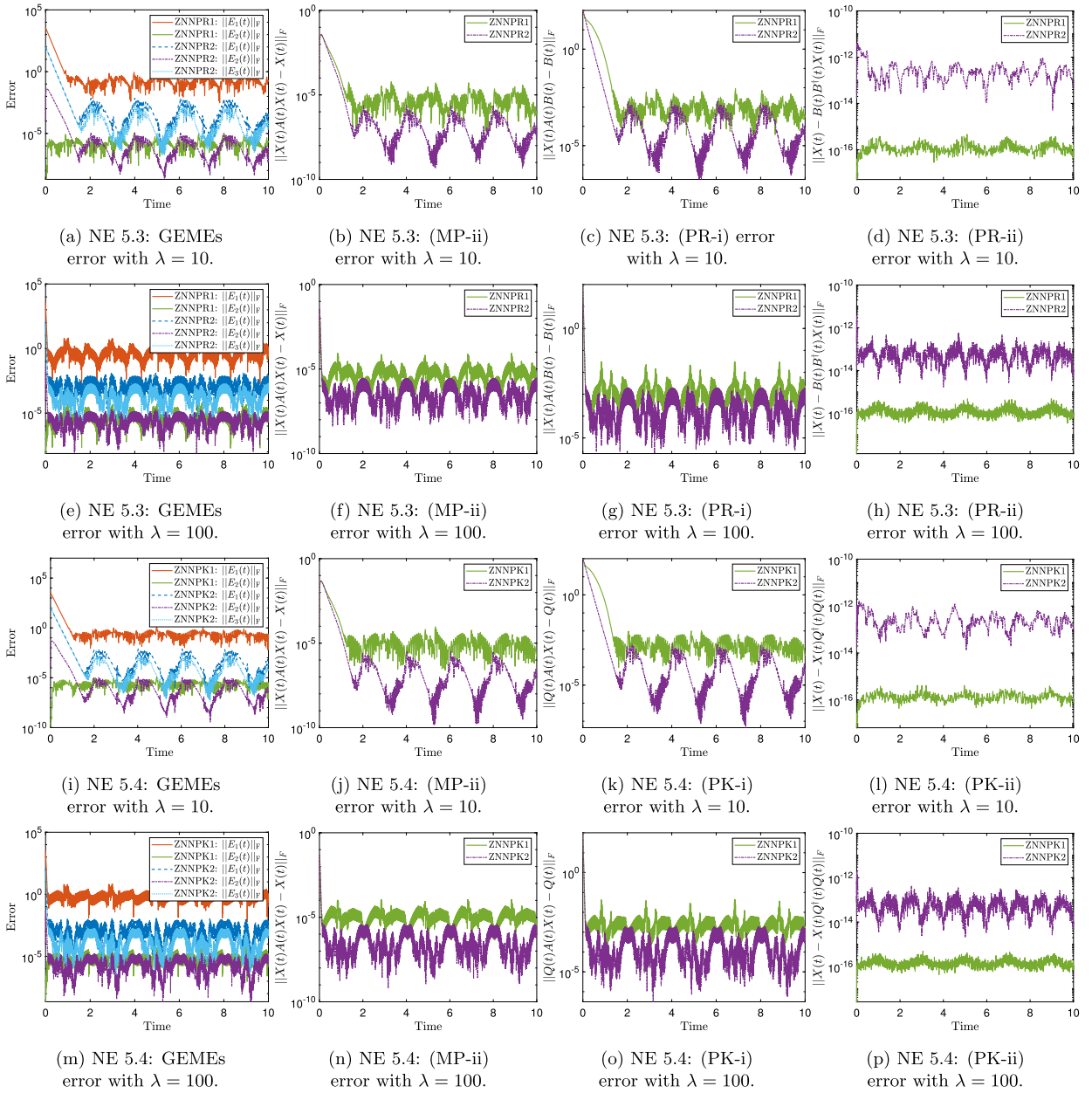
5.5. Numerical example 5

Consider the following matrices:

$$K_1(t)=\begin{bmatrix} 3+\gamma(t) & 1+\alpha(t)/2 & \dots & 1+\alpha(t)/6 \\ 1+\alpha(t)/2 & 3+\gamma(t) & \dots & 1+\alpha(t)/5 \\ \vdots & \vdots & \ddots & \vdots \\ 1+\alpha(t)/10 & 1+\alpha(t)/9 & \dots & 3+\gamma(t) \end{bmatrix},$$

$$K_2(t)=[1+\alpha(t),6+\gamma(t),5+2\alpha(t),2+2\gamma(t),\quad 4+\alpha(t),3+\gamma(t)]^{\mathrm{T}}\odot \mathbf{1}_{6\times 3}.$$

In this example, we set the input matrices $A(t)=K_1(t)+2K_1(t)\imath\in\mathbb{C}^{10\times 6}$ with $\mathrm{nr}(A(t))=6$ and $B(t)=K_2(t)+1/2K_2(t)\imath\in\mathbb{C}^{6\times 3}$ with $\mathrm{nr}(B(t))=1$ to calculate the TV-MROI with PR. As a result, $\mathrm{nr}(A(t)B(t))=\mathrm{nr}(B(t))=1$ and the TV-MROI is $X(t)\in\mathbb{C}^{5\times 4}$ with $\mathrm{nr}(X(t))=1$. The outcomes of the ZNNPR1 and ZNNPR2 models are shown in Fig. 4.



