

Research Article

New Models for Solving Time-Varying LU Decomposition by Using ZNN Method and ZeaD Formulas

Liangjie Ming,^{1,2} Yunong Zhang ,^{2,3} Jinjin Guo,^{2,3} Xiao Liu,^{1,2} and Zhonghua Li⁴

¹School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China

²Research Institute of Sun Yat-sen University in Shenzhen, Shenzhen 518057, China

³School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China

⁴School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510006, China

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

Received 13 November 2020; Revised 22 March 2021; Accepted 28 March 2021; Published 22 April 2021

Academic Editor: Nan-Jing Huang

Copyright © 2021 Liangjie Ming et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, by employing the Zhang neural network (ZNN) method, an effective continuous-time LU decomposition (CTLUD) model is firstly proposed, analyzed, and investigated for solving the time-varying LU decomposition problem. Then, for the convenience of digital hardware realization, this paper proposes three discrete-time models by using Euler, 4-instant Zhang et al. discretization (ZeaD), and 8-instant ZeaD formulas to discretize the proposed CTLUD model, respectively. Furthermore, the proposed models are used to perform the LU decomposition of three time-varying matrices with different dimensions. Results indicate that the proposed models are effective for solving the time-varying LU decomposition problem, and the 8-instant ZeaD LU decomposition model has the highest precision among the three discrete-time models.

1. Introduction

The LU decomposition is generally described as decomposing a matrix A into a unit lower triangular matrix (simply termed, L) and an upper triangular matrix (simply termed, U). Meanwhile, the matrix A equals the product of L and U , that is, $A = LU$. This is also the origin of the name of LU decomposition. In numerical analysis, the LU decomposition is applied to solving linear systems, calculating the matrix determinant, and so on. Moreover, LU decomposition, as an important branch of matrix decompositions, also has many applications in science and engineering fields [1–6], such as in multidimensional frequency estimation [7], array structures [8], and power flow algorithm [9]. In addition, the LU decomposition-based techniques play an important role in solving various engineering problems. For instance, in order to obtain a secure and robust watermarking algorithm, the LU decomposition was utilized in [10]. Besides, the LU decomposition was employed to find the complex nonorthogonal joint diagonalization in [11]. In [12], for fast circuit analysis, the parallel sparse LU

decomposition method was used. Therefore, solving the LU decomposition problem is very meaningful.

For solving the LU decomposition problem, many algorithms or methods were proposed. For example, an LU-spatial decomposition method was proposed in [13]. The method uses a partitioning scheme to compute the LU decomposition rather than the explicit inverse, which reduces the computational cost efficiently. A recursive LU decomposition algorithm was proposed in [14]. The method can also save the computation time but needs to make geometrical modifications to the precomputed scatterer. What is more, as for the nonsingular matrix, a method to find the quasi-LU decomposition was proposed in [15]. The method can obtain the quasi-LU decomposition of a matrix though the matrix does not have LU decomposition. However, those proposed algorithms or methods are not designed to find the LU decomposition of the time-varying matrix. Thus, for the time-varying matrix, those methods have no significant advantages to find the LU decomposition and may have lag errors. For LU decomposition of the time-varying matrix, there is little research while the time-varying

problems are becoming more and more important [16–18]. Besides, an important application of time-varying LU decomposition is solving the angle-of-arrival (AoA) localization problem [19], which is consistent with the time-varying linear system and has been widely applied in various fields [20, 21]. The AoA localization problem can be formulated as $A(t)\mathbf{x}(t) = \mathbf{b}(t)$. Thus, using a model or method (which can obtain the LU decomposition of the time-varying matrix $A(t)$ in real time) can accelerate the solving progress of the AoA localization problem.

Due to the powerful capabilities in time-varying information processing, Zhang neural network (ZNN) method has been applied to solving various time-varying problems [16–18, 22–28]. For example, for finding real-time matrix square root, a finite-time convergent ZNN was proposed in [16]. In order to solve the time-varying matrix inversion problem, a novel discrete-time ZNN was proposed in [17]. Therefore, the ZNN method is employed to solve the time-varying LU decomposition problem in this paper.

In the process of applying the ZNN method to solving the time-varying problems [29–31], an error function is constructed firstly. Then, by using the ZNN design formula, the error function is forced to converge to zero. Furthermore, the continuous-time model for solving the original problem is obtained. In this paper, the corresponding model is termed the continuous-time LU decomposition (CTLUD) model. In addition, by employing three one-step forward finite difference formulas (i.e., Euler, 4-instant Zhang et al. discretization (ZeaD), and 8-instant ZeaD formulas) to discretize the proposed CTLUD model, three corresponding discrete-time models are obtained.

The remainder of this paper is composed of six sections. In Section 2, the problem formulation of the time-varying LU decomposition is presented. In Section 3, the design process of the CTLUD model is presented. In Section 4, three discretization formulas are presented and corresponding discrete-time models are obtained. In Section 5, the proposed CTLUD model is employed to perform LU decomposition of three time-varying matrices with different dimensions, and the corresponding results are shown. In Section 6, the experiment results of three discrete-time models are presented. In Section 7, this paper is concluded. Before ending the introduction part, the main contributions of this paper are recapped as follows:

- (1) Different from static LU decomposition problem analysis, this paper considers and analyzes the time-varying LU decomposition problem.
- (2) An effective CTLUD model is proposed by employing the ZNN method, Kronecker product, and vectorization techniques.
- (3) Three discrete-time LU decomposition models are obtained by employing three discretization formulas to discretize the proposed CTLUD model.
- (4) Experiment results substantiate that the proposed models are effective for solving the time-varying LU decomposition problem.

2. Problem Formulation

Generally, the time-varying LU decomposition problem can be formulated as follows:

$$A(t) = L(t)U(t), \quad (1)$$

where $A(t) \in \mathbb{R}^{n \times n}$ denotes a smoothly time-varying matrix to be decomposed, $L(t) \in \mathbb{R}^{n \times n}$ denotes a unit lower triangular matrix, and $U(t) \in \mathbb{R}^{n \times n}$ denotes an upper triangular matrix. Note that $L(t)$ and $U(t)$ are unknown and time-varying matrices to be obtained.

It is worth noting that not all time-varying matrices have corresponding LU decompositions. In this paper, in order to simplify the problem, we only consider the situation that $A(t)$ is a diagonally dominant matrix. It means that (1) is solvable.

3. Continuous-Time Model

In this section, by employing the ZNN method, a CTLUD model is proposed.

According to the previous work of the ZNN method, if we want to find a theoretical solution matrix $T(t)$ of a time-varying matrix problem, an appropriate error function $E(t)$ is indispensable, where $E(t) = T(t) - R(t)$ indicates the difference between $T(t)$ and actual result matrix $R(t)$. In this paper, $E(t)$ is constructed as follows:

$$E(t) = A(t) - L(t)U(t). \quad (2)$$

Then, $E(t)$ is adjusted dynamically by the following ZNN design formula:

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

which indicates that $E(t)$ is forced to converge to zero matrix globally and exponentially, where $\dot{E}(t)$ denotes the first-order time derivative of $E(t)$. The design parameter $\lambda > 0 \in \mathbb{R}$, which is used to adjust the convergence speed. The convergence speed becomes higher as the value of λ increases.

By substituting (2) into (3), the following dynamic matrix equation is obtained:

$$\dot{A}(t) - \dot{L}(t)U(t) - L(t)\dot{U}(t) = -\lambda E(t), \quad (4)$$

where $\dot{A}(t)$, $\dot{L}(t)$, and $\dot{U}(t)$ denote the first-order time derivatives of $A(t)$, $L(t)$, and $U(t)$, respectively.

Note that some elements in $L(t)$ and $U(t)$ are known. Those known elements do not need to solve. Thus, we construct the following two vectors:

$$\begin{cases} \zeta(t) = [l_{21}(t) \ l_{31}(t) \ \cdots \ l_{n1}(t) \ l_{32}(t) \ \cdots \ l_{n(n-1)}(t)]^T, \\ \mathbf{u}(t) = [u_{11}(t) \ u_{12}(t) \ u_{22}(t) \ u_{13}(t) \ \cdots \ u_{mm}(t)]^T, \end{cases} \quad (5)$$

where $\zeta(t) \in \mathbb{R}^{(1/2)n(n-1) \times 1}$ and $\mathbf{u}(t) \in \mathbb{R}^{(1/2)n(n+1) \times 1}$ are composed by all unknown elements of $L(t)$ and $U(t)$, respectively. Meanwhile, $l_{ij}(t)$ and $u_{ij}(t)$ denote the elements

of $L(t)$ and $U(t)$ in row i and column j , respectively. The superscript T denotes the matrix transpose operator.

