IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 14, 2022, accepted March 21, 2022, date of publication March 28, 2022, date of current version April 1, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3162860

Stability-Constraint-Free Solutions to Solve Time-Varying Linear Equation System

JIAN LI[1,2], JIANWEI FAN[1], XINHUI ZHU[1], YANGE SUN[1], AND YANLING LI[1]
[1]School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China
[2]School of Information Engineering, Zhengzhou University, Zhengzhou 450066, China

Corresponding authors: Jian Li (lijcit@xynu.edu.cn) and Jianwei Fan (fjwlijie@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 62006205 and Grant 62002307; in part by the China Postdoctoral Science Foundation funded project under Grant 2021TQ0299; in part by the Project of Science and Technology Department of Henan Province under Grant 212102210392; in part by the Natural Science Foundation of Henan Province under Grant 222300420274; in part by the Key Scientific Research Projects of Colleges and Universities, Henan, under Grant 21A520039; and in part by the Nanhu Scholars Program for Young Scholars of XYNU.

ABSTRACT Time-varying linear equation systems have been solved by the traditional zeroing neural dynamics approach in recent years. However, this method has to satisfy the stability constraint, which is a very rigorous condition. For this reason, traditional Lagrange-type finite difference formulas fail to lead to effective solutions, and we have to utilize more instants and reduce accuracy so that this condition is satisfied. In this work, we develop a new method of solving a time-varying linear equation system, which is based on theoretical solution decomposition. As a result, the proposed solutions are stability-constraint-free. We do not have to meticulously search effective time-discretization formulas because traditional Lagrange-type formulas are sufficient and especially effective. In addition, the proposed solutions have other advantages. For example, they do not need convergence procedures; they do not have storage requirements for past calculative results; and they are still effective when the sampling gap value is relatively large. Detailed comparisons are presented in this paper. Comparative numerical experiments are also shown to substantiate the effectiveness and advantages of the proposed solutions.

INDEX TERMS Time-varying linear equation system, stability-constraint-free solutions, theoretical solution decomposition, Lagrange finite difference.

I. INTRODUCTION

Linear equation systems are fundamental mathematical problems that are frequently encountered in engineering and scientific fields [1]–[6]. When coefficients in a linear equation system vary over time, the system becomes a time-varying linear equation system (TVLES) [7]. The real-time requirement has gradually increased, and thus, effective resolution of the time-varying problem is urgently needed [8]–[13].

Although TVLES is a static linear equation system when considering a specific time instant, solving TVLES is much more complicated than static solving because real time is the core requirement of solving TVLES [14]–[19]. There are many classical methods to solve static linear equation systems in numerical computation methods, such as Gaussian elimination and decomposition [20]–[24]. However, these methods fail to solve TVLES because of time delays [25]–[29]. Specifically, when these methods are utilized to solve TVLES, they must be viewed as static at different time instants. When time reaches the current instant, these conventional methods can be implemented. Thus, the current instant solution can be obtained only after a few seconds because of the necessary computation time. If we view the obtained solution as the next instant solution, a lagging error exists.

Different from conventional methods, zeroing neural dynamics has been presented, which is especially used for time-varying problems [30]–[33]. Zeroing neural dynamics is a special kind of neural network [32]–[43]. Zeroing neural dynamics to develop solutions of the time-varying problem has the following procedure [44]: First, the problem is viewed as continuous-time problem, and the vector/matrix-form error function $\mathbf{e}(t)$ is defined based on the problem to be solved. Second, we select a suitable design formula, such as $\dot{\mathbf{e}}(t) = -\lambda\mathbf{e}(t)$, to zero the error function. We rearrange the equation as an explicit differentiation equation. Finally,

The associate editor coordinating the review of this manuscript and approving it for publication was Yeliz Karaca.

TABLE 1. Lagrange-type formulas for approximating $\dot{x}(t_k)$ with different truncation errors.

Expression	Truncation error
$\dot{x}(t_k)=\frac{x(t_{k+1})-x(t_{k-1})}{2\tau}+O(\tau^2)$	$O(\tau^2)$
$\dot{x}(t_k)=\frac{2x(t_{k+1})+3x(t_k)-6x(t_{k-1})+x(t_{k-2})}{6\tau}+O(\tau^3)$	$O(\tau^3)$
$\dot{x}(t_k)=\frac{3x(t_{k+1})+10x(t_k)-18x(t_{k-1})+6x(t_{k-2})-x(t_{k-3})}{12\tau}+O(\tau^4)$	$O(\tau^4)$
$\dot{x}(t_k)=\frac{12x(t_{k+1})+65x(t_k)-120x(t_{k-1})+60x(t_{k-2})-20x(t_{k-3})+3x(t_{k-4})}{60\tau}+O(\tau^5)$	$O(\tau^5)$
$\dot{x}(t_k)=\frac{10x(t_{k+1})+77x(t_k)-150x(t_{k-1})+100x(t_{k-2})-50x(t_{k-3})+15x(t_{k-4})-2x(t_{k-5})}{60\tau}+O(\tau^6)$	$O(\tau^6)$

we search for an effective time-discretization formula for the discretization of the explicit differentiation equation so that we can obtain the final solution to solve the original problem.

From the above procedure, we know that time-discretization formulas are necessary, which is also the core point to realize real-time computation [45], [46]. However, finding an effective formula for discretization is challenging work and must satisfy strict conditions [47]. First, it must be 1-step-ahead, i.e., discretizing $\dot{x}(t_k)$ by $x(t_{k+1})$, $x(t_k)$, $x(t_{k-1})$ and $x(t_{k-2})\cdots$, so that the generated model has the prediction ability. For this simple reason, most formulas in the numerical ODE literature, such as backward formulas, are unsuitable. Second, a 1-step-ahead formula does not necessarily lead to a stable model. In [48], a series of Lagrange-type formulas were presented, which are all 1-step-ahead. For convenience of discussion, they are partially repeated, as shown in Table 1. None of these formulas lead to stable solutions when we use the zeroing neural dynamics method.

There also exist some effective discretization formulas due to researchers' unremitting efforts. However, the approximation needs more instants so that the formula satisfies the requirement of stability. In [49], the relationship between the time-instant number and precision of formulas was investigated. Specifically,"two-instant and three-instant formula groups have linear precision at most. Four-instant formula group has quadratic precision at most. Five-instant and six-instant formula groups have cubic precision at most. Seven-instant and eight-instant formula groups have quartic precision at most. Nine-instant formula group has quintic precision at most" [49]. These formulas have lower precision than Lagrange-type formulas when the time-instant number is the same.