(a) NE 5.3: GEMEs error with $\lambda = 10$. (b) NE 5.3: (MP-ii) error with $\lambda = 10$. (c) NE 5.3: (PR-i) error with $\lambda = 10$. (d) NE 5.3: (PR-ii) error with $\lambda = 10$.

(e) NE 5.3: GEMEs error with $\lambda = 100$. (f) NE 5.3: (MP-ii) error with $\lambda = 100$. (g) NE 5.3: (PR-i) error with $\lambda = 100$. (h) NE 5.3: (PR-ii) error with $\lambda = 100$.

(i) NE 5.4: GEMEs error with $\lambda = 10$. (j) NE 5.4: (MP-ii) error with $\lambda = 10$. (k) NE 5.4: (PK-i) error with $\lambda = 10$. (l) NE 5.4: (PK-ii) error with $\lambda = 10$.

(m) NE 5.4: GEMEs error with $\lambda = 100$. (n) NE 5.4: (MP-ii) error with $\lambda = 100$. (o) NE 5.4: (PK-i) error with $\lambda = 100$. (p) NE 5.4: (PK-ii) error with $\lambda = 100$.

Fig. 2. Errors of the ZNNPR1, ZNNPR2, ZNNPK1 and ZNNPK2 models in NEs 5.3 and 5.4.

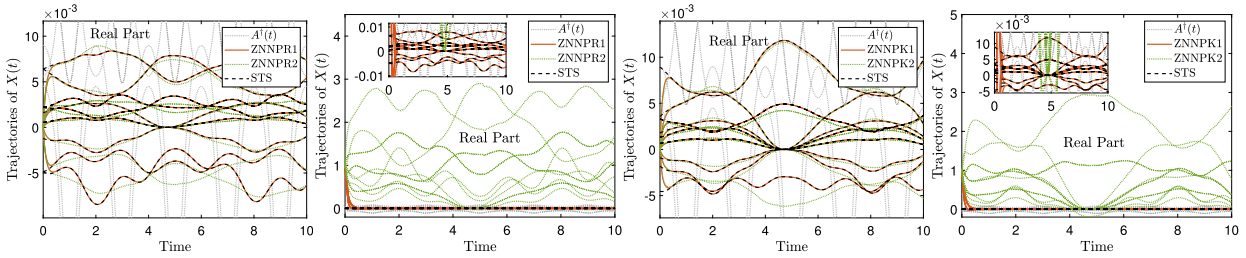
5.6. Numerical example 6

Consider the matrix $K_1(t)$ of NE 5.5 and the following matrices:

$$L_1(t)=K_1^{\mathrm{T}}(t),$$

$$L_2(t)=[\alpha(t)+1, 2+\gamma(t), 3+2\alpha(t), 4+2\gamma(t), 5+\alpha(t), 6+\gamma(t)]\odot \mathbf{1}_3.$$

In this example, we set the input matrices $A(t) = L_1(t) + 2L_1(t)\imath \in \mathbb{C}^{6\times 10}$ with $\mathrm{nr}(A(t)) = 6$ and $Q(t) = L_2(t) + 1/2L_2(t)\imath \in \mathbb{C}^{3\times 6}$ with $\mathrm{nr}(Q(t)) = 1$ to calculate the TV-MROI with PK. As a result, $\mathrm{nr}(Q(t)A(t)) = \mathrm{nr}(Q(t)) = 1$ and the TV-MROI is $X(t) \in \mathbb{C}^{10\times 6}$ with $\mathrm{nr}(X(t)) = 1$. The outcomes of the ZNNPK1 and ZNNPK2 models are shown in Fig. 4.



(a) NE 5.1: $X(t)$ real part with IC1. (b) NE 5.1: $X(t)$ real part with IC2. (c) NE 5.2: $X(t)$ real part with IC1. (d) NE 5.2: $X(t)$ real part with IC2.

(e) NE 5.1: $X(t)$ imag. part with IC1. (f) NE 5.1: $X(t)$ imag. part with IC2. (g) NE 5.2: $X(t)$ imag. part with IC1. (h) NE 5.2: $X(t)$ imag. part with IC2.

