IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 3, 2019, accepted June 17, 2019, date of publication June 24, 2019, date of current version July 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2924469

New Models for Future Problems Solving by Using ZND Method, Correction Strategy and Extrapolation Formulas

ZHIYUAN QI AND YUNONG ZHANG, (Member, IEEE)
School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China
Research Institute, Sun Yat-sen University, Shenzhen 518057, China
Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, Guangzhou 510006, China

Corresponding author: Yunong Zhang (zhynong@mail.sysu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61473323, and in part by the Shenzhen Science and Technology Plan Project under Grant JCYJ20170818154936083.

ABSTRACT Time-variant problems, which can be classified into future and non-future problems, are often encountered in academia and industry. In a future problem, we only know the information on the current and past time instants, and we have to acquire the next-time-instant solution before the next time instant arrives. Zeroing neural dynamics (ZND) and Zhang *et al.* discretization (ZeaD) formula group are two essential tools to build discrete-time ZND (DT-ZND) models for future problems solving. The former uses a systematical design formula to build a continuous-time ZND (CT-ZND) model, and the latter is used to transform the CT-ZND model into the discrete-time forms. Many DT-ZND models have been developed to solve various time-variant problems. In light of DT-ZND models and correction strategy, in this paper, we mainly focus on designing and building improved models for future problems solving. Based on the ZND method, extrapolation formulas, and correction steps, new models and corresponding computational algorithms are proposed to solve future optimization and future matrix inversion problems. The numerical experiments are also carried out to demonstrate the superiority of the proposed algorithms.

INDEX TERMS Future problem, zeroing neural dynamics (ZND), correction strategy, extrapolation formula.

I. INTRODUCTION

Time-variant problems can be divided into two categories, i.e., future problems and non-future problems. Future problems are often encountered in real-time application [1], [2], where the information of future time instants is generally unknown. In other words, the solutions of future problems should be generated or predicted in real time without using the information (e.g., the values of time-variant coefficients and time-derivative information) of future time instants. In contrast, in a non-future problem, some information of future time instants can be acquired. For example, if the specific formulation of the time-variant problem is given in advance, we can use the information of future time instants to help us estimate the next-time-instant solution. However, non-future problems appear less frequently in practical application.

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng H. Zhu.

Zeroing neural dynamics (ZND) is a commonly used method to solve time-variant problems [3]–[15] on the basis of recurrent neural network [16]–[19]. The core of the ZND method is the adoption of a systematic design formula, which guarantees that the defined error function converges rapidly to zero. By using this method, continuous-time ZND (CT-ZND) models can be established to solve various time-variant problems.

In order to implement a CT-ZND model on a digital device, the CT-ZND model should be transformed into a discrete-time form. Zhang *et al.* discretization (ZeaD) formula group is a powerful tool for the transformation [1]–[3], [20]–[24]. By using a ZeaD formula to discretize a CT-ZND model, a discrete-time ZND (DT-ZND) model can be obtained. After setting some parameters such as the sampling period and the stepsize appropriately, the resultant DT-ZND model can successfully track the theoretical solution of the future problem. From previous investigations [1]–[3], [21]–[24], DT-ZND models have excellent performance in solving different future problems.

Correction strategy is widely used in the field of numerical algorithms [25], [26]. By using this strategy, many algorithms have been improved to achieve better performance. Take the Euler method as an example [25]. This method is a first-order numerical algorithm to approximate the solution of a differential equation. It can be improved by taking the following two steps, i.e., prediction and correction. First, the original Euler method is used to predict a preliminary value. Then, by using the predicted value and the trapezoidal method, a corrected value can be obtained which is more accurate than the value generated by the original method. This improved algorithm is also known as the Heun method. The correction strategy has already been used to solve time-variant problems. For example, based on the prediction-correction method, the Newton trajectory tracking (NTT) model has been established for time-variant optimization problem solving [27]. Note that, both of the DT-ZND and NTT models can be adopted to solve time-variant optimization problem. However, the former can be used to solve future optimization problem, while the latter is mainly used in solving non-future optimization problem.

In the field of numerical algorithms, interpolation formulas are often used to predict or generate new data points within the domain of a given data set [25]. In contrast, extrapolation formulas are used to estimate the data points outside the domain [28], [29]. In this paper, in light of ZND method, correction strategy and extrapolation formulas, new improved models are established and proposed for future problems solving. The main contributions of this paper are listed as follows.

- The approach to building new models is proposed to solve future problems.
- New models for future optimization and matrix inversion problems solving are proposed, and the corresponding algorithms are also provided.
- Numerical experiments are carried out, and the superiority of the proposed models is demonstrated by comparison.

Notations: In this paper, symbols representing vectors and matrices are presented as m-dimension vectors and $m \times m$ matrices, respectively, and superscript $^{\mathrm{T}}$ denotes the transpose of a vector/matrix. Besides, we use $\|\cdot\|_2$ and $\|\cdot\|_{\mathrm{F}}$ to denote the two-norm of a vector and the Frobenius-norm of a matrix. The gradient of the function $f(\mathbf{x}, t)$ with respect to $\mathbf{x}$ is denoted by $\nabla_x f(\mathbf{x}, t)$. Similarly, the Hessian matrix of $f(\mathbf{x}, t)$ is denoted by $\nabla_{xx} f(\mathbf{x}, t)$, and $\nabla_{xt} f(\mathbf{x}, t)$ represents the partial derivative of the gradient $\nabla_x f(\mathbf{x}, t)$ with respect to t, i.e., $\nabla_{xt} f(\mathbf{x}, t) = \partial^2 f(\mathbf{x}, t)/(\partial \mathbf{x} \partial t)$.

II. ZND METHOD AND DT-ZND MODELS

In this section, the ZND method is briefly introduced. Future optimization and matrix inversion problems are used as examples to show the process of building CT-ZND models. ZeaD formulas are also introduced to obtain DT-ZND models. Besides, the corresponding computational algorithms are presented as well.

A. CT-ZND MODEL FOR FUTURE OPTIMIZATION PROBLEM SOLVING

The problem formulation of future optimization is stated as follows [22]–[24]:

$$\arg\min_{\mathbf{x}_{k+1}\in\mathbb{R}^m} f(\mathbf{x}_{k+1}, t_{k+1}) \in \mathbb{R}, \quad k = 0, 1, 2, \cdots, \tag{1}$$

where $f(\cdot)$ is assumed to be a smooth strongly convex function, $m \in \mathbb{N}^+$ denotes the length of the vector $\mathbf{x}_{k+1}$, and the subscript $_{k+1}$ denotes a variable at time instant $t = (k+1)g$, with $g \in \mathbb{R}^+$ representing the sampling period. The value of $\mathbf{x}_{k+1}$ should be estimated or predicted in the time interval $[t_k, t_{k+1})$. Thus, in the solution process, the information of future time instants (e.g., t_{k+1}) cannot be used because it is unknown/unavailable.