Furthermore, we have the following two theorems about $L(t)$ and $U(t)$ [32–34].

Theorem 1. *If $L(t) \in \mathbb{R}^{n \times n}$ is a unit lower triangular matrix, then the following equation holds true:*

$$\text{vec}(L(t)) = H\zeta(t) + \text{vec}(I_n), \quad (6)$$

where $\text{vec}(\cdot)$ denotes a column vector obtained by stacking all column vectors of the operational matrix together and

$$\begin{aligned} H\zeta(t) &= [0 \ l_{21}(t) \ l_{31}(t) \ \cdots \ l_{n1}(t) \ 0 \ 0 \ l_{32}(t) \ \cdots \ l_{nn-1}(t) \ \cdots \ 0]^T \\ &= \text{vec}(D(t)) \in \mathbb{R}^{n^2 \times 1}, \end{aligned} \quad (9)$$

where

$$D(t) = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ l_{21}(t) & 0 & \cdots & \cdots & 0 \\ \vdots & l_{32}(t) & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ l_{n1}(t) & l_{n2}(t) & \cdots & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}. \quad (10)$$

Then, we obtain that $L(t) = D(t) + I_n$, that is, $\text{vec}(L(t)) = H\zeta(t) + \text{vec}(I_n)$. The proof is completed. \square

Theorem 2. *If $U(t) \in \mathbb{R}^{n \times n}$ is an upper triangular matrix, then the following equation holds true:*

$$\text{vec}(U(t)) = K\mathbf{u}(t), \quad (11)$$

where matrix K is defined as follows:

$$K = [k_1 \ k_2 \ \cdots \ k_{n-1} \ k_n] \in \mathbb{R}^{n^2 \times (1/2)n(n+1)}, \quad (12)$$

and k_i is defined as

$$k_i = [\mathbf{0}_{n(i-1),i}; \ I_i; \ \mathbf{0}_{n^2-n(i-1)-i,i}] \in \mathbb{R}^{n^2 \times i}. \quad (13)$$

Proof. The proof is similar to that of Theorem 1 and thus omitted.

According to previous statements, we have

$$\begin{cases} \text{vec}(L(t)) = H\zeta(t) + \text{vec}(I_n), \\ \text{vec}(U(t)) = K\mathbf{u}(t), \\ \text{vec}(\dot{L}(t)) = H\dot{\zeta}(t), \\ \text{vec}(\dot{U}(t)) = K\dot{\mathbf{u}}(t), \end{cases} \quad (14)$$

where $\dot{\zeta}(t)$ and $\dot{\mathbf{u}}(t)$ denote the first-order time derivatives of $\zeta(t)$ and $\mathbf{u}(t)$, respectively.

Furthermore, the Kronecker product and vectorization techniques are employed, which are formulated by the following lemma [35, 36]. \square

$I_i \in \mathbb{R}^{i \times i}$ denotes an identity matrix. Matrix H is defined as follows:

$$H = [h_{n-1} \ h_{n-2} \ \cdots \ h_2 \ \mathbf{h}_1] \in \mathbb{R}^{n^2 \times (1/2)n(n-1)}, \quad (7)$$

where h_i is defined as follows:

$$h_i = [\mathbf{0}_{n^2-i(n+1),i}; \ I_i; \ \mathbf{0}_{in,i}] \in \mathbb{R}^{n^2 \times i}, \quad (8)$$

and $\mathbf{0}_{p,q} \in \mathbb{R}^{p \times q}$ denotes a zero matrix. Note that \mathbf{h}_1 is a vector.

Proof. Let H multiply $\zeta(t)$ and we have

Lemma 1. *If $W \in \mathbb{R}^{m \times n}$, $X \in \mathbb{R}^{n \times p}$, $Y \in \mathbb{R}^{p \times q}$, and $Z \in \mathbb{R}^{m \times q}$, then $WXY = Z$ is equivalent to*

$$(Y^T \otimes W)\text{vec}(X) = \text{vec}(Z), \quad (15)$$

where symbol \otimes denotes the Kronecker product. Note that $Y^T \otimes W \in \mathbb{R}^{m \times n \times p}$, $\text{vec}(X) \in \mathbb{R}^{n \times p \times 1}$, and $\text{vec}(Z) \in \mathbb{R}^{m \times q \times 1}$.

By applying Lemma 1, (4) is equivalent to

$$\begin{aligned} \text{vec}(\dot{A}(t) + \lambda E(t)) &= (U^T(t) \otimes I_n)\text{vec}(\dot{L}(t)) \\ &\quad + (I_n \otimes L(t))\text{vec}(\dot{U}(t)), \end{aligned} \quad (16)$$

which is rewritten as

$$\begin{aligned} \text{vec}(\dot{A}(t) + \lambda E(t)) &= [U^T(t) \otimes I_n \ I_n \otimes L(t)] \begin{bmatrix} \text{vec}(\dot{L}(t)) \\ \text{vec}(\dot{U}(t)) \end{bmatrix} \\ &= [(U^T(t) \otimes I_n)H \ (I_n \otimes L(t))K] \begin{bmatrix} \dot{\zeta}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix}. \end{aligned} \quad (17)$$

For simplification, we denote

$$\begin{aligned} \mathbf{v}(t) &= \text{vec}(\dot{A}(t) + \lambda E(t)) \in \mathbb{R}^{n^2 \times 1}, \\ M(t) &= [(U^T(t) \otimes I_n)H \ (I_n \otimes L(t))K] \in \mathbb{R}^{n^2 \times n^2}, \\ \dot{\mathbf{s}}(t) &= \begin{bmatrix} \dot{\zeta}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix} \in \mathbb{R}^{n^2 \times 1}. \end{aligned} \quad (18)$$

Therefore, we get

$$M(t)\dot{\mathbf{s}}(t) = \mathbf{v}(t). \quad (19)$$

Furthermore, we have the following CTLUD model:

$$\dot{\mathbf{s}}(t) = M^\dagger(t)\mathbf{v}(t), \quad (20)$$

where $M^\dagger(t)$ denotes the pseudo-inverse matrix of $M(t)$ [34]. Meanwhile, by giving the initial value of $\mathbf{s}(t)$, $\zeta(t)$ and $\mathbf{u}(t)$ can be obtained via (20). According to Theorems 1 and

2, $L(t)$ and $U(t)$ are also obtained. Note that (20) is also a neurodynamics model (which transforms a matrix decomposition problem into a matrix differential equation problem), can obtain the solution in real time, and has the advantages of parallelizability [22, 36]. Those are also the advantages of the proposed CTLUD model. Besides, Figure 1 shows the block diagram about the solving progress of the proposed CTLUD model for the time-varying LU decomposition problem.

Furthermore, we have the following theorem, which discusses the convergence performance of the proposed CTLUD model.

Theorem 3. *With $A(t)$ being a smoothly time-varying matrix and $M(t)$ being always nonsingular, the elements of $E(t)$ converge to zero globally and exponentially.*

Proof. With $e_{ij}(t)$ denoting the element of $E(t)$ in row i and column j , where $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, n$. According to (3), we have $\dot{e}_{ij}(t) = -\lambda e_{ij}(t)$. Its solution is mathematically expressed as $e_{ij}(t) = e_{ij}(0)\exp(-\lambda t)$. As time $t \rightarrow \infty$, $e_{ij}(t)$ exponentially converges to zero. Then, $E(t)$ globally and exponentially converges to zero matrix; that is, the solution of the CTLUD model globally and exponentially converges to the theoretical solution. The proof is thus completed.

The following remark discusses the computational complexity of the proposed CTLUD model. \square

Remark 1. With using big \mathbf{O} notation, the time complexity of each step in obtaining the proposed CTLUD model is listed as follows. Note that only the highest order in each step is concerned.

The time complexity of obtaining $M(t)$: $\mathbf{O}(n^4)$.

The time complexity of obtaining $\mathbf{v}(t)$: $\mathbf{O}(n^2)$.

The time complexity of matrix multiplication operation: $\mathbf{O}(n^4)$.

The time complexity of matrix pseudo-inverse operation: $\mathbf{O}(n^6)$.

