TYPE Original Research
PUBLISHED 22 May 2024
DOI 10.3389/fnbot.2024.1406604

Check for updates

OPEN ACCESS

EDITED BY
Long Jin,
Lanzhou University, China

REVIEWED BY
Chenfu Yi,
Guangdong Polytechnic Normal University, China
Vasilios N. Katsikis,
National and Kapodistrian University of Athens, Greece

*CORRESPONDENCE
Binbin Qiu
✉ qiubb6@mail.sysu.edu.cn

RECEIVED 25 March 2024
ACCEPTED 03 May 2024
PUBLISHED 22 May 2024

CITATION
Zhang Y, Han Y and Qiu B (2024) An adaptive discretized RNN algorithm for posture collaboration motion control of constrained dual-arm robots.
Front. Neurorobot. 18:1406604.
doi: 10.3389/fnbot.2024.1406604

An adaptive discretized RNN algorithm for posture collaboration motion control of constrained dual-arm robots

Yichen Zhang, Yu Han and Binbin Qiu*

School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen, China

Although there are many studies on repetitive motion control of robots, few schemes and algorithms involve posture collaboration motion control of constrained dual-arm robots in three-dimensional scenes, which can meet more complex work requirements. Therefore, this study establishes the minimum displacement repetitive motion control scheme for the left and right robotic arms separately. On the basis of this, the design mentality of the proposed dual-arm posture collaboration motion control (DAPCMC) scheme, which is combined with a new joint-limit conversion strategy, is described, and the scheme is transformed into a time-variant equation system (TVES) problem form subsequently. To address the TVES problem, a novel adaptive Taylor-type discretized recurrent neural network (ATT-DRNN) algorithm is devised, which fundamentally solves the problem of calculation accuracy which cannot be balanced well with the fast convergence speed. Then, stringent theoretical analysis confirms the dependability of the ATT-DRNN algorithm in terms of calculation precision and convergence rate. Finally, the effectiveness of the DAPCMC scheme and the excellent convergence competence of the ATT-DRNN algorithm is verified by a numerical simulation analysis and two control cases of dual-arm robots.

KEYWORDS

dual-arm robot, dual-arm posture collaboration motion control (DAPCMC), time-variant equation system (TVES), adaptive Taylor-type discretized recurrent neural network (ATT-DRNN), joint-limit conversion strategy

1 Introduction

With the continuous development of electronic information technology, robots, as a key carrier in the realm of artificial intelligence, have been assuming a progressively substantial role in manufacturing (Arents and Greitans, 2022), healthcare (Khan et al., 2020), service industries (McCartney and McCartney, 2020), and beyond (Cheng et al., 2023; Tanyıldızı, 2023; Yang et al., 2023; Liufu et al., 2024), bringing numerous conveniences to human life and work. Many scholars are focusing their attention on robotics research field.

A robotic arm is a mechanical device composed of multiple linked joints, typically equipped with various end-effectors based on the requirements of the work environment. By calculating and adjusting the rotational changes of each joint, the end-effector can be controlled to perform various movements in a predetermined manner, such as position and orientation, thereby accomplishing tasks. For instance, the MATLAB program and particle swarm optimization were utilized for the trajectory planning of the robotic arm (Ekrem and Aksoy, 2023); Chico et al. (2021) employed a hand gesture recognition system and the inertial measurement unit to control the position and orientation of a virtual robotic arm.

A target admittance model was designed in the joint space for hands-on procedures that can be applied in all commercially available general-purpose robotic arms with six or more DOF (Kastritsi and Doulgeri, 2021).

Due to the escalating complexity of task environments, single robotic arms frequently encounter challenges in effectively completing tasks, which highlights the advantages of dual robotic arms in collaborative and efficient task execution. For example, Jiang et al. (2022) presented an adaptive control method for a dual-arm robot to perform bimanual tasks under modeling uncertainties. Bombile and Billard (2022) designed a unified motion generation algorithm that enables a dual-arm robot to grab and release objects quickly. Wang et al. (2023) proposed a sliding mode controller with good robustness against the model uncertainties to capture and stabilize a spinning target in 3D space by a dual-arm space robot.

However, some of the methods mentioned above do not take into account the actual physical constraints of the robotic arms during initial modeling (e.g., Bombile and Billard, 2022; Jiang et al., 2022). This greatly limits the application scenarios of these algorithms and is inconsistent with the real working conditions of the robotic arms. Furthermore, the physical limitations of robotic arms typically pertain to constraints on joint angle and velocity. These constraints do not reside at the same constraint level, thus there are substantial computational challenges when attempting to address them collectively. An optimal approach entails a series of conversion strategies to harmonize these distinct hierarchical constraints to a congruous level (Zhang and Zhang, 2013) (e.g., velocity level). By implementing this approach, the constraints can be effectively unified and dealt without compromising their intended meaning. Some scholars (e.g., Li, 2020) have crafted novel approaches to these conversion strategies stemming from this foundation. Nevertheless, in the process, they have introduced too many supplementary parameters, rendering the strategies less straightforward for apprehension. Additionally, certain studies focus on the control of dual robotic arms based on 2D space, considerably limiting the operating range of robotic arms (Stolfi et al., 2017; Yang S. et al., 2020; Yang et al., 2021).

In recent years, with the rapid advancement of neural network research, many scholars have been committed to applying its formidable nonlinear modeling capability and efficient parallel computing ability to the domain of robotic arm motion control (Wang et al., 2021; Jin et al., 2024). This endeavor has given rise to a special kind of neural network known as the RNN (Xiao et al., 2021; Yan et al., 2022; Fu et al., 2023). For example, Xiao et al. (2021) proposed a noise-enduring and finite-time convergent design formula is suggested to establish a novel RNN. Fu et al. (2023) presented a gradient-feedback RNN to solve the unconstrained time-variant convex optimization problem.

To facilitate the calculation on computers and other digital hardware devices, some scholars focus on discretizing conventional CRNN models through time discretization techniques, leading to the development of DRNN algorithms (Liao et al., 2016; Liu et al., 2023a,b; Shi et al., 2023). The technique of second-order Taylor expansion was used to deal with the discrete time-variant nonlinear system, and a DRNN algorithm was proposed subsequently (Shi et al., 2023). Liao et al. (2016) proposed two Taylor-type DRNN algorithms on account of the Taylor-type formula to perform online dynamic equality-constrained quadratic programming. Liu et al. (2023a) designed a Taylor-type DRNN algorithm based on Taylor-type discrete scheme with smaller TE. It is worth noting that higher accuracy requirements often make the discretization formulas more complicated, inevitably leading to a large amount of computation and increasing the cost of actual production applications. After overall consideration, this study proposes an adaptive DRNN algorithm based on a three-step general Taylor-type discretization formula with an adaptive sampling period introduced, which is of high enough precision for practical applications.

Typically, due to the use of fixed sampling periods and fixed convergence factors in the conventional DRNN algorithms mentioned above, it is difficult for them to achieve a balance in computational precision and convergence rate, resulting in limited algorithmic dynamic and convergence performance. Therefore, some researchers have tried to introduce various adaptive mechanisms into model/algorithm design (Song et al., 2008; Yang M. et al., 2020; Dai et al., 2022; Cai and Yi, 2023). For example, Yang M. et al. (2020) proposed two discretized RNN algorithms with an adaptive Jacobian matrix. Cai and Yi (2023) developed an adaptive gradient-descent-based RNN model to solve time-variant problems based on the Lyapunov theory. Dai et al. (2022) proposed a hybrid RNN model by introducing a fuzzy adaptive control strategy to generate a fuzzy adaptive factor that can change its size adaptively according to the RE. Song et al. (2008) proposed a robust adaptive gradient-descent training algorithm based on an RNN hybrid training concept in discrete-time domain.

In light of the aforementioned circumstances, this study formulates a DAPCMC scheme in 3D space based on the dual-arm robot system and the new JLCS. Subsequently, a novel ATT-DRNN algorithm with adaptive sampling period and adaptive convergence factor is devised to effectively face the challenge of achieving a dynamic balance between great computational precision and rapid convergence rate. When compared with the CTT-DRNN algorithm and the CET-DRNN algorithm, the proposed ATT-DRNN algorithm demonstrates outstanding computational precision and rapid convergence rate. To demonstrate the features and strengths of the proposed ATT-DRNN algorithm, Table 1 shows the comparisons among distinct methods for the motion control of robots.