In order to solve future optimization problem (1), we consider the following continuous-time time-variant optimization problem at first, i.e.,

$$\arg\min_{\mathbf{x}(t)\in\mathbb{R}^m} f(\mathbf{x}(t), t) \in \mathbb{R}, \quad t \in \mathbb{R}^+. \tag{2}$$

When $f(\mathbf{x}(t), t)$ achieves its minimum at a certain time instant, we have $\nabla_x f(\mathbf{x}^*(t), t) = \mathbf{0}$, where $\mathbf{x}^*(t)$ denotes the theoretical solution. Therefore, we can define the following error function [22]–[24]:

$$\mathbf{e}(t) = \nabla_x f(\mathbf{x}(t), t) \in \mathbb{R}^m. \tag{3}$$

In order to make error function (3) converge rapidly to zero, the design formula of the ZND method is adopted [5], i.e.,

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

where $\lambda \in \mathbb{R}^+$ is a parameter related to convergence. Based on definition (3) and formula (4), the CT-ZND model for problem (2) is obtained [22]–[24], [30]–[32]:

$$\dot{\mathbf{x}}(t) = -\nabla_{xx}^{-1} f(\mathbf{x}(t), t) \left(\nabla_{xt} f(\mathbf{x}(t), t) + \lambda \nabla_x f(\mathbf{x}(t), t) \right). \tag{5}$$

We assume that $\nabla_{xx} f(\mathbf{x}(t), t)$ is nonsingular. The corresponding DT-ZND model to solve future optimization (1) is built based on (5), which is presented in a later subsection.

B. CT-ZND MODEL FOR FUTURE MATRIX INVERSION PROBLEM SOLVING

Future matrix inversion problem is described as follows: find or predict $X_{k+1} \in \mathbb{R}^{m\times m}$ in time interval $[t_k, t_{k+1})$ which satisfies

$$M_{k+1} X_{k+1} - I = 0 \in \mathbb{R}^{m\times m},$$

where M_{k+1} denotes the value of a smoothly time-variant nonsingular coefficient matrix $M(t)$ at time instant $t = (k+1)g$. Similar to the previous subsection, let us consider the corresponding continuous-time problem,

Algorithm 1 N-Point DT-ZND Algorithm for Future Optimization Problem Solving

Require: Randomly generated initial value $\mathbf{x}_0$, sampling period g, stepsize h_1 and h_2, coefficients ϑ_i ($i = 0, 1, \cdots, N-1$), initial objective function $f(\mathbf{x}, t_0)$

for $k = 0, 1, \cdots, N-3$ **do**

Generate $N-2$ additional initial values for DT-ZND model (10) by using the 2-point DT-ZND model

$$\mathbf{x}_{k+1} \doteq \mathbf{x}_k - \nabla_{xx}^{-1} f(\mathbf{x}_k, t_k)\big(g\nabla_{xt} f(\mathbf{x}_k, t_k) + h_1 \nabla_x f(\mathbf{x}_k, t_k)\big)$$

end for

for $k = N-2, N-1, \cdots$ **do**

Predict the value $\mathbf{x}_{k+1}$ before t_{k+1} arrives

$$\mathbf{x}_{k+1} \doteq -\frac{\nabla_{xx}^{-1} f(\mathbf{x}_k, t_k)}{\vartheta_0}\big(g\nabla_{xt} f(\mathbf{x}_k, t_k) + h_2 \nabla_x f(\mathbf{x}_k, t_k)\big) - \frac{1}{\vartheta_0}\sum_{i=1}^{N-1}(\vartheta_i \mathbf{x}_{k+1-i})$$

end for

Algorithm 2 N-Point DT-ZND Algorithm for Future Matrix Inversion Problem Solving

Require: Initial value X_0, sampling period g, stepsize h_1 and h_2, coefficients ϑ_i ($i = 0, 1, \cdots, N-1$), initial matrix $M(t_0)$

for $k = 0, 1, \cdots, N-3$ **do**

Generate $N-2$ additional initial values for DT-ZND model (11) by using the 2-point DT-ZND model

$$X_{k+1} \doteq X_k - \big(gX_k\dot{M}_kX_k + h_1 X_k(M_kX_k - I)\big)$$

end for

for $k = N-2, N-1, \cdots$ **do**

Predict the value X_{k+1} before t_{k+1} arrives

$$X_{k+1} \doteq -\frac{1}{\vartheta_0}\big(gX_k\dot{M}_kX_k + h_2 X_k(M_kX_k - I)\big) - \frac{1}{\vartheta_0}\sum_{i=1}^{N-1}(\vartheta_i X_{k+1-i})$$

end for

i.e., continuous-time time-variant matrix inversion problem. The problem formulation is [3], [13]–[15], [33]–[35]:

$$M(t)X(t) - I = 0 \in \mathbb{R}^{m\times m}, \quad t \in \mathbb{R}^+. \tag{6}$$

In terms of problem (6), many definitions of the error function are workable. In this paper, we choose the following definition:

$$E(t) = M(t) - X^{-1}(t). \tag{7}$$

The design formula (4) can be rewritten as [5]:

$$\dot{E}(t) = -\lambda E(t). \tag{8}$$

Considering that $\mathrm{d}(X^{-1}(t))/\mathrm{d}t = -X^{-1}(t)\dot{X}(t)X^{-1}(t)$, we have the following CT-ZND model on the basis of the definition (7) and formula (8) [3], [35]–[39]:

$$\dot{X}(t) = -X(t)\dot{M}(t)X(t) - \lambda X(t)(M(t)X(t) - I),$$

which is also known as the G-M dynamic system for time-variant matrix inversion.

C. DT-ZND MODELS AND ALGORITHMS

In this subsection, we transform the CT-ZND models obtained in the previous subsections into DT-ZND models. A discretization formula is needed in the process of transformation. According to previous studies, traditional numerical differentiation formulas are inappropriate to the time discretization of CT-ZND models. On the contrary, these models can be successfully discretized by using ZeaD formulas. The general computational form of an N-point ZeaD formula is [3], [20]–[23]:

$$\dot{\psi}_k \doteq \frac{1}{g}\sum_{i=0}^{N-1}(\vartheta_i \psi_{k+1-i}), \text{ with } \vartheta_0, \vartheta_{N-1} \neq 0, \tag{9}$$

where $\doteq$ denotes the computational assignment operation, and the coefficients ϑ_i ($i = 0, 1, \cdots, N-1$) satisfy certain rules. The precision of a ZeaD formula is related to the number of points. In recent years, the ZeaD-formula-related studies have been carried out in depth, many specific ZeaD formulas and even some general ZeaD formulas have been found and proposed. Table 1 presents some general ZeaD formulas. In fact, when $N = 2$, the general ZeaD formula is specific and unique, and it has been widely used as the Euler forward difference formula. By adopting the N-point ZeaD formula (9) to approximate $\dot{\mathbf{x}}(t)$ and $\dot{X}(t)$, CT-ZND models (2) and (6) can be transformed into DT-ZND models. The computational DT-ZND model for future optimization problem solving is [22]–[24], [30]–[32]:

$$\mathbf{x}_{k+1} \doteq -\frac{\nabla_{xx}^{-1} f(\mathbf{x}_k, t_k)}{\vartheta_0}\left(g\nabla_{xt} f(\mathbf{x}_k, t_k) + h\nabla_x f(\mathbf{x}_k, t_k)\right) - \frac{1}{\vartheta_0}\sum_{i=1}^{N-1}(\vartheta_i \mathbf{x}_{k+1-i}), \tag{10}$$

where $h = \gamma g \in \mathbb{R}^+$ is termed as the stepsize. Besides, the computational DT-ZND model for future matrix inversion problem solving is [3], [35], [36]:

$$X_{k+1} \doteq -\frac{1}{\vartheta_0}\big(gX_k\dot{M}_kX_k + hX_k(M_kX_k - I)\big) - \frac{1}{\vartheta_0}\sum_{i=1}^{N-1}(\vartheta_i X_{k+1-i}). \tag{11}$$

The corresponding algorithms of DT-ZND models for future optimization and matrix inversion problems solving are presented as Algorithms 1 and 2.

III. CORRECTION STRATEGY AND NEW MODELS

In this section, the NTT model is introduced to show the correction strategy. Based on the thought of correction and

TABLE 1. General ZeaD formulas of N points with $N = 2, 3, 4, 5$.

N	General ZeaD Formula	Reference
2	$\dot{\psi}_k \doteq \dfrac{\psi_{k+1} - \psi_k}{g}$	[25]
3	$\dot{\psi}_k \doteq \dfrac{(2\zeta+1)\psi_{k+1} - 4\zeta\psi_k + (2\zeta-1)\psi_{k-1}}{2g}, \zeta \in \left(0, \dfrac{1}{2}\right) \cup \left(\dfrac{1}{2} + \infty\right)$	[20]
4	$\dot{\psi}_k \doteq \dfrac{(2\zeta+1)\psi_{k+1} - 6\zeta\psi_k + (6\zeta-1)\psi_{k-1} - 2\zeta\psi_{k-2}}{2g}, \zeta \in (0, +\infty)$	[3]
5	$\dot{\psi}_k \doteq \dfrac{(6\zeta+2)\psi_{k+1} - (24\zeta-3)\psi_k + (36\zeta-6)\psi_{k-1} - (24\zeta-1)\psi_{k-2} + 6\zeta\psi_{k-3}}{6g}, \zeta \in \left(\dfrac{1}{12}, \dfrac{1}{6}\right)$	[23]

By setting different values of the parameter ζ, different specific ZeaD formulas can be obtained. Note that the 5-point general ZeaD formula shown in the above table is not the only one. Actually, it is a single-parameter general ZeaD formula, which is a special case of the corresponding multi-parameter 5-point general ZeaD formula [40].

Algorithm 3 NTT Algorithm for Non-Future Optimization Problem Solving

Require: Randomly generated initial value $\mathbf{x}_0$, sampling period g, number of iterations χ, initial objective function $f(\mathbf{x}, t_0)$

for $k = 0, 1, \cdots$ **do**
  Estimate the solution of next time instant $\breve{\mathbf{x}}_{k+1} \doteq \mathbf{x}_k - g\nabla_{xx}^{-1} f(\mathbf{x}_k, t_k)\nabla_{xt} f(\mathbf{x}_k, t_k)$
  Perform χ iterations to correct the estimated solution
  for $j = 0, 1, \cdots, \chi - 1$ **do**
    $\breve{\mathbf{x}}_{k+1} \doteq \breve{\mathbf{x}}_{k+1} - \nabla_{xx}^{-1} f(\breve{\mathbf{x}}_{k+1}, t_{k+1})\nabla_x f(\breve{\mathbf{x}}_{k+1}, t_{k+1})$
  end for
  $\mathbf{x}_{k+1} \doteq \breve{\mathbf{x}}_{k+1}$
end for

extrapolation formulas, new models are proposed to solve future optimization and matrix inversion problems.

A. NTT MODEL FOR NON-FUTURE OPTIMIZAITON

For better understanding of correction strategy, the NTT model is introduced to solve non-future optimization problem [27] (i.e., to solve the problem (1) when the future information is available). This model is built on the basis of prediction-correction method. In the prediction step of the NTT model, we estimate the solution of the next time instant using the following dynamic equation of the theoretical solution [27], i.e.,

$$\dot{\mathbf{x}}^*(t) = -\nabla_{xx}^{-1} f(\mathbf{x}^*(t), t)\nabla_{xt} f(\mathbf{x}^*(t), t). \tag{12}$$

By using the Euler forward difference formula to discretize (12), the solution can be roughly estimated, i.e.,

$$\breve{\mathbf{x}}_{k+1} \doteq \mathbf{x}_k - g\nabla_{xx}^{-1} f(\mathbf{x}_k, t_k)\nabla_{xt} f(\mathbf{x}_k, t_k).$$

Then, in the correction step, the above estimated value is corrected by using the Newton method [25]–[27]:

$$\mathbf{x}_{k+1} \doteq \breve{\mathbf{x}}_{k+1} - \nabla_{xx}^{-1} f(\breve{\mathbf{x}}_{k+1}, t_{k+1})\nabla_x f(\breve{\mathbf{x}}_{k+1}, t_{k+1}).$$

Note that the above formula shows the correction step of the NTT model by using one iteration. Actually, we can correct the estimated value by using multiple iterations to make it closer to the theoretical one. The complete algorithm for the NTT model is presented as Algorithm 3. We can find that the NTT model needs the information of the time instant t_{k+1}. Therefore, it is hard to directly employ the NTT model when future information is unknown. If we have to use this model to solve the future optimization problem, the required future information needs to be estimated.

TABLE 2. Specific extrapolation formulas of P points with $P = 2, 3, 4, 5$.

P	Extrapolation Formula
2	$\hat{\psi}_{k+1} \doteq 2\psi_k - \psi_{k-1}$
3	$\hat{\psi}_{k+1} \doteq 3\psi_k - 3\psi_{k-1} + \psi_{k-2}$
4	$\hat{\psi}_{k+1} \doteq 4\psi_k - 6\psi_{k-1} + 4\psi_{k-2} - \psi_{k-3}$
5	$\hat{\psi}_{k+1} \doteq 5\psi_k - 10\psi_{k-1} + 10\psi_{k-2} - 5\psi_{k-3} + \psi_{k-4}$

B. NEW MODELS FOR FUTURE PROBLEMS SOLVING

In light of the correction strategy and the designing process of the ZND and NTT models, let us consider how to build a more accurate model to solve future optimization problem.