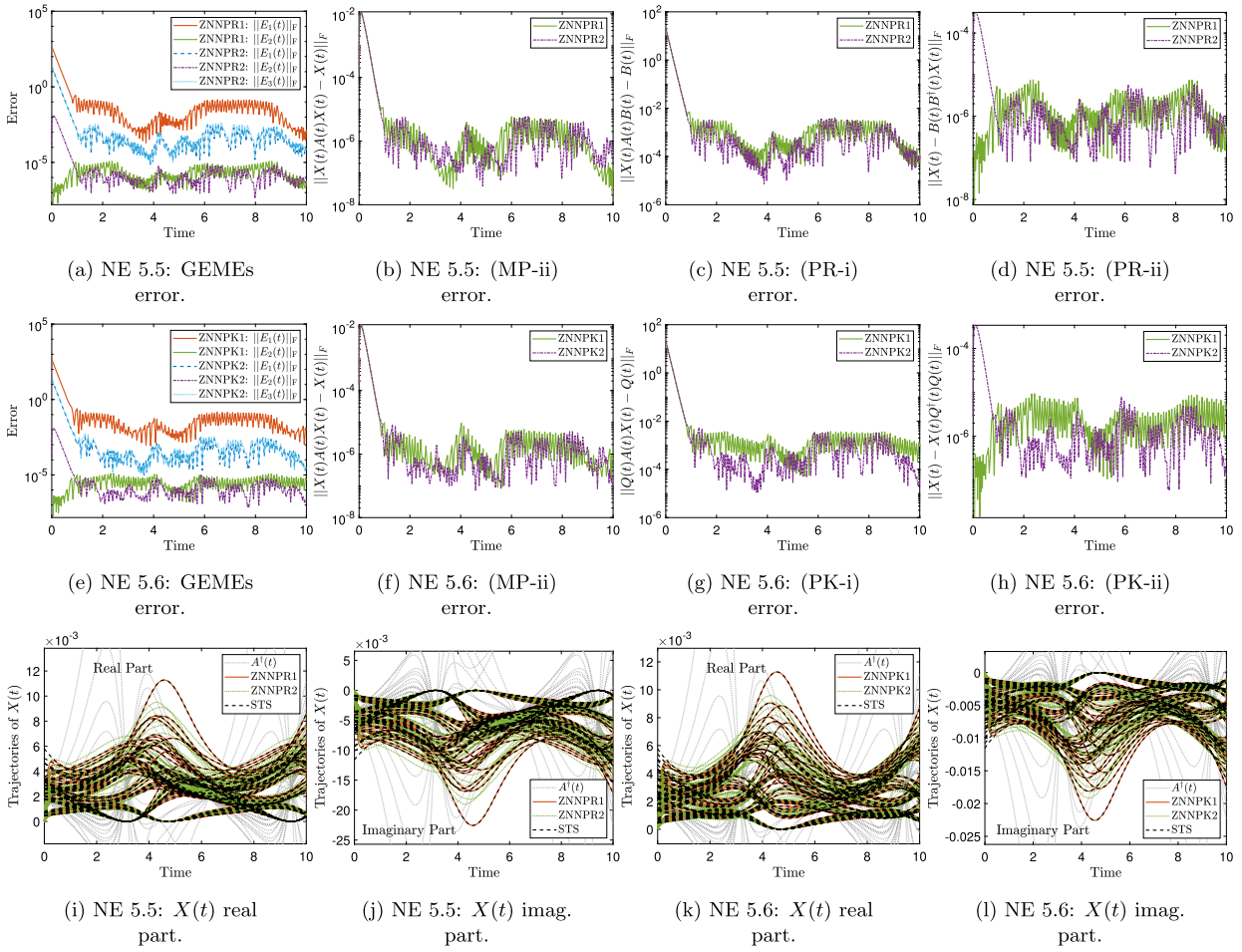
(i) NE 5.3: $X(t)$ real part. (j) NE 5.3: $X(t)$ imag. part. (k) NE 5.4: $X(t)$ real part. (l) NE 5.4: $X(t)$ imag. part.

Fig. 3. Trajectories of the real and imaginary parts of $X(t)$ with PR and PK in NEs 5.1-5.4.

5.7. Discussion and analysis of numerical examples

The capability of the ZNNPR1 and ZNNPR2 models for computing the TV-MROI with PR is examined through NEs 5.1, 5.3 and 5.5, while the capability of the ZNNPK1 and ZNNPK2 models for computing the TV-MROI with PK is examined thru NEs 5.2, 5.4 and 5.6.

