Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 4227512, 21 pages
https://doi.org/10.1155/2021/4227512

Research Article

From Penrose Equations to Zhang Neural Network, Getz–Marsden Dynamic System, and DDD (Direct Derivative Dynamics) Using Substitution Technique

Dongqing Wu [1,2,3,4,5] **and Yunong Zhang** [2,3,4,5]

[1]*School of Computational Science, Zhongkai University of Agriculture and Engineering, Guangzhou 510220, China*
[2]*School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou 510006, China*
[3]*Research Institute of Sun Yat-sen University in Shenzhen, Shenzhen 518057, China*
[4]*Guangdong Key Laboratory of Modern Control Technology, Guangzhou 510070, China*
[5]*Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, Guangzhou 510006, China*

Correspondence should be addressed to Yunong Zhang; zhynong@mail.sysu.edu.cn

Received 19 May 2021; Accepted 25 October 2021; Published 20 November 2021

Academic Editor: Sundarapandian Vaidyanathan

The time-variant matrix inversion (TVMI) problem solving is the hotspot of current research because of its frequent appearance and application in scientific research and industrial production. The generalized inverse problem of singular square matrix and nonsquare matrix can be related to Penrose equations (PEs). The PEs implicitly define the generalized inverse of a known matrix, which is of fundamental theoretical significance. Therefore, the in-depth study of PEs might enlighten problem solving of TVMI in a foreseeable way. For the first time, we construct three different matrix error-monitoring functions based on PEs and propose the corresponding models for TVMI problem solving by using the substitution technique and ZNN design formula. In order to facilitate computer simulation, the obtained continuous-time models are discretized by using ZTD (Zhang time discretization) formulas. Furthermore, the feasibility and effectiveness of the novel Zhang neural network (ZNN) multiple-multiplication model for matrix inverse (ZMMMI) and the PEs-based Getz–Marsden dynamic system (PGMDS) model in solving the problem of TVMI are investigated and shown via theoretical derivation and computer simulation. Computer experiment results also illustrate that the direct derivative dynamics model for TVMI is less effective and feasible.

1. Introduction

During the past decades, scientists and engineers have encountered linear matrix equation problems, i.e., Lyapunov equation, Sylvester equation, and the variational problem, again and again in various scenarios. The matrix inversion problem is one of the most prominent subproblems in the linear matrix equation problems. Among the problems encountered in a variety of optimization problems, the fundamental one is the solution of matrix inversion, such as signal processing [1], biomedical prediction [2], image reconstruction [3], nonlinear optimization [4], and robot inverse kinematics [5–8]. Generally speaking, the matrix inversion problem can be formulated as $AX = I$, where $A \in \mathbb{R}^{n\times n}$ is a known constant matrix, $X \in \mathbb{R}^{n\times n}$ is the unknown matrix to be computed, and $I \in \mathbb{R}^{n\times n}$ is an identity matrix. Over the years, efforts were directed towards computational issues of time-variant matrix inversion (TVMI) and a wealth of algorithms were proposed and applied to solving this problem [9–11]. For example, Yeung and Kumbi [9] developed an inversion method based on the multidimensional discrete Fourier transform of matrix sequences and applied it to an electron amplifier. In [10], Benner et al. adopted Gauss–Huard algorithm to solve TVMI problem. In [11], Xiao et al. designed a complex-valued nonlinear recurrent neural network to solve time-varying complex matrix inversion. Zhang et al. [12] discussed the solution of TVMI problem based on direct

derivative dynamics (DDD) and proposed a zero-stable, consistent, and convergent four-step model.

According to the classic solution of matrix inversion, the time complexity of arithmetic operations is proportional to $O(n^3)$, where n represents the dimension of the square matrix [13]. Evidently, it is an unbearable cost for solving the TVMI problem. Therefore, it is an urgent need for effective methods in TVMI problem solving.

In recent years, with the rise of the artificial neural network, the recurrent neural network (RNN) has been thought and manifested to be an effective option for TVMI problem solving. Methods based on RNN have been proved to be promising for RNN's nature of high-speed parallel-processing and convenience of hardware implementation [14–18]. To solve the TVMI problem, Zhang et al. firstly proposed a new class of RNN, Zhang neural network, abbreviated as ZNN [19]. The essence of ZNN model is to construct a carefully selected error-monitoring function, termed Zhang function (ZF). The ZF can be negative, zero, positive, bounded, or even lower-unbounded. The traditional error-monitoring functions are often some norm-based positive-definite energy functions, which are not so flexible as ZFs. Moreover, since the time-derivative information of the time-variant matrix is fully utilized by ZFs, the resultant ZNN model can decrease the lagging errors which are almost unavoidable in the traditional methods. In theoretical and practical researches, the ZNN models claim their natural advantages in convergence and high accuracy. Inspired by this idea, the ZFs have been found to speed up and consolidate the development of various ZNN models [20–26].

With the deepening of the research on solving TVMI, designing a new ZF through specific classic equations, deriving new solution models, and discussing the convergence and accuracy of the models have become the key to the solution of the extended TVMI problem.

As a generalization of matrix inverse, the Moore–Penrose generalized inverse has been widely investigated [27]. It is applied to finding the least norm square solution of nonuniform linear equations and makes the form of solution simple. The Moore–Penrose generalized inverse of a matrix is unique in both real and complex fields, which can be obtained theoretically by singular value decomposition (SVD) algorithm [27]. Although the SVD algorithm provides a direction for TVMI problem solving, it is deeply troubled by the high time complexity of the algorithm. For example, according to the actual number of multiplication operations, SVD algorithm needs $2mn^2 + 4n^3$ operations to find the Moore–Penrose generalized inverse of an $m \times n$ matrix [27]. Therefore, based on the mathematical definition of inverse matrix, there have been many attempts to solve the TVMI problem [28–30].

In view of the fundamental theoretical value of Penrose equations (PEs) in the definition of generalized inverse, it is necessary to study the novel TVMI solution models and find equivalence between existing solution models from PEs. But as far as we know, there is no relevant research so far. This paper focuses on the PE relevant models. By constructing different matrix error-monitoring functions from PEs, three corresponding time-variant matrix inverse solution models are derived. Through theoretical derivation and computer simulation, the feasibility and the effectiveness of the new models in TVMI problem solving are verified.

The remainder of this paper is organized as follows. Section 2 explains the PEs and some necessary equations. In Section 3, we derive a new ZNN multiple-multiplication model for matrix inversion, abbreviated as ZMMMI, from PEs. Thereafter, we employ the Euler forward finite difference (two-instant) formula, the four-instant Taylor–Zhang formula, the six-instant Zhang time discretization (ZTD) formula [31–33], the eight-instant ZTD formula, and the ten-instant ZTD formula to develop five discrete algorithms to compute TVMI. In Section 4, we derive a new PEs-based Getz–Marsden dynamic system (PGMDS) model for time-variant matrix inversion from PEs. Thereafter, we employ the five discrete formulas to present five discrete algorithms to compute TVMI. In Section 5, we present a DDD (direct derivative dynamics) neural network model for time-variant matrix inversion on the basis of PEs. Thereafter, we employ the five discrete formulas to develop five discrete algorithms to compute TVMI. In Section 6, we conduct numerical experiments to verify the convergence and precision of the three new models. Based on the experiment results, we compare and discuss the effectiveness of the three models. Section 7 gives the conclusions and future directions of research.

Through the research of this paper, more TVMI solution models will be available to researchers. The main contributions of this paper can be outlined as follows.

(1) Three different models for solving the TVMI are proposed by defining different error-monitoring functions from PEs for the first time. This is the main motivation of this paper.

(2) We propose and provide a novel ZMMMI model from PEs, of which the stability and convergence are proved theoretically, thereby being rare complements to the problem solving of TVMI.

(3) The paper investigates and provides the theoretical analysis of the continuous-time models and finds that the classic GMDS model can also be derived from the PEs.

(4) For comparison, by exploiting the multiple-instant ZTD formulas, five discrete algorithms for each of the continuous-time ZMMMI, PGMDS, and DDD models are presented and constructed, respectively.

(5) The stability and convergence of the ZMMMI2i through ZMMMI10i algorithms and PGMDS2i through PGMDS10i algorithms are substantiated by numerical experiments of two benchmark examples.

(6) Numerical experiment results further indicate that compared with the effective ZMMMI algorithms and PGMDS algorithms, the DDD algorithms are less effective for TVMI.

2. Penrose Equations

For the purpose of laying a basis for direction, some necessary preliminaries of the time-variant matrix

pseudoinverse are given. For a given real time-variant matrix $A(t) \in \mathbb{R}^{m\times n}$, if one assumes that there exists an unknown real time-variant $X(t) \in \mathbb{R}^{n\times m}$ which [27, 34]

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

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

$$(A(t)X(t))^{\mathrm{T}} = A(t)X(t), \tag{1c}$$

$$(X(t)A(t))^{\mathrm{T}} = X(t)A(t), \tag{1d}$$

where t denotes the time, superscript $^{\mathrm{T}}$ indicates the matrix transpose operator, and $X(t)$ represents the time-variant pseudoinverse of $A(t)$, which is often denoted by $A^{+}(t)$. When rank $(A(t))$ is equal to m and $A(t)A^{\mathrm{T}}(t)$ is nonsingular, the unique $X(t)$ can be obtained as

$$X^{+}(t) \coloneqq A^{\mathrm{T}}(t)\left(A(t)A^{\mathrm{T}}(t)\right)^{-1} \in \mathbb{R}^{n\times m}. \tag{2}$$

Equation (2) holds true when $A(t)$ is full row rank. Similarly, when $A(t)$ is full column rank and $A(t)A^{\mathrm{T}}(t)$ is nonsingular, unique $X(t)$ can be obtained as

$$X^{+}(t) \coloneqq \left(A^{\mathrm{T}}(t)A(t)\right)^{-1}A^{\mathrm{T}}(t) \in \mathbb{R}^{n\times m}. \tag{3}$$

In equations (2) and (3), we use superscript $^{-1}$ to indicate the matrix inversion operator. Also, $X^{+}(t)$ is called the right and left pseudoinverse of $A(t)$ in (2) and (3), respectively.

3. New ZNN Model and Algorithms

This section focuses on constructing another new ZF from PEs and then derives a new continuous-time solution model for Penrose pseudoinverse based on ZNN design formula [35].

3.1. Continuous-Time Model from PEs. The design formula of ZNN is shown below:

$$\dot{Z}(t) = -\gamma Z(t), \tag{4}$$

where $\gamma \in \mathbb{R}^{+}$ is the parameter whose physical meaning is the reciprocal of the product of the corresponding capacitance parameter and the resistance parameter. According to [28], (4) is a first-order differential equation whose norm of the general solution is inversely proportional to the exponential function of γ. Therefore, in order to make $Z(t)$ converge to zero as rapid as possible, γ should be set as large as the physical hardware allows. Next, we discuss the ZNN model in the real situation. We choose (1a) to be the ZF as

$$Z(t) = A(t)X(t)A(t) - A(t), \tag{5}$$

and when $t \longrightarrow +\infty$, $Z(t) \longrightarrow 0$ theoretically. Then, we take the derivative of both sides of (5) and get

$$\begin{aligned}\dot{Z}(t) = {} & \dot{A}(t)X(t)A(t) + A(t)\dot{X}(t)A(t) \\ & + A(t)X(t)\dot{A}(t) - \dot{A}(t).\end{aligned} \tag{6}$$

Substituting (5) and (6) into (4), we obtain

$$\begin{aligned}\dot{Z}(t) &= -\gamma(A(t)X(t)A(t) - A(t)) \\ &= \dot{A}(t)X(t)A(t) + A(t)\dot{X}(t)A(t) \\ &\quad + A(t)X(t)\dot{A}(t) - \dot{A}(t).\end{aligned} \tag{7}$$

Reformulating (6), we have

$$\begin{aligned}A(t)\dot{X}(t)A(t) = {} & -\dot{A}(t)X(t)A(t) - A(t)X(t)\dot{A}(t) + \dot{A}(t) \\ & - \gamma(A(t)X(t)A(t) - A(t)).\end{aligned} \tag{8}$$

In order to derive the multiple-multiplication model, one assumes that $A(t)$ is square and of full rank. To get the explicit $\dot{X}(t)$, we need to left multiply both sides of (8) by $A^{-1}(t)$ and right multiply both sides of (8) by $A^{-1}(t)$ and then get explicit $\dot{X}(t)$ as

$$\begin{aligned}\dot{X}(t) = {} & A^{-1}(t)(-\dot{A}(t)X(t)A(t) - A(t)X(t)\dot{A}(t) \\ & + \dot{A}(t) - \gamma(A(t)X(t)A(t) - A(t)))A^{-1}(t).\end{aligned} \tag{9}$$

The goal of the model design in this section is to use the known matrix to estimate $\dot{A}(t)$, but $A^{-1}(t)$ appears on the right-hand side of (9), so it is necessary to use the substitution technique to replace $A^{-1}(t)$ with the appropriate known matrix. Based on Theorem 1 in [36], the state matrix $X(t)$ of (1a) globally converges to $A^{-1}(t)$ when the sampling gap $\epsilon \longrightarrow 0$ and t evolves large enough. Therefore, $X(t)$ would be a feasible substitute for $A^{-1}(t)$. This substitution technique has been proven to be effective in [37–39]. Therefore, we have the explicit dynamics of the ZNN models as

$$\begin{aligned}\dot{X}(t) = {} & X(t)(-\dot{A}(t)X(t)A(t) - A(t)X(t)\dot{A}(t) + \dot{A}(t) \\ & - \gamma(A(t)X(t)A(t) - A(t)))X(t).\end{aligned} \tag{10}$$

This novel model differs from any previous TVMI solution model. Note that the right-hand side of (10) involves multiple matrix-multiplication, so the model is termed as continuous-time ZMMMI model (10). To prepare for the following discussion, a lemma is given below [28].