The remainder of this study consists of four parts. Section 2 formulates the DAPCMC scheme and designs the ATT-DRNN algorithm. Section 3 presents the theoretical analyses of the

Abbreviations: 3D, Three dimensional; 2D, Two dimensional; RNN, Recurrent neural network; TE, Truncation error; CRNN, Continuous recurrent neural network; DRNN, Discretized recurrent neural network; DAPCMC, Dual-arm posture collaboration motion control; JLCS, Joint-limit conversion strategy; ATT-DRNN, Adaptive Taylor-type discretized recurrent neural network; CTT-DRNN, Conventional Taylor-type discretized recurrent neural network; CET-DRNN, Conventional Euler-type discretized recurrent neural network; TVES, Time-variant equation system; MDRMC, Minimum displacement repetitive motion control; DOF, Degrees of freedom; LA, Left arm; RA, Right arm; TVQP, Time-variant quadratic programming; EE, Error equation; ACRNN, Adaptive continuous recurrent neural network; RE, Residual error; D-H, Denavit-Hartenberg; UAV, Unmanned aerial vehicle.

TABLE 1 Comparisons among distinct methods for motion control of robots.

Method	Posture control	Inequality constraint	Discretized handling	Adaptive mechanism	Applicable scene	Robotic arm number
Jiang et al. (2022)	No	No	No	Yes	2D	Dual
Yang et al. (2021)	Yes	Yes	Yes	No	2D	Dual
Yang S. et al. (2020)	No	Yes	No	No	2D	Dual
Fu et al. (2023)	No	No	No	No	3D	Single
Shi et al. (2023)	No	No	Yes	No	2D	Single
Liao et al. (2016)	No	No	Yes	No	2D	Single
Wu and Zhang (2023)	Yes	No	Yes	Yes	3D	Single
Yang M. et al. (2020)	No	No	Yes	Yes	3D	Single
ATT-DRNN	Yes	Yes	Yes	Yes	3D	Dual

proposed ATT-DRNN algorithm. Section 4 provides illustrative examples, and Section 5 concludes this study. Finally, the primary contributions/novelties of this paper can be summarized as follows.

1) Distinguishing from common dual-arm robot motion control schemes in 2D space, a novel construction methodology of the DAPCMC scheme in 3D space is provided, which can make a spatial dual-arm robot collaboratively execute repetitive tracking of a desired trajectory while adhering to a predetermined posture.
2) Distinguishing from existing strategies, an innovative JLCS is proposed, which has a ubiquitously differentiable and more succinct expression.
3) Distinguishing from conventional discretization methods, an innovative ATT-DRNN algorithm is engineered to address the DAPCMC scheme, which introduces a new adaptive convergence factor and sampling period to guarantee a notable convergence rate and exceptional convergence precision.
4) Distinguishing from the simple path-tracking task of single-arm robots, the posture collaboration motion control experiments of a UR5 dual-arm robot with the joint-angle and joint-velocity bound constraints considered substantiate the effectiveness of the proposed DAPCMC scheme and the outstanding convergence capability of the proposed ATT-DRNN algorithm.

2 Scheme formulation and algorithm design

This section describes how to construct a DAPCMC scheme that can be converted into a TVES problem and processed by the proposed ATT-DRNN algorithm.

2.1 Rudimentary knowledge

For the convenience of comprehension, let us construct a single robot arm motion control scheme with n DOF, which takes into account joint physical limits and can simultaneously ensure position control and orientation control during the MDRMC. Specifically, such a scheme can be described as below:

$$\min_{\mathfrak{z}(t)} \quad \frac{1}{2}\dot{\mathfrak{z}}^{\mathrm{T}}(t)\boldsymbol{U}(t)\dot{\mathfrak{z}}(t) + \boldsymbol{\varphi}^{\mathrm{T}}(t)\dot{\mathfrak{z}}(t), \tag{1}$$

$$\text{s.t.} \quad \boldsymbol{J}_1(\mathfrak{z}(t))\dot{\mathfrak{z}}(t) = \dot{\boldsymbol{\Upsilon}}_{\mathrm{I}}(t) - \alpha\left[\boldsymbol{\Upsilon}_{\mathrm{R}}(t) - \boldsymbol{\Upsilon}_{\mathrm{I}}(t)\right], \tag{2}$$

$$\boldsymbol{J}_2(\mathfrak{z}(t))\dot{\mathfrak{z}}(t) = \dot{\boldsymbol{o}}_{\mathrm{I}}(t) - \beta\left[\boldsymbol{o}_{\mathrm{R}}(t) - \boldsymbol{o}_{\mathrm{I}}(t)\right], \tag{3}$$

$$\mathfrak{z}^{-} \leq \mathfrak{z}(t) \leq \mathfrak{z}^{+}, \tag{4}$$

$$\dot{\mathfrak{z}}^{-} \leq \dot{\mathfrak{z}}(t) \leq \dot{\mathfrak{z}}^{+}, \tag{5}$$

where superscript $^{\mathrm{T}}$ represents the transpose operator; $\mathfrak{z}(t) = \left[\dot{\mathfrak{z}}_1(t), \dot{\mathfrak{z}}_2(t), ..., \dot{\mathfrak{z}}_n(t)\right]^{\mathrm{T}} \in \mathbb{R}^n$ indicates the angle values of the robotic joints, and $\dot{\mathfrak{z}}(t) \in \mathbb{R}^n$ means the angular velocities of the robotic joints; matrix $\boldsymbol{U}(t) = \boldsymbol{I}_{n\times n} \in \mathbb{R}^{n\times n}$ is an identity matrix; vector $\boldsymbol{\varphi}(t) = \xi\left[\mathfrak{z}(t) - \mathfrak{z}(0)\right] \in \mathbb{R}^n$ with design parameter $\xi > 0$ and $\mathfrak{z}(0)$ means the initial joint-angle vector; $\boldsymbol{J}_1(\mathfrak{z}(t)) \in \mathbb{R}^{3\times n}$ and $\boldsymbol{J}_2(\mathfrak{z}(t)) \in \mathbb{R}^{3\times n}$ represent the position Jacobian matrix and the orientation Jacobian matrix, respectively; $\boldsymbol{\Upsilon}_{\mathrm{I}}(t) \in \mathbb{R}^3$ and $\boldsymbol{\Upsilon}_{\mathrm{R}}(t) \in \mathbb{R}^3$ represent the ideal path and the real position of the end-executor, separately; $\boldsymbol{o}_{\mathrm{I}}(t) \in \mathbb{R}^3$ and $\boldsymbol{o}_{\mathrm{R}}(t) \in \mathbb{R}^3$ represent the ideal orientation and the real orientation of the end-executor, respectively; $\alpha > 0$ and $\beta > 0$ are both the error-feedback gains; $\mathfrak{z}^{\pm}$ and $\dot{\mathfrak{z}}^{\pm}$ denote the upper and lower limits of $\mathfrak{z}(t)$ and $\dot{\mathfrak{z}}(t)$, separately.

Remark 2.1: In accordance with previous experience (Zhang and Zhang, 2013), when $t \to \infty$, the objective function (1) at the joint-velocity level is equivalent to $\|\mathfrak{z}(t) - \mathfrak{z}(0)\|_2^2/2$ at the joint-angle level, where the design parameter $\xi > 0$ ought to be adjusted as large as allowed by the manipulator conditions. Note that the robot arm's repetitive motion planning scheme under minimal displacement can be regarded as an optimization objective that can be resolved at the joint-velocity level.

Remark 2.2: Referring to the contributions of previous scholars (Yang et al., 2021), the equality constraint (2) at the joint-velocity level is equivalent to $\boldsymbol{f}(\mathfrak{z}(t)) = \boldsymbol{\Upsilon}_{\mathrm{I}}(t)$ at the joint-angle level, when $t \to \infty$ and the error-feedback gain $\alpha > 0$ is at an appropriate value, where $\boldsymbol{f}(\cdot): \mathbb{R}^n \to \mathbb{R}^3$ represents the forward kinematics mapping function of a robotic arm.