Thus, in the case that we use the traditional serial numerical algorithm, the total time complexity of the proposed CTLUD model is $\mathbf{O}(n^6)$. It is worth pointing out that the proposed CTLUD model is a neurodynamics model and can be implemented by parallel distributed processing. In the case that we use parallel distributed processing, the time complexities of obtaining $M(t)$ and $\mathbf{v}(t)$ can be reduced to $\mathbf{O}(n^2)$ and $\mathbf{O}(n)$ [36], respectively. Besides, the capability that the ZNN method can find the matrix pseudo-inverse in parallel, has been revealed in many works [37, 38]. The time complexities of matrix multiplication operation and matrix pseudo-inverse operation can be reduced to $\mathbf{O}(n^2)$ by sacrificing the space complexity [36].

4. Discrete-Time Models

In this section, for the convenience of digital hardware realization [34, 39–41], three discrete-time LU decomposition models are proposed, discussed, and investigated. Note that the three discrete-time models are obtained by applying three corresponding discretization formulas.

4.1. Discretization Formulas. The Euler formula [42], which is also a one-step-ahead discretization formula, is presented as follows:

$$\dot{s}(t_k) = \frac{s(t_{k+1}) - s(t_k)}{\eta} + O(\eta), \quad (21)$$

which contains values of two time instants, where $s(t_{k+i}) = s((k+i)\eta)$ and η denotes the sampling gap. Besides, $O(\eta)$ denotes the first-order truncation error.

A 4-instant ZeaD formula [43], which contains values of four time instants, is presented as follows:

$$\dot{s}(t_k) = \frac{2s(t_{k+1}) - 3s(t_k) + 2s(t_{k-1}) - s(t_{k-2})}{2\eta} + O(\eta^2), \quad (22)$$

where $O(\eta^2)$ denotes the second-order truncation error.

The last discretization formula employed in this paper is an 8-instant ZeaD formula [44] and presented as follows:

$$\begin{aligned} \dot{s}(t_k) = & \left(\frac{200}{483}s(t_{k+1}) + \frac{1097}{9660}s(t_k) - \frac{160}{483}s(t_{k-1}) \right. \\ & - \frac{20}{69}s(t_{k-2}) + \frac{20}{483}s(t_{k-3}) + \frac{55}{1932}s(t_{k-4}) \\ & \left. + \frac{44}{805}s(t_{k-5}) - \frac{5}{161}s(t_{k-6}) \right) \frac{1}{\eta} + O(\eta^4), \end{aligned} \quad (23)$$

where $O(\eta^4)$ denotes the fourth-order truncation error.

4.2. Corresponding Models. According to the three discretization formulas presented in the last subsection, the following three equations are obtained:

$$\mathbf{s}(t_{k+1}) = \eta \dot{\mathbf{s}}(t_k) + \mathbf{s}(t_k) + \mathbf{O}(\eta^2), \quad (24)$$

$$\mathbf{s}(t_{k+1}) = \eta \dot{\mathbf{s}}(t_k) + \frac{3}{2}\mathbf{s}(t_k) + \mathbf{s}(t_{k-1}) + \frac{1}{2}\mathbf{s}(t_{k-2}) + \mathbf{O}(\eta^3),$$

$$\begin{aligned} \mathbf{s}(t_{k+1}) = & \frac{483}{200}\eta \dot{\mathbf{s}}(t_k) - \frac{393}{1433}\mathbf{s}(t_k) \\ & + \frac{4}{5}\mathbf{s}(t_{k-1}) + \frac{7}{10}\mathbf{s}(t_{k-2}) - \frac{1}{10}\mathbf{s}(t_{k-3}) \\ & - \frac{11}{160}\mathbf{s}(t_{k-4}) - \frac{33}{250}\mathbf{s}(t_{k-5}) + \frac{3}{40}\mathbf{s}(t_{k-6}) + \mathbf{O}(\eta^5), \end{aligned} \quad (25)$$

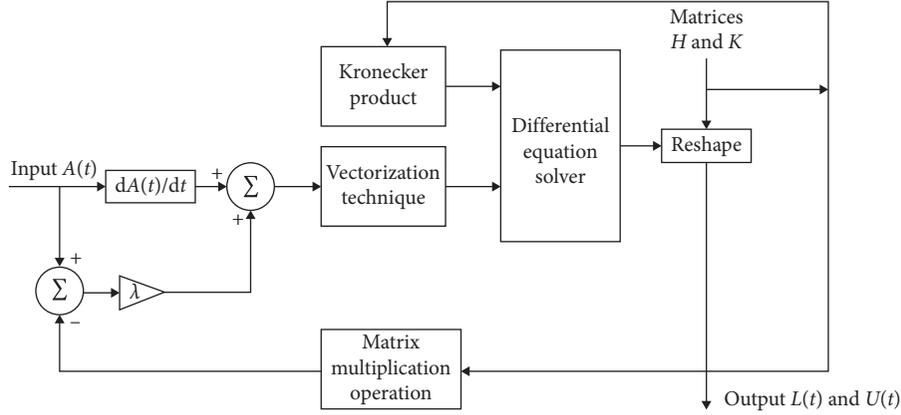


FIGURE 1: Block diagram of the proposed CTLUD model.

where $\mathbf{O}(\eta^i)$ denotes a vector with entries of order $O(\eta^i)$ and $\dot{\mathbf{s}}(t_k) = M^\dagger(t_k)\mathbf{v}(t_k)$. Based on the above equations, three corresponding discrete-time models are obtained as follows:

$$\mathbf{s}(t_{k+1}) \doteq \eta \dot{\mathbf{s}}(t_k) + \mathbf{s}(t_k), \quad (26)$$

$$\mathbf{s}(t_{k+1}) \doteq \eta \dot{\mathbf{s}}(t_k) + \frac{3}{2}\mathbf{s}(t_k) + \mathbf{s}(t_{k-1}) + \frac{1}{2}\mathbf{s}(t_{k-2}), \quad (27)$$

$$\mathbf{s}(t_{k+1}) \doteq \frac{483}{200}\eta \dot{\mathbf{s}}(t_k) - \frac{393}{1433}\mathbf{s}(t_k) + \frac{4}{5}\mathbf{s}(t_{k-1}) + \frac{7}{10}\mathbf{s}(t_{k-2}) - \frac{1}{10}\mathbf{s}(t_{k-3}) - \frac{11}{160}\mathbf{s}(t_{k-4}) - \frac{33}{250}\mathbf{s}(t_{k-5}) + \frac{3}{40}\mathbf{s}(t_{k-6}), \quad (28)$$

where “ \doteq ” denotes the computation assignment operator from its right-hand side to its left-hand side. In this paper, (26)–(28) are termed Euler discrete-time LU decomposition (EDTLUD) model, 4-instant ZeaD LU decomposition (4IZLUD) model, and 8-instant ZeaD LU decomposition (8IZLUD) model, respectively. Besides, we have the following theorem.

Theorem 4. *With $\eta \in (0, 1)$ denoting the sampling gap, the EDTLUD, 4IZLUD, and 8IZLUD models are 0-stable, consistent, and convergent, and their truncation error orders are $\mathbf{O}(\eta^2)$, $\mathbf{O}(\eta^3)$, and $\mathbf{O}(\eta^5)$, respectively.*

Proof. The EDTLUD model is chosen as an example. The characteristic polynomial of the EDTLUD model is expressed as follows:

$$P(\gamma) = \gamma - 1. \quad (29)$$

Evidently, (29) has only one root on unit circle, that is, $\gamma = 1$. According to Lemma 2 in the appendix, the EDTLUD model is 0-stable. Moreover, based on (24), the EDTLUD model has a truncation error of $\mathbf{O}(\eta^2)$. Therefore, we come to the conclusion that the EDTLUD model is consistent and convergent according to Lemmas 3–5 in the appendix. As for the 4IZLUD model and the 8IZLUD model, the proofs are similar to that of above, and they are thus omitted here. The proof is completed.

Algorithm 1 shows the pseudocode describing the various steps of the proposed 4IZLUD model. As for the EDLUD model and the 8IDLUD model, their operation processes are very similar to that of the 4IDLUD model and are thus omitted. The following remark shows the analyses about the computational complexities of the proposed discrete-time models. \square

Remark 2. With using big \mathbf{O} notation, the time complexity of each step in obtaining the solution of the time-varying LU decomposition is listed as follows. Note that only the highest order in each step is concerned.

The time complexity of obtaining $E(t_k), \dot{A}(t_k), \mathbf{v}(t_k)$: $\mathbf{O}(n^2)$.