Note that the initial value in the ZND model is not necessarily the theoretical one. By using the DT-ZND model, the residual error will quickly converge to a near-zero steady state, and the order of the residual error is closely related to the ZeaD formula applied. When the generated P-point sequence $\{\mathbf{x}_k, \mathbf{x}_{k-1}, \cdots, \mathbf{x}_{k-P+1}\}$ is sufficiently close to the theoretical solution sequence $\{\mathbf{x}_k^*, \mathbf{x}_{k-1}^*, \cdots, \mathbf{x}_{k-P+1}^*\}$, we can use a P-point numerical extrapolation formula to generate $\mathbf{x}_{k+1}$. The P-point numerical extrapolation formula we adopt has the following general form [28], [29] (with the truncation error considered):

$$\psi_{k+1} = \sum_{j=0}^{P-1}(\omega_j \psi_{k-j}) + O(g^P).$$

Algorithm 4 New Algorithm for Future Optimization Problem Solving

Require: Randomly generated initial value $\mathbf{x}_0$, sampling period g, stepsize h_1 and h_2, coefficients ϑ_i ($i = 0, 1, \cdots, N-1$) and ω_j ($j = 0, 1, \cdots, P-1$), threshold value η_e, initial objective function $f(\mathbf{x}, t_0)$

for $k = 0, 1, \cdots, N-3$ **do**

  Generate the corrected value $\check{\mathbf{x}}_k$

$$\check{\mathbf{x}}_k \doteq \mathbf{x}_k - \nabla_{xx}^{-1} f(\mathbf{x}_k, t_k) \nabla_x f(\mathbf{x}_k, t_k)$$

  Generate $N-2$ additional initial values by using the 2-point DT-ZND model

$$\mathbf{x}_{k+1} \doteq \mathbf{x}_k - \nabla_{xx}^{-1} f(\mathbf{x}_k, t_k) \big(g \nabla_{xt} f(\mathbf{x}_k, t_k) + h_1 \nabla_x f(\mathbf{x}_k, t_k)\big)$$

end for

for $k = N-2, N-1, \cdots$ **do**

  Generate the corrected value $\check{\mathbf{x}}_k$

$$\check{\mathbf{x}}_k \doteq \mathbf{x}_k - \nabla_{xx}^{-1} f(\mathbf{x}_k, t_k) \nabla_x f(\mathbf{x}_k, t_k)$$

  if $||\mathbf{e}_k||_2 > \eta_e$ **then**

    Predict the value $\mathbf{x}_{k+1}$ by using the N-point DT-ZND model

$$\mathbf{x}_{k+1} \doteq - \frac{\nabla_{xx}^{-1} f(\mathbf{x}_k, t_k)}{\vartheta_0} \big(g \nabla_{xt} f(\mathbf{x}_k, t_k) + h_2 \nabla_x f(\mathbf{x}_k, t_k)\big) - \frac{1}{\vartheta_0} \sum_{i=1}^{N-1} (\vartheta_i \mathbf{x}_{k+1-i})$$

  else

    Predict the value $\mathbf{x}_{k+1}$ by using the P-point extrapolation formula

$$\mathbf{x}_{k+1} \doteq \sum_{j=0}^{P-1} (\omega_j \check{\mathbf{x}}_{k-j})$$

  end if

end for

Table 2 presents some specific multi-point extrapolation formulas. By using an accurate and appropriate extrapolation formula, the estimated next-time-instant solution (i.e., $\mathbf{x}_{k+1}$) is obtained. It is worth mentioning that, if we keep on using the extrapolation formula to predict the next-time-instant solution, the computational error will accumulate. Therefore, a correction step is needed. Although we cannot use the future information to correct the predicted next-time-instant solution $\mathbf{x}_{k+1}$, we can use the current information (i.e., information of t_k) to correct the previously predicted value $\mathbf{x}_k$. The Newton method is used in the correction step [25]–[27], i.e.,

$$\check{\mathbf{x}}_k \doteq \mathbf{x}_k - \nabla_{xx}^{-1} f(\mathbf{x}_k, t_k) \nabla_x f(\mathbf{x}_k, t_k).$$

The corresponding algorithm of the new model to solve the future optimization problem is presented as Algorithm 4,

Algorithm 5 New Algorithm for Future Matrix Inversion Problem Solving

Require: Initial value X_0, sampling period g, stepsize h_1 and h_2, coefficients ϑ_i ($i = 0, 1, \cdots, N-1$) and ω_j ($j = 0, 1, \cdots, P-1$), threshold value η_e, initial matrix $M(t_0)$

for $k = 0, 1, \cdots, N-3$ **do**

  Generate the corrected value $\check{X}_k$

$$\check{X}_k = 2X_k - X_k M_k X_k$$

  Generate $N-2$ additional initial values by using the 2-point DT-ZND model

$$X_{k+1} \doteq X_k - \big(g X_k \dot{M}_k X_k + h_1 X_k (M_k X_k - I)\big)$$

end for

for $k = N-2, N-1, \cdots$ **do**

  Generate the corrected value $\check{X}_k$

$$\check{X}_k = 2X_k - X_k M_k X_k$$

  if $||E_k||_{\mathrm{F}} > \eta_e$ **then**

    Predict the value X_{k+1} by using the N-point DT-ZND model

$$X_{k+1} \doteq -\frac{1}{\vartheta_0} \big(g X_k \dot{M}_k X_k + h_2 X_k (M_k X_k - I)\big) - \frac{1}{\vartheta_0} \sum_{i=1}^{N-1} (\vartheta_i X_{k+1-i})$$

