

Adams–Bashforth-Type Discrete-Time Zeroing Neural Networks Solving Time-Varying Complex Sylvester Equation With Enhanced Robustness

Zeshan Hu¹, Kenli Li¹, *Senior Member, IEEE*, Lin Xiao¹, Yaonan Wang¹, Mingxing Duan¹,
and Keqin Li², *Fellow, IEEE*

Abstract—In this article, two Adams–Bashforth-type integration-enhanced discrete-time zeroing neural dynamic (ADTIZD) models are proposed to solve the time-varying complex Sylvester equation (TVCSE) problem in the first time. In ADTIZD models, Adams–Bashforth discrete formulas as novel discrete formulas are used, giving our ADTIZD models higher accuracy [truncation error being $O(\tau^5)$] but less time and space complexity than the ordinary multi-instant models. Enhanced by the integration part, the ADTIZD models can resist large additive noises, where even constant noises cannot decrease their precision. All convergence and robustness performance conclusions about our ADTIZD models are supported by rigorous theoretical proofs and numerical experiments. More comparisons between ADTIZD models and other discrete-time zeroing neural network models are shown in these experiments too. The efficacy of ADTIZD models is finally been validated in the simulation of adopting them in controlling a robotic manipulator.

Index Terms—Adams–Bashforth discretization formula, derivative approximation, discrete-time zeroing neural dynamic (DTZND), noise resistance, robot manipulator control, time-varying complex Sylvester equation (TVCSE).

I. INTRODUCTION

SYLVESTER equation as a fundamental mathematical problem, has played an important role in various fields. These fields include signal processing [1], image recovery [2],

Manuscript received June 29, 2020; revised November 18, 2020; accepted February 22, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0204302; and in part by the Programs of National Natural Science Foundation of China under Grant 61866013, Grant 61625202, and Grant 61860206011. This article was recommended by Associate Editor S. S. S. Ge. (*Corresponding authors: Lin Xiao; Kenli Li.*)

Zeshan Hu, Kenli Li, and Mingxing Duan are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: huzeshan@hnu.edu.cn; lk1@hnu.edu.cn; duanmingxing16@nudt.edu.cn).

Lin Xiao is with the Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, Changsha 410081, China (e-mail: xiaolin860728@163.com).

Yaonan Wang is with the College of Electrical and Information Engineering, Hunan University, Changsha 410082, China (e-mail: yaonan@hnu.edu.cn).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA, and also with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: lik@newpaltz.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TSMC.2021.3065091>.

Digital Object Identifier 10.1109/TSMC.2021.3065091

optimization problem [3] and robot kinematics control [4]. The wide applications of the Sylvester equation indicate that finding and optimizing solutions of this problem are of great value. This problem has thus attracted more and more researchers' interest. Although the time-invariant Sylvester equation problem has been well studied, solutions for a more generalized time-varying Sylvester equation (TVSE) problem are still not good enough yet. When we define the TVSE problem in a complex number field, such a problem is called the time-varying complex Sylvester equation (TVCSE) problem and becomes even more general with more applications, however its solving methods are also more insufficient.

Recently, the research works of the artificial neural network (ANN) are emerging, one reason for this phenomenon is that ANN inherently possesses a parallel processing structure and, thus, it can easily utilize the powerful computing resource of a multicore processor, GPU, or even supercomputer. Since calculating the solution of the Sylvester equation problem is computing intensive, researchers have also tried to solve such a problem using ANN-based schemes. Gradient neural network (GNN), as a special type of the recurrent neural network, (RNN) is then proposed to solve matrix problems including the Sylvester equation problem in [5]. Technically speaking, GNN uses a scalar number, called the performance index, to represent the error size. In order to reduce the solution error, GNN adjusts its state output along gradient decent direction of the performance index when evolving. Experiments in [6] and [7] show that GNN has good performance in dealing with the time-invariant matrix equation problem, however it performs bad in solving the time-varying matrix equation problems with persistent large residual error. After discovering the GNN's above flaw, another RNN variation, named the zeroing neural network (ZNN), is investigated in [8]. Compared to GNN, ZNN uses an error matrix as its performance indicator, every element in this performance indicator is forced to decrease toward 0 separately during neural network evolving. In this way, ZNN has shown advantage in dealing with the time-invariant and time-varying problems such as the Sylvester equation problem [1], [9]–[12], whose residual error of the output solution always decreases to 0.

Although ZNN has been proved to be effective in solving the TVSE problem, it is worth pointing out that ZNN is limited when it comes to practical applications. Originating

from the famous Hopfield neural network [13], ZNNs are designed to be realized by analog circuit too. However, this means that the original ZNN cannot handle digital information directly. It is also difficult for the original ZNN to cooperate with existing digital equipments, since two times of conversions between digital and analog signals may introduce huge precision loss. In this day with numerous digital computing resources, a method that can solve the TVSE problem in the digital circuit or computer may be more suitable. To overcome the above disadvantages, our first possible choice can be many existing numerical algorithms for the static Sylvester equation problem like in [14]–[16], which mostly belong to iterative algorithms. Then, with the above numerical algorithms, the TVSE problem will be treated discretely in time and solutions need to be calculated at every time instant. But, this discretization operation has two fatal problems. One problem is that with high computational complexity of these numerical algorithms, the larger scale the Sylvester equation is, the longer computation time each time instant will need. Another problem is that these algorithms' solutions will always be behind the theoretical solutions in time, such time delay can result in a large lagging error.

In light of that continuous-time ZNN (CTZNNs) models can solve various time-varying matrix problems quickly and accurately, researchers begin to design discrete-time ZNN models that can both maintain CTZNNs' solving capability and run in the digital circuit [17]–[21]. For these discrete-time ZNN models, although they also treat time-varying problems as discrete-time problems, they will calculate solutions of each time point before rather than behind that time point. Therefore, discrete-time ZNN models are able to avoid the lagging error caused by solution time delay. Moreover, since discrete-time ZNN models are actually computing future solutions of time-varying problems, such problems are termed the future problems.

Nevertheless, the investigation of discrete-time ZNN models is not abundant. As far as we have known, there is no published work that uses the discrete-time ZNN model to solve the TVCSE problem yet. Moreover, the existing literature about discrete-time ZNN models [19]–[22] typically only use discrete formulas that are derived from the approximation formula of the first-order derivative. Then, in these literature, higher precision of the discrete-time ZNN model is achieved by designing the discrete formula that involves more time steps, which means more storage and computing time consumption. However, building discrete-time ZNN model should not be confined to this way. It is also true that there exists better discrete formula than the previous one that is used in [19]–[22]. Therefore, in this article, we introduce the Adams–Bashforth formulas as discrete formulas. Then, a novel integration-enhanced continuous ZNN model for solving the TVCSE problem will be discretized by such a discrete formula. Such novel discrete-time ZNN models we propose for the complex number TVSE problem will be named as Adams–Bashforth-type discrete-time integration-enhanced zeroing neural dynamic (ADTIZD) models, which provides better accuracy but with less space and time complexity.

The remainder of this article is outlined as follows. Section II first introduces the formulation of the continuous

TVCSE (CTVCSE) problem and its corresponding discrete-time version: the discrete-time-varying complex Sylvester equation (DTVCSE) problem. Section II then clarifies the design of the continuous-time integration-enhanced ZNN model for solving the CTVCSE problem. In Section III, the Adams–Bashforth discrete formula is introduced, then two different ADTIZD models named ADTIZD-K and Adams–Bashforth-type DTIZD with unknown derivatives (ADTIZD-U) model are proposed for solving the DTVCSE problem with or without derivative information. Section III also theoretically proves these two ADTIZD models' maximum steady-state residual error (MSSRE) convergence order and their inherent noise resistance against different additive noises. Section IV presents three existing conventional multi-instant discrete-time integration-enhanced zeroing neural dynamic (DTIZD) models to be compared with our ADTIZD models. Section V provides numerical validations for theoretical conclusions about our ADTIZD models and comparisons across different models. Section VI applies our ADTIZD-U model in inverse kinematics controlling a robot arm, verifying our ADTIZD models' efficacy again. Section VII concludes this article briefly. Finally, contributions of this article are summarized as follows.

- 1) In this article, two discrete-time ZNN models termed the ADTIZD models for solving the TVCSE problem are proposed for the first time.
- 2) The novel Adams–Bashforth formula is utilized to design discrete-time ZNN models, giving the ADTIZD models a higher solution accuracy with even less calculations and storage consumption than the traditional models. The ADTIZD models possess powerful noise suppression abilities in the same time.
- 3) Theoretical proofs are presented to support ADTIZD models' performance. Numerical experiments are conducted to validate these conclusions and they include comprehensive comparisons between different models.
- 4) The ADTIZD-U model is successfully used in robot arm control application and the results have shown ADTIZD models' superiority in practical usage.

II. PROBLEM FORMULATION AND CTIZNN DESIGN

The problem that we aim here is the TVCSE and we first introduce the CTVCSE problem. Consider the following equation:

$$\mathbf{A}(t)\mathbf{X}(t) + \mathbf{X}(t)\mathbf{B}(t) = \mathbf{C}(t) \quad (1)$$

where $\mathbf{A}(t) \in \mathbb{C}^{p \times p}$, $\mathbf{B}(t) \in \mathbb{C}^{q \times q}$, and $\mathbf{C}(t) \in \mathbb{C}^{p \times q}$ are known time-dependent complex matrices and $\mathbf{X}(t) \in \mathbb{C}^{p \times q}$ is an unknown complex matrix. Equation (1) is the continuous TVSE and we need to obtain $\mathbf{X}(t)$ at any time instant in real time in the CTVCSE problem. In order to facilitate the investigation of the CTVCSE problem, according to [23], we assume that $\mathbf{A}(t)$, $\mathbf{B}(t)$, and $\mathbf{C}(t)$ are all smooth matrices with their derivatives about time and themselves being uniformly bounded. The assumption that $\mathbf{A}(t)$ and $-\mathbf{B}(t)$ share no identical eigenvalues is also provided to ensure that the CTVCSE problem (1) has a unique solution.

A. Problem Formulation

On the basis of CTVCSE (1), we now present the DTVCSE problem that we are going to tackle

$$\mathbf{A}_{n+1}\mathbf{X}_{n+1} + \mathbf{X}_{n+1}\mathbf{B}_{n+1} = \mathbf{C}_{n+1} \quad (2)$$

where $\mathbf{A}_{n+1} = \mathbf{A}(t_{n+1})$, $\mathbf{B}_{n+1} = \mathbf{B}(t_{n+1})$, and $\mathbf{C}_{n+1} = \mathbf{C}(t_{n+1})$ are acquired from $\mathbf{A}(t)$, $\mathbf{B}(t)$, and $\mathbf{C}(t)$ in (1) by sampling at time point $t_{n+1} = (n+1)\tau$ with $n \in \mathbb{N}$ stands for the sampling index and $\tau \in \mathbb{R}^+$ denotes the sampling gap. In the DTVCSE problem, due to the real-time solution requirement, the unknown matrix \mathbf{X}_{n+1} has to be figured out between the updating interval $[n\tau, (n+1)\tau] \in [t_0, t_f]$, where $t_f \in \mathbb{R}^+$ is the end time of the DTVCSE problem. However, the future information of \mathbf{A}_{n+1} , \mathbf{B}_{n+1} , and \mathbf{C}_{n+1} for solving \mathbf{X}_{n+1} within $t \in [n\tau, (n+1)\tau]$ are practically unreachable, thus such a kind of problem like the DTVCSE problem (2) can be termed the future problem and is extremely hard to solve.

B. Design of CTIZNN

To lay the foundation for the DTIZD models for solving the DTVCSE problem, the continuous-time integration-enhanced ZNN (CTIZNN) model needs to be designed first.

