Filomat 34:15 (2020), 5103–5111
https://doi.org/10.2298/FIL2015103X

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Four Discrete-Time ZD Algorithms for Dynamic Linear Matrix-Vector Inequality Solving

Feng Xu[a], Zexin Li[a], Dongsheng Guo[a]

[a] *College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China*

Abstract. Recently, a typical neural dynamics called Zhang dynamics (ZD) has been developed for online solution of dynamic linear matrix-vector inequality. This paper show a summary result by presenting the discrete-time forms of such a ZD for dynamic linear matrix-vector inequality solving. Specifically, by exploiting four different kinds of Taylor-type difference rules, the resultant discrete-time ZD (DTZD) algorithms, which are called respectively the DTZD-I, DTZD-II, DTZD-III, and DTZD-IV algorithms, are established. These algorithms can achieve excellent computational performance in solving dynamic linear matrix-vector inequality. The theoretical and numerical results are presented to further substantiate the efficacy of the presented four DTZD algorithms.

1. Introduction

In recent years, dynamic linear matrix-vector inequality has been considered as a powerful formulation and design technique for many scientific and engineering problems [1]–[5]. Solving the dynamic linear matrix-vector inequality effectively is a significant issue in numerous fields [1]–[3], [6]–[10]. To dynamic linear matrix-vector inequality, a typical neural dynamics termed Zhang dynamics (ZD) has been studied by Zhang *et al* [6]–[9]. In [7], a variant of the exponent-type formula was designed, and the resultant continuous-time ZD (CTZD) model was developed. Differing from [7] that focuses on solving the dynamic linear matrix-vector inequality directly, [9] presented another CTZD model to solve this dynamic inequality aided with an equality conversion. For the CTZD models in [7] and [9], the former is depicted in an implicit dynamics, while the latter is depicted in an explicit dynamics. The theoretical analysis and simulation results have further shown the efficacy of such two CTZD models.

For a continuous-time model, it is generally and necessary to develop the corresponding discrete-time form for the purposes of potential digital hardware implementation and numerical algorithm development [10]–[15]. The common way to discretize a continuous-time model is the exploitation of the Euler-type difference rule [16]. The related researches have indicated that the numerical algorithm derived via this difference rule generally has an order $O(\tau^2)$ steady-state residual error (SSRE) for solving a dynamic problem [2], [16], where τ denotes the sampling gap. That is, the SSRE reduces by 100 times when the τ value decreases by 10 times. Based on the Taylor series expansion [16],

2010 *Mathematics Subject Classification*. 15A39; 65E99

Keywords. Zhang dynamics (ZD); discrete-time algorithms; difference rules; dynamic linear matrix-vector inequality; numerical verifications

Received: 25 October 2018; Revised: 27 November 2018; Accepted: 10 May 2019

Communicated by Bolin Liao

Research supported by the National Natural Science Foundation of China under Grant 61603143, the Training Program for Outstanding Young Scientific Talents in Fujian Province University under Grant 50X19020, the Quanzhou City Science and Technology Program of China under Grant 2018C111R, and also the Promotion Program for Young and Middle-aged Teacher in Science and Technology Research of Huaqiao University under Grant ZQN-YX402.

Email addresses: xufeng110119@163.com (Feng Xu), lizexin620@126.com (Zexin Li), gdongsh@hqu.edu.cn (Dongsheng Guo)

four kinds of difference rules have been reported in [13]–[15], [17]–[20] for the first-order derivative approximation. Each of Taylor-type difference rules has a smaller truncation error than the Euler-type difference rule, i.e., $O(\tau^2)$ versus $O(\tau)$. The related researches on ZD have verified that the numerical algorithm derived via the Taylor-type difference rule always has an order $O(\tau^3)$ SSRE for solving a dynamic problem [13]–[15], [17]–[20]. That is, when the value of τ decreases by 10 times, the SSRE reduces by 1000 times. Recently, more other forms of Taylor-type difference rules have been designed and studied [21]–[24]. By summarizing these results [13]–[15], [17]–[24], the Taylor-type difference rule is more effective than the Euler-type difference rule on the discretization of a continuous-time model, especially on ZD discretization.

In this paper, a summary result based on the previous work [6]–[9] is provided by presenting and investigating four discrete-time ZD (DTZD) algorithms to solve dynamic linear matrix-vector inequality. Such four algorithms are derived by exploiting the Taylor-type difference rules in [13]–[15], [17]–[20], which are called respectively the DTZD-I, DTZD-II, DTZD-III, and DTZD-IV algorithms in this paper. Then, theoretical results are given to highlight the excellent properties of the presented four DTZD algorithms. Numerical results are also presented to substantiate the efficacy of the presented DTZD-I, DTZD-II, DTZD-III, and DTZD-IV algorithms. The theoretical and numerical results further indicate that the SSRE changes in an $O(\tau^3)$ manner for each of the presented algorithms.

The rest of this paper is organized into five sections. Section 2 shows the preliminary of solving dynamic linear matrix-vector inequality. In section 3, four Taylor-type difference rules are presented, and the resultant DTZD-I, DTZD-II, DTZD-III, and DTZD-IV algorithms are established. In Section 4, numerical results are presented, which are synthesized using the presented DTZD-I, DTZD-II, DTZD-III, and DTZD-IV algorithms. Section 5 concludes this paper with final remarks. The main contributions of this paper are as follows.

- In this paper, four different types of DTZD algorithms are presented and studied for dynamic linear matrix-vector inequality solving. This paper is an important improvement by showing a summary result of four DTZD algorithms to solve dynamic linear matrix-vector inequality.