Remark 2.3: Similarly, the equality constraint (3) at the joint-velocity level is equivalent to $\boldsymbol{g}(\mathfrak{z}(t)) = \boldsymbol{o}_{\mathrm{I}}(t)$ at the joint-angle level, when $t \to \infty$ and the error-feedback gain $\beta > 0$ is at an appropriate value, where nonlinear function $\boldsymbol{g}(\mathfrak{z}(t)) = \boldsymbol{o}_{\mathrm{R}}(t) = \left[o_{\mathrm{Rx}}(t), o_{\mathrm{Ry}}(t), o_{\mathrm{Rz}}(t)\right]^{\mathrm{T}} \in \mathbb{R}^3$ and the 2-norm of the real orientation vector $\boldsymbol{o}_{\mathrm{R}}(t)$ satisfies $\left\|\boldsymbol{o}_{\mathrm{R}}(t)\right\|_2 = 1$.

Note that the inequality constraint (4) is at the joint-angle level of the system. In order to integrate inequality constraints (4) and (5) of distinct constraint levels into a unified formulation at the joint-velocity level as below:

$$\mho^-(t) \le \dot{\mathfrak{z}}(t) \le \mho^+(t), \tag{6}$$

previous studies (Zhang and Zhang, 2013; Zhang et al., 2018; Li, 2020; Li et al., 2023; Qiu et al., 2023) supply a large number of JLCSs.

Nevertheless, the JLCS in Zhang and Zhang (2013) is unable to guarantee $\mho^-(t)$ or $\mho^+(t)$ to be differentiable anywhere. Meanwhile, as regard to the JLCS in Li (2020), $\mho^-(t)$ and $\mho^+(t)$ are designed as piecewise functions, respectively, and complex compound functions are embedded in them. In addition, the JLCS in Li et al. (2023); Qiu et al. (2023) adopt numerous design parameters and construct pretty complex expressions.

Therefore, as one of the contributions of this study, we provide a new JLCS. The ith ($i = 1, 2, ..., n$) elements of $\mho^-(t)$ and $\mho^+(t)$ in (6) are designed as follows:

$$\begin{cases} \mho_i^-(t) = \dot{\mathfrak{z}}_i^- \exp\left[\dfrac{\gamma \mathfrak{z}_i(t)}{\mathfrak{z}_i(t) - \mathfrak{z}_i^- + \varepsilon_1}\right], \varepsilon_1 \to 0_+, & (7) \\ \mho_i^+(t) = \dot{\mathfrak{z}}_i^+ \exp\left[\dfrac{\gamma \mathfrak{z}_i(t)}{\mathfrak{z}_i(t) - \mathfrak{z}_i^+ + \varepsilon_2}\right], \varepsilon_2 \to 0_-, & (8) \end{cases}$$

where $\dot{\mathfrak{z}}_i(t), \mathfrak{z}_i^-, \mathfrak{z}_i^+, \dot{\mathfrak{z}}_i^-, \dot{\mathfrak{z}}_i^+$ denote the ith element of $\mathfrak{z}(t), \mathfrak{z}^-, \mathfrak{z}^+, \dot{\mathfrak{z}}^-, \dot{\mathfrak{z}}^+$ in (4) and (5), separately; ε_1 and ε_2 are both non-zero minimum terms to ensure that the above equations are able to differentiable everywhere; design parameter $\gamma \in (0, 1)$ should be as small as possible.

Remark 2.4: To present the proposed JLCS (7)-(8) more specifically, Figure 1 exhibits the relationship between the ith joint angle $\mathfrak{z}_i(t)$ and the ith joint velocity $\dot{\mathfrak{z}}_i(t)$. It is worth noting that, when the joint approaches its lower or upper limit, the value of γ has a crucial effect on the changing rate of the joint-velocity boundary.

2.2 DAPCMC scheme

Finally, upon the previous section, we construct a dual-arm collaborative control system consisting of the LA and RA.

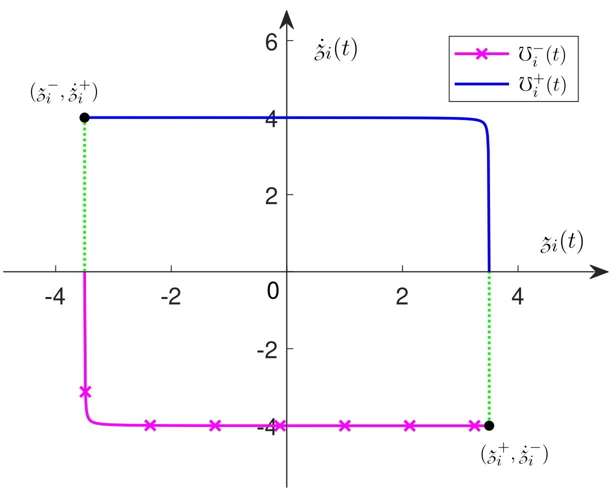

FIGURE 1
The relationship between the ith joint angle $\mathfrak{z}_i(t)$ and the ith joint velocity $\dot{\mathfrak{z}}_i(t)$ in the proposed JLCS (7)–(8), with $i = 1, 2, ..., n$.

2.2.1 LA collaborative control subsystem

According to (1)–(5), we construct the MDRMC scheme for the n-DOF LA as follows:

$$\min_{\dot{\mathfrak{z}}_{\mathcal{L}}(t)} \frac{1}{2}\dot{\mathfrak{z}}_{\mathcal{L}}^{\mathrm{T}}(t)\boldsymbol{U}_{\mathcal{L}}(t)\dot{\mathfrak{z}}_{\mathcal{L}}(t) + \boldsymbol{\varphi}_{\mathcal{L}}^{\mathrm{T}}(t)\dot{\mathfrak{z}}_{\mathcal{L}}(t), \tag{9}$$

$$\begin{aligned} \text{s.t.}\boldsymbol{J}_{1\mathcal{L}}(\mathfrak{z}_{\mathcal{L}}(t))\dot{\mathfrak{z}}_{\mathcal{L}}(t) = {} & \dot{\boldsymbol{\Upsilon}}_{\mathrm{I}\mathcal{L}}(t) \\ & - \alpha_{\mathcal{L}}\left[\boldsymbol{\Upsilon}_{\mathrm{R}\mathcal{L}}(t) - \boldsymbol{\Upsilon}_{\mathrm{I}\mathcal{L}}(t)\right], && (10) \\ \boldsymbol{J}_{2\mathcal{L}}(\mathfrak{z}_{\mathcal{L}}(t))\dot{\mathfrak{z}}_{\mathcal{L}}(t) = {} & \dot{\boldsymbol{o}}_{\mathrm{I}\mathcal{L}}(t) \\ & - \beta_{\mathcal{L}}\left[\boldsymbol{o}_{\mathrm{R}\mathcal{L}}(t) - \boldsymbol{o}_{\mathrm{I}\mathcal{L}}(t)\right], && (11) \\ \mathfrak{z}_{\mathcal{L}}^- \le \mathfrak{z}_{\mathcal{L}}(t) \le \mathfrak{z}_{\mathcal{L}}^+, & && (12) \\ \dot{\mathfrak{z}}_{\mathcal{L}}^- \le \dot{\mathfrak{z}}_{\mathcal{L}}(t) \le \dot{\mathfrak{z}}_{\mathcal{L}}^+, & && (13) \end{aligned}$$

where the subscript $\mathcal{L}$ denotes the LA; vector $\mathfrak{z}_{\mathcal{L}}(t) = \left[\mathfrak{z}_{\mathcal{L}1}(t), \mathfrak{z}_{\mathcal{L}2}(t), ..., \mathfrak{z}_{\mathcal{L}n}(t)\right]^{\mathrm{T}} \in \mathbb{R}^n$; vector $\boldsymbol{\varphi}_{\mathcal{L}}(t) = \xi_{\mathcal{L}}\left[\mathfrak{z}_{\mathcal{L}}(t) - \mathfrak{z}_{\mathcal{L}}(0)\right] \in \mathbb{R}^n$ with the design parameter $\xi_{\mathcal{L}} > 0$ and $\mathfrak{z}_{\mathcal{L}}(0)$ means the LA's initial joint-angle vector. Additionally, the meanings represented by the other symbols are similar to those in the MDRMC scheme (1)–(5).

