Time-Varying Matrix Inversion Based on Lagrange-Type Finite Difference

Jian Li, Shuang Pan, Jingjing Chen, Wenjing Sun, Qing Chen, Zhipeng Fu

Abstract—**Discretization is a vital component of solving time-varying matrix inversion, which determines the effectiveness, the ability of real-time computation and the precision. Zeroing neural dynamics is a classical method for solving time-varying matrix inversion. In that method, conventional Lagrange-type finite difference formulas cannot be used for discretization because of the constraint of 0-stability. It thus leads to a lot of research in the development of effective discretization formulas. The developed formulas are uniformly called ZeaD. In this work, we develop a new method instead of zeroing neural dynamics to solve time-varying matrix inversion. We obtain a direct calculation scheme instead of a form of differential equation. Thus, Lagrange-type formulas are still effective for discretization in our method. Finally, a series of models based on Lagrange-type formulas as well as ZeaD formulas are proposed. Note that, compared with ZeaD formulas, Lagrange-type formulas have higher precision when the numbers of instances are the same.**

Index Terms—**Time-varying matrix inversion, Lagrange-type finite difference formulas, discretization, zeroing neural dynamics, 0-stability.**

I. Introduction

MATRIX inversion is a fundamental mathematic problem. It is widely encountered in scientific and engineering fields [1]–[5]. Numerous methods have been developed and investigated for solving this problem [6]–[9]. For example, in [6] a recurrent implicit dynamics was presented for online matrix inversion. In [8] a completely block recursive algorithm with low complexity was introduced for generalized matrix inversion. In [9], different stochastic algorithms were presented to obtain matrix inversion. The quoted methods mainly aim to solve time-invariant matrix inversion in terms of the enhancement of computation precision and the reduction in time consumption.

Manuscript received January 14, 2022; revised June 17, 2022. This work was supported in part by the National Natural Science Foundation of China (No. 62006205), by the China Postdoctoral Science Foundation funded project (No. 2021TQ0299), by Nanhu Scholars Program for Young Scholars of XYNU and Scientific Research Fund for College Students of XYNU (No. 2022-D-XS-146). (Corresponding author: Jian Li.)

Jian Li is a Lecturer of the School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China (phone: 0376-6392285, e-mail: lijcit@xynu.edu.cn).

Shuang Pan is a postgraduate student of the School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China (e-mail: 2085929142@qq.com).

Jingjing Chen is an undergraduate student of the School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China (e-mail: 3040569976@qq.com).

Wenjing Sun is an undergraduate student of the School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China (e-mail: 1684994701@qq.com).

Qing Chen is an undergraduate student of the School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China (e-mail: 2499024163@qq.com).

Zhipeng Fu is a postgraduate student of the School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China (e-mail: 1254747132@qq.com).

Time-varying matrix inversion originates from time-invariant matrix inversion, and differs from the later by adding time variable [10]–[13]. Time-varying matrix inversion is also a fundamental mathematic problem, which has been investigated back in 1997 and even earlier [14], [15]. Solution of time-varying matrix inversion varies over time. Conventional methods aiming to solve time-invariant matrix inversion have disappointed performance when solving time-varying matrix inversion because of time-delay [16]–[18].

Zeroing neural dynamics has been developed recent years to specially solve time-varying problems [12], [19]–[21]. Since zeroing neural dynamics was developed, various types of time-varying problems have been solved [22]–[26], such as time-varying matrix inversion, linear equation system, nonlinear equation system, nonlinear optimization. There are three steps to develop models by zeroing neural dynamics method [27]–[29]. First, we construct vector or matrix error function according to problem to be solved. Then, we design dynamic formula to zero out every elements of error function. In this step, continuous-time model can be obtained, which has a form of differential equation. Finally, we develop discretization formula to discretize continuous-time model and obtain the final model to solve the original problem. During the developing process, the third step (discretization) is quite important, which determines the effectiveness, the ability of real-time computation and the precision [30]–[34]. First, discretization formula must be one-step-ahead, which makes the model has the ability of real-time computation. Second, discretization formula must satisfy the constraint of 0-stability, such that the model is convergent and effective. Thirdly, discretization formula with higher precision leads to model with higher precision.

As aforementioned, discretization is quite important and challenging. A lot of researchers have made a great effort to this point and some effective formulas have been developed [30]–[34]. For example, in [30], a formula with four instances used was developed, which has square precision. In [31], a third-order-accuracy formula was developed with five instances utilized. In [32], a fourth-order-accuracy formula was developed with eight instances utilized. In [33], a general four-instant discretization formula was proposed, which has second order accuracy. In [34], a general five-instant discretization formula was proposed, which has third order accuracy. Those formulas are developed in the frame of zeroing neural dynamics and are called ZeaD.

Finite difference plays an important role in numerical computation [35]–[39]. Compared with ZeaD formulas, Lagrange-type finite difference formulas have been developed earlier [35]. In addition, Lagrange-type finite difference formulas have higher precision when the numbers of instances are the same. Unfortunately, Lagrange-type finite difference formulas are not suitable in zeroing neural dynamics because of the constraint of 0-stability [40], [41].

In this work, we develop a new method instead of zeroing neural dynamics to solve time-varying matrix inversion. We obtain a direct calculation formula instead of a form of differential equation. Thus, Lagrange-type formulas are still effective for discretization in our method. A series of models based on Lagrange-type formulas as well as ZeaD formulas are proposed. The contributions of this work are listed as follows.

(1) We develop a new method to solve time-varying matrix inversion, which differs from zeroing neural dynamics method.
(2) A series of models based on Lagrange-type finite difference formulas as well as ZeaD formulas are proposed.
(3) Both cases of derivative information known and unknown for time-varying matrix are investigated.

II. Problem Formulation

The significant point of solving time-varying problems is real time. For the convenience of problem formulation, the problem is given in a discrete-time form as follows [28], [42]–[44]. Given a sequence of matrices $C(\tau_j)$ at time instances $\tau_j \leq \tau_k$ (i.e., past and current information of time stream), we should obtain the discrete-time matrix inverse $Y(\tau_{k+1})$ of $C(\tau_{k+1})$ (i.e., future information) on each computational time interval $[\tau_k, \tau_{k+1}) \subseteq [0, \tau_{\mathrm{f}}]$ so that

$$C(\tau_{k+1})Y(\tau_{k+1}) - I = \mathbf{0}, \tag{1}$$

where $C(\tau_{k+1}) = C((k+1)\iota) \in \mathbb{R}^{n\times n}$ is a time-varying equidistant matrix sequence; and $Y(\tau_{k+1}) \in \mathbb{R}^{n\times n}$ is unknown, which needs to be computed in real-time for each time interval $[\tau_k, \tau_{k+1}) \subseteq [0, \tau_{\mathrm{f}}]$. I is identity matrix; and $\mathbf{0} \in \mathbb{R}^{n\times n}$ is the zero matrix. In addition, $k = 0, 1, \cdots$ denotes the updating index; τ_{f} denotes the task duration; and ι denotes the constant sampling gap of the time-varying matrix sequence $C(\tau_{k+1})$.

III. Development and Analyses of Models

In this section, we present the development process of our models. Then, we discuss and compare the proposed models and conventional models.

A. Lagrange-Type Finite Difference Versus ZeaD

Lagrange-type finite difference formulas are classical approximation of first-order derivative. They have one-step ahead form, which is necessary for discretization. However, when Lagrange-type finite difference formulas are employed for time-discretization in zeroing neural dynamics method, they lead to instability of solution. Thus, many researchers have found some effective time-discretization formulas in zeroing neural dynamics method. The developed formulas are called ZeaD. Parts of Lagrange-type finite difference formulas and ZeaD formulas are shown in Table I for the convenience of comparisons. First, both kinds of formulas have one-step-ahead form, which is needed for discretization. Second, when we utilize zeroing neural dynamics method to solve time-varying problems, ZeaD formulas lead to stable solutions, while Lagrange-type finite difference formulas fail to do it. Third, Lagrange-type finite difference formulas have higher precision than ZeaD formulas when the same number of instant is used. For example, the first Lagrange-type finite difference formula in Table I (i.e., L1) only uses two instances and has precision of $O(\iota^2)$. However, the second ZeaD formula in Table I (i.e., Z2) uses four instances to achieve $O(\iota^2)$ precision. Similarly, the third Lagrange-type finite difference formula in Table I (i.e., L3) only uses five instances to achieve precision of $O(\iota^4)$. However, the last ZeaD formula in Table I (i.e., Z4) uses seven instances to achieve $O(\iota^4)$ precision. In this work, we break the limit of zeroing neural dynamics method and use a new method to solve time-varying matrix inversion, such that Lagrange-type finite difference formulas can be employed for discretization. It means that the proposed models to solve time-varying matrix inversion have higher precision when the same number of instant is used compared with ZeaD formulas.