The complex Sylvester (1) is reformulated as

$$(I_q \otimes \mathbf{A}(t) + \mathbf{B}^T(t) \otimes I_p)\mathbf{x}(t) = \mathbf{c}(t) \quad (3)$$

where \otimes stands for the Kronecker product, whose detail properties can be found in [24], $\mathbf{x}(t) = \text{vec}(\mathbf{X}(t)) \in \mathbb{C}^{pq}$ and $\mathbf{c}(t) = \text{vec}(\mathbf{C}(t)) \in \mathbb{C}^{pq}$ are the column vectors obtained by piling columns from $\mathbf{X}(t)$ and $\mathbf{C}(t)$ into a single column. Besides, $I_p \in \mathbb{R}^{p \times p}$ and $I_q \in \mathbb{R}^{q \times q}$ are the identity matrices. The vector form of (3) is then further simplified to

$$\mathbf{M}(t)\mathbf{x}(t) = \mathbf{c}(t) \quad (4)$$

where $\mathbf{M}(t) = I_q \otimes \mathbf{A}(t) + \mathbf{B}^T(t) \otimes I_p \in \mathbb{C}^{pq \times pq}$. It is worth noting that the existing design and analyzing methods of discrete-time zeroing neural dynamic (DTZND) are mostly based on the real number field, which promotes us to extend these methods to the DTVCSE problem that has complex matrices. Therefore, we decompose the complex coefficients in (4) into $\mathbf{M} = \mathbf{M}_r + i\mathbf{M}_{ig}$, $\mathbf{x} = \mathbf{x}_r + i\mathbf{x}_{ig}$, $\mathbf{c} = \mathbf{c}_r + i\mathbf{c}_{ig}$, in which i denotes the imaginary unit, $\mathbf{M}_r \in \mathbb{R}^{pq \times pq}$, $\mathbf{x}_r, \mathbf{c}_r \in \mathbb{R}^{pq}$ and $\mathbf{M}_{ig} \in \mathbb{R}^{pq \times pq}$, $\mathbf{x}_{ig}, \mathbf{c}_{ig} \in \mathbb{R}^{pq}$ are all real matrices. It follows from (4) that:

$$\begin{aligned} (\mathbf{M}_r + i\mathbf{M}_{ig})(\mathbf{x}_r + i\mathbf{x}_{ig}) &= \mathbf{c}_r + i\mathbf{c}_{ig} \\ (\mathbf{M}_r\mathbf{x}_r - \mathbf{M}_{ig}\mathbf{x}_{ig}) + i(\mathbf{M}_r\mathbf{x}_{ig} + \mathbf{M}_{ig}\mathbf{x}_r) &= \mathbf{c}_r + i\mathbf{c}_{ig}. \end{aligned} \quad (5)$$

Solving the above equations is equivalent to handling the following linear equations:

$$\begin{aligned} \mathbf{N}(t)\mathbf{y}(t) &= \mathbf{d}(t) \\ \mathbf{N} &= \begin{bmatrix} \mathbf{M}_r & -\mathbf{M}_{ig} \\ \mathbf{M}_{ig} & \mathbf{M}_r \end{bmatrix}, \mathbf{y} = \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_{ig} \end{bmatrix}, \mathbf{d} = \begin{bmatrix} \mathbf{c}_r \\ \mathbf{c}_{ig} \end{bmatrix}. \end{aligned} \quad (6)$$

Now, we have successfully transformed the CTVCSE problem into real numbered linear (6) and the solution $\mathbf{X}(t)$

can be easily recovered from $\mathbf{y}(t)$. Let us consider the CTIZNN formula [25]–[27]

$$\dot{\mathbf{e}}(t) = -\gamma\mathbf{e}(t) - \xi \int_0^t \mathbf{e}(\sigma)d\sigma \quad (7)$$

where $\gamma > 0$ and $\xi > 0$ are the design parameters used to adjust the model's convergence rate. Constructing the error index $\mathbf{e}(t) = \mathbf{N}(t)\mathbf{y}(t) - \mathbf{d}(t)$ for (6) and substituting it into ZNN formula (7), the CTIZNN model for the CTVCSE problem is obtained

$$\begin{aligned} \mathbf{N}(t)\dot{\mathbf{y}}(t) &= \dot{\mathbf{d}}(t) - \dot{\mathbf{N}}(t)\mathbf{y}(t) - \gamma(\mathbf{N}(t)\mathbf{y}(t) - \mathbf{d}(t)) \\ &\quad - \xi \int_0^t (\mathbf{N}(\sigma)\mathbf{y}(\sigma) - \mathbf{d}(\sigma))d\sigma. \end{aligned} \quad (8)$$

According to [25]–[27], we have the following lemma about the stability of CTIZNN design formula (7).

Lemma 1: Error vector $\mathbf{e}(t)$ of the CTIZNN formula (7) globally and exponentially converges to zero vector $\mathbf{0}$.

Because the unique solution of the CTVCSE problem is obtained when and only when (6) is solved, we conclude from Lemma 1 that CTIZNN model (8) globally converges to the precise solution of (6) as well as the CTVCSE problem.

Finally, for convenience of ensuing investigation, the CTIZNN model (8) is reconstructed as in [28]

$$\dot{\mathbf{y}}(t) = \mathbf{N}^\dagger(t)(\dot{\mathbf{d}}(t) - \dot{\mathbf{N}}(t)\mathbf{y}(t) - \gamma(\mathbf{N}(t)\mathbf{y}(t) - \mathbf{d}(t) - \xi\mathbf{v}(t))) \quad (9)$$

where $\mathbf{N}^\dagger(t) \in \mathbb{R}^{2pq \times 2pq}$ represents the Moore–Penrose inverse of $\mathbf{N}(t)$, while $\gamma \in \mathbb{R}^+$ is a predefined parameter used to scale CTIZNN's convergence rate and $\mathbf{v}(t) = \int_0^t (\mathbf{N}(\sigma)\mathbf{y}(\sigma) - \mathbf{d}(\sigma))d\sigma \in \mathbb{R}^{2pq}$ is the integral term. Thus, the CTIZNN model (9) can be applied to solve (6) and retrieve the solution matrix $\mathbf{X}(t)$ of the CTVCSE problem from $\mathbf{y}(t)$. To lay the foundations for the following model design and analyses, we have the following definitions.

Definition 1: A linear k -step method is consistent (i.e., have consistency) of order m if its truncation error for smooth exact solution is $O(\tau^{m+1})$ [29].

Definition 2: By obtaining the roots of characteristic polynomial $P_k(\theta) = \sum_{j=0}^k \lambda_j \theta^j$, a k -step method $\sum_{j=0}^k \lambda_j y_{n+j} = \tau \sum_{j=0}^k \varphi_j \phi_{n+j}$ is able to be checked if it is 0-stable. Using θ to represent all roots of $P_k(\theta)$, the above k -step method is 0-stable (i.e., has 0-stability) if $|\theta| \leq 1$ and $|\theta| = 1$ is simple [29].

Definition 3: If and only if the k -step method is consistent and 0-stable, it is convergent. When a k -step method is convergent, it satisfies: $y_{[t/\tau]} \rightarrow y^*(t)$ for all $t \in [0, t_f]$ with $\tau \rightarrow 0$. Moreover, a convergent k -step method converges with the order of its truncation error [29].

III. ADAMS-TYPE DTIZD MODEL

In this section, we introduce the effective Adams–Bashforth formula to the discrete-time ZNN field and propose two novel Adams-type DTIZD (ADTIZD) models, which possess higher precision while still maintaining a simple formula and low calculation steps as well as low storage consumption. In this article, $S^v[t_0, t_f]$ denotes the set of functions ψ , where first v -order derivatives of all ψ are continuous within interval $[t_0, t_f]$. Besides, we assume that $\mathbf{y}(t) \in S^5[t_0, t_f]$ in (6).

The general discrete k -step method for solving the ordinary differential equation (ODE) such as CTIZNN dynamic formula (9) is formulated as in [29] and [30]

$$y_{n+k} = \sum_{j=0}^{k-1} \alpha_j y_{n+j} + \tau \sum_{j=0}^k \beta_j \dot{y}_{n+j} \quad (10)$$

where $y(t)$ is an element of state solution $\mathbf{y}(t)$, and y_{n+k} is the approximate value of $y(t_{n+k})$ calculated using the right-hand side of (10). In addition, $y_{n+j} = y((n+j)\tau)$, $\dot{y}_{n+j} = \dot{y}((n+j)\tau)$, τ is the time gap in Section II-A, and $\alpha_j, \beta_j \in \mathbb{R}$ are constant values that satisfy $\alpha_0 \beta_0 \neq 0$. Since we are solving future problem (6), $\beta_k = 0$ is always true for utilizing no future information.

For the purpose of constructing the k -step method (10), we define the truncation error T_{n+k} of (10) as

$$T_{n+k} = y_{n+k}^* - \sum_{j=0}^{k-1} \alpha_j y_{n+j} - \tau \sum_{j=0}^k \beta_j \dot{y}_{n+j} \quad (11)$$

where y_{n+k}^* is the accurate of $y(t_{n+k})$ in the CTIZNN model (8). In light of the Taylor expansion, we have

$$\begin{aligned} y(t_{n+j}) &= y(t_n) + \dot{y}(t_n)j\tau + \frac{\ddot{y}(t_n)}{2!}(j\tau)^2 + \dots \\ \dot{y}(t_{n+j}) &= \dot{y}(t_n) + \ddot{y}(t_n)j\tau + \frac{y^{(3)}(t_n)}{2!}(j\tau)^2 + \dots \end{aligned} \quad (12)$$

where $j = 0, 1, 2, \dots, k$. Substituting (12) into (11) leads to

$$\begin{aligned} T_{n+k} &= l_0 y(t_n) + l_1 \tau \dot{y}(t_n) + \dots + l_m \tau^m y^{(m)}(t_n) + \dots \quad (13) \\ \begin{cases} l_0 = 1 - (\alpha_0 + \dots + \alpha_{k-1}) \\ l_1 = k - (\alpha_1 + \dots + (k-1)\alpha_{k-1}) - (\beta_0 + \dots + \beta_k) \\ l_m = \frac{1}{m!}(k^m - (\alpha_1 + \dots + (k-1)^m \alpha_{k-1})) - \frac{1}{(m-1)!} \\ \quad (\beta_1 + 2^{m-1}\beta_2 + \dots + k^{m-1}\beta_k), \quad m = 2, 3, \dots \end{cases} \end{aligned}$$

Note that if the magnitude of truncation error T_{n+k} is $O(\tau^{p+1})$, we say that (10) is a p -order discrete formula.

Theorem 1: The Adams–Bashforth-type discrete formulas for solving (9) as an ODE problem with different orders of truncation error are, respectively, listed as follows [30], [31]:

$$y_{n+1} \approx y_n + \tau \dot{y}_n \quad (14)$$

$$y_{n+2} \approx y_{n+1} + \frac{\tau}{2}(3\dot{y}_{n+1} - \dot{y}_n) \quad (15)$$

$$y_{n+3} \approx y_{n+2} + \frac{\tau}{12}(23\dot{y}_{n+2} - 16\dot{y}_{n+1} + 5\dot{y}_n) \quad (16)$$

$$y_{n+4} \approx y_{n+3} + \frac{\tau}{24}(55\dot{y}_{n+3} - 59\dot{y}_{n+2} + 37\dot{y}_{n+1} - 9\dot{y}_n) \quad (17)$$

where the truncation error T_{n+k} of (14)–(17) are $O(\tau^2)$, $O(\tau^3)$, $O(\tau^4)$, and $O(\tau^5)$, respectively. According to Definition 1, we then term (14)–(17) as the 1, 2, 3, and 4-order Adams–Bashforth discrete formulas, respectively.