In the cases of NEs 5.1 and 5.2, two different initial conditions (i.e. IC1 and IC2) have been used. Particularly, Figs. 1a and 1e show the GEMEs convergence in NE 5.1 under IC1 and IC2, respectively, and Figs. 1i and 1m show the GEMEs convergence in NE 5.2 under IC1 and IC2, respectively. We observe, there in, that the convergence begins at $t = 0$ with high errors and ends prior to $t = 2$ with errors in the range $[10^{-7}, 10^{-1}]$. Figs. 1b and 1f show the (MP-ii) convergence in NE 5.1 under IC1 and IC2, respectively, and Figs. 1j and 1n show the (MP-ii) convergence in NE 5.2 under IC1 and IC2, respectively. In Fig. 1b, the convergence speeds of the ZNNPR1 and ZNNPR2 under IC1 are identical despite the ZNNPR2 model having a smaller total error. In Fig. 1f, the convergence speed of the ZNNPR1 is higher than the ZNNPR2, and it also generates a smaller total error is lower than the ZNNPR2. In Fig. 1j, the convergence speeds of the ZNNPK1 and ZNNPK2 under IC1 are identical despite the ZNNPK2 model having a smaller total error. In Fig. 1n, the convergence speed of the ZNNPK1 is higher than the ZNNPK2, and it also has a smaller total error than the ZNNPK2. Figs. 1c and 1g show the (PR-i) convergence in NE 5.1 under IC1 and IC2, respectively, and Figs. 1k and 1o show the (PK-i) convergence in NE 5.2 under IC1 and IC2, respectively. There in, we can see that the initial conditions do not affect the convergence speed of the models, but affect the total error. That is, ZNNPR2 and ZNNPK2, respectively, have a smaller total error than the ZNNPR1 and ZNNPK1 under IC1, whereas ZNNPR1 and ZNNPK1, respectively, have a smaller total error than the ZNNPR2 and ZNNPK2 under IC1. Notice that Figs. 1d and 1h, which show the (PR-ii) convergence in NE 5.1 under IC1 and IC2, respectively, and Figs. 1l and 1p, which show the (PK-ii) convergence in NE 5.2 under IC1 and IC2, respectively, demonstrate the similar error trend. Figs. 3a and 3b show the real part trajectories of the solution $X(t)$, STS and $A^{\dagger}(t)$ in NE 5.1 under IC1 and IC2, respectively, while Figs. 3e and 3f show their imaginary part trajectories. Notably, the ZNNPR1 solution matches the STS, while STS has different trajectories from $A^{\dagger}(t)$. In addition, the ZNNPR2 solution is different than the STS and $A^{\dagger}(t)$. Figs. 3c and 3d show the real part trajectories of the solution $X(t)$, STS and $A^{\dagger}(t)$ in NE 5.2 under IC1 and IC2, respectively, while Figs. 3g and 3h show their imaginary part trajectories. Notably, the ZNNPK1 solution matches the STS, while STS has different trajectories from $A^{\dagger}(t)$. In addition, the ZNNPK2 solution is different than



(a) NE 5.5: GEMEs error. (b) NE 5.5: (MP-ii) error. (c) NE 5.5: (PR-i) error. (d) NE 5.5: (PR-ii) error.

(e) NE 5.6: GEMEs error. (f) NE 5.6: (MP-ii) error. (g) NE 5.6: (PK-i) error. (h) NE 5.6: (PK-ii) error.

(i) NE 5.5: $X(t)$ real part. (j) NE 5.5: $X(t)$ imag. part. (k) NE 5.6: $X(t)$ real part. (l) NE 5.6: $X(t)$ imag. part.

Fig. 4. Errors of the ZNNPR1, ZNNPR2, ZNNPK1 and ZNNPK2 models, and trajectories of the real and imaginary parts of $X(t)$ with PR and PK in NEs 5.5 and 5.6.

the STS and $A^\dagger(t)$. It is important to notice that the initial conditions have an impact on the solutions of ZNNPR2 and ZNNPK2 but not ZNNPR1 and ZNNPK1, which always match the STS. In other words, for various initial conditions, the ZNNPR2 generates various solutions to the TV-MROI with PR and the ZNNPK2 generates various solutions to the TV-MROI with PK.