2) The numerical results show that the SSRE of each DTZD algorithm is in the order $O(\tau^3)$. This paper is the first attempt to present four algorithm with $O(\tau^3)$ error pattern for dynamic linear matrix-vector inequality solving.

3) This paper reveals that different kinds of difference rules would lead to different types of DTZD algorithms. This finding presents a potential for the development of numerical algorithm with excellent computational performance for various dynamic problems solving.

2. Dynamic linear matrix-vector inequality

In this section, the problem formulation of dynamic linear matrix-vector inequality is presented. Then, the CTZD model in the previous work [9] is given for dynamic linear matrix-vector inequality solving.

The following problem of dynamic linear matrix-vector inequality [6]–[9] is studied in this paper:

$$A(t)x(t) \leqslant b(t), \tag{1}$$

where coefficients $A(t) \in R^{n\times n}$ and $b(t) \in R^n$ are smoothly time-varying, and $x(t) \in R^n$ is the unknown vector to be obtained. The goal of this paper is to determine a feasible solution $x(t)$ such that (1) holds true for any time instant $t_k = k\tau$ with $k = 0, 1, 2, \cdots$. To guarantee the existence of $x(t)$ in (1), we limit the investigation of this paper to the situation where $A(t)$ is nonsingular at any time instant $t_k = k\tau$.

In [9], by introducing elegantly a vector, the dynamic linear matrix-vector inequality (1) is converted to a dynamic equation, which is formulated as follows:

$$A(t)x(t) - b(t) + D(t)y(t) = 0, \tag{2}$$

where $D(t) = \text{diag}\{y_1(t), y_2(t), \cdots, y_n(t)\} \in R^{n\times n}$ and $y(t) = [y_1(t), y_2(t), \cdots, y_n(t)]^{\mathrm{T}} \in R^n$ with superscript $^{\mathrm{T}}$ denoting the transpose operator. To solve the dynamic linear matrix-vector inequality (1) and the dynamic matrix-vector equation (2), the following CTZD model is presented in [9]:

$$A(t)\dot{x}(t) + 2D(t)\dot{y}(t) = -\dot{A}(t)x(t) + \dot{b}(t) - \gamma(A(t)x(t) - b(t) + D(t)y(t)), \tag{3}$$

where $\gamma > 0 \in R$ is used to scale the solution convergence, $\dot{x}(t)$, $\dot{y}(t)$, $\dot{A}(t)$, and $\dot{b}(t)$ denote the time derivatives of $x(t)$, $y(t)$, $A(t)$, and $b(t)$, respectively.

By defining $u(t) = [x^{\mathrm{T}}(t), y^{\mathrm{T}}(t)]^{\mathrm{T}} \in R^{2n}$, $W(t) = [A(t), 2D(t)] \in R^{n\times 2n}$, $P(t) = [-\dot{A}(t), 0] \in R^{n\times 2n}$, and $Q(t) = [A(t), D(t)] \in R^{n\times 2n}$, (3) is reformulated as follows:

$$\dot{u}(t) = W^{\dagger}(t)(P(t)u(t) + \dot{b}(t)) - \gamma W^{\dagger}(t)(Q(t)u(t) - b(t)), \tag{4}$$

where $\dot{u}(t) \in R^{2n}$ denotes the time derivative of $u(t)$, and $W^{\dagger}(t)$ denotes the right pseudoinverse of $W(t)$ [9], [25]. For details about the CTZD model (4), please refer to . In the ensuing section, such a model is discretized for the purpose of potential hardware implementation [10]–[15].

3. DTZD algorithms

In this section, by exploiting the Taylor-type difference rules in [13]–[15], [17]–[20], four different DTZD algorithms are developed for dynamic linear matrix-vector inequality solving.

Lemma 1. *The Taylor-type difference rules exploited in this paper are formulated as follows [13]–[15], [17]–[20]:*

$$\dot{u}_k \approx \frac{6u_{k+1} - 3u_k - 2u_{k-1} - u_{k-2}}{10\tau}, \tag{5}$$

$$\dot{u}_k \approx \frac{2u_{k+1} - 3u_k + 2u_{k-1} - u_{k-2}}{2\tau}, \tag{6}$$

$$\dot{u}_k \approx \frac{5u_{k+1} - 3u_k - u_{k-1} - u_{k-2}}{8\tau}, \tag{7}$$

$$\dot{u}_k \approx \frac{26u_{k+1} - 33u_k + 18u_{k-1} - 11u_{k-2}}{30\tau}, \tag{8}$$

where $u_k = u(t = k\tau)$ *and* $k = 2, 3, 4, \cdots$.

Proof. See [17]–[20]. □

By using (5) to discretize the CTZD model (4), the corresponding DTZD-I algorithm for dynamic linear matrix-vector inequality solving is obtained as follows:

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

where $h = 5\gamma\tau/3 > 0 \in R$ is the step size. In addition $W_k^{\dagger}$, P_k, Q_k, b_k and $\dot{b}_k$ denote $W^{\dagger}(t = k\tau)$, $P(t = k\tau)$, $Q(t = k\tau)$, $b(t = k\tau)$ and $\dot{b}(t = k\tau)$, respectively.