B. TVMI-K Models

Considering that time-varying matrix inversion (1) is a problem of time stream and time is continuous in reality, we investigate the continuous-time form of (1) as follows:

$$C(\tau)Y(\tau) - I = \mathbf{0}. \tag{2}$$

Defining theoretical solution of (2) as $Y^*(\tau)$, we have the following equation:

$$C(\tau)Y^*(\tau) - I = \mathbf{0}. \tag{3}$$

Direct derivation of (3) yields

$$C(\tau)\dot{Y}^*(\tau) + \dot{C}(\tau)Y^*(\tau) = \mathbf{0}. \tag{4}$$

Equation (4) can be rewritten as

$$\dot{Y}^*(\tau) = -C^+(\tau)\dot{C}(\tau)Y^*(\tau), \tag{5}$$

where the matrix operator $^+$ denotes the inverse of a matrix. In addition, we know that $Y^*(\tau) = C^+(\tau)$, and thus, equation (5) can be rewritten as

$$\dot{Y}^*(\tau) = -C^+(\tau)\dot{C}(\tau)C^+(\tau). \tag{6}$$

Then, we can use discretization formulas including Lagrange-type finite difference and ZeaD formulas to discretize equation (6). Specifically, using L1 formula in Table I for discretization, we have

$$\frac{Y^*(\tau_{k+1}) - Y^*(\tau_{k-1})}{2\iota} = -C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k), \tag{7}$$

which is rewritten as

$$Y^*(\tau_{k+1}) = -2\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k) + Y^*(\tau_{k-1}). \tag{8}$$

We know that $Y^*(\tau_{k-1}) = C^+(\tau_{k-1})$, and thus equation (8) is rewritten as

$$Y^*(\tau_{k+1}) = -2\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k) + C^+(\tau_{k-1}). \tag{9}$$

Finally, the Lagrange-type model based on L1 formula to solve time-varying matrix inversion (1) is obtained as

$$Y(\tau_{k+1}) = -2\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k) + C^+(\tau_{k-1}). \tag{10}$$

Note that model (10) to solve time-varying matrix inversion (1) needs to know the derivative information, i.e., value of $\dot{C}(\tau_k)$, and is based on Lagrange-type finite difference (i.e.,

TABLE I
LAGRANGE-TYPE FINITE DIFFERENCE [40] VERSUS ZEAD FOR APPROXIMATING $\dot{x}(\tau_k)$ WITH DIFFERENT TRUNCATION ERRORS.

	Name	Expression	Instant number	Truncation error
Lagrange	L1	$\dot{x}(\tau_k)=\frac{x(\tau_{k+1})-x(\tau_{k-1})}{2\iota}$	2	$O(\iota^2)$
	L2	$\dot{x}(\tau_k)=\frac{2x(\tau_{k+1})+3x(\tau_k)-6x(\tau_{k-1})+x(\tau_{k-2})}{6\iota}$	4	$O(\iota^3)$
	L3	$\dot{x}(\tau_k)=\frac{3x(\tau_{k+1})+10x(\tau_k)-18x(\tau_{k-1})+6x(\tau_{k-2})-x(\tau_{k-3})}{12\iota}$	5	$O(\iota^4)$
	L4	$\dot{x}(\tau_k)=\frac{12x(\tau_{k+1})+65x(\tau_k)-120x(\tau_{k-1})+60x(\tau_{k-2})-20x(\tau_{k-3})+3x(\tau_{k-4})}{60\iota}$	6	$O(\iota^5)$
	L5	$\dot{x}(\tau_k)=\frac{10x(\tau_{k+1})+77x(\tau_k)-150x(\tau_{k-1})+100x(\tau_{k-2})-50x(\tau_{k-3})+15x(\tau_{k-4})-2x(\tau_{k-5})}{60\iota}$	7	$O(\iota^6)$
ZeaD	Z1	$\dot{x}(\tau_k)=\frac{3x(\tau_{k+1})-2x(\tau_k)-x(\tau_{k-1})}{4\iota}$	3	$O(\iota)$
	Z2	$\dot{x}(\tau_k)=\frac{2x(\tau_{k+1})-3x(\tau_k)+2x(\tau_{k-1})-x(\tau_{k-2})}{2\iota}$	4	$O(\iota^2)$
	Z3	$\dot{x}(\tau_k)=\frac{8x(\tau_{k+1})+x(\tau_k)-6x(\tau_{k-1})-5x(\tau_{k-2})+2(\tau_{k-3})}{18\iota}$	5	$O(\iota^3)$
	Z4	$\dot{x}(\tau_k)=\frac{83x(\tau_{k+1})+45x(\tau_k)-84x(\tau_{k-1})-82x(\tau_{k-2})+27x(\tau_{k-3})+21x(\tau_{k-4})-10x(\tau_{k-5})}{216\iota}$	7	$O(\iota^4)$

L1 formula). Thus, it is termed as L1-TVMI-K model. When we use different Lagrange-type finite difference formulas shown in Table I, more other TVMI-K models with higher precision can be obtained.

In addition, when we use ZeaD formulas for discretization, the corresponding models are obtained. For example, when we use Z2 formula in Table I for discretization, we have

$$\begin{aligned}&\frac{2Y^*(\tau_{k+1})-3Y^*(\tau_k)+2Y^*(\tau_{k-1})-Y^*(\tau_{k-2})}{2\iota}\\&=-C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k),\end{aligned}\tag{11}$$

which is rewritten as

$$\begin{aligned}Y^*(\tau_{k+1})=&-\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k)+\frac{3}{2}Y^*(\tau_k)\\&-Y^*(\tau_{k-1})+\frac{1}{2}Y^*(\tau_{k-2}).\end{aligned}\tag{12}$$

We know that $Y^*(\tau_k)=C^+(\tau_k)$, $Y^*(\tau_{k-1})=C^+(\tau_{k-1})$ and $Y^*(\tau_{k-2})=C^+(\tau_{k-2})$, and thus equation (12) is rewritten as

$$\begin{aligned}Y^*(\tau_{k+1})=&-\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k)+\frac{3}{2}C^+(\tau_k)\\&-C^+(\tau_{k-1})+\frac{1}{2}C^+(\tau_{k-2}).\end{aligned}\tag{13}$$

Finally, the ZeaD-type model based on Z2 formula to solve time-varying matrix inversion (1) is obtained as

$$\begin{aligned}Y(\tau_{k+1})=&-\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k)+\frac{3}{2}C^+(\tau_k)\\&-C^+(\tau_{k-1})+\frac{1}{2}C^+(\tau_{k-2}).\end{aligned}\tag{14}$$

It is termed as Z2-TVMI-K model. When we use different ZeaD formulas shown in Table I, more other TVMI-K models with higher precision are obtained. The TVMI-K models based on Lagrange-type finite difference and ZeaD formulas are listed in Table II.

Theoretical analysis is shown as follows to guarantee the effectiveness and precision of proposed models.

Theorem 1: If matrix $C(\tau)$ in TVMI problem (1) is nonsingular and has high-order derivatives, residual error of L1-TVMI-K model (10) to solve this problem is $O(\iota^3)$, where residual error is defined as $\|C(\tau_{k+1})Y(\tau_{k+1})-I\|$. In addition, the orders of residual errors of different TVMI-K models based on different lagrange-type finite difference formulas are one-rank higher than the error orders of corresponding formulas.