Lemma 1. *For a time-variant real matrix $A(t) \in \mathbb{R}^{n\times n}$ and its inverse $A^{-1}(t)$, we have $\dot{A}^{-1}(t) = -A^{-1}(t)\dot{A}(t)A^{-1}(t)$.*

Proof. Because $A(t)A^{-1}(t) = I$ with $I \in \mathbb{R}^{n\times n}$ being the identity matrix, we have

$$\frac{\mathrm{d}\left(A(t)A^{-1}(t)\right)}{\mathrm{d}t} = \frac{\mathrm{d}I}{\mathrm{d}t} = 0. \tag{11}$$

Expanding the left-hand side of the above equation, we obtain

$$\frac{\mathrm{d}(A(t))}{\mathrm{d}t}A^{-1}(t) + A(t)\frac{\mathrm{d}\left(A^{-1}(t)\right)}{\mathrm{d}t} = 0. \tag{12}$$

Next, left multiply $A^{-1}(t)$ on both sides of the above equation and then reformulate it. Thus, we obtain

$$\frac{\mathrm{d}\left(A^{-1}(t)\right)}{\mathrm{d}t} = -A^{-1}(t)\frac{\mathrm{d}(A(t))}{\mathrm{d}t}A^{-1}(t). \tag{13}$$

For simplicity, the notation $\mathrm{d}(\cdot)/\mathrm{d}t$ is substituted by $\dot{(\cdot)}$. Then, we obtain

$$\dot{A}^{-1}(t) = -A^{-1}(t)\dot{A}(t)A^{-1}(t). \tag{14}$$

The proof is thus complete. □

For the continuous-time ZMMMI model (10), we provide the following proposition on its exponential convergence performance with a proper random initial state.

Proposition 1. *For a smoothly time-variant real matrix $A(t) \in \mathbb{R}^{n\times n}$ of full rank, the state matrix $X(t)$ of ZMMMI model (10) starting from proper random initial state $X(t)$ exponentially converges to the time-variant theoretical inverse of $A(t)$.*

Proof. Let $\alpha(t) = X(t) - A^{-1}(t)$ represent the difference between the solution generated by ZMMMI model (10) and the theoretical inverse of $A(t)$. Substituting $X(t) = \alpha(t) + A^{-1}(t)$ into (8), we obtain

$$\begin{aligned} A\left(\dot{\alpha} + \dot{A}^{-1}\right)A = &-\dot{A}\left(\alpha + A^{-1}\right)A - A\left(\alpha + A^{-1}\right)\dot{A} + \dot{A} \\ &- \gamma\left(A\left(\alpha + A^{-1}\right)A - A\right), \end{aligned} \tag{15}$$

where A, α, $\dot{A}$, and $\dot{\alpha}$ denote the time-variant matrices $A(t)$, $\alpha(t)$, $\dot{A}(t)$, and $\dot{\alpha}(t)$, respectively, for conciseness. According to Lemma 1 and $A^{-1}A = AA^{-1} = I$, (15) can be reformulated to

$$\dot{A}\alpha A + A\dot{\alpha}A + A\alpha\dot{A} = -\gamma A\alpha A. \tag{16}$$

Let $W = A\alpha A$. The above equation can be rewritten as

$$\dot{W} = -\gamma W, \tag{17}$$

which is the compact matrix form of the following set of $n \times n$ first-order differential equations:

$$\dot{\omega}_{ij}(t) = -\gamma\omega_{ij}(t), \quad \forall i, j \in 1, 2, \ldots, n. \tag{18}$$

Evidently, a Lyapunov function candidate $l_{ij} = \omega_{ij}^2/2 \geq 0$ can be defined for ijth subsystem (18), which is also positive definite, i.e., $l_{ij} > 0$ for $\omega_{ij} \neq 0$ and $l_{ij} = 0$ for $\omega_{ij} = 0$. Therefore, we have its time derivative

$$\frac{\mathrm{d}l_{ij}(t)}{\mathrm{d}t} = \omega_{ij}\dot{\omega}_{ij} = -\gamma\omega_{ij}^2. \tag{19}$$

Evidently, $\dot{l}_{ij}$ is negative definite, i.e., $\dot{l}_{ij} < 0$ for $\omega_{ij} \neq 0$ and $\dot{l}_{ij} = 0$ for $\omega_{ij} = 0$. Besides, when $|\omega_{ij}| \longrightarrow \infty$, the Lyapunov function candidate $l_{ij} = |\omega_{ij}|^2/2 \longrightarrow \infty$, where symbol $|\cdot|$ represents the absolute value of a scalar value. According to the Lyapunov stability theory, $\omega_{ij}(t)$ (globally) converges to zero for any $i, j \in 1, 2, \ldots, n$ [40]. Therefore, for ZMMMI model (10), the state matrix $X(t)$ can converge to the time-variant theoretical inverse $A^{-1}(t)$ starting from proper random initial state $X(0)$, e.g., sufficiently close to $A^{-1}(0)$ [36, 41, 42]. Approximately and conditionally, we prove the exponential convergence performance of ZMMMI model (10).

We get the analytic solution of (18) in the following matrix form:

$$W(t) = W(0)e^{-\gamma t}. \tag{20}$$

Then, we further obtain

$$\|W(t)\|_{\mathrm{F}} = \|W(0)\|_{\mathrm{F}}e^{-\gamma t}, \tag{21}$$

where symbol $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm of a matrix and e represents Euler number. According to (21), as $t \longrightarrow \infty$, $\|W(t)\|_{\mathrm{F}} \longrightarrow 0$ with rate $\gamma > 0$ exponentially. The proof of exponential convergence of ZMMMI model (10) is thus completed. □

3.2. Discrete Algorithms. The hardware implementation of the continuous-time ZMMMI model (10) needs discrete algorithm. In this section, we discuss five discrete algorithms for the continuous-time ZMMMI model (10).

3.2.1. Euler Forward Formula-Based Algorithm. For simplicity, the following Euler forward difference method is referred [43, 44]:

$$\dot{X}(t_k) = \frac{1}{\epsilon}\left(X(t_{k+1}) - X(t_k)\right) + O(\epsilon), \tag{22}$$

where $\epsilon > 0$ represents the sampling gap and $k = 0, 1, 2, \ldots$ represents the iteration number. $t_k = k\epsilon$ denotes the sampling time at iteration number k. For simplicity, we denote $X_k = X(t_k)$, and then (22) is transformed as

$$\dot{X}_k = \frac{1}{\epsilon}\left(X_{k+1} - X_k\right) + O(\epsilon). \tag{23}$$

With (23), we discretize the continuous-time ZMMMI model (10) as

$$\frac{1}{\epsilon}\left(X_{k+1} - X_k\right) = X_k\left(-\dot{A}_k X_k A_k - A_k X_k \dot{A}_k + \dot{A}_k - \gamma\left(A_k X_k A_k - A_k\right)\right)X_k + O(\epsilon), \tag{24}$$

which is further formulated as

$$X_{k+1} \doteq X_k + \epsilon X_k\left(-\dot{A}_k X_k A_k - A_k X_k \dot{A}_k + \dot{A}_k\right)X_k - hX_k\left(A_k X_k A_k - A_k\right)X_k, \tag{25}$$

where $h = \epsilon\gamma > 0$ denotes sufficiently small step length. Because it only depends on two sampling points in the past time, algorithm (25) is named as ZMMMI2i algorithm. It is worth pointing out that $\epsilon > 0$ should be set sufficiently small to ensure the convergence of the discrete model of ZMMMI model (10).

3.2.2. Taylor–Zhang Discretization Formula-Based Algorithm. According to [45], TZDF is presented as

$$\dot{u}_k = \frac{1}{\epsilon}\left(u_{k+1} - \frac{3}{2}u_k + u_{k-1} - \frac{1}{2}u_{k-2}\right) + O\left(\epsilon^2\right). \tag{26}$$

With (10) and (26), the four-instant ZMMMI algorithm is named as ZMMMI4i algorithm, which is expressed as follows:

$$\begin{aligned} X_{k+1} \doteq{}& \epsilon X_k\left(-\dot{A}_k X_k A_k - A_k X_k \dot{A}_k + \dot{A}_k\right)X_k - hX_k\left(A_k X_k A_k - A_k\right)X_k \\ &+ \frac{3}{2}X_k - X_{k-1} + \frac{1}{2}X_{k-2}, \end{aligned} \tag{27}$$

where $h > 0$ denotes the step size again as before.

With (10) and (28), the six-instant ZMMMI algorithm is named as ZMMMI6i algorithm, which is given as follows:

3.2.3. Six-Instant ZTD Formula-Based Algorithm. In [37], a six-instant ZTD formula is given as follows:

$$\begin{aligned} \dot{u}_k = \frac{1}{\epsilon}\Big(&\frac{1}{2}u_{k+1} - \frac{5}{48}u_k - \frac{1}{4}u_{k-1} - \frac{1}{8}u_{k-2} \\ &- \frac{1}{12}u_{k-3} + \frac{1}{16}u_{k-4}\Big) + O\left(\epsilon^3\right). \end{aligned} \tag{28}$$

$$\begin{aligned} X_{k+1} \doteq{}& 2\epsilon X_k\left(-\dot{A}_k X_k A_k - A_k X_k \dot{A}_k + \dot{A}_k\right)X_k - 2hX_k\left(A_k X_k A_k - A_k\right)X_k \\ &+ \frac{5}{24}X_k + \frac{1}{2}X_{k-1} + \frac{1}{4}X_{k-2} + \frac{1}{6}X_{k-3} - \frac{1}{8}X_{k-4}, \end{aligned} \tag{29}$$

where $h > 0$ denotes the step size again as before.

With (10) and (30), the eight-instant ZMMMI algorithm is named as ZMMMI8i algorithm, which is given as follows:

3.2.4. Eight-Instant ZTD Formula-Based Algorithm. In [37], an eight-instant ZTD formula is given as follows:

$$\begin{aligned} \dot{u}_k = \frac{1}{111\epsilon}\Big(&50u_{k+1} - \frac{51}{10}u_k - 20u_{k-1} - 30u_{k-2} \\ &- 10u_{k-3} + \frac{35}{4}u_{k-4} + \frac{44}{5}u_{k-5} - 5u_{k-6}\Big) + O\left(\epsilon^4\right). \end{aligned} \tag{30}$$

$$\begin{aligned} X_{k+1} \doteq{}& \epsilon X_k\left(-\dot{A}_k X_k A_k - A_k X_k \dot{A}_k + \dot{A}_k\right)X_k - hX_k\left(A_k X_k A_k - A_k\right)X_k \\ &+ \frac{111}{50}\left(\frac{51}{10}X_k + 20X_{k-1} + 30X_{k-2} + 10X_{k-3} - \frac{35}{4}X_{k-4} - \frac{44}{5}X_{k-5} + 5X_{k-6}\right). \end{aligned} \tag{31}$$

3.2.5. Ten-Instant ZTD Formula-Based Algorithm. In [25], one ten-instant formula is given as follows:

$$\dot{u}_k = \frac{1}{\epsilon}\sum_{i=0}^{9} c_i u_{k+1-i} + O\left(\epsilon^5\right), \tag{32}$$

where $c_0 = 100/237$, $c_1 = 947/66360$, $c_2 = -20/237$, $c_3 = -100/237$, $c_4 = -10/79$, $c_5 = 40/237$, $c_6 = 88/1185$, $c_7 = 10/237$, $c_8 = -230/1659$, and $c_9 = 95/1896$ are the coefficients presented in [25]. With (10) and (32), the ten-instant ZMMMI algorithm is named as ZMMMI10i algorithm, which is thus obtained:

$$X_{k+1} \doteq \frac{1}{c_0}\left(\epsilon X_k\left(-\dot{A}_k X_k A_k - A_k X_k \dot{A}_k + \dot{A}_k\right)X_k - hX_k\left(A_k X_k A_k - A_k\right)X_k\right) - \sum_{i=1}^{9} c_i X_{k+1-i}, \tag{33}$$

where $h = \epsilon\gamma$. It should be noticed that when h is a given constant, the truncation error is consistent with $O(\epsilon^6)$. When h is not fixed and γ is a constant, the error is upgraded to $O(\epsilon^5)$ as a result.

3.3. Steady-State Residual Errors of Discrete Algorithms. Next, a theorem is given to show that the steady-state residual errors of $\lim_{k\longrightarrow+\infty}\sup\|A_{k+1}X_{k+1}A_{k+1} - A_{k+1}\|_F$ are equivalent to the precision of the corresponding discrete algorithm in this section.

Theorem 1. *Consider a smoothly time-variant real matrix $A(t) \in \mathbb{R}^{n\times n}$ of full rank. With sufficiently small sampling gap $\epsilon \in (0,1)$, the maximal steady-state residual error $\lim_{k\longrightarrow+\infty}\sup\|A_{k+1}X_{k+1}A_{k+1} - A_{k+1}\|_F$ of discrete algorithm (25), (27), (29), (31), or (33) is $O(\epsilon^p)$, where $p = 2, 3, 4, 5$ or 6.*

Proof. According to Theorems 1 and 2 in [46], we know the fact that (25) is 0-stable, consistent, and convergent. As a result, it converges with the order of its truncation error. Assume that B_{k+1} is the exact solution of $A_{k+1}B_{k+1} = I$. According to (25), (27), (29), (31), and (33), we have $X_{k+1} = B_{k+1} + O(\epsilon^p)$, with $\epsilon \in (0,1)$, and further have

$$\begin{aligned}
&= \lim_{k\longrightarrow+\infty}\sup\left\|A_{k+1}X_{k+1}A_{k+1} - A_{k+1}\right\|_F \\
&= \lim_{k\longrightarrow+\infty}\sup\left\|A_{k+1}\left(B_{k+1} + O\left(\epsilon^p\right)\right)A_{k+1} - A_{k+1}\right\|_F \\
&= \lim_{k\longrightarrow+\infty}\sup\left\|\left(I + A_{k+1}O\left(\epsilon^p\right)\right)A_{k+1} - A_{k+1}\right\|_F \\
&= \lim_{k\longrightarrow+\infty}\sup\left\|A_{k+1} + A_{k+1}^2 O\left(\epsilon^p\right) - A_{k+1}\right\|_F \\
&= \lim_{k\longrightarrow+\infty}\sup\left\|A_{k+1}^2 O\left(\epsilon^p\right)\right\|_F
\end{aligned} \tag{34}$$

The proof is thus completed. □

4. GMDS Model and Algorithms

This section focuses on constructing new ZF from PEs and then derives continuous-time PGMDS solution model for matrix inverse based on ZNN design formula. Furthermore, we develop five discrete algorithms for continuous-time PGMDS model by exploiting the multiple-instant ZTD formulas.