Moreover, according to the JLCS (7)–(8), (12) and (13) in the LA's MDRMC scheme can be converted into the following form:

$$\mho_{\mathcal{L}}^-(t) \le \mho_{\mathcal{L}}(t) \le \mho_{\mathcal{L}}^+(t), \tag{14}$$

where $\mho_{\mathcal{L}}(t) = \dot{\mathfrak{z}}_{\mathcal{L}}(t) = \left[\dot{\mathfrak{z}}_{\mathcal{L}1}(t), \dot{\mathfrak{z}}_{\mathcal{L}2}(t), , ..., \dot{\mathfrak{z}}_{\mathcal{L}n}(t)\right]^{\mathrm{T}} \in \mathbb{R}^n$; the upper and lower limit values of $\mho_{\mathcal{L}}^-(t)$ and $\mho_{\mathcal{L}}^+(t)$ are correspondence to those of $\mho^-(t)$ and $\mho^+(t)$ in the JLCS (7)-(8).

Furthermore, by reorganizing the LA's MDRMC scheme (9)-(13), we obtain the LA collaborative control subsystem scheme, which has a briefer representation:

$$\min_{\mho_{\mathcal{L}}(t)} \quad \frac{1}{2}\mho_{\mathcal{L}}^{\mathrm{T}}(t)\boldsymbol{U}_{\mathcal{L}}(t)\mho_{\mathcal{L}}(t) + \boldsymbol{\varphi}_{\mathcal{L}}^{\mathrm{T}}(t)\mho_{\mathcal{L}}(t), \tag{15}$$

$$\text{s.t.} \quad \boldsymbol{A}_{\mathcal{L}}(t)\mho_{\mathcal{L}}(t) = \boldsymbol{c}_{\mathcal{L}}(t), \tag{16}$$

$$\boldsymbol{B}_{\mathcal{L}}(t)\mho_{\mathcal{L}}(t) \le \boldsymbol{d}_{\mathcal{L}}(t), \tag{17}$$

where matrices $\boldsymbol{A}_{\mathcal{L}}(t) \in \mathbb{R}^{6\times n}$ and $\boldsymbol{B}_{\mathcal{L}}(t) \in \mathbb{R}^{2n\times n}$ and vectors $\boldsymbol{c}_{\mathcal{L}}(t) \in \mathbb{R}^{6}$ and $\boldsymbol{d}_{\mathcal{L}}(t) \in \mathbb{R}^{2n}$ are expressed as below:

$$\boldsymbol{A}_{\mathcal{L}}(t) = \begin{bmatrix} \boldsymbol{J}_{1\mathcal{L}}(\mathfrak{z}_{\mathcal{L}}(t)) \\ \boldsymbol{J}_{2\mathcal{L}}(\mathfrak{z}_{\mathcal{L}}(t)) \end{bmatrix},$$

$$\boldsymbol{c}_{\mathcal{L}}(t) = \begin{bmatrix} \dot{\boldsymbol{\Upsilon}}_{\mathrm{I}\mathcal{L}}(t) - \alpha_{\mathcal{L}}\left[\boldsymbol{\Upsilon}_{\mathrm{R}\mathcal{L}}(t) - \boldsymbol{\Upsilon}_{\mathrm{I}\mathcal{L}}(t)\right] \\ \dot{\boldsymbol{o}}_{\mathrm{I}\mathcal{L}}(t) - \beta_{\mathcal{L}}\left[\boldsymbol{o}_{\mathrm{R}\mathcal{L}}(t) - \boldsymbol{o}_{\mathrm{I}\mathcal{L}}(t)\right] \end{bmatrix}$$

$$\boldsymbol{B}_{\mathcal{L}}(t) = \begin{bmatrix} \boldsymbol{I}_{n\times n} \\ -\boldsymbol{I}_{n\times n} \end{bmatrix}, \qquad \boldsymbol{d}_{\mathcal{L}}(t) = \begin{bmatrix} \mho^{+}_{\mathcal{L}}(t) \\ -\mho^{-}_{\mathcal{L}}(t) \end{bmatrix}.$$

2.2.2 RA collaborative control subsystem

Similar to the (2.2.1), the MDRMC scheme for the n-DOF RA is follows:

$$\min_{\dot{\mathfrak{z}}_{\mathcal{R}}(t)} \ \frac{1}{2}\dot{\mathfrak{z}}^{\mathrm{T}}_{\mathcal{R}}(t)\boldsymbol{U}_{\mathcal{R}}(t)\dot{\mathfrak{z}}_{\mathcal{R}}(t) + \boldsymbol{\varphi}^{\mathrm{T}}_{\mathcal{R}}(t)\dot{\mathfrak{z}}_{\mathcal{R}}(t), \tag{18}$$

$$\text{s.t.}\boldsymbol{J}_{1\mathcal{R}}(\mathfrak{z}_{\mathcal{R}}(t))\dot{\mathfrak{z}}_{\mathcal{R}}(t) = \dot{\boldsymbol{\Upsilon}}_{\mathrm{I}\mathcal{R}}(t) - \alpha_{\mathcal{R}}\left[\boldsymbol{\Upsilon}_{\mathrm{R}\mathcal{R}}(t) - \boldsymbol{\Upsilon}_{\mathrm{I}\mathcal{R}}(t)\right], \tag{19}$$

$$\boldsymbol{J}_{2\mathcal{R}}(\mathfrak{z}_{\mathcal{R}}(t))\dot{\mathfrak{z}}_{\mathcal{R}}(t) = \dot{\boldsymbol{o}}_{\mathrm{I}\mathcal{R}}(t) - \beta_{\mathcal{R}}\left[\boldsymbol{o}_{\mathrm{R}\mathcal{R}}(t) - \boldsymbol{o}_{\mathrm{I}\mathcal{R}}(t)\right], \tag{20}$$

$$\mathfrak{z}^{-}_{\mathcal{R}} \le \mathfrak{z}_{\mathcal{R}}(t) \le \mathfrak{z}^{+}_{\mathcal{R}}, \tag{21}$$

$$\dot{\mathfrak{z}}^{-}_{\mathcal{R}} \le \dot{\mathfrak{z}}_{\mathcal{R}}(t) \le \dot{\mathfrak{z}}^{+}_{\mathcal{R}}, \tag{22}$$

where the subscript $\mathcal{R}$ denotes the RA; vector $\mathfrak{z}_{\mathcal{R}}(t) = \left[\dot{\mathfrak{z}}_{\mathcal{R}1}(t), \dot{\mathfrak{z}}_{\mathcal{R}2}(t), ..., \dot{\mathfrak{z}}_{\mathcal{R}n}(t)\right]^{\mathrm{T}} \in \mathbb{R}^{n}$; vector $\boldsymbol{\varphi}_{\mathcal{R}}(t) = \xi_{\mathcal{R}}\left[\mathfrak{z}_{\mathcal{R}}(t) - \mathfrak{z}_{\mathcal{R}}(0)\right] \in \mathbb{R}^{n}$ with design parameter $\xi_{\mathcal{R}} > 0$ and $\mathfrak{z}_{\mathcal{R}}(0)$ means the RA's initial joint-angle vector. Additionally, the meanings represented by the other symbols are similar to those in the MDRMC scheme for the LA (9)–(13).

Similarly, (21) and (22) in the RA's MDRMC scheme can be transformed into the following form:

$$\mho^{-}_{\mathcal{R}}(t) \le \mho_{\mathcal{R}}(t) \le \mho^{+}_{\mathcal{R}}(t), \tag{23}$$

where $\mho_{\mathcal{R}}(t) = \dot{\mathfrak{z}}_{\mathcal{R}}(t) = \left[\dot{\mathfrak{z}}_{\mathcal{R}1}(t), \dot{\mathfrak{z}}_{\mathcal{R}2}(t), , ..., \dot{\mathfrak{z}}_{\mathcal{R}n}(t)\right]^{\mathrm{T}} \in \mathbb{R}^{n}$; the upper and lower limit values of $\mho^{-}_{\mathcal{R}}(t)$ and $\mho^{+}_{\mathcal{R}}(t)$ are parallelism to those of $\mho^{-}_{\mathcal{L}}(t)$ and $\mho^{+}_{\mathcal{L}}(t)$ in the LA's JLCS (14).