Proof: When we consider the truncation error during the derivation process of L1-TVMI-K model (10), equation (7) is exactly

$$\frac{Y^*(\tau_{k+1})-Y^*(\tau_{k-1})}{2\iota}+\mathbf{O}(\iota^2)=-C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k),$$

which is rewritten as follows by multiplying both sides by 2ι:

$$\begin{aligned}Y^*(\tau_{k+1})=&-2\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k)\\&+Y^*(\tau_{k-1})+\mathbf{O}(\iota^3).\end{aligned}\tag{15}$$

Then equation (15) is rewritten as

$$\begin{aligned}Y^*(\tau_{k+1})=&-2\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k)\\&+C^+(\tau_{k-1})+\mathbf{O}(\iota^3).\end{aligned}\tag{16}$$

We know that L1-TVMI-K model (10) is

$$Y(\tau_{k+1})=-2\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k)+C^+(\tau_{k-1}).\tag{17}$$

Combing (17) and (16) yields

$$\|Y(\tau_{k+1})-Y^*(\tau_{k+1})\|=\|\mathbf{O}(\iota^3)\|.\tag{18}$$

Then, the residual error

$$\begin{aligned}&\|C(\tau_{k+1})Y(\tau_{k+1})-I\|\\=&\|C(\tau_{k+1})(Y^*(\tau_{k+1})+\mathbf{O}(\iota^3))-I\|\\=&\|C(\tau_{k+1})Y^*(\tau_{k+1})-I+\mathbf{O}(\iota^3)\|\\=&\|\mathbf{O}(\iota^3)\|\\=&O(\iota^3).\end{aligned}\tag{19}$$

It is proved that residual error of L1-TVMI-K model (10) to solve TVMI problem (1) is $O(\iota^3)$.

In addition, focusing on equation (15), we find that the truncation error is $\mathbf{O}(\iota^3)$, while the corresponding L1 formula has $O(\iota^2)$ truncation error. It is just because the multiplication of both sides by 2ι. Similarly, when we use L2 formula for discretization, truncation error of L2-TVMI-K model is $\mathbf{O}(\iota^4)$ because the truncation error of L2 model is $O(\iota^3)$. Thus, it is concluded that the orders of residual errors

TABLE II
DIFFERENT MODELS TO SOLVE TIME-VARYING MATRIX INVERSION (1) USING DIFFERENT DISCRETIZATION FORMULAS INCLUDING LAGRANGE-TYPE FINITE DIFFERENCE AND ZEAD WITH VALUE OF $\dot{C}(\tau_k)$ KNOWN.

Discretization	Model	Truncation error
L1	$Y(\tau_{k+1}) = -2\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k) + C^+(\tau_{k-1})$	$O(\iota^3)$
L2	$Y(\tau_{k+1}) = -3\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k) - \frac{3}{2}C^+(\tau_k) + 3C^+(\tau_{k-1}) - \frac{1}{2}C^+(\tau_{k-2})$	$O(\iota^4)$
L3	$Y(\tau_{k+1}) = -4\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k) - \frac{10}{3}C^+(\tau_k) + 6C^+(\tau_{k-1}) - 2C^+(\tau_{k-2}) + \frac{1}{3}C^+(\tau_{k-3})$	$O(\iota^5)$
L4	$Y(\tau_{k+1}) = -5\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k) - \frac{65}{12}C^+(\tau_k) + 10C^+(\tau_{k-1}) - 5C^+(\tau_{k-2}) + \frac{5}{3}C^+(\tau_{k-3}) - \frac{1}{4}C^+(\tau_{k-4})$	$O(\iota^6)$
L5	$Y(\tau_{k+1}) = -6\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k) - \frac{77}{10}C^+(\tau_k) + 15C^+(\tau_{k-1}) - 10C^+(\tau_{k-2}) + 5C^+(\tau_{k-3}) - \frac{3}{2}C^+(\tau_{k-4}) + \frac{1}{5}C^+(\tau_{k-5})$	$O(\iota^7)$
Z1	$Y(\tau_{k+1}) = -\frac{4}{3}\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k) + \frac{2}{3}C^+(\tau_k) + \frac{1}{3}C^+(\tau_{k-1})$	$O(\iota^2)$
Z2	$Y(\tau_{k+1}) = -\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k) + \frac{3}{2}C^+(\tau_k) - C^+(\tau_{k-1}) + \frac{1}{2}C^+(\tau_{k-2})$	$O(\iota^3)$
Z3	$Y(\tau_{k+1}) = -\frac{9}{4}\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k) - \frac{1}{8}C^+(\tau_k) + \frac{3}{4}C^+(\tau_{k-1}) + \frac{5}{8}C^+(\tau_{k-2}) - \frac{1}{4}C^+(\tau_{k-3})$	$O(\iota^4)$
Z4	$Y(\tau_{k+1}) = -\frac{216}{83}\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k) - \frac{45}{83}C^+(\tau_k) + \frac{84}{83}C^+(\tau_{k-1}) + \frac{82}{83}C^+(\tau_{k-2}) - \frac{27}{83}C^+(\tau_{k-3}) - \frac{21}{83}C^+(\tau_{k-4}) + \frac{10}{83}C^+(\tau_{k-5})$	$O(\iota^5)$

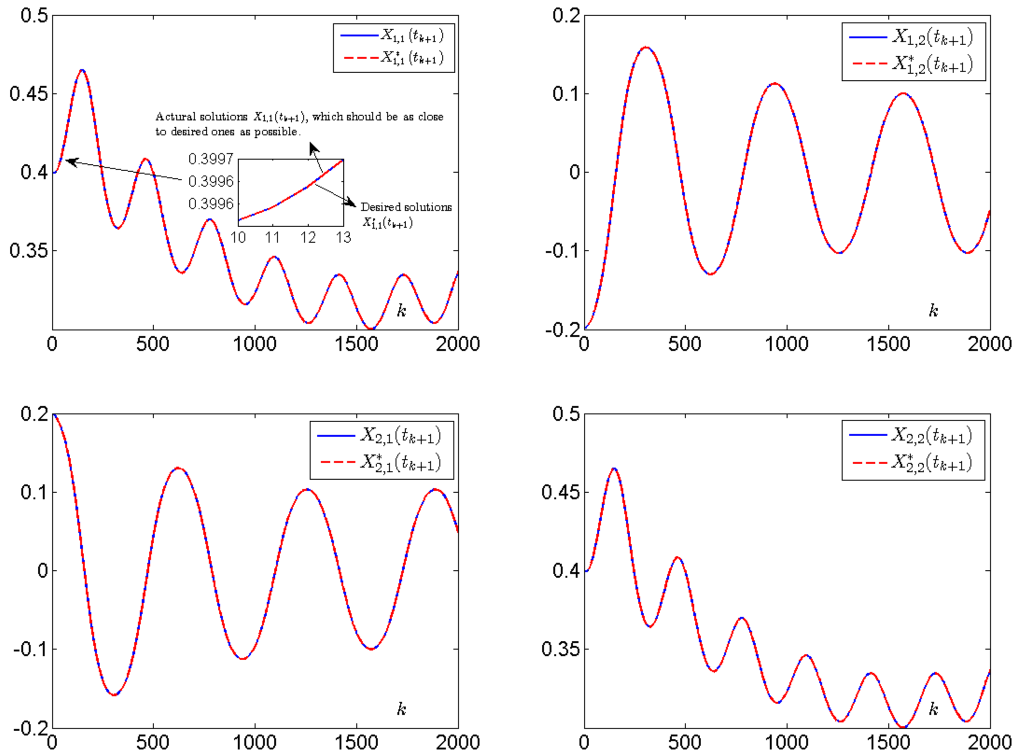


Fig. 1. Trajectories of four elements of solution matrix $Y(\tau_{k+1})$ generated by L1-TVMI-K model as well as theoretical solution paths, i.e., $Y^*(\tau_{k+1})$, with sampling gap $\iota = 0.01$ s.

of different TVMI-K models based on different lagrange-type finite difference formulas are one-rank higher than the error orders of corresponding formulas. ■

C. TVMI-U Models

Focusing on the aforementioned TVMI-K models, they all need to know the value of $\dot{C}(\tau_k)$ for each updating. However, this information may be unknown in some applications. Thus, we investigate the case of unknown $\dot{C}(\tau_k)$ and propose some TVMI-U models.

