IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 22, 2020, accepted December 29, 2020, date of publication January 5, 2021, date of current version January 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3049244

Discrete-Time Zeroing Dynamics With Quadruplicate Error Pattern for Time-Varying Linear Inequality

JIANHUANG CAI, QINGSHAN FENG, AND DONGSHENG GUO, (Member, IEEE)
College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China

Corresponding author: Dongsheng Guo (gdongsh@hqu.edu.cn)

This work was supported in part by the Training Program for Outstanding Young Scientific Talents in Fujian Province University under Grant 50X19020, in part by the Quanzhou City Science and Technology Program of China under Grant 2018C111R, and in part by the Subsidized Project for Cultivating Postgraduates' Innovative Ability in Scientific Research of Huaqiao University under Grant 18013082037.

ABSTRACT Linear inequality (LI) plays an important role in many fields of science and engineering. Recently, a typical neural dynamics called zeroing dynamics (ZD) has been reported for online solution of time-varying LI (TVLI). On the basis of the previous work, the discrete-time form of the ZD with superior computational property is studied in this paper. Specifically, a Taylor-type difference rule is first presented for the first-order derivative approximation. By utilizing such a difference rule to discrete the previous ZD model, the new discrete-time ZD (DTZD) algorithm is thus established and proposed for TVLI solving. Such an algorithm performs better computational performance than the existing DTZD algorithm. Theoretical results show that the proposed DTZD algorithm has a quadruplicate error pattern on solving the TVLI. Comparative numerical results with two illustrative examples further substantiate the efficacy and superiority of the proposed DTZD algorithm over the existing DTZD algorithm.

INDEX TERMS Zeroing dynamics (ZD), discrete-time algorithm, time-varying linear inequality (TVLI), difference rule, numerical validation.

I. INTRODUCTION

In recent years, linear inequality (LI) has attracted considerable attention in different scientific and engineering fields. It has been viewed as a powerful formulation and design technique for many problems [1]–[7]. In mathematics, the linear inequality problem is generally formulated as follows:

$$Ax \leqslant b, \tag{1}$$

where coefficient matrix $A \in R^{n\times n}$ and vector $b \in R^n$ are constant, and $x \in R^n$ is the unknown vector to be obtained.

To solve (1), many numerical algorithms and neural networks have been developed and investigated [4]–[13]. For example, in [8], different iterative methods were presented by Yang *et al.* to solve the LI system. In [9], three continuous-time neural networks were developed by Cichocki and Bargiela for LI solving. In [10], the implementation of a relaxation-projection algorithm by artificial neural networks to solve sets of linear inequalities was investigated by Labonte. Note that these approaches are designed intrinsically to solve the static LI, i.e., in the form of (1). As many systems in practical applications always change as time evolves, the corresponding LI should be the time-varying one with coefficients being time-varying [i.e., $A(t)$ and $b(t)$ with $t \geqslant 0$]. When the aforementioned approaches are exploited directly to solve the time-varying LI (TVLI), they may be less effective [15]–[20].

Aiming at solving TVLI, a typical neural dynamics called Zhang dynamics (ZD) has been developed and investigated by Zhang *et al.* [15]–[20]. Especially, in [20], by introducing a nonnegative vector, the TVLI was converted to a matrix-vector equation. Then, by exploiting the exponent-type design formula, the continuous-time ZD (CTZD) model depicted in an explicit dynamics was developed. Its efficacy was substantiated via theoretical and simulation results. In [21], the discrete-time form of this CTZD model, namely discrete-time ZD (DTZD) algorithm, was further studied by utilizing the Euler-type difference rule [22]. Both theoretical

The associate editor coordinating the review of this manuscript and approving it for publication was Xiao-Jun Yang.

and numerical results indicated that such a DTZD algorithm has an $O(\tau^2)$ error pattern when solving TVLI, where τ denotes the sampling gap. That is, the computational error in the steady-state reduces by 100 times, as the value of τ decreases by 10 times.

Recently, a new Taylor-type difference rule has been constructed in [23], which has been proven to has a smaller truncation error than the Euler-type difference rule [22] for the first-order derivative approximation, i.e., $O(\tau^3)$ versus $O(\tau)$. Moreover, the results shown in [23] have revealed that the Taylor-type difference rule can achieve better performance on CTZD discretization, as compared with the Euler-type difference rule. Thus, in this paper, by utilizing the Taylor-type difference rule, the new discrete-time ZD (DTZD) algorithm is proposed for TVLI solving. The theoretical and numerical results are presented to substantiate the efficacy and superiority of the proposed DTZD algorithm. These results further indicate that the computational error in the steady-state for the proposed algorithm is in the order $O(\tau^4)$. That is, decreasing the value of τ by 10 times leads to reduction of the computational error in the steady-state by 10000 times. In this sense, the proposed DTZD algorithm is superior to the previous DTZD algorithm for TVLI solving. It is worth pointing out that this paper is the first attempt to provide a numerical algorithm with $O(\tau^4)$ error pattern for generating a time-varying solution to the TVLI. This paper presents a valuable insight into the development of the new numerical algorithm with high precision for solving time-varying linear and/or nonlinear inequality.

The rest of this paper is organized into four sections. Section II shows the previous results for TVLI solving. In section III, the DTZD algorithm proposed in this paper is presented, together with theoretical results. Section IV provides comparative numerical results that are synthesized using the proposed DTZD algorithm and the existing DTZD algorithm in [21]. Section V concludes this paper final remark.

II. TIME-VARYING LINEAR INEQUALITY

In this section, the TVLI problem is formulated. Then, the CTZD model and the DTZD algorithm in the previous work [19]–[21] are presented for TVLI solving.

A. PROBLEM FORMULATION

The following problem of TVLI is studied in this paper:

$$A(t)x(t) \leqslant b(t), \tag{2}$$

where $A(t) \in R^{n\times n}$ and $b(t) \in R^n$ are smoothly time-varying coefficients, and $x(t) \in R^n$ is the unknown vector to be obtained. In this paper, a feasible solution $x(t)$ needs to be determined to make (2) hold true for any $t_k = k\tau$ with $k = 0, 1, 2, \cdots$. To guarantee the existence of $x(t)$, $A(t)$ (2) is assumed to be nonsingular at any $t_k = k\tau$.