Then, by reorganizing the RA's MDRMC scheme (18)–(22), we obtain the RA collaborative control subsystem scheme:

$$\min_{\mho_{\mathcal{R}}(t)} \ \frac{1}{2}\mho^{\mathrm{T}}_{\mathcal{R}}(t)\boldsymbol{U}_{\mathcal{R}}(t)\mho_{\mathcal{R}}(t) + \boldsymbol{\varphi}^{\mathrm{T}}_{\mathcal{R}}(t)\mho_{\mathcal{R}}(t), \tag{24}$$

$$\text{s.t.} \ \ \boldsymbol{A}_{\mathcal{R}}(t)\mho_{\mathcal{R}}(t) = \boldsymbol{c}_{\mathcal{R}}(t), \tag{25}$$

$$\boldsymbol{B}_{\mathcal{R}}(t)\mho_{\mathcal{R}}(t) \le \boldsymbol{d}_{\mathcal{R}}(t), \tag{26}$$

where matrices $\boldsymbol{A}_{\mathcal{R}}(t) \in \mathbb{R}^{6\times n}$ and $\boldsymbol{B}_{\mathcal{R}}(t) \in \mathbb{R}^{2n\times n}$ and vectors $\boldsymbol{c}_{\mathcal{R}}(t) \in \mathbb{R}^{6}$ and $\boldsymbol{d}_{\mathcal{R}}(t) \in \mathbb{R}^{2n}$ are expressed as follows:

$$\boldsymbol{A}_{\mathcal{R}}(t) = \begin{bmatrix} \boldsymbol{J}_{1\mathcal{R}}(\mathfrak{z}_{\mathcal{R}}(t)) \\ \boldsymbol{J}_{2\mathcal{R}}(\mathfrak{z}_{\mathcal{R}}(t)) \end{bmatrix},$$

$$\boldsymbol{c}_{\mathcal{R}}(t) = \begin{bmatrix} \dot{\boldsymbol{\Upsilon}}_{\mathrm{I}\mathcal{R}}(t) - \alpha_{\mathcal{R}}\left[\boldsymbol{\Upsilon}_{\mathrm{R}\mathcal{R}}(t) - \boldsymbol{\Upsilon}_{\mathrm{I}\mathcal{R}}(t)\right] \\ \dot{\boldsymbol{o}}_{\mathrm{I}\mathcal{R}}(t) - \beta_{\mathcal{R}}\left[\boldsymbol{o}_{\mathrm{R}\mathcal{R}}(t) - \boldsymbol{o}_{\mathrm{I}\mathcal{R}}(t)\right] \end{bmatrix}$$

$$\boldsymbol{B}_{\mathcal{R}}(t) = \begin{bmatrix} \boldsymbol{I}_{n\times n} \\ -\boldsymbol{I}_{n\times n} \end{bmatrix}, \qquad \boldsymbol{d}_{\mathcal{R}}(t) = \begin{bmatrix} \mho^{+}_{\mathcal{R}}(t) \\ -\mho^{-}_{\mathcal{R}}(t) \end{bmatrix}.$$

Furthermore we combine the LA collaborative control subsystem scheme (15)–(17) with the RA collaborative control subsystem scheme (24)–(26) to obtain a complete DAPCMC scheme, which is also a TVQP problem:

$$\min_{\mho(t)} \ \frac{1}{2}\mho^{\mathrm{T}}(t)\boldsymbol{U}(t)\mho(t) + \boldsymbol{\varphi}^{\mathrm{T}}(t)\mho(t), \tag{27}$$

$$\text{s.t.} \ \ \boldsymbol{A}(t)\mho(t) = \boldsymbol{c}(t), \tag{28}$$

$$\boldsymbol{B}(t)\mho(t) \le \boldsymbol{d}(t), \tag{29}$$

where

$$\boldsymbol{A}(t) = \begin{bmatrix} \boldsymbol{A}_{\mathcal{L}}(t) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_{\mathcal{R}}(t) \end{bmatrix} \in \mathbb{R}^{12\times 2n}, \ \mho(t) = \begin{bmatrix} \mho_{\mathcal{L}}(t) \\ \mho_{\mathcal{R}}(t) \end{bmatrix} \in \mathbb{R}^{2n},$$

$$\boldsymbol{c}(t) = \begin{bmatrix} \boldsymbol{c}_{\mathcal{L}}(t) \\ \boldsymbol{c}_{\mathcal{R}}(t) \end{bmatrix} \in \mathbb{R}^{12}$$

$$\boldsymbol{B}(t) = \begin{bmatrix} \boldsymbol{B}_{\mathcal{L}}(t) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}_{\mathcal{R}}(t) \end{bmatrix} \in \mathbb{R}^{4n\times 2n}, \ \boldsymbol{d}(t) = \begin{bmatrix} \boldsymbol{d}_{\mathcal{L}}(t) \\ \boldsymbol{d}_{\mathcal{R}}(t) \end{bmatrix} \in \mathbb{R}^{4n},$$

$$\boldsymbol{\varphi}(t) = \begin{bmatrix} \boldsymbol{\varphi}_{\mathcal{L}}(t) \\ \boldsymbol{\varphi}_{\mathcal{R}}(t) \end{bmatrix} \in \mathbb{R}^{2n}$$

$$\boldsymbol{U}(t) = \begin{bmatrix} \boldsymbol{U}_{\mathcal{L}}(t) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{U}_{\mathcal{R}}(t) \end{bmatrix} \in \mathbb{R}^{2n\times 2n}.$$

In order to resolve the proposed DAPCMC scheme (27)–(29), that is, to seek the optimal solution to the TVQP problem (27)–(29), it is necessary for us to concentrate on how to translate such a TVQP problem (27)–(29) into a more computationally convenient TVES problem. After that, solving the TVES problem is tantamount to finding the optimal solution to the TVQP problem (27)–(29).

With reference to Wei et al. (2022), the optimal solution to the TVQP problem (27)–(29) can be obtained by dealing with the following TVES problem:

$$\mathcal{H}(t)\boldsymbol{\chi}(t) + \boldsymbol{g}(t) = \boldsymbol{0}, \tag{30}$$

where the coefficient matrix $\mathcal{H}(t) \in \mathbb{R}^{\varpi\times\varpi}$ and the vectors $\boldsymbol{\chi}(t) \in \mathbb{R}^{\varpi}$ and $\boldsymbol{g}(t) \in \mathbb{R}^{\varpi}$ can be described as follows:

$$\mathcal{H}(t) = \begin{bmatrix} \boldsymbol{U}(t) & \boldsymbol{A}^{\mathrm{T}}(t) & \boldsymbol{B}^{\mathrm{T}}(t) \\ \boldsymbol{A}(t) & \boldsymbol{0}_{12\times 12} & \boldsymbol{0}_{12\times 4n} \\ -\boldsymbol{B}(t) & \boldsymbol{0}_{4n\times 12} & \boldsymbol{I}_{4n\times 4n} \end{bmatrix}$$

$$\boldsymbol{\chi}(t) = \begin{bmatrix} \mho(t) \\ \boldsymbol{\lambda}(t) \\ \boldsymbol{\mu}(t) \end{bmatrix}, \qquad \boldsymbol{g}(t) = \begin{bmatrix} \boldsymbol{\varphi}(t) \\ -\boldsymbol{c}(t) \\ \boldsymbol{r}(t) \end{bmatrix}$$

where the Lagrange multiplier $\boldsymbol{\lambda}(t) \in \mathbb{R}^{12}$ is connected with the equality constraint (28) and the Lagrange multiplier $\boldsymbol{\mu}(t) \in \mathbb{R}^{4n}$ connected with the inequality constraint (29); $\boldsymbol{r}(t) = \boldsymbol{d}(t) - \sqrt{\boldsymbol{v}(t)\circ\boldsymbol{v}(t) + \boldsymbol{\mu}(t)\circ\boldsymbol{\mu}(t) + \boldsymbol{\varepsilon}_3}$ and $\boldsymbol{v}(t) = \boldsymbol{d}(t) - \boldsymbol{B}(t)\mho(t)$; $\circ$ bespeaks the Hadamard product operator; $\boldsymbol{\varepsilon}_3 \to 0_{+}$ and $\varpi = 6n + 12$.