  else

    Predict the value X_{k+1} by using the P-point extrapolation formula

$$X_{k+1} \doteq \sum_{j=0}^{P-1} (\omega_j \check{X}_{k-j})$$

  end if

end for

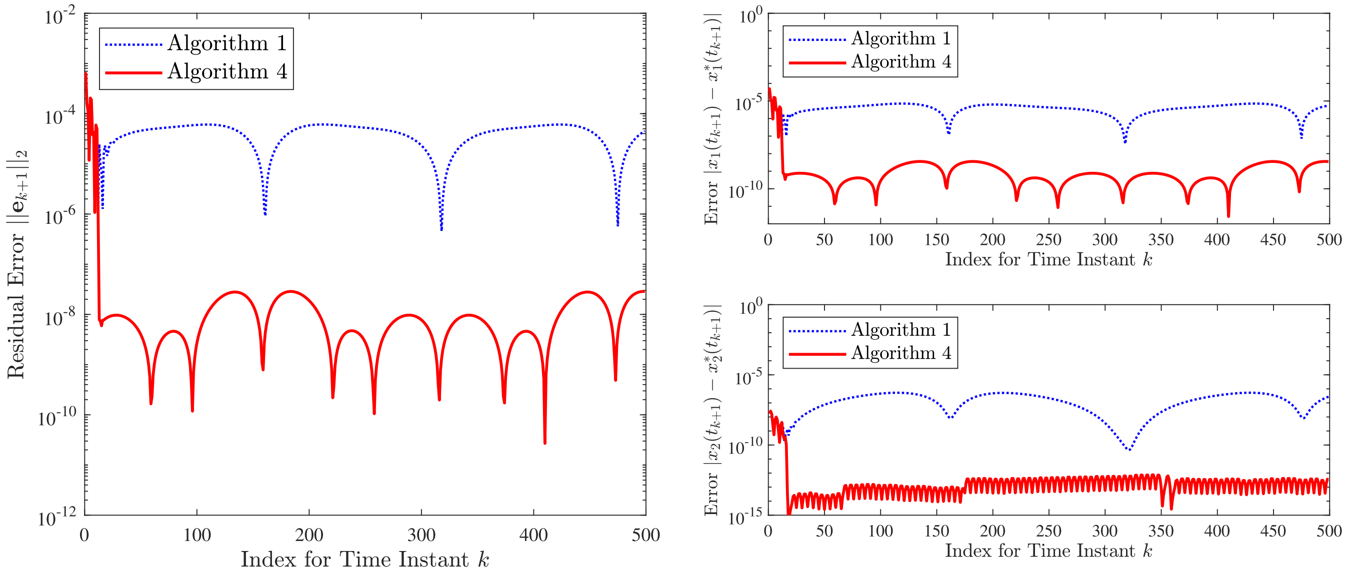
where $||\cdot||_2$ denotes the two-norm operator. In Algorithm 4, η_e is a pre-set threshold value. If the residual error is smaller than η_e, we assume that the generated P-point sequence is sufficiently close to the theoretical one, and a P-point extrapolation formula is then used to predict the next-time-instant solution.

Similarly, in terms of the future matrix inversion problem, we can use the following equation to correct the previously predicted value X_k before using the extrapolation formula:

$$\check{X}_k = M_k^{-1}.$$

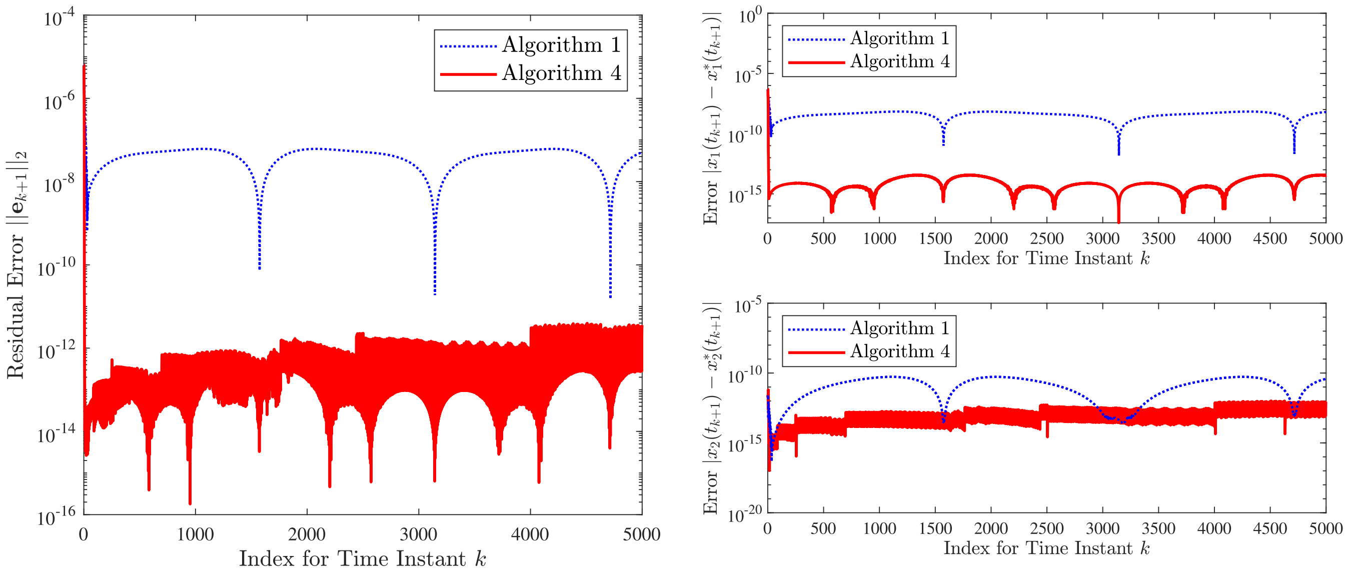
However, computing the inverse of a matrix is time-consuming, which is unsuitable for practical application. Therefore, the Newton method can be used for correction [25], [38], [39], [41], i.e.,

$$\check{X}_k = 2X_k - X_k M_k X_k.$$



(a) Residual error (b) Tracking error

FIGURE 1. **Algorithms 1 and 4 solving future optimization problem (13) with $h_1 = 1$, $h_2 = 0.240$, $\eta_e = 10^{-6}$, and $g = 0.01$ s.**