In [19], [20], by introducing a time-varying nonnegative vector $\Lambda(t) \in R^n$, the TVLI (2) is transformed into the following matrix-vector equation:

$$A(t)x(t) - b(t) + \Lambda(t) = 0, \tag{3}$$

where $\Lambda(t) = y(t) \odot y(t)$ with the multiplication operator $\odot$ denoting the Hadamard product [19], [20]. Note that the time-varying vector $y(t) \in R^n$ also needs to be automatically determined during the solving process of (3).

B. CTZD MODEL

To solve the TVLI (2), on the basis of (3), the error function is defined as $e(t) = A(t)x(t) - b(t) + \Lambda(t) = A(t)x(t) - b(t) + D(t)y(t) \in R^n$, where the diagonal matrix $D(t)$ denotes $D(t) = \mathrm{diag}\{y_1(t), y_2(t), \cdots, y_n(t)\} \in R^{n\times n}$. To make this error function converge to zero, the following exponent-type design formula is exploited [24]:

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

where $\dot{e}(t)$ denotes the time derivative of $e(t)$, and $\gamma > 0 \in R$ is the design parameter that affects the solution convergence. By expanding (4), the following computational model is obtained:

$$\begin{aligned} A(t)\dot{x}(t) + 2D(t)\dot{y}(t) = &-\dot{A}(t)x(t) + \dot{b}(t) \\ &-\gamma(A(t)x(t) - b(t) + D(t)y(t)), \end{aligned} \tag{5}$$

where $\dot{x}(t)$, $\dot{y}(t)$, $\dot{A}(t)$, and $\dot{b}(t)$ denote the time derivatives of $x(t)$, $y(t)$, $A(t)$, and $b(t)$, respectively. By defining $u(t) = [x^{\mathrm{T}}(t), y^{\mathrm{T}}(t)]^{\mathrm{T}} \in R^{2n}$ with superscript $^{\mathrm{T}}$ denoting the transpose operator, (5) is reformulated as follows:

$$\dot{u}(t) = W^{\dagger}(t)(P(t)u(t) + \dot{b}(t)) - \gamma W^{\dagger}(t)(Q(t)u(t) - b(t)), \tag{6}$$

where $\dot{u}(t) \in R^{2n}$ denotes the time derivative of $u(t)$, $W(t) = [A(t), 2D(t)] \in R^{n\times 2n}$, $P(t) = [-\dot{A}(t), 0] \in R^{n\times 2n}$, and $Q(t) = [A(t), D(t)] \in R^{n\times 2n}$. In addition, $W^{\dagger}(t)$ denotes the pseudoinverse of $W(t)$ [25]. (6) is the CTZD model presented in [20] to solve the TVLI (2), of which the neuron structure is shown in Fig. 1. Both theoretical and simulation results indicate the efficacy of (6).

C. DTZD ALGORITHM

For the purposes of potential hardware implementation and numerical algorithm development [22], [23], [26], [27], the discrete-time form of the CTZD model (6) studied in [21].

Specifically, by utilizing the Euler-type difference rule [22] discretize (6), the following DTZD algorithm for solving the TVLI (2) is obtained [21]:

$$u_{k+1} = u_k + \tau W_k^{\dagger}(P_k u_k + \dot{b}_k) - hW_k^{\dagger}(Q_k u_k - b_k), \tag{7}$$

where $u_k = u(t_k = k\tau)$, $W_k^{\dagger} = W^{\dagger}(t_k = k\tau)$, $P_k = P(t_k = k\tau)$, $Q_k = Q(t_k = k\tau)$, $b_k = b(t_k = k\tau)$, and $\dot{b}_k = \dot{b}(t_k = k\tau)$. In addition, $h = \gamma\tau > 0 \in R$ denotes the step size, $\tau > 0 \in R$ denotes the sampling gap, and $k = 0, 1, 2, \cdots$ denotes the iteration number. As demonstrated in [21], given an initial state u_0 for the DTZD algorithm (7),

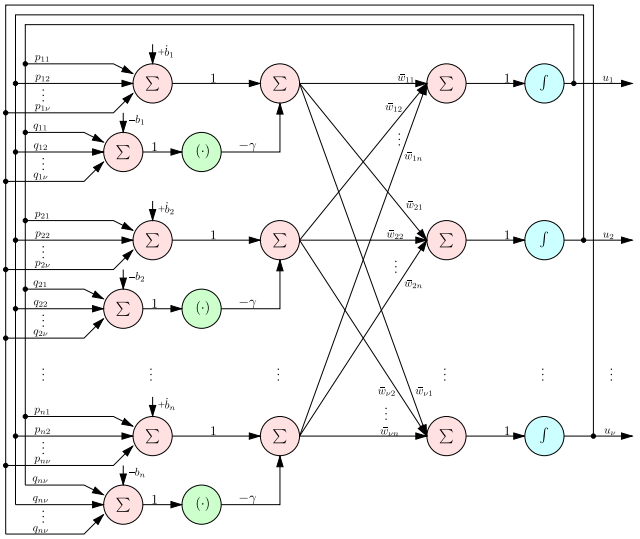

FIGURE 1. Neuron structure of in the CTZD model (6) to solve the TVLI (2), where $\nu = 2n$, and u_i is the ith (with $i = 1, 2, \cdots, \nu$) neuron element of (6). The weights $\bar{w}_{ik}$, p_{kj}, and q_{kj} represent the ikth (with $k = 1, 2, \cdots, n$) element of $W^{\dagger}(t)$, the kjth element of $P(t)$, and the kjth element of $Q(t)$. The thresholds $\dot{b}_k$ and b_k represent the kth elements of $\dot{b}(t)$ and $b(t)$.

its computational error is convergent, which combines with a small the steady-state error, thereby showing the effectiveness of (7). The theoretical and numerical results presented in [21] also indicate that the DTZD algorithm (7) has an $O(\tau^2)$ error pattern when solving (2).

III. NEW DTZD ALGORITHM

In this section, by utilizing the Taylor-type difference rule [23], a new DTZD algorithm with superior computational performance is developed for TVLI solving.

Proposition: The Taylor-type difference rule utilized in this paper is formulated as follows [23]:

$$\dot{u}_k \approx \frac{24u_{k+1} - 5u_k - 12u_{k-1} - 6u_{k-2} - 4u_{k-3} + 3u_{k-4}}{48\tau}, \tag{8}$$

where $\tau = t_{k+1} - t_k = t_k - t_{k-1} = t_{k-1} - t_{k-2} = t_{k-2} - t_{k-3} = t_{k-3} - t_{k-4}$, *and* $k = 4, 5, 6, \cdots$.