Based on previous work [28], we use backward finite difference formulas to approximate $\dot{C}(\tau_k)$ because only current and past information is known during each calculative step. In addition, we employ different backward formulas for different models to avoid loss of precision. Specifically, for L1-TVMI-K model (10) and Z2-TVMI-K model (14), we employ the following backward finite difference formula to approximate $\dot{C}(\tau_k)$:

$$\dot{u}(\tau_k) = \frac{3u(\tau_k) - 4u(\tau_{k-1}) + u(\tau_{k-2})}{2\iota}. \tag{20}$$

It has a truncation error of $O(\iota^2)$, which avoids the loss of precision. Specifically, using backward finite difference formula (20) to approximate $\dot{C}(\tau_k)$, we have

$$\dot{C}(\tau_k) = \frac{3C(\tau_k) - 4C(\tau_{k-1}) + C(\tau_{k-2})}{2\iota}. \tag{21}$$

Combining (21) and L1-TVMI-K model (10) yields

$$\begin{aligned} Y(\tau_{k+1}) &= C^+(\tau_{k-1}) \\ &- C^+(\tau_k)(3C(\tau_k) - 4C(\tau_{k-1}) + C(\tau_{k-2}))C^+(\tau_k), \end{aligned} \tag{22}$$

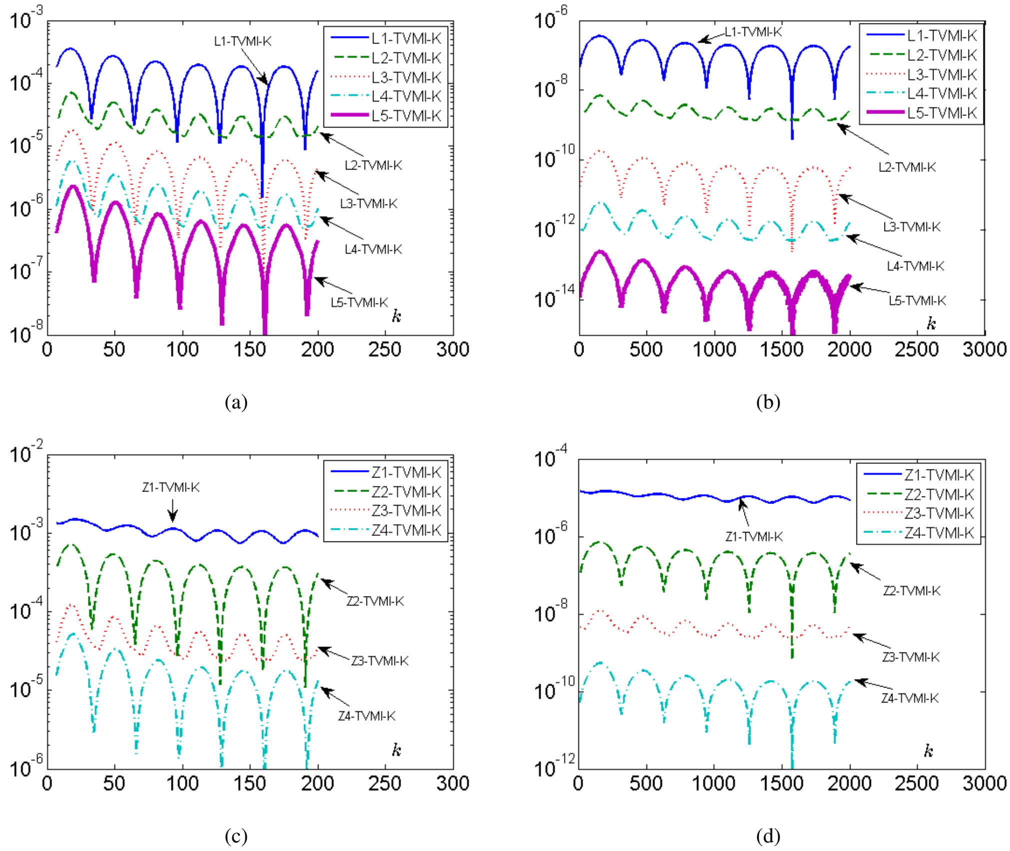


Fig. 2. Residual errors $C(\tau_{k+1})Y(\tau_{k+1})-I$ generated by different TVMI-K models with different values of ι. (a) TVMI-K models based on Lagrange finite difference with $\iota = 0.1$ s. (b) TVMI-K models based on Lagrange finite difference with $\iota = 0.01$ s. (c) TVMI-K models based on ZeaD with $\iota = 0.1$ s. (d) TVMI-K models based on ZeaD with $\iota = 0.01$ s.

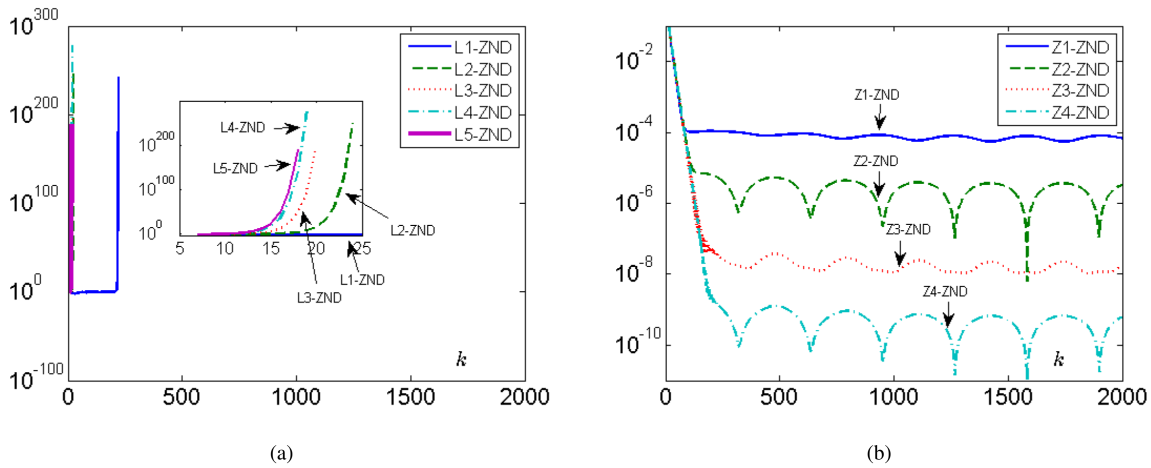


Fig. 3. Residual errors generated by different ZND models based on conventional zeroing neural dynamics method. (a) L-ZND models based on Lagrange finite difference. (b) Z-ZND models based on ZeaD.

which is termed as L1-TVMI-U model. Similarly, Combining (21) and Z2-TVMI-K model (14) yields Z2-TVMI-U model:

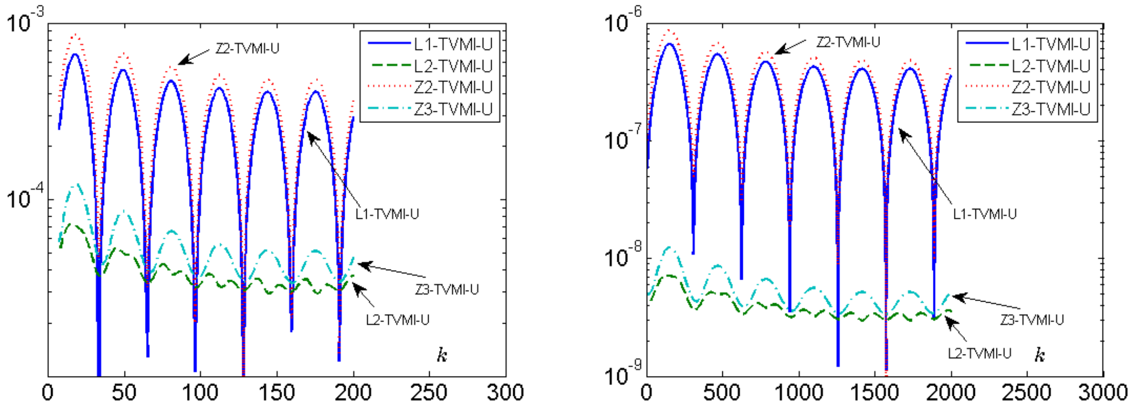
$$\begin{aligned}Y(\tau_{k+1}) =& -C^{+}(\tau_k)\Big(\frac{3}{2}C(\tau_k)-2C(\tau_{k-1})\\&+\frac{1}{2}C(\tau_{k-2})\Big)C^{+}(\tau_k)+\frac{3}{2}C^{+}(\tau_k)\\&-C^{+}(\tau_{k-1})+\frac{1}{2}C^{+}(\tau_{k-2}).\end{aligned} \tag{23}$$