4.1. Continuous-Time Model from PEs. In this section, we choose (1b) to be the ZF as

$$Z(t) = X(t)A(t)X(t) - X(t), \tag{35}$$

and when $t \longrightarrow +\infty$, $Z(t) \longrightarrow 0$ theoretically. Then, we take the derivative of both sides of (1b) and get

$$\dot{Z}(t) = \dot{X}(t)A(t)X(t) + X(t)\dot{A}(t)X(t) + X(t)A(t)\dot{X}(t) - \dot{X}(t). \tag{36}$$

Substituting (35) and (36) into (4), we obtain

$$\begin{aligned}
&\dot{X}(t)A(t)X(t) + X(t)\dot{A}(t)X(t) + X(t)A(t)\dot{X}(t) - \dot{X}(t) \\
&\quad = -\gamma\left(X(t)A(t)X(t) - X(t)\right).
\end{aligned} \tag{37}$$

According to the fact that $A(t)X(t) = X(t)A(t) = I$, we reformulate (36) and then obtain

$$\dot{X}(t) = -X(t)\dot{A}(t)X(t) - \gamma\left(X(t)A(t)X(t) - X(t)\right). \tag{38}$$

Model (38), which is derived from PE (1b), is exactly the Getz–Marsden dynamic system mentioned in [36]. Therefore, the model is named as continuous-time PGMDS model (38).

4.2. Discrete Algorithms. In this part, we discretize the continuous-time PGMDS model (38) with the following five discrete formulas as those in Section 3.

4.2.1. Euler Forward Formula-Based Algorithm. First, (38) is discretized as

$$\dot{X}_k = -X_k\dot{A}_k X_k - \gamma\left(X_k A_k X_k - X_k\right) + O(\epsilon). \tag{39}$$

With (39) and (23), the two-instant PGMDS algorithm is named as PGMDS2i algorithm, which is expressed as follows:

$$X_{k+1} \doteq X_k - \epsilon X_k\dot{A}_k X_k - h\left(X_k A_k X_k - X_k\right), \tag{40}$$

where $h = \epsilon\gamma > 0$ denotes the step length as before.

4.2.2. Taylor–Zhang Discretization Formula-Based Algorithm. With (39) and (26), the four-instant PGMDS

algorithm is named as PGMDS4i algorithm, which is expressed as follows:

$$X_{k+1} \doteq -\epsilon X_k \dot{A}_k X_k - h\left(X_k A_k X_k - X_k\right) + \frac{3}{2}X_k - X_{k-1} + \frac{1}{2}X_{k-2}, \tag{41}$$

where $h > 0$ denotes the step size again as before.

4.2.3. Six-Instant ZTD Formula-Based Algorithm. With (39) and (28), the six-instant PGMDS algorithm is named as PGMDS6i algorithm, which is given as follows:

$$\begin{aligned} X_{k+1} \doteq & -\epsilon 2 X_k \dot{A}_k X_k - 2h\left(X_k A_k X_k - X_k\right) + \frac{5}{24}X_k \\ & + \frac{1}{2}X_{k-1} + \frac{1}{4}X_{k-2} + \frac{1}{6}X_{k-3} - \frac{1}{8}X_{k-4}, \end{aligned} \tag{42}$$

where $h > 0$ denotes the step size again as before.

4.2.4. Eight-Instant ZTD Formula-Based Algorithm. With (39) and (30), the eight-instant PGMDS algorithm is named as PGMDS8i algorithm, which is given as follows:

$$\begin{aligned} X_{k+1} \doteq & -\epsilon X_k \dot{A}_k X_k - h\left(X_k A_k X_k - X_k\right) + \frac{111}{50}\left(\frac{51}{10}X_k + 20X_{k-1} + 30X_{k-2}\right. \\ & \left. + 10X_{k-3} - \frac{35}{4}X_{k-4} - \frac{44}{5}X_{k-5} + 5X_{k-6}\right), \end{aligned} \tag{43}$$

where $h = \epsilon\gamma > 0$ denotes the step length as before.

4.2.5. Ten-Instant ZTD Formula-Based Algorithm. With (39) and (32), the ten-instant PGMDS algorithm is named as PGMDS10i algorithm, which is thus obtained:

$$X_{k+1} \doteq \frac{1}{c_0}\left(-\epsilon X_k \dot{A}_k X_k - h\left(X_k A_k X_k - X_k\right)\right) - \sum_{i=1}^{9} c_i X_{k+1-i}, \tag{44}$$

where $h = \epsilon\gamma > 0$ denotes the step length as before.

4.3. Steady-State Residual Errors of Discrete Algorithms. Next, a theorem is given to show that the steady-state residual of $\lim_{k\longrightarrow+\infty}\sup\|X_{k+1}A_{k+1}X_{k+1} - X_{k+1}\|_{\mathrm{F}}$ is equivalent to the precision of the corresponding discrete algorithm in this section.

Theorem 2. *Consider a smoothly time-variant real matrix $A(t) \in \mathbb{R}^{n\times n}$ of full rank. With sufficiently small sampling gap $\epsilon \in (0, 1)$, the maximal steady-state residual error $\lim_{k\longrightarrow+\infty}\sup\|X_{k+1}A_{k+1}X_{k+1} - X_{k+1}\|_F$ of discrete algorithm (40)–(44) is $O(\epsilon^p)$, where $p = 2, 3, 4, 5,$ and 6, respectively.*

Proof. With the same conditions as Theorem 1, we know the fact that (39) is 0-stable, consistent, and convergent [46]. As a result, it converges with the order of its truncation error. Assume that B_{k+1} satisfies $A_{k+1}B_{k+1} = I$ and $X_{k+1} = B_{k+1} + O(\epsilon^p)$ with $\epsilon \in (0, 1)$ as before. We further have

$$\begin{aligned} & = \lim_{k\longrightarrow+\infty}\sup\left\|X_{k+1}A_{k+1}X_{k+1} - X_{k+1}\right\|_{\mathrm{F}} \\ & = \lim_{k\longrightarrow+\infty}\sup\left\|\left(B_{k+1} + O\left(\epsilon^p\right)\right)A_{k+1}X_{k+1} - X_{k+1}\right\|_{\mathrm{F}} \\ & = \lim_{k\longrightarrow+\infty}\sup\left\|\left(B_{k+1}A_{k+1} + A_{k+1}O\left(\epsilon^p\right)\right)X_{k+1} - X_{k+1}\right\|_{\mathrm{F}} \\ & = \lim_{k\longrightarrow+\infty}\sup\left\|\left(I + A_{k+1}O\left(\epsilon^p\right)\right)X_{k+1} - X_{k+1}\right\|_{\mathrm{F}} \\ & = \lim_{k\longrightarrow+\infty}\sup\left\|X_{k+1} + A_{k+1}X_{k+1}O\left(\epsilon^p\right) - X_{k+1}\right\|_{\mathrm{F}} \\ & = \lim_{k\longrightarrow+\infty}\sup\left\|A_{k+1}\left(B_{k+1} + O\left(\epsilon^p\right)\right)O\left(\epsilon^p\right)\right\|_{\mathrm{F}} \\ & = \lim_{k\longrightarrow+\infty}\sup\left\|O\left(\epsilon^p\right) + A_{k+1}O\left(\epsilon^{2p}\right)\right\|_{\mathrm{F}} = O\left(\epsilon^p\right). \end{aligned} \tag{45}$$

The proof is thus completed. □

5. DDD Model and Algorithms

This section focuses on constructing direct derivative dynamics from one appropriate Penrose equation and then derives direct derivative solution model for matrix inversion.

5.1. Continuous-Time Model from PEs. We take the derivative on both sides of (1a) and then obtain

$$\dot{A}(t)X(t)A(t) + A(t)\dot{X}(t)A(t) + A(t)X(t)\dot{A}(t) = \dot{A}(t). \tag{46}$$

To get the explicit expression of $\dot{X}(t)$, we reformulate above equation and obtain

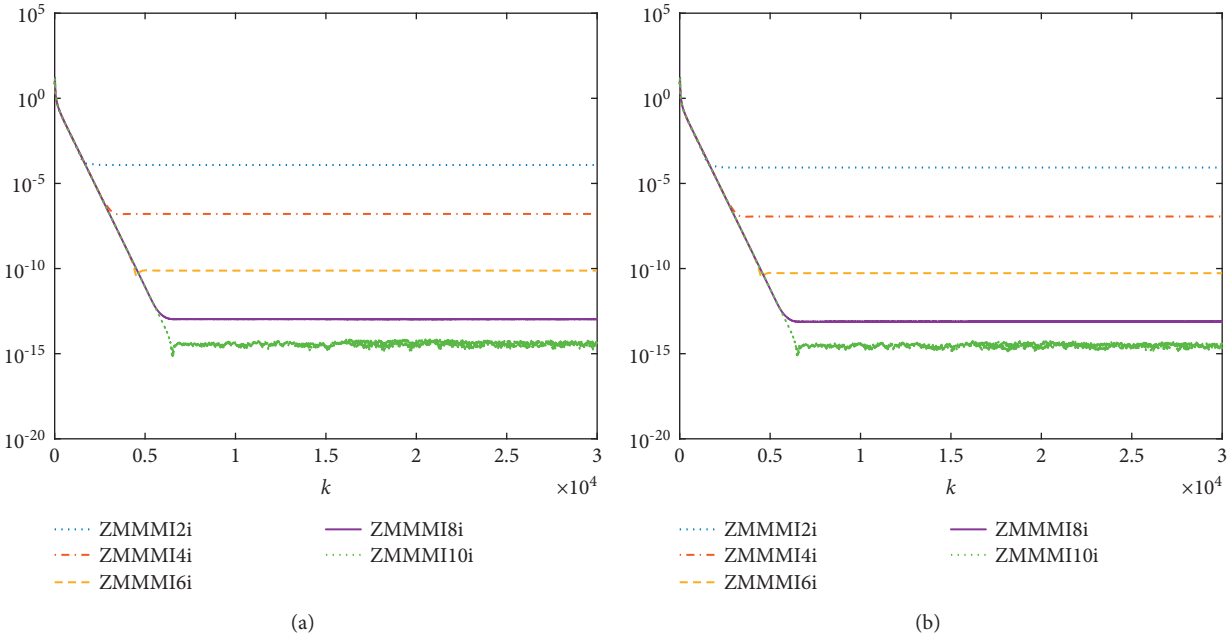

Figure 1: Residual error trajectories synthesized by five ZMMMI algorithms for example 1 (with $\epsilon = 0.001$ s and $\gamma = 5.0$). (a) Trajectories of $\|X_{k+1}A_{k+1} - I\|_F$. (b) Trajectories of $\|X_{k+1} - A_{k+1}^{-1}\|_F$.

$$\dot{X}(t) = -A^{-1}(t)\dot{A}(t)X(t) - X(t)\dot{A}(t)A^{-1} \cdot (t) + A^{-1}(t)\dot{A}(t)A^{-1}(t). \tag{47}$$

According to the assumption of the matrix inverse problem, we can substitute $A^{-1}(t)$ with $X(t)$ in (47) and thus obtain

$$\dot{X}(t) = -X(t)\dot{A}(t)X(t) - X(t)\dot{A}(t)X(t) + X(t)\dot{A}(t)X(t). \tag{48}$$

The right-hand side of above equation can be simplified as

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

Finally, we present the continuous-time direct derivative dynamics model (49), which is termed as continuous-time DDD model (49) for short.

5.2. Discrete Algorithms. In this part, we discretize the continuous-time DDD model (49) with the following five discrete formulas as those in Section 3.

5.2.1. Euler Forward Formula-Based Algorithm. First, (49) is discretized as

$$\dot{X}_k = -X_k\dot{A}_k X_k + O(\epsilon). \tag{50}$$

With (50) and (23), the two-instant DDD algorithm is named as DDD2i algorithm, which is expressed as follows:

$$X_{k+1} \doteq X_k - \epsilon X_k \dot{A} X_k. \tag{51}$$

5.2.2. Taylor–Zhang Discretization Formula-Based Algorithm. With (50) and (26), the four-instant DDD algorithm is named as DDD4i algorithm, which is expressed as follows:

$$X_{k+1} \doteq -\epsilon X_k \dot{A}_k X_k + \frac{3}{2}X_k - X_{k-1} + \frac{1}{2}X_{k-2}. \tag{52}$$

5.2.3. Six-Instant ZTD Formula-Based Algorithm. With (50) and (28), the six-instant DDD algorithm is named as DDD6i algorithm, which is given as follows:

$$X_{k+1} \doteq -\epsilon X_k \dot{A}_k X_k + \frac{5}{24}X_k + \frac{1}{2}X_{k-1} + \frac{1}{4}X_{k-2} + \frac{1}{6}X_{k-3} - \frac{1}{8}X_{k-4}. \tag{53}$$

5.2.4. Eight-Instant ZTD Formula-Based Algorithm. With (50) and (30), the eight-instant DDD algorithm is named as DDD8i algorithm, which is given as follows:

$$X_{k+1} \doteq -\epsilon X_k \dot{A}_k X_k + \frac{111}{50}\left(\frac{51}{10}X_k + 20X_{k-1} + 30X_{k-2} + 10X_{k-3} - \frac{35}{4}X_{k-4} - \frac{44}{5}X_{k-5} + 5X_{k-6}\right). \tag{54}$$

5.2.5. Ten-Instant ZTD Formula-Based Algorithm. With (50) and (32), the ten-instant DDD algorithm is named as DDD10i algorithm, which is thus obtained:

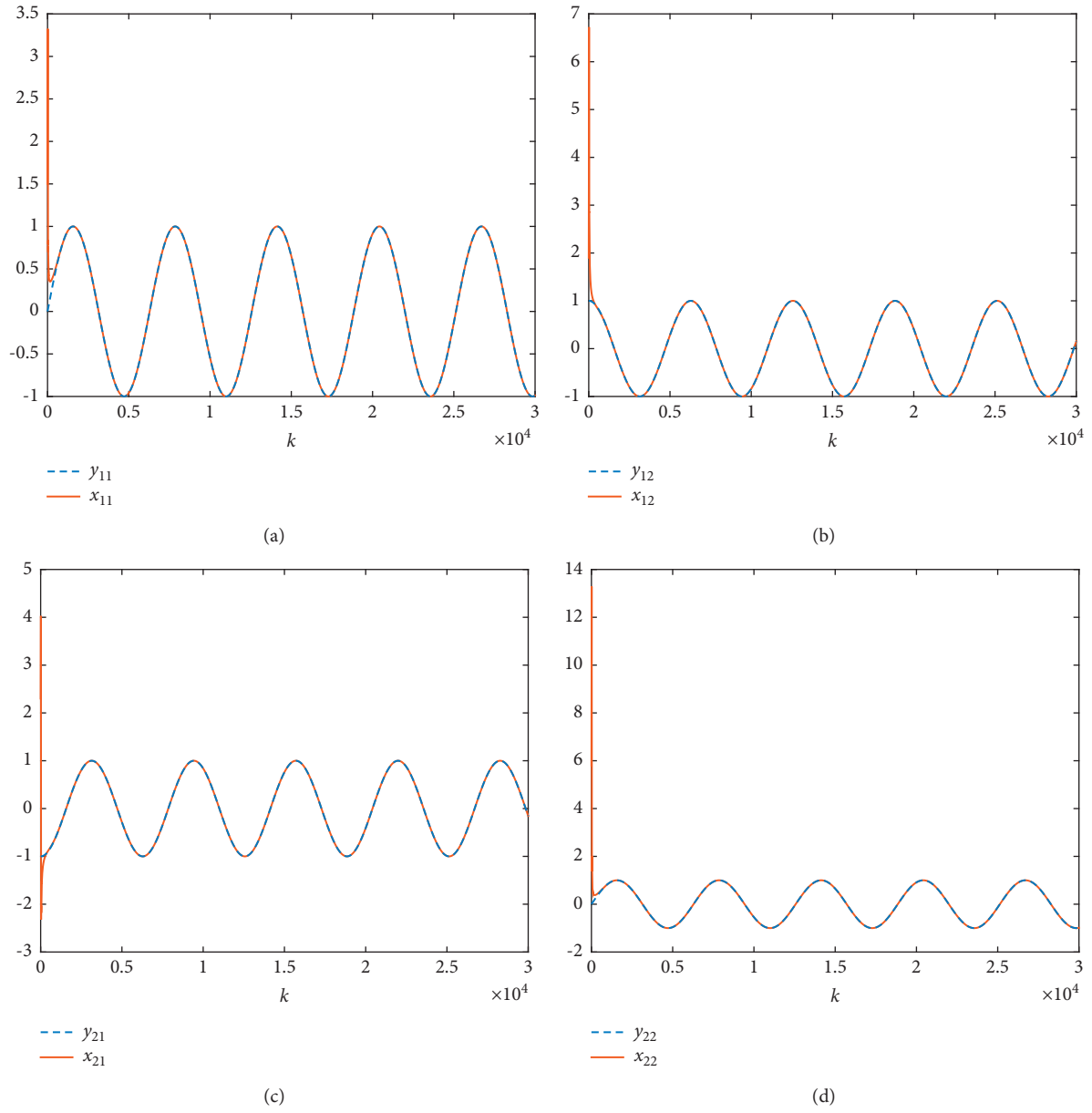

Figure 2: Solution trajectories synthesized by Y_{k+1} (i.e., A_{k+1}^{-1}) and by ZMMMI10i algorithm (33) for example 1 (with $\epsilon = 0.001$ s and $\gamma = 5.0$). (a) Trajectories of y_{11} and x_{11}. (b) Trajectories of y_{12} and x_{12}. (c) Trajectories of y_{21} and x_{21}. (d) Trajectories of y_{22} and x_{22}.

$$X_{k+1} \doteq -\epsilon X_k \dot{A}_k X_k - \sum_{i=1}^{9} c_i X_{k+1-i}. \tag{55}$$

6. Computer Experiments and Results

In this section, computer experiments is carried out to verify the effectiveness of the presented three models, ZMMMI model (10), PGMDS model (38), and DDD model (49), on three time-variant matrix inversion examples.

6.1. Example 1. Let us consider the following discrete-time matrix inversion problem with X_{k+1} to be obtained during $[t_k, t_{k+1})$, of which A_k is defined as

$$A(t_k) = \begin{pmatrix} \sin(t_k) & -\cos(t_k) \\ \cos(t_k) & \sin(t_k) \end{pmatrix} \in \mathbb{R}^{2\times 2}. \tag{56}$$

The task duration (i.e., final time) is uniformly set as $t_d = 30$ s. To verify the computational results, the theoretical inversion of matrix (56) can be obtained as

$$Y(t_k) = A^{-1}(t_k) = \begin{pmatrix} \sin(t_k) & \cos(t_k) \\ -\cos(t_k) & \sin(t_k) \end{pmatrix} \in \mathbb{R}^{2\times 2}, \tag{57}$$

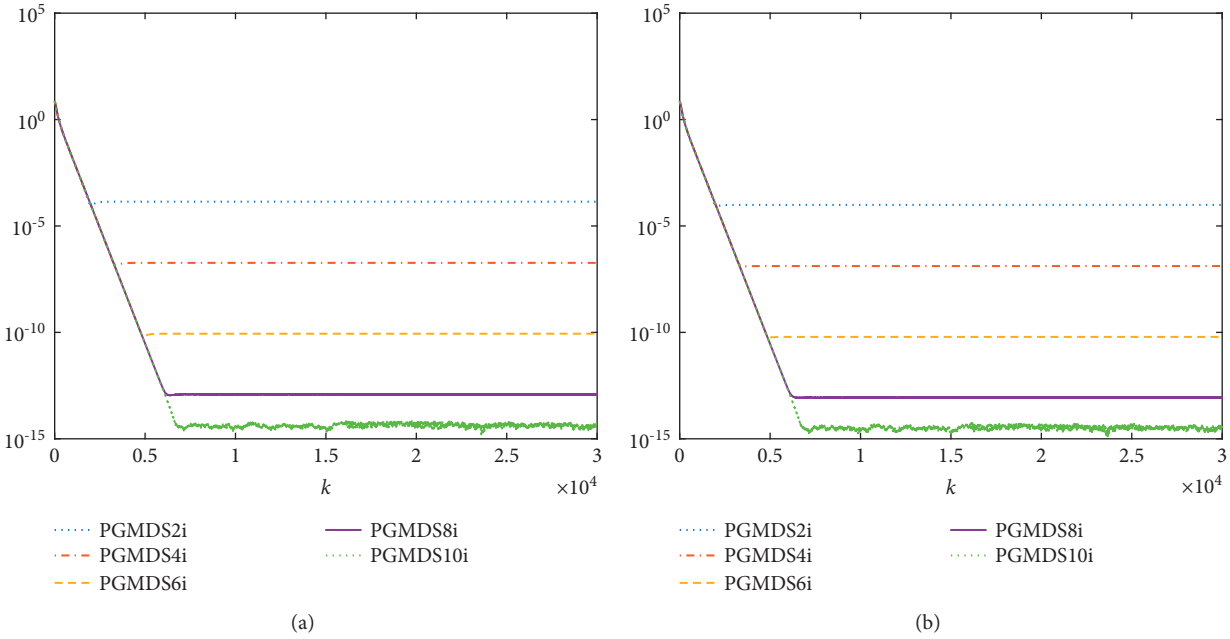

Figure 3: Residual error trajectories synthesized by five PGMDS algorithms for example 1 (with $\epsilon = 0.001$ s and $\gamma = 5.0$). (a) Trajectories of $\|X_{k+1}A_{k+1} - I\|_F$. (b) Trajectories of $\|X_{k+1} - A_{k+1}^{-1}\|_F$.

which is given for checking the effectiveness and correctness of the three models and corresponding algorithms. The initial values of the example are arbitrarily set as

$$Y(0) = \begin{pmatrix} 1.5 & 1.9 \\ 2.3 & 5.8 \end{pmatrix}, \tag{58}$$

for all algorithms.

6.1.1. New Models of Zhang Neural Network. Firstly, we check the effectiveness and the correctness of ZMMMI model (10) with five discrete algorithms, i.e., ZMMMI2i algorithm (25), ZMMMI4i algorithm (27), ZMMMI6i algorithm (29), ZMMMI8i algorithm (31), and ZMMMI10i algorithm (33). The step size and task duration are uniformly set as $h = 0.005$ and $t_d = 30$ s. The corresponding results of computer experiments are shown in Figures 1 and 2. Figure 1(a) shows the residual errors $\|X_{k+1}A_{k+1} - I\|_F$ of the five ZMMMI algorithms, with $\epsilon = 0.001$ s and $h = 0.005$. Figure 1(b) illustrates the residual errors $\|X_{k+1} - A_{k+1}^{-1}\|_F$ of the same five ZMMMI algorithms as before. From the trajectories of all entries in Figures 2(a)–2(d), the solution of the model coincides with the theoretical solution perfectly. In addition, we see that the residual error trajectories synthesized by different ZMMMI algorithms quickly stabilized to the steady-state error level, after undergoing the initial hundreds of recursions.

6.1.2. PGMDS Model. First, we check the effectiveness and the correctness of PGMDS model (38) with five discrete algorithms, i.e., PGMDS2i algorithm (40), PGMDS4i algorithm (41), PGMDS6i algorithm (42), PGMDS8i algorithm (43), and PGMDS10i algorithm (44). The step size and task duration are uniformly set as $h = 0.005$ and $t_d = 30$ s. The corresponding results of computer experiments are shown in Figures 3 and 4. Figure 3(a) shows the residual errors $\|X_{k+1}A_{k+1} - I\|_F$ of the five PGMDS algorithms, with $\epsilon = 0.001$ s and $h = 0.005$. Figure 3(b) illustrates the residual errors $\|X_{k+1} - A_{k+1}^{-1}\|_F$ of the same five PGMDS algorithms as before. From the trajectories of all entries in Figures 4(a)–4(d), the solution of the model coincides with the theoretical solution perfectly. In addition, we see that the residual error trajectories synthesized by different PGMDS algorithms quickly stabilized to the theoretical error level, after undergoing the initial hundreds of recursions.

6.1.3. DDD Model. First, we check the effectiveness and correctness of DDD model (49) with five discrete algorithms, i.e., DDD2i algorithm (51), DDD4i algorithm (52), DDD6i algorithm (53), DDD8i algorithm (54), and DDD10i algorithm (55). The step size and task duration are uniformly set as $h = 0.0001$ and $t_d = 30$ s. The corresponding results of computer experiments are shown in Figures 5 and 6. Figure 5(a) shows the residual errors $\|X_{k+1}A_{k+1} - I\|_F$ of the five DDD algorithms, with $\epsilon = 0.001$ s and $h = 0.005$. Figure 5(b) illustrates the residual errors $\|X_{k+1} - A_{k+1}^{-1}\|_F$ of the same five DDD algorithms as before. From the trajectories of all entries in Figures 6(a) and 6(b), even for the model with the highest accuracy, DDD10i algorithm (55), its solution still cannot converge to the theoretical solution.

6.2. Example 2. The second time-variant matrix is a 3×3 real matrix which is shown as follows:

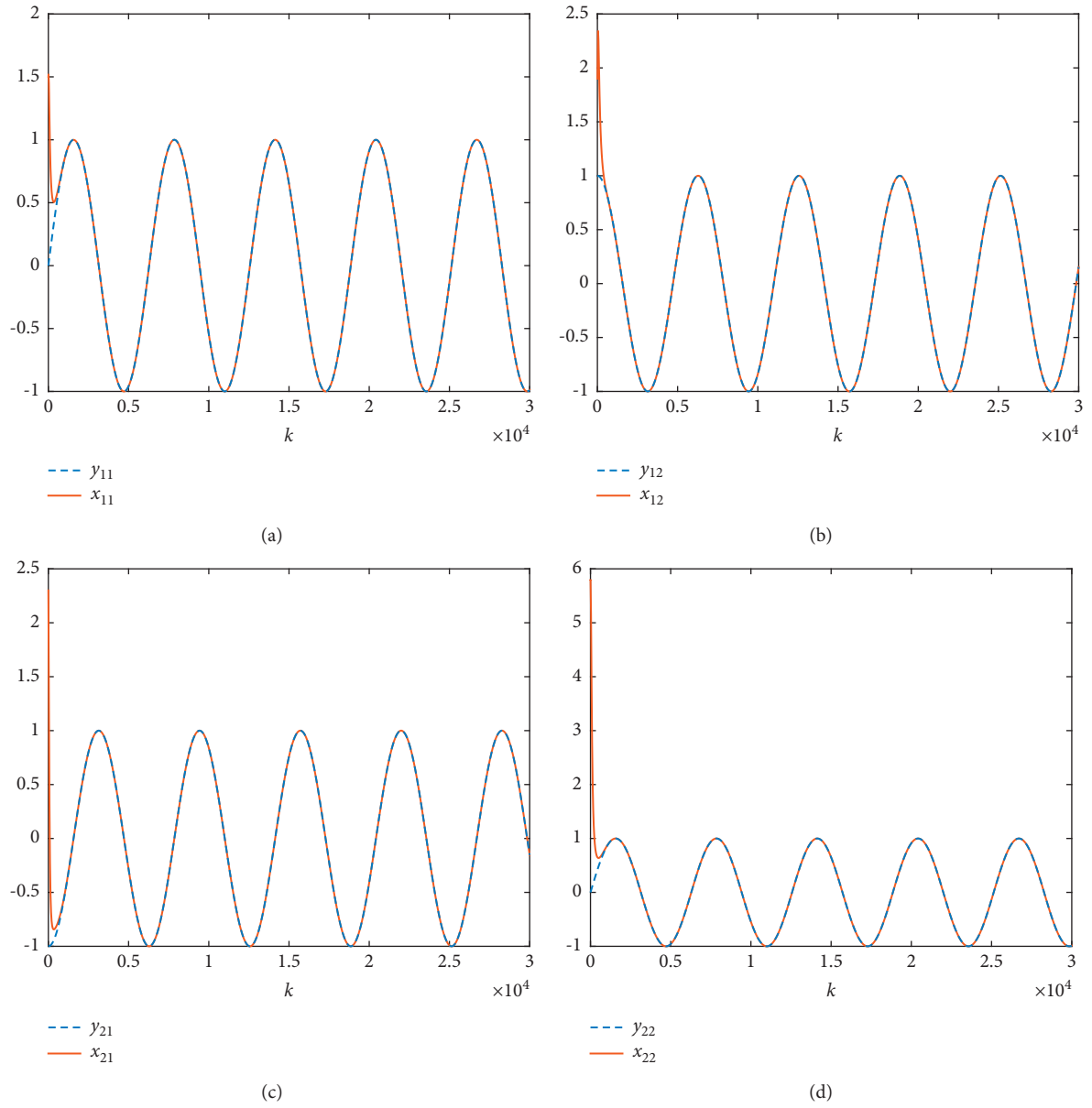

FIGURE 4: Solution trajectories synthesized by Y_{k+1} (i.e., A_{k+1}^{-1}) and by PGMDS10i algorithm (39) for example 1 (with $\epsilon = 0.001$ s and $\gamma = 5.0$). (a) Trajectories of y_{11} and x_{11}. (b) Trajectories of y_{12} and x_{12}. (c) Trajectories of y_{21} and x_{21}. (d) Trajectories of y_{22} and x_{22}.