Similarly, by using (6) to discretize (4), the corresponding DTZD-II algorithm for dynamic linear matrix-vector inequality solving is obtained as follows:

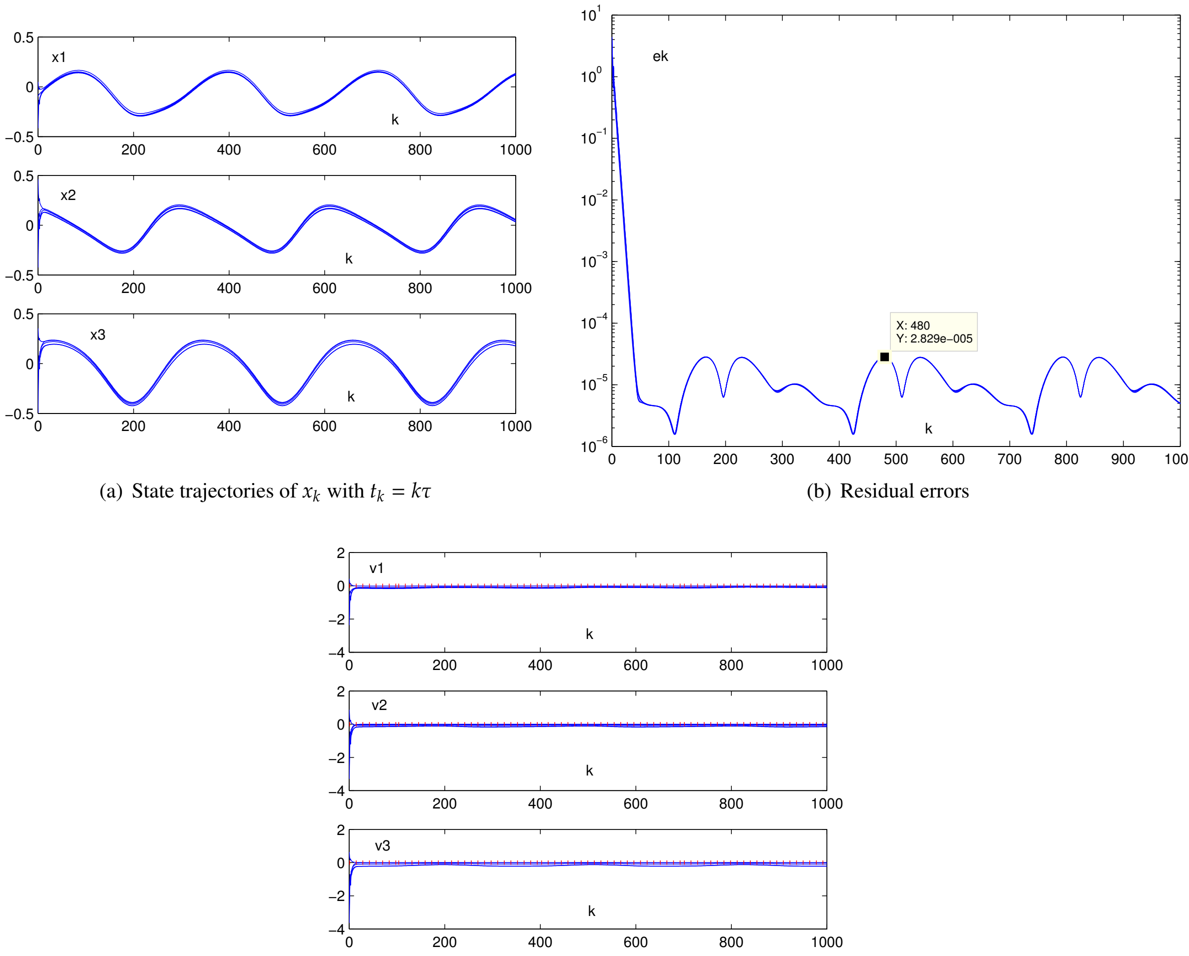
$$u_{k+1} = \frac{3}{2}u_k - u_{k-1} + \frac{1}{2}u_{k-2} + \tau W_k^{\dagger}(P_k u_k + \dot{b}_k) - hW_k^{\dagger}(Q_k u_k - b_k), \tag{10}$$

where $h = \gamma\tau > 0 \in R$ is the step size.

Furthermore, based on (7) and (8), the other two DTZD termed the DTZD-III and DTZD-IV algorithms for dynamic linear matrix-vector inequality solving are derived as follows:

$$u_{k+1} = \frac{3}{5}u_k + \frac{1}{5}u_{k-1} + \frac{1}{5}u_{k-2} + \frac{8}{5}\tau W_k^{\dagger}(P_k u_k + \dot{b}_k) - hW_k^{\dagger}(Q_k u_k - b_k), \tag{11}$$

$$u_{k+1} = \frac{33}{26}u_k - \frac{9}{13}u_{k-1} + \frac{11}{26}u_{k-2} + \frac{15}{13}\tau W_k^{\dagger}(P_k u_k + \dot{b}_k) - hW_k^{\dagger}(Q_k u_k - b_k). \tag{12}$$



(a) State trajectories of x_k with $t_k = k\tau$

(b) Residual errors

(c) Profiles of the testing error function ε_k

Figure 1: Numerical results using the presented DTZD-I algorithm (9) with $h = 0.5$ and $\tau = 0.01$ to solve dynamic linear matrix-vector inequality.