In other words, as long as we can explore the solution $\boldsymbol{\chi}(t)$ suitable for the TVES problem (30), it means that we have found the optimal solution to the TVQP problem (27)–(29); Next, we will explain the derivation of the ATT-DRNN algorithm and employ it to work out the TVES problem (30), the TVQP problem (27)–(29), and the proposed DAPCMC scheme (27)–(29).

2.3 Algorithm design

First, we set up the following vector-valued EE in the light of the TVES problem (30):

$$\boldsymbol{e}(t) = \mathcal{H}(t)\boldsymbol{\chi}(t) + \boldsymbol{g}(t). \tag{31}$$

Finally, we utilize the RNN evolution rule (Shi and Zhang, 2018) as below:

$$\dot{\boldsymbol{e}}(t) = \frac{\mathrm{d}\boldsymbol{e}(t)}{\mathrm{d}t} = -\zeta\boldsymbol{e}(t), \tag{32}$$

where the fixed convergence factor $\zeta > 0$ has an important impact on the global exponential convergence rate. The larger the ζ chooses, the faster the convergence rate one acquires.

Then, the RNN evolution rule (32) can be further expanded as the following equation on account of the EE :

$$\dot{\mathcal{H}}(t)\boldsymbol{\chi}(t) + \mathcal{H}(t)\dot{\boldsymbol{\chi}}(t) + \dot{\boldsymbol{g}}(t) = -\zeta\left[\mathcal{H}(t)\boldsymbol{\chi}(t) + \boldsymbol{g}(t)\right]. \tag{33}$$

For the handiness of figuring out the optimal solution to the TVQP problem (27)–(28), we reformulate (33) as

$$\boldsymbol{D}(t)\dot{\boldsymbol{\chi}}(t) = -\boldsymbol{V}(t)\boldsymbol{\chi}(t) - \boldsymbol{\varrho}(t) - \zeta\left[\mathcal{H}(t)\boldsymbol{\chi}(t) + \boldsymbol{g}(t)\right], \tag{34}$$

where

$$\boldsymbol{D}(t) = \begin{bmatrix} \boldsymbol{U}(t) & \boldsymbol{A}^{\mathrm{T}}(t) & \boldsymbol{B}^{\mathrm{T}}(t) \\ \boldsymbol{A}(t) & \boldsymbol{0}_{12\times 12} & \boldsymbol{0}_{12\times 4n} \\ \left[\boldsymbol{\ell}_1(t) - \boldsymbol{I}_{4n\times 4n}\right]\boldsymbol{B}(t) & \boldsymbol{0}_{4n\times 12} & \boldsymbol{I}_{4n\times 4n} - \boldsymbol{\ell}_2(t) \end{bmatrix} \in \mathbb{R}^{\varpi\times\varpi}$$

$$\boldsymbol{V}(t) = \begin{bmatrix} \dot{\boldsymbol{U}}(t) & \dot{\boldsymbol{A}}^{\mathrm{T}}(t) & \dot{\boldsymbol{B}}^{\mathrm{T}}(t) \\ \dot{\boldsymbol{A}}(t) & \boldsymbol{0}_{12\times 12} & \boldsymbol{0}_{12\times 4n} \\ \left[\boldsymbol{\ell}_1(t) - \boldsymbol{I}_{4n\times 4n}\right]\dot{\boldsymbol{B}}(t) & \boldsymbol{0}_{4n\times 12} & \boldsymbol{0}_{4n\times 4n} \end{bmatrix} \in \mathbb{R}^{\varpi\times\varpi}$$

$$\boldsymbol{\varrho}(t) = \begin{bmatrix} \dot{\boldsymbol{\varphi}}(t) \\ -\dot{\boldsymbol{c}}(t) \\ \left[\boldsymbol{I}_{4n\times 4n} - \boldsymbol{\ell}_1(t)\right]\dot{\boldsymbol{d}}(t) \end{bmatrix} \in \mathbb{R}^{\varpi},$$

with

$$\begin{cases} \boldsymbol{\ell}_1(t) = \wedge\left[\boldsymbol{\Omega}(t)\circ\boldsymbol{\nu}(t)\right] \\ \boldsymbol{\ell}_2(t) = \wedge\left[\boldsymbol{\Omega}(t)\circ\boldsymbol{\mu}(t)\right] \\ \boldsymbol{\Omega}(t) = \left[\boldsymbol{\nu}(t)\circ\boldsymbol{\nu}(t) + \boldsymbol{\mu}(t)\circ\boldsymbol{\mu}(t) + \boldsymbol{\varepsilon}_3\right]^{-\frac{1}{2}}, \boldsymbol{\varepsilon}_3 \to \boldsymbol{0}_+. \end{cases}$$

We treat (34) as a CRNN model. In order to facilitate its realization in computer system and digital hardware, the CTT-DRNN algorithm and the ATT-DRNN algorithm are introduced in the following subsections.

2.3.1 CTT-DRNN algorithm

In this subsection, a conventional Taylor-type discretization formula is given, and the CTT-DRNN algorithm is obtained by combining it with the CRNN model (34).

Based on Hu et al. (2018), the three-step general Taylor-type discretization formula is formulated as follows:

$$\dot{x}_k = \frac{(-2a+1)x_{k+1} + 6ax_k - (6a+1)x_{k-1} + 2ax_{k-2}}{2\sigma} + O(\sigma^2), \quad k = 2, 3, 4, ..., \tag{35}$$

where the argument $a < 0$; k is the updating index; $\sigma > 0$ is the fixed sampling period; $x_k = x(t_k)$ denotes the samping value of function $x(t)$ at time instant $t_k = k\sigma$; $O(\sigma^2)$ is the TE.

By applying the three-step general Taylor-type discretization formula (35) to discretize the CRNN model (34), we can acquire CTT-DRNN algorithm as below:

$$\boldsymbol{\chi}_{k+1} \doteq \frac{6a\boldsymbol{\chi}_k - (6a+1)\boldsymbol{\chi}_{k-1} + 2a\boldsymbol{\chi}_{k-2} - 2\boldsymbol{M}_k\left[\sigma\left(-\boldsymbol{V}_k\boldsymbol{\chi}_k - \boldsymbol{\varrho}_k\right) - h\left(\mathcal{H}_k\boldsymbol{\chi}_k + \boldsymbol{g}_k\right)\right]}{2a-1} \tag{36}$$

where symbol $\doteq$ denotes the computational assignment operation; $\boldsymbol{M}_k, \boldsymbol{V}_k, \mathcal{H}_k, \boldsymbol{\varrho}_k, \boldsymbol{g}_k$, and $\boldsymbol{\chi}_k$ mean the instantaneous values of $\boldsymbol{M}(t), \boldsymbol{V}(t), \mathcal{H}(t), \boldsymbol{\varrho}(t), \boldsymbol{g}(t)$, and $\boldsymbol{\chi}(t)$ sampling at time instant t_k with $\boldsymbol{M}(t)$ denoting the pseudoinverse of $\boldsymbol{D}(t)$; parameter $h = \sigma\zeta$ represents the solution step size generally set at the range of $(0, 1)$.

2.3.2 ATT-DRNN algorithm

According to the analysis of Subsection (2.3.1), on the one hand, the larger the fixed convergence factor ζ, the faster the global convergence rate of the system, thus we should naturally set ζ as large as possible at the beginning to ensure a sublime exponential converging capability of the CRNN model (34). On the other hand, it is recognized that the fixed argument σ as the sampling period is a significant factor affecting the convergence precision of the CTT-DRNN algorithm (36). Generally, the more remarkable convergence precision is guaranteed by a smaller value of σ taken at the initial stage. However, blindly setting a small value of σ may directly lead to an exiguous solution step size h, making it knotty for the solution process to converge rapidly or even proceed normally. Similarly, an excessively huge ζ also makes it hard to ensure a brilliant exactness of the algorithm due to incurring a gigantic solution step size h. It can be seen that the above situations are contradictory to each other. Moreover, according to the changes in system conditions, fixed parameters cannot meet the needs of different states. In view of this, to autonomously adjust the convergence factor ζ and the sampling period σ according to the actual convergence situation, and assure that the global state both has a remarkable convergence rate and outstanding convergence precision, a novel ATT-DRNN algorithm is designed as described in the following text.