Two λ values (10 and 100) have been used in NEs 5.3 and 5.4. Particularly, Figs. 2a and 2e show the GEMEs convergence in NE 5.3 under $\lambda = 10$ and $\lambda = 100$, respectively, and Figs. 2i and 2m show the GEMEs convergence in NE 5.4 under $\lambda = 10$ and $\lambda = 100$, respectively. In the case of $\lambda = 10$, we see that the convergence begins at $t = 0$ with high errors and ends prior to $t = 2$ with errors in the range $[10^{-9}, 10^{-1}]$. In the case of $\lambda = 100$, we can see that the convergence starts at $t = 0$ with high errors and ends prior to $t = 0.2$ with errors in the range $[10^{-9}, 10^{-1}]$. Figs. 2b and 2f show the (MP-ii) convergence in NE 5.3 under $\lambda = 10$ and $\lambda = 100$, respectively, and Figs. 2j and 2n show the (MP-ii) convergence in NE 5.4 under $\lambda = 10$ and $\lambda = 100$, respectively. In Figs. 2b and 2f, the convergence speeds of the ZNNPR1 and ZNNPR2 are almost identical despite the ZNNPR2 model having a smaller total error, and in Figs. 2j and 2n, the convergence speeds of the ZNNPK1 and ZNNPK2 are almost identical despite the ZNNPR2 model having a smaller total error. Notice that Figs. 2c and 2g (which show the (PR-i) convergence in NE 5.3 under $\lambda = 10$ and $\lambda = 100$, respectively), and Figs. 2k and 2o (which show the (PK-i) convergence in NE 5.4 under $\lambda = 10$ and $\lambda = 100$, respectively), demonstrate the similar error trend. Figs. 2d and 2h show the (PR-ii) convergence in NE 5.3 under $\lambda = 10$ and $\lambda = 100$, respectively, and Figs. 2l and 2p show the (PK-ii) convergence in NE 5.4 under $\lambda = 10$ and $\lambda = 100$, respectively. In Figs. 2d and 2h, the ZNNPR2 is already converged and has much smaller total error than the ZNNPR1, and in Figs. 2l and 2p, the ZNNPK2 converges and has much smaller total error than the ZNNPK1. Notice that when the value of λ is 100 as opposed to 10, the overall error is reduced much more. Fig. 3i shows the real part trajectories of the solution $X(t)$, STS and $A^\dagger(t)$ in NE 5.3 under $\lambda = 10$, while Fig. 3j shows their imaginary part trajectories. Notably, the ZNNPR1 solution matches the STS, while STS has different trajectories from $A^\dagger(t)$. In addition, the ZNNPR2 solution is different than the STS and $A^\dagger(t)$. Fig. 3k shows the real part trajectories of the solution $X(t)$, STS and $A^\dagger(t)$ in NE 5.4 under $\lambda = 10$, while Fig. 3l shows their imaginary part trajectories. Notably, the ZNNPK1 solution matches the STS, while STS has different trajectories from $A^\dagger(t)$. In addition, the ZNNPK2 solution is different than the STS.

In the cases of NEs 5.5 and 5.6, input matrices with relatively large dimensions have been used. Particularly, Figs. 4a and 4e show the GEMEs convergence in NEs 5.5 and 5.6, respectively. There we can see that the convergence begins at $t = 0$ with high

errors and ends prior to $t = 2$ with errors in the range $[10^{-8}, 10^{-1}]$. Figs. 4b and 4f show the (MP-ii) convergence in NEs 5.5 and 5.6, respectively, Fig. 4c shows the (PR-i) convergence in NE 5.5, and Fig. 4g shows the (PK-i) convergence in NE 5.6. We find that both the convergence speeds and the total error of the ZNNPR1 and ZNNPR2 are identical, as are the convergence speeds and total error of the ZNNPK1 and ZNNPK2. Further, Fig. 4d shows the (PR-ii) convergence in NE 5.5, and Fig. 4h shows the (PK-ii) convergence in NE 5.6. In Fig. 4d, the ZNNPR2 is already converged but ZNNPR1 has smaller total error, and In Fig. 4h, the ZNNPK2 is already converged but ZNNPK1 has smaller total error. Fig. 4i shows the real part trajectories of the solution $X(t)$, STS and $A^{\dagger}(t)$ in NE 5.5, while Fig. 4j shows their imaginary part trajectories. Notably, the ZNNPR1 solution matches the STS, while STS has different trajectories from $A^{\dagger}(t)$. In addition, the ZNNPR2 solution is different than the STS and $A^{\dagger}(t)$. Fig. 4k shows the real part trajectories of the solution $X(t)$, STS and $A^{\dagger}(t)$ in NE 5.6, while Fig. 4l shows their imaginary part trajectories. Notably, the ZNNPK1 solution matches the STS, while STS has different trajectories from $A^{\dagger}(t)$. In addition, the ZNNPK2 solution is different than the STS.

The NEs performed in this section lead to the following conclusions.
- The ZNNPR1, ZNNPR2, ZNNPK1, and ZNNPK2 models converge to the zero matrix quickly after starting from an initial condition that is not ideal.
- The initial conditions do not have an impact on the solutions ZNNPR1 and ZNNPK1, which always match the STS.
- The initial conditions have an impact on the solutions of ZNNPR2 and ZNNPK2. That is, for various initial conditions, the ZNNPR2 generates various solutions to the TV-MROI with PR and the ZNNPK2 generates various solutions to the TV-MROI with PK.
- The value of λ regulates the GEME convergence ending period t.
- The $X(t)$ solutions trajectories, as well as the (MP-ii), (PR-i) and (PK-i) convergence, act in the same manner because of the GEMEs' convergence features.
- In the case of (PR-ii), the ZNNPR1 and ZNNPK1 models outperform the ZNNPR2 and ZNNPK2 models, respectively. However, according to Section 4 findings, the ZNNPR2 and ZNNPK2 models have lower computational complexity than the ZNNPR1 and ZNNPK1 models, respectively.
- The ZNNPR1, ZNNPR2, ZNNPK1 and ZNNPK2 models can handle input matrices with relatively large dimensions.

In essence, the TV-MROI with PR is calculated with outstanding performance by the ZNNPR1 and ZNNPR2 models, and the TV-MROI with PK is calculated with effective performance by the ZNNPK1 and ZNNPK2 models.