By using (8) to discretize the CTZD model (6), the new DTZD algorithm proposed in this paper for solving the TVLI (2) is thus obtained as follows:

$$\begin{aligned} u_{k+1} = {} & \frac{5}{24}u_k + \frac{1}{2}u_{k-1} + \frac{1}{4}u_{k-2} + \frac{1}{6}u_{k-3} - \frac{1}{8}u_{k-4} \\ & + 2\tau W_k^{\dagger}(P_k u_k + \dot{b}_k) - hW_k^{\dagger}(Q_k u_k - b_k), \end{aligned} \tag{9}$$

where the step sized $h = 2\gamma\tau > 0 \in R$. Comparing the proposed DTZD algorithm (9) with the previous DTZD algorithm (7), we can find that the former is a five-step iteration, whereas the latter is a one-step iteration. The difference on structures for such two algorithms leads to different computational performances on TVLI solving (which will be presented in Section IV).

For the proposed DTZD algorithm (9), five initial states (i.e., u_0, u_1, u_2, u_3, and u_4) are needed. To initiate the iterative computation of (9), the previous DTZD algorithm (7) is used. Specifically, given an initial state u_0, the other four initial states for (9) are obtained through the following iterations [i.e., the first to fourth iterations of (7)]:

$$\begin{cases} u_1 = u_0 + \tau W_0^{\dagger}(P_0 u_0 + \dot{b}_0) - hW_0^{\dagger}(Q_0 u_0 - b_0), \\ u_2 = u_1 + \tau W_1^{\dagger}(P_1 u_1 + \dot{b}_1) - hW_k^{\dagger}(Q_1 u_1 - b_1), \\ u_3 = u_2 + \tau W_2^{\dagger}(P_2 u_2 + \dot{b}_2) - hW_k^{\dagger}(Q_2 u_2 - b_2), \\ u_4 = u_3 + \tau W_3^{\dagger}(P_3 u_3 + \dot{b}_3) - hW_k^{\dagger}(Q_3 u_3 - b_3). \end{cases}$$

Then, the procedure of the proposed DTZD algorithm (9) to solve the TVLI (2) is provided as follows.

i. Initialization:

Given time duration T, sampling gap τ, and step size h.
Initialize t_0, u_0, A_0, $\dot{A}_0$, b_0, and $\dot{b}_0$.
Receive $W_0^{\dagger}$, P_0, and Q_0.
Compute $\|e_0\|_2 = \|Q_0 u_0 - b_0\|_2$.

ii. First Loop (with $k = 0, 1, 2, 3$):

Compute u_{k+1} through (7).
Receive $W_{k+1}^{\dagger}$, P_{k+1}, and Q_{k+1}.
Compute $\|e_{k+1}\|_2 = \|Q_{k+1}u_{k+1} - b_{k+1}\|_2$.

iii. Second Loop (with $k = 4, \cdots,$ int$(T)/\tau$):

Compute u_{k+1} through (9).
Receive $W_{k+1}^{\dagger}$, P_{k+1}, and Q_{k+1}.
Compute $\|e_{k+1}\|_2 = \|Q_{k+1}u_{k+1} - b_{k+1}\|_2$.

iv. Output:

Save u_k and $\|e_k\|_2$, and plot figures.

With regard to the proposed DTZD algorithm (9), its accuracy can be measured by the computational error $\|e_k\|_2 = \|Q_k u_k - b_k\|_2$. Evidently, $\|e_k\|_2 = 0$ yields $Q_k u_k - b_k = A_k x_k - b_k + D_k y_k = 0$. As defined in Section II, $-D_k y_k \leqslant 0$, then $A_k x_k - b_k = -D_k y_k \leqslant 0$, and further $A_k x_k \leqslant b_k$. Evidently, if the computational error of (9) possesses the convergence characteristic and the steady-state error (i.e., $\|e_k\|_2$ with a large k) is small enough, then it can be concluded that the solution of x_k computed by (9) is exactly a time-varying solution of the TVLI (2). Besides, the following theoretical results on the computational property of the proposed DTZD algorithm (9) are given.

Lemma 1: The proposed DTZD algorithm (9) is a convergent method.

Proof: It can be analyzed via zero-stability and consistency [28]. □

Lemma 2: Consider a solvable TVLI (2). The computational error of the proposed DTZD algorithm (9) in the steady-state is of order $O(\tau^4)$.

Proof: It can be generalized from the previous work [23]. □

In sum, these results have theoretically guaranteed the superior computational performance of the proposed DTZD algorithm (9) for TVLI solving.

IV. COMPARATIVE NUMERICAL RESULTS

In this section, numerical experiments with two examples are conducted to substantiate the efficacy and superiority of