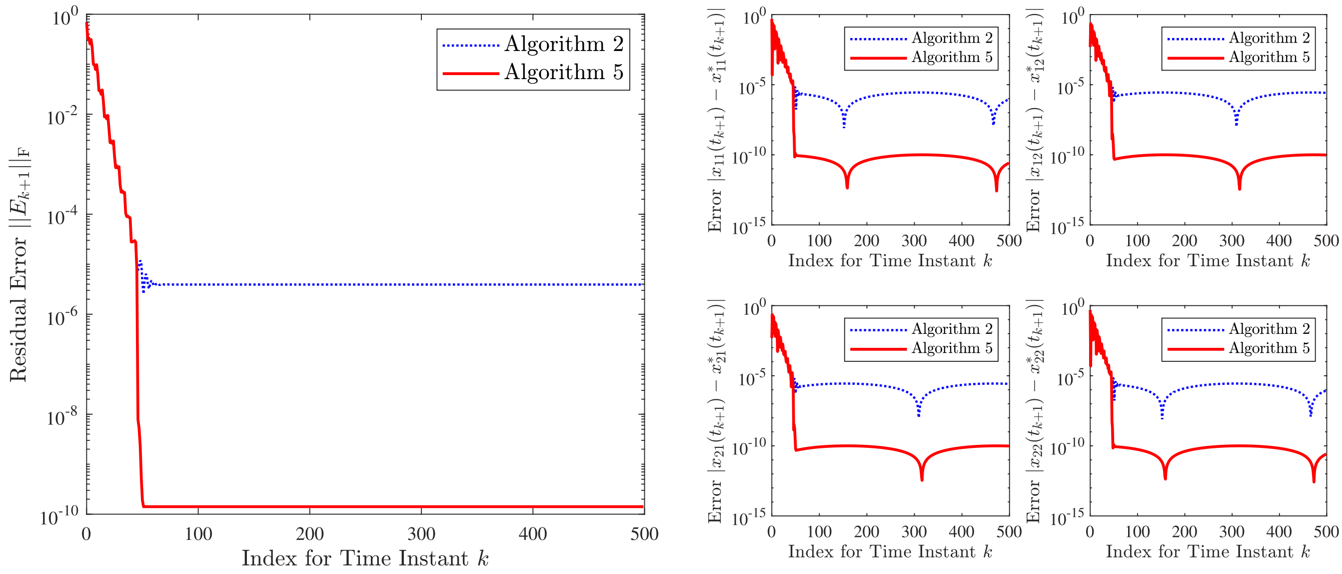


(a) Residual error (b) Tracking error

FIGURE 2. **Algorithms 1 and 4 solving future optimization problem (13) with $h_1 = 1$, $h_2 = 0.240$, $\eta_e = 10^{-6}$, and $g = 0.001$ s.**

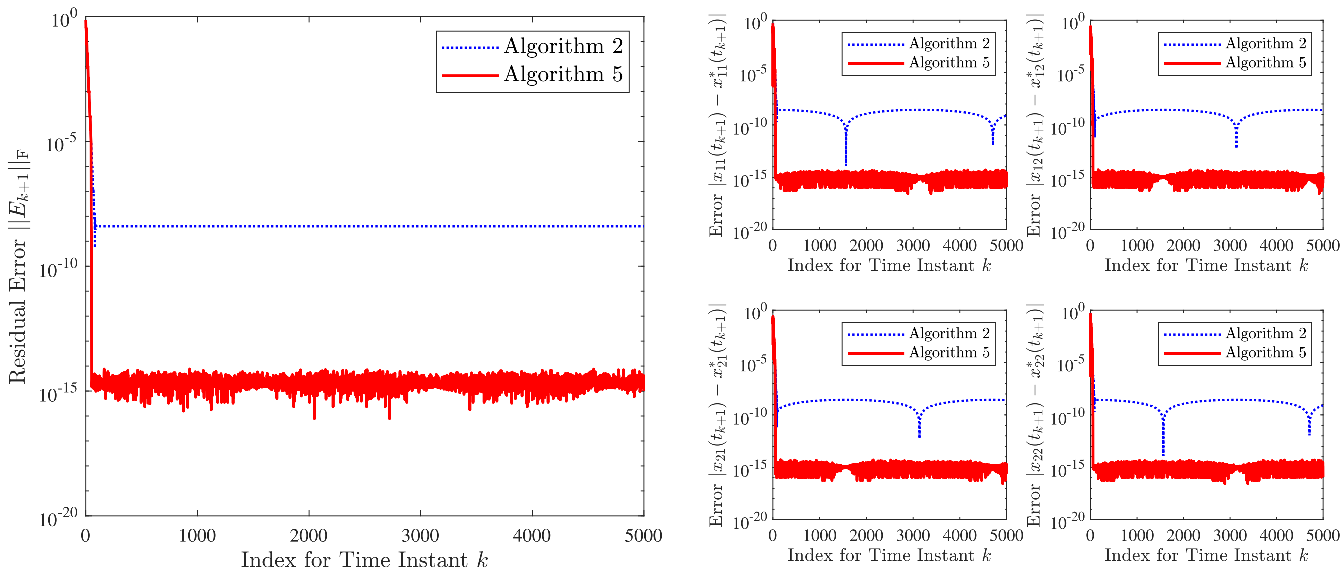
Algorithm 5 shows the corresponding computational steps of solving the future matrix inversion problem, where $||\cdot||_{\mathrm{F}}$ denotes the Frobenius-norm operator. The residual error at time instant t_k is defined as $||E_k||_{\mathrm{F}} = ||M_k X_k - I||_{\mathrm{F}}$, which is different from the definition (7) in that computing the inverse of the matrix X_k requires a large amount of computation and is thus not recommended.

Remarks: It is worth mentioning that the convergence of a DT-ZND model is related to the stepsize h. If the value of the stepsize is set improperly, the DT-ZND model will diverge in terms of the residual error. Besides, different values of the stepsize may bring different convergence rates for a DT-ZND model. Some stepsize-related investigations have already been carried out, such as the theoretical stepsize interval and the optimum of the stepsize (the optimum of the stepsize makes the DT-ZND model converge with the fastest rate in terms of the residual error) corresponding to a general/specific DT-ZND model [1], [2], [22], [24]. Note that, when the residual error is larger than the pre-set threshold value η_e, the proposed new models actually have the same operation with the corresponding DT-ZND models. Therefore, the conclusions of the previous stepsize-related investigations can be applicable to the new models.



(a) Residual error (b) Tracking error

FIGURE 3. Algorithms 2 and 5 solving future matrix inversion problem (13) with $h_1 = 1$, $h_2 = 0.240$, $\eta_e = 10^{-5}$, and $g = 0.01$ s.



(a) Residual error (b) Tracking error

FIGURE 4. Algorithms 2 and 5 solving future matrix inversion problem (13) with $h_1 = 1$, $h_2 = 0.240$, $\eta_e = 10^{-5}$, and $g = 0.001$ s.

IV. NUMERICAL EXPERIMENTS

In this section, numerical experiments are conducted to show the superiority of the new models. The new models are compared with the corresponding DT-ZND models for future optimization and matrix inversion problems solving.

A. FUTURE OPTIMIZATION PROBLEM

Let us consider the following time-variant function for optimization:

$$f(\mathbf{x}_k, t_k) = \exp\left(2\tanh\left(x_1(t_k)\right) - x_2(t_k) + t_k - \sin^2 t_k\right) + \frac{\exp\left(x_2(t_k) - t_k\right)}{2\tanh\left(x_1(t_k)\right) + \cos^2 t_k}. \tag{13}$$

The theoretical solution of the above future optimization problem is

$$\begin{aligned}\mathbf{x}^*_{k+1} &= [x^*_1(t_{k+1}), x^*_2(t_{k+1})]^{\mathrm{T}} \\ &= \left[\frac{1}{2}\ln\frac{2+\sin^2\ t_{k+1}}{2-\sin^2\ t_{k+1}}, t_{k+1}\right]^{\mathrm{T}},\end{aligned}$$

and $f(\mathbf{x}^*_{k+1}, t_{k+1}) = 2$. A specific ZeaD formula is needed to perform the DT-ZND and the new models. In this numerical experiment, the following 4-point ZeaD formula is adopted:

$$\dot{\psi}_k \doteq \frac{2\psi_{k+1} - 3\psi_k + 2\psi_{k-1} - \psi_{k-2}}{2g}, \tag{14}$$

which is also the second formula in Table 1 with $\zeta = 0.5$. In this situation, DT-ZND model (10) can be rewritten as

$$\mathbf{x}_{k+1} \doteq -\nabla_{xx}^{-1} f(\mathbf{x}_k, t_k)\,(g\nabla_{xt} f(\mathbf{x}_k, t_k) + h\nabla_x f(\mathbf{x}_k, t_k)) + \frac{3}{2}\mathbf{x}_k - \mathbf{x}_{k-1} + \frac{1}{2}\mathbf{x}_{k-2}.$$

