



General four-step discrete-time zeroing and derivative dynamics applied to time-varying nonlinear optimization

Yunong Zhang^{a,*}, Liu He^a, Chaowei Hu^{a,b}, Jinjin Guo^a, Jian Li^a, Yang Shi^a

^a School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China

^b College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China

ARTICLE INFO

Article history:

Received 3 November 2017

Received in revised form 4 July 2018

Keywords:

Time-varying nonlinear optimization

Derivative dynamics

Zeroing dynamics

Zhang et al. discretization

General ZeaD formula

ABSTRACT

Time-varying nonlinear optimization (TVNO) problems are considered as important issues in various scientific disciplines and industrial applications. In this paper, the continuous-time derivative dynamics (CTDD) model is developed for obtaining the real-time solutions of TVNO problems. Furthermore, aiming to remedy the weaknesses of CTDD model, a continuous-time zeroing dynamics (CTZD) model is presented and investigated. For potential digital hardware realization, by using bilinear transform, a general four-step Zhang et al discretization (ZeaD) formula is proposed and applied to the discretization of both CTDD and CTZD models. A general four-step discrete-time derivative dynamics (general four-step DTDD) model and a general four-step discrete-time zeroing dynamics (general four-step DTZD) model are proposed on the basis of this general four-step ZeaD formula. Further theoretical analyses indicate that the general four-step DTZD model is zero-stable, consistent and convergent with the truncation error of $O(g^4)$, which denotes a vector with every entries being $O(g^4)$ with g denoting the sampling period. Theoretical analyses also indicate that the maximal steady-state residual error (MSSRE) has an $O(g^4)$ pattern confirmed. The efficacy and accuracy of the general four-step DTDD and DTZD models are further illustrated by numerical examples.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

In numerous scientific disciplines and industrial applications, time-varying optimization problems have been widely encountered, and such problems are fundamental and essential [1–10]. Since nonlinear optimization is an important subtopic of optimization problems, researches on this are abundant and many of which have been applied in the engineering fields such as system control [11–13] and signal processing [14–17]. Various algorithms have been proposed and analyzed for nonlinear optimization problems solving [10–27]. For example, in [11], a decomposition scheme is proposed to solve parametric non-convex programs. Such a scheme consists of a fixed number of proximal linearized alternating minimizations and a dual update per time step. The proposed approach is attractive in a real-time distributed context and the performance of the optimality-tracking scheme can be enhanced via a continuation technique. In [13], a self-triggered algorithm is presented to solve a class of convex optimization problems with time-varying objective functions. The algorithm predicts the temporal evolution of the gradient by using known upper bounds on higher order derivatives of the objective function. The method guarantees convergence to arbitrarily small neighborhood of the optimal trajectory in finite time and without incurring Zeno behavior. In [17], based on prediction and correction steps, Simonetto et al. proposed two algorithms with a discrete

* Corresponding author.

E-mail address: zhynong@mail.sysu.edu.cn (Y. Zhang).

time-sampling scheme. Considering the correction step consists either of one or multiple gradient steps or Newton steps, the two algorithms are termed gradient trajectory tracking (GTT) and Newton trajectory tracking (NTT) algorithms. The two algorithms behave as $O(g^2)$ and in some cases as $O(g^4)$ with g denotes the sampling period. Furthermore, in [16], online algorithms are developed to track solutions of time-varying constrained optimization problems. Resembling workhorse Kalman filtering-based approaches for dynamical systems, the proposed methods involve prediction–correction steps to provably track the trajectory of the optimal solutions of time-varying convex problems and improve the convergence speed of existing prediction–correction methods when applied to unconstrained problems. However, most of these algorithms and methods are dedicated in solving static nonlinear optimization problems, which means that they may not possess the capability of handling time-varying nonlinear optimization (TVNO) problems [20–22]. The main difference between TVNO problems and static nonlinear optimization problems is evidently that TVNO problems change with time. This difference makes the time derivative play an important role in obtaining accurate real-time solutions for TVNO problems.

Traditional numerical algorithms such as the Newton–Raphson iteration (NRI) and other methods [10,18–20] are designed intrinsically for static optimization. Generally speaking, these methods are under the assumption that the optimization problems do not change during the computational time. Thus, the calculated solutions are directly used in the optimization problems after the calculation. Comparing to the traditional numerical algorithms, the neural dynamical approach is superior because of the potential real-time application advantages such as self-adaption, parallel processing, distributed storage and hardware applications [28,29]. In this paper, the derivative dynamics approach is exploited as an intuitive one while the zeroing dynamics approach is exploited for potential predictive power [30–32]. More specifically speaking, a continuous-time derivative dynamics (CTDD) model and a continuous-time zeroing dynamics (CTZD) model are generalized, developed and investigated for TVNO problems.

For potential digital hardware realization, it is necessary to discretize the continuous-time dynamics models. Fortunately, a new class of finite-difference methods and formulas termed Zhang et al. discretization (ZeaD) has been proposed, named and applied by Zhang et al. since 2014 after nearly 8-year search and preparation [33]. It is worth pointing out that ZeaD formulas can be used for stable, convergent and accurate discretization of neural dynamics (i.e., derivative dynamics and zeroing dynamics) and other continuous-time ordinary differential equation (ODE) systems [33].

The rest of this paper consists of five sections. Section 2 presents the general formulation of nonlinear optimization. Then the CTDD model is developed. Since the performance of CTDD model is related to the initial state and perturbations, the CTDD model may not solve TVNO problems in a robust manner. Aiming to remedy this weakness, a CTZD model is then developed. In Section 3, for potential digital hardware realization, a general four-step ZeaD formula is proposed for the discretization of the CTDD and CTZD models. The design procedure is also presented in Section 3. The design procedure guarantees that the general four-step ZeaD formula is zero-stable, consistent and convergent with the truncation error of $O(g^3)$. Based on the general four-step ZeaD formula, the CTDD and CTZD models, the general four-step discrete-time derivative dynamics (general four-step DTDD) model and discrete-time zeroing dynamics (general four-step DTZD) model are proposed, respectively. Then in Section 4, further theoretical analyses are conducted, and the characteristics of the proposed general four-step DTZD model are investigated. We further find out that the general four-step DTZD model has a truncation error of $O(g^4)$, which denotes a vector with every entries being $O(g^4)$, and that its maximal steady-state residual error (MSSRE) also has an $O(g^4)$ pattern. Furthermore, numerical results of TVNO examples illustrate the efficacy and accuracy of the proposed DTDD and DTZD models in Section 5. Finally in Section 6, the conclusion along with the final remarks of this paper is presented. Before proceeding, a summarization of the main contributions of this paper is worth listing out.

- (1) For the first time, by using bilinear transform, a general four-step ZeaD formula having $O(g^3)$ truncation error is proposed and investigated in this paper. Such a general four-step formula can be used for the stable, convergent and accurate discretization of neural dynamics and other continuous-time ODE systems.
- (2) By adopting the general four-step ZeaD formula to discretize CTDD and CTZD models, the general four-step DTDD and DTZD models are proposed in this paper. Both general four-step discrete-time dynamics models solve TVNO problems successfully. Note that the general four-step DTDD model is sensitive to the initial state and shows a residual error of $O(g^2)$, and the general four-step DTZD model is robust and has a residual error of $O(g^4)$.
- (3) The stability and convergence results of the general four-step DTZD model for TVNO problems solving are proved, with the MSSRE of the model having an $O(g^4)$ pattern.
- (4) Numerical experimental results synthesized by the general four-step DTDD and DTZD models are presented, which further substantiate the efficacy and accuracy of the general four-step DTDD and DTZD models. These results collaborate well with the theoretical analyses.

2. Problem formulation, continuous-time models and general ZeaD formula

In this section, the problem formulation of TVNO is presented for further investigation. Based on this problem formulation, the CTDD model and CTZD model are developed. Then, the general ZeaD formula is presented to lay the foundation of the general four-step ZeaD formula.

2.1. Problem formulation

The problem formulation of the TVNO is as follows:

$$\min_{\mathbf{x}(t) \in \mathbb{R}^n} \varphi(\mathbf{x}(t), t) \in \mathbb{R}, \quad t \in [0, t_f], \quad (1)$$

where the time-varying nonlinear mapping function $\varphi(\cdot, \cdot) : \mathbb{R}^n \times [0, t_f] \rightarrow \mathbb{R}$ should be second-order differentiable and bounded. The real-time optimization solution is denoted by $\mathbf{x}(t) \in \mathbb{R}^n$. Such solution is ought to be found in real time and guarantees that the TVNO function (1) achieves its minimum value for all time t .

In order to obtain the real-time solution of TVNO problem (1), we define a differentiable nonlinear mapping function as

$$\begin{aligned} \Psi(\mathbf{x}(t), t) &= \frac{\partial \varphi(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)} = \left[\frac{\partial \varphi}{\partial x_1}, \frac{\partial \varphi}{\partial x_2}, \dots, \frac{\partial \varphi}{\partial x_n} \right]^T \\ &= [\psi_1(\mathbf{x}(t), t), \psi_2(\mathbf{x}(t), t), \dots, \psi_n(\mathbf{x}(t), t)]^T \in \mathbb{R}^n, \end{aligned}$$

where $\partial \varphi / \partial x_i = \partial \varphi(\mathbf{x}(t), t) / \partial x_i(t) = \psi_i(\mathbf{x}(t), t)$, with $i = 1, 2, \dots, n$, and superscript T denotes the transpose operator of a matrix/vector. Furthermore, we define the time-varying set:

$$\Omega^* = \{\mathbf{x}^*(t) \mid \partial \varphi(\mathbf{x}^*(t), t) / \partial \mathbf{x}^*(t) = \mathbf{0}\},$$

for time $t \in [0, t_f]$. It is worth pointing out that, in order to obtain the desired solution $\mathbf{x}^*(t)$, which can guarantee the minimization of TVNO problem (1), $\Psi(\mathbf{x}(t), t)$ should be set to zero.