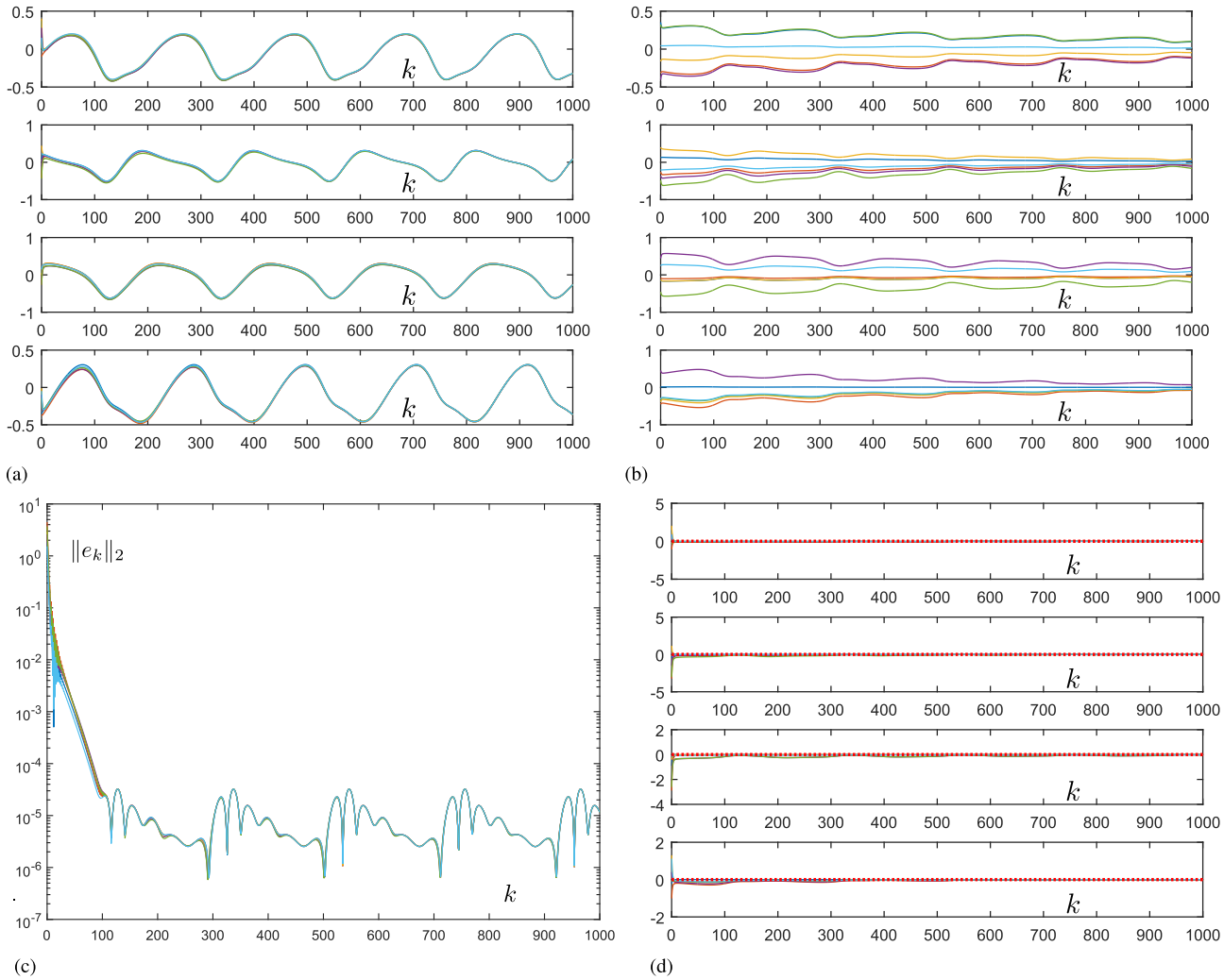

FIGURE 2. Numerical results synthesized using the proposed DTZD algorithm (9) with $h = 0.5$ and $\tau = 0.01$ to solve (2). (a) State trajectories of $x_k \in R^4$ with $t_k = k\tau$. (b) State trajectories of $y_k \in R^4$ with $t_k = k\tau$. (c) Computational errors. (d) Testing errors $\varepsilon_k = A_k x_k - b_k \in R^4$.

the proposed DTZD algorithm (9) for solving the TVLI (2), as compared with the previous DTZD algorithm (7)

Example 1: In this example, we consider the TVLI (2) with the following coefficients:

$$A(t) = \begin{bmatrix} 4+\sin(3t) & \cos(3t) & \cos(3t)/2 & \cos(3t)/3 \\ \cos(3t) & 4+\sin(3t) & \cos(3t) & \cos(3t)/2 \\ \cos(3t)/2 & \cos(3t) & 4+\sin(3t) & \cos(3t) \\ \cos(3t)/3 & \cos(3t)/2 & \cos(3t) & 4+\sin(3t) \end{bmatrix},$$

$$b(t) = \begin{bmatrix} \sin(3t) \\ \cos(3t) \\ \sin(3t)+\cos(3t) \\ \sin(3t)-\cos(3t) \end{bmatrix}.$$

To solve this TVLI, the proposed DTZD algorithm (9), of which the state vector is $u_k = [x_k^{\mathrm{T}}, y_k^{\mathrm{T}}]^{\mathrm{T}} \in R^8$, is exploited. The related results are given in Figs. 2–4 and Table 1.

Fig. 2 shows the numerical results synthesized using the proposed DTZD algorithm (9) with $h = 0.5$ and $\tau = 0.01$. As seen from Fig. 2(a) and (b), starting from six randomly-generated initial states, the state trajectories of x_k [being the first 4 elements of u_k in (9)] and y_k [being the rest elements of u_k] always change with time $t_k = k\tau$. From Fig. 2(c), we can observe that the computational errors of (9) possess the characteristic of convergence, where $\|e_k\|_2$ is calculated as $\|e_k\|_2 = \|A_k x_k - b_k + D_k y_k\|_2$. In addition, Fig. 2(c) shows that the corresponding computational errors in the steady-state (i.e., $\|e_k\|_2$ with a large k) are small enough with the order being 10^{-5}. This statement means that x_k and y_k in Fig. 2(a) and (b) satisfy (3), which further means that the solution of x_k computed by (9) is exactly a time-varying solution to the TVLI (2). For better understanding, a testing error is defined as $\varepsilon_k = A_k x_k - b_k$, and Fig. 2(d) shows its profiles. As seen from Fig. 2(d), all elements of ε_k are less than or equal to zero. That is, $\varepsilon_k = A_k x_k - b_k \leqslant 0$, and then $A_k x_k \leqslant b_k$, thereby indicating that the x_k solution in Fig. 2(a) is an exact solution to (2). Evidently, these numerical results substantiate the efficacy of the proposed DTZD algorithm (9) for TVLI solving.

TABLE 1. Steady-state computational errors of the previous DTZD algorithm (7) and the proposed DTZD algorithm (9) using different values of h and τ to solve the TVLI (2).

#	h	$\tau = 0.01$	$\tau = 0.001$	manner
DTZD algorithm (7)	0.2	$\leqslant 8.393 \times 10^{-3}$	$\leqslant 8.884 \times 10^{-5}$	$O(\tau^2)$
	0.3	$\leqslant 5.785 \times 10^{-3}$	$\leqslant 5.925 \times 10^{-5}$	
	0.4	$\leqslant 4.391 \times 10^{-3}$	$\leqslant 4.446 \times 10^{-5}$	
	0.5	$\leqslant 3.530 \times 10^{-3}$	$\leqslant 3.557 \times 10^{-5}$	
	0.6	$\leqslant 2.951 \times 10^{-3}$	$\leqslant 2.964 \times 10^{-5}$	
DTZD algorithm (9)	0.2	$\leqslant 5.480 \times 10^{-5}$	$\leqslant 9.204 \times 10^{-9}$	$O(\tau^4)$
	0.3	$\leqslant 4.550 \times 10^{-5}$	$\leqslant 6.172 \times 10^{-9}$	
	0.4	$\leqslant 3.801 \times 10^{-5}$	$\leqslant 4.639 \times 10^{-9}$	
	0.5	$\leqslant 3.233 \times 10^{-5}$	$\leqslant 3.714 \times 10^{-9}$	
	0.6	$\leqslant 2.797 \times 10^{-5}$	$\leqslant 3.097 \times 10^{-9}$	