Proof: We take the proof of discrete formula (17) for example. When the step number is $k = 4$, let $\alpha_3 = 1, \alpha_0 = \alpha_1 = \alpha_2 = 0, l_0 = l_1 = l_2 = l_3 = l_4 = 0$, as well as $\beta_4 = 0$ in (11), and from (13), we have the following linear equations:

$$\begin{cases} \beta_0 + \beta_1 + \beta_2 + \beta_3 + \beta_4 = 1 \\ 2(\beta_1 + 2\beta_2 + 3\beta_3 + 4\beta_4) = 4^2 - 3^2 \\ 3(\beta_1 + 2^2\beta_2 + 3^2\beta_3 + 4^2\beta_4) = 4^3 - 3^3 \\ 4(\beta_1 + 2^3\beta_2 + 3^3\beta_3 + 4^3\beta_4) = 4^4 - 3^4. \end{cases}$$

Solving the above equations, we have the solution as

$$\beta_0 = -\frac{9}{24}, \beta_1 = \frac{37}{24}, \beta_2 = -\frac{59}{24}, \beta_3 = \frac{55}{24}.$$

Therefore, with known α_j and β_j , discrete formula (17) can be derived. Besides, we have $l_5 = 251/720 \neq 0$ for formula (17); thus, (17) has the truncation error of $T_{n+4} = O(\tau^5)$ consider that $y^{(5)}(t_n)$ is bounded in (13). As for other Adams–Bashforth-type discrete formulas (14)–(16), the proof can be obtained similar to the case of $k = 4$ and is thus omitted here. ■

We exploit the Adams–Bashforth-type discrete (17) to discretize the CTIZNN model (9)

$$\begin{aligned} \mathbf{y}_{n+1} &= \mathbf{y}_n + \frac{\tau}{24}(55\dot{\mathbf{y}}_n - 59\dot{\mathbf{y}}_{n-1} + 37\dot{\mathbf{y}}_{n-2} - 9\dot{\mathbf{y}}_{n-3}) \\ \dot{\mathbf{y}}_{n+j} &= \mathbf{N}_{n+j}^\dagger(\dot{\mathbf{d}}_{n+j} - \dot{\mathbf{N}}_{n+j}\mathbf{y}_{n+j} - \gamma(\mathbf{N}_{n+j}\mathbf{y}_{n+j} - \mathbf{d}_{n+j}) - \xi\mathbf{v}_{n+j}) \end{aligned} \quad (18)$$

where \mathbf{N}_{n+j} and \mathbf{d}_{n+j} are obtained by sampling the corresponding matrices at time point t_{n+j} similar to obtaining \mathbf{A}_{n+1} in (2). Different from \mathbf{N}_{n+j} and \mathbf{d}_{n+j} , $\mathbf{v}_{n+j} = \mathbf{v}(t_{n+j})$ cannot be directly computed. In order to obtain \mathbf{v}_{n+j} , we take $\mathbf{v}(t)$ as the solution of another ODE problem $\dot{\mathbf{v}}(t) = \mathbf{e}(t)$ and calculate its approximate value with (17) as

$$\mathbf{v}_{n+4} = \mathbf{v}_{n+3} + \frac{\tau}{24}(55\mathbf{e}_{n+3} - 59\mathbf{e}_{n+2} + 37\mathbf{e}_{n+1} - 9\mathbf{e}_n) \quad (19)$$

where $\mathbf{e}_{n+j} = \mathbf{N}_{n+j}\mathbf{x}_{n+j} - \mathbf{d}_{n+j}$. Here, in model (18), we assume that the time derivatives $\dot{\mathbf{N}}_{n+j}$ and $\dot{\mathbf{d}}_{n+j}$ are known and directly acquired. Thus, we name (18) as the Adams-type DTIZD with the known derivative (ADTIZD-K) model.

Theorem 2: The ADTIZD-K model (18) is consistent and convergent. Furthermore, the truncation error with which (18) converges is $O(\tau^5)$ for $t_n \in [t_0, t_f]$.

Proof: First, we know from Theorem 1 that the Adams–Bashforth formula (17) is consistent of order 4. Then, we come to investigate the residual error of ADTIZD-K model (18). We let \mathbf{v}_{n+j}^* and $\dot{\mathbf{y}}_{n+j}^*$ be the accurate values of $\mathbf{v}(t_{n+j})$ and $\dot{\mathbf{y}}(t_{n+j})$, respectively, then (18) leads to

$$\begin{aligned} \dot{\mathbf{y}}_{n+j} &= \mathbf{N}_{n+j}^\dagger(\dot{\mathbf{d}}_{n+j} - \dot{\mathbf{N}}_{n+j}\mathbf{y}_{n+j} - \gamma(\mathbf{N}_{n+j}\mathbf{y}_{n+j} - \mathbf{d}_{n+j}) \\ &\quad - \xi\mathbf{v}_{n+j}) \\ &= \mathbf{N}_{n+j}^\dagger(\dot{\mathbf{d}}_{n+j} - \dot{\mathbf{N}}_{n+j}\mathbf{y}_{n+j} - \gamma(\mathbf{N}_{n+j}\mathbf{y}_{n+j} - \mathbf{d}_{n+j}) \\ &\quad - \xi(\mathbf{v}_{n+j}^* + \mathbf{O}(\tau^5))) \\ &= \mathbf{N}_{n+j}^\dagger(\dot{\mathbf{d}}_{n+j} - \dot{\mathbf{N}}_{n+j}\mathbf{y}_{n+j} - \gamma(\mathbf{N}_{n+j}\mathbf{y}_{n+j} - \mathbf{d}_{n+j}) \\ &\quad - \xi\mathbf{v}_{n+j}^*) + \mathbf{O}(\tau^5) \\ &= \dot{\mathbf{y}}_{n+j}^* + \mathbf{O}(\tau^5) \end{aligned} \quad (20)$$

where $\dot{\mathbf{y}}_{n+j}^*$ is the precise value of $\dot{\mathbf{y}}(t_{n+j})$ in the CTIZNN model (8). Furthermore, from (20) and (18), considering (17)'s truncation error $O(\tau^5)$, we have

$$\begin{aligned} \mathbf{y}_{n+1} &= \mathbf{y}_n + \frac{\tau}{24}(55\dot{\mathbf{y}}_n^* - 59\dot{\mathbf{y}}_{n-1}^* + 37\dot{\mathbf{y}}_{n-2}^* - 9\dot{\mathbf{y}}_{n-3}^* + \mathbf{O}(\tau^5)) \\ &= (\mathbf{y}_{n+1}^* + \mathbf{O}(\tau^5)) + \mathbf{O}(\tau^6) \\ &= \mathbf{y}_{n+1}^* + \mathbf{O}(\tau^5) \end{aligned} \quad (21)$$

where \mathbf{y}_{n+j}^* is the precise value of $\mathbf{y}(t_{n+j})$ in the CTIZNN model (8). Hence, the ADTIZD-K model (18) is consistent with $\mathbf{O}(\tau^5)$ according to Definition 1. Second, the characteristic polynomial of (18) is $\theta^4 - \theta^3 = 0$, which has two roots $\theta_1 = 0$ and $\theta_2 = 1$ and root θ_2 is a simple root. Therefore, it yields from Definition 2 that the ADTIZD-K model is also 0-stable. Finally, we have the conclusion ADTIZD-K model being convergent with the truncation error being $\mathbf{O}(\tau^5)$ from Definition 3. ■

As mentioned above, we assume that the time derivatives of input matrices to be known in ADTIZD-K model (18). Nevertheless, when dealing with the DTVCSE problem (2) in real world, the derivative information of \mathbf{A}_n , \mathbf{B}_n , and \mathbf{C}_n are commonly unknown, and $\dot{\mathbf{N}}_n$ and $\dot{\mathbf{d}}_n$ are consequently unknown too. Under this situation, we apply the backward discrete formulas in [30] to approximate $\dot{\mathbf{N}}_n$ and $\dot{\mathbf{d}}_n$ in (18), which are

$$\dot{y}_n \approx \frac{1}{\tau}(y_n - y_{n-1}) \quad (22)$$

$$\dot{y}_n \approx \frac{1}{2\tau}(3y_n - 4y_{n-1} + y_{n-2}) \quad (23)$$

$$\dot{y}_n \approx \frac{1}{6\tau}(11y_n - 18y_{n-1} + 9y_{n-2} - 2y_{n-3}) \quad (24)$$

$$\dot{y}_n \approx \frac{1}{12\tau}(25y_n - 48y_{n-1} + 36y_{n-2} - 16y_{n-3} + 3y_{n-4}) \quad (25)$$

where the truncation error of (22)–(25) are $O(\tau)$, $O(\tau^2)$, $O(\tau^3)$, and $O(\tau^4)$, respectively. In order to match with the precision of ADTIZD-K model (18), approximation formula (25) is selected to approximate first-order derivatives. Thus, the following ADTIZD-U model is deduced:

$$\begin{aligned} \mathbf{y}_{n+1} &= \mathbf{y}_n + \frac{\tau}{24}(55\tilde{\mathbf{y}}_n - 59\tilde{\mathbf{y}}_{n-1} + 37\tilde{\mathbf{y}}_{n-2} - 9\tilde{\mathbf{y}}_{n-3}) \\ \tilde{\mathbf{y}}_{n+j} &= \mathbf{N}_{n+j}^\dagger (\dot{\mathbf{D}}_{n+j} - \dot{\mathcal{N}}_{n+j}\mathbf{y}_{n+j} - \gamma(\mathbf{N}_{n+j}\mathbf{y}_{n+j} - \mathbf{d}_{n+j}) \\ &\quad - \xi\mathbf{v}_{n+j}) \end{aligned} \quad (26)$$

where $\dot{\mathbf{D}}_n = (25\mathbf{d}_n - 48\mathbf{d}_{n-1} + 36\mathbf{d}_{n-2} - 16\mathbf{d}_{n-3} + 3\mathbf{d}_{n-4})/(12\tau)$ and $\dot{\mathcal{N}}_n = (25\mathbf{N}_n - 48\mathbf{N}_{n-1} + 36\mathbf{N}_{n-2} - 16\mathbf{N}_{n-3} + 3\mathbf{N}_{n-4})/(12\tau)$. Besides, $\dot{\mathbf{v}}_{n+j}$ is defined and obtained as in ADTIZD-K model (18).

Theorem 3: The ADTIZD-U model (26) is consistent and convergent, which converges with the truncation error being $O(\tau^5)$ for $t_n \in [t_0, t_f]$.

Proof: On the basis of the ADTIZD-U model (26) and the fact that derivative approximation formula (25) has the residual error with a magnitude of $O(\tau^4)$, in (26), we obtain

$$\begin{aligned} \tilde{\mathbf{y}}_{n+j} &= \mathbf{N}_{n+j}^\dagger \left((\dot{\mathbf{d}}_{n+j} + \mathbf{O}(\tau^4)) - (\dot{\mathbf{N}}_{n+j} + \mathbf{O}(\tau^4))\mathbf{y}_{n+j} \right. \\ &\quad \left. - \gamma(\mathbf{N}_{n+j}\mathbf{y}_{n+j} - \mathbf{d}_{n+j}) - \xi(\mathbf{v}_{n+j}^* + \mathbf{O}(\tau^5)) \right) \\ &= \mathbf{N}_{n+j}^\dagger \left(\dot{\mathbf{d}}_{n+j} - \dot{\mathbf{N}}_{n+j}\mathbf{y}_{n+j} - \gamma(\mathbf{N}_{n+j}\mathbf{y}_{n+j} - \mathbf{d}_{n+j}) \right. \\ &\quad \left. - \xi\mathbf{v}_{n+j}^* + \mathbf{O}(\tau^4) \right) \\ &= \mathbf{N}_{n+j}^\dagger \left(\dot{\mathbf{d}}_{n+j} - \dot{\mathbf{N}}_{n+j}\mathbf{y}_{n+j} - \gamma(\mathbf{N}_{n+j}\mathbf{y}_{n+j} - \mathbf{d}_{n+j}) \right. \\ &\quad \left. - \xi\mathbf{v}_{n+j}^* \right) + \mathbf{O}(\tau^4) \\ &= \dot{\mathbf{y}}_{n+j}^* + \mathbf{O}(\tau^4) \end{aligned}$$

where $\dot{\mathbf{y}}_{n+j}^*$ is the accurate value of $\dot{\mathbf{y}}(t_{n+j})$ as in the ADTIZD-K model (18). Substituting the above equation back to model (26), we have

$$\begin{aligned} \mathbf{y}_{n+1} &= \mathbf{y}_n + \frac{\tau}{24} \left(55\dot{\mathbf{y}}_n^* - 59\dot{\mathbf{y}}_{n-1}^* + 37\dot{\mathbf{y}}_{n-2}^* - 9\dot{\mathbf{y}}_{n-3}^* \right. \\ &\quad \left. + \mathbf{O}(\tau^4) \right) \\ &= \mathbf{y}_n + \frac{\tau}{24} (55\dot{\mathbf{y}}_n^* - 59\dot{\mathbf{y}}_{n-1}^* + 37\dot{\mathbf{y}}_{n-2}^* - 9\dot{\mathbf{y}}_{n-3}^*) \\ &\quad + \mathbf{O}(\tau^5) \end{aligned}$$

which means that ADTIZD-U model (26) is the approximation of ADTIZD-K (18) model with error being $\mathbf{O}(\tau^5)$. Thus, for the ODE (9), ADTIZD-U's truncation error is $\mathbf{O}(\tau^5)$ too. Using the same technique, proving Theorem 2, ADTIZD-U model (26) is consistent and convergent with residual error being $\mathbf{O}(\tau^5)$. ■

Theorem 4: When utilizing the ADTIZD-K (18) model and ADTIZD-U (26) model to solve the future DTVCSE problem (2), the MSSREs $\lim_{n \rightarrow \infty} \sup \|\mathbf{A}_{n+1}\mathbf{X}_{n+1} + \mathbf{X}_{n+1}\mathbf{B}_{n+1} - \mathbf{C}_{n+1}\|_F$ of such two models both are $O(\tau^5)$.