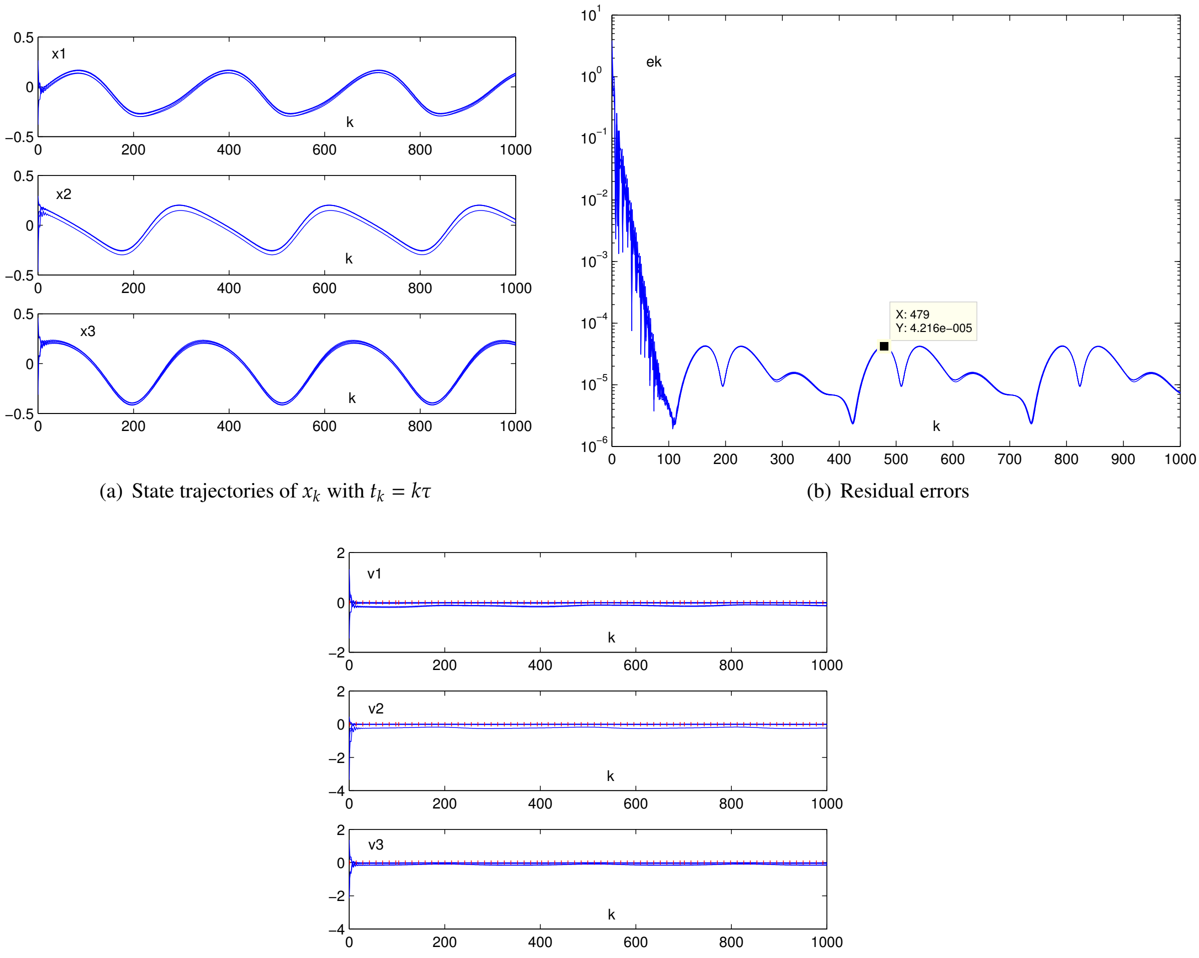
Therefore, we have obtained four DTZD algorithms [i.e., (9), (10), (11), and (12)] to solve the dynamic linear matrix-vector inequality (1). For each DTZD algorithm, three initial states (i.e., u_0, u_1, and u_2) are needed to start the iteration. In this case, based on an initial state u_0, the other two initial states u_1 and u_2 are obtained via the following iterations:

$$\begin{cases} u_1 = u_0 + \tau W_0^{\dagger}(P_0 u_0 + \dot{b}_0) - hW_0^{\dagger}(Q_0 u_0 - b_0), \\ u_2 = u_1 + \tau W_1^{\dagger}(P_1 u_1 + \dot{b}_1) - hW_k^{\dagger}(Q_1 u_1 - b_1). \end{cases}$$

For the presented four DTZD algorithms, i.e., (9)–(12), we have the following theoretical results (with the corresponding proofs being generalized from the previous work [17]–[20] and thus omitted here).

Lemma 2. *Each of the presented four DTZD algorithms (9)–(12) is a convergent method, which combines with order of the truncation error being $\mathbf{O}(\tau^3)$ for all $t_k \in [t_0, t_{\text{final}}]$, where $\mathbf{O}(\tau^3)$ is a vector with every element being $O(\tau^3)$.*

Lemma 3. *Consider a solvable dynamic linear matrix-vector inequality problem (1). For general case of $h \in (0, 1)$, the SSRE of each of the presented four DTZD algorithms (9)–(12) is of order $O(\tau^3)$.*



(a) State trajectories of x_k with $t_k = k\tau$

(b) Residual errors

(c) Profiles of the testing error function ε_k

Figure 2: Numerical results using the presented DTZD-II algorithm (10) with $h = 0.5$ and $\tau = 0.01$ to solve dynamic linear matrix-vector inequality.