In this work, we develop a new way to solve TVLES, which is based on theoretical solution decomposition. As a result, the proposed solutions are stability-constraint-free. We do not have to meticulously search effective time-discretization formulas because classical Lagrange-type formulas are sufficient and especially effective. In addition, the proposed solutions have other advantages. For example, they do not need convergence procedures; they do not require storage past calculative results; and they are still effective when the sampling gap value is relatively large. Detailed comparisons are presented in this paper. Comparative numerical experiments are also shown to substantiate the effectiveness and advantages of the proposed solutions.

The remainder of this paper is organized into six sections. Section II shows the problem formulation of TVLES. Section III develops the stability-constraint-free solutions. Section IV compares the proposed and conventional solutions. Section V gives a theoretical analysis of the proposed solutions. Section VI presents some comparative numerical experimental results. Section VII concludes this paper with final remarks. The main contributions are listed as follows.

1) We propose a method to solve a time-varying linear equation system that is based on theoretical solution decomposition, which is quite different from zeroing neural dynamics.
2) Stability-constraint-free solutions are proposed based on this new method. The proposed solutions have some advantages in terms of stability, convergence procedures and so on.

II. PROBLEM FORMULATION

A time-varying linear equation system (TVLES) is a stream of linear equation systems over time. It is formulated as

$$A(t_{k+1})\mathbf{x}(t_{k+1})=\mathbf{b}(t_{k+1}), \tag{1}$$

where $A(t_{k+1})$ is the time-varying nonsingular square matrix and vector, and $\mathbf{b}(t_{k+1})$ is time-varying vector, and thus TVLES (1) is determined. Note that future solution $\mathbf{x}(t_{k+1})$ should be obtained during time interval $[t_k, t_{k+1})$ to satisfy the requirement of real-time computation.

III. STABILITY-CONSTRAINT-FREE SOLUTIONS

We consider the continuous-time form of TVLES (1) and define $\mathbf{x}^*(t)$ as its theoretical solution. Then, we have the following equation:

$$A(t)\mathbf{x}^*(t)=\mathbf{b}(t).$$

Direct derivation of the above equation leads to

$$A(t)\dot{\mathbf{x}}^*(t)+\dot{A}(t)\mathbf{x}^*(t)=\dot{\mathbf{b}}(t),$$

where $\dot{\mathbf{x}}^*(t)$, $\dot{A}(t)$ and $\dot{\mathbf{b}}(t)$ are time derivatives of $\mathbf{x}^*(t)$, $A(t)$ and $\mathbf{b}(t)$, respectively. It is rewritten as

$$\dot{\mathbf{x}}^*(t)=A^{-1}(t)(\dot{\mathbf{b}}(t)-\dot{A}(t)\mathbf{x}^*(t)), \tag{2}$$

where $A^{-1}(t)$ is inverse of $A(t)$. We use the classical central difference formula

$$\dot{x}(t_k) = \frac{x(t_{k+1}) - x(t_{k-1})}{2\tau} + O(\tau^2) \tag{3}$$

to discretize $\dot{\mathbf{x}}^*(t)$ in equation (2), where τ is the sampling gap, and obtain

$$\dot{\mathbf{x}}^*(t_k) = \frac{\mathbf{x}^*(t_{k+1}) - \mathbf{x}^*(t_{k-1})}{2\tau} + \mathbf{O}(\tau^2).$$

We combine the above discretization equation and equation (2) with truncation errors omitted and obtain

$$\frac{\mathbf{x}^*(t_{k+1}) - \mathbf{x}^*(t_{k-1})}{2\tau} = A^{-1}(t_k)(\dot{\mathbf{b}}(t_k) - \dot{A}(t_k)\mathbf{x}^*(t_k)). \tag{4}$$

Furthermore, the backward finite difference formula

$$\dot{u}(t_k) = \frac{3u(t_k) - 4u(t_{k-1}) + u(t_{k-2})}{2\tau} + O(\tau^2)$$

is employed to approximate $\dot{\mathbf{b}}(t_k)$ and $\dot{A}(t_k)$, i.e.,

$$\begin{cases} \dot{\mathbf{b}}(t_k) = \dfrac{3\mathbf{b}(t_k) - 4\mathbf{b}(t_{k-1}) + \mathbf{b}(t_{k-2})}{2\tau} + \mathbf{O}(\tau^2), \\ \dot{A}(t_k) = \dfrac{3A(t_k) - 4A(t_{k-1}) + A(t_{k-2})}{2\tau} + \mathbf{O}(\tau^2). \end{cases}$$

Then, employing the above equations to approximate $\dot{\mathbf{b}}(t_k)$ and $\dot{A}(t_k)$ in equation (4), we have

$$\begin{aligned} &\frac{\mathbf{x}^*(t_{k+1}) - \mathbf{x}^*(t_{k-1})}{2\tau} \\ &= A^{-1}(t_k)\Bigg(\frac{3\mathbf{b}(t_k) - 4\mathbf{b}(t_{k-1}) + \mathbf{b}(t_{k-2})}{2\tau} \\ &\quad - \frac{3A(t_k) - 4A(t_{k-1}) + A(t_{k-2})}{2\tau}\mathbf{x}^*(t_k)\Bigg). \end{aligned} \tag{5}$$

It is rewritten as

$$\begin{aligned} \mathbf{x}^*(t_{k+1}) = {}& \mathbf{x}^*(t_{k-1}) + A^{-1}(t_k)\big(3\mathbf{b}(t_k) - 4\mathbf{b}(t_{k-1}) \\ &+ \mathbf{b}(t_{k-2})\text{-}(3A(t_k) - 4A(t_{k-1}) + A(t_{k-2}))\mathbf{x}^*(t_k)\big). \end{aligned}$$

Note that the allowed period of time to calculate solution $\mathbf{x}(t_{k+1})$ is from t_k to t_{k+1}. During this time interval, $A(t_k)$, $A(t_{k-1})$, $A(t_{k-2})$, $\mathbf{b}(t_k)$, $\mathbf{b}(t_{k-1})$, and $\mathbf{b}(t_{k-2})$ are all known. Thus, $\mathbf{x}^*(t_k)$ and $\mathbf{x}^*(t_{k-1})$ can be obtained directly by

$$\begin{cases} \mathbf{x}^*(t_k) = A^{-1}(t_k)\mathbf{b}(t_k), \\ \mathbf{x}^*(t_{k-1}) = A^{-1}(t_{k-1})\mathbf{b}(t_{k-1}). \end{cases}$$

Finally, the stability-constraint-free solution to solve TVLES, which is termed the TVLES-SCF-I solution, is obtained as follows:

$$\begin{aligned} \mathbf{x}(t_{k+1}) = {}& \bar{\mathbf{x}}(t_{k-1}) + A^{-1}(t_k)\big(3\mathbf{b}(t_k) - 4\mathbf{b}(t_{k-1}) \\ &+ \mathbf{b}(t_{k-2})\text{-}(3A(t_k) - 4A(t_{k-1}) + A(t_{k-2}))\bar{\mathbf{x}}(t_k)\big), \end{aligned} \tag{6}$$

where $\bar{\mathbf{x}}(t_k)$ and $\bar{\mathbf{x}}(t_{k-1})$ are obtained by $\bar{\mathbf{x}}(t_k) = A^{-1}(t_k)\mathbf{b}(t_k)$ and $\bar{\mathbf{x}}(t_{k-1}) = A^{-1}(t_{k-1})\mathbf{b}(t_{k-1})$.