Proof: First, we consider the ADTIZD-K (18) model. According to Lemma 1 and Theorem 2, we can obtain $\mathbf{y}_{n+1} = \mathbf{y}_{n+1}^* + \mathbf{O}(\tau^5)$, where \mathbf{y}_{n+1} is the solution outputted by ADTIZD-K model (18) while \mathbf{y}_{n+1}^* is the theoretical solution of (9) when $t = t_{n+1}$. Thus, when $n \rightarrow \infty$, it yields from (6) that

$$\begin{aligned} \mathbf{N}_{n+1}\mathbf{y}_{n+1} - \mathbf{d}_{n+1} &= \mathbf{N}_{n+1}(\mathbf{y}_{n+1}^* + \mathbf{O}(\tau^5)) - \mathbf{d}_{n+1} \\ &= \mathbf{N}_{n+1}\mathbf{y}_{n+1}^* - \mathbf{d}_{n+1} + \mathbf{O}(\tau^5) \\ &= \mathbf{O}(\tau^5) \end{aligned} \quad (27)$$

where \mathbf{N}_{n+1} is uniformly bounded because \mathbf{A}_{n+1} and \mathbf{B}_{n+1} are uniformly bounded and \mathbf{M}_{n+1} is consequently bounded. Moreover, when $n \rightarrow \infty$, it follows from (5), (6), and (27) that the equation $\mathbf{N}_{n+1}\mathbf{y}_{n+1} - \mathbf{d}_{n+1}$ can be reformulated as:

$$\mathbf{N}_{n+1}\mathbf{y}_{n+1} - \mathbf{d}_{n+1} = \begin{bmatrix} \text{Re}(\mathbf{M}_{n+1}\mathbf{x}_{n+1} - \mathbf{c}_{n+1}) \\ \text{Im}(\mathbf{M}_{n+1}\mathbf{x}_{n+1} - \mathbf{c}_{n+1}) \end{bmatrix} = \mathbf{O}(\tau^5) \quad (28)$$

where for any complex matrix \mathcal{M} , we have $\mathcal{M} = \text{Re}(\mathcal{M}) + i\text{Im}(\mathcal{M})$. $\text{Re}(\mathcal{M})$ and $\text{Im}(\mathcal{M})$ are real matrices and i is the imaginary unit. From the discrete form of linear (4) and above equality (28), we can get

$$\begin{aligned} \lim_{n \rightarrow \infty} \sup \|\mathbf{A}_{n+1}\mathbf{X}_{n+1} + \mathbf{X}_{n+1}\mathbf{B}_{n+1} - \mathbf{C}_{n+1}\|_F &= \lim_{n \rightarrow \infty} \sup \|\mathbf{M}_{n+1}\mathbf{x}_{n+1} - \mathbf{c}_{n+1}\|_2 \\ &= \lim_{n \rightarrow \infty} \sup \left\{ \|\text{Re}(\mathbf{M}_{n+1}\mathbf{x}_{n+1} - \mathbf{c}_{n+1})\|_2^2 \right. \\ &\quad \left. + \|\text{Im}(\mathbf{M}_{n+1}\mathbf{x}_{n+1} - \mathbf{c}_{n+1})\|_2^2 \right\}^{\frac{1}{2}} \\ &= \lim_{n \rightarrow \infty} \sup \|\mathbf{N}_{n+1}\mathbf{y}_{n+1} - \mathbf{d}_{n+1}\|_2 \\ &= \lim_{n \rightarrow \infty} \sup \|\mathbf{O}(\tau^5)\|_2 \\ &= O(\tau^5). \end{aligned} \quad (29)$$

Equalities (29) have shown that the MSSRE of ADTIZD-K (18) is with a magnitude of $O(\tau^5)$.

We now consider the MSSRE of ADTIZD-U (26) model. Since its truncation error is $\mathbf{O}(\tau^5)$ too, using Theorem 3 similar to the analysis of ADTIZD-K model, we conclude that its MSSRE is $O(\tau^5)$ too. The proof is now complete. ■

Theorems 2–4 have illustrated the novelties of ADTIZD models (18) and (26) with respect to accuracy. In fact, ADTIZD-K model (18) and ADTIZD-U model (26) not only converge with the truncation error of $\mathbf{O}(\tau^5)$ but also solve the DTVCSE problem (3) accurately with MSSRE being the same level of $O(\tau^5)$. However, the improvement of ADTIZD models also lies in their robustness under the perturbation of additive noises. We consider the noise-polluted integration-enhanced ZNN design formula

$$\dot{\mathbf{e}}(t) = -\gamma\mathbf{e}(t) - \xi \int_0^t \mathbf{e}(\sigma)d\sigma + \Delta(t) \quad (30)$$

where $\Delta(t) \in \mathbb{R}^{2pq}$ is the time-varying additive noise. Thus, the noise-polluted CTIZNN model from (30) is

$$\begin{aligned} \dot{\mathbf{y}}(t) &= \mathbf{N}^\dagger(t)(\dot{\mathbf{d}}(t) - \dot{\mathbf{N}}(t)\mathbf{y}(t) - \gamma(\mathbf{N}(t)\mathbf{y}(t) - \mathbf{d}(t)) \\ &\quad - \xi\mathbf{v}(t) + \Delta(t)). \end{aligned} \quad (31)$$

Consequently, the noise-polluted ADTIZD-K model transformed from (31) is

$$\begin{aligned} \mathbf{y}_{n+1} &= \mathbf{y}_n + \frac{\tau}{24}(55\dot{\mathbf{y}}_n - 59\dot{\mathbf{y}}_{n-1} + 37\dot{\mathbf{y}}_{n-2} - 9\dot{\mathbf{y}}_{n-3}), \\ \dot{\mathbf{y}}_{n+j} &= \mathbf{N}_{n+j}^\dagger(\dot{\mathbf{d}}_{n+j} - \dot{\mathbf{N}}_{n+j}\mathbf{y}_{n+j} - \gamma(\mathbf{N}_{n+j}\mathbf{y}_{n+j} - \mathbf{d}_{n+j}) \\ &\quad - \xi\mathbf{v}_{n+j} + \Delta_{n+j}) \end{aligned} \quad (32)$$

where $\Delta_{n+j} = \Delta(t_{n+j})$ and \mathbf{v}_{n+j} is calculated the same as in ADTIZD-K model (18). Note that the noise-polluted ADTIZD-U model can be obtained similar to obtaining (32), thus is omitted here. The robustness of noise-polluted ADTIZND models will be discussed as follows.

Theorem 5: Consider the time-varying noise in noise-polluted CTIZNN model (31) being linear type $\Delta(t) = \mu t + \delta$ with $\mu, \delta \in \mathbb{R}^{2pq}$. The noise-polluted ADTIZD-K model (32) and noise-polluted ADTIZD-U model will converge toward accurate solution of DTVCSE problem (6) with MSSRE being $\lim_{n \rightarrow \infty} \sup \|\mathbf{A}_{n+1}\mathbf{X}_{n+1} + \mathbf{X}_{n+1}\mathbf{B}_{n+1} - \mathbf{C}_{n+1}\|_F \leq \|\mu\|_2/\xi + O(\tau^5)$.

Proof: First, according to [25] and [26], we know that under conditions of this theorem, the noise-polluted CTIZNN model (31) converges toward the precise solution of (6) with the stable residual error being $\lim_{t \rightarrow \infty} \|\mathbf{e}(t)\|_2 = \|\mu\|_2/\xi$. Then, consider the noise-polluted ADTIZD-K model (32), following the deducing process of (27) and (29) in Theorem 4, we have

$$\begin{aligned} &\lim_{n \rightarrow \infty} \sup \|\mathbf{A}_{n+1}\mathbf{X}_{n+1} + \mathbf{X}_{n+1}\mathbf{B}_{n+1} - \mathbf{C}_{n+1}\|_F \\ &= \lim_{n \rightarrow \infty} \sup \|\mathbf{N}_{n+1}\mathbf{y}_{n+1} - \mathbf{d}_{n+1}\|_2 \\ &= \lim_{n \rightarrow \infty} \sup \|\mathbf{N}_{n+1}\mathbf{y}_{n+1}^* - \mathbf{d}_{n+1} + \mathbf{O}(\tau^5)\|_2 \\ &= \lim_{n \rightarrow \infty} \sup \|\mathbf{e}_{n+1}^* + \mathbf{O}(\tau^5)\|_2 \\ &\leq \lim_{n \rightarrow \infty} \sup \left(\|\mathbf{e}_{n+1}^*\|_2 + \|\mathbf{O}(\tau^5)\|_2 \right) \\ &= \|\mu\|_2/\xi + O(\tau^5) \end{aligned}$$

where $\mathbf{e}_{n+1}^* = \mathbf{e}(t_{n+1})$ is the exact error vector of noise-polluted CTIZNN model (31) at $t = t_{n+1}$.

If we take the noise-polluted ADTIZD-U model into consideration, its MSSRE is also bounded by $\|\mu\|_2/\xi + O(\tau^5)$ and can be proved with the same techniques. Now, Theorem 5 has been proved. ■

Remark 1: In terms of robustness, Theorem 5 has demonstrated the powerful noise resistance abilities of noise-polluted ADTIZD models. When perturbed by constant additive noise $\Delta(t) = \delta$, such noise is treated as one specific variation of linear noise $\Delta(t) = \mu t + \delta$ with $\mu = \mathbf{0}$. Thus, the noise-polluted ADTIZD-K model (32) and noise-polluted ADTIZD-U model can solve the DTVCSE problem (2) effectively with the MSSRE still being $O(\tau^5)$, i.e., almost no accuracy loss. In other cases when $\mu \neq \mathbf{0}$, the MSSREs of noise-polluted ADTIZD-K (32) and ADTIZD-U models are still bounded as long as $\|\mu\|_2$ is bounded and this upper bound can be reduced by increasing ξ .