First, according to the actual solution status, the adaptive sampling period $\sigma_k = \sigma(t_k)$ is designed as follows:

$$\sigma_k = \frac{q}{\left(p + \|\boldsymbol{e}_k\|_2\right)^{\delta}}, \tag{37}$$

where fixed arguments $p, q > 0$ are applied to adjust the solution accuracy of the algorithm; variable argument δ is utilized to ensure the algorithm precision while adjusting the sampling period change rate; error $\boldsymbol{e}_k = \mathcal{H}_k\boldsymbol{\chi}_k + \boldsymbol{g}_k$ and symbol $\|\cdot\|_2$ represents the 2-norm of a vector.

Accordingly, the adaptive convergence factor $\zeta_k = \zeta(t_k)$ is designed as follows:

$$\zeta_k = \frac{h\left(p + \|\boldsymbol{e}_k\|_2\right)^{\delta}}{q}, \tag{38}$$

where variable argument δ is utilized to ensure the algorithm accuracy while adjusting the global convergence rate of the algorithm; $h = \sigma_k\zeta_k$ is the same as that in (36).

Meanwhile, the corresponding continuous adaptive convergence factor $\zeta(t)$ can be written as follows:

$$\zeta(t) = \frac{h\left(p + \left\|\boldsymbol{e}(t)\right\|_2\right)^{\delta}}{q}. \tag{39}$$

With the help of the continuous adaptive convergence factor $\zeta(t)$ (39), a novel RNN evolution rule can be written as follows:

$$\dot{\boldsymbol{e}}(t) = \frac{\mathrm{d}\boldsymbol{e}(t)}{\mathrm{d}t} = -\zeta(t)\boldsymbol{e}(t). \tag{40}$$

Thus, on the account of the EE (31), the novel RNN evolution rule (40) can be further expanded and reformulated as the ACRNN model:

$$\dot{\boldsymbol{\chi}}(t) = \boldsymbol{M}(t)\left\{-\boldsymbol{V}(t)\boldsymbol{\chi}(t) - \boldsymbol{\varrho}(t) - \zeta(t)\left[\mathcal{H}(t)\boldsymbol{\chi}(t) + \boldsymbol{g}(t)\right]\right\}, \tag{41}$$

where the corresponding parameters are all the same as in the previous section.

Besides, by taking into account the adaptive sampling period σ_k (37), the adaptive three-step general Taylor-type discretization formula can be expressed as follows:

$$\dot{x}_k = \frac{(-2a+1)x_{k+1} + 6ax_k - (6a+1)x_{k-1} + 2ax_{k-2}}{2\sigma_k} + O(\sigma_k^2), \quad k = 2, 3, 4, ..., \tag{42}$$

Then, we can acquire the ATT-DRNN algorithm by using the adaptive three-step general Taylor-type discretization formula (42) to discretize the ACRNN model (41), which can be written as follows:

$$\boldsymbol{\chi}_{k+1} \doteq \tfrac{6a}{2a-1}\boldsymbol{\chi}_k - \tfrac{6a+1}{2a-1}\boldsymbol{\chi}_{k-1} + \tfrac{2a}{2a-1}\boldsymbol{\chi}_{k-2} - \tfrac{2}{2a-1}\boldsymbol{M}_k\left[\sigma_k\left(-\boldsymbol{V}_k\boldsymbol{\chi}_k - \boldsymbol{\varrho}_k\right) - h\left(\mathcal{H}_k\boldsymbol{\chi}_k + \boldsymbol{g}_k\right)\right], \tag{43}$$

where the solution step size $h = \sigma_k\zeta_k$ is generally set at the range of $(0, 1)$. Moreover, three initial state vectors $\boldsymbol{\chi}_k$ with $k = 0, 1, 2$ are necessary to start up the proposed ATT-DRNN algorithm (43). The first one $\boldsymbol{\chi}_0$ consists of $\mho_0$, $\boldsymbol{\lambda}_0$, and $\boldsymbol{\mu}_0$, where $\mho_0$ is determined by the initial joint-velocity vectors of the LA and RA, while $\boldsymbol{\lambda}_0$ and $\boldsymbol{\mu}_0$ are relatively arbitrarily set. The remaining initial state vectors can be generated by utilizing an adaptive Euler-type DRNN algorithm, which can be obtained by applying adaptive Euler forward formula to discretize the ACRNN model (41), i.e., $\boldsymbol{\chi}_{k+1} \doteq \boldsymbol{\chi}_k + \sigma_k\dot{\boldsymbol{\chi}}_k$ with $\dot{\boldsymbol{\chi}}_k = \boldsymbol{M}_k\left[-\boldsymbol{V}_k\boldsymbol{\chi}_k - \boldsymbol{\varrho}_k - \zeta_k\left(\mathcal{H}_k\boldsymbol{\chi}_k + \boldsymbol{g}_k\right)\right]$.

Remark 2.5: By observing (37), it is evident that the adaptive sampling period σ_k continuously adjusts according to the changes in the RE $\|\boldsymbol{e}_k\|_2$, with an increase in the RE $\|\boldsymbol{e}_k\|_2$ and a decrease in the sampling period σ_k, and vice versa.

Remark 2.6: By observing (38), it is evident that the adaptive convergence factor ζ_k continuously adjusts according to the changes in the RE $\|\boldsymbol{e}_k\|_2$, when the RE $\|\boldsymbol{e}_k\|_2$ increases, the adaptive convergence factor ζ_k grows, leading to a higher convergence rate, and vice versa.

Remark 2.7: The solution step size h procured through multiplying σ_k and ζ_k is always a positive constant. By observing Equations (37) and (38) simultaneously, it can be easily found that σ_k and ζ_k exhibit the reciprocal states to each other. That is to say, when the RE $\|\boldsymbol{e}_k\|_2$ is large, the algorithm will adjust and yield a smaller sampling period σ_k and a larger convergence factor ζ_k to guarantee a rapid convergence of the algorithm in an extremely short sampling time; on the contrary, when the RE $\|\boldsymbol{e}_k\|_2$ reduces, the algorithm will adaptively increase the sampling period σ_k and simultaneously decrease the convergence factor ζ_k. By decreasing the sampling period and increasing the convergence rate, the algorithm can promptly complete the calculation and improve its computational efficiency. Therefore, the ATT-DRNN algorithm (43) can consider both computational accuracy and convergence efficiency during the calculation process.

3 Theoretical analyses and results

This section theoretically analyzes the convergence property of the ACRNN model (41) and the computational precision of the ATT-DRNN algorithm (43) for solving the TVQP problem (27)–(29).

Theorem 1: With the parameters $h, p, q, \delta > 0$ of the continuous adaptive convergence factor $\zeta(t)$, the RE $\left\|\boldsymbol{e}(t)\right\|_2$ generated by the ACRNN model (41) exponentially converges to zero in a large-scale manner with the exponential convergence rate at least being hp^{δ}/q.

Proof: To begin with, by exploiting the EE (31), a Lyapunov function can be chosen as follows:

$$\mathbb{L}(t) = \frac{\left\|\boldsymbol{e}(t)\right\|_2^2}{2}. \tag{44}$$

Then, the time derivative of the function $\mathbb{L}(t)$ is obtained by referring to (40):

$$\dot{\mathbb{L}}(t) = \boldsymbol{e}^{\mathrm{T}}(t)\dot{\boldsymbol{e}}(t) = -\zeta(t)\left\|\boldsymbol{e}(t)\right\|_2^2. \tag{45}$$

Observing (44) and (45), one can draw the following conclusions.

(1) If and only if $\boldsymbol{e}(t) = \boldsymbol{0}$, $\mathbb{L}(t) = 0$; otherwise, $\mathbb{L}(t) > 0$.

(2) If and only if $\boldsymbol{e}(t) = \boldsymbol{0}$, $\dot{\mathbb{L}}(t) = 0$; otherwise, $\dot{\mathbb{L}}(t) < 0$.

In other words, the function $\mathbb{L}(t)$ is positive definite and its derivative $\dot{\mathbb{L}}(t)$ is negative definite, which satisfies the Lyapunov stability theory conditions (Isidori, 1989). Thus, it can be concluded that the RE $\left\|\boldsymbol{e}(t)\right\|_2$ converges to zero in a large-scale manner.