For L2-TVMI-K model and Z3-TVMI-K model shown in Table II, we employ the following backward finite difference formula to approximate $\dot{C}(\tau_k)$:

$$\begin{aligned}\dot{u}(\tau_k) =& \frac{11}{6\iota}u(\tau_k)-\frac{3}{\iota}u(\tau_{k-1})+\frac{3}{2\iota}u(\tau_{k-2})\\&-\frac{1}{3\iota}u(\tau_{k-3}),\end{aligned} \tag{24}$$

It has a truncation error of $O(\iota^3)$, which also avoids the loss of precision. Pars of TVMI-U models are presented in Table III.

The following theorem is to guarantee the effectiveness and precision of proposed TVMI-U models.



(a) (b)

Fig. 4. Residual errors generated by different TVMI-U models with different values of ι. (a) with $\iota = 0.1$ s. (b) with $\iota = 0.01$ s.

Theorem 2: If matrix $C(\tau)$ in TVMI problem (1) is nonsingular and has high-order derivatives, residual error of L1-TVMI-U model (22) to solve this problem is $O(\iota^3)$.

Proof: We know that L1-TVMI-U model (22) is based on L1-TVMI-K model (10) with further approximation of $\dot{C}(\tau_k)$. Specifically, we know that backward finite difference formula (20) has truncation error of $O(\iota^2)$, and thus equation (21) is exactly

$$\dot{C}(\tau_k) = \frac{3C(\tau_k) - 4C(\tau_{k-1}) + C(\tau_{k-2})}{2\iota} + \mathbf{O}(\iota^2). \quad (25)$$

We know that $Y^*(\tau_{k+1}) = -2\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k) + C^+(\tau_{k-1}) + \mathbf{O}(\iota^3)$ from equation (16). Then, for L1-TVMI-U model (22) we have

$$\begin{aligned}
&Y(\tau_{k+1}) - Y^*(\tau_{k+1})\\
=& -2\iota C^+(\tau_k)\Big(\frac{3C(\tau_k) - 4C(\tau_{k-1}) + C(\tau_{k-2})}{2\iota}\Big)C^+(\tau_k)\\
&\quad - (-2\iota C^+(\tau_k)\dot{C}(\tau_k)C^+(\tau_k)) + \mathbf{O}(\iota^3)\\
=& -2\iota C^+(\tau_k)\Big(\frac{3C(\tau_k) - 4C(\tau_{k-1}) + C(\tau_{k-2})}{2\iota} - \dot{C}(\tau_k)\Big)\\
&\quad \cdot C^+(\tau_k) + \mathbf{O}(\iota^3)\\
=& -2C^+(\tau_k)(\mathbf{O}(\iota^3))C^+(\tau_k) + \mathbf{O}(\iota^3)\\
=& \mathbf{O}(\iota^3).
\end{aligned}$$

Then, similar to the process of (19), we have that residual error of L1-TVMI-U model (22) to solve this problem is $O(\iota^3)$. ■

D. Our Method Versus Zeroing Neural Dynamics Method

Based on previous work [10], [27], [28], using zeroing neural dynamics method to solve TVMI problem (1), we have the following process. First, we define error function as $E(\tau) = Y(\tau) - C^+(\tau)$. Second, we employ formula $\dot{E}(\tau) = -\lambda E(\tau)$ to zero out error function and continuous-time model is obtained:

$$\dot{Y}(\tau) = -\lambda(Y(\tau)C(\tau)Y(\tau) - Y(\tau)) + Y(\tau)\dot{C}(\tau)Y(\tau). \quad (26)$$

Finally, we use ZeaD formulas to discretize the continuous-time model and the final models to solve TVMI problem (1) are obtained. For example, when we use Z2 formula for discretization, we have

$$\begin{aligned}
Y(\tau_{k+1}) =& -Y(\tau_k)(h(C(\tau_k)Y(\tau_k) - I) - \iota\dot{C}(\tau_k)Y(\tau_k))\\
&+ \frac{3}{2}Y(\tau_k) - Y(\tau_{k-1}) + \frac{1}{2}Y(\tau_{k-2}),
\end{aligned} \quad (27)$$

where $h = \lambda\iota$. In this paper, we term these models based on zeroing neural dynamics method as ZND models. For example, when L1 formula is employed for discretization, the model is termed as L1-ZND.

There are some differences between our method and zeroing neural dynamics as follows. First, the models generated by zeroing neural dynamics have the form of iteration and the current step calculation is based on previous calculated results. However, our method directly utilizes known information to calculate each step results and does not use previous calculated results. Second, our method can use lagrange-type finite difference while zeroing neural dynamics method does not. It is because that our method is not be restricted by 0-stability constraint. This advantage makes our method has higher precision when we use the same number instances for discretization. Third, zeroing neural dynamics method needs a number of steps to converge to theoretical solution. However, our method directly calculates the results with higher precision.

IV. SIMULATIONS

In this section, some simulation results are shown to verify the effectiveness and superiority of our method. The following time-varying nonsingular matrix is investigated as example:

$$C(\tau_k) = \begin{bmatrix} \sin(0.1\tau_k) + 2 & \cos(\tau_k) \\ \cos(\tau_k) & \sin(0.1\tau_k) + 2 \end{bmatrix} \quad (28)$$

As described in problem formulation, current instant is τ_k, and at current instant only current and past information can be utilized for the calculation of inverse matrix. In addition, before next instant τ_{k+1} comes, the value of $Y(\tau_{k+1})$ should be obtained by calculation.

First, we take L1-TVMI-K model as example to substantiate the effectiveness of proposed models. Specifically, task duration $\tau_f = 20$ s and sampling gap is 0.01 s. Results are