2.2. CTDD model

When considering of obtaining the real-time solution of TVNO problem (1), the derivative dynamics approach is an intuitive attempt. More specifically speaking, in order to achieve the desired solution $\mathbf{x}^*(t)$ that can maintain $\Psi(\mathbf{x}(t), t) = \mathbf{0}$, for time $t \in [0, t_f]$, the derivative of the gradient $\Psi(\mathbf{x}(t), t)$ with respect to time t should be zero, and we have

$$\begin{aligned} \frac{d\Psi(\mathbf{x}(t), t)}{dt} &= \frac{\partial \Psi(\mathbf{x}(t), t)}{\partial t} + \frac{\partial \Psi(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)} \frac{d\mathbf{x}(t)}{dt} \\ &= \dot{\Psi}_t(\mathbf{x}(t), t) + H(\mathbf{x}(t), t) \frac{d\mathbf{x}(t)}{dt} \\ &= \mathbf{0}, \end{aligned} \quad (2)$$

where the Hessian matrix $H(\mathbf{x}(t), t)$ and the time-derivative vector $\dot{\Psi}_t(\mathbf{x}(t), t)$ are defined respectively as follows:

$$\begin{aligned} H(\mathbf{x}(t), t) &= \begin{bmatrix} \frac{\partial \psi_1}{\partial x_1} & \frac{\partial \psi_1}{\partial x_2} & \dots & \frac{\partial \psi_1}{\partial x_n} \\ \frac{\partial \psi_2}{\partial x_1} & \frac{\partial \psi_2}{\partial x_2} & \dots & \frac{\partial \psi_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \psi_n}{\partial x_1} & \frac{\partial \psi_n}{\partial x_2} & \dots & \frac{\partial \psi_n}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 \varphi}{\partial x_1 \partial x_1} & \frac{\partial^2 \varphi}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 \varphi}{\partial x_1 \partial x_n} \\ \frac{\partial^2 \varphi}{\partial x_2 \partial x_1} & \frac{\partial^2 \varphi}{\partial x_2 \partial x_2} & \dots & \frac{\partial^2 \varphi}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \varphi}{\partial x_n \partial x_1} & \frac{\partial^2 \varphi}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 \varphi}{\partial x_n \partial x_n} \end{bmatrix} \in \mathbb{R}^{n \times n}, \\ \dot{\Psi}_t(\mathbf{x}(t), t) &= \frac{\partial \Psi(\mathbf{x}(t), t)}{\partial t} = \frac{\partial^2 \varphi(\mathbf{x}(t), t)}{\partial \mathbf{x}(t) \partial t} \in \mathbb{R}^n. \end{aligned}$$

Note that ψ_i and x_i denote the i th element of $\Psi(\mathbf{x}(t), t)$ and $\mathbf{x}(t)$, respectively. Assuming that Hessian matrix $H(\mathbf{x}(t), t)$ is nonsingular for any $t \in [0, t_f]$, the CTDD model for TVNO problem (1) can be formulated as [2,3,7–9]:

$$\dot{\mathbf{x}}(t) = -H^{-1}(\mathbf{x}(t), t) \frac{\partial^2 \varphi(\mathbf{x}(t), t)}{\partial \mathbf{x}(t) \partial t}, \quad \text{with } \mathbf{x}(0) = \mathbf{x}^*(0). \quad (3)$$

According to Lemma 1 [3], assume that Hessian matrix $H(\mathbf{x}(t), t)$ is positive definite, the solution of Eq. (3) corresponds to the real-time theoretical solution of TVNO problem (1).

Lemma 1. With $\mathbf{x}(t) \in \mathbb{R}^n$ and $n \geq 2$, assume that the n -variable bounded function $\varphi(\mathbf{x}(t), t)$ has the second order continuous partial derivatives in the neighborhood of point $\mathbf{p}^* = \mathbf{x}^*(t_p) = [x_1^*(t_p), \dots, x_n^*(t_p)]^T$ at time instant t_p , which satisfies $\partial \varphi(\mathbf{x}(t), t) / \partial x_i(t)|_{t=t_p, \mathbf{x}(t_p)=\mathbf{p}^*} = 0$, with $i = 1, 2, \dots, n$. If the corresponding Hessian matrix $H(\mathbf{p}^*, t_p)$ is positive definite, then $\varphi(\mathbf{x}(t), t)$ achieves its minimum at point $\mathbf{p}^*$ of time instant t_p .

Note that the CTDD model does not possess the nature of exploiting the error-feedback information throughout the real-time solution process. Moreover, supposing the initial state $\mathbf{x}(0)$ is not selected appropriately, more specifically speaking if $\Psi(\mathbf{x}(0), 0) \neq \mathbf{0}$, $\Psi(\mathbf{x}(t), t)$ would remain nonzero during the whole computing process. Furthermore, even if $\mathbf{x}(0)$ is accurately found, potential perturbations or computational errors may lead to a result that $\Psi(\mathbf{x}(t), t) \neq \mathbf{0}$ for all $t \geq 0$. Thus, in viewpoint of control, this nature of not utilizing error-feedback information may cost the CTDD model (3) its robustness over perturbations and computational errors [3]. Therefore, a new method utilizing both derivative information and error-feedback information simultaneously is required.

2.3. CTZD model

In this subsection, the formulation of CTZD model is presented. It is worth pointing out that, according to our previous work [31–34], when comparing with derivative dynamics models, zeroing dynamics models not only have higher accuracy, but also have better robustness for obtaining the real-time solution for TVNO problem (1), because it exploits error-feedback information as well as the time-derivative information [3].

For monitoring and controlling purposes during the solving process of TVNO problem (1), we firstly define a vector-valued indefinite error-function:

$$\mathbf{e}(t) = [e_1(t), e_2(t), \dots, e_n(t)]^T = \Psi(\mathbf{x}(t), t), \quad (4)$$

where $e_i(t) = \psi_i(\mathbf{x}(t), t)$, with $i = 1, 2, \dots, n$, denotes the i th element of $\mathbf{e}(t)$. It is evident that the convergence of solution $\mathbf{x}(t)$ toward the desired solution $\mathbf{x}^*(t)$ can be achieved when the error function $\mathbf{e}(t)$ converges to zero. In order to make every element of $\mathbf{e}(t)$ converge to zero, based on our previous work on zeroing dynamics, the CTZD design formula can be obtained as follows:

$$\frac{d\mathbf{e}(t)}{dt} = -\lambda\Phi(\mathbf{e}(t)), \text{ i.e., } \frac{d\Psi(\mathbf{x}(t), t)}{dt} = -\lambda\Phi(\Psi(\mathbf{x}(t), t)), \quad (5)$$

where the value of design parameter $\lambda > 0$ should be selected accordingly for numerical stability, since it is used to scale the convergence rate. Besides, $\Phi(\cdot): \mathbb{R}^n \rightarrow \mathbb{R}^n$ denotes a vector activation function array of dynamics and $\phi(\cdot)$ is used to denote the element of $\Phi(\cdot)$. It is worth mentioning that CTZD design formula (5) is asymptotically stable (specifically, exponentially stable), which is proven in our previous work [30–33]. Besides, generally speaking, any monotonically-increasing odd activation function $\phi(\cdot)$ can be used for constructing the dynamics model and different choices of activation function $\phi(\cdot)$ may lead to different convergence performance. In this paper, the activation function applied to construct the CTZD model is the simple linear activation function:

$$\phi_{\text{lin}}(u) = u.$$

After we expand CTZD design formula (5), the following differential equation is obtained:

$$H(\mathbf{x}(t), t)\dot{\mathbf{x}}(t) = -\lambda\Phi(\Psi(\mathbf{x}(t), t)) - \dot{\Psi}_t(\mathbf{x}(t), t). \quad (6)$$

Then, the above dynamics could be rewritten as below:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= -H^{-1}(\mathbf{x}(t), t) (\lambda\Phi(\Psi(\mathbf{x}(t), t)) + \dot{\Psi}_t(\mathbf{x}(t), t)) \\ &= -H^{-1}(\mathbf{x}(t), t) \left(\lambda\Phi \left(\frac{\partial\varphi(\mathbf{x}(t), t)}{\partial\mathbf{x}(t)} \right) + \frac{\partial^2\varphi(\mathbf{x}(t), t)}{\partial\mathbf{x}(t)\partial t} \right), \end{aligned} \quad (7)$$

and the CTZD model we finally obtained can be formulated as [2,3,7–9]:

$$\dot{\mathbf{x}}(t) = -H^{-1}(\mathbf{x}(t), t) \left(\lambda \frac{\partial\varphi(\mathbf{x}(t), t)}{\partial\mathbf{x}(t)} + \frac{\partial^2\varphi(\mathbf{x}(t), t)}{\partial\mathbf{x}(t)\partial t} \right), \quad (8)$$

where $\mathbf{x}(t)$ denotes the state of the dynamics and starts with randomly-generated initial condition $\mathbf{x}(0) \in \mathbb{R}^n$.

2.4. General n -step ZeaD formula

Based on our previous work [33], the general n -step ZeaD formula can be concluded as

$$\dot{f}_k = \frac{1}{g} \left(\sum_{i=1}^{n+1} a_i f_{k-n+i} \right) + O(g^p), \quad (9)$$

where n represents the amount of the steps of ZeaD formula (9); a_i denotes the coefficients, with $i = 1, 2, \dots, n+1$; g denotes the sampling period; $O(g^p)$ denotes the truncation error; f_k denotes the value of $f(t)$ at time instant $t_k = kg$, i.e., $f_k = f(t_k)$; $\dot{f}_k = \dot{f}(t_k)$ for simplicity; k denotes the updating index. For example, the general four-step ZeaD formula can be proposed as

$$\dot{f}_k = \frac{a_5 f_{k+1} + a_4 f_k + a_3 f_{k-1} + a_2 f_{k-2} + a_1 f_{k-3}}{g} + O(g^p); \quad (10)$$

however, the coefficients are remaining to be determined in Section 3.

3. General four-step ZeaD formula and discrete-time models

For potential digital hardware realization, it is necessary to discretize the CTDD model and CTZD model. In this section, a general four-step ZeaD formula is designed, and both general four-step DTDD model and DTZD model are proposed.

3.1. Definitions of convergence of discrete-time models

Before proceeding, it is necessary for us to list the following important definitions of convergence of discrete-time models. From these definitions, we acknowledge that a discrete-time model must satisfy convergent conditions.

Definition 1. An n -step method $\sum_{j=0}^n \alpha_j f_{k+j} = g \sum_{j=0}^n \beta_j s_{k+j}$ can be checked zero-stability by determining the roots of characteristic polynomial $\rho(\gamma) = \sum_{j=0}^n \alpha_j \gamma^j$. If the roots of $\rho(\gamma) = 0$ are such that all roots lie in the disk, i.e., $|\gamma| \leq 1$, with $|\gamma| = 1$ being simple, then the n -step method is zero-stable. In addition, the zero-stability is sometimes called Dahlquist stability or root stability.

Definition 2. An n -step method is said to be of consistency with order p , if its truncation error is $O(g^p)$ with $p > 0$ for the smooth exact solution.

Definition 3. An n -step method is convergent, i.e., $x_{[(t-t_0)/g]} \rightarrow x^*(t)$, for all $t \in [t_0, t_f]$, as $g \rightarrow 0$, if and only if the method is consistent and zero-stable (please see Definitions 1 and 2). In other words, consistency plus zero-stability leads to convergence. In particular, a zero-stable consistent method converges with the order of its truncation error.