6. Conclusion

This paper presented four ZNN models (i.e. ZNNPR1, ZNNPR2, ZNNPK1 and ZNNPK2) for calculating the TV-MROI with PR and with PK. Particularly, the ZNNPR1 and ZNNPR2 models calculate the TV-MROI with PR and the ZNNPK1 and ZNNPK2 models the TV-MROI with PK. For investigating the performance of the models, six NEs were presented. With the exception of one error measurement, where the ZNNPR1 and ZNNPK1 models outperform the ZNNPR2 and ZNNPK2 models, the performance of the ZNNPR1 and ZNNPR2 models, as well as the ZNNPK1 and ZNNPK2 models, is almost identical. However, the ZNNPR2 and ZNNPK2 models have lower computational complexity than the ZNNPR1 and ZNNPK1 models, respectively. To conclude, the findings show that all models generate the TS with outstanding and effective performance.

The list below includes some possible research topics.
1. The ZNNPR1, ZNNPR2, ZNNPK1 and ZNNPK2 models could be enhanced using ZNN designs with terminal convergence.
2. As algebra relies heavily on generalized inverses, future works can explore particular varieties of generalized inverses, including the time-varying {1,3} and {1,4}-inverses.
3. ZNN designs built on a nature-inspired metaheuristic optimization technique, such those shown in [43,44], could be used to improve the models described in this paper.

CRediT authorship contribution statement

Predrag S. Stanimirović: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Spyridon D. Mourtas:** Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Dijana Mosić:** Investigation, Methodology, Writing – original draft, Writing – review & editing. **Vasilios N. Katsikis:** Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Xinwei Cao:** Formal analysis, Methodology, Software, Validation. **Shuai Li:** Formal analysis, Methodology, Software, Validation.

Data availability

No data was used for the research described in the article.

Acknowledgements

Predrag Stanimirović is supported by the Science Fund of the Republic of Serbia, #GRANT No 7750185, Quantitative Automata Models: Fundamental Problems and Applications - QUAM.
Predrag Stanimirović and Dijana Mosić are supported by Ministry of Education, Science and Technological Development, Republic of Serbia, Contract No. 451-03-47/2023-01/200124.
This work was supported by the Ministry of Science and Higher Education of the Russian Federation (Grant No. 075-15-2022-1121).