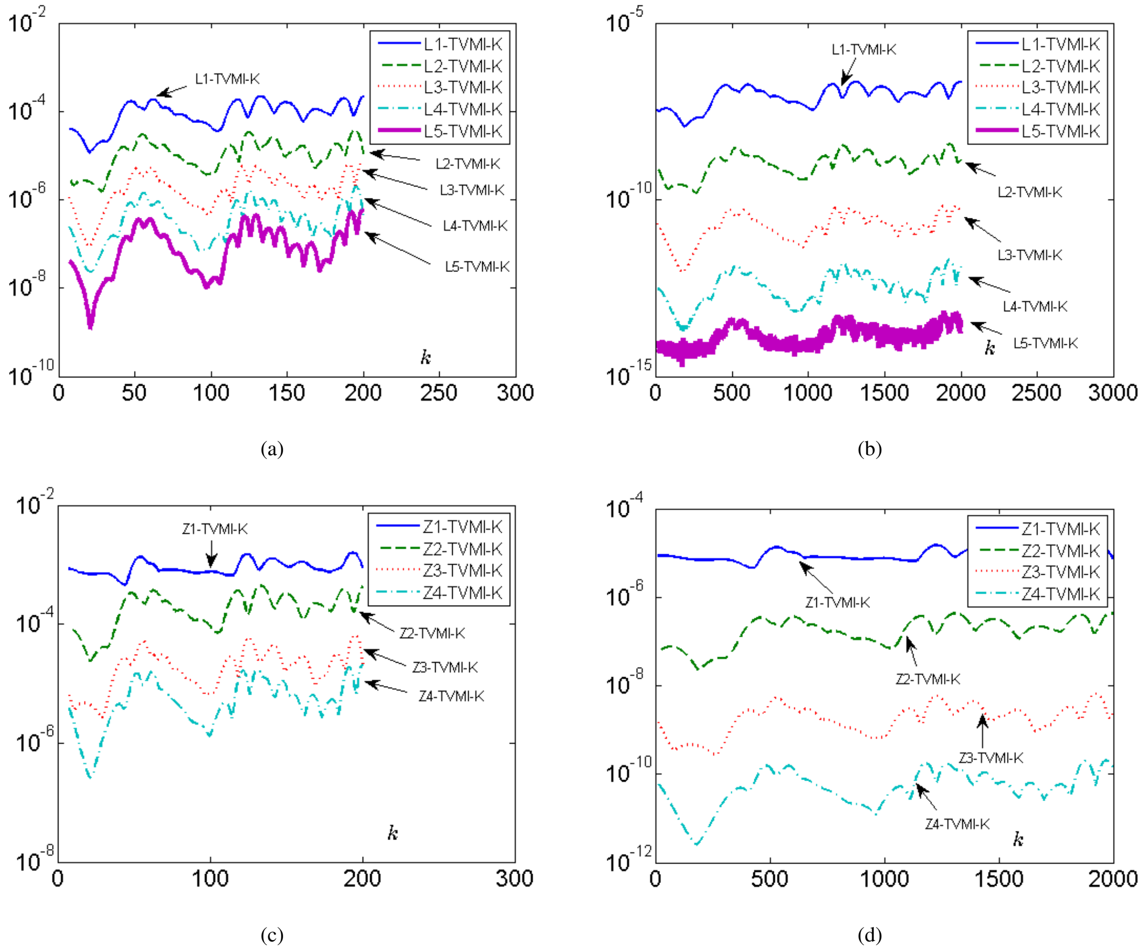


Fig. 5. Residual errors $C(\tau_{k+1})Y(\tau_{k+1}) - I$ generated by different TVMI-K models with different values of ι for solving problem (29). (a) TVMI-K models based on Lagrange finite difference with $\iota = 0.1$ s. (b) TVMI-K models based on Lagrange finite difference with $\iota = 0.01$ s. (c) TVMI-K models based on ZeaD with $\iota = 0.1$ s. (d) TVMI-K models based on ZeaD with $\iota = 0.01$ s.

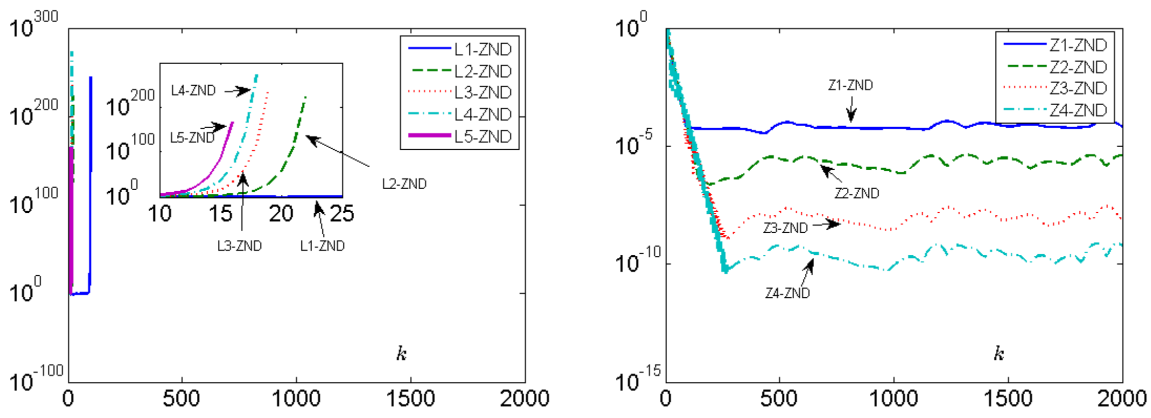
shown in Figure 1. This figure displays trajectories of four elements of solution matrix $Y(\tau_{k+1})$ generated by L1-TVMI-K model as well as theoretical solution paths, i.e., $Y^*(\tau_{k+1})$. It is observed that the trajectories generated by model overlap with theoretical ones all the time, which substantiate the effectiveness of proposed model.

Second, simulations about TVMI-K models are conducted. Corresponding models include a series of L-TVMI-K models, which are based on Lagrange-type finite difference, and a series of Z-TVMI-K models, which are based on ZeaD formulas. In addition, we take different values of sampling gap ι to substantiate the precision of different models. Results are shown in Figure 2. Specifically, Figure 2(a) shows residual errors generated by TVMI-K models based on Lagrange finite difference (L1/L2/L3/L4/L5-TVMI-K models) with $\iota = 0.1$ s, in which residual error is defined as $C(\tau_{k+1})Y(\tau_{k+1}) - I$. Figure 2(b) shows residual errors generated by TVMI-K models based on Lagrange finite difference with $\iota = 0.01$ s. Figure 2(c) shows residual errors generated by TVMI-K models based on ZeaD (Z1/Z2/Z3/Z4-TVMI-K models) with $\iota = 0.1$ s. Figure 2(d) shows residual errors generated by TVMI-K models based on ZeaD with $\iota = 0.01$ s. It can be observed that all proposed models perform well. In addition, when sampling gap is the same, discretization with more instances lead to higher precision. When sampling gap becomes 0.01 s from 0.1 s, residual errors of L1/L2/L3/L4/L5-TVMI-K models decrease by 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} and 10^{-7}, respectively. Thus, precisions of L1/L2/L3/L4/L5-TVMI-K models are $O(\iota^3)$, $O(\iota^4)$, $O(\iota^5)$, $O(\iota^6)$ and $O(\iota^7)$, respectively. Similarly, Z1/Z2/Z3/Z4-TVMI-K models have precision of $O(\iota^2)$, $O(\iota^3)$, $O(\iota^4)$ and $O(\iota^5)$, respectively.

Third, some simulations about conventional zeroing neural dynamics method are conducted to substantiate the superiority of our method. Both kinds of discretizations including lagrange-type finite difference and ZeaD formulas are utilized in zeroing neural dynamics method, the corresponding results are shown in Figure 3(a) and (b), respectively. It is observed that L1/L2/L3/L4/L5-ZND models fail to solve problem (28) and Z1/Z2/Z3/Z4-ZND models perform well. In addition, initial errors of ZND models are relatively large and tend to be stable after hundreds steps. It coincides with aforementioned discussions.

In addition, some simulations about TVMI-U models are conducted and the corresponding results are shown in Figure 4. From this figure we can observe that TVMI-U models still perform well without losing precision compared with TVMI-K models.

In addition, a relatively large time-varying nonsingular matrix $C(\tau_k) \in \mathbb{R}^{9\times 9}$ is investigated with elements are



(a) (b)

Fig. 6. Residual errors generated by different ZND models based on conventional zeroing neural dynamics method for solving problem (29). (a) L-ZND models based on Lagrange finite difference. (b) Z-ZND models based on ZeaD.

TABLE III
DIFFERENT MODELS TO SOLVE TIME-VARYING MATRIX INVERSION (1) USING DIFFERENT DISCRETIZATION FORMULAS INCLUDING LAGRANGE-TYPE FINITE DIFFERENCE AND ZEAD WITH VALUE OF $\dot{C}(\tau_k)$ UNKNOWN.