It is worth noting that the conventional CTZNN formula has been extensively used in constructing DTZDs. The conventional continuous ZNN formula is $\dot{\mathbf{e}}(t) = -\gamma\mathbf{e}(t)$, whose global stability has been proved in [19], [20], and [22]. Correspondingly, the additive noise-polluted continuous-time ZNN formula is

$$\dot{\mathbf{e}}(t) = -\gamma\mathbf{e}(t) + \Delta(t). \quad (33)$$

Our ADTIZD models have made great improvements in robustness, especially when compared with the noise-polluted CTZNN formula (33)-based DTZDs. To lay the basis for ensuing comparisons, we directly present the noise-polluted Adams-type DTZND with know derivatives (ADTIZD-K) model as

$$\begin{aligned} \mathbf{y}_{n+1} &= \mathbf{y}_n + \frac{\tau}{24}(55\dot{\mathbf{y}}_n - 59\dot{\mathbf{y}}_{n-1} + 37\dot{\mathbf{y}}_{n-2} - 9\dot{\mathbf{y}}_{n-3}) \\ \dot{\mathbf{y}}_{n+j} &= \mathbf{N}_{n+j}^\dagger(\dot{\mathbf{d}}_{n+j} - \dot{\mathbf{N}}_{n+j}\mathbf{y}_{n+j} - \gamma(\mathbf{N}_{n+j}\mathbf{y}_{n+j} - \mathbf{d}_{n+j}) + \Delta_{n+j}) \end{aligned} \quad (34)$$

which bases on the noise-polluted traditional CTZNN (33).

IV. MULTI-INSTANT DTIZD MODELS

In Section III, two different Adams-type DTIZD models for solving the DTVCSE (2) problem have been proposed based on the Adams–Bashforth discrete formulas (14)–(17). Nevertheless, there are some existing discrete formulas that can be applied to the CTIZNN model (9) for such a problem. For comparison purpose, some representative multi-instant DTIZD models will be developed for the DTVCSE problem in this section. Note that for avoiding repetitive contents, we only provide models, whose coefficient derivatives from (9) are known and the inverse matrix are directly computed similar to the ADTIZD-K (18) model.

A. 6-Instant DTIZD Model

In Section III, it has been assumed that $\mathbf{y}(t) \in S^5[t_0, t_f]$, thus $\mathbf{y}(t) \in S^4[t_0, t_f]$ is true. Then, the 6-instant discrete formula has been presented to discretize various continuous ZNN models,

whose formula is written as [19]

$$\begin{aligned} \mathbf{y}_{n+1} \approx & \frac{6}{13}\mathbf{y}_n + \frac{2}{13}\mathbf{y}_{n-1} + \frac{4}{13}\mathbf{y}_{n-2} + \frac{3}{13}\mathbf{y}_{n-3} - \frac{2}{13}\mathbf{y}_{n-4} \\ & + \frac{24}{13}\tau\dot{\mathbf{y}}_n. \end{aligned} \quad (35)$$

Discrete formula (35) is convergent and has the truncation error of $\mathbf{O}(\tau^4)$ [19], we utilize it on the CTIZNN model (9) leads to

$$\begin{aligned} \mathbf{y}_{n+1} = & \frac{6}{13}\mathbf{y}_n + \frac{2}{13}\mathbf{y}_{n-1} + \frac{4}{13}\mathbf{y}_{n-2} + \frac{3}{13}\mathbf{y}_{n-3} - \frac{2}{13}\mathbf{y}_{n-4} \\ & + \frac{24}{13}\tau\dot{\mathbf{y}}_n \\ \dot{\mathbf{y}}_n = & \mathbf{N}_n^\dagger(\dot{\mathbf{d}}_n - \dot{\mathbf{N}}_n\mathbf{y}_n - \gamma(\mathbf{N}_n\mathbf{y}_n - \mathbf{d}_n) - \xi\mathbf{v}_{n+1}). \end{aligned} \quad (36)$$

Then, (36) is termed as the 6-instant DTIZD with the known derivative (6IDTIZD-K) model. It can be proved by exploiting techniques from Theorem 2 and 4 that, for the DTVCSSE (2) problem, the 6IDTIZD model is convergent with its MSSRE $\lim_{n \rightarrow \infty} \sup \|\mathbf{A}_{n+1}\mathbf{X}_{n+1} + \mathbf{X}_{n+1}\mathbf{B}_{n+1} - \mathbf{C}_{n+1}\|_F$ being $\mathbf{O}(\tau^4)$.

B. 4-Instant DTIZD Model

As in Section IV-A, $\mathbf{y} \in \mathcal{S}^3[t_0, t_f]$ is also true. Then, the 4-instant discrete formula is presented in [32], whose expression is as follows:

$$\mathbf{y}_{n+1} \approx \frac{3}{2}\mathbf{y}_n - \mathbf{y}_{n-1} + \frac{1}{2}\mathbf{y}_{n-2} + \tau\dot{\mathbf{y}}_n. \quad (37)$$

The residual error of the above formula (37) is $\mathbf{O}(\tau^3)$. Therefore, the following model can be derived by combining (37) and CTIZNN model (9):

$$\begin{aligned} \mathbf{y}_{n+1} = & \frac{3}{2}\mathbf{y}_n - \mathbf{y}_{n-1} + \frac{1}{2}\mathbf{y}_{n-2} + \tau\dot{\mathbf{y}}_n \\ \dot{\mathbf{y}}_n = & \mathbf{N}_n^\dagger(\dot{\mathbf{d}}_n - \dot{\mathbf{N}}_n\mathbf{y}_n - \gamma(\mathbf{N}_n\mathbf{y}_n - \mathbf{d}_n) - \xi\mathbf{v}_{n+1}). \end{aligned} \quad (38)$$

We name (38) as the 4-instant DTIZD with the known derivative (4IDTIZD-K) model. Besides, Guo *et al.*[32] have proved that (37) is convergent, following the proof process of Theorems 2 and 4, we are able to prove that the MSSRE of the 4IDTIZD-K model on solving DTVCSSE problem (2) is at the level of $\mathbf{O}(\tau^3)$ with its proof being omitted.

C. 2-Instant DTIZD Model

The 2-instant discrete formula or the so-called Euler-type discrete formula is widely used to construct the DTIZD models [19]. Derived from the Euler forward difference formula, the 2-instant discrete formula is written as

$$\mathbf{y}_{n+1} \approx \mathbf{y}_n + \tau\dot{\mathbf{y}}_n \quad (39)$$

which possesses the truncation error of $\mathbf{O}(\tau^2)$. We use (39) to discretize our CTIZNN model (9) and obtain the 2-instant DTIZD with the known derivative (2IDTIZD-K) model:

$$\begin{aligned} \mathbf{y}_{n+1} = & \mathbf{y}_n + \tau\dot{\mathbf{y}}_n \\ \dot{\mathbf{y}}_n = & \mathbf{N}_n^\dagger(\dot{\mathbf{d}}_n - \dot{\mathbf{N}}_n\mathbf{y}_n - \gamma(\mathbf{N}_n\mathbf{y}_n - \mathbf{d}_n) - \xi\mathbf{v}_{n+1}). \end{aligned} \quad (40)$$

In [19], the discrete formula (39) has been proved to be convergent. Thus, similar to proving Theorems 2 and 4, we conclude

that the MSSRE level of the 2IDTIZD-K model for solving DTVCSSE (2) problem is $\mathbf{O}(\tau^2)$ with its proof being omitted.

Remark 2: For demonstrating the differences as well as novelties of our ADTIZD-K (18) model for the DTVCSSE problem (2), three conventional multi-instant discrete formulas together with their corresponding DTIZD models have been proposed in this section. Since the CTIZNN models that we used in this section are the same, the main difference between ADTIZD models and multi-instant DTIZD models is clearly their discrete formulas. For conventional multi-instant discrete formulas (35), (37), and (39), we can find that: in each update, they all use several sampling points of $\mathbf{y}(t)$ while only using one sampling point of $\dot{\mathbf{y}}(t)$. This characteristic of conventional multi-instant discrete formulas exists because they all originate from the first-order derivative approximation formulas, which only contain one derivative element. However, such a designing method has limited the investigation process of obtaining more efficient discrete formulas. For such a reason, we have introduced the more general form of discrete formulas (10) and use it to obtain the novel Adams–Bashforth formulas (14)–(17) in Section III.

Benefiting from its special origin (13), the 4-order Adams–Bashforth discrete formula (17) and ADTIZD-K (18) model have made comprehensive performance improvements, which are listed as follows.

- 1) Let us consider model accuracy. When $\tau < 1$, the MSSRE of the ADTIZD-K model is $\mathbf{O}(\tau^5)$, which is one order smaller than $\mathbf{O}(\tau^4)$ of the 6IDTIZD-K (36) model and is more smaller than MSSRE of 4IDTIZD-K (38) or 2IDTIZD-K (38) model. This means that when $\tau < 1$, the ADTIZD-K model is more accurate than three multi-instant DTIZD models that are from Section IV. Such a precision gap will become even greater when τ decreases.
- 2) The ADTIZD-K (18) model has also made improvements in the aspect of neural network complexity. We can reasonably assume that all history information to be used in the next update is stored in the neural network, such as past sampling points of $\mathbf{y}(t)$ and $\dot{\mathbf{y}}(t)$. Then, calculations (additions and subtractions) in the proposed ADTIZD-K (18) model's each update are even $2pq$ less than that of the 6IDTIZD-K (36) model and only equal to that of 4IDTIZD-K (38) model. As for the storage complexity, the ADTIZD-K (18) model stores $2pq$ numbers less than the 6IDTIZD-K (36) model, only equal to 4IDTIZD-K (38) model's storage usage.

The above improvements mainly comes from the novel Adams–Bashforth discrete formula, which inversely demonstrates the effectiveness of introducing the Adams–Bashforth discrete formula.

V. NUMERICAL VALIDATIONS AND COMPARISONS

In this section, for validating the efficacy and superiority of our ADTIZD models in solving DTVCSSE problems, we will conduct numerical experiments and provide comparisons between different models. The CTVCSE problem used in the

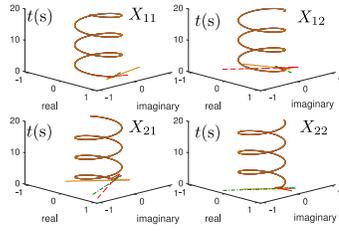


Fig. 1. State solutions \mathbf{X}_n outputted by the ADTIZD-K model (18) for solving DTVCSSE problem (2). Blue dotted lines are theoretical solutions, red dotted lines, green dotted lines and orange solid lines are state solutions when $\tau = 0.01, 0.005$ and 0.001 s, respectively.

following experiments is considered:

$$\begin{aligned} \mathbf{A}(t) &= \begin{bmatrix} \exp(it) & -i\exp(-it) \\ -i\exp(it) & \exp(-it) \end{bmatrix} \\ \mathbf{B}(t) &= \begin{bmatrix} i\exp(-it) & 2\exp(it) \\ 2\exp(it) & i\exp(-it) \end{bmatrix} \\ \mathbf{C}(t) &= \begin{bmatrix} \frac{1}{2}i\exp(-2it) + 1 + 1i & 1 - \frac{1}{2}\exp(-2it) \\ \exp(2it) - \frac{1}{2} & i\exp(2it) + 1 + \frac{1}{2}i \end{bmatrix}. \end{aligned} \quad (41)$$

Following the definition of CTVCSSE (1) problem's input matrices (41), the theoretical solution of the corresponding CTVCSSE (41) problem can be calculated:

$$\mathbf{X}^*(t) = \frac{1}{2} \begin{bmatrix} \exp(-it) & i\exp(-it) \\ i\exp(it) & \exp(it) \end{bmatrix}.$$

Thus, the accurate solution of the corresponding DTVCSSE problem (2) is $\mathbf{X}_n^* = \mathbf{X}^*(t_n)$ and can be exploited to verify the output solutions in experiments. Besides, the problem duration in the following experiments is set to 20 s.