The time complexity of obtaining $M(t_k)$: $\mathbf{O}(n^4)$.

The time complexity of obtaining $M^\dagger(t_k)$: $\mathbf{O}(n^6)$.

The time complexity of obtaining $\dot{\mathbf{s}}(t_k)$: $\mathbf{O}(n^4)$.

The time complexity of obtaining $\mathbf{s}(t_{k+1})$: $\mathbf{O}(n^2)$.

Thus, in the case that we use the traditional serial numerical algorithm, the total time complexity of the proposed discrete-time models is $\mathbf{O}(n^6)$. Similar to the statements in Remark 1, the total time complexity of the proposed discrete-time models can also be reduced to $\mathbf{O}(n^2)$ by sacrificing the space complexity [36].

Data: Matrix $A(t)$, initial values $\mathbf{s}(0)$, sampling gap η , design parameter λ , and final computation time t_f .

Result: Matrices $L(t_{k+1})$ and $U(t_{k+1})$.

- (1) $\mathbf{s}(t_{-1})$ and $\mathbf{s}(t_{-2}) \leftarrow \mathbf{0}$;
- (2) $L(0)$ and $U(0) \leftarrow \text{reshape } \mathbf{s}(0)$;
- (3) $k \leftarrow 0$;
- (4) **while** $k < t_f/\eta$ **do**
- (5) $E(t_k) \leftarrow A(t_k) - L(t_k)U(t_k)$;
- (6) $\dot{A}(t_k) \leftarrow \text{diff}(A(t_k))$;
- (7) $\mathbf{v}(t_k) \leftarrow \text{vec}(\dot{A}(t_k) + \lambda E(t_k))$;
- (8) $M(t_k) \leftarrow [(U^T(t_k) \otimes I_n)H \ (I_n \otimes L(t_k))K]$;
- (9) $\dot{\mathbf{s}}(t_k) \leftarrow M^\dagger(t_k)\mathbf{v}(t_k)$;
- (10) $\mathbf{s}(t_{k+1}) \leftarrow \eta\dot{\mathbf{s}}(t_k) + 1.5\mathbf{s}(t_k) + \mathbf{s}(t_{k-1}) + 0.5\mathbf{s}(t_{k-2})$;
- (11) $L(t_{k+1})$ and $U(t_{k+1}) \leftarrow \text{reshape } \mathbf{s}(t_{k+1})$;
- (12) $k \leftarrow k + 1$;
- (13) **end**

ALGORITHM 1: 4IZLUD algorithm.

$$A(t) = \begin{bmatrix} 50 + \sin(2t) & \cos(t) & \log(t+1) & \frac{1}{t+1} & \cos(t) & \sin(3t) & \cos(2t) \\ \cos(t) & 50 + 2t & 2 & \sin(t+2) & 5 & 10 + \sin(t) & 3 \\ \sin^2(t) & 3 & 200 - \cos(t) & 3 \sin(t) & 200 - \cos(t) & \cos^2(t) & t \\ t & 2 + \frac{1}{t+1} & 0 & 100 + \sin(t) & 3 \cos(2t) & 3t & \cos(t) \\ \cos(t) & 0 & 3 \cos(2t) & \sin^2(t) & 300 & 200 - \cos(t) & 3 \sin(t) \\ 50 + \sin(2t) & \cos(t) & \log(t+1) & 1/(t+1) & \cos(t) & 500 + \sin(3t) & \cos(2t) \\ \cos(t) & 50 + 2t & 2 & \sin(t+2) & 5 & 10 + \sin(t) & 300 \end{bmatrix}. \quad (30)$$

5. Computer Simulations and Results of CTLUD Model

In this section, for verifying the effectiveness of the proposed CTLUD model, three time-varying matrices are presented. Note that the three matrices have different dimensions, and the initial state $\mathbf{s}(0)$ of the CTLUD model can be randomly set. In this paper, we randomly use some integers to be the initial state of the CTLUD model in all examples. Besides, the corresponding experiment results are shown.

5.1. Example 1. Firstly, in order to test the performance of the proposed CTLUD model on solving simple LU decomposition problem, the following 2-dimensional time-varying matrix is considered:

$$A(t) = \begin{bmatrix} 10 + \sin(t) & \cos(2t) \\ 3 & 2 \sin(t) + 10 \end{bmatrix}. \quad (31)$$

In this example, λ is set as 3 and computation time is limited to $[0, t_f]$ s, where $t_f = 10$ denotes the final computation time. In addition, the initial state of $\mathbf{s}(t)$ is set as

$$\mathbf{s}(0) = [2 \ 1 \ 2 \ 3]^T \in \mathbb{R}^{4 \times 1}. \quad (32)$$

In order to perform the LU decomposition of $A(t)$, the proposed CTLUD model is employed. The corresponding experiment results are shown in Figure 2. As seen from Figure 2(a), the trajectory of $\|E(t)\|_F$ is shown, where $\|E(t)\|_F$ denotes the Frobenius-norm of $E(t)$. Evidently, $\|E(t)\|_F$ converges to 0 in a short time. At about 2 s, $\|E(t)\|_F$ is already near 0. Therefore, a conclusion is obtained that the 2-dimensional time-varying LU decomposition problem (i.e., Example 1) is solved effectively by the proposed CTLUD model.

What is more, Figure 2(b) shows the trajectories of $e_{11}(t)$, $e_{21}(t)$, $e_{12}(t)$, and $e_{22}(t)$, where $e_{ij}(t)$ denotes the element of $E(t)$ in row i and column j . It can be evidently