4. Numerical results

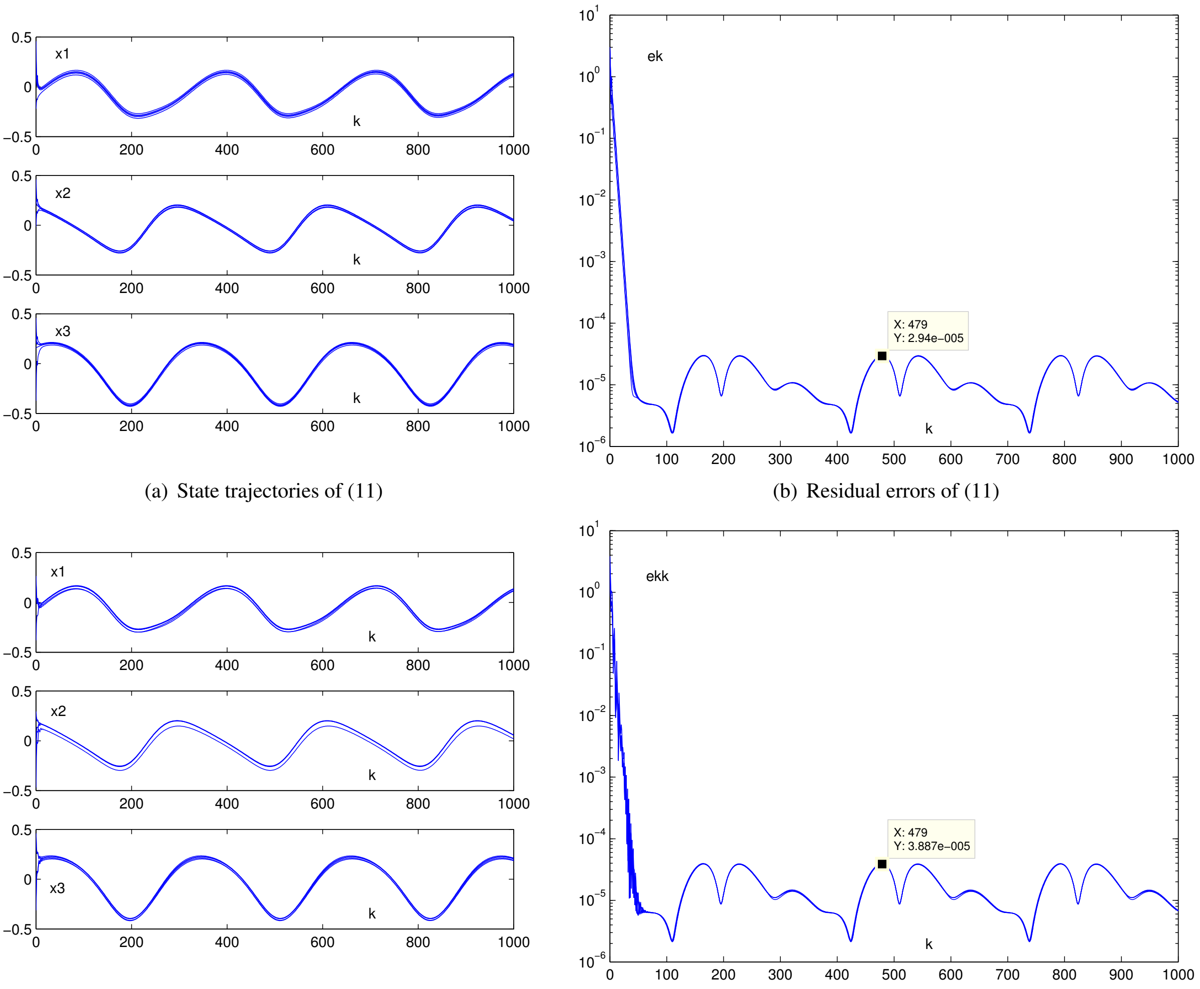
In this section, numerical results are presented to substantiate the efficacy of the presented DTZD-I, DTZD-II, DTZD-III, and DTZD-IV algorithms.

For illustration and verification, the dynamic linear matrix-vector inequality (1) is considered, with the coefficient matrix $A(t)$ and vector $b(t)$ being as follows:

$$A(t) = \begin{bmatrix} 5+\sin(2t) & \cos(2t)/2 & \cos(2t) \\ \cos(2t)/2 & 5+\sin(2t) & \cos(2t)/2 \\ \cos(2t) & \cos(2t)/2 & 5+\sin(2t) \end{bmatrix} \in R^{3\times 3} \text{ and } b(t) = \begin{bmatrix} \sin(2t) \\ \cos(2t) \\ \sin(2t)+\cos(2t) \end{bmatrix} \in R^{3}.$$

The presented four DTZD algorithm are exploited to sovle this dynamic linear matrix-vector inequality, and the related numerical results are given in Figs. 1–4 and Table 1.

First, Fig. 1 shows the numerical results synthesized by the presented DTZD-I algorithm (9) using $h = 0.5$ and $\tau = 0.01$. As seen from Fig. 1(a), starting from five randomly-generated initial states, the state trajectories of x_k [being the first 3 elements of u_k in (9)] are time-varying (with time t_k being $t_k = k\tau$). In addition, as shown in Fig. 1(b), the residual errors of (9) are all convergent, where $\|e_k\|_2 = \|Q_k u_k - b_k\|_2 = \|A_k x_k - b_k + D_k y_k\|_2$. Fig. 1(c)



(a) State trajectories of (11)

(b) Residual errors of (11)