In addition, we can employ other one-step-ahead formulas with higher precision, such as

$$\begin{aligned} \dot{x}(t_k) = {}& \frac{1}{3\tau}x(t_{k+1}) + \frac{1}{2\tau}x(t_k) - \tau x(t_{k-1}) \\ &+ \frac{1}{6\tau}x(t_{k+2}) + O(\tau^3), \end{aligned} \tag{7}$$

to discretize $\dot{\mathbf{x}}^*(t)$, and the following backward formula is employed to approximate $\dot{\mathbf{b}}(t_k)$ and $\dot{A}(t_k)$:

$$\begin{aligned} \dot{u}(t_k) = {}& \frac{11}{6\tau}u(t_k) - \frac{3}{\tau}u(t_{k-1}) + \frac{3}{2\tau}u(t_{k-2}) \\ &- \frac{1}{3\tau}u(t_{k-3}) + O(\tau^3). \end{aligned}$$

Similarly, the stability-constraint-free solution to solve TVLES, which is termed the TVLES-SCF-II solution, is obtained as follows:

$$\begin{aligned} \mathbf{x}(t_{k+1}) = {}& -\frac{3}{2}\bar{\mathbf{x}}(t_k) + 3\bar{\mathbf{x}}(t_{k-1}) - \frac{1}{2}\bar{\mathbf{x}}(t_{k-2}) + A^{-1}(t_k)\Big(\frac{11}{2}\mathbf{b}(t_k) \\ &- 9\mathbf{b}(t_{k-1}) + \frac{9}{2}\mathbf{b}(t_{k-2}) - \mathbf{b}(t_{k-3}) - \big(\frac{11}{2}A(t_k) \\ &- 9A(t_{k-1}) + \frac{9}{2}A(t_{k-2}) - A(t_{k-3})\big)\bar{\mathbf{x}}(t_k)\Big), \end{aligned} \tag{8}$$

where $\bar{\mathbf{x}}(t_k)$, $\bar{\mathbf{x}}(t_{k-1})$ and $\bar{\mathbf{x}}(t_{k-2})$ are obtained by $\bar{\mathbf{x}}(t_k) = A^{-1}(t_k)\mathbf{b}(t_k)$, $\bar{\mathbf{x}}(t_{k-1}) = A^{-1}(t_{k-1})\mathbf{b}(t_{k-1})$ and $\bar{\mathbf{x}}(t_{k-2}) = A^{-1}(t_{k-2})\mathbf{b}(t_{k-2})$.

In solutions (6) and (8), the calculations of inverse $A^{-1}(t_k)$, $A^{-1}(t_{k-1})$ etc are necessary. In this work, we only investigate the case of determined TVLES for the research rigor. For an underdetermined TVLES, the solutions of inverse become pseudo-inverse, which are not unique. It may lead to the failure of solving TVLES. In this case, we could use the minimum norm solution, which minimizes the norm $\|\mathbf{x}\|_2$, such that the solutions of pseudo-inverse are unique. Furthermore, considering the characteristic of continuous change over time for TVLES, the proposed solutions would be still effective.

IV. COMPARISONS WITH CONVENTIONAL SOLUTIONS

When we develop solutions in conventional ways to solve TVLES [46], an error function is first defined as $A(t)\mathbf{x}(t) - \mathbf{b}(t)$. Then, the zeroing neural dynamics method is utilized, and a continuous-time solution with a differential equation form is obtained as

$$\dot{\mathbf{x}}(t) = A^{-1}(t)\left(-\dot{A}(t)\mathbf{x}(t) - \lambda(A(t)\mathbf{x}(t) - \mathbf{b}(t)) + \dot{\mathbf{b}}(t)\right),$$

where coefficient $\lambda > 0$. Furthermore, discretization formulas are employed for the discretization of the above continuous-time solution. For example, when the Taylor discretization formula [50]

$$\begin{aligned} \dot{x}(t_k) = {}& \frac{1}{\tau}x(t_{k+1}) - \frac{3}{2\tau}x(t_k) + \frac{1}{\tau}x(t_{k-1}) \\ &- \frac{1}{2\tau}x(t_{k-2}) + O(\tau^2). \end{aligned} \tag{9}$$

is employed, the conventional solution is obtained as follows:

$$\begin{aligned} \mathbf{x}(t_{k+1}) = {}& -A^{-1}(t_k)\big(\tau\dot{A}(t_k)\mathbf{x}(t_k) + h(A(t_k)\mathbf{x}(t_k) \\ &- \mathbf{b}(t_k)) - \tau\dot{\mathbf{b}}(t_k)\big) + \frac{3}{2}\mathbf{x}(t_k) - \mathbf{x}(t_{k-1}) + \frac{1}{2}\mathbf{x}(t_{k-2}) \end{aligned} \tag{10}$$

with $h = \tau\lambda$. When the five-instant discretization formula [47]

$$\dot{x}(t_k) = \frac{4}{9\tau}x(t_{k+1}) + \frac{1}{18\tau}x(t_k) - \frac{1}{3\tau}x(t_{k-1}) - \frac{5}{18\tau}x(t_{k-2}) + \frac{1}{9\tau}x(t_{k-3}) + O(\tau^3) \quad (11)$$

is employed, the conventional solution is obtained as follows:

$$\begin{aligned}\mathbf{x}(t_{k+1}) = &-\frac{9}{4}A^{-1}(t_k)\big(\tau\dot{A}(t_k)\mathbf{x}(t_k) + h(A(t_k)\mathbf{x}(t_k) \\ &- \mathbf{b}(t_k)) - \tau\dot{\mathbf{b}}(t_k)\big) - \frac{1}{8}\mathbf{x}(t_k) + \frac{3}{4}\mathbf{x}(t_{k-1}) \\ &+ \frac{5}{8}\mathbf{x}(t_{k-2}) - \frac{1}{4}\mathbf{x}(t_{k-3}). \end{aligned} \quad (12)$$