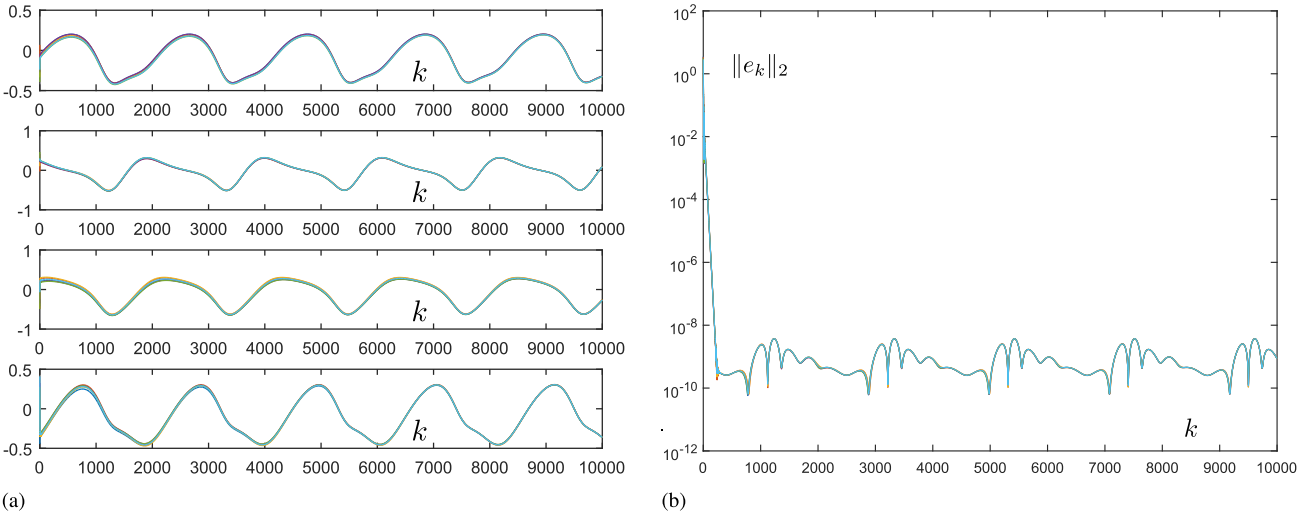

FIGURE 3. Numerical results synthesized using the proposed DTZD algorithm (9) with $h = 0.5$ and $\tau = 0.001$ to solve (2). (a) State trajectories of $x_k \in R^4$ with $t_k = k\tau$. (b) Computational errors.

By decreasing the value of τ (i.e., from 0.01 to 0.001), the proposed DTZD algorithm (9) is tested, and the results are presented in Fig. 3. As seen from Fig. 3(a), the state trajectories of x_k, starting from six randomly-generated initial states, are always changing. In addition, Fig. 3(b) shows that the computational errors of (9) also possess the convergence characteristic, and the corresponding steady-state errors are much smaller [in comparison with those in Fig. 2(c)] and are in the order 10^{-9}. These results demonstrate again that the proposed DTZD algorithm (9) is effective on solving the TVLI (2). More importantly, comparing Fig. 2(c) with Fig. 3(b), we can obtain that (9) exhibits excellent computational performance by fixing h and decreasing τ. That is, as the τ value decreases by 10 times, the computational error of (9) in the steady-state reduces by 10000 times. Thus, τ should be set to be small enough for the proposed DTZD algorithm (9) to achieve the precision requirement.

For further investigation and comparison, we investigate the previous DTZD algorithm (7) and the proposed DTZD algorithm (9) with the same initial state (i.e., $u_0 = \mathbf{0.1} \in R^8$) and using different values of h and τ to solve the TVLI (2). The corresponding numerical results are presented in Table 1 and Fig. 4, which indicate that such two DTZD algorithms are effective on solving (2) in view of a small computational error. In addition, as shown in Fig. 4, in the same condition, the proposed DTZD algorithm (9) has a better computational property than the previous DTZD algorithm (7) for TVLI solving. Moreover, the following results are summarized from Table 1 and Fig. 4.

- As to the previous DTZD algorithm (7), its computational error in the steady-state changes in an $O(\tau^2)$ manner; i.e., decreasing τ decreases by 10 times leads to the reduction of the steady-state computational error by 100 times.
- As to the proposed DTZD algorithm (9), its computational error in the steady-state changes in an $O(\tau^4)$ manner; i.e., decreasing τ decreases by 10 times leads to the reduction of the steady-state computational error by 10000 times.