According to previous investigations, the stepsize interval of the above DT-ZND model is $h \in (0, 1)$, and the optimum of the stepsize is 0.240 [2]. We perform Algorithms 1 and 4 with initial value $\mathbf{x}_0 = [0, 0]^{\mathrm{T}}$, $h_1 = 1$, $h_2 = 0.240$ and different values of g (i.e., $g = 0.01$ s and 0.001 s). Besides, in Algorithm 4, we use a 5-point extrapolation formula, i.e., the fourth formula shown in Table 2, and the error threshold η_e is set to 10^{-6}.

The corresponding numerical experimental results are shown in Figs. 1 and 2. Figs. 1(a) and 2(a) show that the steady-state residual error of Algorithm 4 is smaller than that of Algorithm 1. Additionally, as seen in Figs. 1(b) and 2(b), the next-time-instant solution generated by Algorithm 4 is closer to the theoretical solution than that by Algorithm 1, which means that Algorithm 4 has a better performance than Algorithm 1 in terms of tracking error.

B. FUTURE MATRIX INVERSION PROBLEM

In this subsection, Algorithms 2 and 5 are compared with each other for future matrix inversion problem solving. We use the following time-variant matrix for inversion [39]:

$$M_k = \begin{bmatrix} \sin t_k & \cos t_k \\ -\cos t_k & \sin t_k \end{bmatrix}. \tag{15}$$

In order to better compare the performance of Algorithms 2 and 5, the theoretical inversion of M_{k+1} is given by

$$X_{k+1}^* = M_{k+1}^{-1} = \begin{bmatrix} x_{11}^*(t_{k+1}) & x_{12}^*(t_{k+1}) \\ x_{21}^*(t_{k+1}) & x_{22}^*(t_{k+1}) \end{bmatrix} = \begin{bmatrix} \sin t_{k+1} & -\cos t_{k+1} \\ \cos t_{k+1} & \sin t_{k+1} \end{bmatrix}.$$

Algorithms 2 and 5 are performed to solve the matrix inversion problem of (15) by using the 4-point ZeaD formula (14), and the fourth extrapolation formula in Table 2 is used in Algorithm 5. Additionally, the initial value X_0 is set to $[0.5, -0.5; 0.5, 0.5]$. In this case, Figs. 3 and 4 show the corresponding numerical results with $h_1 = 1$, $h_2 = 0.240$, and $\eta_e = 10^{-5}$ for $g = 0.01$ s and 0.001 s. Specifically, Figs. 3(a) and 4(a) illustrate the curves of the residual error $||E_{k+1}||_{\mathrm{F}} = ||M_{k+1}X_{k+1} - I||_{\mathrm{F}}$. The curves of Algorithms 2 and 5 coincide with each other at the beginning in that $||E_k||_{\mathrm{F}}$ is larger than the threshold η_e, and these two algorithms perform the same operation in this situation. As shown, the steady-state residual error of Algorithm 5 is much less than that of Algorithm 2. Besides, the tracking error of each element of the generated matrix X_{k+1} is shown in Figs. 3(b) and 4(b). It is clear that the tracking error of Algorithm 5 is smaller than that of Algorithm 2. These numerical results demonstrate the validity and superiority of Algorithm 5.

V. CONCLUSION

In this paper, ZND method and ZeaD formulas have been introduced to build DT-ZND models for time-variant problems solving. The correction strategy has also been introduced. Then, based on ZND method, extrapolation formulas, and correction strategy, a new approach to building discrete-time models for future problems solving has been given. Two new models and the corresponding algorithms have been proposed to solve future optimization and matrix inversion problems. Comparative numerical experiments have been carried out to substantiate the validity and the superiority of these new models.

There are some future directions. First, the convergence of the proposed new models, such as the order of the residual error, can be investigated much further. Second, in this paper, we use future optimization and matrix inversion problems as examples to demonstrate the approach to building new models. Actually, for other future problems such as future linear system solving, nonlinear equation solving, and quadratic programming, the corresponding new models may also be built by using the similar steps. Third, practical physical experiments can also be carried out to further explore the superiority of the proposed new models.

REFERENCES

[1] J. Guo and Y. Zhang, "Stepsize interval confirmation of general four-step DTZN algorithm illustrated with future quadratic programming and tracking control of manipulators," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[2] Y. Zhang, H. Gong, M. Yang, J. Li, and X. Yang, "Stepsize range and optimal value for Taylor—Zhang discretization formula applied to zeroing neurodynamics illustrated via future equality-constrained quadratic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 959–966, Mar. 2019.

[3] C. Hu, X. Kang, and Y. Zhang, "Three-step general discrete-time Zhang neural network design and application to time-variant matrix inversion," *Neurocomputing*, vol. 306, pp. 108–118, Sep. 2018.

[4] D. Guo, F. Xu, L. Yan, Z. Nie, and H. Shao, "A new noise-tolerant obstacle avoidance scheme for motion planning of redundant robot manipulators," *Frontiers Neurorobot.*, vol. 12, pp. 1–13, Aug. 2018.

[5] Y. Zhang and D. Guo, *Zhang Functions and Various Models*. Berlin, Germany:, Springer, 2015.

[6] P. S. Stanimirović, V. N. Katsikis, and S. Li, "Hybrid GNN-ZNN models for solving linear matrix equations," *Neurocomputing*, vol. 316, pp. 124–134, Nov. 2018.

[7] P. S. Stanimirović, V. N. Katsikis, and S. Li, "Integration enhanced and noise tolerant ZNN for computing various expressions involving outer inverses," *Neurocomputing*, vol. 329, pp. 129–143, Feb. 2019.

[8] L. Ding, L. Xiao, K. Zhou, Y. Lan, and Y. Zhang, "A new RNN model with a modified nonlinear activation function applied to complex-valued linear equations," *IEEE Access*, vol. 6, pp. 62954–62962, 2018.

[9] L. Xiao, S. Li, F. Lin, Z. Tan, and A. H. Khan, "Zeroing neural dynamics for control design: Comprehensive analysis on stability, robustness, and convergence speed," *IEEE Trans. Ind. Inform.*, vol. 15, no. 5, pp. 2605–2616, May 2019.