References

[1] A. Ben-Israel, T.N.E. Greville, Generalized Inverses: Theory and Applications, 2nd edition, CMS Books in Mathematics, Springer, New York, NY, 2003.
[2] G. Wang, Y. Wei, S. Qiao, P. Lin, Y. Chen, Generalized Inverses, Theory and Computations, vol. 53, Springer, Singapore, 2018.
[3] Y. Yuan, Z. Wang, L. Guo, Event-triggered strategy design for discrete-time nonlinear quadratic games with disturbance compensations: the noncooperative case, IEEE Trans. Syst. Man Cybern. Syst. 48 (2018) 1885–1896, https://doi.org/10.1109/tsmc.2017.2704278.
[4] S. Zhang, Y. Dong, Y. Ouyang, Z. Yin, K. Peng, Adaptive neural control for robotic manipulators with output constraints and uncertainties, IEEE Trans. Neural Netw. Learn. Syst. 29 (2018) 5554–5564, https://doi.org/10.1109/tnnls.2018.2803827.
[5] Y. Shi, W. Zhao, S. Li, B. Li, X. Sun, Novel discrete-time recurrent neural network for robot manipulator: a direct discretization technical route, IEEE Trans. Neural Netw. Learn. Syst. 34 (6) (2023) 2781–2790, https://doi.org/10.1109/TNNLS.2021.3108050.
[6] Y. Shi, J. Wang, S. Li, B. Li, X. Sun, Tracking control of cable-driven planar robot based on discrete-time recurrent neural network with immediate discretization method, IEEE Trans. Ind. Inform. 19 (6) (2023) 7414–7423, https://doi.org/10.1109/TII.2022.3210255.
[7] T.E. Simos, V.N. Katsikis, S.D. Mourtas, P.S. Stanimirović, Solving time-varying nonsymmetric algebraic Riccati equations with zeroing neural dynamics, IEEE Trans. Syst. Man Cybern. Syst. (2023) 1–13, https://doi.org/10.1109/tsmc.2023.3284533.
[8] E.S. Ghith, F.A.A. Tolba, Real-time implementation of tuning PID controller based on whale optimization algorithm for micro-robotics system, in: 2022 14th International Conference on Computer and Automation Engineering (ICCAE), 2022.
[9] X. Yang, H. He, Self-learning robust optimal control for continuous-time nonlinear systems with mismatched disturbances, Neural Netw. 99 (2018) 19–30, https://doi.org/10.1016/j.neunet.2017.11.022.
[10] S.D. Mourtas, A weights direct determination neuronet for time-series with applications in the industrial indices of the federal reserve bank of St. Louis, J. Forecast. 14 (7) (2022) 1512–1524, https://doi.org/10.1002/for.2874.
[11] S. Li, J. He, Y. Li, M.U. Rafique, Distributed recurrent neural networks for cooperative control of manipulators: a game-theoretic perspective, IEEE Trans. Neural Netw. Learn. Syst. 28 (2) (2017) 415–426, https://doi.org/10.1109/TNNLS.2016.2516565.
[12] E. Canuto, C. Novara, L. Massotti, D. Carlucci, C.P. Montenegro, Chapter 9 - Orbit and attitude actuators, in: E. Canuto, C. Novara, L. Massotti, D. Carlucci, C.P. Montenegro (Eds.), Spacecraft Dynamics and Control, Aerospace Engineering, Butterworth-Heinemann, 2018, pp. 463–520, https://www.sciencedirect.com/science/article/pii/B978008100700600009X.
[13] D.N. Nenchev, A. Konno, T. Tsujita, Chapter 2 - Kinematics, in: D.N. Nenchev, A. Konno, T. Tsujita (Eds.), Humanoid Robots, Butterworth-Heinemann, 2019, pp. 15–82, https://www.sciencedirect.com/science/article/pii/B9780128045602000092.
[14] B. Ciubotaru, M. Staroswiecki, C. Christophe, Fault tolerant control of the Boeing 747 short-period mode using the admissible model matching technique, in: H.-Y. Zhang (Ed.), Fault Detection, Supervision and Safety of Technical Processes 2006, Elsevier Science Ltd, Oxford, 2007, pp. 819–824, https://www.sciencedirect.com/science/article/pii/B978008044485750138X.
[15] H. Huang, D. Fu, X. Xiao, Y. Ning, H. Wang, L. Jin, S. Liao, Modified Newton integration neural algorithm for dynamic complex-valued matrix pseudoinversion applied to mobile object localization, IEEE Trans. Ind. Inform. 17 (2020) 2432–2442.
[16] A.G. Dempster, E. Cetin, Interference localization for satellite navigation systems, Proc. IEEE 104 (6) (2016) 1318–1326.
[17] A. Noroozi, A.H. Oveis, S.M. Hosseini, M.A. Sebt, Improved algebraic solution for source localization from tdoa and fdoa measurements, IEEE Wirel. Commun. Lett. 7 (3) (2018) 352–355.
[18] A.J. Getson, F.C. Hsuan, {2}-Inverses and Their Statistical Application, vol. 47, Springer Science & Business Media, 2012.
[19] M.Z. Nashed, Generalized inverses and applications, in: Proceedings of an Advanced Seminar Sponsored by the Mathematics Research Center, the University of Wisconsin, Madison, October 8–10, 1973, Elsevier, 2014, 32.
[20] D. Mosić, P.S. Stanimirović, S.D. Mourtas, Minimal rank properties of outer inverses with prescribed range and null space, Mathematics 11 (2023) 1732, https://doi.org/10.3390/math11071732.
[21] P.S. Stanimirović, V.N. Katsikis, D. Pappas, Computing {2, 4} and {2, 3}-inverses by using the Sherman-Morrison formula, Appl. Math. Comput. 273 (2015) 584–603.
[22] Y. Zhang, S.S. Ge, Design and analysis of a general recurrent neural network model for time-varying matrix inversion, IEEE Trans. Neural Netw. 16 (6) (2005) 1477–1490, https://doi.org/10.1109/TNN.2005.857946.
[23] Y. Chai, H. Li, D. Qiao, S. Qin, J. Feng, A neural network for Moore-Penrose inverse of time-varying complex-valued matrices, Int. J. Comput. Intell. Syst. 13 (1) (2020) 663–671, https://doi.org/10.2991/ijcis.d.200527.001.
[24] Z. Sun, F. Li, L. Jin, T. Shi, K. Liu, Noise-tolerant neural algorithm for online solving time-varying full-rank matrix Moore-Penrose inverse problems: a control-theoretic approach, Neurocomputing 413 (2020) 158–172, https://doi.org/10.1016/j.neucom.2020.06.050.
[25] W. Wu, B. Zheng, Improved recurrent neural networks for solving Moore-Penrose inverse of real-time full-rank matrix, Neurocomputing 418 (2020) 221–231, https://doi.org/10.1016/j.neucom.2020.08.026.
[26] Y. Zhang, Y. Yang, N. Tan, B. Cai, Zhang neural network solving for time-varying full-rank matrix Moore-Penrose inverse, Computing 92 (2) (2011) 97–121, https://doi.org/10.1007/s00607-010-0133-9.
[27] S. Qiao, X.-Z. Wang, Y. Wei, Two finite-time convergent Zhang neural network models for time-varying complex matrix Drazin inverse, Linear Algebra Appl. 542 (2018) 101–117, https://doi.org/10.1016/j.laa.2017.03.014.
[28] S. Qiao, Y. Wei, X. Zhang, Computing time-varying ML-weighted pseudoinverse by the Zhang neural networks, Numer. Funct. Anal. Optim. 41 (2020) 1672–1693, https://doi.org/10.1080/01630563.2020.1740887.
[29] X. Wang, P.S. Stanimirovic, Y. Wei, Complex ZFs for computing time-varying complex outer inverses, Neurocomputing 275 (2018) 983–1001, https://doi.org/10.1016/j.neucom.2017.09.034.
[30] M. Zhou, J. Chen, P.S. Stanimirovic, V.N. Katsikis, H. Ma, Complex varying-parameter Zhang neural networks for computing core and core-EP inverse, Neural Process. Lett. 51 (2) (2020) 1299–1329, https://doi.org/10.1007/s11063-019-10141-6.
[31] J. Liu, H. Cai, C. Jiang, X. Han, Z. Zhang, An interval inverse method based on high dimensional model representation and affine arithmetic, Appl. Math. Model. 63 (2018) 732–743, https://doi.org/10.1016/j.apm.2018.07.009.
[32] S.D. Mourtas, V.N. Katsikis, Exploiting the Black-Litterman framework through error-correction neural networks, Neurocomputing 498 (2022) 43–58, https://doi.org/10.1016/j.neucom.2022.05.036.
[33] S.D. Mourtas, C. Kasimis, Exploiting mean-variance portfolio optimization problems through zeroing neural networks, Mathematics 10 (3079) (2022) 3079, https://doi.org/10.3390/math10173079, https://www.mdpi.com/2227-7390/10/17/3079.
[34] W. Jiang, C.-L. Lin, V.N. Katsikis, S.D. Mourtas, P.S. Stanimirović, T.E. Simos, Zeroing neural network approaches based on direct and indirect methods for solving the Yang–Baxter-like matrix equation, Mathematics 10 (2022) 1950, https://doi.org/10.3390/math10111950.
[35] H. Jerbi, H. Alharbi, M. Omri, L. Ladhar, T.E. Simos, S.D. Mourtas, V.N. Katsikis, Towards higher-order zeroing neural network dynamics for solving time-varying algebraic Riccati equations, Mathematics 10 (4490) (2022) 4490, https://doi.org/10.3390/math10234490, https://www.mdpi.com/2227-7390/10/23/4490.
[36] L. Xiao, W. Huang, X. Li, F. Sun, Q. Liao, L. Jia, J. Li, S. Liu, ZNNs with a varying-parameter design formula for dynamic Sylvester quaternion matrix equation, IEEE Trans. Neural Netw. Learn. Syst. (2022) 1–11, https://doi.org/10.1109/tnnls.2022.3163293.
[37] V.N. Katsikis, P.S. Stanimirović, S.D. Mourtas, L. Xiao, D. Karabasević, D. Stanujkić, Zeroing neural network with fuzzy parameter for computing pseudoinverse of arbitrary matrix, IEEE Trans. Fuzzy Syst. 30 (9) (2022) 3426–3435, https://doi.org/10.1109/tfuzz.2021.3115969.