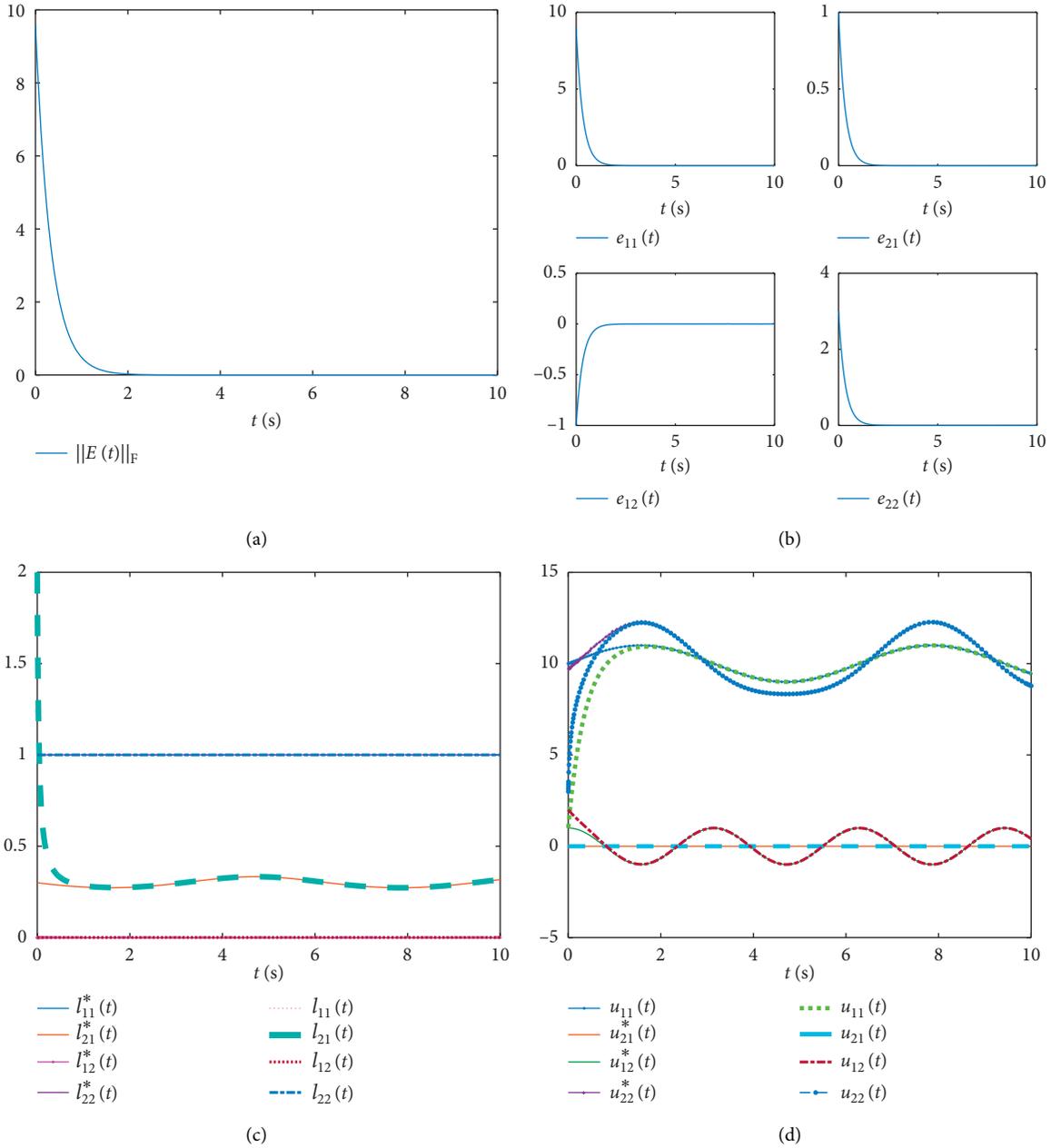


FIGURE 2: Experiment results synthesized by CTLUD model with $\lambda = 3$ and $\mathbf{s}(0) = [2 \ 1 \ 2 \ 3]^T$ for solving 2-dimensional time-varying LU decomposition problem in Example 1. (a) Frobenius-norm trajectory of error matrix $E(t)$. (b) Trajectories of $e_{11}(t)$, $e_{21}(t)$, $e_{12}(t)$, and $e_{22}(t)$. (c) Trajectories of $l_{11}^*(t)$, $l_{21}^*(t)$, $l_{12}^*(t)$, $l_{22}^*(t)$, $l_{11}(t)$, $l_{21}(t)$, $l_{12}(t)$, and $l_{22}(t)$. (d) Trajectories of $u_{11}^*(t)$, $u_{21}^*(t)$, $u_{12}^*(t)$, $u_{22}^*(t)$, $u_{11}(t)$, $u_{21}(t)$, $u_{12}(t)$, and $u_{22}(t)$.

seen that all elements of $E(t)$ converge to 0 in a short time no matter $e_{ij}(0) > 0$ or $e_{ij}(0) < 0$.

In Figures 2(c) and 2(d), we show the element trajectories of $L(t)$ and $U(t)$, where $l_{ij}^*(t)$ and $u_{ij}^*(t)$ denote the corresponding theoretical values of $L(t)$ and $U(t)$, respectively. As seen from Figures 2(c) and 2(d), all elements of $L(t)$ and $U(t)$ track their theoretical values. In other examples, the element trajectories of $L(t)$ and $U(t)$ are not shown because the results are similar to that of Example 1 except for convergence speed.

In this example, because the design parameter λ is set as 3, the convergence speed is not high enough seemingly, but the convergence process is shown clearly. The experiment results with bigger λ are shown in Examples 2 and 3.

5.2. Example 2. Furthermore, in order to verify the validity of the proposed CTLUD model for the common LU decomposition problem, we consider the following 3-dimensional time-varying matrix:

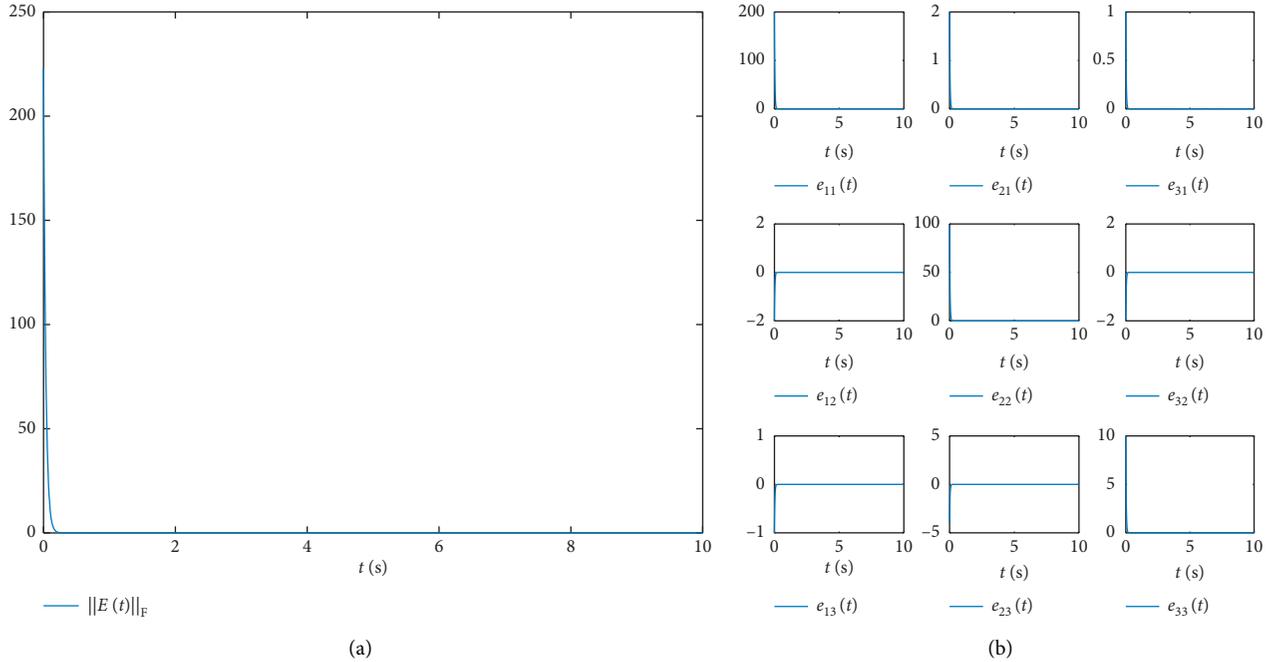


FIGURE 3: Experiment results synthesized by CTLUD model with $\lambda = 30$ and $\mathbf{s}(0) = [0 \ 1 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6]^T$ for solving 3-dimensional time-varying LU decomposition problem in Example 2. (a) Frobenius-norm trajectory of error matrix $E(t)$. (b) Trajectories of $e_{11}(t)$, $e_{21}(t)$, $e_{31}(t)$, $e_{12}(t)$, $e_{22}(t)$, $e_{32}(t)$, $e_{13}(t)$, $e_{23}(t)$, and $e_{33}(t)$.

$$A(t) = \begin{bmatrix} 200 + \cos(t) & 0 & 3 \cos(2t) \\ -\sin(t) + 2 & 100 + \cos(t) + \sin(2t) & \cos(2t) \\ 2 \cos(t) & \sin(3t) & 20 \end{bmatrix}. \quad (33)$$

Similar to Example 1, the computation time is limited to $[0, 10]$ s. Different from Example 1, the design parameter λ is set as 30, which is greater than that of Example 1. In addition, the initial state of $\mathbf{s}(t)$ is set as

$$\mathbf{s}(0) = [0 \ 1 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6]^T \in \mathbb{R}^{9 \times 1}. \quad (34)$$

By employing the proposed CTLUD model to solve the 3-dimensional time-varying LU decomposition problem, matrices $L(t)$ and $U(t)$ are obtained. The corresponding experiment results are obtained and shown in Figure 3. In Figure 3(a), the trajectory of $\|E(t)\|_F$ is shown. $\|E(t)\|_F$ converges to 0 with a high convergence speed. What is more, the convergence speed is much higher than that of Example 1. At about 0.5 s, $\|E(t)\|_F$ is already near 0. Thus, the following conclusion is obtained. The 3-dimensional time-varying LU decomposition problem is solved effectively through the proposed CTLUD model.

In addition, Figure 3(b) shows the trajectories of all elements of $E(t)$. From Figure 3(b), all elements of $E(t)$ converge to 0 rapidly. In particular, as for $e_{11}(0) = 200$, which is much bigger than initial values of other elements, $e_{11}(t)$ and other elements converge to 0 near-simultaneously. Therefore, the effectiveness of the proposed CTLUD model is verified by the experiment results.

Therefore, for a 3-dimensional time-varying matrix, the CTLUD model can also find its LU decomposition effectively.

5.3. Example 3. Thirdly, in order to observe the performance of model (20) for solving the high-dimensional LU decomposition problem, a 7-dimensional time-varying diagonally dominant matrix is considered, that is, (30).