First, let us examine the performance of ADTIZD-K model (18) in solving DTVCSSE problem (2) within a noise-free environment. To further present novelties of the proposed ADTIZD-K model (18), other multi-instant DTIZD models proposed in Section IV are included in this experiments for comparison. For illustration convenience, $s_t = 0.2$ and $\xi = \gamma^2$ are set, where $s_t = \gamma\tau$ is the step size of DTIZD models. Evidently, γ and ξ vary with different τ and fixed s_t . Elements of the initial vector $\mathbf{y}(0)$ are randomly generated in interval $[-1, 1]$, meaning that the real parts and imaginary parts of $\mathbf{x}(0)$ are both randomly generated within $[-1, 1]$. Under such conditions, with different sampling time periods $\tau = 0.01, 0.005, 0.001$ s, the output solutions of our ADTIZD-K model (18) are plotted in Fig. 1 together with DTVCSSE (2) problem's theoretical solution. As it is depicted in Fig. 1, beginning from different original states, the output solutions of ADTIZD-K models (18) always converge to the accurate solution of DTVCSSE (2) problem quickly and stably.

For comparison purpose, we have conducted more experiments with the ADTIZD-K model and other multi-instant DTIZD models. The residual errors $\|\mathbf{E}_n\|_F$ generated by these models in solving DTVCSSE problem (2) are depicted in Fig. 2(a)–(d), where $\mathbf{E}_n = \mathbf{A}_n\mathbf{X}_n + \mathbf{X}_n\mathbf{B}_n - \mathbf{C}_n$. Fig. 2(a)–(d) has shown that when $\tau = 0.05, 0.01, 0.005$, and 0.001 s, the steady-state residual errors $\|\mathbf{E}_n\|_F$ of the proposed ADTIZD-K models (18) converge toward 0 at order of $10^{-7}, 10^{-10}, 10^{-12}$, and 10^{-15} , i.e.,

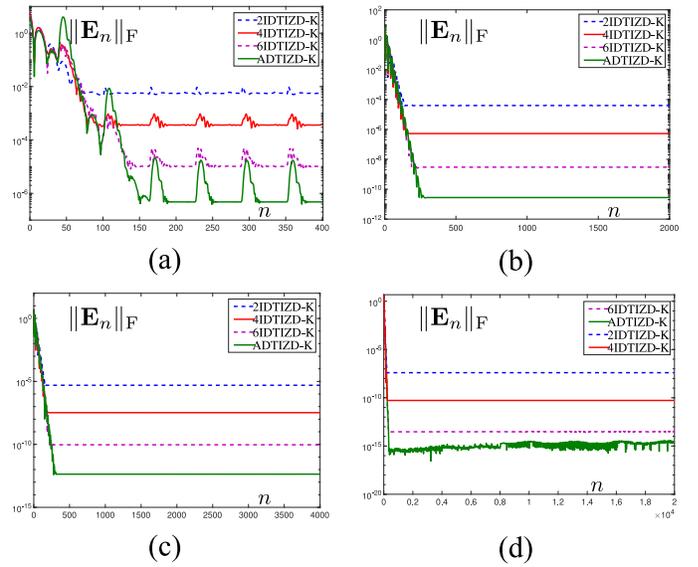


Fig. 2. Residual error $\|\mathbf{E}_n\|_F$ trajectories of ADTIZD-K (18), 2IDTIZD-K (40), 4IDTIZD-K (38) and 6IDTIZD-K (36) models with different τ, γ , and ξ for solving the DTVCSSE problem (2). (a) $\tau = 0.05, \gamma = 4$ and $\xi = 4^2$. (b) $\tau = 0.01, \gamma = 20$ and $\xi = 20^2$. (c) $\tau = 0.005, \gamma = 40$ and $\xi = 40^2$. (d) $\tau = 0.001, \gamma = 200$, and $\xi = 200^2$.

$O(0.05^5), O(0.01^5), O(0.005^5)$, and $O(0.001^5)$, respectively. Evidently, the ADTIZD-K model (18) can solve the DTVCSSE problem (2) efficiently with different τ . Furthermore, it can be observed from Fig. 2(a)–(d) that in each condition, our ADTIZD-K model (18) always possesses the smallest steady-state residual error $\|\mathbf{E}_n\|_F$ among all the models. In contrast, $\|\mathbf{E}_n\|_F$ of 6IDTIZD-K model (36), 4IDTIZD-K model (38), and 2IDTIZD-K model (40) only change in $O(\tau^2), O(\tau^3)$, and $O(\tau^4)$ patterns, respectively. The above results have verified the conclusions in Theorem 4 and Section IV. Thus, we conclude from Theorem 4 that if the sampling time interval τ drops by ten times, the MSSRE $\lim_{n \rightarrow \infty} \sup \|\mathbf{E}_n\|_F$ of ADTIZD-K model (18) decreases rapidly by 100 000 times. This kind of fast precision improvement not only gives the ADTIZD-K model (18) better performance but also provides the model with more computation time when certain accuracy requirement is given.

Now, let us consider examining the robustness performance of the ADTIZD-K model (18) in solving the DTVCSSE problem (2). Therefore, the noise-polluted ADTIZD-K model (32) is examined in the following robustness experiments. Besides, the ordinary noise-polluted ADTIZD-K model (34) is added in these experiments as a comparison model. First, the parameters are set as $s_t = \gamma\tau = 0.2, \tau = 0.001$, and $\xi = \gamma^2/10 = 4 \times 10^3$. Under the pollution of constant additive noise $\Delta(t) = \delta_1 = [5, 1, -2, 1, 1, -1, 2, 0]^T$, the solution \mathbf{X}_n calculated by two noise-polluted models (32) and (34) are presented in Fig. 3(a), so are the corresponding computation errors $\|\mathbf{E}_n\|_F$. Clearly, just the constant noise δ_1 has seriously reduced the computation accuracy of noise-polluted ADTIZD-K model (34), whose error $\|\mathbf{E}_n\|_F$ in Fig. 3(b) keeps stable at a relative high level (about 3×10^{-2}). However, the error $\|\mathbf{E}_n\|_F$ of noise-polluted ADTIZD-K model (32) becomes stable at about 2×10^{-15} , which

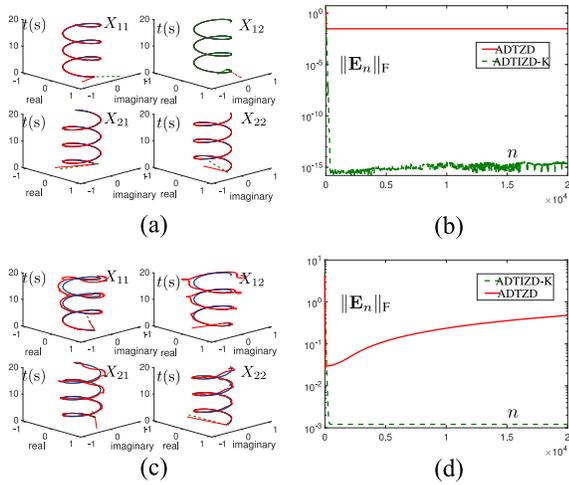


Fig. 3. Output solutions and corresponding computation errors $\|\mathbf{E}_n\|_F$ of the noise-polluted ADTIZD-K model (34) and the noise-polluted ADTIZD-K model (32) under different types of additive noise perturbations. (a) State solutions \mathbf{X}_n with constant noise δ_1 . (b) Residual errors $\|\mathbf{E}_n\|_F$ with constant noise δ_1 . (c) Solutions \mathbf{X}_n with linear noise $\mu_1 + \delta_1$. (d) Residual errors $\|\mathbf{E}_n\|_F$ with linear noise $\mu_1 + \delta_1$. [In solutions, blue dotted lines are accurate solutions, red solid lines are from noise-polluted ADTIZD-K model (34) while green dotted lines are from noise-polluted ADTIZD-K model (32).]

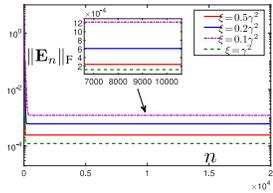


Fig. 4. Computation errors $\|\mathbf{E}_n\|_F$ of noise-polluted ADTIZD-K model (32) on solving the DTVCSSE problem (2) under perturbation of linear noise $\mu_2 t + \delta_2$ and with different ξ .

can be seen as no accuracy loss when compared with noise-free results like in Fig. 2(d). Correspondingly, it can be seen from Fig. 3(a) that the state solution of ADTIZD-K model (32) overlaps with the accurate solution very well. We then change the additive noise to more complex linear type $\Delta(t) = \mu_1 t + \delta_1$, where $\mu_1 = [1, -2, 3, 3, 0, 1, 0, -2]^T$ and $\|\mu_1\|_2/\xi \approx 1.3229 \times 10^{-3}$, then the state outputs of noise-polluted ADTIZD-K model (32) and noise-polluted ADTIZD-K model (34) are presented in Fig. 3(c). According to Fig. 3(c), the solution of noise-polluted ADTIZD-K model (34) becomes more inaccurate and diverges away from the theoretical solution, but the solution of noise-polluted ADTIZD-K model (32) still matches with the theoretical solution well. Moreover, the stable residual error of noise-ADTIZD-K (32) holds stable at about 1.2×10^{-3} in Fig. 3(d) and this inversely verifies our analysis results in Theorem 5. On the contrary, the residual error of noise-polluted ADTIZD-K model (34) keeps going high above 0.1 while showing no sign of stopping.

As we have discussed in Remark 1, the noise resistance ability of the noise-polluted ADTIZD-K model (32) is enhanced with the increase of design parameter ξ . In this case, to avoid occasional factors, the constant additive noise we use is $\Delta(t) = \delta_2 = [4, -2, 5, 0, -1, 3, 4, 1]^T$ while the linear additive noise is $\Delta(t) = \mu_2 t + \delta_2$, $\mu_2 = [-3, 0, 1, 2, -1, 3, 1, 2]^T$ with $\|\mu_2\|_2 \approx 5.3852$. We then change ξ to $\xi = 0.1\gamma^2 = 4 \times 10^3$, $0.2\gamma^2 = 8 \times 10^3$, $0.5\gamma^2 = 2 \times 10^4$, and $\gamma^2 = 4 \times 10^4$,

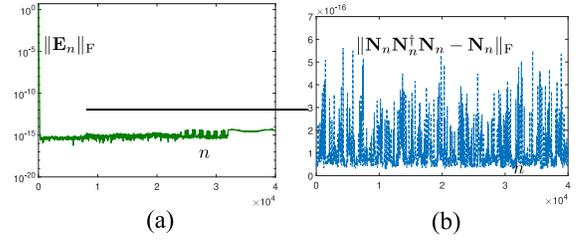


Fig. 5. Profiles of ADTIZD-K model (18) on solving DTVCSSE problem (2) with $\tau = 5 \times 10^{-4}$ s. (a) Residual error $\|\mathbf{E}_n\|_F$. (b) Moore-Penrose inverse error $\|\mathbf{N}_n \mathbf{N}_n^+ \mathbf{N}_n - \mathbf{N}_n\|_F$ of MATLAB built-in function.

i.e., $\|\mu_2\|_2/\xi \approx 1.3463 \times 10^{-3}$, 6.7315×10^{-4} , 2.6926×10^{-4} , and 1.3463×10^{-4} , other settings are kept the same. Thus, the residual error $\|\mathbf{E}_n\|_F$ of noise-polluted ADTIZD-K model (32) is illustrated in Fig. 4, which conforms to the bound of MSSRE $\|\mu_2\|_2/\xi + O(\tau^5)$ exactly again. The superior robustness performance of ADTIZD-K model (18) has now been validated by the above comparison numerical experiments.