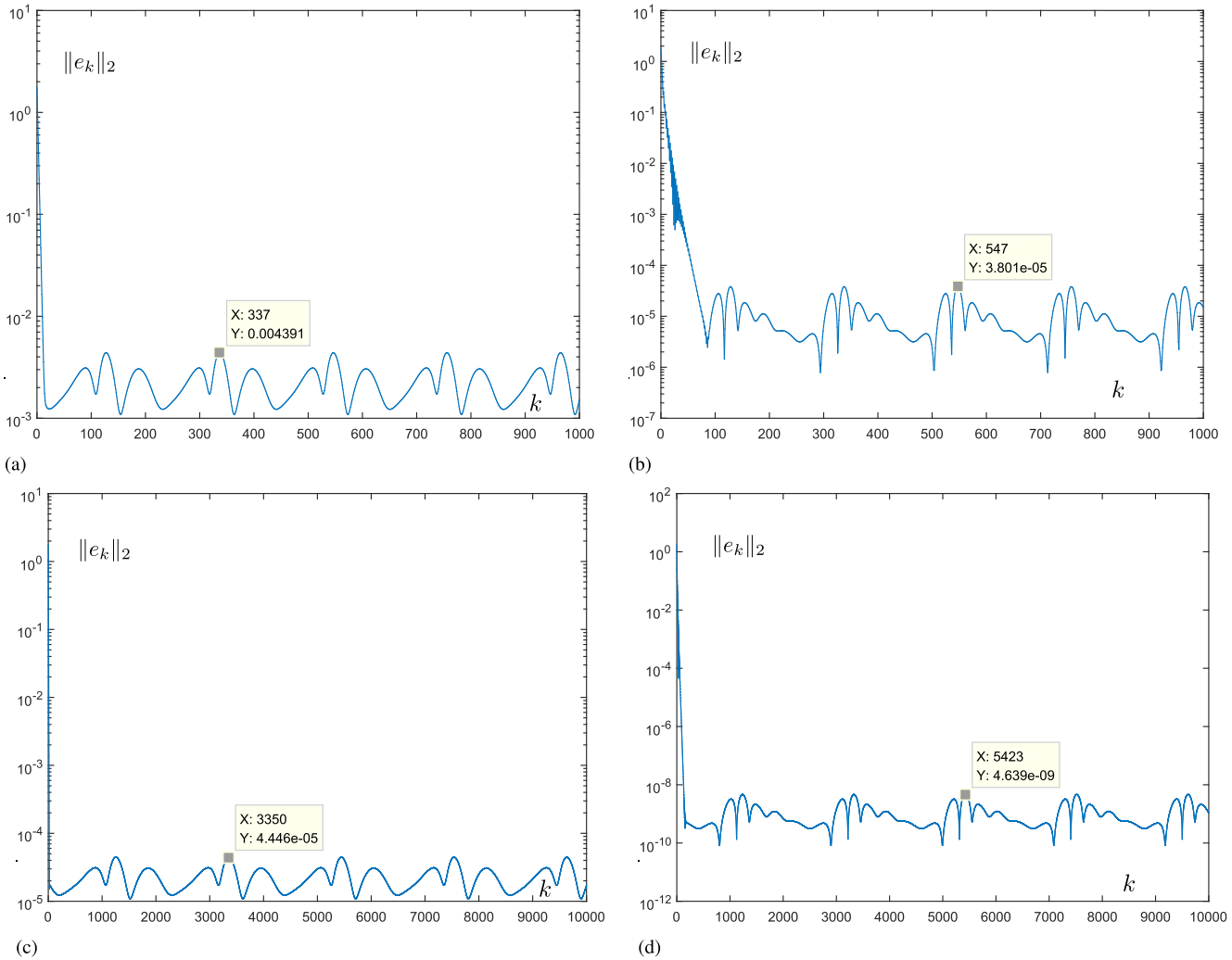

FIGURE 4. Computational errors of the previous DTZD algorithm (7) and the proposed DTZD algorithm (9) with $h = 0.4$ and using different values of τ to solve (2). (a) Via (7) with $\tau = 0.01$. (b) Via (9) with $\tau = 0.01$. (c) Via (7) with $\tau = 0.001$. (d) Via (9) with $\tau = 0.001$.

- As to such two DTZD algorithms, their computational performances can be further improved by increasing the value of h.

On the basis of these results, it can be concluded that the computational performance of (9) is improved more effectively than that of (7) by decreasing the value of τ. In this sense, the proposed DTZD algorithm (9) is superior to the previous DTZD algorithm (7).

In sum, the above numerical results (i.e., Figs. 2–4 and Table 1) have substantiated the efficacy and superiority of the proposed DTZD algorithm (9) for TVLI solving.

Example 2: In this example, the proposed DTZD algorithm (9) is extended to solving the following time-varying linear matrix inequality (TVLMI) [14]:

$$M(t)X(t)N(t) \leqslant C(t), \tag{10}$$

where $M(t) \in R^{m\times m}$, $N(t) \in R^{n\times n}$, and $C(t) \in R^{m\times n}$ are smoothly time-varying matrices, and $X(t) \in R^{m\times n}$ is the unknown matrix to be obtained. Note that the above inequality "$\leqslant$" is satisfied, if and only if each element of the left matrix is less than or equal to that of the right matrix. On the basis of the Kronecker-product and vectorization techniques [14], [25], online solution of the TVLMI (10) is equivalent to solving the TVLI as follows:

$$(N^{\mathrm{T}}(t) \otimes M(t))\mathrm{vec}(X(t)) \leqslant \mathrm{vec}(C(t)), \tag{11}$$

where $\otimes$ denotes the Kronecker product and $\mathrm{vec}(\cdot)$ denotes the vectorization operator. Thus, the proposed DTZD algorithm (9) is exploited to solve the above TVLI (11) so as to achieve the purpose of solving the TVLMI (10).

In the numerical experiments, we consider the TVLMI (10) with the following coefficients:

$$M(t) = \begin{bmatrix} m_1(t) & m_2(t) & m_3(t) & m_4(t) & m_5(t) \\ m_2(t) & m_1(t) & m_2(t) & m_3(t) & m_4(t) \\ m_3(t) & m_2(t) & m_1(t) & m_2(t) & m_3(t) \\ m_4(t) & m_3(t) & m_2(t) & m_1(t) & m_2(t) \\ m_5(t) & m_4(t) & m_3(t) & m_2(t) & m_1(t) \end{bmatrix},$$

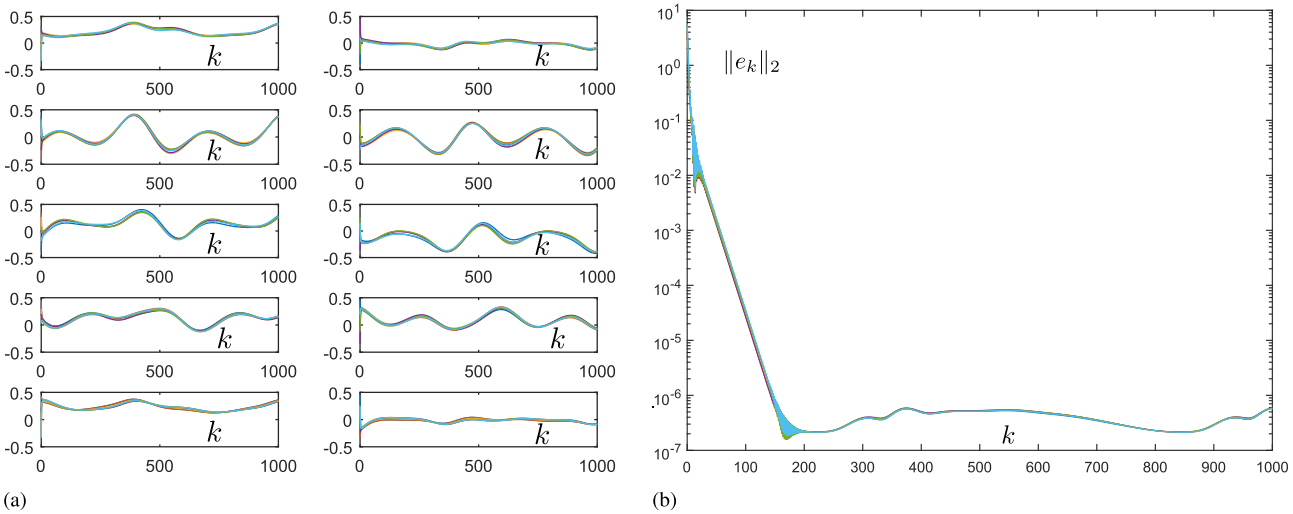

FIGURE 5. Numerical results synthesized using the proposed DTZD algorithm (9) with $h = 0.5$ and $\tau = 0.01$ to solve (10). (a) State trajectories of $X_k \in R^{5\times 2}$ with $t_k = k\tau$. (b) Computational errors.