In this example, the computation time is still limited to $[0, 10]$ s. Different from the previous two examples, the design parameter $3 < \lambda = 10 < 30$. What is more, the initial state of $\mathbf{s}(t)$ is set as

$$\mathbf{s}(0) = [1 \ 2 \ \dots \ 21 \ 1 \ 2 \ \dots \ 28]^T \in \mathbb{R}^{49 \times 1}. \quad (35)$$

The corresponding experiment results synthesized by the CTLUD model are shown in Figure 4. As seen from Figure 4(a), although $A(t)$ is a 7-dimensional matrix, the $\|E(t)\|_F$ has converged to 0 in about 1 s, and the convergence speed is higher than that of Example 1 and lower than that of Example 2 because the $3 < \lambda = 10 < 30$, which verifies the statements in Section 3.

What is more, the trajectories of all elements of $E(t)$ are shown in Figure 4(b). Evidently, all elements converge to 0 near-simultaneously, although the initial values of some elements are bigger than others. As for the high-dimensional and time-varying LU decomposition problem, the proposed CTLUD model is also effective.

In summary, the LU decomposition problems of three time-varying matrices with different dimensions and different initial states, are solved effectively by the CTLUD model.

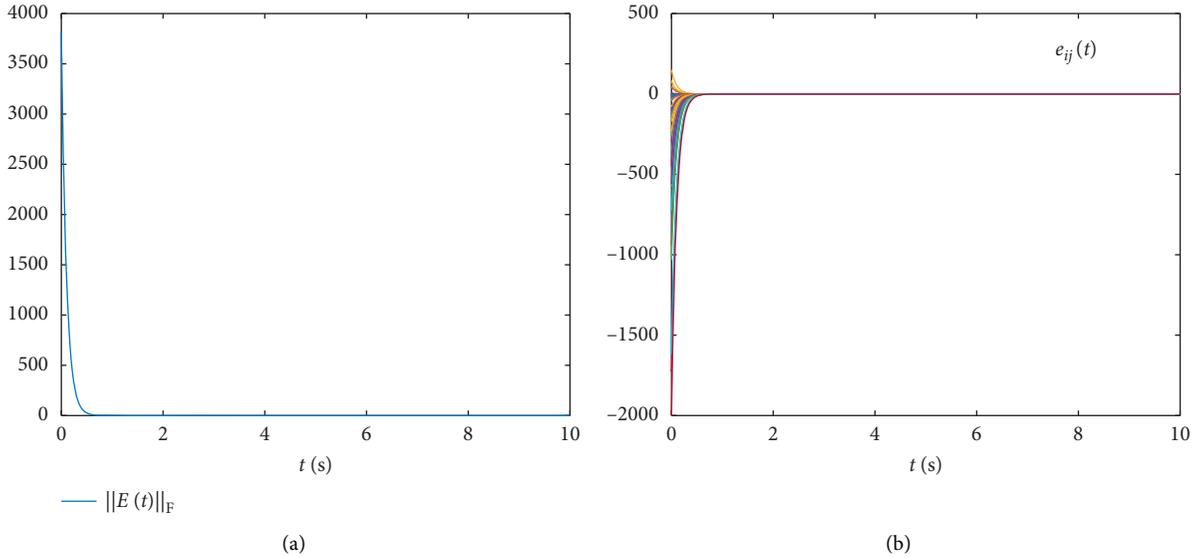


FIGURE 4: Experiment results synthesized by CTLUD model with $\lambda = 10$ and $\mathbf{s}(0) = [1 \ 2 \ \dots \ 21 \ 1 \ 2 \ \dots \ 28]^T$ for solving 7-dimensional time-varying LU decomposition problem in Example 3. (a) Frobenius-norm trajectory of error matrix $E(t)$. (b) Trajectories of $e_{11}(t)$, $e_{21}(t)$, \dots , $e_{67}(t)$, and $e_{77}(t)$.

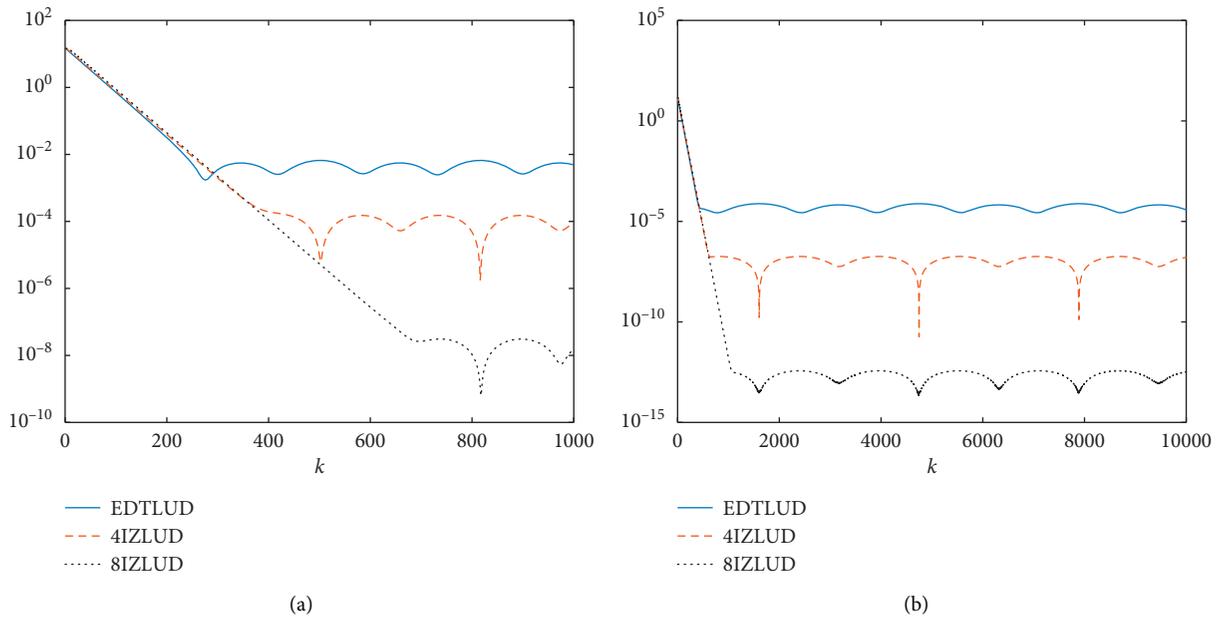


FIGURE 5: Residual errors of three discrete-time models with $g = 0.03$ and $\mathbf{s}(0) = [2 \ 1 \ 2 \ 3]^T$ for solving 2-dimensional time-varying LU decomposition problem in Example 1. (a) When $\eta = 0.01$ s. (b) When $\eta = 0.001$ s.

6. Numerical Experiments and Results of Discrete-Time Models

In this section, in order to verify the effectiveness of three proposed discrete-time models, we use EDTLUD, 4IZLUD, and 8IZLUD models to solve three time-varying LU decomposition problems in the last section, respectively. Besides, the numerical results verify the statements of Theorem 3.

For Example 1, $E(t_k) = A(t_k) - L(t_k)U(t_k)$ is the error matrix and the residual error is defined as $\|E(t_k)\|_F$. In this paper, $g = \eta\lambda$ is termed step length. In all experiments involved in this section, g is set as 0.03. The results synthesized by the three proposed discrete-time models with $\mathbf{s}(0) = [2 \ 1 \ 2 \ 3]^T$, $\mathbf{s}(t_k) = \mathbf{0}$ ($k < 0$), and $t_f = 10$ are shown in Figure 5. Specifically, Figures 5(a) and 5(b) show the results with $\eta = 0.01$ and $\eta = 0.001$, respectively. Evidently, with different values of η , the residual errors of the three models