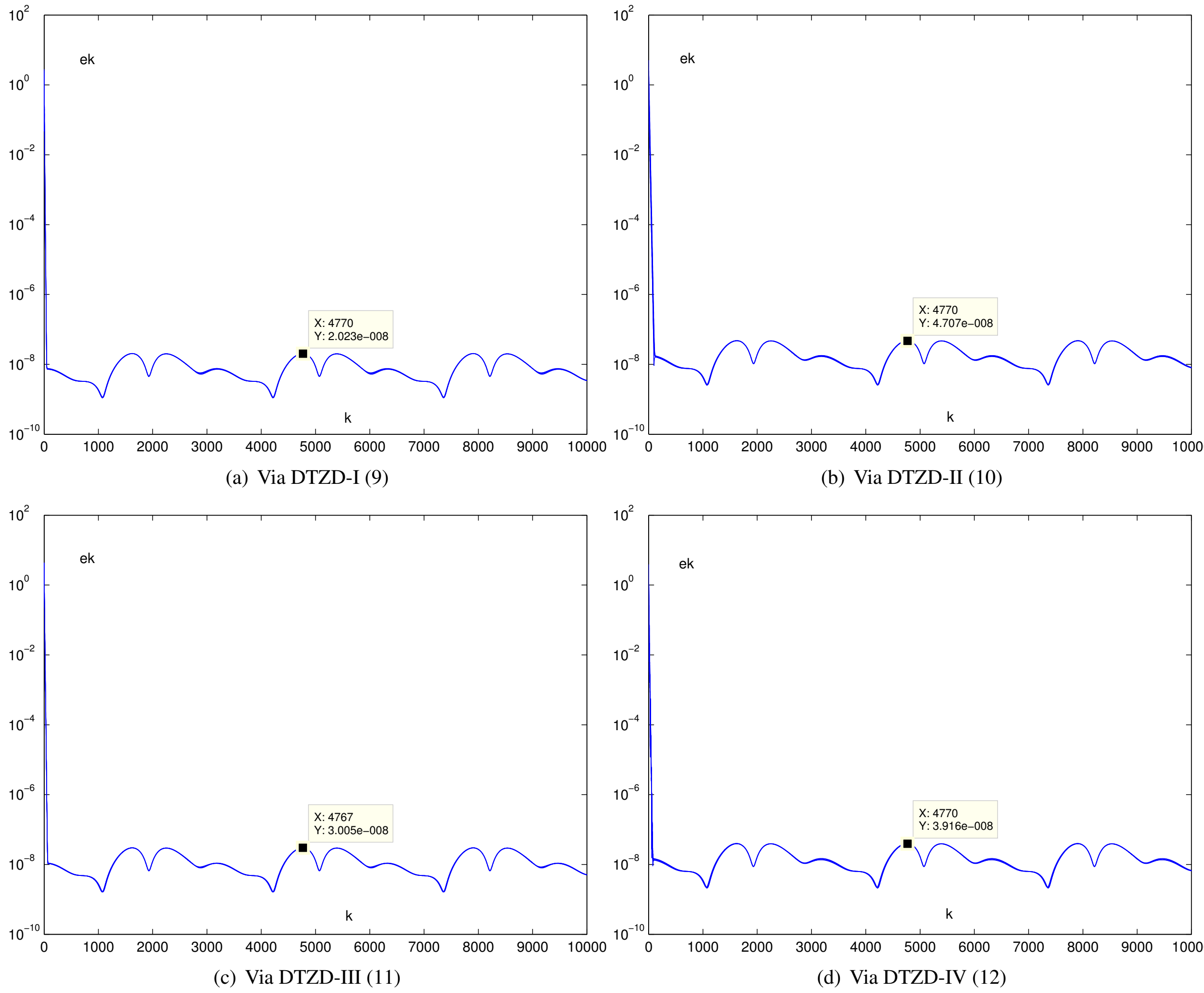
(c) State trajectories of (12)

(d) Residual errors of (12)

Figure 3: Numerical results using the presented DTZD-III and DTZD-IV algorithms (11) and (12) with $h = 0.5$ and $\tau = 0.01$ to solve dynamic linear matrix-vector inequality.

also indicates that the corresponding SSREs (i.e., $\|e_k\|_2$ with k large enough) are small enough with the maximal value being 2.829×10^{-5}. This statement means that the solutions presented in Fig. 1(a) are the solutions to (2). That is to say, the x_k solution via (9) is exactly an solution to the dynamic linear matrix-vector inequality (1). For better understanding, Fig. 1(c) shows the profiles of the testing error function $\varepsilon_k = A_k x_k - b_k$. As seen from Fig. 1(d), all elements of ε_k are less than or equal to zero, which also indicates that the above solution of x_k is an exact solution to (1). These numerical results substantiate the efficacy of the presented DTZD-I algorithm (9) for dynamic linear matrix-vector inequality solving.

Second, Fig. 2 shows the numerical results synthesized by the presented DTZD-II algorithm (10) using $h = 0.5$ and $\tau = 0.01$. As seen from Fig. 2(a), the state trajectories of x_k, starting from five randomly-generated initial states, are time-varying. In addition, as shown in Fig. 2(b), the residual errors of (10) are all convergent, and the corresponding SSREs are also small enough with the maximal value being 4.216×10^{-5}. Thus showing that the x_k solution via (10) is exactly an solution to the dynamic linear matrix-vector inequality (1). Furthermore, Fig. 2(c) indicates that all elements of ε_k are less than or equal to zero, showing again that the above solution of x_k is an exact solution to (1). These numerical results substantiate that the presented DTZD-II algorithm (10) is effective in solving dynamic linear matrix-vector inequality.



(a) Via DTZD-I (9)

(b) Via DTZD-II (10)

(c) Via DTZD-III (11)

(d) Via DTZD-IV (12)

Figure 4: Residual errors of the presented four DTZD algorithms with $h = 0.5$ and $\tau = 0.001$ to solve dynamic linear matrix-vector inequality.