3.2. General four-step ZeaD formula design

For the readability of this paper and the integrity of the structure, the important achievements during the design procedure of general four-step ZeaD formula are presented in the following theorem and corollary.

Theorem 1. The general four-step ZeaD formula can be proposed as

$$\dot{f}_k \doteq \frac{(a_1 + \frac{1}{3})f_{k+1} - (4a_1 - \frac{1}{2})f_k + (6a_1 - 1)f_{k-1} - (4a_1 - \frac{1}{6})f_{k-2} + a_1 f_{k-3}}{g}, \quad (11)$$

where symbol $\doteq$ denotes the computational assignment operation, and the effective domain of a_1 is $1/12 < a_1 < 1/6$. This general four-step formula is convergent with the truncation error of $O(g^3)$.

Proof. From Section 3.1, we can conclude that as an n -step method, formula (11) is convergent if it is both consistent and zero-stable. The design procedure of formula (11) is illustrated in the remaining part of this subsection.

Firstly, the m th-order Taylor expansion [34] can be presented as

$$f_{k+1} = f_k + g\dot{f}_k + \frac{g^2}{2}\ddot{f}_k + \cdots + \frac{g^m}{m!}f_k^{(m)} + O(g^{m+1}). \quad (12)$$

In order to obtain the general four-step ZeaD formula with the truncation error of $O(g^4)$, the following Taylor expansions of $f_{k+1}, f_{k-1}, f_{k-2}$ and f_{k-3} are given as

$$f_{k+1} = f_k + g\dot{f}_k + \frac{g^2}{2}\ddot{f}_k + \frac{g^3}{6}\dddot{f}_k + \frac{g^4}{24}\ddddot{f}_k + O(g^5), \quad (13)$$

$$f_{k-1} = f_k - g\dot{f}_k + \frac{g^2}{2}\ddot{f}_k - \frac{g^3}{6}\dddot{f}_k + \frac{g^4}{24}\ddddot{f}_k + O(g^5), \quad (14)$$

$$f_{k-2} = f_k - 2g\dot{f}_k + 2g^2\ddot{f}_k - \frac{4g^3}{3}\dddot{f}_k + \frac{2g^4}{3}\ddddot{f}_k + O(g^5) \quad (15)$$

and

$$f_{k-3} = f_k - 3g\dot{f}_k + \frac{9g^2}{2}\ddot{f}_k - \frac{9g^3}{2}\dddot{f}_k + \frac{27g^4}{8}\ddddot{f}_k + O(g^5). \quad (16)$$

By substituting (13)–(16) into (10), the following equation can be obtained:

$$b_0 f_k + b_1 g\dot{f}_k + b_2 g^2\ddot{f}_k + b_3 g^3\dddot{f}_k + b_4 g^4\ddddot{f}_k + O(g^5) = O(g^{p+1}), \quad (17)$$

where $b_0 = a_5 + a_4 + a_3 + a_2 + a_1$, $b_1 = a_5 - a_3 - 2a_2 - 3a_1 - 1$, $b_2 = a_5/2 + a_3/2 + 2a_2 + 9a_1/2$, $b_3 = a_5/6 - a_3/6 - 4a_2/3 - 9a_1/2$ and $b_4 = a_5/24 + a_3/24 + 2a_2/3 + 27a_1/8$. If $p = 4$ and general four-step ZeaD formula (10) is consistent and has a truncation error of $O(g^4)$, the conditions below must be satisfied:

$$\begin{cases} b_0 = a_5 + a_4 + a_3 + a_2 + a_1 = 0 \\ b_1 = a_5 - a_3 - 2a_2 - 3a_1 - 1 = 0 \\ b_2 = a_5/2 + a_3/2 + 2a_2 + 9a_1/2 = 0 \\ b_3 = a_5/6 - a_3/6 - 4a_2/3 - 9a_1/2 = 0 \\ b_4 = a_5/24 + a_3/24 + 2a_2/3 + 27a_1/8 = 0. \end{cases} \quad (18)$$

By solving (18), we obtain the solution as

$$\begin{cases} a_1 = -\frac{1}{12} \\ a_2 = \frac{1}{2} \\ a_3 = -\frac{3}{2} \\ a_4 = \frac{5}{6} \\ a_5 = \frac{1}{4}. \end{cases} \quad (19)$$

Therefore, we can obtain the characteristic polynomial of (10) as follows:

$$\rho(\gamma) = 3\gamma^4 + 10\gamma^3 - 18\gamma^2 + 6\gamma - 1 = 0, \quad (20)$$

of which the roots are -4.7028 , 1 , $0.1847 + 0.1917i$ and $0.1847 - 0.1917i$ with i denoting imaginary unit. It does not satisfy the zero-stability since one root lies outside of unit disk. Thus, there are no coefficients that can guarantee $p = 4$ and four-step ZeaD formula (10) has a truncation error of $O(g^4)$.

If $p = 3$ and general four-step ZeaD formula (10) has a truncation error of $O(g^3)$, the conditions below must be satisfied:

$$\begin{cases} b_0 = a_5 + a_4 + a_3 + a_2 + a_1 = 0 \\ b_1 = a_5 - a_3 - 2a_2 - 3a_1 - 1 = 0 \\ b_2 = a_5/2 + a_3/2 + 2a_2 + 9a_1/2 = 0 \\ b_3 = a_5/6 - a_3/6 - 4a_2/3 - 9a_1/2 = 0. \end{cases} \quad (21)$$

By computing the solution of (21), we obtain

$$\begin{cases} a_5 = a_1 + 3 \\ a_4 = -4a_1 + \frac{1}{2} \\ a_3 = 6a_1 - 1 \\ a_2 = -4a_1 + \frac{1}{6}. \end{cases} \quad (22)$$

Besides, from the above computation process, we obtain $b_4 = a_1 + 1/12$. Then Eq. (17) can be formulated as

$$\left(a_1 + \frac{1}{12}\right)g^4\ddot{f}_k + O(g^5) = O(g^4). \quad (23)$$

Evidently Eq. (23) holds true. Thus according to Definition 2 in Section 3.1, we obtain a general four-step ZeaD formula which is consistent and with the truncation error of $O(g^3)$.

Furthermore, according to Definition 3 in Section 3.1, we need to further investigate the zero-stability of general four-step ZeaD formula (10) to have a better understanding of its convergency. General four-step ZeaD formula (10) can be rewritten as

$$a_5f_{k+1} + a_4f_k + a_3f_{k-1} + a_2f_{k-2} + a_1f_{k-3} = g\dot{f}_k, \quad (24)$$

which gives us the characteristic polynomial of (24) as below according to Definition 1 in Section 3.1:

$$\rho(\gamma) = a_5\gamma^4 + a_4\gamma^3 + a_3\gamma^2 + a_2\gamma + a_1. \quad (25)$$

In order to investigate the roots of Eq. (25), by adopting bilinear transform $\gamma = (1 + \omega\tau/2)/(1 - \omega\tau/2)$ [35], the following equation can be obtained:

$$c_4\left(\frac{\omega\tau}{2}\right)^4 + c_3\left(\frac{\omega\tau}{2}\right)^3 + c_2\left(\frac{\omega\tau}{2}\right)^2 + c_1\left(\frac{\omega\tau}{2}\right) + c_0 = 0. \quad (26)$$

Table 1
Collection of four-step ZeaD formulas with various values of a_1 .

a_1	Formula
$1/12 < a_1 < 1/6$	$\dot{f}_k = \frac{(a_1 + \frac{1}{3})f_{k+1} - (4a_1 - \frac{1}{2})f_k + (6a_1 - 1)f_{k-1} - (4a_1 - \frac{1}{6})f_{k-2} + a_1f_{k-3}}{g}$
$1/11$	$\dot{f}_k = \frac{28f_{k+1} + 9f_k - 30f_{k-1} - 13f_{k-2} + 6f_{k-3}}{66g}$
$1/10$	$\dot{f}_k = \frac{13f_{k+1} + 3f_k - 12f_{k-1} - 7f_{k-2} + 3f_{k-3}}{30g}$
$1/9$	$\dot{f}_k = \frac{8f_{k+1} + f_k - 6f_{k-1} - 5f_{k-2} + 2f_{k-3}}{18g}$
$1/8$	$\dot{f}_k = \frac{11f_{k+1} - 6f_{k-1} - 8f_{k-2} + 3f_{k-3}}{24g}$
$1/7$	$\dot{f}_k = \frac{20f_{k+1} - 3f_k - 6f_{k-1} - 17f_{k-2} + 6f_{k-3}}{42g}$

After substituting solution (22) to Eq. (26), we obtain $c_0 = 0$, $c_1 = 2$, $c_2 = 4$, $c_3 = 2/3$ and $c_4 = 16a_1 - 4/3$, which means Eq. (26) can be reformulated as

$$\left(16a_1 - \frac{4}{3}\right) \left(\frac{\omega\tau}{2}\right)^4 + \frac{2}{3} \left(\frac{\omega\tau}{2}\right)^3 + 4 \left(\frac{\omega\tau}{2}\right)^2 + 2 \left(\frac{\omega\tau}{2}\right) + 0 = 0.$$

According to Routh's stability criterion [36], we can conclude that

$$\frac{1}{12} < a_1 < \frac{1}{6}.$$

Based on solution (22) and Taylor expansions (13) through (16), we have

$$\begin{aligned} & \frac{(a_1 + \frac{1}{3})f_{k+1} - (4a_1 - \frac{1}{2})f_k + (6a_1 - 1)f_{k-1} - (4a_1 - \frac{1}{6})f_{k-2} + a_1f_{k-3}}{g} \\ &= \frac{gf_k + (a_1 + \frac{1}{12})g^4f_k'''' + O(g^5)}{g} = \dot{f}_k + \left(a_1 + \frac{1}{12}\right)g^3f_k'''' + O(g^4). \end{aligned} \quad (27)$$

After rearranging Eq. (27), the general four-step ZeaD formula can be presented as

$$\begin{aligned} & \dot{f}_k \\ &= \frac{(a_1 + \frac{1}{3})f_{k+1} - (4a_1 - \frac{1}{2})f_k + (6a_1 - 1)f_{k-1} - (4a_1 - \frac{1}{6})f_{k-2} + a_1f_{k-3}}{g} \\ & \quad - \left(a_1 + \frac{1}{12}\right)g^3f_k'''' + O(g^4), \end{aligned} \quad (28)$$

or

$$\begin{aligned} & \dot{f}_k \\ &= \frac{(a_1 + \frac{1}{3})f_{k+1} - (4a_1 - \frac{1}{2})f_k + (6a_1 - 1)f_{k-1} - (4a_1 - \frac{1}{6})f_{k-2} + a_1f_{k-3}}{g} \\ & \quad + O(g^3), \end{aligned}$$

where $1/12 < a_1 < 1/6$. Apparently we can see that the truncation error of (28) is $-(a_1 + 1/12)g^3f_k'''' = O(g^3)$. In addition, the collection of four-step ZeaD formulas with different values of a_1 is shown in Table 1. Thus, the proof is completed. $\square$

Corollary 1. The truncation error of general four-step formula (11) becomes smaller when a_1 approaches toward $1/12$.