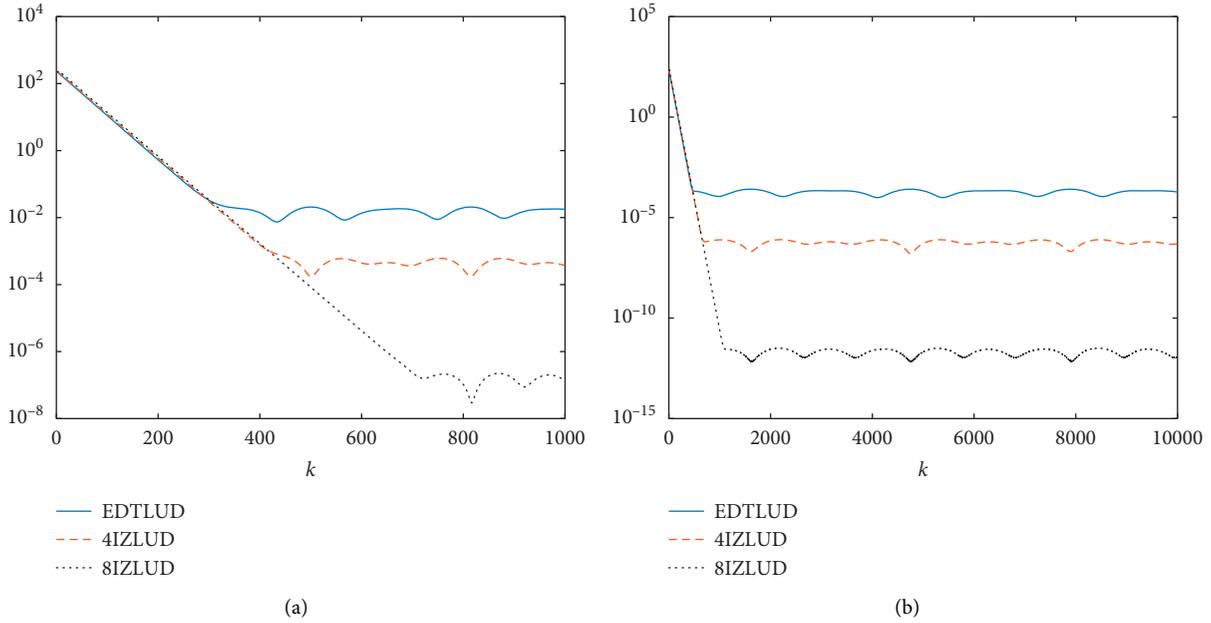


FIGURE 6: Residual errors of three discrete-time models with $g = 0.03$ and $\mathbf{s}(0) = [0 \ 1 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6]^T$ for solving 3-dimensional time-varying LU decomposition problem in Example 2. (a) When $\eta = 0.01$ s. (b) When $\eta = 0.001$ s.

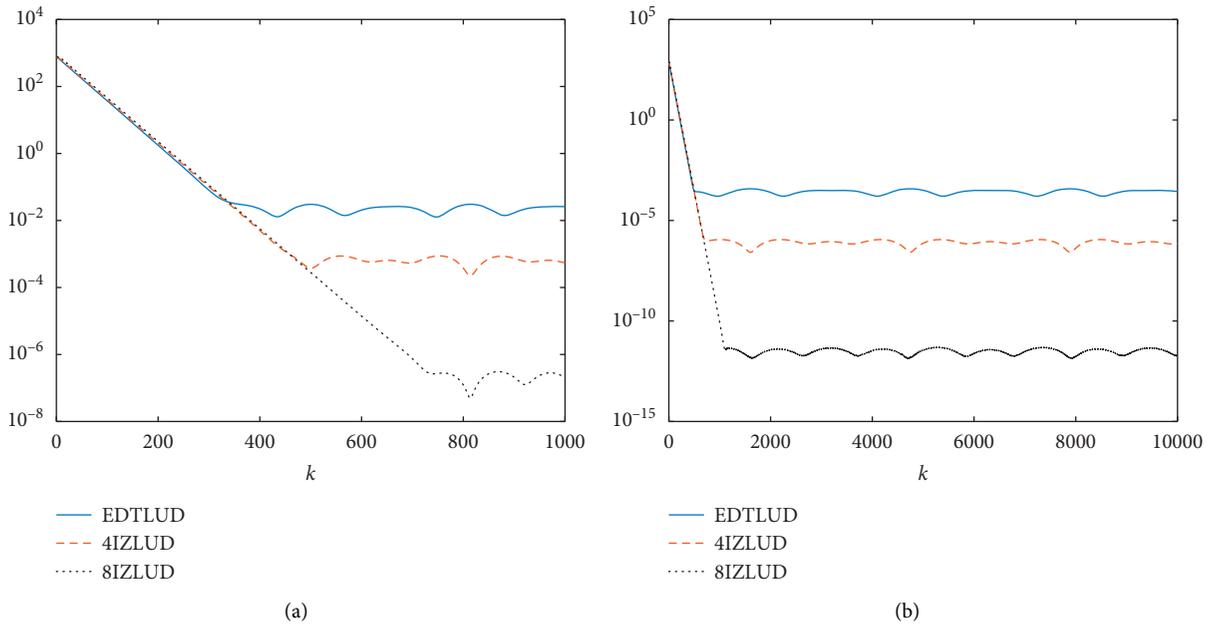


FIGURE 7: Residual errors of three discrete-time models with $g = 0.03$ and $\mathbf{s}(0) = [1 \ 2 \ \dots \ 21 \ 1 \ 2 \ \dots \ 28]^T$ for solving 7-dimensional time-varying LU decomposition problem in Example 3. (a) When $\eta = 0.01$ s. (b) When $\eta = 0.001$ s.

converge to near 0 but with different precision as time goes by. Compared with EDTLUD and 4IZLUD models, the 8IZLUD model has the highest precision for solving the time-varying LU decomposition problem.

It is worth mentioning that, as the value of η decreases by 10 times, the maximal steady-state residual errors of three models reduce by 10^2 , 10^3 , and 10^5 times, respectively, which coincides with Theorem 4.

For Example 2 and Example 3, the numerical experiment results synthesized by three models with $\mathbf{s}(0) = [0 \ 1 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6]^T$, $\mathbf{s}(0) = [1 \ 2 \ \dots \ 21 \ 1 \ 2 \ \dots \ 28]^T$, and $t_f = 10$ are shown in Figures 6 and 7, respectively. From Figures 6 and 7, we can obtain a conclusion similar to the one above; that is, the 8IZLUD model has the highest precision among the three discrete-time models.

In summary, numerical results illustrate that the three discrete-time models are effective for solving the time-varying LU decomposition problem.

7. Conclusion

In this paper, the time-varying LU decomposition problem has been discussed and investigated. By applying the ZNN method, an effective CTLUD model has been proposed. Then, based on the Euler formula, 4-instant ZeaD formula, and 8-instant ZeaD formula, three discrete-time LU decomposition models have been proposed and discussed by discretizing the proposed CTLUD model. Furthermore, in order to verify the effectiveness of the proposed models, three time-varying matrices with different dimensions have been presented. By using the proposed models to perform the LU decomposition of the presented matrices, the corresponding results have been obtained, shown, and discussed. A conclusion has been obtained that the proposed models are effective for solving the time-varying LU decomposition problem, and the 8IZLUD model has the highest precision among the three discrete-time models. More investigations and experiments about other time-varying matrix decompositions and corresponding applications would be future work directions.

Appendix

In this appendix, four lemmas for a linear N -step method are presented below [44–46].

Lemma 2. *As for a linear N -step method described by $\sum_{p=0}^N \beta_p x_{j+p} = \eta \sum_{p=0}^N \alpha_p \sigma_{j+p}$, whose 0-stability can be checked by determining the roots of its characteristic polynomial $P_N(\gamma) = \sum_{p=0}^N \beta_p \gamma^p$, if every root of $P_N(\gamma) = 0$ satisfies $|\gamma| \leq 1$ with $|\gamma| = 1$ being unique, the linear N -step method has strong 0-stability (i.e., is strongly 0-stable).*

Lemma 3. *With $q > 0$, if the truncation error for the exact solution is of order $O(\eta^{q+1})$, the N -step method is generally consistent with order q .*

Lemma 4. *For any time instant $t \in [0, t_f]s$, as $\eta \rightarrow 0$, if and only if a linear N -step method is 0-stable and consistent, the method is convergent, which is written as $x_{[t/\eta]} \rightarrow x^*(t)$ mathematically. That is, 0-stability plus consistency means convergence, which is also known as the Dahlquist equivalence theorem.*

Lemma 5. *A linear N -step method is convergent with the order of its truncation error, if the method is 0-stable and consistent.*

Data Availability

The experiment data used to support the findings of this study are included within the article.

Disclosure

Yunong Zhang is a co-first author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was aided 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 also the Shenzhen Science and Technology Plan Project (no. JCYJ20170818154936083).