For comparisons of conventional solutions and the proposed solutions in this work, the central difference formula (3) is employed for the discretization of conventional continuous-time solutions because the proposed TVLES-SCF-I solution (6) is also based on the central difference formula (3). The corresponding solution based on the central difference formula (3) is shown below:

$$\begin{aligned}\mathbf{x}(t_{k+1}) = &-\frac{1}{2}A^{-1}(t_k)\big(\tau\dot{A}(t_k)\mathbf{x}(t_k) + h(A(t_k)\mathbf{x}(t_k) \\ &- \mathbf{b}(t_k)) - \tau\dot{\mathbf{b}}(t_k)\big) + \mathbf{x}(t_{k-1}). \end{aligned} \quad (13)$$

Similarly, a finite difference formula (7) is employed for the discretization of the conventional continuous-time solution because the proposed TVLES-SCF-II solution (8) is also based on this formula. The corresponding solution based on this formula is shown below:

$$\begin{aligned}\mathbf{x}(t_{k+1}) = &-3A^{-1}(t_k)\big(\tau\dot{A}(t_k)\mathbf{x}(t_k) + h(A(t_k)\mathbf{x}(t_k) \\ &- \mathbf{b}(t_k)) - \tau\dot{\mathbf{b}}(t_k)\big) - \frac{3}{2}\mathbf{x}(t_k) \\ &+ 3\mathbf{x}(t_{k-1}) - \frac{1}{2}\mathbf{x}(t_{k-2}). \end{aligned} \quad (14)$$

Note that solutions (13) and (14) fail to solve TVLES because of nonstability, which is caused by the ineffective discretization formulas (3) and (7). In contrast, the proposed TVLES-SCF-I (6) and TVLES-SCF-II solutions (8) are effective in solving TVLES, which are also based on these two formulas. Specifically, the proposed TVLES-SCF solutions have some differences and advantages compared with conventional solutions, as shown below:

- TVLES-SCF solutions are not bound by the constraint of zero stability. For example, the TVLES-SCF-I solution (6) is based on the central difference formula (3). Based on previous work, we know that when the central difference formula (3) is employed, the corresponding solution does not satisfy the constraint of zero stability, which would lead to divergence of the solution. However, the TVLES-SCF-I solution (6) is still effective in solving the TVLES problem (1).
- TVLES-SCF solutions are still effective when the sampling gap value is relatively large. In conventional solutions such as (12), parameter $h = \tau\lambda$ exists, which affects the convergence of the solution. Limited by this parameter, the sampling gap value has to be relatively small, especially when employing high-precision discretization formulas. However, TVLES-SCF solutions have no such parameter, and the sampling gap value is only bound by truncation error. Thus, TVLES-SCF solutions are still effective when the sampling gap value is relatively large compared with conventional solutions. In addition, it is a challenge to set the value of parameter h. A smaller value of h leads to slower convergence, and a larger value of h leads to divergence of the solution.
- TVLES-SCF solutions do not need convergence procedures and directly calculate and obtain future-instant results. For conventional solutions, the calculation procedures are iterations. Current-step calculation is based on past results. Starting from a random initial state, the calculated results gradually converge to theoretical time-varying results. However, TVLES-SCF solutions are not iterations, which directly calculate and obtain future-instant results with quite high precision.
- TVLES-SCF solutions do not require storage past calculative results. For conventional solutions, the current-step calculation is based on past calculative results such as $\mathbf{x}(t_{k-1})$ and $\mathbf{x}(t_{k-2})$ and thus has to use extra space to store them. However, TVLES-SCF solutions only need past known information such as $A(t_{k-1})$ and $A(t_{k-2})$.

V. THEORETICAL ANALYSIS

Theoretical analysis is shown below to guarantee the effectiveness and precision of the proposed TVLES-SCF solutions.

Theorem 1: If the TVLES problem (1) has high-order derivatives, the errors of the TVLES-SCF-I (6) and TVLES-SCF-II solutions (8) to solve this problem are $O(\tau^3)$ and $O(\tau^4)$, respectively, where the solution error is defined as $\|\mathbf{x}(t_{k+1}) - \mathbf{x}^*(t_{k+1})\|$.

Proof: When we consider the truncation error during the derivation process of the TVLES-SCF-I solution (6), equation (4) is exactly

$$\frac{\mathbf{x}^*(t_{k+1}) - \mathbf{x}^*(t_{k-1})}{2\tau} + \mathbf{O}(\tau^2) = A^{-1}(t_k)(\dot{\mathbf{b}}(t_k) - \dot{A}(t_k)\mathbf{x}^*(t_k)).$$

Furthermore, equation (5) is exactly

$$\begin{aligned}&\frac{\mathbf{x}^*(t_{k+1}) - \mathbf{x}^*(t_{k-1})}{2\tau} + \mathbf{O}(\tau^2) \\ &= A^{-1}(t_k)\Big(\frac{3\mathbf{b}(t_k) - 4\mathbf{b}(t_{k-1}) + \mathbf{b}(t_{k-2})}{2\tau} + \mathbf{O}(\tau^2) \\ &\quad - \big(\frac{3A(t_k) - 4A(t_{k-1}) + A(t_{k-2})}{2\tau} + \mathbf{O}(\tau^2)\big)\mathbf{x}^*(t_k)\Big),\end{aligned}$$

which can be rewritten as

$$\begin{aligned}\mathbf{x}^*(t_{k+1}) = &\mathbf{x}^*(t_{k-1}) + A^{-1}(t_k)\big(3\mathbf{b}(t_k) - 4\mathbf{b}(t_{k-1}) \\ &+ \mathbf{b}(t_{k-2})\text{-}(3A(t_k) - 4A(t_{k-1}) \\ &+ A(t_{k-2}))\mathbf{x}^*(t_k)\big) + \mathbf{O}(\tau^3).\end{aligned}$$