$$A(t_k) = \begin{pmatrix} 5 + \cos(t_k) & \cos(t_k) & -\sin(t_k) \\ \cos(t_k) & 3 + \sin(t_k) & \sin(t_k) \\ \cos(t_k) & -\cos(t_k) & 2 + \sin(t_k) \end{pmatrix}, \quad (59)$$

where $A(t_k) \in \mathbb{R}^{3\times 3}$. To verify the computational results, we utilize the theoretical inverse of $A(t_k)$, which is denoted as $Y(t_k) = (y_{ij}(t_k))$. Because $Y(t_k)$ is too complicated, it is omitted here. The initial values of the example are arbitrarily set as

$$Y(0) = \begin{pmatrix} 9.5 & 0.7 & 0.6 \\ -1.1 & 12 & 1.3 \\ -1.6 & 0.2 & 9.0 \end{pmatrix}. \quad (60)$$

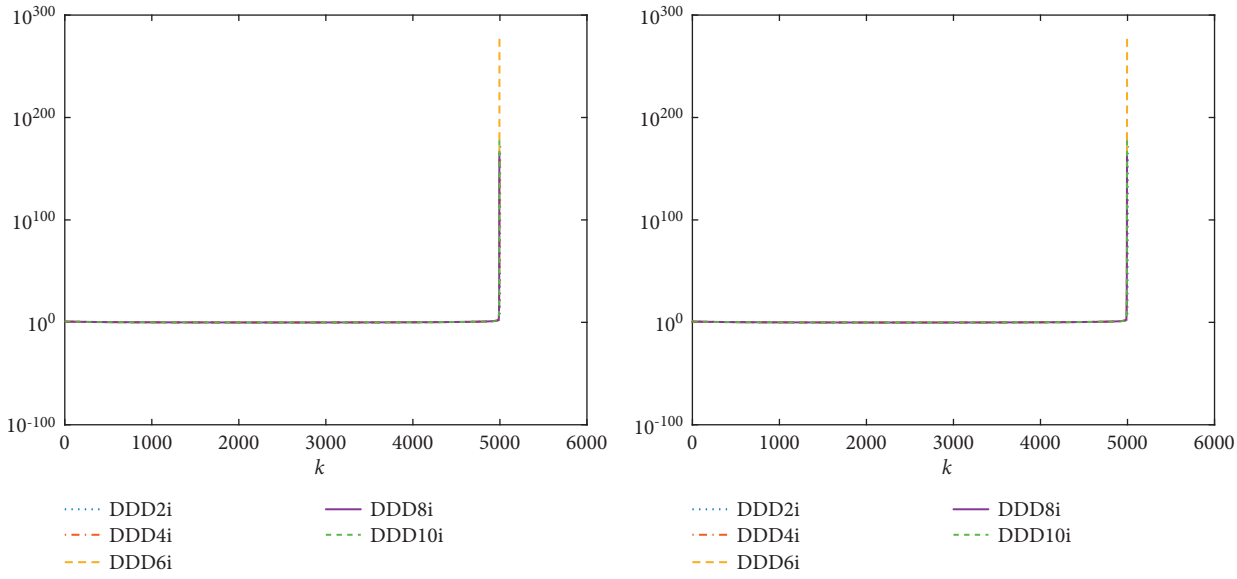

Figure 5: Residual error trajectories synthesized by five DDD algorithms for example 1 (with $\epsilon = 0.001$ s and $\gamma = 5.0$). (a) Trajectories of $\|X_{k+1}A_{k+1} - I\|_F$. (b) Trajectories of $\|X_{k+1} - A_{k+1}^{-1}\|_F$.

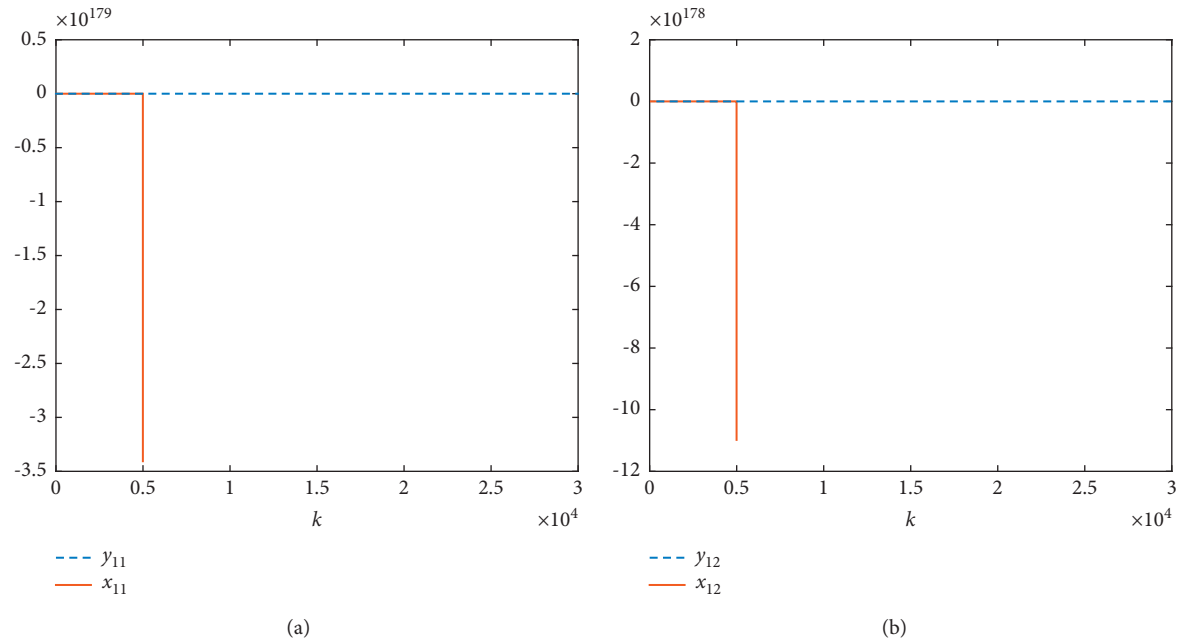

Figure 6: Continued.

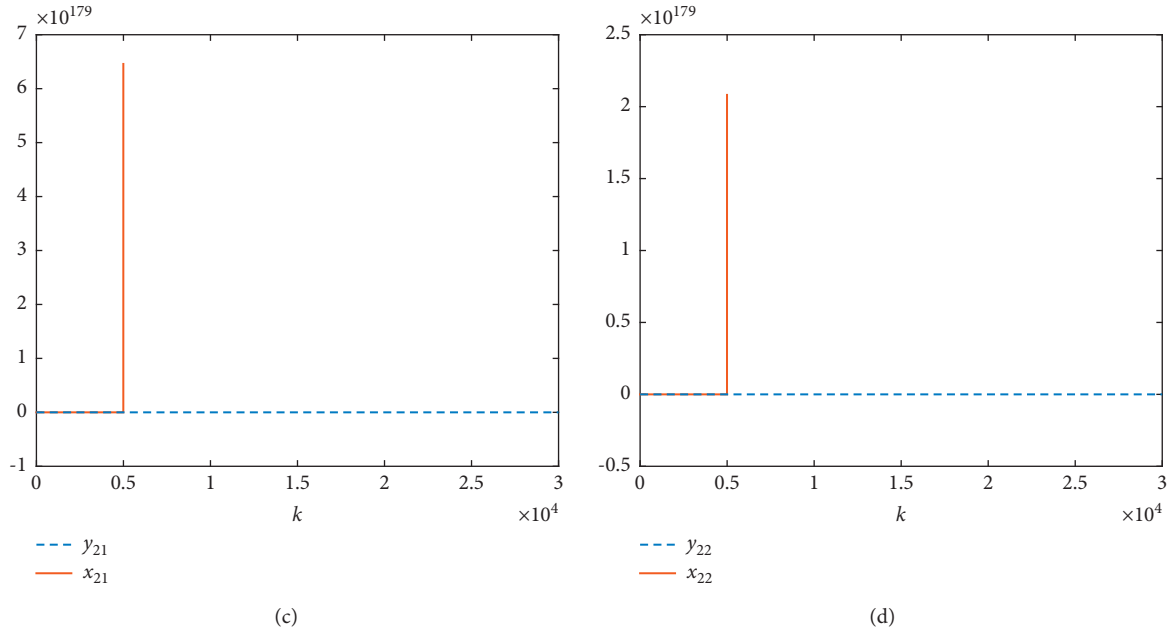

FIGURE 6: Solution trajectories synthesized by Y_{k+1} (i.e., A_{k+1}^{-1}) and by DDD10i algorithm (55) for example 1 (with $\epsilon = 0.001$ s and $\gamma = 5.0$). (a) Trajectories of y_{11} and x_{11}. (b) Trajectories of y_{12} and x_{12}. (c) Trajectories of y_{21} and x_{21}. (d) Trajectories of y_{22} and x_{22}.

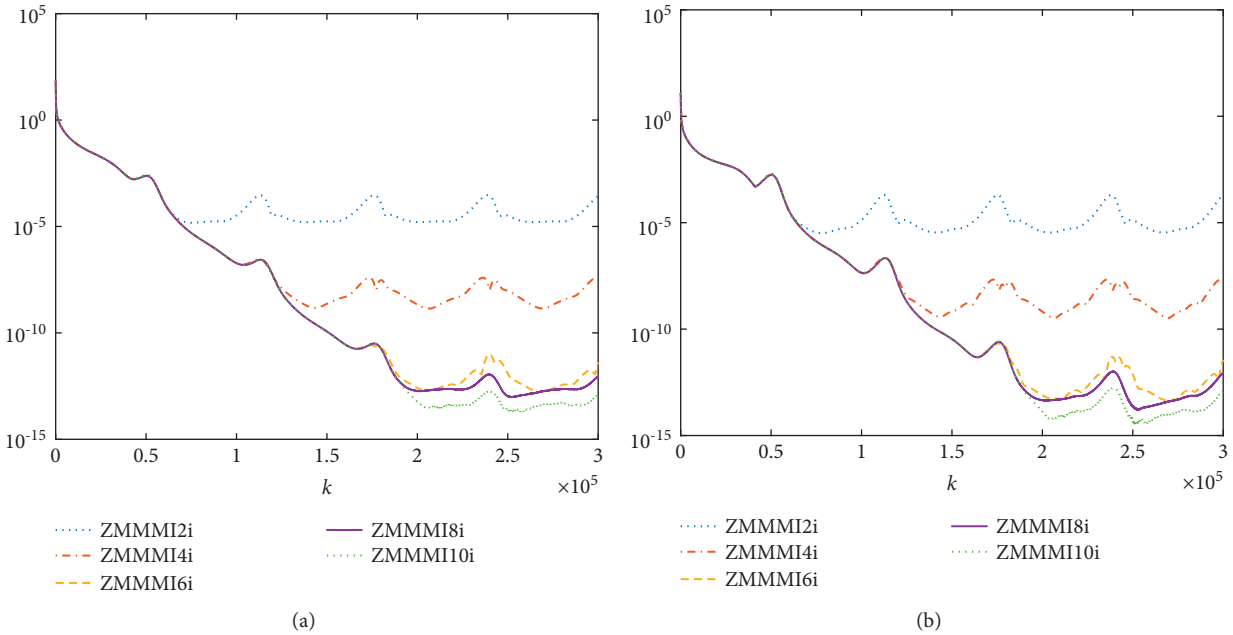

FIGURE 7: Residual error trajectories synthesized by five ZMMMI algorithms for example 2 (with $\epsilon = 0.0001$ s and $\gamma = 2.0$). (a) Trajectories of $\|X_{k+1}A_{k+1} - I\|_F$. (b) Trajectories of $\|X_{k+1} - A_{k+1}^{-1}\|_F$.

The task duration is uniformly set as $t_d = 30$ s.

6.2.1. ZMMMI Model. The results of the numerical experiments for ZMMMI2i algorithm (25), ZMMMI4i algorithm (27), ZMMMI6i algorithm (29), ZMMMI8i algorithm (31), and ZMMMI10i algorithm (33) are shown in Figures 7 and 8. The experimental results are satisfactory as expected.