References

- [1] J. Choi, J. J. Dongarra, L. S. Ostrouchov, A. P. Petitet, D. W. Walker, and R. C. Whaley, "Design and implementation of the ScaLAPACK LU, QR, and cholesky factorization routines," *Scientific Programming*, vol. 5, no. 3, pp. 173–184, 1996.
- [2] X. Li, J. Niu, M. K. Khan, and Z. Wang, "Applying LU decomposition of matrices to design anonymity bilateral remote user authentication scheme," *Mathematical Problems in Engineering*, vol. 2013, Article ID 910409, , 2013.
- [3] R. Wu and X. Xie, "Two-stage column block parallel LU factorization algorithm," *IEEE Access*, vol. 8, pp. 2645–2655, 2019.
- [4] Y.-T. Tsai, C.-Y. Lee, and M.-K. Tsai, "Levelized incomplete LU method and its application to semiconductor device simulation," *Solid-State Electronics*, vol. 44, no. 6, pp. 1069–1075, 2000.
- [5] S. Han and Z. Zhang, "A novel zero-watermark algorithm based on LU decomposition in NSST domain," in *Proceedings of the IEEE International Conference on Signal Processing*, pp. 1592–1596, Beijing, China, April 2013.
- [6] M. Eljammaly, Y. Hanafy, A. Wahdan, and A. Bayoumi, "Hardware implementation of LU decomposition using dataflow architecture on FPGA," in *Proceedings of the International Conference on Computer Science and Information Technology*, pp. 298–302, Amman, Jordan, March 2013.
- [7] M. M. M. Omar, K. A. Eskaf, and B. A. Ghreiwati, "Multi-dimensional frequency estimation using LU decomposition eigenvector-based algorithm," *Annals of Telecommunications*, vol. 75, no. 1–2, pp. 17–25, 2019.
- [8] H. Shao, J. Hu, H. Guo, F. Ye, W. Lu, and Z. Nie, "Fast simulation of array structures using T-EPA with hierarchical LU decomposition," *IEEE Antennas and Wireless Propagation Letters*, vol. 11, pp. 1556–1559, 2012.
- [9] R. Fang, S. Dong, B. Zhu, Q. Xu, and Y. Song, "Improved loop-current-method power flow algorithm for distribution network based on accelerated parallel LU decomposition," *Power System Technology*, vol. 46, no. 6, pp. 2179–2186, 2019.

- [10] O. Jane, H. G. Ilk, and E. Elbase, "A secure and robust watermarking algorithm based on the combination of DWT, SVD, and LU decomposition with Arnold's cat map approach," in *Proceedings of the International Conference on Electrical and Electronics Engineering*, pp. 306–310, Bursa, Turkey, November 2013.
- [11] K. Wang, X.-F. Gong, and Q.-H. Lin, "Complex non-orthogonal joint diagonalization based on LU and LQ decompositions," *Latent Variable Analysis and Signal Separation*, vol. 7191, pp. 50–57, 2012.
- [12] K. He, S. X.-D. Tan, H. Wang, and G. Shi, "GPU-accelerated parallel sparse LU factorization method for fast circuit analysis," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 3, pp. 1140–1150, 2016.
- [13] R. Kastner, "An LU-spatial decomposition method," in *Proceedings of the Antennas and Propagation Society International Symposium*, pp. 864–867, Ann Arbor, MI, USA, June 1993.
- [14] T. Yigal and K. Raphael, "A recursive LU decomposition algorithm," *Microwave and Optical Technology Letters*, vol. 7, no. 2, pp. 48–52, 1994.
- [15] M. I. Bueno and C. R. Johnson, "Minimum deviation, quasi-LU factorization of nonsingular matrices," *Linear Algebra and its Applications*, vol. 427, no. 1, pp. 99–118, 2007.
- [16] L. Xiao, "A finite-time convergent Zhang neural network and its application to real-time matrix square root finding," *Neural Computing and Applications*, vol. 31, no. S2, pp. 793–800, 2019.
- [17] 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.
- [18] Y. Zhang and C. Yi, *Zhang Neural Networks and Neural-Dynamic Method*, Nova Science Press, New York, NY, USA, 2011.
- [19] S. Kawakami and T. Ohtsuki, "Localization using iterative angle of arrival method sharing snapshots of coherent sub-arrays," *EURASIP Journal on Advances in Signal Processing*, vol. 46, pp. 1–7, 2011.
- [20] A. Noroozi, A. H. Oveis, S. M. Hosseini, and M. A. Sebt, "Improved algebraic solution for source localization from TDOA and FDOA measurements," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 352–355, 2018.
- [21] A. G. Dempster and E. Cetin, "Interference localization for satellite navigation systems," *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1318–1326, 2016.
- [22] S. Qiao, X.-Z. Wang, and Y. Wei, "Two finite-time convergent Zhang neural network models for time-varying complex matrix Drazin inverse," *Linear Algebra and its Applications*, vol. 542, pp. 101–117, 2018.
- [23] L. Xiao, B. Liao, S. Li, and K. Chen, "Nonlinear recurrent neural networks for finite-time solution of recurrent neural networks for finite-time solution of general time-varying linear matrix equations," *Neural Networks*, vol. 98, pp. 102–113, 2017.
- [24] D. Guo, Z. Nie, and L. Yan, "Theoretical analysis, numerical verification and geometrical representation of new three-step DTZD algorithm for time-varying nonlinear equations solving," *Neurocomputing*, vol. 214, pp. 516–526, 2016.
- [25] N. Liu and S. Qin, "A novel neurodynamic approach to constrained complex-variable pseudoconvex optimization," *IEEE Transactions on Cybernetics*, vol. 49, no. 11, pp. 3946–3956, 2019.
- [26] S. Qin and X. Xue, "A two-layer recurrent neural network for nonsmooth convex optimization problems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 6, pp. 1149–1160, 2015.
- [27] 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, 2017.
- [28] N. Liu and S. Qin, "A neurodynamic approach to nonlinear optimization problems with affine equality and convex inequality constraints," *Neural Networks*, vol. 109, pp. 147–158, 2019.
- [29] L. Jin, S. Li, and B. Hu, "RNN models for dynamic matrix inversion: a control-theoretical perspective," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 1, pp. 189–199, 2018.
- [30] J. Li, Y. Zhang, and M. Mao, "General square-pattern discretization formulas via second-order derivative elimination for zeroing neural network illustrated by future optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 891–901, 2019.
- [31] L. Jin, S. Li, H. Wang, and Z. Zhang, "Nonconvex projection activated zeroing neurodynamic models for time-varying matrix pseudoinversion with accelerated finite-time convergence," *Applied Soft Computing*, vol. 62, pp. 840–850, 2018.
- [32] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
- [33] Y. Zhang, L. Ming, H. Huang, J. Chen, and Z. Li, "Time-varying Schur decomposition via Zhang neural dynamics," *Neurocomputing*, vol. 419, pp. 251–258, 2020.
- [34] J. Chen and Y. Zhang, "Online singular value decomposition of time-varying matrix via zeroing neural dynamics," *Neurocomputing*, vol. 383, pp. 314–323, 2020.
- [35] 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, 2012.
- [36] Y. Zhang, X. Liu, Y. Ling, M. Yang, and H. Huang, "Continuous and discrete Zeroing dynamics models using JMP function array and design formula for solving time-varying Sylvester-transpose matrix inequality," *Numerical Algorithms*, vol. 86, no. 4, pp. 1591–1614, 2021.
- [37] 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.
- [38] Y. Zhang, Y. Yang, N. Tan, and B. Cai, "Zhang neural network solving for time-varying full-rank matrix Moore-Penrose inverse," *Computing*, vol. 92, no. 2, pp. 97–121, 2011.
- [39] Y. Zhang, M. Yang, H. Huang, M. Xiao, and H. Hu, "New discrete-solution model for solving future different-level linear inequality and equality with robot manipulator control," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 1975–1984, 2019.
- [40] 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.
- [41] C. Hu, X. Kang, and Y. Zhang, "Three-step general discrete-time Zhang neural network design and application to time-variant matrix inversion," *Neurocomputing*, vol. 306, pp. 108–118, 2018.
- [42] 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*, pp. 6262–6267, Shenyang, China, June 2018.

- [43] L. Jin and Y. Zhang, “Discrete-time Zhang neural network of $O(\tau^3)$ pattern for time-varying matrix pseudoinversion with application to manipulator motion generation,” *Neuro-computing*, vol. 142, pp. 165–173, 2014.
- [44] J. Chen and Y. Zhang, “Discrete-time ZND models solving ALRMPC via eight-instant general and other formulas of ZeaD,” *IEEE Access*, vol. 7, pp. 125909–125918, 2019.
- [45] E. Suli and D. F. Mayers, *An Introduction to Numerical Analysis*, Cambridge University Press, Cambridge, UK, 2003.
- [46] D. F. Griffiths and D. J. Higham, *Numerical Methods for Ordinary Differential Equations: Initial Value Problems*, Springer Press, London, UK, 2010.