Proof. From Eq. (28), we can see it clearly that as the expression of truncation error being $-(a_1 + 1/12)g^3f_k''''$. Since the effective domain of parameter a_1 is $1/12 < a_1 < 1/6$, it is evident that when the value of a_1 approaches toward $1/12$, the truncation error of general four-step ZeaD formula (11) becomes smaller. Thus completes the proof. $\square$

3.3. General discrete-time solutions

Due to potential digital hardware realization such as digital computer and digital circuit realization, the development for discrete-time models is necessary. In this section, based on the CTDD model and CTZD model, both general four-step DTDD model and DTZD model are proposed by adopting general four-step ZeaD formula (11).

Firstly, let us consider the following discrete-time TVNO problem with computational time interval $[t_k, t_{k+1}) \subseteq [0, t_f]$:

$$\min_{\mathbf{x}_{k+1} \in \mathbb{R}^n} \varphi(\mathbf{x}_{k+1}, t_{k+1}) \in \mathbb{R}, \quad (29)$$

where $\varphi(\mathbf{x}_{k+1}, t_{k+1})$ is generated or measured from the smoothly time-varying signal $\varphi(\mathbf{x}(t), t)$ by sampling at time instant $t = (k+1)g$ (which is denoted as t_{k+1}). Besides, $g > 0$ and $k = 0, 1, 2, \dots$ denotes the updating index.

In the real-time solution process of TVNO problem (29), computation has to be performed based on the present and/or previous data. For example, at time instant t_k , during the computational time interval $[t_k, t_{k+1})$, we could only use known information such as $\mathbf{x}(t_k)$, $\varphi(\mathbf{x}(t_k), t_k)$ and its derivatives, instead of unknown information such as $\varphi(\mathbf{x}_{k+1}, t_{k+1})$ and its derivatives. Thus, the objective is by utilizing the present and/or previous data to find the unknown vector $\mathbf{x}_{k+1}$ of the unknown function $\varphi(\mathbf{x}_{k+1}, t_{k+1})$ during $[t_k, t_{k+1})$, such that (29) achieves its minimal value at each time instant. Note that, at the start of the computation with $t_0 = 0$, we could not compute $\mathbf{x}(0)$ based on previous data; thus $\mathbf{x}(0)$ can be randomly generated or directly set as $\mathbf{x}^*(0)$, where $\mathbf{x}^*(0)$ denotes the theoretical solution of (29) at start time instant t_0 .

3.3.1. General four-step DTDD model

Firstly in this subsection, in view of general four-step ZeaD formula (10) in an element-wise manner, the following formula can be obtained:

$$\begin{aligned} \dot{\mathbf{x}}_k &= \frac{(a_1 + \frac{1}{3})\mathbf{x}_{k+1} - (4a_1 - \frac{1}{2})\mathbf{x}_k + (6a_1 - 1)\mathbf{x}_{k-1} - (4a_1 - \frac{1}{6})\mathbf{x}_{k-2} + a_1\mathbf{x}_{k-3}}{g} \\ &\quad + \mathbf{O}(g^3). \end{aligned} \quad (30)$$

According to Section 2.2, after adopting formula (30) into CTDD model (3), we obtain

$$\begin{aligned} &\frac{(a_1 + \frac{1}{3})\mathbf{x}_{k+1} - (4a_1 - \frac{1}{2})\mathbf{x}_k + (6a_1 - 1)\mathbf{x}_{k-1} - (4a_1 - \frac{1}{6})\mathbf{x}_{k-2} + a_1\mathbf{x}_{k-3}}{g} \\ &\quad \doteq -H^{-1}(\mathbf{x}_k, t_k) \frac{\partial^2 \varphi(\mathbf{x}_k, t_k)}{\partial \mathbf{x}_k \partial t_k}, \end{aligned} \quad (31)$$

which can be formulated as

$$\begin{aligned} \mathbf{x}_{k+1} &\doteq \left(\frac{-4a_1 + \frac{1}{2}}{a_1 + \frac{1}{3}} \right) \mathbf{x}_k - \left(\frac{6a_1 - 1}{a_1 + \frac{1}{3}} \right) \mathbf{x}_{k-1} - \left(\frac{-4a_1 + \frac{1}{6}}{a_1 + \frac{1}{3}} \right) \mathbf{x}_{k-2} \\ &\quad - \left(\frac{a_1}{a_1 + \frac{1}{3}} \right) \mathbf{x}_{k-3} - \left(\frac{1}{a_1 + \frac{1}{3}} \right) gH^{-1}(\mathbf{x}_k, t_k) \dot{\Psi}_t(\mathbf{x}_k, t_k). \end{aligned} \quad (32)$$

Thus, the design procedure of DTDD model is completed.

3.3.2. General four-step DTZD model

According to Section 2.3, in order to obtain the accurate discretization, general four-step formula (30) is adopted to discretize CTZD model (8). Therefore, we have

$$\begin{aligned} &\frac{(a_1 + \frac{1}{3})\mathbf{x}_{k+1} - (4a_1 - \frac{1}{2})\mathbf{x}_k + (6a_1 - 1)\mathbf{x}_{k-1} - (4a_1 - \frac{1}{6})\mathbf{x}_{k-2} + a_1\mathbf{x}_{k-3}}{g} \\ &\quad + \mathbf{O}(g^3) = -H^{-1}(\mathbf{x}_k, t_k) \left(\lambda \frac{\partial \varphi(\mathbf{x}_k, t_k)}{\partial \mathbf{x}_k} + \frac{\partial^2 \varphi(\mathbf{x}_k, t_k)}{\partial \mathbf{x}_k \partial t_k} \right), \end{aligned} \quad (33)$$

which can be formulated as

$$\begin{aligned} \mathbf{x}_{k+1} &= - \left(\frac{-4a_1 + \frac{1}{2}}{a_1 + \frac{1}{3}} \right) \mathbf{x}_k - \left(\frac{6a_1 - 1}{a_1 + \frac{1}{3}} \right) \mathbf{x}_{k-1} \\ &\quad - \left(\frac{-4a_1 + \frac{1}{6}}{a_1 + \frac{1}{3}} \right) \mathbf{x}_{k-2} - \left(\frac{a_1}{a_1 + \frac{1}{3}} \right) \mathbf{x}_{k-3} \\ &\quad - \left(\frac{1}{a_1 + \frac{1}{3}} \right) gH^{-1}(\mathbf{x}_k, t_k) \left(\lambda \frac{\partial \varphi(\mathbf{x}_k, t_k)}{\partial \mathbf{x}_k} + \frac{\partial^2 \varphi(\mathbf{x}_k, t_k)}{\partial \mathbf{x}_k \partial t_k} \right) + \mathbf{O}(g^4), \end{aligned} \quad (34)$$

or

$$\begin{aligned} \mathbf{x}_{k+1} = & -\left(\frac{-4a_1 + \frac{1}{2}}{a_1 + \frac{1}{3}}\right)\mathbf{x}_k - \left(\frac{6a_1 - 1}{a_1 + \frac{1}{3}}\right)\mathbf{x}_{k-1} \\ & - \left(\frac{-4a_1 + \frac{1}{6}}{a_1 + \frac{1}{3}}\right)\mathbf{x}_{k-2} - \left(\frac{a_1}{a_1 + \frac{1}{3}}\right)\mathbf{x}_{k-3} \\ & - \left(\frac{1}{a_1 + \frac{1}{3}}\right)H^{-1}(\mathbf{x}_k, t_k)(h\Psi(\mathbf{x}_k, t_k) + g\dot{\Psi}_t(\mathbf{x}_k, t_k)) + \mathbf{O}(g^4), \end{aligned} \quad (35)$$

where design parameter $h = \lambda g$ is used to scale the convergence rate of general four-step DTZD model (35).

Thus, the design procedure of general four-step DTZD model is completed.

4. Further theoretical guarantee for general four-step DTZD model

Based on the three definitions listed in Section 3.1, we have the following theoretical results about general four-step DTZD model (35).

Theorem 2. *The proposed general four-step DTZD model (35) for time-varying nonlinear optimization is zero-stable, consistent and convergent with the truncation error of $\mathbf{O}(g^4)$ when $1/12 < a_1 < 1/6$.*

Proof. According to Section 3.3.2, general four-step DTZD model (35) can be written as

$$\begin{aligned} \mathbf{x}_{k+1} = & -\left(\frac{-4a_1 + \frac{1}{2}}{a_1 + \frac{1}{3}}\right)\mathbf{x}_k - \left(\frac{6a_1 - 1}{a_1 + \frac{1}{3}}\right)\mathbf{x}_{k-1} \\ & - \left(\frac{-4a_1 + \frac{1}{6}}{a_1 + \frac{1}{3}}\right)\mathbf{x}_{k-2} - \left(\frac{a_1}{a_1 + \frac{1}{3}}\right)\mathbf{x}_{k-3} \\ & - \left(\frac{1}{a_1 + \frac{1}{3}}\right)H^{-1}(\mathbf{x}_k, t_k)(h\Psi(\mathbf{x}_k, t_k) + g\dot{\Psi}_t(\mathbf{x}_k, t_k)) + \mathbf{O}(g^4), \end{aligned}$$

where apparently the truncation error is $\mathbf{O}(g^4)$. In addition, according to Definitions 1 and 2 in Section 3.1, based on the design procedure of general four-step ZeaD formula (10) and general four-step DTZD model (35), it is known that general four-step DTZD model (35) is zero-stable and consistent with the truncation error of $\mathbf{O}(g^4)$ when $1/12 < a_1 < 1/6$. Therefore, according to Definition 3 in Section 3.1, we know that general four-step DTZD model (35) is zero-stable, consistent and convergent with the truncation error of $\mathbf{O}(g^4)$ when $1/12 < a_1 < 1/6$. Thus, the proof is completed. $\square$