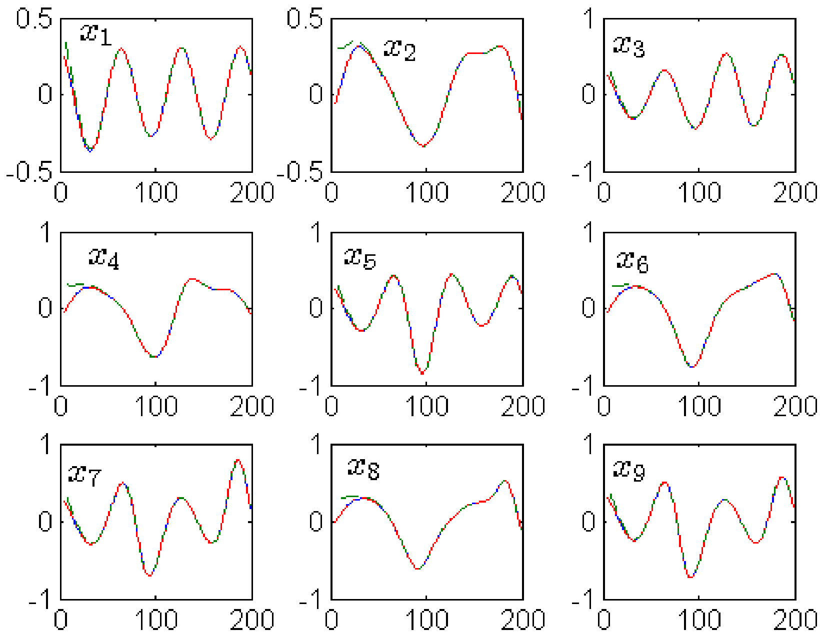
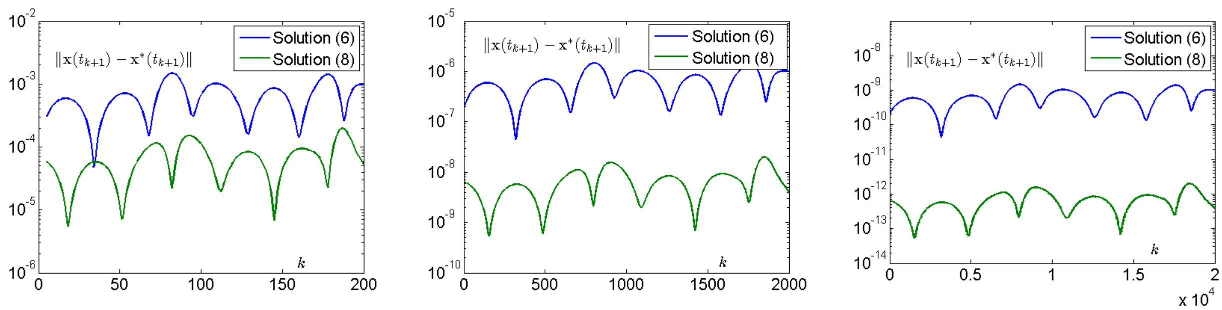
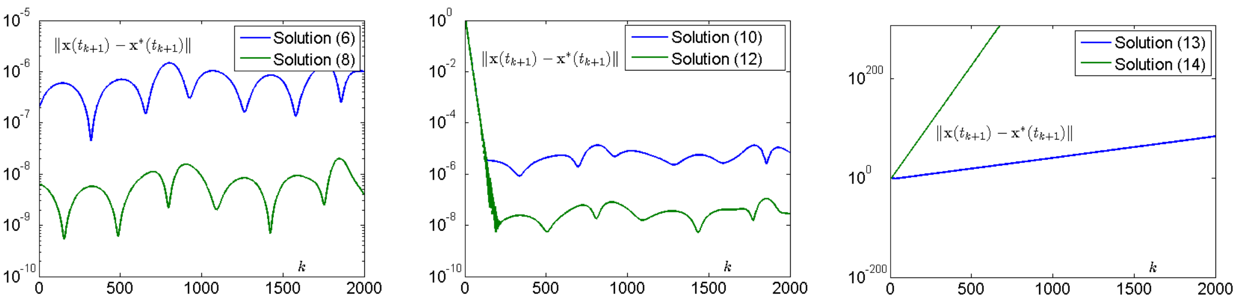


FIGURE 1. Nine time-varying elements of the vector-form solution $\mathbf{x}(t_{k+1})$ generated by the TVLES-SCF-I solution (6) (i.e., blue lines), conventional solution (10) (i.e., green lines) and theoretical solution (i.e., red lines).



(a) (b) (c)

FIGURE 2. Solution errors defined as $\|\mathbf{x}(t_{k+1}) - \mathbf{x}^*(t_{k+1})\|$ of TVLES-SCF-I (6) and TVLES-SCF-II solutions (8) with (a) $\tau = 0.1$ s, (b) $\tau = 0.01$ s, and (c) $\tau = 0.001$ s to solve the TVLES problem (1).



(a) (b) (c)

FIGURE 3. Comparisons of the errors of (a) TVLES-SCF solutions (6) and (8), (b) conventional stability-constraint solutions (10) and (12) satisfying stability-constraint, (c) conventional stability-constraint solutions (13) and (14) without satisfying stability-constraint when sampling gap $\tau = 0.01$ s.

Then, when defining the error of the TVLES-SCF-I solution (6) as $\|\mathbf{x}(t_{k+1}) - \mathbf{x}^*(t_{k+1})\|$, where $\|\cdot\|$ is the norm, we have

$$\|\mathbf{x}(t_{k+1}) - \mathbf{x}^*(t_{k+1})\| = \|\mathbf{O}(\tau^3)\| = O(\tau^3).$$

Thus, it is proven that the TVLES-SCF-I solution (6) to solve the TVLES problem (1) is effective, of which the solution error is $O(\tau^3)$. Similar to the above proven process, the TVLES-SCF-II solution (8) to solve the TVLES problem (1) is effective, of which the solution error is $O(\tau^4)$. □

VI. NUMERICAL EXPERIMENTS

In this section, some numerical experimental results are shown to substantiate the effectiveness and advantages of the

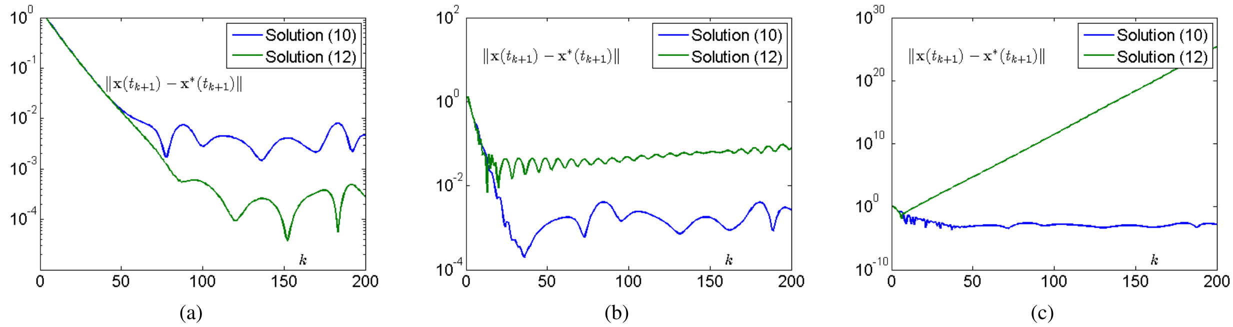


FIGURE 4. Errors of conventional stability-constraint solutions (10) and (12) when (a) $h = 0.1$, (b) $h = 0.3$ and (c) $h = 0.5$ with sampling gap $\tau = 0.1$ s.