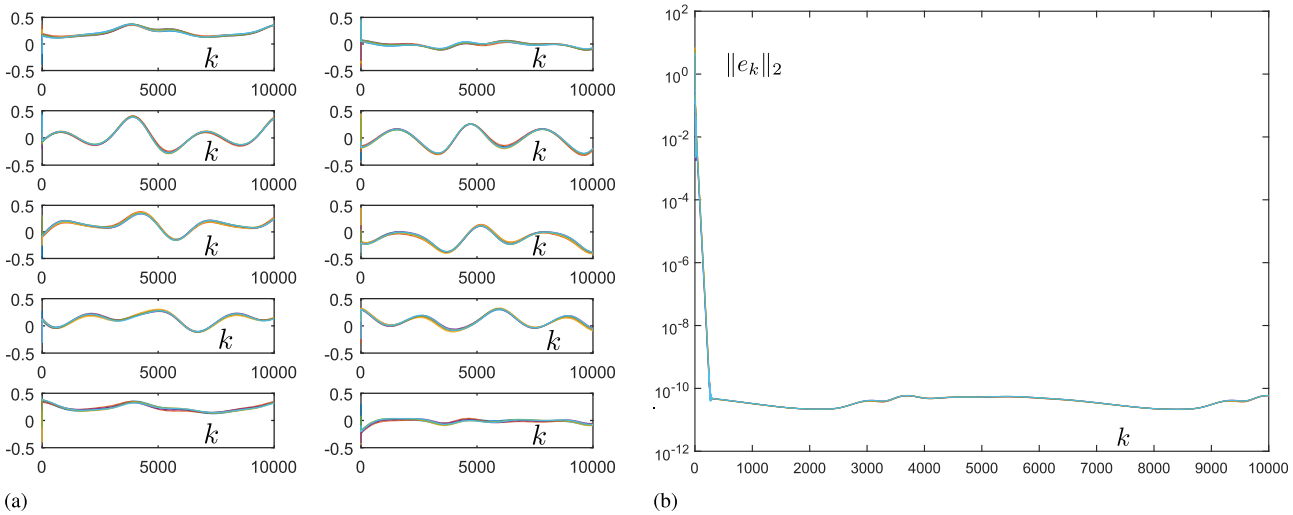

FIGURE 6. Numerical results synthesized using the proposed DTZD algorithm (9) with $h = 0.5$ and $\tau = 0.001$ to solve (10). (a) State trajectories of $X_k \in R^{5\times 2}$ with $t_k = k\tau$. (b) Computational errors.

$$N(t) = \begin{bmatrix} \sin(t) & \cos(t) \\ -\cos(t) & \sin(t) \end{bmatrix},$$

$$C(t) = \begin{bmatrix} \sin(t) & \cos(t) \\ \cos(t) & \sin(t) \\ \sin(t)+\cos(t) & \sin(t)\cos(t) \\ \sin(t)-\cos(t) & \exp(-t) \\ \sin(t)+\exp(-t) & \cos(t)+\exp(-t) \end{bmatrix},$$

where $m_1(t) = 5 + \sin(t)$, and $m_k(t) = \cos(t)/(k-1)$ with $k = 2, 3, 4, 5$. The related numerical results synthesized using the proposed DTZD algorithm (9) are presented in Figs. 5 and 6.

Specifically, Fig. 5(a) shows that, as synthesized by (9) with $h = 0.4$ and $\tau = 0.01$, the state trajectories of X_k are changing. Fig. 5(b) shows that the computational errors of (9) exhibit the convergence characteristic, where $\|e_k\|_2$ is calculated as $\|e_k\|_2 = \|(N_k^{\mathrm{T}} \otimes M_k)\mathrm{vec}(X_k) - \mathrm{vec}(C_k) + D_k y_k\|_2$. Fig. 5(b) also indicates that the computational errors in the steady-state are small and are in the order 10^{-7}. This statement means that the solution of X_k computed by (9) is exactly a time-varying solution to (10), i.e., $M_k X_k N_k \leqslant C_k$. These numerical results substantiate that the proposed DTZD algorithm (9) is effective on solving the TVLMI (10).

With the value of h fixed, the proposed DTZD algorithm (9) is tested by decreasing the value of τ. Fig. 6 shows the corresponding results, which demonstrate again the efficacy of (9) for TVLMI solving (in terms of a small computational error). In addition, it follows from Fig. 5(b) and Fig. 6(b) that the computational error of (9) in the steady-state decreases from 10^{-7} to 10^{-11} as τ decreases from 0.01 to 0.001. That is, when solving the TVLMI (10), the computational performance of (9) is still improved by decreasing the value of τ.

Thus, it can be concluded once again that τ is an important parameter in the proposed DTZD algorithm (9) and should be set appropriately to be a small value.

In sum, the above numerical results (i.e., Figs. 5 and 6) have substantiated the efficacy of the proposed DTZD algorithm (9) for TVLMI solving [as an extension study of (9) on solving the TVLI (2)].

V. CONCLUSION

In this paper, by utilizing the Taylor-type difference rule (8), the new DTZD algorithm (9) has been proposed and investigated to solve the TVLI (2). The theoretical results have also been presented to show the computational properties of such an algorithm. Comparative numerical results with two examples have further substantiated the efficacy and superiority of the proposed DTZD algorithm (9) for TVLI solving, in comparison with the DTZD algorithm (7) in the previous work [21]. Both theoretical and numerical results have indicated that the computational error of (9) in the steady-state changes in an $O(\tau^4)$ manner, showing that its computational performance is improved effectively by decreasing the value of τ.