Theorem 3. *Consider discrete-time TVNO problem (1). The steady-state residual error (MSSRE) $\lim_{k \rightarrow \infty} \|\Psi(\mathbf{x}_{k+1}, t_{k+1})\|_2$ of general four-step DTZD model (35) is $\mathbf{O}(g^4)$, where $\|\cdot\|_2$ denotes the 2-norm of a vector.*

Proof. Let $\mathbf{x}_{k+1}^* \in \mathbb{R}^n$ denote the theoretical solution of TVNO problem (1). According to Lemma 1 from [3], we know that $\Psi(\mathbf{x}_{k+1}^*, t_{k+1}) = \mathbf{0}$. Furthermore, from Theorem 2, it can be concluded that $\mathbf{x}_{k+1} = \mathbf{x}_{k+1}^* + \mathbf{O}(g^4)$ when k is large enough. Therefore, the following equation can be obtained:

$$\lim_{k \rightarrow \infty} \|\Psi(\mathbf{x}_{k+1}, t_{k+1})\|_2 = \lim_{k \rightarrow \infty} \|\Psi(\mathbf{x}_{k+1}^* + \mathbf{O}(g^4), t_{k+1})\|_2.$$

By using Taylor expansion, we obtain

$$\begin{aligned} \lim_{k \rightarrow \infty} \|\Psi(\mathbf{x}_{k+1}, t_{k+1})\|_2 &= \|\Psi(\mathbf{x}_{k+1}^*, t_{k+1}) + H(\mathbf{x}_{k+1}^*, t_{k+1})\mathbf{O}(g^4) + \mathbf{O}(g^8)\|_2 \\ &= \|H(\mathbf{x}_{k+1}^*, t_{k+1})\mathbf{O}(g^4) + \mathbf{O}(g^8)\|_2. \end{aligned}$$

Considering that $H(\mathbf{x}_{k+1}^*, t_{k+1})$ is a constant matrix at a certain time instant, we further have

$$\lim_{k \rightarrow \infty} \|\Psi(\mathbf{x}_{k+1}, t_{k+1})\|_2 = \mathbf{O}(g^4).$$

Thus, we complete the proof. $\square$

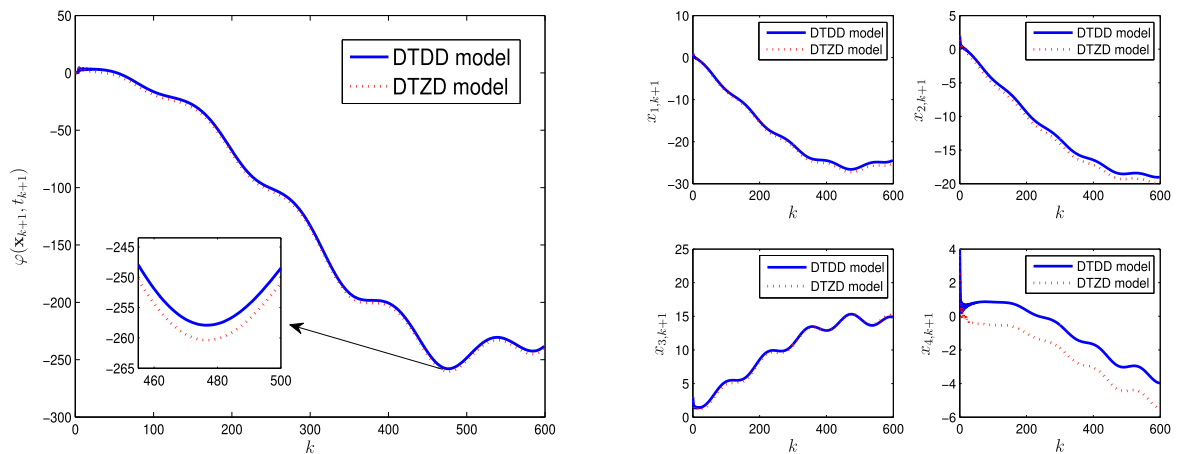


Fig. 1. Trajectories of $\varphi(\mathbf{x}_{k+1}, t_{k+1})$ and $\mathbf{x}_{k+1}$ of TVNO problem (36) synthesized by both general four-step DTDD model (32) and general four-step DTZD model (35) with $a_1 = 1/9$ and $h = 0.05$.

5. Numerical experiments and verifications

In order to verify the efficacy and accuracy of general four-step DTDD model (32) and general four-step DTZD model (35), numerical experiments are conducted. Without loss of generality, these experiments are carried out in MATLAB version 7.6.0 environment with a personal digital computer [34].

5.1. Example 1: TVNO problem solving by DTDD model and DTZD model

In this subsection, for illustrative and comparative purposes, one same example of TVNO problem (1) is synthesized by general four-step DTDD model (32) and general four-step DTZD model (35) simultaneously. Furthermore, in order to investigate the robustness of these models, several experiments are started with various initial states.

The continuous-time form of the example of TVNO problem (1) can be expressed as

$$\begin{aligned}
 & \min_{\mathbf{x}(t) \in \mathbb{R}^4} \varphi(\mathbf{x}(t), t) \\
 & = \left(x_1(t) + 10 \sin\left(\frac{\pi t}{40}\right) \right)^2 + \left(x_2(t) + \frac{t}{4} \right)^2 \\
 & \quad + \left(x_3(t) - \exp\left(-\frac{t}{4}\right) \right)^2 + 0.025(t-1)x_3(t)x_4(t) \\
 & \quad + \left(x_1(t) + \ln\left(\sin\left(\frac{\pi t}{40}\right) + 1\right) \right) \left(x_2(t) + \sin\left(\frac{t}{8}\right) + \cos\left(\frac{t}{8}\right) \right) \\
 & \quad - (x_1(t) + (x_1(t) + \sin(t))x_3(t) + \left(x_4(t) + \exp\left(-\frac{t}{4}\right) \right)^2,
 \end{aligned} \tag{36}$$

and, according to expression (29) in Section 3.3, the corresponding discrete-time form is expressed as

$$\min_{\mathbf{x}_{k+1} \in \mathbb{R}^n} \varphi(\mathbf{x}_{k+1}, t_{k+1}) \in \mathbb{R}.$$

Firstly, general four-step DTDD model (32) and general four-step DTZD model (35) are applied to solve TVNO problem (36) with $g = 0.05$ s, $h = 0.05$ and $a_1 = 1/9$. The corresponding results are presented in Fig. 1.

In Fig. 1(a), the red dotted curve and the blue solid curve denote the values of $\varphi(\mathbf{x}_{k+1}, t_{k+1})$ corresponding to the solutions of TVNO problem (36) synthesized by general four-step DTDD model (32) and general four-step DTZD model (35), respectively. Note that the numerical experiments of both DTDD model (32) and DTZD model (35) start with the same initial state vector $\mathbf{x}_0 = \mathbf{x}(0) = [1, 2, 3, 4]^T$ with the time duration being 30 s. Then according to our previous work [3,33], the state vectors of the second and the third sampling nodes are computed by using the Euler-type discrete-time model since the model requires just the state vector of one preceding sampling node. After the state vectors of the first four sampling nodes are acquired, both general four-step DTDD model (32) and general four-step DTZD model (35) are adopted to solve the TVNO problem (36).

From Fig. 1, it is easy for us to obtain an intuitive understanding that: when the initial state $\mathbf{x}(0)$ does not satisfy $\mathbf{x}(0) = \mathbf{x}^*(0)$, although both general four-step DTDD model (32) and general four-step DTZD model (35) are adequate for

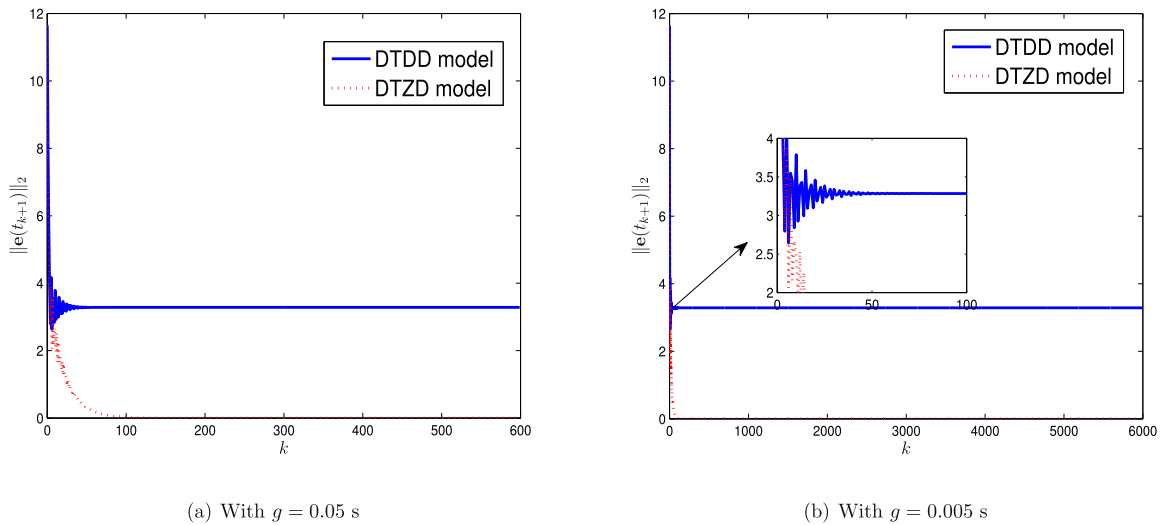


Fig. 2. Trajectories of residual error $\|\mathbf{e}(t_{k+1})\|_2$ of TVNO problem (36) synthesized by both general four-step DTDD model (32) and general four-step DTZD model (35) with $a_1 = 1/9$ and $h = 0.05$.

Table 2

MSSRE data synthesized by both general four-step DTZD model (35) and general four-step DTDD model (32) for solving TVNO problem (36) when initial states satisfy $\mathbf{x}(0) = \mathbf{x}^*(0)$ with $a_1 = 1/9$ and $h = 0.05$.

	$g = 0.05$ s	$g = 0.005$ s	$g = 0.0005$ s
DTZD	1.6144×10^{-5}	2.2714×10^{-9}	2.5334×10^{-13}
DTDD	6.9452×10^{-4}	4.8745×10^{-6}	6.6863×10^{-8}

solving TVNO problem (36), general four-step DTZD model (35) is more accurate. More specifically speaking, as shown in Fig. 1(a), the value of $\varphi(\mathbf{x}_{k+1}, t_{k+1})$ synthesized by general four-step DTZD model (35) remains to be smaller than the value of $\varphi(\mathbf{x}_{k+1}, t_{k+1})$ synthesized by general four-step DTDD model (32).