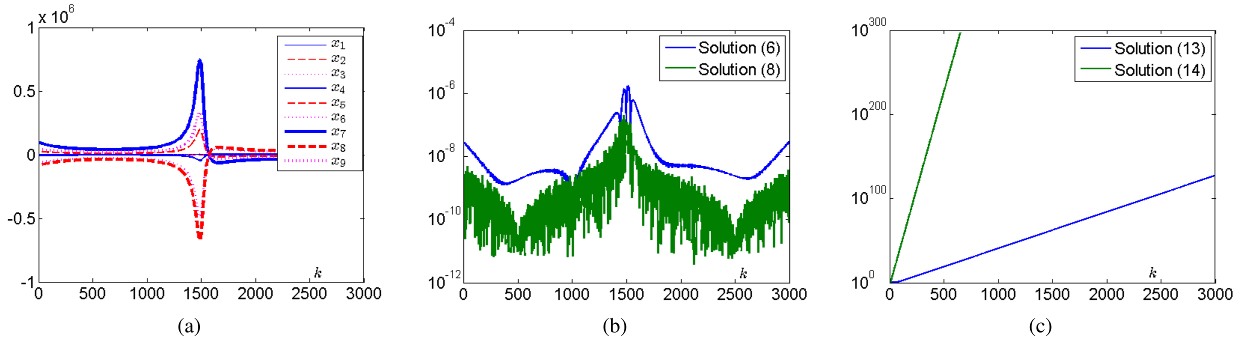


FIGURE 5. Comparisons of TVLES-SCF solutions (6) and (8) and conventional stability-constraint solutions (13) and (14) when sampling gap $\tau = 0.01$ s for solving TVLES with sometimes-ill-conditioned and even sometimes-singular matrix. (a) Solution trajectories of TVLES-SCF solution (6). (b) Residual errors of TVLES-SCF solutions (6) and (8). (c) Residual errors of conventional stability-constraint solutions (13) and (14).

proposed TVLES-SCF solutions. Specifically, we consider the TVLES problem (1) with the entries of $A(t_k) \in \mathbb{R}^{9\times 9}$ and $\mathbf{b}(t_k) \in \mathbb{R}^9$ being

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

and

$$b^i(t_k) = \begin{cases} \cos(t_k), & \text{when } i \text{ is odd} \\ \sin(0.5t_k), & \text{when } i \text{ is even,} \end{cases} \tag{15}$$

where $i, j = 1, 2, \cdots, 9$.

First, to substantiate the effectiveness of the proposed solutions, we conduct a numerical experiment in which we employ the TVLES-SCF-I solution (6) and conventional solution (10) to solve the above TVLES problem (15). The sampling gap is set as 0.1 s and the coefficient $h = 0.1$. The task duration is 20 s. The numerical result is shown in Fig. 1, which displays all nine time-varying elements of the vector-form solution $\mathbf{x}(t_{k+1})$ generated by the TVLES-SCF-I solution (6) (i.e., blue lines), conventional solution (10) (i.e., green lines) and theoretical solution (i.e., red lines). It can be observed that all solution trajectories generated by the TVLES-SCF-I solution (6) and theoretical path always overlap during the process. Thus, the effectiveness of the TVLES-SCF-I solution (6) to solve the TVLES problem (15) is substantiated. In addition, trajectories of conventional solution (10) also successfully track the theoretical ones after dozens of iterations.

Second, to substantiate the precision of the proposed TVLES-SCF solutions, i.e., the errors of the TVLES-SCF-I (6) and TVLES-SCF-II solutions (8) are $O(\tau^3)$ and $O(\tau^4)$, respectively, as proven in the theorem, we set the sampling gap τ to different values, i.e., $\tau = 0.1$, 0.01, and 0.001 s, and investigate the error of the TVLES-SCF-I (6) and TVLES-SCF-II solutions (8), which is defined as $\|\mathbf{x}(t_{k+1}) - \mathbf{x}^*(t_{k+1})\|$. Numerical results are shown in Fig. 2. When the sampling gap $\tau = 0.1, 0.01, 0.001$ s, the errors of the TVLES-SCF-I solution (6) are of order 10^{-3}, 10^{-6} and 10^{-9}, respectively. Similarly, the errors of the TVLES-SCF-II solution (8) are of order 10^{-4}, 10^{-8} and 10^{-12}, respectively. Thus, the precision of the proposed TVLES-SCF solutions is substantiated, which coincides with the theoretical results.

Third, to substantiate the advantages of the proposed TVLES-SCF solutions compared with conventional methods, as discussed in the above section, TVLES-SCF solutions (6) and (8), conventional stability-constraint solutions (10), (12), (13) and (14), including satisfying and not satisfying stability-constraint solutions, are all employed to solve the same problem with the same settings. In this group of numerical experiments, the sampling gap $\tau = 0.1$ s and coefficient $h = 0.1$. The task duration is 20 s. The errors of all solutions are shown in Fig. 3. When we compare the results in Figs. 3(a) and 3(b), it is observed that solutions (6) and (8) have high precision from start. In contrast, solutions (10) and (12) have high precision after nearly one hundred iterations. Thus, it is substantiated that TVLES-SCF solutions do not have convergence procedures and directly calculate and obtain future-instant results, while conventional solutions do not. When we compare the results in Figs. 3(a) and 3(c), TVLES-SCF solutions successfully solve the problem, while solutions (13) and (14) do not, although they are all based on nonstable discretization formulas such as central difference. Thus, it is substantiated that TVLES-SCF solutions are not bound by the constraint of zero stability, while conventional solutions fail to do so.

Besides, we investigate TVLES problem with sometimes-ill-conditioned and even sometimes-singular matrix as in Example 2 of [46]. Specifically, a 10-by-10 time-varying matrix is considered, of which the entries are the same as those of Hilbert matrix except the first entry. The first entry $a_k = 2 + \sin(0.1\pi t_k)$. The matrix sometimes becomes the Hilbert matrix, which is a classical ill-conditioned matrix. Numerical results are shown in Fig. 5, which shows residual errors defined as $\|A(t_k)\mathbf{x}(t_k) - \mathbf{b}(t_k)\|$ of TVLES-SCF solutions (6) and (8) and conventional stability-constraint solutions (13) and (14) when sampling gap $\tau = 0.01$ s as well as solution trajectories of TVLES-SCF solution (6). From this figure, TVLES-SCF solutions still have relatively good performances.