[38] H. Alharbi, H. Jerbi, M. Kchaou, R. Abbassi, T.E. Simos, S.D. Mourtas, V.N. Katsikis, Time-varying pseudoinversion based on full-rank decomposition and zeroing neural networks, Mathematics 11 (3) (2023) 600, https://doi.org/10.3390/math11030600.
[39] L. Xiao, P. Cao, W. Song, L. Luo, W. Tang, A fixed-time noise-tolerance ZNN model for time-variant inequality-constrained quaternion matrix least-squares problem, IEEE Trans. Neural Netw. Learn. Syst. (2023) 1–10, https://doi.org/10.1109/tnnls.2023.3242313.
[40] J. Dai, P. Tan, X. Yang, L. Xiao, L. Jia, Y. He, A fuzzy adaptive zeroing neural network with superior finite-time convergence for solving time-variant linear matrix equations, Knowl.-Based Syst. 242 (2022) 108405, https://doi.org/10.1016/j.knosys.2022.108405.
[41] L. Xiao, H. Tan, J. Dai, L. Jia, W. Tang, High-order error function designs to compute time-varying linear matrix equations, Inf. Sci. 576 (2021) 173–186, https://doi.org/10.1016/j.ins.2021.06.038.
[42] N. Zhong, Q. Huang, S. Yang, F. Ouyang, Z. Zhang, A varying-parameter recurrent neural network combined with penalty function for solving constrained multi-criteria optimization scheme for redundant robot manipulators, IEEE Access 9 (2021) 50810–50818, https://doi.org/10.1109/access.2021.3068731.
[43] A.H. Khan, S. Li, X. Luo, Obstacle avoidance and tracking control of redundant robotic manipulator: an RNN-based metaheuristic approach, IEEE Trans. Ind. Inform. 16 (2020) 4670–4680, https://doi.org/10.1109/TII.2019.2941916.
[44] A.H. Khan, S. Li, D. Chen, L. Liao, Tracking control of redundant mobile manipulator: an RNN based metaheuristic approach, Neurocomputing 400 (2020) 272–284, https://doi.org/10.1016/j.neucom.2020.02.109.