In this work, based on the general ADTIZD-K model (18), another ADTIZD-U (26) model has also been developed for situations with insufficient information when coefficients time derivatives are unknown. Therefore, we are now going to verify and compare these two models' efficacy in solving DTVCSSE problem (2). This time, we have conducted more numerical experiments using different combinations of model design parameters. There are totally four changing parameters involved in the combinations, which are s_t , τ , model type, and noise type. Besides, we have $\xi = 0.5 \times \gamma^2$. Corresponding experiments results are summarized in Table I, where MSSRE and average computing time per updating (ACTPU) are provided. It is worth noting that in these experiments, the MSSRE is defined as $\max\{\|\mathbf{E}(t)\|_F\} \forall t \in [19, 20]$ s, because the strict definition of MSSRE is $\max\{\|\mathbf{E}(t)\|_F\}, t \rightarrow +\infty$ and is unachievable. As can be observed in Table I, the performances of ADTIZD-K model (18) and ADTIZD-U model (26) are close. In every equivalent experiment, the MSSRE of ADTIZD-U (26) model is only slightly larger than that of ADTIZD-K model (18) due to the accuracy loss of the time derivative approximation. All the data in Table in I also match with our theoretical conclusions in above sections well. Furthermore, it can be observed in Table I that the changing patterns of MSSREs of both ADTIZD models conform to the theoretical conclusions in Section III well. All ACTPUs are less than 5×10^{-4} s, meaning that our ADTIZD models work perfectly with designated τ . Finally, as it is illustrated in Table I, when $\xi = s_t/\tau$ increases, the MSSREs of noise-polluted ADTIZD models decrease accordingly, which obeys the conclusion in Theorem 5.

It should be noted that in testing experiments, we have tried $\tau = 5 \times 10^{-4}$ s, which means $O(\tau^5) \approx O(10^{-17})$, however, the MSSRE of ADTIZD-K model (18) has stuck at about 10^{-15} – 10^{-16} as shown in Fig. 5(a). The reason for such problem is that we use the built function in MATLAB to calculate Moore-Penrose inverse \mathbf{N}_n^+ , but such inverse results have precision limit. The Moore-Penrose inverse error $\|\mathbf{N}_n \mathbf{N}_n^+ \mathbf{N}_n - \mathbf{N}_n\|_F$ presented in Fig. 5(b) is exactly about 1×10^{15} to 1×10^{16} . Based on the result of above trial, the smallest τ we use in above experiments is $\tau = 0.001$ s for illustration purpose.

TABLE I
MSSRE AND ACTPU OF DIFFERENT ADTIZD MODELS WITH DIFFERENT PARAMETER SETTINGS ON SOLVING THE DTVCSSE PROBLEM (2) WITH DIFFERENT NOISE POLLUTIONS WHERE ACTPU ($\times 10^{-4}$ s) IS IN THE PARENTHESES

| Model | Noise | s_t | $\tau = 0.05$ s | $\tau = 0.01$ s | $\tau = 0.005$ s | $\tau = 0.001$ s |
|----------|----------------------|-------|----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| ADTIZD-K | None | 0.2 | 7.8405×10^{-7} (3.3666) | 5.5111×10^{-11} (2.4842) | 8.6349×10^{-13} (2.0712) | 8.4144×10^{-15} (1.9274) |
| | | 0.3 | 1.1001×10^{-6} (3.1981) | 2.4500×10^{-11} (2.4578) | 3.8499×10^{-13} (2.0825) | 9.7821×10^{-15} (1.8349) |
| | δ_1 | 0.2 | 7.8405×10^{-7} (2.5166) | 5.5111×10^{-11} (2.4153) | 8.6353×10^{-13} (2.0873) | 8.4813×10^{-15} (1.9172) |
| | | 0.3 | 1.1001×10^{-6} (3.2227) | 2.4500×10^{-11} (2.4212) | 3.8498×10^{-13} (2.1553) | 9.9041×10^{-15} (1.9597) |
| | $\mu_1 t + \delta_1$ | 0.2 | 6.1057×10^{-1} (2.4346) | 2.4232×10^{-2} (2.4794) | 6.0579×10^{-3} (2.0378) | 2.4232×10^{-4} (1.9573) |
| | | 0.3 | 3.0991×10^{-1} (2.5476) | 1.0770×10^{-2} (2.3601) | 2.6924×10^{-3} (1.9817) | 1.0770×10^{-4} (2.0758) |
| ADTIZD-U | None | 0.2 | 1.4749×10^{-5} (3.3458) | 1.0352×10^{-9} (2.9208) | 1.6218×10^{-11} (2.5032) | 1.4152×10^{-13} (2.2880) |
| | | 0.3 | 2.1587×10^{-5} (3.2876) | 4.6024×10^{-10} (2.9651) | 7.2420×10^{-12} (2.5127) | 1.7008×10^{-13} (2.5028) |
| | δ_1 | 0.2 | 1.4749×10^{-5} (3.3758) | 1.0352×10^{-9} (3.1957) | 1.6218×10^{-11} (2.4488) | 1.4148×10^{-13} (2.5054) |
| | | 0.3 | 2.1587×10^{-5} (3.7773) | 4.6024×10^{-10} (2.7251) | 7.2422×10^{-12} (2.6772) | 1.7064×10^{-13} (2.2756) |
| | $\mu_1 t + \delta_1$ | 0.2 | 6.1056×10^{-1} (3.0805) | 2.4232×10^{-2} (3.0305) | 6.0579×10^{-3} (2.5007) | 2.4232×10^{-4} (2.4849) |
| | | 0.3 | 3.0990×10^{-1} (3.5098) | 1.0770×10^{-2} (2.6582) | 2.6924×10^{-3} (2.3869) | 1.0770×10^{-4} (2.1053) |

VI. APPLICATION TO ROBOT KINEMATICS CONTROL

It has been introduced in Section I that a variety of practical applications take TVSE problems as their core problems. Thus, one of our ADTIZD models (18), (26) will be used to inverse kinematically control a robot arm.

Here, in this section, the redundant robot arm to be controlled is the PUMA 560, which is a robot manipulator with 6 degree of freedoms (DOFs) and works in three-dimensional (3-D) space. Before entering the robot inverse kinematics control part, let us first introduce the basic forward-kinematics equation of such a robot arm [33], [34]

$$\mathbf{r}(t) = \boldsymbol{\phi}(\boldsymbol{\eta}(t))$$

where $\mathbf{r}(t) \in \mathbb{R}^3$ is the coordinates of the robot arm's end-effector in 3-D space, $\boldsymbol{\eta}(t) \in \mathbb{R}^6$ is the joint angle vector. Then, $\boldsymbol{\phi}(\cdot) : \mathbb{R}^6 \rightarrow \mathbb{R}^3$ denotes the mapping function from the manipulator's state parameters to end-effector position, and $\boldsymbol{\phi}(\cdot)$ is determined by PUMA 560's physical structure and is known.

Generally speaking, the inverse kinematics control of a robot manipulator is to figure out the dynamic control parameter $\boldsymbol{\eta}(t)$ when given a dynamic end-effector path $\mathbf{r}(t)$. However, for a redundant robot arm, such as PUMA560, its DOF redundancy results in infinite solutions of $\boldsymbol{\eta}(t)$ for one same $\mathbf{r}(t)$ while the continuous property of $\boldsymbol{\eta}(t)$ is also required. Therefore, the inverse kinematics control of PUMA560 is a very tough job and we usually solve it in the velocity level with the following method [35]:

$$\begin{aligned} \mathbf{W}_J(\boldsymbol{\eta}(t))\dot{\boldsymbol{\eta}}(t) &= \dot{\mathbf{r}}(t), \\ \dot{\boldsymbol{\eta}}(t) &= \mathbf{W}_J^\dagger(\boldsymbol{\eta}(t))\dot{\mathbf{r}}(t) \end{aligned} \quad (42)$$

where $\mathbf{W}_J(\boldsymbol{\eta}(t)) = \partial\boldsymbol{\phi}(\boldsymbol{\eta}(t))/\partial\boldsymbol{\eta}(t) \in \mathbb{R}^{3 \times 6}$ is the Jacobian matrix and is known, and $\mathbf{W}_J^\dagger(\boldsymbol{\eta}(t)) \in \mathbb{R}^{6 \times 3}$ stands for $\mathbf{W}_J(\boldsymbol{\eta}(t))$'s Moore-Penrose inverse. Evidently, the robot manipulator control job can be accomplished by obtaining $\mathbf{W}_J^\dagger(\boldsymbol{\eta}(t))$ in real time and we can obtain this inverse matrix with the following formula:

$$\mathbf{X}(t)\mathbf{W}_J(\boldsymbol{\eta}(t))\mathbf{W}_J^\dagger(\boldsymbol{\eta}(t)) = \mathbf{W}_J^\dagger(\boldsymbol{\eta}(t))$$

where $\mathbf{X}(t)$ represents the theoretical value of $\mathbf{W}_J^\dagger(\boldsymbol{\eta}(t))$. If we set the coefficient matrices in CTVCSE (1) problem as $\mathbf{A}(t) = \mathbf{0}$, $\mathbf{B}(t) = \mathbf{W}_J(\boldsymbol{\eta}(t))\mathbf{W}_J^\dagger(\boldsymbol{\eta}(t))$, $\mathbf{C}(t) = \mathbf{W}_J^\dagger(\boldsymbol{\eta}(t))$, the above equation can be transformed into a special CTVCSE problem. With the above robot arm tracking control model, the proposed ADTIZD models (18), (26) can now be used to control the robot manipulator. Note that the control scheme (42) not works in position level, thus a position error feedback is added to (42) as follows:

$$\dot{\boldsymbol{\eta}}(t) = \mathbf{W}_J^\dagger(\boldsymbol{\eta}(t))(\dot{\mathbf{r}}_d(t) + g(\mathbf{r}_d(t) - \mathbf{r}_a(t))) \quad (43)$$

where $\mathbf{r}_d(t) \in \mathbb{R}^3$ denotes the given desired path, $\mathbf{r}_a(t) \in \mathbb{R}^3$ denotes the actual end-effector coordinates and $g > 0$ is a scale parameter for adjusting feedback. Also note that to better simulate the complex real-world application scene where necessary information for the control model may be missing, the more general ADTIZD-U model (26) is used in the following simulation. Furthermore, we assume that the derivative of $\mathbf{r}_d(t)$ is unknown and use the first-order derivative approximation formula (25) to get $\dot{\mathbf{r}}_d(t)$. The information above the control process can directly obtain is very limited, which may reduced the tracking control accuracy, but the simulation results can better reflect real-world performance.

In this simulation, parameters are selected as $\tau = 0.001$ s, $s_t = \tau\gamma = 0.2$, and $\xi = \gamma^2/2 = 2 \times 10^4$ in ADTIZD-U model (26) and $g = 50$ in (43), the simulation duration is 20 s. Then, the PUMA560 is controlled to track a rotated Lissajous shape path with all corresponding results plotted in Figs. 6 and 7. First, the trajectory of the end effector and the desired Lissajous path in 3-D space are shown in Fig. 6(a), it is obvious that end-effector traces the given path precisely and they overlap strictly. The movement of the robot manipulator in the hole tracking process can be seen in Fig. 6(b), where we can find that all joints of PUMA560 work coordinately and have successfully finished the tracking task. The motion of robot arm's joints can also be reflected by changing curves of joint angles $\boldsymbol{\eta}(t)$ or joint angle velocities $\dot{\boldsymbol{\eta}}(t)$, which are depicted in Fig. 7(a) and (b), respectively. It can be observed from Fig. 7(a) and (b) that both joint angles and joint angle velocities change continuously and smoothly, thus the control

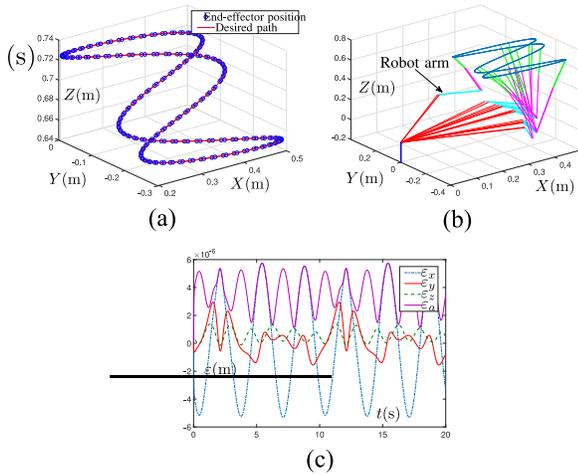


Fig. 6. Profiles of using the ADTIZD-K model (18) to control robot manipulator PUMA560 to track rotated Lissajous shape path. (a) Actual trace of the end-effector and the desired path in 3-D space. (b) Motion of the hole robot arm during tracking control. (c) End-effector position errors in three dimensions being ε_x , ε_y , ε_z and in space being ε_a .