VII. CONCLUSION

In this work, stability-constraint-free solutions have been proposed to solve time-varying linear equation systems. These solutions are based on a new method, i.e., theoretical solution decomposition, instead of traditional zeroing neural dynamics. The proposed solutions have some advantages compared with traditional zeroing neural dynamics solutions, which have been presented in this paper. Comparative numerical experiments have been conducted, and the corresponding results substantiate the effectiveness and advantages. Besides, our method may be effective for other time-varying problems, which is our future research direction.

REFERENCES

[1] L. Xiao, S. Li, K. Li, L. Jin, and B. Liao, "Co-design of finite-time convergence and noise suppression: A unified neural model for time varying linear equations with robotic applications," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 12, pp. 5233–5243, Dec. 2020.

[2] L. Jin, S. Li, B. Hu, M. Liu, and J. Yu, "A noise-suppressing neural algorithm for solving the time-varying system of linear equations: A control-based approach," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 236–246, Jan. 2019.

[3] Z. Zhang, L. Zheng, T. Qiu, and F. Deng, "Varying-parameter convergent-differential neural solution to time-varying overdetermined system of linear equations," *IEEE Trans. Autom. Control*, vol. 65, no. 2, pp. 874–881, Feb. 2020.

[4] J. Li, Y. Shi, and H. Xuan, "Unified model solving nine types of time-varying problems in the frame of zeroing neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 1896–1905, May 2021.

[5] Z. Yin, K. Liu, Y. Chen, R. Xue, J. Li, and Y. Shi, "Performance analyses of four-instant discretization formulas with application to generalized-Sylvester-type future matrix equation," *IEEE Access*, vol. 7, pp. 152258–152266, 2019.

[6] J. Cai, Q. Feng, and D. Guo, "Discrete-time zeroing dynamics with quadruplicate error pattern for time-varying linear inequality," *IEEE Access*, vol. 9, pp. 7985–7993, 2021.

[7] Z. Zhang, L. Zheng, and T. Qiu, "A gain-adjustment neural network based time-varying underdetermined linear equation solving method," *Neurocomputing*, vol. 458, pp. 184–194, Oct. 2021.

[8] Z. Li, F. Xu, Q. Feng, J. Cai, and D. Guo, "The application of ZFD formula to kinematic control of redundant robot manipulators with guaranteed motion precision," *IEEE Access*, vol. 6, pp. 64777–64783, 2018.

[9] H. Liu, T. Wang, and D. Guo, "Design and validation of zeroing neural network to solve time-varying algebraic Riccati equation," *IEEE Access*, vol. 8, pp. 211315–211323, 2020.

[10] T. Wang and D. Guo, "Investigation on a new discrete-time synchronous motion planning scheme for dual-arm robot systems," *IEEE Access*, vol. 8, pp. 201545–201554, 2020.

[11] J. Li, X. Zhu, Y. Shi, J. Wang, and H. Guo, "Real-time robot manipulator tracking control as multilayered time-varying problem," *Appl. Math. Model.*, vol. 96, pp. 355–366, Aug. 2021.

[12] 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.

[13] J. Li, M. Mao, Y. Zhang, and B. Qiu, "Different-level algorithms for control of robotic systems," *Appl. Math. Model.*, vol. 77, pp. 922–933, Jan. 2020.

[14] W. Qi, X. Yang, J. H. Park, J. Cao, and J. Cheng, "Fuzzy SMC for quantized nonlinear stochastic switching systems with semi-Markovian process and application," *IEEE Trans. Cybern.*, early access, Apr. 19, 2021, doi: 10.1109/TCYB.2021.3069423.

[15] W. Qi, G. Zong, and S. F. Su, "Fault detection for semi-Markov switching systems in the presence of positivity constraints," *IEEE Trans. Cybern.*, early access, Aug. 3, 2021, doi: 10.1109/TCYB.2021.3096948.

[16] 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.

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

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

[19] Y. Lei, B. Liao, and J. Chen, "Comprehensive analysis of ZNN models for computing complex-valued time-dependent matrix inverse," *IEEE Access*, vol. 8, pp. 91989–91998, 2020.

[20] H.-H. Chen, M. T. Manry, and H. Chandrasekaran, "A neural network training algorithm utilizing multiple sets of linear equations," *Neurocomputing*, vol. 25, nos. 1–3, pp. 55–72, Apr. 1999.

[21] K. N. B. Murthy and C. S. R. Murthy, "A new Gaussian elimination-based algorithm for parallel solution of linear equations," *Comput. Math. Appl.*, vol. 29, no. 7, pp. 39–54, Apr. 1995.

[22] D. S. Parker, "Schur complements obey Lambek's categorial grammar: Another view of Gaussian elimination and LU decomposition," *Linear Algebra Appl.*, vol. 278, nos. 1–3, pp. 63–84, Jul. 1998.

[23] D. Jakovetić, N. Krejić, N. K. Jerinkić, G. Malaspina, and A. Micheletti, "Distributed fixed point method for solving systems of linear algebraic equations," *Automatica*, vol. 134, Dec. 2021, Art. no. 109924.

[24] M. Mimura and N. Tokushige, "Solving linear equations in a vector space over a finite field," *Discrete Math.*, vol. 344, no. 12, Dec. 2021, Art. no. 112603.

[25] J. Chen and Y. Zhang, "Discrete-time ZND models solving ALRMPC via eight-instant general and other formulas of ZeaD," *IEEE Access*, vol. 7, pp. 125909–125918, 2019.

[26] Z. Qi and Y. Zhang, "New models for future problems solving by using ZND method, correction strategy and extrapolation formulas," *IEEE Access*, vol. 7, pp. 84536–84544, 2019.

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

[28] L. Xiao, Y. He, B. Liao, and J. Dai, "An accelerated ZNN-based algorithm with piecewise time-varying parameters to solve time-variant linear equations," *J. Comput. Appl. Math.*, vol. 398, Dec. 2021, Art. no. 113665.

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

[30] Y. Zhang, Y. Ling, S. Li, M. Yang, and N. Tan, "Discrete-time zeroing neural network for solving time-varying Sylvester-transpose matrix inequation via exp-aided conversion," *Neurocomputing*, vol. 386, pp. 126–135, Apr. 2020.

[31] Y. Zhang, L. He, C. Hu, J. Guo, J. Li, and Y. Shi, "General four-step discrete-time zeroing and derivative dynamics applied to time-varying nonlinear optimization," *J. Comput. Appl. Math.*, vol. 347, pp. 314–329, Feb. 2019.

[32] Y. Zhang, L. Ming, H. Huang, J. Chen, and Z. Li, "Time-varying Schur decomposition via Zhang neural dynamics," *Neurocomputing*, vol. 419, pp. 251–258, Jan. 2021.