Third, Fig. 3 presents the numerical results synthesized by the presented DTZD-III and DTZD-IV algorithms (11) and (12) using $h = 0.5$ and $\tau = 0.01$. As shown in Fig. 3, for each of (11) and (12), the state trajectories of x_k are time-varying and the residual errors are convergent. In addition, Fig. 3 indicates that the SSREs of (11) and (12) are small enough and are in the order of 10^{-5}. Note that, for each DTZD algorithm, all elements of the testing error function ε_k are less than or equal to zero (which are omitted here due to results similarity). These numerical results substantiate that the presented DTZD-III and DTZD-IV algorithms (11) and (12) are also effective for dynamic linear matrix-vector inequality solving.

Fourth, the presented four DTZD algorithms [i.e., DTZD-I (9), DTZD-II (10), DTZD-III (11), and DTZD-IV (12)] are tested by decreasing the τ value, with the related numerical results presented in Fig. 4. As shown in Fig. 4, for each of the presented algorithms, the residual errors are all convergent, and the corresponding SSREs are small enough (i.e., in the order of 10^{-8}). This result indicates again the efficacy of the presented four DTZD algorithms for dynamic linear matrix-vector inequality solving. More importantly, it follows from Figs. 1–4 that the SSRE of each DTZD algorithm decreases from 10^{-5} to 10^{-8} as the value of τ decreases from 0.01 to 0.001. That is, the computational performances of (9), (10), (11), and (12) can be improved effectively by decreasing the value of τ. Thus, it can be concluded that τ plays an important role in the presented DTZD algorithms and should be selected appropriately small to meet the precision requirement in practice.

Table 1: SSREs of the presented four DTZD algorithms (9), (10), (11), and (12) using different values of h and τ to solve dynamic linear matrix-vector inequality.

#	h	$\tau = 0.1$	$\tau = 0.01$	$\tau = 0.001$	manner
(9)	0.3	$\leqslant 2.373 \times 10^{-2}$	$\leqslant 4.580 \times 10^{-5}$	$\leqslant 4.704 \times 10^{-8}$	$O(\tau^3)$
	0.4	$\leqslant 2.177 \times 10^{-2}$	$\leqslant 3.518 \times 10^{-5}$	$\leqslant 3.535 \times 10^{-8}$	
	0.5	$\leqslant 1.943 \times 10^{-2}$	$\leqslant 2.788 \times 10^{-5}$	$\leqslant 2.847 \times 10^{-8}$	
	0.6	$\leqslant 1.706 \times 10^{-2}$	$\leqslant 2.331 \times 10^{-5}$	$\leqslant 2.397 \times 10^{-8}$	
	0.7	$\leqslant 1.554 \times 10^{-2}$	$\leqslant 2.011 \times 10^{-5}$	$\leqslant 2.039 \times 10^{-8}$	
(10)	0.3	$\leqslant 5.020 \times 10^{-2}$	$\leqslant 6.963 \times 10^{-5}$	$\leqslant 7.062 \times 10^{-8}$	$O(\tau^3)$
	0.4	$\leqslant 4.249 \times 10^{-2}$	$\leqslant 5.332 \times 10^{-5}$	$\leqslant 5.378 \times 10^{-8}$	
	0.5	$\leqslant 3.593 \times 10^{-2}$	$\leqslant 4.215 \times 10^{-5}$	$\leqslant 4.291 \times 10^{-8}$	
	0.6	$\leqslant 3.137 \times 10^{-2}$	$\leqslant 3.528 \times 10^{-5}$	$\leqslant 3.533 \times 10^{-8}$	
	0.7	$\leqslant 2.755 \times 10^{-2}$	$\leqslant 3.049 \times 10^{-5}$	$\leqslant 3.035 \times 10^{-8}$	
(11)	0.3	$\leqslant 2.711 \times 10^{-2}$	$\leqslant 4.092 \times 10^{-5}$	$\leqslant 4.946 \times 10^{-8}$	$O(\tau^3)$
	0.4	$\leqslant 2.279 \times 10^{-2}$	$\leqslant 3.710 \times 10^{-5}$	$\leqslant 3.732 \times 10^{-8}$	
	0.5	$\leqslant 2.013 \times 10^{-2}$	$\leqslant 2.953 \times 10^{-5}$	$\leqslant 3.005 \times 10^{-8}$	
	0.6	$\leqslant 1.808 \times 10^{-2}$	$\leqslant 2.445 \times 10^{-5}$	$\leqslant 2.499 \times 10^{-8}$	
	0.7	$\leqslant 1.652 \times 10^{-2}$	$\leqslant 2.134 \times 10^{-5}$	$\leqslant 2.137 \times 10^{-8}$	
(12)	0.3	$\leqslant 4.313 \times 10^{-2}$	$\leqslant 6.544 \times 10^{-5}$	$\leqslant 6.557 \times 10^{-8}$	$O(\tau^3)$
	0.4	$\leqslant 3.632 \times 10^{-2}$	$\leqslant 4.868 \times 10^{-5}$	$\leqslant 4.876 \times 10^{-8}$	
	0.5	$\leqslant 3.159 \times 10^{-2}$	$\leqslant 3.922 \times 10^{-5}$	$\leqslant 3.929 \times 10^{-8}$	
	0.6	$\leqslant 2.784 \times 10^{-2}$	$\leqslant 3.240 \times 10^{-5}$	$\leqslant 3.263 \times 10^{-8}$	
	0.7	$\leqslant 2.397 \times 10^{-2}$	$\leqslant 2.783 \times 10^{-5}$	$\leqslant 2.798 \times 10^{-8}$	

Fifth, for further investigation, we simulate the DTZD-I (9), DTZD-II (10), DTZD-III (11), and DTZD-IV (12) using different values of h and τ to solve the dynamic linear matrix-vector inequality (1). The corresponding numerical results are provided in Table 1, which indicates that all of the presented DTZD algorithms are effective on solving (1), e.g., in the order of 10^{-5} and 10^{-8}. Furthermore, the following summaries are obtained from Table 1 (as well as Fig. 4).