6.2.2. PGMDS Model. The results of the numerical experiments for PGMDS2i algorithm (40), PGMDS4i

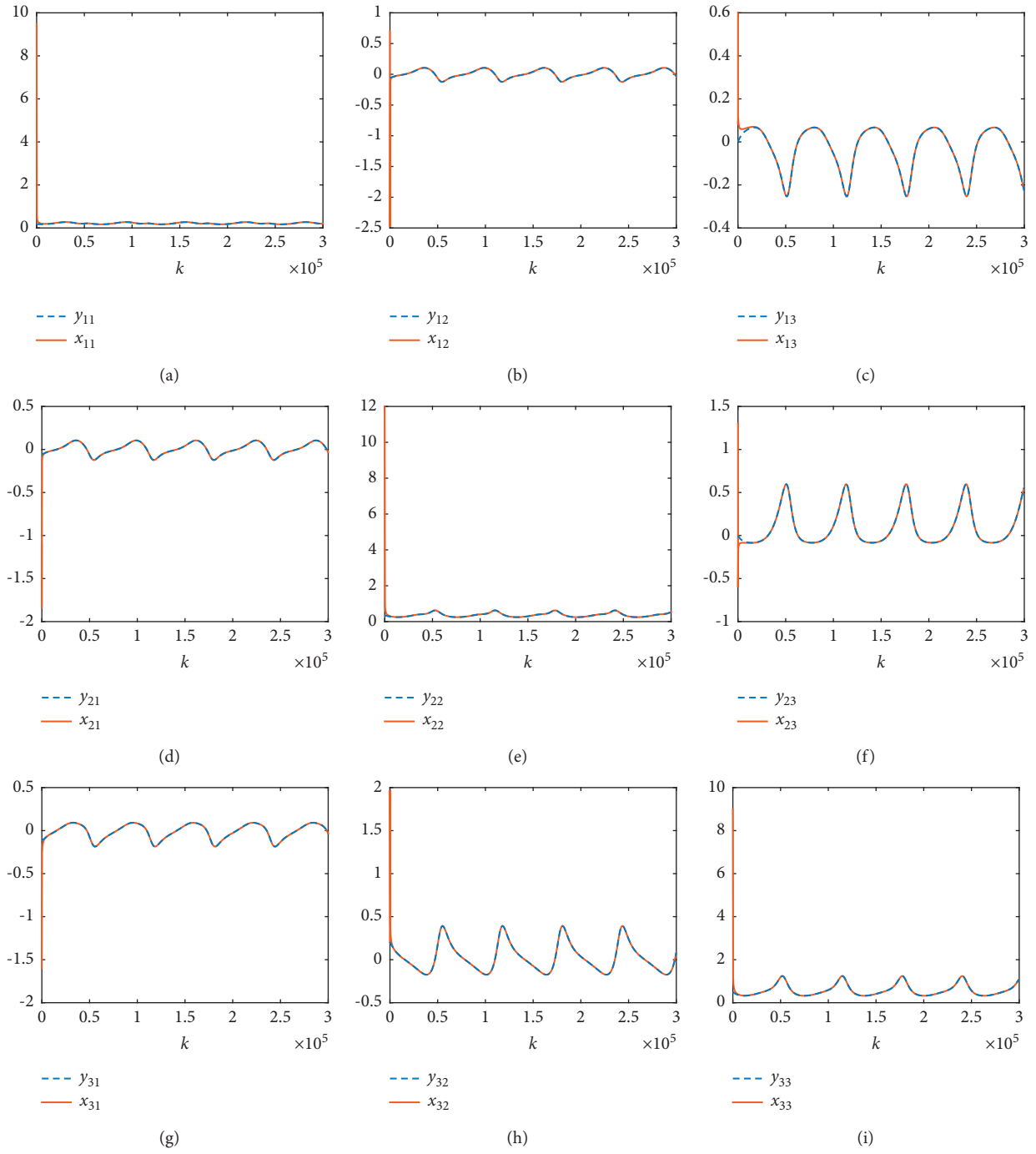

FIGURE 8: Solution trajectories synthesized by Y_{k+1} (i.e., A_{k+1}^{-1}) and by ZMMMI10i algorithm (33) for example 2 (with $\epsilon = 0.0001$ s and $\gamma = 2.0$). (a) Trajectories of y_{11} and x_{11}. (b) Trajectories of y_{12} and x_{12}. (c) Trajectories of y_{13} and x_{13}. (d) Trajectories of y_{21} and x_{21}. (e) Trajectories of y_{22} and x_{22}. (f) Trajectories of y_{23} and x_{23}. (g) Trajectories of y_{31} and x_{31}. (h) Trajectories of y_{32} and x_{32}. (i) Trajectories of y_{33} and x_{33}.

algorithm (41), PGMDS6i algorithm (42), PGMDS8i algorithm (43), and PGMDS10i algorithm (44) are illustrated in Figures 9 and 10. The experimental results are satisfactory as expected.

6.2.3. DDD Model. The results of the numerical experiments for DDD2i algorithm (51), DDD4i algorithm (52), DDD6i algorithm (53), DDD8i algorithm (54), and DDD10i algorithm (55) are shown in Figures 11 and 12. In terms of

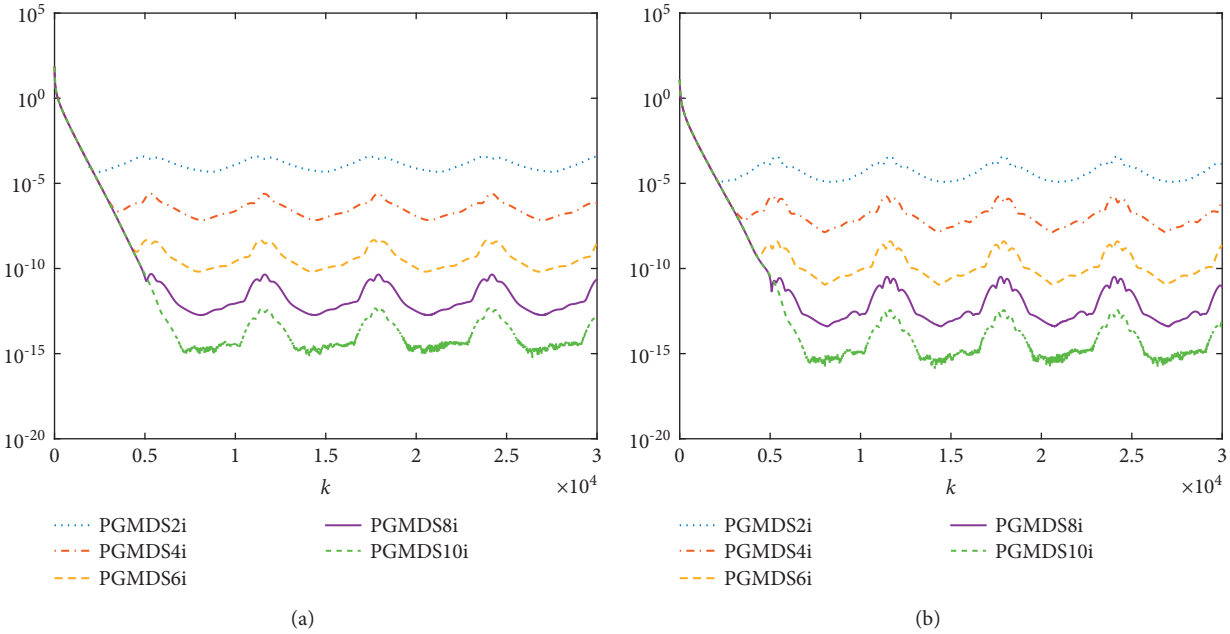

FIGURE 9: Residual error trajectories synthesized by five PGMDS algorithms for example 2 (with $\epsilon = 0.001$ s and $\gamma = 5.0$). (a) Trajectories of $\|X_{k+1}A_{k+1} - I\|_F$. (b) Trajectories of $\|X_{k+1} - A_{k+1}^{-1}\|_F$.

convergence and accuracy, the experimental results in this example are consistent with those in example 1.

6.3. Example 3. This example compares solely the effective models, i.e., ZMMMI (10) and PGMDS (38), with a 4×4 real matrix, which is shown as follows:

$$A(t_k) = \begin{pmatrix} 6 + c_k & c_k & c_k & c_k \\ c_k & 5 + s_k & -s_k & c_k \\ c_k & c_k & 4 + s_k & -s_k \\ s_k & s_k & c_k & 2 + s_k \end{pmatrix}, \tag{61}$$

where $A(t_k) \in \mathbb{R}^{4\times 4}$ and s_k and c_k denote $\sin(t_k)$ and $\cos(t_k)$, respectively. To verify the computational results, we utilize the theoretical inverse of $A(t_k)$, which is denoted as $Y(t_k) = (y_{ij}(t_k))$. Because $Y(t_k)$ is too complicated, it is omitted here. The initial values of the example are arbitrarily set as

$$Y(0) = \begin{pmatrix} 6 & 1 & 1 & 1 \\ 1 & 5 & 0 & 1 \\ 1 & 1 & 4 & 0 \\ 0 & 0 & 1 & 2 \end{pmatrix}. \tag{62}$$

The task duration is uniformly set as $t_d = 30$ s.

6.3.1. ZMMMI Model. The results of the numerical experiments for ZMMMI2i algorithm (25), ZMMMI4i algorithm (27), ZMMMI6i algorithm (29), ZMMMI8i algorithm (31), and ZMMMI10i algorithm (33) are shown in Figures 13(a) and 13(b). The experimental results are satisfactory as expected.

6.3.2. PGMDS Model. The results of the numerical experiments for PGMDS2i algorithm (40), PGMDS4i algorithm (41), PGMDS6i algorithm (42), PGMDS8i algorithm (43), and PGMDS10i algorithm (44) are illustrated in Figures 14(a) and 14(b). The experimental results are satisfactory as expected.

6.3.3. Comparison and Discussion. We mainly investigate the residual error of the models and the coincidence between the solution matrix with the ground-truth matrix inverse. From the discrete simulation results of two examples of each five discrete algorithms, it can be seen that the convergence of ZMMMI model (10) and PGMDS model (38) is good, which is completely consistent with the conclusion of Proposition 1 and Theorems 1 and 2. Therefore, it can be concluded that the two models and corresponding discrete algorithms are effective. However, in the case of DDD model (49), the residual errors and the coincidence of the solution matrix entries are not satisfactory, so we evidently summarize that the effectiveness of DDD model (49) and corresponding discrete algorithms is not enough.

6.3.4. Remark. Note that the convergence of DDD models is shown in [12, 47], where the DDD models are utilized to solve the time-varying nonlinear optimization problems. There are three points worth further discussing. First, in [12], the initial value is set to be the theoretical value, and the

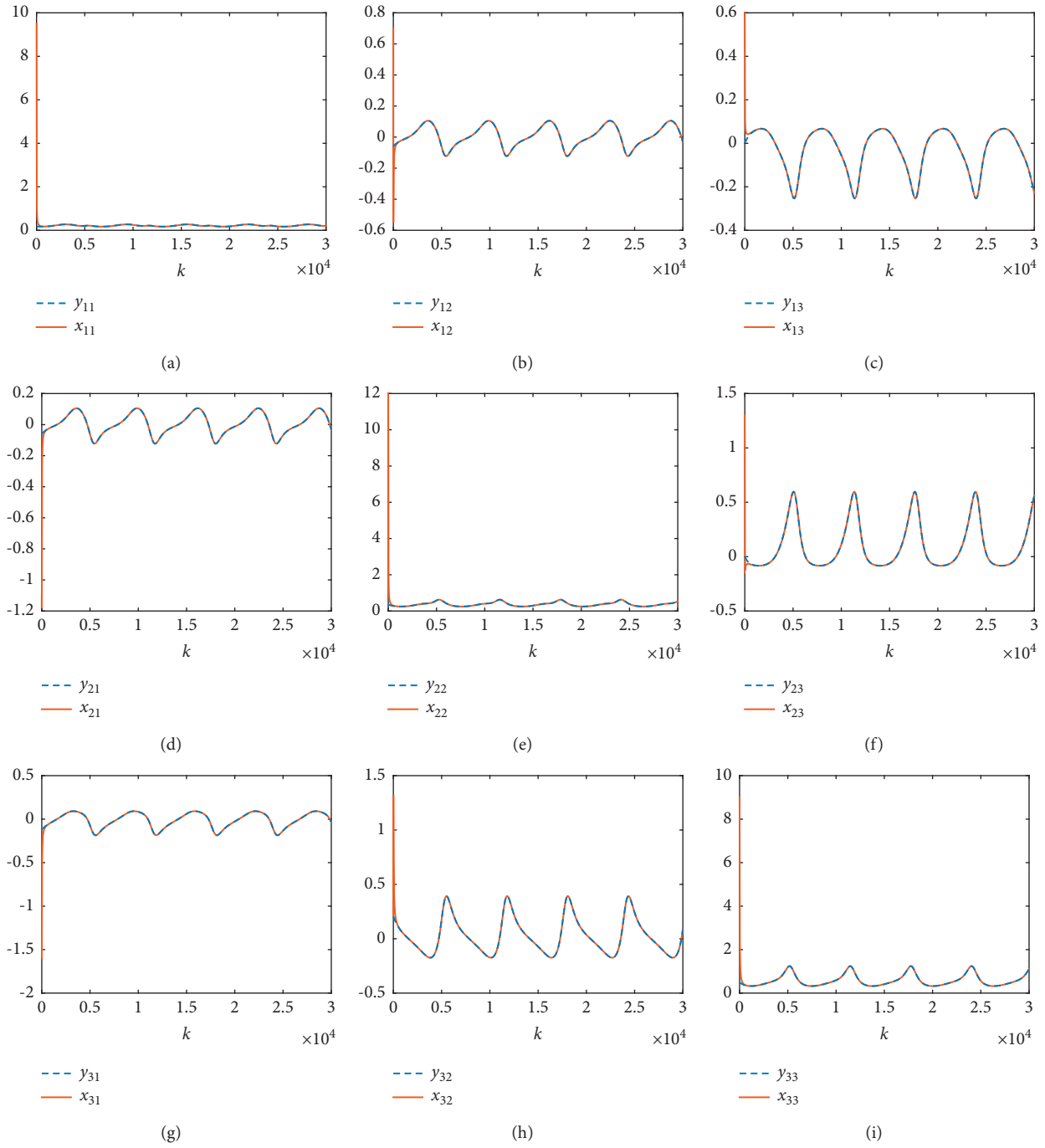

FIGURE 10: Solution trajectories synthesized by Y_{k+1} (i.e., A_{k+1}^{-1}) and by PGMDS10i algorithm (44) for example 2 (with $\epsilon = 0.001$ s and $\gamma = 5.0$). (a) Trajectories of y_{11} and x_{11}. (b) Trajectories of y_{12} and x_{12}. (c) Trajectories of y_{13} and x_{13}. (d) Trajectories of y_{21} and x_{21}. (e) Trajectories of y_{22} and x_{22}. (f) Trajectories of y_{23} and x_{23}. (g) Trajectories of y_{31} and x_{31}. (h) Trajectories of y_{32} and x_{32}. (i) Trajectories of y_{33} and x_{33}.