In order to have a better understanding of the real-time convergence of general four-step DTDD model (32) and general four-step DTZD model (35), we investigate the residual error $\|\mathbf{e}(t_{k+1})\|_2 = \|\partial\varphi(\mathbf{x}_{k+1}, t_{k+1})/\partial\mathbf{x}_{k+1}\|_2 = \|\Psi(\mathbf{x}_{k+1}, t_{k+1})\|_2$, as the optimization of TVNO problem (1) can be achieved when $\|\mathbf{e}(t_{k+1})\|_2$ is convergent to zero.

The trajectories of residual error $\|\mathbf{e}(t_{k+1})\|_2$ when the initial state $\mathbf{x}(0)$ does not satisfy $\mathbf{x}(0) = \mathbf{x}^*(0)$ are illustrated in Fig. 2.

As seen from Fig. 2(a), when $g = 0.05$ s, the residual error of general four-step DTZD model (35) converges to near zero, while the residual error of general four-step DTDD model (32) does not converge to zero and remains relatively large, which is being around 3.28. Moreover, from Fig. 2(b), we can see that when $g = 0.005$ s there is an evident enhancement of the performance of general four-step DTZD model (35), while the residual error of general four-step DTDD model (32) still does not converge to zero.

Secondly, to further investigate the performance of both discrete-time dynamics models when $\mathbf{x}(0) = \mathbf{x}^*(0)$, a few more experiments are conducted and the corresponding results are presented in Table 2 and Fig. 3. Specifically, Table 2 presents the corresponding maximum steady-state residual error (MSSRE). From Fig. 3, when the initial state is delicately selected to satisfy $\mathbf{x}(0) = \mathbf{x}^*(0)$, the residual error $\|\mathbf{e}(t_{k+1})\|_2$ of general four-step DTDD model (32) converges to near zero after a short time of oscillation, but general four-step DTZD model (35) shows to be more accurate. More specifically speaking, as seen from Fig. 3(a), the residual error $\|\mathbf{e}(t_{k+1})\|_2$ of general four-step DTDD model (32) is of order 10^{-4} when sampling period being $g = 0.05$ s, and in Fig. 3(b) the residual error is of order 10^{-6} when sampling period being $g = 0.005$ s. It is shown that the residual error $\|\mathbf{e}(t_{k+1})\|_2$ of general four-step DTDD model (32) is of order $O(g^2)$ when the initial state $\mathbf{x}(0)$ satisfies $\mathbf{x}(0) = \mathbf{x}^*(0)$.

As to the performance of general four-step DTZD model (35), from Table 2 we can see it clearly that the MSSRE of general four-step DTZD model (35) is of order 10^{-5} when $g = 0.05$ s, of order 10^{-9} when $g = 0.005$ s and of order 10^{-13} when $g = 0.0005$ s. We can draw the conclusion that the MSSRE of general four-step DTZD model (35) is of order $O(g^4)$, and this conclusion collaborates with Theorem 3 in Section 4. It can be also seen from Table 2 that the MSSRE of general four-step DTDD model (32) shows a pattern of $O(g^2)$.

Furthermore, from Figs. 1 and 2, it is evident that general four-step DTDD model (32) is sensitive to the initial state, while general four-step DTZD model (35) is robust.

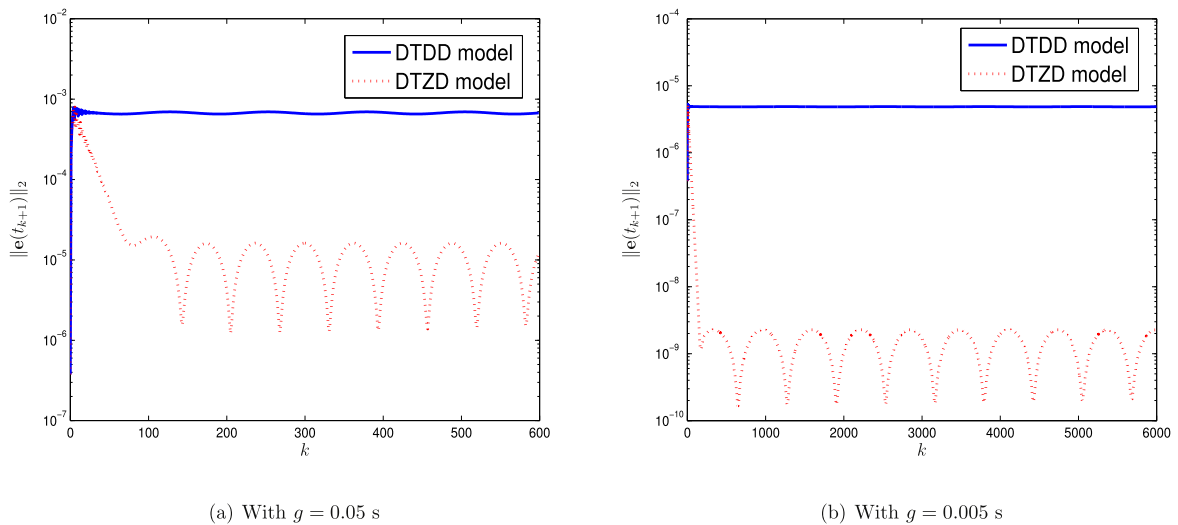


Fig. 3. Trajectories of residual error $\|e(t_{k+1})\|_2$ of TVNO problem (36) synthesized by both general four-step DTDD model (32) and general four-step DTZD model (35) when $\mathbf{x}(0) = \mathbf{x}^*(0)$ with $a_1 = 1/9$ and $h = 0.05$.

Table 3

MSSRE data of general four-step DTZD model (35) with different values of a_1 and $h = 0.05$.

a_1	$g = 0.05$ s	$g = 0.005$ s	$g = 0.0005$ s
1/7	3.6688×10^{-5}	2.8024×10^{-9}	3.0990×10^{-13}
15/112	1.9130×10^{-5}	2.6918×10^{-9}	3.0326×10^{-13}
1/11	1.5343×10^{-5}	2.1588×10^{-9}	2.6126×10^{-13}

Finally, to further investigate the influence of parameter a_1 over the performance of general four-step DTZD model (35), general four-step DTZD model (35) is exploited to solve TVNO problem (36) with the value of a_1 being $a_1 = 1/6$, $a_1 = 1/7$, $a_1 = 15/112$, $a_1 = 1/11$ and $a_1 = 1/12$. Without loss of generality, the numerical experiments begin with the same randomly generated initial state. The corresponding results are presented in Fig. 4 and Table 3.

From Fig. 4(a) and (b) we can obtain an intuitive conclusion that there exists notable enhancements of the performance of general four-step DTZD model (35) as a_1 approaches $1/12$. More specifically speaking, from both Fig. 4(a) and (b) we can see it clearly that although the residual errors of all three DTZD models have an $O(g^4)$ pattern, the DTZD model with $a_1 = 1/11$ has both the highest accuracy and convergence rate, while the DTZD model with $a_1 = 1/7$ has the lowest accuracy and convergence rate, and the performance of the DTZD model with $a_1 = 15/112$ is between these two DTZD models. From more detailed MSSRE data that are listed in Table 3, we can see that after all three DTZD models reach the steady state, there still exists a small difference between the MSSRE data of these three DTZD models. From Table 3, we can draw the conclusion that all three DTZD models have a residual error being the order of $O(g^4)$ and general four-step DTZD model (35) has a smaller MSSRE as a_1 approaches toward $1/12$. This conclusion collaborates with both Theorem 3 and Corollary 1. Furthermore, as shown in Fig. 4(c) and (d), general four-step DTZD model (35) does not converge when $a_1 = 1/6$ and $a_1 = 1/12$, which collaborates with Theorem 1 that the effective domain of parameter a_1 is $1/12 < a_1 < 1/6$.

5.2. Computational complexity

In this subsection, we present the computational complexity of general four-step DTDD model (32) and general four-step DTZD model (35) for solving TVNO problem (36), including computational complexity analyses and numerical comparisons.

For the convenience of comparisons on the performance of computational complexity, we present the classical Euler-type DTZD model for the time-varying nonlinear optimization [2,3] as

$$\mathbf{x}_{k+1} \doteq \mathbf{x}_k - H^{-1}(\mathbf{x}_k, t_k)(h\Psi(\mathbf{x}_k, t_k) + g\dot{\Psi}(\mathbf{x}_k, t_k)), \quad (37)$$

which is convergent with the truncation error of $O(g^2)$. For the n -dimension time-varying nonlinear optimization, $\mathbf{x}_k$, $\Psi(\mathbf{x}_k, t_k)$ and $\dot{\Psi}(\mathbf{x}_k, t_k)$ are n -dimension vectors, and $H(\mathbf{x}_k, t_k)$ is an n -dimension square matrix. Excluding the same matrix-inverse operation, the multiplications of Euler-type DTZD model (37), proposed DTDD model (32) and proposed DTZD model (35) are $n^2 + 2n$, $n^2 + 6n$ and $n^2 + 7n$ per updating, respectively. The additions of Euler-type DTZD model (37), proposed DTDD model (32) and proposed DTZD model (35) are $n^2 + n$, $n^2 + 3n$ and $n^2 + 4n$ per updating, respectively. Besides, the