[33] M. Yang, Y. Zhang, and H. Hu, "Discrete ZNN models of Adams-Bashforth (AB) type solving various future problems with motion control of mobile manipulator," *Neurocomputing*, vol. 384, pp. 84–93, Apr. 2020.

[34] Y. Zhang, S. Li, and G. Geng, "Initialization-based k-winners-take-all neural network model using modified gradient descent," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 13, 2021, doi: 10.1109/TNNLS.2021.3123240.

[35] L. Jin, L. Wei, and S. Li, "Gradient-based differential neural-solution to time-dependent nonlinear optimization," *IEEE Trans. Autom. Control*, early access, Jan. 20, 2022, doi: 10.1109/TAC.2022.3144135.

[36] M. Liu, L. Chen, X. Du, L. Jin, and M. Shang, "Activated gradients for deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 1, 2021, doi: 10.1109/TNNLS.2021.3106044.

[37] Y. Shi, L. Jin, S. Li, J. Li, J. Qiang, and D. K. Gerontitis, "Novel discrete-time recurrent neural networks handling discrete-form time-variant multi-augmented Sylvester matrix problems and manipulator application," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 587–599, Feb. 2022.

[38] Y. Shi, W. Zhao, S. Li, B. Li, and X. Sun, "Novel discrete-time recurrent neural network for robot manipulator: A direct discretization technical route," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 15, 2021, doi: 10.1109/TNNLS.2021.3108050.

[39] L. Jia, L. Xiao, J. Dai, Z. Qi, Z. Zhang, and Y. Zhang, "Design and application of an adaptive fuzzy control strategy to zeroing neural network for solving time-variant QP problem," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 6, pp. 1544–1555, Jun. 2021.

[40] L. Xiao, Y. Zhang, J. Dai, Q. Zuo, and S. Wang, "Comprehensive analysis of a new varying parameter zeroing neural network for time varying matrix inversion," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 1604–1613, Mar. 2021.

[41] L. Xiao, Y. Cao, J. Dai, L. Jia, and H. Tan, "Finite-time and predefined-time convergence design for zeroing neural network: Theorem, method, and verification," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4724–4732, Jul. 2021.

[42] L. Xiao, J. Dai, L. Jin, W. Li, S. Li, and J. Hou, "A noise-enduring and finite-time zeroing neural network for equality-constrained time-varying nonlinear optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 8, pp. 4729–4740, Aug. 2021.

[43] W. Qi, Y. Hou, G. Zong, and C. K. Ahn, "Finite-time event-triggered control for semi-Markovian switching cyber-physical systems with FDI attacks and applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 6, pp. 2665–2674, Jun. 2021.

[44] J. Guo, B. Qiu, C. Hu, and Y. Zhang, "Discrete-time nonlinear optimization via zeroing neural dynamics based on explicit linear multi-step methods for tracking control of robot manipulators," *Neurocomputing*, vol. 412, pp. 477–485, Oct. 2020.

[45] J. Li, Y. Zhang, and M. Mao, "General square-pattern discretization formulas via second-order derivative elimination for zeroing neural network illustrated by future optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 891–901, Mar. 2019.

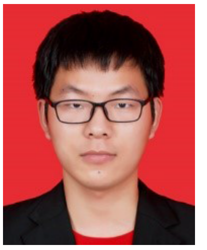
[46] J. Li, Y. Zhang, and M. Mao, "Five-instant type discrete-time ZND solving discrete time-varying linear system, division and quadratic programming," *Neurocomputing*, vol. 331, pp. 323–335, Feb. 2019.

[47] J. Li, M. Mao, F. Uhlig, and Y. Zhang, "A 5-instant finite difference formula to find discrete time-varying generalized matrix inverses, matrix inverses, and scalar reciprocals," *Numer. Algorithms*, vol. 81, no. 2, pp. 609–629, Jun. 2019.

[48] Y. Zhang, Y. Chou, J. Chen, Z. Zhang, and L. Xiao, "Presentation, error analysis and numerical experiments on a group of 1-step-ahead numerical differentiation formulas," *J. Comput. Appl. Math.*, vol. 239, pp. 406–414, Feb. 2013.

[49] M. Yang, Y. Zhang, and H. Hu, "Relationship between time-instant number and precision of ZeaD formulas with proofs," *Numer. Algorithms*, vol. 88, no. 2, pp. 883–902, Oct. 2021.

[50] 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.



JIAN LI received the B.E. degree in computer science and technology from Hainan University, Haikou, China, in 2014, and the Ph.D. degree in information and communication engineering from Sun Yat-sen University, Guangzhou, China, in 2019.

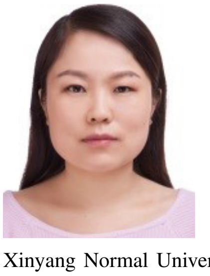
He is currently a Lecturer with the School of Computer and Information Technology, Xinyang Normal University, Xinyang, China. His current research interests include neural networks, numerical computation, and robotics.

JIANWEI FAN received the B.E. degree in electronical information science and technology from the Henan University of Science and Technology, Luoyang, China, in 2011, and the Ph.D. degree in pattern recognition and intelligent system from Xidian University, Xi'an, China, in 2017.

He is currently a Lecturer with the School of Computer and Information Technology, Xinyang Normal University, Xinyang, China. His current research interests include remote sensing image registration and feature extraction.

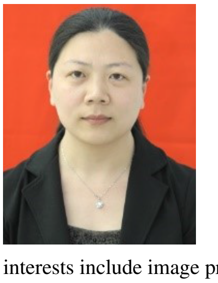
XINHUI ZHU received the B.E. degree in computer science and technology from Hainan University, Haikou, China, in 2014. She is currently pursuing the M.S. degree in computer software and theory with the School of Information and Technology, Xinyang Normal University, Xinyang, China.

Her current research interests include neural networks, numerical computation, and robotics.



YANGE SUN received the B.S. degree in computer science from Xinyang Normal University, Xinyang, China, in 2004, the M.S. degree in computer application technology from Central China Normal University, Wuhan, China, in 2007, and the Ph.D. degree in computer science from the Beijing Jiaotong University, Beijing, China, in 2019.

She is currently an Associate Professor with the School of Computer and Information Technology, Xinyang Normal University, Xinyang. Her research interests include data mining and machine learning.



YANLING LI received the B.S. degree in computer science and technology and the M.S. degree in basic mathematics from the Xinyang Normal University, Xinyang, China, in 1998 and 2003, respectively, and the Ph.D. degree in control science and engineering from the Huazhong University of Science and Technology, in 2009.

She is currently a Professor with the School of Computer and Information Technology, Xinyang Normal University, Xinyang. Her current research interests include image processing and machine learning.