One future research direction involves the construction of new Taylor-type difference rules for designing more DTZD algorithms to solve the TVLI (2). Another future research direction involves the study of designing new DTZD algorithms with anti-noise capability. In a follow-up study, the proposed DTZD algorithm (9) will be further investigated by using different activation functions [24].

ACKNOWLEDGMENT

The authors would like to thank the editors and anonymous reviewers for the time and effort they spent reviewing their article as well as for their detailed and constructive comments for the article's improvement in terms of presentation and quality.

REFERENCES

[1] L. Xiao and Y. Zhang, "Dynamic design, numerical solution and effective verification of acceleration-level obstacle-avoidance scheme for robot manipulators," *Int. J. Syst. Sci.*, vol. 47, no. 4, pp. 932–945, Mar. 2016.
[2] D. Guo, Q. Feng, and J. Cai, "Acceleration-level obstacle avoidance of redundant manipulators," *IEEE Access*, vol. 7, pp. 183040–183048, 2019.
[3] Y. Zhang, M. Yang, H. Huang, M. Xiao, and H. Hu, "New discrete-solution model for solving future different-level linear inequality and equality with robot manipulator control," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 1975–1984, Apr. 2019.
[4] Y. Lei, "The inexact fixed matrix iteration for solving large linear inequalities in a least squares sense," *Numer. Algorithms*, vol. 69, no. 1, pp. 227–251, May 2015.
[5] A. Dax, "A hybrid algorithm for solving linear inequalities in a least squares sense," *Numer. Algorithms*, vol. 50, no. 2, pp. 97–114, Feb. 2009.
[6] Y. Zhang, "A set of nonlinear equations and inequalities arising in robotics and its online solution via a primal neural network," *Neurocomputing*, vol. 70, nos. 1–3, pp. 513–524, Dec. 2006.
[7] X. Hu, "Dynamic system methods for solving mixed linear matrix inequalities and linear vector inequalities and equalities," *Appl. Math. Comput.*, vol. 216, no. 4, pp. 1181–1193, Apr. 2010.
[8] K. Yang and K. G. Murty, "New iterative methods for linear inequalities," *J. Optim. Theory Appl.*, vol. 72, no. 1, pp. 163–185, 1992.
[9] A. Cichocki and A. Bargiela, "Neural networks for solving linear inequality systems," *Parallel Comput.*, vol. 22, no. 11, pp. 1455–1475, Jan. 1997.
[10] G. Labonte, "On solving systems of linear inequalities with artificial neural networks," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 590–600, May 1997.
[11] Y. Xia, J. Wang, and D. L. Hung, "Recurrent neural networks for solving linear inequalities and equations," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 46, no. 4, pp. 452–462, Apr. 1999.
[12] M. Sun and Q. Sui, "A new self-adaptive alternating direction method for variational inequality problems with linear equality and inequality constraints," *J. Appl. Math. Comput.*, vol. 37, nos. 1–2, pp. 69–84, 2011.
[13] H. Li, J. Luo, Q. Wang, "Solvability and feasibility of interval linear equations and inequalities," *Linear Algebra Appl.*, vol. 463, pp. 78–94, Dec. 2014.
[14] D. Guo and Y. Zhang, "Zhang neural network for online solution of time-varying linear matrix inequality aided with an equality conversion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 370–382, Feb. 2014.
[15] 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.
[16] Y. Zhang and D. Guo, *Zhang Functions and Various Models*. Heidelberg, Germany: Springer-Verlag, 2015.
[17] L. Xiao and Y. Zhang, "Zhang neural network versus gradient neural network for solving time-varying linear inequalities," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1676–1684, Oct. 2011.
[18] D. Guo and Y. Zhang, "A new variant of the Zhang neural network for solving online time-varying linear inequalities," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 468, no. 2144, pp. 2255–2271, 2012.
[19] L. Xiao and Y. Zhang, "Different Zhang functions resulting in different ZNN models demonstrated via time-varying linear matrix–vector inequalities solving," *Neurocomputing*, vol. 121, pp. 140–149, Dec. 2013.
[20] D. Guo and Y. Zhang, "ZNN for solving online time-varying linear matrix–vector inequality via equality conversion," *Appl. Math. Comput.*, vol. 259, pp. 327–338, May 2015.
[21] D. Guo and F. Xu, "Design and analysis of discrete algorithm for time-varying linear inequality solving," *J. Huaqiao Univ. (Natural Sci.)*, vol. 38, no. 5, pp. 732–736, 2017.
[22] J. H. Mathews and K. D. Fink, *Numerical Methods Using MATLAB*, 4th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2004.
[23] D. Guo, Z. Nie, and L. Yan, "Novel discrete-time Zhang neural network for time-varying matrix inversion," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 8, pp. 2301–2310, Aug. 2017.
[24] D. Guo, S. Li, and P. S. Stanimirovic, "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.
[25] Y. Zhang and C. Yi, *Zhang Neural Networks and Neural-Dynamic Method*. Hauppauge, NY, USA: Nova Science Publishers, 2011.
[26] 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, Jan. 2018.
[27] L. Jin, Y. Zhang, and B. Qiu, "Neural network-based discrete-time Z-type model of high accuracy in noisy environments for solving dynamic system of linear equations," *Neural Comput. Appl.*, vol. 29, no. 11, pp. 1217–1232, Jun. 2018.
[28] D. F. Griffiths and D. J. Higham, *Numerical Methods for Ordinary Differential Equations: Initial Value Problems*. London, U.K.: Springer, 2010.

JIANHUANG CAI received the B.S. degree in automation from Huaqiao University, Xiamen, China, in 2018, where he is currently pursuing the M.S. degree in control science and engineering with the College of Information Science and Engineering.

His current research interests include numerical methods and neural networks.

QINGSHAN FENG received the B.S. degree in automation from Huaqiao University, Xiamen, China, in 2018, where he is currently pursuing the M.S. degree in control science and engineering with the College of Information Science and Engineering.

His current research interests include robotics and numerical methods.

DONGSHENG GUO (Member, IEEE) received the B.S. degree in automation from Sun Yat-sen University, Guangzhou, China, in 2010, and the Ph.D. degree in communication and information systems from the School of Information Science and Technology, Sun Yat-sen University, in 2015.

He is currently an Associate Professor with the College of Information Science and Engineering, Huaqiao University, Xiamen, China, and with the Fujian Engineering Research Center of Motor Control and System Optimal Schedule, Xiamen, for cooperative research. His current research interests include numerical methods, neural networks, and robotics.