[10] F. Charif, A. Benchabane, and A. Bennia, "A simplified architecture of the Zhang neural network for Toeplitz linear systems solving," *Neural Process. Lett.*, vol. 47, no. 2, pp. 391–401, Apr. 2018.

[11] L. Xiao, Z. Zhang, and S. Li, "Solving time-varying system of nonlinear equations by finite-time recurrent neural networks with application to motion tracking of robot manipulators," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[12] L. Jin, S. Li, B. Liao, and Z. Zhang, "Zeroing neural networks: A survey," *Neurocomputing*, vol. 267, pp. 597–604, Dec. 2017.

[13] L. Xiao, "A new design formula exploited for accelerating Zhang neural network and its application to time-varying matrix inversion," *Theor. Comput. Sci.*, vol. 647, pp. 50–58, Sep. 2016.
[14] 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.
[15] Z. Zhang, X. Deng, X. Qu, B. Liao, L. D. Kong, and L. Li, "A varying-gain recurrent neural network and its application to solving online time-varying matrix equation," *IEEE Access*, vol. 6, pp. 77940–77952, 2018.
[16] J. J. Hopfield and D. W. Tank, "Computing with neural circuits: A model," *Science*, vol. 233, pp. 625–633, Aug. 1986.
[17] S. Qin, J. Feng, J. Song, X. Wen, and C. Xu, "A one-layer recurrent neural network for constrained complex-variable convex optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 534–544, Mar. 2018.
[18] N. Liu and S. Qin, "A novel neurodynamic approach to constrained complex-variable pseudoconvex optimization," *IEEE Trans. Cybern.*, to be published.
[19] X. He, C. Li, T. Huang, C. Li, and J. Huang, "A recurrent neural network for solving bilevel linear programming problem," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 4, pp. 824–830, Apr. 2014.
[20] Y. Zhang, M. Zhu, C. Hu, J. Li, and M. Yang, "Euler-precision general-form of Zhang et al discretization (ZeaD) formulas, derivation, and numerical experiments," in *Proc. Chin. Control Decision Conf. (CCDC)*, Jun. 2018, pp. 6262–6267.
[21] 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," *J. Comput. Appl. Math.*, vol. 273, pp. 29–40, Jan. 2015.
[22] M. Sun and Y. Wang, "General five-step discrete-time Zhang neural network for time-varying nonlinear optimization," *Bull. Malaysian Math. Sci. Soc.*, vol. 3, pp. 1–20, May 2019.
[23] 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.
[24] D. Guo, X. Lin, Z. Su, S. Sun, and Z. Huang, "Design and analysis of two discrete-time ZD algorithms for time-varying nonlinear minimization," *Numer. Algorithms*, vol. 77, no. 1, pp. 23–36, 2018.
[25] J. H. Mathews and K. D. Fink, *Numerical Methods Using MATLAB*. Upper Saddle River, NJ, USA: Pearson Prentice-Hall, 2004.
[26] M. Fazlyab, S. Paternain, V. M. Preciado, and A. Ribeiro, "Prediction-correction interior-point method for time-varying convex optimization," *IEEE Trans. Autom. Control*, vol. 63, no. 7, pp. 1973–1986, Jul. 2018.
[27] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, "A class of prediction-correction methods for time-varying convex optimization," *IEEE Trans. Signal Process.*, vol. 64, no. 17, pp. 4576–4591, Sep. 2016.
[28] Y. Zhang, Z. Yang, J. Chen, Z. Qi, and G. Yang, "Presentation, derivation and numerical experiments of a group of extrapolation formulas," in *Proc. 9th Int. Conf. Inf. Sci. Technol.*, Hulunbuir, China, Aug. 2019, pp. 1–7.
[29] Z. Huang, "Multipoint smoothing extrapolation of measured data," *Radio Eng. China*, vol. 4, pp. 1–15, Mar. 1981.
[30] Y. Zhao and W. Lu, "Training neural networks with time-varying optimization," in *Proc. Int. Conf. Neural Netw.*, Oct. 1993, pp. 1693–1696.
[31] Y. Zhao and M. N. S. Swamy, "A novel technique for tracking time-varying minimum and its applications," in *Proc. IEEE Can. Conf. Elect. Comput. Eng.*, May 1998, pp. 910–913.
[32] Y. Zhao and C. Feng, "Time dependent optimization for information processing and its applications (I) basic concept and system identification," *J. Southeast Univ.*, vol. 29, no. 4, pp. 92–97, Jul. 1999.
[33] Z. Tan, Y. Hu, L. Xiao, and K. Chen, "Robustness analysis and robotic application of combined function activated RNN for time-varying matrix pseudo inversion," *IEEE Access*, vol. 7, pp. 33434–33440, 2019.
[34] L. Jin, S. Li, and B. Hu, "RNN models for dynamic matrix inversion: A control-theoretical perspective," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 189–199, Jan. 2018.
[35] Y. Zhang, W. Ma, and B. Cai, "From Zhang neural network to Newton iteration for matrix inversion," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 7, pp. 1405–1415, Jul. 2009.
[36] N. E. Getz and J. E. Marsden, "A dynamic inverse for nonlinear maps," in *Proc. 34th IEEE Conf. Decision Control*, Dec. 1995, pp. 4218–4223.
[37] P. S. Stanimirović and M. D. Petković, "Improved GNN models for constant matrix inversion," *Neural Process. Lett.*, vol. 28, pp. 1–19, Mar. 2019.
[38] Y. Zhang, D. Guo, C. Yi, L. Li, and Z. Ke, "More than Newton iterations generalized from Zhang neural network for constant matrix inversion aided with line-search algorithm," in *Proc. IEEE ICCA*, Jun. 2010, pp. 399–404.
[39] 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.
[40] 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.
[41] L. Jin, S. Li, H. Wang, and Z. Zhang, "Nonconvex projection activated zeroing neurodynamic models for time-varying matrix pseudoinversion with accelerated finite-time convergence," *Appl. Soft Comput.*, vol. 62, pp. 840–850, Jan. 2018.

ZHIYUAN QI is currently pursuing the B.E. degree in software engineering with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China.

His current research interests include time-varying problems solving, neural networks, and intelligent computing. His webpage is now available at http://www.qizhy.com.

YUNONG ZHANG (S'02–M'03) received the B.S. degree in industrial electrical automation from the Huazhong University of Science and Technology, Wuhan, China, in 1996, the M.S. degree in control theory and control engineering from the South China University of Technology, Guangzhou, China, in 1999, and the Ph.D. degree in mechanical and automation engineering from The Chinese University of Hong Kong, Hong Kong, in 2003.

He is currently a Professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou. Before joining Sun Yat-sen University, in 2006, he has been with the National University of Singapore, the University of Strathclyde, and the National University of Ireland at Maynooth, since 2003. His main research interests include robotics, neural networks, computation, and optimization. His webpage is now available at http://sdcs.sysu.edu.cn/content/2477.