- The SSRE of each DTZD algorithm changes in an $O(\tau^3)$ manner. That is, by fixing h, when τ decreases by 10 times, the SSRE of each DTZD algorithm is reduced by 100 times, which coincides with Lemmas 2 and 3. This statement also means that the computational performances of the presented four DTZD algorithms are improved effectively when the τ value is decreased.
- By fixing τ, when h is increased, the SSRE of each DTZD algorithm is decreased. This statement shows that the computational performances of the presented four DTZD algorithms can be further improved by increasing the value of h appropriately.

Based on these qualitative and quantitative results, it can be concluded that, for the presented four DTZD algorithms, different computational performances can be achieved by selecting appropriate h and τ.

In summary, the above results (i.e., Figs. 1–4 and Table 1) indicate the efficacy of the presented the presented DTZD-I, DTZD-II, DTZD-III, and DTZD-IV algorithms for dynamic linear matrix-vector inequality solving.

5. Conclusions

In this paper, four different types of DTZD algorithms [i.e., DTZD-I (9), DTZD-II (10), DTZD-III (11), and DTZD-IV (12)] have been presented and investigated to solve the dynamic linear matrix-vector inequality (1). Then, theoretical results have been provided to highlight the excellent computational properties of the presented algorithms. That is, the SSRE for each DTZD algorithm presented in this paper changes in an $O(\tau^3)$ manner, showing that the computational performance is improved effectively by decreasing the value of τ. Numerical results have been presented to further substantiate the efficacy of the presented four DTZD algorithms. A future research direction can be the study of the the presented four DTZD algorithms using different types of (nonlinear) activation functions [25] for dynamic linear matrix-vector inequality solving. Another future research direction is the development of more DTZD algorithms by referring to more Taylor-type difference rules [21]–[24].

References

[1] F. Xu, Z. Li, Z. Nie, H. Shao, and D. Guo, Zeroing neural network for solving time-varying linear equation and inequality systems, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 8, pp. 2346–2357, 2019

[2] Y. Shi and Y. Zhang, New discrete-time models of zeroing neural network solving systems of time-variant linear and nonlinear inequalities, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, in press, 2018.

[3] D. Guo and F. Xu, Design and analysis of discrete algorithm for time-varying linear inequality solving, *Journal of Huaqiao University(Natural Science)*, vol. 38, no. 5, pp. 732–736, 2017.

[4] L. Xiao and Y. Zhang, Dynamic design, numerical solution and effective verification of acceleration-level obstacle avoidance scheme for robot manipulators, *International Journal of Systems Science*, vol. 47, no. 4, pp. 932–945, 2016.

[5] D. Guo and K. Li, Acceleration-level obstacle-avoidance scheme for motion planning of redundant robot manipulators, *Proceedings of IEEE International Conference on Robotics and Biomimetics*, pp. 1313–1318, 2016.

[6] L. Xiao and Y. Zhang, Zhang neural network versus gradient neural network for solving time-varying linear inequalities, *IEEE Transactions on Neural Networks*, vol. 22, no. 10, pp. 1676–1684, 2011.

[7] D. Guo and Y. Zhang, A new variant of the Zhang neural network for solving online time-varying linear inequalities, *Proceedings of the Royal Society A*, vol. 468, no. 2144, pp. 2255–2271, 2012.

[8] 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, 2013.

[9] D. Guo and Y. Zhang, ZNN for solving online time-varying linear matrix-vector inequality via equality conversion, *Applied Mathematics and Computation*, vol. 259, pp. 327–338, 2015.

[10] D. Guo, A. Li, X. Lin, F. Xu, and Z. Su, Three-step DTZNN algorithm for time-varying linear matrix inequality solving, *Proceedings of the International Joint Conference on Neural Networks*, pp. 2590–2595, 2017.

[11] Y. Xia, J. Wang, and D. L. Hung, Recurrent neural networks for solving linear inequalities and equations, *IEEE Transactions on Circuits and Systems I*, vol. 46, no. 4, pp. 452–462, 1999.

[12] J. Singh and N. Barabanov, Stability of discrete time recurrent neural networks and nonlinear optimization problems, *Neural Network*, vol. 74, pp. 58–72, 2016.

[13] D. Guo, Z. Nie, and L. Yan, Theoretical analysis, numerical verification and geometrical representation of new three-step DTZD algorithm for time-varying nonlinear equations solving, *Neurocomputing*, vol. 214, pp. 516–526, 2016.

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

[15] 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, *Numerical Algorithms*, vol. 77, no. 1, pp. 23–36, 2018.

[16] J. H. Mathews and F. D. Fink, Numerical Methods Using MATLAB, fourth ed., Prentice Hall, New Jersey, 2004.

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

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

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

[20] J. Li, M. Mao, F. Uhlig, and Y. Zhang, Z-type neural-dynamics for time-varying nonlinear optimization under a linear equality constraint with robot application, *Journal of Computational and Applied Mathematics*, vol. 327, pp. 155–166, 2018.

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

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

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

[24] M. D. Petkovic, P. S. Stanimirovic, and V. N. Katsikis, Modified discrete iterations for computing the inverse and pseudoinverse of the time-varying matrix, *Neurocomputing*, vol. 289, pp. 155–165, 2018.

[25] D. Guo, S. Li, and P. S. Stanimirovic, Analysis and application of modified ZNN design with robustness against harmonic noise, *IEEE Transactions on Industrial Informatics*, in press, 2019.