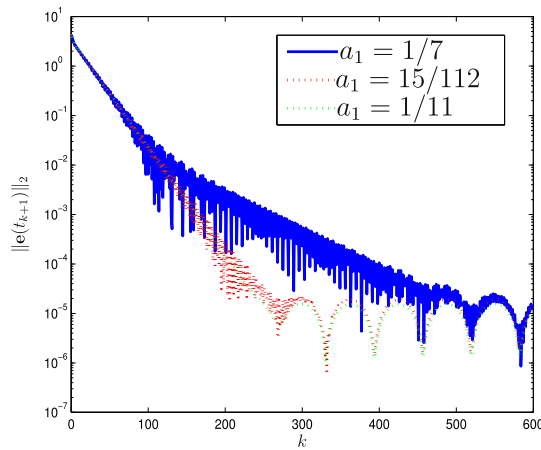
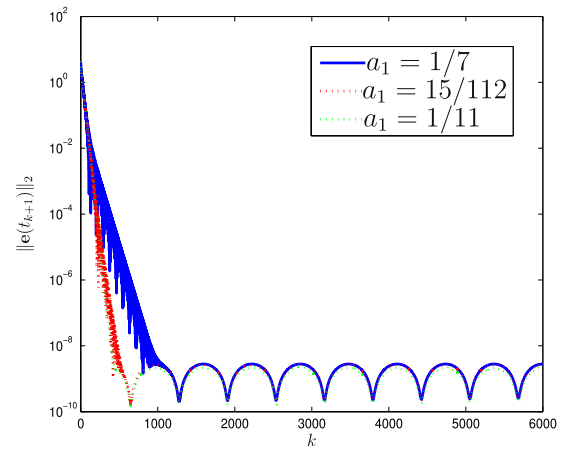
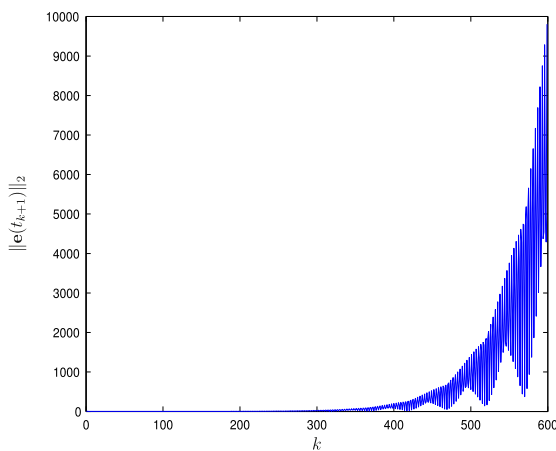
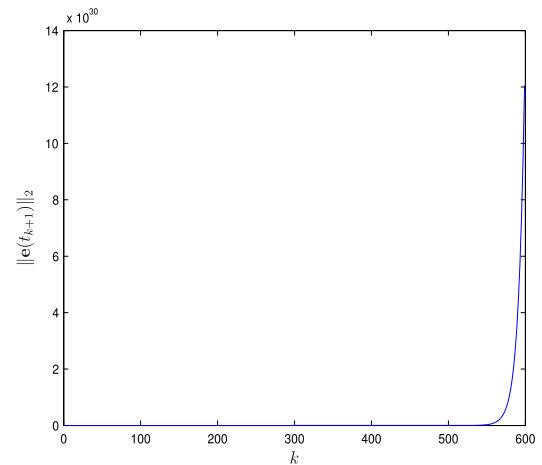
(a) With $g = 0.05$ s(b) With $g = 0.005$ s(c) With $g = 0.05$ s and $a_1 = 1/6$ (d) With $g = 0.05$ s and $a_1 = 1/12$

Fig. 4. Trajectories of residual error $\|e(t_{k+1})\|_2$ of TVNO problem (36) synthesized by general four-step DTZD model (35) with $h = 0.05$ and different values of a_1 .

main multiplications and additions come from the same matrix operation of the above three models for a high dimensional time-varying nonlinear optimization, which are n^2 and $n^2 - n$, respectively. It means that the main computational complexity of these models originates from the problem itself. In this sense, the slightly more multiplications and additions of proposed DTDD model (32) and proposed DTZD model (35) can be ignored.

To substantiate the above analyses, we apply Euler-type DTZD model (37), proposed DTDD model (32) and DTZD model (35) for solving TVNO problem (36). Average computing time per updating generated by the three models is considered in numerical experiments, which are presented as Table 4. In order to guarantee the accuracy of the results, we run the program in the same environments of hardware and software and take average testing values. The results are presented as Table 4. We can observe from Table 4 that the data of average computing time per updating generated by the three models are similar, coinciding with the above analyses.

According to the computational complexity analyses and numerical comparisons, we know that proposed DTDD model (32) and proposed DTZD model (35) are similar to Euler-type DTZD model (37) in terms of computational complexity for solving the time-varying nonlinear optimization, but they (i.e., the former) can achieve computing precision superior to Euler-type DTZD model (37).

Table 4

Data of average computing time per updating generated by Euler-type DTZD model (37), proposed DTDD model (32) and proposed DTZD model (35) with $a_0 = 1/9$ and different g .

Model	$g = 0.01$ s	$g = 0.02$ s	$g = 0.03$ s	$g = 0.04$ s	$g = 0.05$ s
(37)	9.55×10^{-5} s	9.75×10^{-5} s	9.49×10^{-5} s	9.84×10^{-5} s	9.80×10^{-5} s
(32)	9.88×10^{-5} s	1.02×10^{-4} s	1.02×10^{-4} s	1.02×10^{-4} s	1.03×10^{-4} s
(35)	1.00×10^{-4} s	1.03×10^{-4} s	1.05×10^{-4} s	1.06×10^{-4} s	1.05×10^{-4} s

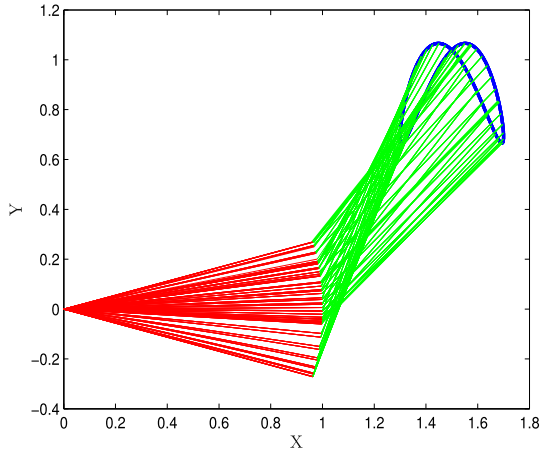
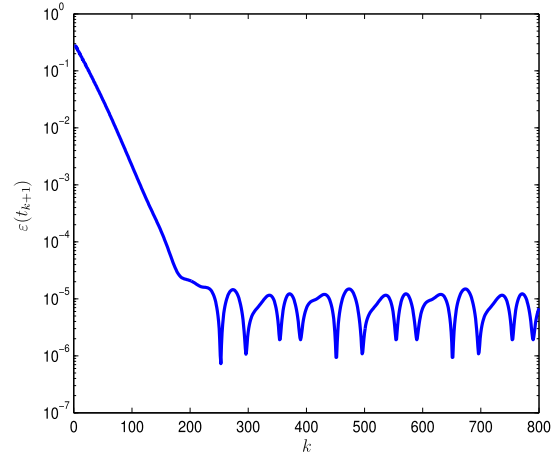
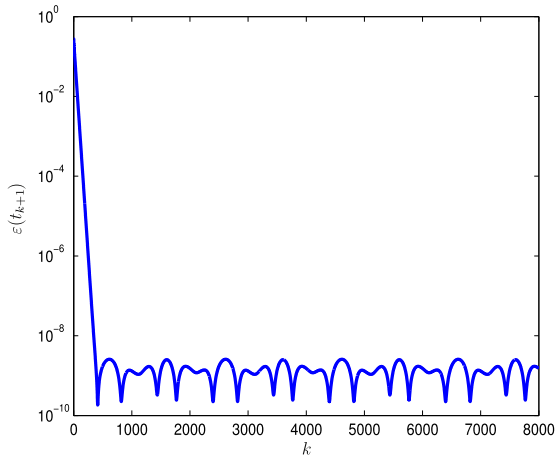
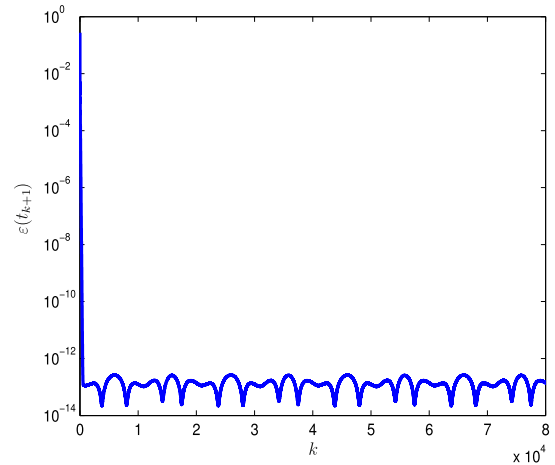
(a) With $a_1 = 1/9$ (b) With $g = 0.05$ s(c) With $g = 0.005$ s(d) With $g = 0.0005$ s

Fig. 5. Robot trajectories synthesized by model (39) and residual error $\varepsilon(t_{k+1})$ of model (39) with $h = 0.05$ and different values of g .

5.3. Example 2: Real-time motion control of robot manipulator

In this subsection, the real-time motion control of a two-joint planar robotic manipulator, is investigated and solved by the above-mentioned general four-step DTZD model (35). Firstly, the discrete-time kinematics equation of two-joint planar robot manipulator at the position level is given as $f(\boldsymbol{\theta}_k) = \mathbf{r}_k$, where $f(\cdot)$ is the kinematics mapping function with known structure and parameters with $\boldsymbol{\theta}_k = \boldsymbol{\theta}(t_k)$. Besides, $\boldsymbol{\theta}_k \in \mathbb{R}^2$ is the joint angle vector and $\mathbf{r}_k \in \mathbb{R}^2$ is the end-effector position vector. In view of robotic control, the following problem is to be solved at each computational time interval $[t_k, t_{k+1}) = [kg, (k+1)g) \subseteq [0, t_f]$:

$$\min \|\mathbf{r}_{dk} - \mathbf{r}_k\|_2^2,$$

where $\mathbf{r}_{dk}$ denotes the desired path vector at time instant t_k . Besides, the above TVNO problem can be rewritten as

$$\min \frac{\mathbf{r}_k^T \mathbf{r}_k}{2} - \mathbf{r}_k^T \mathbf{r}_{dk}. \quad (38)$$

Moreover, the real-time optimization solution of TVNO problem (38) at time instant t_k is denoted by $\boldsymbol{\theta}_k^* \in \mathbb{R}^2$. Such solution is ought to be found during each computational time interval and guarantees that TVNO problem (38) achieves its minimum value for each time instant t_k .

Furthermore, according to general four-step DTZD model (35) and in view of kinematics, we have the following model, which can solve TVNO problem (38) successfully.

$$\begin{aligned} \boldsymbol{\theta}_{k+1} = & - \left(\frac{-4a_1 + \frac{1}{2}}{a_1 + \frac{1}{3}} \right) \boldsymbol{\theta}_k - \left(\frac{6a_1 - 1}{a_1 + \frac{1}{3}} \right) \boldsymbol{\theta}_{k-1} \\ & - \left(\frac{-4a_1 + \frac{1}{6}}{a_1 + \frac{1}{3}} \right) \boldsymbol{\theta}_{k-2} - \left(\frac{a_1}{a_1 + \frac{1}{3}} \right) \boldsymbol{\theta}_{k-3} \\ & - \left(\frac{1}{a_1 + \frac{1}{3}} \right) (J_k^T J_k)^{-1} (h J_k^T (\mathbf{r}_k - \mathbf{r}_{dk}) + g J_k^T \dot{\mathbf{r}}_{dk}) + \mathbf{O}(g^4), \end{aligned} \quad (39)$$

where J_k stands for $J(\boldsymbol{\theta}_k)$ for convenience purposes and $J(\boldsymbol{\theta}_k) \in \mathbb{R}^{2 \times 2}$ denotes the Jacobian matrix of the two-joint planar manipulator at time instant t_k .