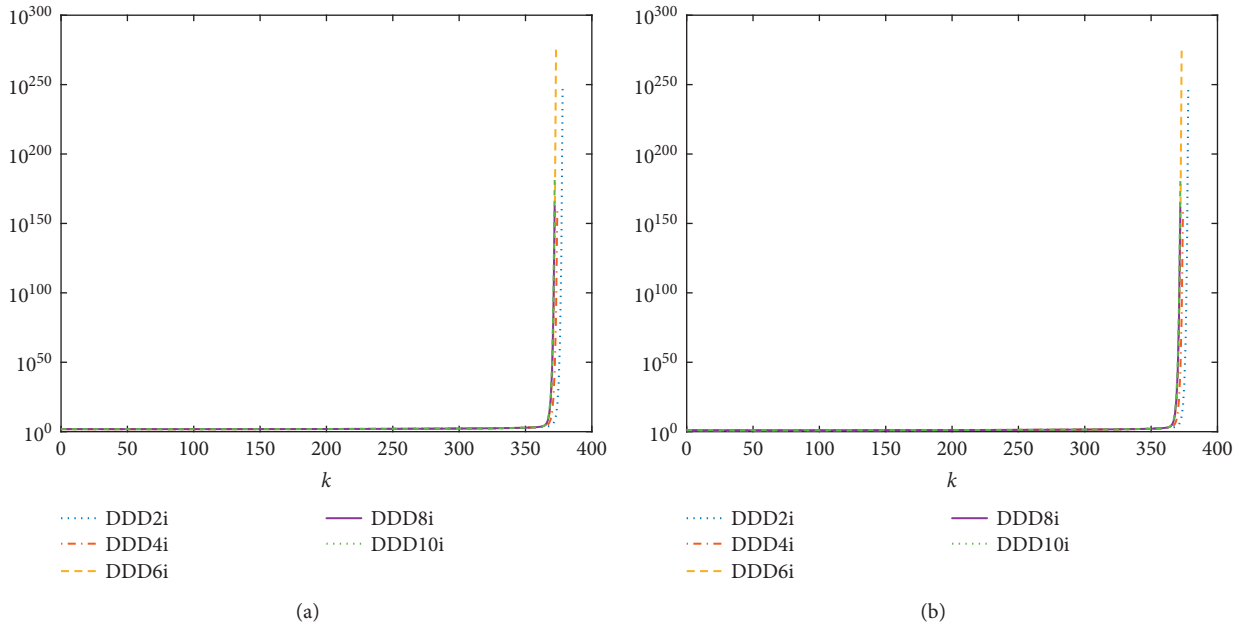

Figure 11: Residual error trajectories synthesized by five DDD algorithms for example 2 (with $\epsilon = 0.001$ s and $\gamma = 1.0$). (a) Trajectories of $\|X_{k+1}A_{k+1} - I\|_F$. (b) Trajectories of $\|X_{k+1} - A_{k+1}^{-1}\|_F$.

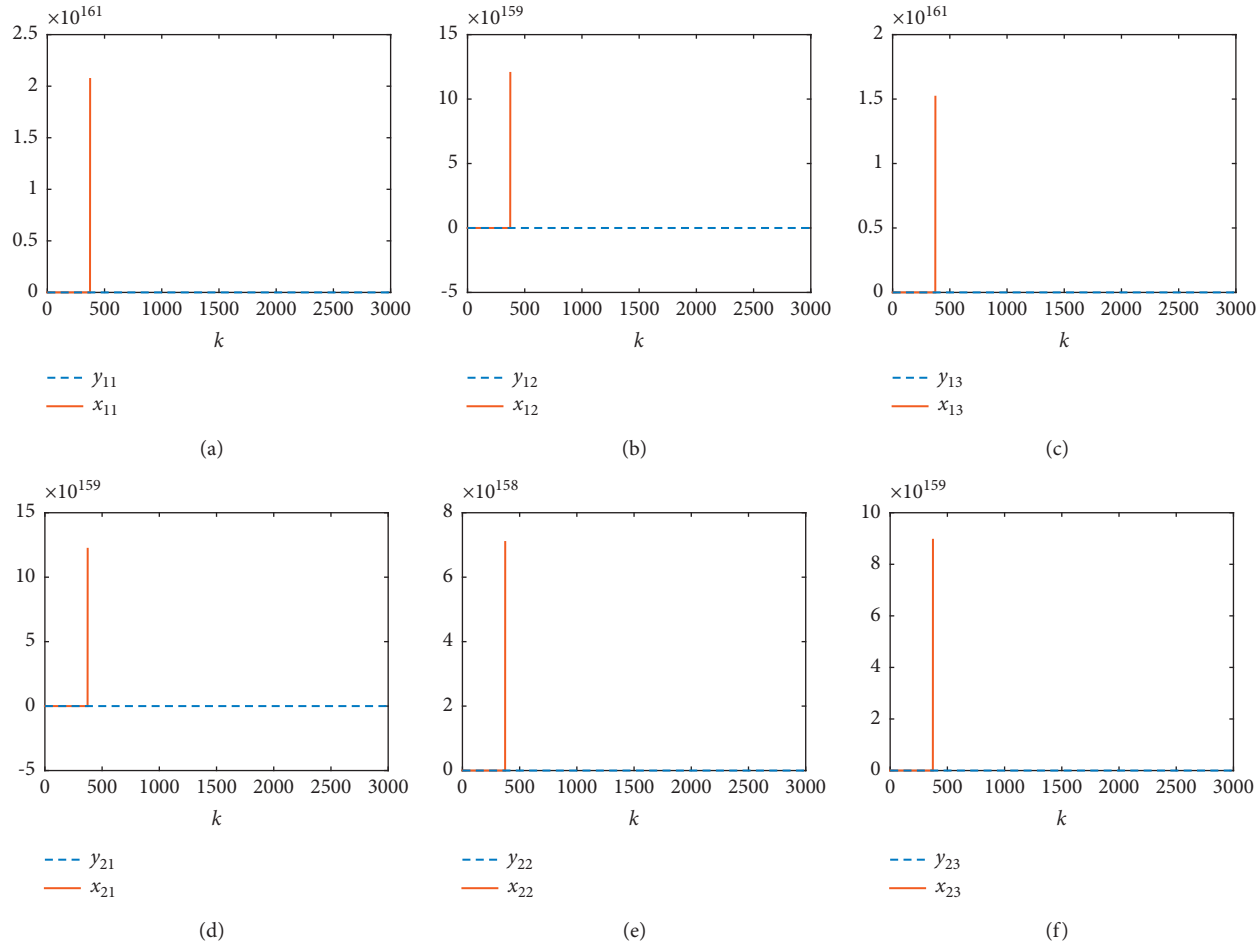

Figure 12: Continued.

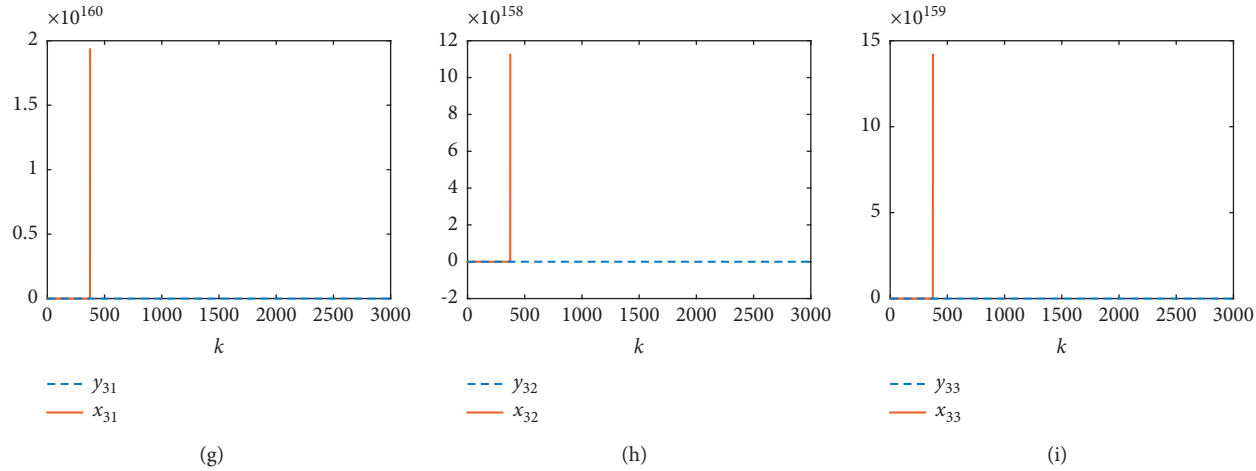

Figure 12: Solution trajectories synthesized by Y_{k+1} (i.e., A_{k+1}^{-1}) and by DDD10i algorithm (55) for example 2 (with $\epsilon = 0.001$ s and $\gamma = 1.0$). (a) Trajectories of y_{11} and x_{11}. (b) Trajectories of y_{12} and x_{12}. (c) Trajectories of y_{13} and x_{13}. (d) Trajectories of y_{21} and x_{21}. (e) Trajectories of y_{22} and x_{22}. (f) Trajectories of y_{23} and x_{23}. (g) Trajectories of y_{31} and x_{31}. (h) Trajectories of y_{32} and x_{32}. (i) Trajectories of y_{33} and x_{33}.

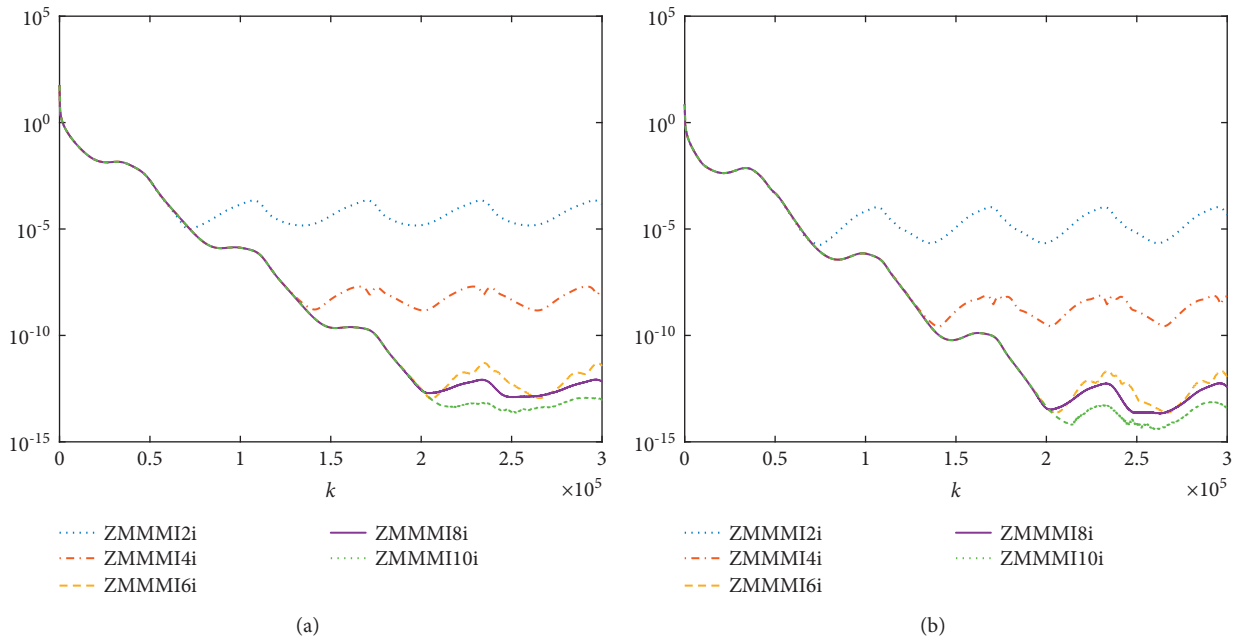

Figure 13: Residual error trajectories synthesized by five ZMMMI algorithms for example 3 (with $\epsilon = 0.0001$ s and $\gamma = 2.0$). (a) Trajectories of $\|X_{k+1}A_{k+1} - I\|_F$. (b) Trajectories of $\|X_{k+1} - A_{k+1}^{-1}\|_F$.