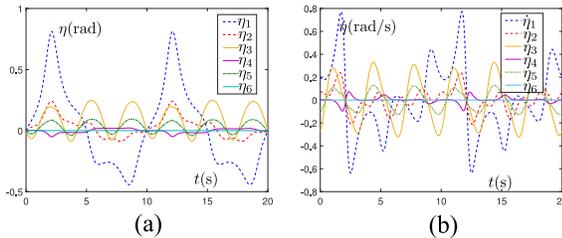


Fig. 7. Joint profiles of PUMA560 during tracking process. (a) Joint angular trajectories. (b) Joint angle velocity trajectories.

signals output by our ADTIZD-U model (26)-based robot control model are suitable for practical applications. Last but not least, the position errors of the end effector are demonstrated in Fig. 6(c), where ε_x , ε_y , and ε_z are the position errors in three dimensions, respectively, while ε_a is the total position error in 3-D space. As can be found in Fig. 6(c), all these tracking errors are always tiny with their magnitudes being around 10^{-6} – 10^{-7} m. Therefore, we conclude that the positioning precision in above simulation experiment is very high, even though in our previous setting the control model is only given highly limited information. Now, the efficacy and application potential of our proposed ADTIZD-U model (26) have been validated by the above simulation results.

VII. CONCLUSION

After introducing the novel Adams–Bashforth discrete formulas, in this article, we have used them to discretize a integration-enhanced CTZNN. Then, the resultant ADTIZD-K model (18) and ADTIZD-U model (26) for solving the DTVCSE problem (2) are proposed for the first time. The convergence, accuracy, and noise tolerance of ADTIZD models have all been theoretically analyzed and proved, where ADTIZD models have high accuracy but low calculation and storage complexity. ADTIZD models are resistant to additive noises, constant noises cannot influence their accuracy

and they are stable even under large linear noises. In addition, numerical experiments have been conducted to validate ADTIZD models' performance, and comparisons with other conventional discrete-time ZNN models have demonstrated our ADTIZD models' superiority. Finally, the ADTIZD-U model (26) is successfully applied in the robot manipulator path tracking control application, showing ADTIZD models' efficacy. Further reducing ADTIZD models' computation complexity is very meaningful and can be future work.

REFERENCES

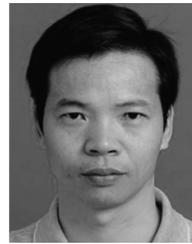
- [1] L. Jin, J. Yan, X. Du, X. Xiao, and D. Fu, "RNN for solving time-variant generalized Sylvester equation with applications to robots and acoustic source localization," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6359–6369, Oct. 2020.
- [2] Q. Wei, N. Dobigeon, and J.-Y. Tourneret, "Fast fusion of multi-band images based on solving a Sylvester equation," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4109–4121, Nov. 2015.
- [3] S. Dolgov, J. Pearson, D. Savostyanov, and M. Stoll, "Fast tensor product solvers for optimization problems with fractional differential equations as constraints," *Appl. Math. Comput.*, vol. 273, pp. 604–623, Jan. 2016.
- [4] L. Xiao, Z. Zhang, Z. Zhang, W. Li, and S. Li, "Design, verification and robotic application of a novel recurrent neural network for computing dynamic Sylvester equation," *Neural Netw.*, vol. 105, pp. 185–196, Sep. 2018.
- [5] L. Xiao, B. Liao, J. Luo, and L. Ding, "A convergence-enhanced gradient neural network for solving Sylvester equation," in *Proc. 36th Chin. Control Conf. (CCC)*, 2017, pp. 3910–3913.
- [6] Z. Tan, Y. Hu, and K. Chen, "On the investigation of activation functions in gradient neural network for online solving linear matrix equation," *Neurocomputing*, vol. 413, pp. 185–192, Nov. 2020.
- [7] S. Liu, H. Tan, and L. Xiao, "A fully complex-valued and robust ZNN model for dynamic complex matrix inversion under external noises," *IEEE Access*, vol. 8, pp. 87478–87489, 2020.
- [8] D. Guo, S. Li, and P. S. Stanimirović, "Analysis and application of modified ZNN design with robustness against harmonic noise," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4627–4638, Jul. 2020.
- [9] L. Jin, S. Li, L. Xiao, R. Lu, and B. Liao, "Cooperative motion generation in a distributed network of redundant robot manipulators with noises," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 10, pp. 1715–1724, Oct. 2018.
- [10] L. Xiao, Y. Zhang, J. Dai, J. Li, and W. Li, "New noise-tolerant ZNN models with predefined-time convergence for time-variant Sylvester equation solving," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Aug. 5, 2019, doi: [10.1109/TSMC.2019.2930646](https://doi.org/10.1109/TSMC.2019.2930646).
- [11] J. Jin, L. Xiao, M. Lu, and J. Li, "Design and analysis of two FTRNN models with application to time-varying Sylvester equation," *IEEE Access*, vol. 7, pp. 58945–58950, 2019.
- [12] 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.
- [13] B. Liao and Y. Zhang, "Different complex ZFs leading to different complex ZNN models for time-varying complex generalized inverse matrices," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 9, pp. 1621–1631, Sep. 2014.
- [14] M. Hajarian, "Matrix iterative methods for solving the Sylvester-transpose and periodic Sylvester matrix equations," *J. Frankl. Inst. Eng. Appl. Math.*, vol. 350, no. 10, pp. 3328–3341, 2013.
- [15] M. Dehghan and A. Shirilord, "A generalized modified Hermitian and skew-Hermitian splitting (GMHSS) method for solving complex Sylvester matrix equation," *Appl. Math. Comput.*, vol. 348, pp. 632–651, May 2019.
- [16] X. Zhang and X. Sheng, "The relaxed gradient based iterative algorithm for the symmetric (Skew symmetric) solution of the Sylvester equation," *Math. Problem Eng.*, vol. 2017, no. 2, 2017, Art. no. 1624969.
- [17] Y. Shi and Y. Zhang, "New discrete-time models of zeroing neural network solving systems of time-variant linear and nonlinear inequalities," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 2, pp. 565–576, Feb. 2020.
- [18] L. Wei, L. Jin, C. Yang, K. Chen, and W. Li, "New noise-tolerant neural algorithms for future dynamic nonlinear optimization with estimation on Hessian matrix inversion," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, May 27, 2019, doi: [10.1109/TSMC.2019.2916892](https://doi.org/10.1109/TSMC.2019.2916892).

- [19] 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 Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5767–5776, Nov. 2018.
- [20] J. Li, Y. Zhang, and M. Mao, "Continuous and discrete zeroing neural network for different-level dynamic linear system with robot manipulator control," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 11, pp. 4633–4642, Nov. 2020.
- [21] 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," *Neurocomputing*, vol. 142, pp. 165–173, Oct. 2014.
- [22] J. Li, Y. Zhang, and M. Mao, "Five-instant type discrete-time ZND solving discrete time-varying linear system, division and quadratic programming," *Neurocomputing*, vol. 331, pp. 323–335, Feb. 2019.
- [23] Z. Zhang and L. Zheng, "A complex varying-parameter convergent-differential neural-network for solving online time-varying complex Sylvester equation," *IEEE Trans. Cybern.*, vol. 49, no. 10, pp. 3627–3639, Oct. 2019.
- [24] A. Graham, *Kronecker Products and Matrix Calculus with Applications*. New York, NY, USA: Courier Dover Publ., 2018.
- [25] L. Jin, Y. Zhang, S. Li, and Y. Zhang, "Modified ZNN for time-varying quadratic programming with inherent tolerance to noises and its application to kinematic redundancy resolution of robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 6978–6988, Nov. 2016.
- [26] 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 Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2615–2627, Dec. 2016.
- [27] W. Li, "Design and analysis of a novel finite-time convergent and noise-tolerant recurrent neural network for time-variant matrix inversion," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 11, pp. 4362–4376, Nov. 2020.
- [28] Y. Zhang, Y. Wang, L. Jin, B. Mu, and H. Zheng, "Different ZFs leading to various ZNN models illustrated via online solution of time-varying underdetermined systems of linear equations with robotic application," in *Proc. Int. Symp. Neural Netw.*, 2013, pp. 481–488.
- [29] D. Griffiths and D. Higham, *Numerical Methods for Ordinary Differential Equations: Initial Value Problems*. London, U.K.: Springer, 2010.
- [30] J. H. Mathews and K. D. Fink, *Numerical Methods Using MATLAB*, vol. 4. Upper Saddle River, NJ, USA: Prentice Hall, 2004.
- [31] K. Diethelm, N. J. Ford, and A. D. Freed, "Detailed error analysis for a fractional Adams method," *Numer. Algorithms*, vol. 36, no. 1, pp. 31–52, 2004.
- [32] D. Guo, X. Lin, Z. Su, S. Sun, and Z. Huang, "Design and analysis of two discrete-time ZD algorithms for time-varying nonlinear minimization," *Numer. Algorithms*, vol. 77, no. 1, pp. 23–36, 2018.
- [33] D. Chen, S. Li, W. Li, and Q. Wu, "A multi-level simultaneous minimization scheme applied to jerk-bounded redundant robot manipulators," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 463–474, Jan. 2020.
- [34] D. Chen, S. Li, Q. Wu, and X. Luo, "New disturbance rejection constraint for redundant robot manipulators: An optimization perspective," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2221–2232, Apr. 2020.
- [35] D. Chen, S. Li, F.-J. Lin, and Q. Wu, "New super-twisting zeroing neural-dynamics model for tracking control of parallel robots: A finite-time and robust solution," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2651–2660, Jun. 2020.



Zeshan Hu received the B.S. degree in computer science and technology from Fuzhou University, Fuzhou, China, in 2018. He is currently pursuing the master's degree in computer science and technology with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China.

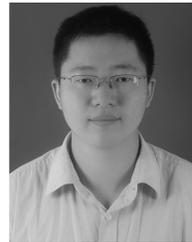
His main research interests include neural networks and robotics.



Kenli Li (Senior Member, IEEE) received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2003.

He was a Visiting Scholar with the University of Illinois at Urbana-Champaign, Champaign, IL, USA, from 2004 to 2005. He is currently a Full Professor of Computer Science and Technology with Hunan University, Changsha, China, and also the Deputy Director of the National Supercomputing Center, Changsha. His current research interests

include parallel computing, cloud computing, big data computing, and neural computing.



Lin Xiao received the B.S. degree in electronic information science and technology from Hengyang Normal University, Hengyang, China, in 2009, and the Ph.D. degree in communication and information systems from Sun Yat-sen University, Guangzhou, China, in 2014.

He is currently a Professor with the College of Information Science and Engineering, Hunan Normal University, Changsha, China. He has authored over 100 papers in international conferences and journals. His main research interests include neural networks and robotics.



Yaonan Wang received the B.S. degree in computer engineering from the East China Science and Technology University, Fuzhou, China, in 1981, and the M.S. and Ph.D. degrees in electrical engineering from Hunan University, Changsha, China, in 1990 and 1994, respectively.

He has been a Professor with Hunan University since 1995. He is an Academician of the Chinese Academy of Engineering, Beijing, China. His current research interests include robotics, intelligent perception and control, and computer vision for industrial applications.

Mingxing Duan, photograph and biography not available at the time of publication.



Keqin Li (Fellow, IEEE) received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1985, and the Ph.D. degree in computer science from the University of Houston, Texas, USA, in 1990.

He is a SUNY Distinguished Professor of Computer Science with the State University of New York, New Paltz, NY, USA. He is also a Distinguished Professor with Hunan University, Changsha, China. He has authored or coauthored more than 760 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. His current research interests include cloud computing, fog computing and mobile-edge computing, and energy-efficient computing and communication.

Dr. Li is currently an Associate Editor of the *ACM Computing Surveys*.