The end-effector is controlled to track a Lissajous curve, which is expressed as

$$\mathbf{r}_{dk} = \begin{bmatrix} 1.5 + 0.2 \sin\left(\frac{\pi t_k}{5}\right) \\ \frac{\sqrt{3}}{2} + 0.2 \sin\left(\frac{2\pi t_k}{5} + \frac{\pi}{3}\right) \end{bmatrix} \quad (40)$$

Moreover, since $\partial(\mathbf{r}_k^T \mathbf{r}_k / 2 - \mathbf{r}_k^T \mathbf{r}_{dk}) / \partial \boldsymbol{\theta} = J_k^T (\mathbf{r}_k - \mathbf{r}_{dk})$, we define a residual error $\varepsilon(t_{k+1})$ as $\varepsilon(t_{k+1}) = \|J_{k+1}^T (\mathbf{r}_{k+1} - \mathbf{r}_{dk+1})\|_2$. Apparently, the joint angle vector $\boldsymbol{\theta}_{k+1}$ converges to the real-time optimal solution $\boldsymbol{\theta}_{k+1}^*$ when residual error $\varepsilon(t_{k+1})$ converges to zero.

Discrete-time model (39) with $a_1 = 1/9$ is employed to solve this problem, and the corresponding results are presented in Fig. 5. In all three simulations, the initial joint states are set as $\boldsymbol{\theta}_0 = [0, \pi/3]^T$, the parameter $h = 0.05$ and the task duration $t_f = 40$ s. Numerical results are presented in Fig. 5. Specifically, Fig. 5(a) shows robot trajectories synthesized by model (39). Fig. 5(b), (c) and (d) show the residual error of model (39) with different g . From Fig. 5(a), we can see that it is evident that model (39) completes the task successfully. Furthermore, the residual error $\varepsilon(t_{k+1})$ is of 10^{-5} when $g = 0.05$ s, of 10^{-9} when $g = 0.005$ s and of 10^{-13} when $g = 0.0005$ s, respectively. Apparently, the residual error $\varepsilon(t_{k+1})$ has an $O(g^4)$ pattern as well.

6. Conclusion

In this paper, for solving TVNO problems, a CTDD model and a CTZD model have been developed and presented. Then, a general four-step ZeaD formula using bilinear transform has been proposed and investigated. This new general four-step ZeaD formula is consistent, zero-stable and convergent with the truncation error of $O(g^3)$. Based on this general four-step ZeaD formula, the discrete-time dynamics models have been developed and investigated for potential digital hardware realization. Through theoretical analyses, the fact that the MSSRE of general four-step DTZD model has a pattern of $O(g^4)$ has been theoretically proved. Moreover, the efficacy and accuracy of both general four-step DTDD and general four-step DTZD model when solving for TVNO problems have been illustrated through numerical experiments. Note that the DTDD model is sensitive to the initial states and its residual error shows a pattern of $O(g^2)$, and the general four-step DTZD model is robust with the residual error having a pattern of $O(g^4)$.

Acknowledgments

This work is supported by the National Natural Science Foundation of China [http://dx.doi.org/10.13039/501100001809] (with number 61473323), by the Foundation of Key Laboratory of Autonomous Systems and Networked Control, Ministry of Education, China (with number 2013A07), and also by the Laboratory Open Fund of Sun Yat-sen University, China (with number 20160209).

References

- [1] J. Lee, S. Leyffer, *Mixed Integer Nonlinear Programming*, Springer, New York, 2012.
- [2] D. Guo, Y. Zhang, Neural dynamics and Newton–Raphson iteration for nonlinear optimization, *J. Comput. Nonlinear Dyn.* 9 (2) (2014) 1016–1026.
- [3] L. Jin, Y. Zhang, Continuous and discrete Zhang dynamics for real-time varying nonlinear optimization, *Numer. Algorithms* 73 (1) (2016) 115–140.
- [4] J. Li, M. Mao, F. Uhlig, Y. Zhang, Z-type neural-dynamics for time-varying nonlinear optimization under a linear equality constraint with robot application, *J. Comput. Appl. Math.* 327 (2018) 155–166.
- [5] Y. Zhang, S.S. Ge, T.H. Lee, A unified quadratic programming based dynamical system approach to joint torque optimization of physically constrained redundant manipulators, *IEEE Trans. Syst. Man Cybern. B* 34 (5) (2004) 2126–2132.
- [6] Y. Zhao, S. Liu, Global optimization algorithm for mixed integer quadratically constrained quadratic program, *J. Comput. Appl. Math.* 319 (2017) 159–169.
- [7] Y. Zhang, W. Li, B. Qiu, Y. Ding, D. Zhang, Three-state space reformulation and control of MD-included one-link robot system using direct-derivative and Zhang-dynamics methods, in: *Proc. the 29th Chinese Control and Decision Conference*, 2017, pp. 3785–3790.
- [8] Y. Zhao, C. Feng, Time-dependent optimization for information processing and its applications, *J. Southeast Univ.* 29 (4) (1999) 1038–1041.
- [9] Y. Zhao, M.N.S. Swamy, A novel technique for tracking time-varying minimum and its applications, in: *Proc. IEEE Canadian Electrical and Computer Engineering Conference*, 1998, pp. 910–913.
- [10] M.B. Aroux, N.E. Echebest, E.A. Pilotta, Inexact restoration method for nonlinear optimization without derivatives, *J. Comput. Appl. Math.* 290 (2015) 26–43.
- [11] J.-H. Hours, C.N. Jones, Parametric non-convex decomposition algorithm for real-time and distributed NMPC, *IEEE Trans. Automat. Control* 61 (2) (2016) 287–302.
- [12] S. Rahili, W. Ren, Distributed convex optimization for continuous-time dynamics with time-varying cost functions, *IEEE Trans. Automat. Control* 62 (4) (2017) 1590–1605.
- [13] M. Fazlyab, C. Nowzari, G.J. Pappas, A. Ribeiro, V.M. Preciado, Self-triggered time-varying convex optimization, in: *Proc. the 55th IEEE Conference on Decision and Control*, 2016, pp. 3090–3097.
- [14] F.Y. Jakubiec, A. Ribeiro, D-MAP: Distributed maximum a posteriori probability estimation of dynamic systems, *IEEE Trans. Signal Process.* 61 (2) (2013) 450–466.
- [15] Q. Ling, A. Ribeiro, Decentralized dynamic optimization through the alternating direction method of multipliers, *IEEE Trans. Signal Process.* 62 (5) (2014) 1185–1197.
- [16] A. Simonetto, E. Dall’Anese, Prediction–correction algorithms for time-varying constrained optimization, *IEEE Trans. Signal Process.* 65 (20) (2017) 5481–5494.
- [17] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, A. Ribeiro, A class of prediction–correction methods for time-varying convex optimization, *IEEE Trans. Signal Process.* 64 (17) (2016) 4576–4591.
- [18] M. Fatemi, A new efficient conjugate gradient method for unconstrained optimization, *J. Comput. Appl. Math.* 300 (2016) 207–216.
- [19] J. Cu, L. Zhang, X. Xiao, Log-Sigmoid nonlinear Lagrange method for nonlinear optimization problems over second-order cones, *J. Comput. Appl. Math.* 229 (2009) 129–144.
- [20] E.G. Birgin, J.M. Martínez, A spectral conjugate gradient method for unconstrained optimization, *Appl. Math. Opt.* 43 (2) (2001) 117–128.
- [21] Y. Dai, L. Liao, New conjugacy conditions and related nonlinear conjugate gradient methods, *Appl. Math. Opt.* 43 (1) (2001) 87–101.
- [22] Y. Narushima, H. Yabe, Conjugate gradient methods based on secant conditions that generate descent search directions for unconstrained optimization, *J. Appl. Math. Comput.* 236 (17) (2012) 4303–4317.
- [23] A.L. Dontchev, M.I. Krastanov, R.T. Rockafellar, V.M. Veliov, An Euler–Newton continuation method for tracking solution trajectories of parametric variational inequalities, *SIAM J. Control Optim.* 51 (5) (2013) 1823–1840.
- [24] V.M. Zavala, M. Anitescu, Real-time nonlinear optimization as a generalized equation, *SIAM J. Control Optim.* 48 (8) (2010) 5444–5467.
- [25] A. Nagurny, J. Pan, Evolution variational inequalities and projected dynamical systems with application to human migration, *Math. Comput. Modelling* 43 (5–6) (2006) 646–657.
- [26] W. Su, *Traffic Engineering and Time-Varying Convex Optimization*, (Ph.D. thesis), Pennsylvania State University, 2009.
- [27] E. Dall’Anese, A. Simonetto, Optimal power flow pursuit, *IEEE Trans. Smart Grid* 9 (2) (2016) 942–952.
- [28] B. Liao, Y. Zhang, Different complex ZFs leading to different complex ZNN models for time-varying complex generalized inverse matrices, *IEEE Trans. Neural Netw. Learn. Syst.* 25 (9) (2014) 1621–1631.
- [29] M.P. Barbarosou, N.G. Maratos, A nonfeasible gradient projection recurrent neural network for equality-constrained optimization problems, *IEEE Trans. Neural Netw.* 19 (10) (2008) 1665–1677.
- [30] Y. Zhang, D.C. Jiang, J. Wang, A recurrent neural network for solving Sylvester equation with time-varying coefficients, *IEEE Trans. Neural Netw.* 13 (5) (2002) 1053–1063.
- [31] Y. Zhang, S.S. Ge, Design and analysis of a general recurrent neural network model for time-varying matrix inversion, *IEEE Trans. Neural Netw.* 16 (6) (2005) 1477–1490.
- [32] Y. Zhang, Y. Chou, J. Chen, Z. Zhang, L. Xiao, Presentation, error analysis and numerical experiments on a group of 1-step-ahead numerical differentiation formulas, *J. Comput. Appl. Math.* 239 (2013) 406–414.
- [33] Y. Zhang, L. Jin, D. Guo, Y. Yin, Y. Chou, Taylor-type 1-step-ahead numerical differentiation rule for first-order derivative approximation and ZNN discretization, *J. Comput. Appl. Math.* 273 (2015) 29–40.
- [34] J.H. Mathews, K.D. Fink, *Numerical Methods Using MATLAB*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 2005.
- [35] A.V. Oppenheim, *Discrete-Time Signal Processing*, Pearson Higher Education Inc., New Jersey, NJ, 2010.
- [36] K. Ogata, *Modern Control Engineering*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 2001.