Discretization	Model	Truncation error
L1	$\tilde{C}(\tau_k) = 3C(\tau_k) - 4C(\tau_{k-1}) + C(\tau_{k-2})$ $Y(\tau_{k+1}) = -C^+(\tau_k)\tilde{C}(\tau_k)C^+(\tau_k) + C^+(\tau_{k-1})$	$O(\iota^3)$
L2	$\tilde{C}(\tau_k) = \frac{11}{2}C(\tau_k) - 9C(\tau_{k-1}) + \frac{9}{2}C(\tau_{k-2}) - C(\tau_{k-3})$ $Y(\tau_{k+1}) = -C^+(\tau_k)\tilde{C}(\tau_k)C^+(\tau_k) - \frac{3}{2}C^+(\tau_k) + 3C^+(\tau_{k-1}) - \frac{1}{2}C^+(\tau_{k-2})$	$O(\iota^4)$
Z2	$\tilde{C}(\tau_k) = \frac{3}{2}C(\tau_k) - 2C(\tau_{k-1}) + \frac{1}{2}C(\tau_{k-2})$ $Y(\tau_{k+1}) = -C^+(\tau_k)\tilde{C}(\tau_k)C^+(\tau_k) + \frac{3}{2}C^+(\tau_k) - C^+(\tau_{k-1}) + \frac{1}{2}C^+(\tau_{k-2})$	$O(\iota^3)$
Z3	$\tilde{C}(\tau_k) = \frac{33}{8}C(\tau_k) - \frac{27}{4}C(\tau_{k-1}) + \frac{27}{8}C(\tau_{k-2}) - \frac{3}{4}C(\tau_{k-3})$ $Y(\tau_{k+1}) = -C^+(\tau_k)\tilde{C}(\tau_k)C^+(\tau_k) - \frac{1}{8}C^+(\tau_k) + \frac{3}{4}C^+(\tau_{k-1}) + \frac{5}{8}C^+(\tau_{k-2}) - \frac{1}{4}C^+(\tau_{k-3})$	$O(\iota^4)$

shown as

$$a^{i,j}(\tau_k) = \begin{cases} -\sin(0.1(i-j)\tau_k)/(i-j), & \text{when } i > j \\ \sin(0.1i\tau_k) + 3, & \text{when } i = j \\ \cos(0.1(j-i)\tau_k)/(j-i), & \text{when } i < j \end{cases} \tag{29}$$

Simulation results are shown in Figures 5 and 6. Similar to the above first example, it can be observed that all proposed models have good performances. In addition, when sampling gap is the same, discretization with more instances lead to higher precision. Besides, It is observed that L1/L2/L3/L4/L5-ZND models fail to solve problem (29) and Z1/Z2/Z3/Z4-ZND models perform well. In addition, initial errors of ZND models are relatively large and tend to be stable after hundreds steps. It coincides with aforementioned discussions.

We consider an extreme condition with the object matrix is sometimes-singular. Specifically, we construct the object matrix

$$C(\tau_k) = \begin{bmatrix} \sin(0.01\pi\tau_k) & \cos(0.01\pi\tau_k) \\ \cos(0.01\pi\tau_k) & \sin(0.01\pi\tau_k) \end{bmatrix}, \tag{30}$$

which is evidently sometimes-singular. Simulation results are shown in Figures 7. It is observed that TVMI-K models based on Lagrange finite difference and ZeaD have good performances and can pass through singularity. However, ZND models based on Lagrange finite difference diverge at the beginning. ZND models based on ZeaD can not pass through singularity although converging at the beginning. Thus, our models have better performances compared with conventional models.

V. CONCLUSION

In this work, time-varying matrix inversion has been solved by a new method instead of classical zeroing neural dynamics method. A series of models based on not only ZeaD formulas but also Lagrange-type finite difference formulas have been proposed. In this method, we directly decompose the equations about theoretical solutions, and obtain a direct calculation formula instead of a form of differential equation. The proposed models are not constrained by 0-stability. Not only ZeaD formulas but also Lagrange-type finite difference formulas are effective to solve time-varying matrix inversion in our method. Finally, plenty of numerical experiments have been conducted to substantiate the effectiveness and superiority of our models.

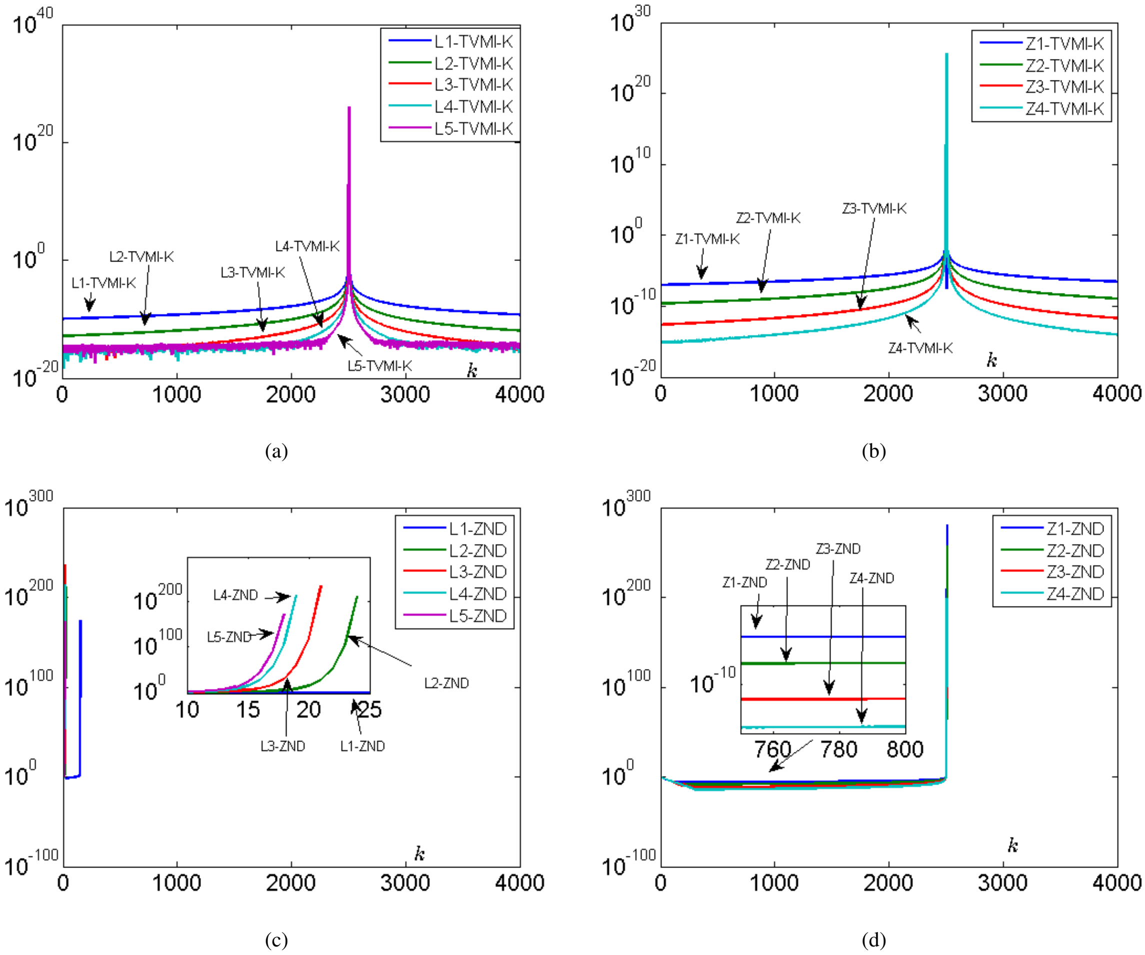


Fig. 7. Residual errors $C(\tau_{k+1})Y(\tau_{k+1}) - I$ generated by different models for solving problem (30) with $\iota = 0.01$ s. (a) TVMI-K models based on Lagrange finite difference. (b) TVMI-K models based on ZeaD. (c) ZND models based on Lagrange finite difference. (d) ZND models based on ZeaD.

REFERENCES

[1] K. Liu, Y. Liu, Y. Zhang, L. Wei, Z. Sun, and L. Jin, "Five-step discrete-time noise-tolerant zeroing neural network model for time-varying matrix inversion with application to manipulator motion generation," *Engineering Applications of Artificial Intelligence*, vol. 103, 104306, 2021.

[2] Z. Huang, L. Wang, and Z. Xu, "Some new bounds for the Hadamard product of a nonsingular M-matrix and its inverse," *IAENG International Journal of Applied Mathematics*, vol. 46, no. 3, pp. 388-397, 2016.