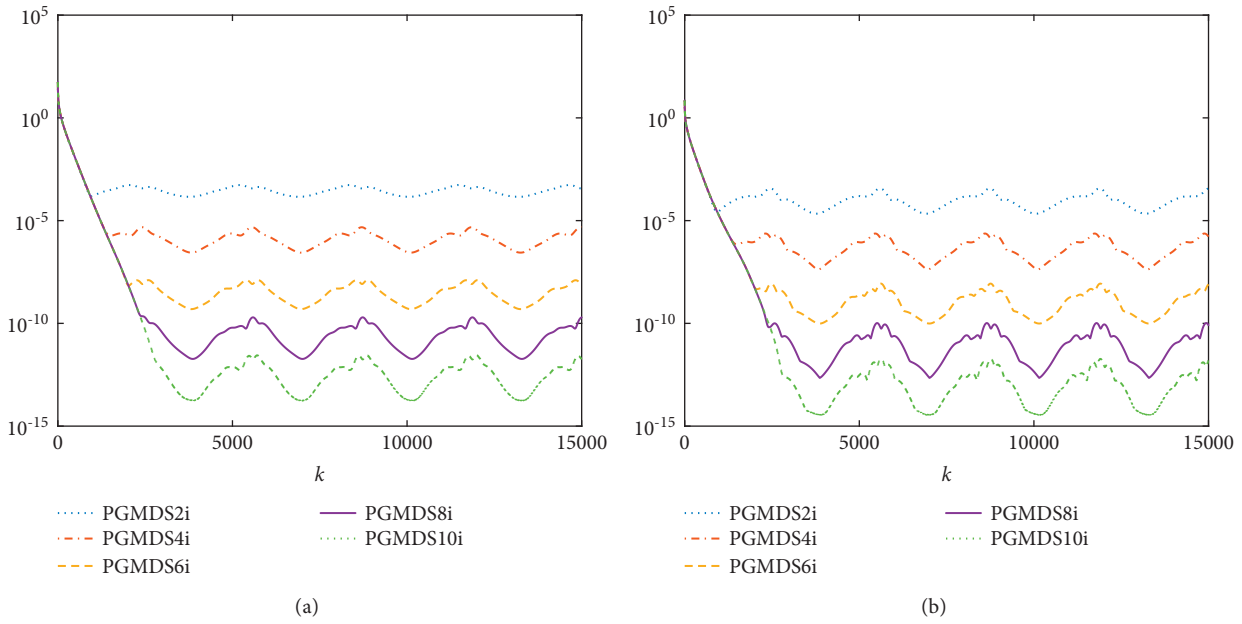

FIGURE 14: Residual error trajectories synthesized by five PGMDS algorithms for example 3 (with $\epsilon = 0.002$ s and $\gamma = 5.0$). (a) Trajectories of $\|X_{k+1}A_{k+1} - I\|_F$. (b) Trajectories of $\|X_{k+1} - A_{k+1}^{-1}\|_F$.

computational process is not disturbed. Second, comparatively, in this study, the initial value is not close enough to the theoretical value. Third, the higher-order discrete algorithm (i.e., the ten-instant discrete algorithm) is more easily affected by the rounding error disturbance [44]. Thus, the experimental results of the DDD model in this study showing the divergence are actually complementing the previous research studies, in addition to the confirmation of [47] about divergence. Therefore, we summarize that the DDD model is generally less effective, with further in-depth investigation being also a future research direction.

7. Conclusion

In this paper, we have shed some light on the matrix inversion solution models derivation, i.e., ZMMMI model (10), PGMDS model (38), and DDD model (49), from PEs. It has provided a new perspective to make full use of the theoretical value of PEs. First, with the substitution technique and design formula of ZNN, we have investigated and proposed the new model of ZNN for matrix inversion problem. Then, we have discussed the convergence and accuracy of ZMMMI model (10) and presented two theorems and proofs about it. On the basis of the model, we have developed five ZMMMI algorithms to discretize continuous-time ZMMMI model (10). Second, with the substitution technique and design formula of ZNN, we have investigated and presented PGMDS model (38) for matrix inversion problem. On the basis of the model, we have shown five PGMDS algorithms to discretize continuous-time PGMDS model (38). Third, with the substitution technique and design formula of ZNN, we have presented DDD model (49) for matrix inversion problem. On the basis of the model, we have developed five DDD algorithms to discretize continuous-time DDD model (38). Fourth, we have prepared three examples to calculate the inverse of matrices by using the above three models, respectively. The results illustrate that ZMMMI model (10) and PGMDS model (38) are effective, while DDD model (49) is less effective. In the future, based on PEs, we will use the substitution technique and ZNN design formula to discuss more efficient RNN models for TVMI problem, including pseudo-inversion and complex matrix inversion.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (no. 61976230), the project supported by Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (no. 2018), the Key-Area Research and Development Program of Guangzhou (no. 202007030004), the Research Fund Program of Guangdong Key Laboratory of Modern Control Technology (no. 2017B030314165), the China Postdoctoral Science Foundation (no. 2018M643306), the Fundamental Research Funds for the Central Universities (no. 19lgpy227), and the Shenzhen Science and Technology Plan Project (no. JCYJ20170818154936083).

References

[1] J. H. Wilkinson, "Error analysis of direct methods of matrix inversion," *Journal of the ACM*, vol. 8, no. 3, pp. 281–330, 1961.

[2] B. Cai and X. Jiang, "A novel artificial neural network method for biomedical prediction based on matrix pseudo-inversion," *Journal of Biomedical Informatics*, vol. 48, pp. 114–121, 2014.

[3] R. J. Steriti and M. A. Fiddy, "Regularized image reconstruction using SVD and a neural network method for matrix inversion," *IEEE Transactions on Signal Processing*, vol. 41, no. 10, pp. 3074–3077, 1993.

[4] R. M. Gower, F. Hanzely, P. Richtarik, and S. U. Stich, "Accelerated stochastic matrix inversion: general theory and speeding up BFGS rules for faster second-order optimization," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 1626–1636, Montreal, Canada, December 2018.

[5] R. H. Sturges, "Analog matrix inversion (robot kinematics)," *IEEE Journal of Robotics and Automation*, vol. 4, no. 2, pp. 157–162, 1988.

[6] L.-L. Wang and L.-X. Pan, "Research on SBMPC algorithm for path planning of rescue and detection robot," *Discrete Dynamics in Nature and Society*, vol. 2020, Article ID 7821942, 11 pages, 2020.

[7] J. Guo, B. Qiu, and Y. Zhang, "New-type DTZ model for solving discrete time-dependent nonlinear equation system with robotic-arm application," in *Proceedings of the International Conference on Information Science and Technology (ICIST)*, pp. 153–162, Bath, UK, September 2020.

[8] W. Yang, J. Chen, Y. Zhang, J. Sun, and Z. Zhang, "Abundant computer and robot experiments verifying minimum joint motion planning and control of redundant arms via Zhang neural network," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Shenzhen, China, July 2021.

[9] K. Yeung and F. Kumbi, "Symbolic matrix inversion with application to electronic circuits," *IEEE Transactions on Circuits and Systems*, vol. 35, pp. 235–238, 1988.

[10] P. Benner, P. Ezzatti, E. S. Quintana-Orti, and A. Remon, "Extending the Gauss-Huard method for the solution of Lyapunov matrix equations and matrix inversion," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 9, pp. 1–13, 2017.

[11] L. Xiao, Y. Zhang, K. Li, B. Liao, and Z. Tan, "A novel recurrent neural network and its finite-time solution to time-varying complex matrix inversion," *Neurocomputing*, vol. 331, pp. 483–492, 2019.

[12] Y. Zhang, L. He, C. Hu, J. Guo, J. Li, and Y. Shi, "General four-step discrete-time zeroing and derivative dynamics applied to time-varying nonlinear optimization," *Journal of Computational and Applied Mathematics*, vol. 347, pp. 314–329, 2019.

[13] C. K. Koc and G. Chen, "Inversion of all principal submatrices of a matrix," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 1, pp. 280-281, 1994.

[14] A. C. Tsoi and S. Tan, "Recurrent neural networks: a constructive algorithm, and its properties," *Neurocomputing*, vol. 15, no. 3-4, pp. 309–326, 1997.

[15] Z. Uykan, "Discrete pseudo-SINR-balancing nonlinear recurrent system," *Discrete Dynamics in Nature and Society*, vol. 2013, Article ID 480560, 19 pages, 2013.

[16] K. Chen and C. Yi, "Robustness analysis of a hybrid of recursive neural dynamics for online matrix inversion," *Applied Mathematics and Computation*, vol. 273, pp. 969–975, 2016.

[17] S. Qin, X. Yang, X. Xue, and J. Song, "A one-layer recurrent neural network for pseudoconvex optimization problems with equality and inequality constraints," *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3063–3074, 2016.

[18] Y. Hong and Q. Zhang, "Indicator selection for topic popularity definition based on AHP and deep learning models," *Discrete Dynamics in Nature and Society*, vol. 2020, Article ID 9634308, 11 pages, 2020.

[19] Y. Zhang, D. Jiang, and J. Wang, "A recurrent neural network for solving Sylvester equation with time-varying coefficients," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1053–1063, 2002.

[20] B. Liao and Y. Zhang, "Different complex ZFs leading to different complex ZNN models for time-varying complex generalized inverse matrices," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 9, pp. 1621–1631, 2014.

[21] I. Stojanovic, P. Stanimirovic, I. Zivkovic, D. Gerontitis, and X.-Z. Wang, "ZNN models for computing matrix inverse based on hyperpower iterative methods," *Filomat*, vol. 31, no. 10, pp. 2999–3014, 2017.

[22] Z. Zhang, X. Deng, X. Qu, B. Liao, L.-D. Kong, and L. Li, "A varying-gain recurrent neural network and its application to solving online time-varying matrix equation," *IEEE Access*, vol. 6, pp. 77940–77952, 2018.

[23] Z. Qi and Y. Zhang, "New models for future problems solving by using ZND method, correction strategy and extrapolation formulas," *IEEE Access*, vol. 7, pp. 84536–84544, 2019.

[24] M. Sun and J. Liu, "General six-step discrete-time Zhang neural network for time-varying tensor absolute value equations," *Discrete Dynamics in Nature and Society*, vol. 2019, Article ID 4861912, 12 pages, 2019.

[25] D. Wu, Y. Zhang, J. Guo, Z. Li, and L. Ming, "GMDS-ZNN model 3 and its ten-instant discrete algorithm for time-variant matrix inversion compared with other multiple-instant ones," *IEEE Access*, vol. 8, pp. 228188–228198, 2020.

[26] M. Sun, M. Tian, and Y. Wang, "Discrete-time Zhang neural networks for time-varying nonlinear optimization," *Discrete Dynamics in Nature and Society*, vol. 201914 pages, Article ID 4745759, 2019.

[27] A. Ben-Israel and T. Greville, *Generalized Inverse: Theory and Applications*, Springer-Verlag, New York, NY, USA, 2003.

[28] B. Liao and Y. Zhang, "From different ZFs to different ZNN models accelerated via li activation functions to finite-time convergence for time-varying matrix pseudoinversion," *Neurocomputing*, vol. 133, pp. 512–522, 2014.

[29] X. Xiao, C. Jiang, H. Lu et al., "A parallel computing method based on zeroing neural networks for time-varying complex-valued matrix Moore-Penrose inversion," *Information Sciences*, vol. 524, pp. 216–228, 2020.

[30] P. Stanimirovic, V. N. Katsikis, and S. Li, "Higher-order ZNN dynamics," *Neural Processing Letters*, vol. 51, no. 1, pp. 697–721, 2020.

[31] Y. Zhang, L. Jin, D. Guo, Y. Yin, and Y. Chou, "Taylor-type 1-step-ahead numerical differentiation rule for first-order derivative approximation and ZNN discretization," *Journal of Computational and Applied Mathematics*, vol. 273pp. 29–40, C, 2015.

[32] Y. Zhang, X. Yang, J. Wang, J. Li, and L. He, "New formula 4IgSFDL of Zhang finite difference for 1st-order derivative approximation with numerical experiment verification," in *Proceedings of the International Conference on Computer Science and Network Technology (ICCSNT)*, pp. 206–209, Changchun, China, 2016.

[33] Y. Zhang, M. Zhu, C. Hu, J. Li, and M. Yang, "Euler-precision general-form of Zhang et al discretization (ZeaD) formulas, derivation, and numerical experiments," in *Proceedings of the Chinese Control and Decision Conference (CCDC)*, pp. 6273–6278, Shenyang, China, 2018.

[34] J. Wang, "Recurrent neural networks for computing pseudoinverses of rank-deficient matrices," *SIAM Journal on Scientific Computing*, vol. 18, no. 5, pp. 1479–1493, 1997.

[35] Y. Zhang, W. Ma, and B. Cai, "From Zhang neural network to Newton iteration for matrix inversion," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 7, pp. 1405–1415, 2009.

[36] D. Guo and Y. Zhang, "Zhang neural network, Getz-Marsden dynamic system, and discrete-time algorithms for time-varying matrix inversion with application to robots' kinematic control," *Neurocomputing*, vol. 97, pp. 22–32, 2012.

[37] D. Guo, Z. Nie, and L. Yan, "Novel discrete-time Zhang neural network for time-varying matrix inversion," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2301–2310, 2017.

[38] D. Guo, D. Liang, Y. Wang, D. Chen, and Y. Zhang, "Solving for ZGI via ZNN and discrete-time algorithms with application to robot control," in *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, vol. 2015, pp. 31–36, Shenyang, China, March 2015.

[39] M. Mao, J. Li, L. Jin, S. Li, and Y. Zhang, "Enhanced discrete-time Zhang neural network for time-variant matrix inversion in the presence of bias noises," *Neurocomputing*, vol. 207, pp. 220–230, 2016.

[40] Y. Zhang, Y. Shi, K. Chen, and C. Wang, "Global exponential convergence and stability of gradient-based neural network for online matrix inversion," *Applied Mathematics and Computation*, vol. 215, no. 3, pp. 1301–1306, 2009.

[41] N. H. Getz and J. E. Marsden, "Dynamical methods for polar decomposition and inversion of matrices," *Linear Algebra and its Applications*, vol. 258, pp. 311–343, 1997.

[42] N. H. Getz, *Dynamic inversion of nonlinear maps with applications to nonlinear control and robotics*, Ph.D. dissertation, University of California at Berkeley, Berkeley, CA, USA, 1995.

[43] F. R. Villatoro and J. I. Ramos, "On the method of modified equations. I: asymptotic analysis of the Euler forward difference method," *Applied Mathematics and Computation*, vol. 103, no. 2-3, pp. 111–139, 1999.

[44] J. H. Mathews and K. K. Fink, *Numerical Methods Using MATLAB*, Pearson Prentice Hall, Upper Saddle River, NJ, USA, 2004.

[45] Y. Zhang, H. Gong, M. Yang, J. Li, and X. Yang, "Stepsize range and optimal value for Taylor-Zhang discretization formula applied to zeroing neurodynamics illustrated via future equality-constrained quadratic programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, pp. 959–966, 2019.

[46] J. Chen, J. Guo, and Y. Zhang, "General ten-instant DTDMSR model for dynamic matrix square root finding," *Cybernetics & Systems*, vol. 52, no. 1, pp. 127–143, Oct. 2020.

[47] Y. Zhang, N. Shi, M. Yang, J. Guo, and J. Chen, "Output optimization of scalar and 2-dimension time-varying nonlinear systems using zeroing dynamics," *Asian Journal of Control*, vol. 23, no. 4, pp. 1643–1657, 2021.