[3] J. Cui, G. Peng, Q. Lu, and Z. Huang, "A class of nonstationary upper and lower triangular splitting iteration methods for ill-posed inverse problems," *IAENG International Journal of Computer Science*, vol. 47, no. 1, pp. 118-129, Feb. 2020.

[4] F. Ries, T. D. Marco, and R. Guerrieri, "Triangular matrix inversion on heterogeneous multicore systems, *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 177-184, Jan. 2012.

[5] L. Ma, K. Dickson, J. McAllister, and J. McCanny, "QR decomposition-based matrix inversion for high performance embedded MIMO receivers," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1858-1867, Apr. 2011.

[6] K. Chen, "Recurrent implicit dynamics for online matrix inversion," *Applied Mathematics and Computation*, vol. 219, no. 20, pp. 10218-10224, Jun. 2013.

[7] X. Lv and T. Huang, "A note on inversion of Toeplitz matrices," *Applied Mathematics Letters*, vol. 20, no.12, pp. 1189-1193, Dec. 2007.

[8] M. D. Petkovic and P. S. Stanimirovic, "Generalized matrix inversion is not harder than matrix multiplication," *Journal of Computational and Applied Mathematics*, vol. 230, pp. 270-282, 2009.

[9] B. F. Vajargah, "Different stochastic algorithms to obtain matrix inversion," *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1841-1846, Jun. 2007.

[10] 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, Aug. 2017.

[11] Z. Zhang, L. Zheng, and M. Wang, "An exponential-enhanced-type varying-parameter RNN for solving time-varying matrix inversion," *Neurocomputing*, vol. 338, pp. 126-138, Apr. 2019.

[12] Y. Zhang and S. S. Ge, "Design and analysis of a general recurrent neural network model for time-varying matrix inversion," *IEEE Transactions on Neural Networks*, vol. 16, no. 6, pp. 1477-1490, Nov. 2005.

[13] Q. Zuo, L. Xiao, and K. Li, "Comprehensive design and analysis of time-varying delayed zeroing neural network and its application to matrix inversion," *Neurocomputing*, vol. 379, pp. 273-283, Feb. 2020.

[14] 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.

[15] 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.

[16] Y. Zhang, Z. Li, and K. Li, "Complex-valued Zhang neural network for online complex-valued time-varying matrix inversion," *Applied Mathematics and Computation*, vol. 217, no. 24, pp. 10066-10073, Aug. 2011.

[17] Y. Zhang, K. Chen, and H. Tan, "Performance analysis of gradient neural network exploited for online time-varying matrix inversion," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1940-1945, Aug. 2009.

[18] Z. Tan, Y. Hu, L. Xiao, and K. Chen, "Robustness analysis and robotic application of combined function activated RNN for time-varying matrix pseudo inversion," *IEEE Access*, vol. 7, pp. 33434-33440, 2019.

[19] L. Jin, Y. Zhang, and S. Li, "Integration-enhanced Zhang neural network for real-time-varying matrix inversion in the presence of various kinds of noises," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 12, pp. 2615-2627, Dec. 2016.

[20] L. Xiao, "A new design formula exploited for accelerating Zhang

neural network and its application to time-varying matrix inversion," *Theoretical Computer Science*, vol. 647, pp. 50-58, Sept. 2016.

[21] L. Xiao, Y. Zhang, J. Dai, Q. Zuo, and S. Wang, "Comprehensive analysis of a new varying parameter zeroing neural network for time varying matrix inversion," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1604-1613, March 2021.

[22] B. Qiu, Y. Zhang, and Z. Yang, "New discrete-time ZNN models for least-squares solution of dynamic linear equation system with time-varying rank-deficient coefficient," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5767-5776, Nov. 2018.

[23] Q. Xiang, W. Li, B. Liao, and Z. Huang, "Noise-resistant discrete-time neural dynamics for computing time-dependent Lyapunov equation," *IEEE Access*, vol. 6, pp. 45359-45371, Aug. 2018.

[24] Y. Lei, B. Liao, and Q. Yin, "A noise-acceptable ZNN for computing complex-valued time-dependent matrix pseudoinverse," *IEEE Access*, vol. 7, pp. 13832-13841, Jan. 2019.

[25] N. Zhong, Q. Huang, S. Yang, F. Ouyang, and Z. Zhang, "A varying-parameter recurrent neural network combined with penalty function for solving constrained multi-criteria optimization scheme for redundant robot manipulators," *IEEE Access*, vol. 9, pp. 50810-50818, March 2021.

[26] B. Liao and Q. Xiang, "Discrete-time noise-suppressing Zhang neural network for dynamic quadratic programming with application to manipulators," *Engineering Letters*, vol. 25, no. 4, pp. 431-437, 2017.

[27] 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, Jun. 2014.

[28] J. Li, M. Mao, F. Uhlig, and Y. Zhang. "A 5-instant finite difference formula to find discrete time-varying generalized matrix inverses, matrix inverses, and scalar reciprocals," *Numerical Algorithms*, vol. 81, pp. 609-629, 2019.

[29] M. D. Petkovi, P. S. Stanimirovi, and V. N. Katsikis, "Modified discrete iterations for computing the inverse and pseudoinverse of the time-varying matrix," *Neurocomputing*, vol. 289, pp. 155-165, May 2018.

[30] Y. Zhang, M. Yang, J. Li, L. He, S. Wu, "ZFD formula 4IgSFD_Y applied to future minimization," *Physics Letters A*, vol. 381, pp. 1677-1681, 2017.

[31] J. Li, Y. Zhang, S. Li, and M. Mao, "New discretization-formula-based zeroing dynamics for real-time tracking control of serial and parallel manipulators," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3416-3425, Aug. 2018.

[32] J. Guo, B. Qiu, J. Chen, and Y. Zhang, "Solving future different-layer nonlinear and linear equation system using new eight-node DZNN model," *IEEE Transactions on Industrial Informatics*, vol. 16, pp. 2280-2289, 2020.

[33] 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.

[34] 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, Feb. 2019.

[35] E. Suli and D. F. Mayers, *An Introduction to Numerical Analysis*. Oxford, UK: Cambridge University Press, 2003.

[36] P. Othata and N. Pochai "A mathematical model of salinity control in a river with an effect of internal waves using two explicit finite difference methods," *Engineering Letters*, vol. 29, no. 2, pp. 689-696, 2021.

[37] O. E. Abolarin and S. W. Akingbade, "Derivation and application of fourth stage inverse polynomial scheme to initial value problems,", vol. 47, no. 4, pp. 459-464, Nov. 2017.

[38] L. Wang, H. Li, and Y. Meng, "Numerical solution of coupled Burgers' equation using finite difference and sinc collocation method," *Engineering Letters*, vol. 29, no. 2, pp. 668-674, 2021.

[39] P. Othata and N. Pochai, "Irrigation water management strategies for salinity control in the chao phraya river using sualyev finite difference method with Lagrange interpolation technique, *Engineering Letters*, vol. 29, no. 2, pp. 332-338, 2021.

[40] Y. Zhang, Y. Chou, J. Chen, Z. Zhang and L. Xiao, "Presentation, error analysis and numerical experiments on a group of 1-step-ahead numerical differentiation formulas," *Journal of Computational and Applied Mathematics*, vol. 239, pp. 406-414, 2013.

[41] L. Jin and Y. Zhang, "Discrete-time Zhang neural network of $O(\iota^3)$ pattern for time-varying matrix pseudoinversion with application to manipulator motion generation," *Neurocomputing*, vol. 142, pp. 165-173, 2014.

[42] 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, Feb. 2019.

[43] Z. Tan, W. Li, L. Xiao, and Y. Hu, "New varying-parameter ZNN models with finite-time convergence and noise suppression for time-varying matrix mooreCpenrose inversion," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 8, pp. 2980-2992, Aug. 2020.

[44] P. Stanimirovi, D. Gerontitis, P. Tzekis, R. Behera, and J. K. Sahoo, "Simulation of varying parameter recurrent neural network with application to matrix inversion," *Mathematics and Computers in Simulation*, vol. 185, pp. 614-628, July 2021.