Second, by reconstructing and expanding (45), we acquire

$$\dot{\mathbb{L}}(t) = -2\frac{h\left(p + \left\|\boldsymbol{e}(t)\right\|_2\right)^{\delta}}{q}\frac{\left\|\boldsymbol{e}(t)\right\|_2^2}{2} = -2\frac{h\left(p + \left\|\boldsymbol{e}(t)\right\|_2\right)^{\delta}}{q}\mathbb{L}(t). \tag{46}$$

Furthermore, based on (46), the following inequality can be further formulated as follows:

$$\dot{\mathbb{L}}(t) = -2\frac{h\left(p + \left\|\boldsymbol{e}(t)\right\|_2\right)^{\delta}}{q}\mathbb{L}(t) \le -2\frac{hp^{\delta}}{q}\mathbb{L}(t). \tag{47}$$

Attempting to figure out the inequality (47), we get

$$\frac{\left\|\boldsymbol{e}(t)\right\|_2^2}{2} \le \frac{\left\|\boldsymbol{e}(0)\right\|_2^2}{2}\exp\left(-2\frac{hp^{\delta}}{q}t\right), \tag{48}$$

and the inequality (48) can be further formulated as

$$\left\|\boldsymbol{e}(t)\right\|_2 \leq \left\|\boldsymbol{e}(0)\right\|_2 \exp\left(-\frac{hp^{\delta}}{q}t\right). \tag{49}$$

Until now, in accordance with the inequality (49), we can conclude that the RE $\left\|\boldsymbol{e}(t)\right\|_2$ of the ACRNN model (41) exponentially converges to zero in a large-scale manner with the exponential convergence rate at least being hp^{δ}/q, which completes the proof. ■

Theorem 2: With the parameters $h, p, q, \delta > 0$ of the adaptive sampling period σ_k, the ATT-DRNN algorithm (43) is zero-stable and convergent with the TE of order $\boldsymbol{O}((q/p^{\delta})^3)$. In addition, the theoretical solution of TVES problem (30) converged by the ATT-DRNN algorithm (43) with a maximal steady-status RE $\lim_{k\to\infty}\sup\left\|\boldsymbol{e}_{k+1}\right\|_2$ bing of order $O((q/p^{\delta})^3)$.

Proof: First, drawing on the experience of Hu et al. (2018), it testified that the adaptive three-step general Taylor-type discretization formula (42) meets the conditions of zero-stable and convergent, which has a TE term $\boldsymbol{O}(\sigma^2)$.

By using the adaptive three-step general Taylor-type discretization formula (42) to discretize the ACRNN model (41), the new ATT-DRNN algorithm (43) can be rewritten as follows:

$$\begin{aligned}\boldsymbol{\chi}_{k+1} &= \tfrac{6a}{2a-1}\boldsymbol{\chi}_k - \tfrac{6a+1}{2a-1}\boldsymbol{\chi}_{k-1} + \tfrac{2a}{2a-1}\boldsymbol{\chi}_{k-2}\\ &-\tfrac{2}{2a-1}\boldsymbol{M}_k\left[\sigma_k\left(-\boldsymbol{V}_k\boldsymbol{\chi}_k - \boldsymbol{\varrho}_k\right) - h\left(\mathcal{H}_k\boldsymbol{\chi}_k + \boldsymbol{g}_k\right)\right] + \boldsymbol{O}(\sigma_k^3),\end{aligned} \tag{50}$$

where $\boldsymbol{O}(\sigma_k^3)$ is the TE term.

Due to the development process recommended above, it is distinct that the ATT-DRNN algorithm (43) originating from (50) is also zero-stable and similarly convergent with the TE of order $\boldsymbol{O}(\sigma_k^3)$. Therefore, we get

$$\lim_{k\to\infty}\sigma_k = \lim_{k\to\infty}\frac{q}{\left(p+\|\boldsymbol{e}_k\|_2\right)^{\delta}} = \frac{q}{p^{\delta}}, \tag{51}$$

which means that the TE term for the ATT-DRNN algorithm (43) is $\boldsymbol{O}((q/p^{\delta})^3)$.

Then, based on the ATT-DRNN algorithm (43), the theoretical solution $\boldsymbol{\chi}_{k+1}^*$ of TVES problem (30) can be expressed as follows:

$$\boldsymbol{\chi}_{k+1}^* = \boldsymbol{\chi}_{k+1} + \boldsymbol{O}((q/p^{\delta})^3). \tag{52}$$

In addition, it is known that the theoretical solution $\boldsymbol{\chi}_{k+1}^*$ of the TVES problem (30) satisfies $\mathcal{H}_{k+1}\boldsymbol{\chi}_{k+1}^* + \boldsymbol{g}_{k+1} = \boldsymbol{0}$. Thus, by combining (51) with (52), we can draw the following conclusion:

$$\begin{aligned}&\lim_{k\to\infty}\sup\left\|\boldsymbol{e}_{k+1}\right\|_2\\ &= \lim_{k\to\infty}\sup\left\|\mathcal{H}_{k+1}\boldsymbol{\chi}_{k+1} + \boldsymbol{g}_{k+1}\right\|_2\\ &= \lim_{k\to\infty}\sup\left\|\mathcal{H}_{k+1}\boldsymbol{\chi}_{k+1}^* + \boldsymbol{g}_{k+1} - \mathcal{H}_{k+1}\boldsymbol{O}(\sigma_k^3)\right\|_2\\ &= \lim_{k\to\infty}\sup\left\|\mathcal{H}_{k+1}\boldsymbol{O}(\sigma_k^3)\right\|_2\\ &\leq \lim_{k\to\infty}\sup(\left\|\mathcal{H}_{k+1}\right\|_{\mathrm{F}}\left\|\boldsymbol{O}(\sigma_k^3)\right\|_2) = O((q/p^{\delta})^3).\end{aligned} \tag{53}$$

Based on (53), it can be concluded that the maximal steady-status RE $\lim_{k\to\infty}\sup\left\|\boldsymbol{e}_{k+1}\right\|_2$ generated by the ATT-DRNN algorithm (43) is $O((q/p^{\delta})^3)$. Thus, we complete the proof. ■

4 Illustrative examples

In this section, a numerical simulation example is provided first and explored to state explicitly the remarkable competence of the devised ATT-DRNN algorithm (43) when tackling the TVQP problem (27)–(29). Then, two examples of dual-arm robot control are provided to demonstrate the effectiveness of the devised ATT-DRNN algorithm (43) in addressing the proposed DAPCMC scheme (27)–(29). Meanwhile, we utilize the CTT-DRNN algorithm (36) and the CET-DRNN algorithm in Wu and Zhang (2023) for comparisons to show the superior performance of the devised ATT-DRNN algorithm (43). To help readers understand the algorithm implementation process, the pseudo-code of the proposed ATT-DRNN algorithm (43) for addressing the DAPCMC scheme (27)–(29) is presented in Algorithm 1.

```
Set: $\alpha$, $\beta$, $\xi$, $a$, $h$, $p$, $q$, $\delta$, $T_{\mathrm{e}}$, $\gamma$, $\mathfrak{z}_{\mathcal{L}}^{\pm}$, $\mathfrak{z}_{\mathcal{R}}^{\pm}$, $\dot{\mathfrak{z}}_{\mathcal{L}}^{\pm}$, $\dot{\mathfrak{z}}_{\mathcal{R}}^{\pm}$;
Initialize: $\mathfrak{z}_{\mathcal{L}}(0)$, $\mathfrak{z}_{\mathcal{R}}(0)$, $\mho_{\mathcal{L}}(0)$, $\mho_{\mathcal{R}}(0)$, $\boldsymbol{\lambda}(0)$, $\boldsymbol{\mu}(0)$, $t(0)$;
while $t_k \leq T_{\mathrm{e}}$ do
  Generate $\mho_{\mathcal{L}}^{\pm}(t_k)$, $\mho_{\mathcal{R}}^{\pm}(t_k)$ from the JCLS;
  Compute $\mathcal{H}(t_k)$, $\boldsymbol{\chi}(t_k)$, $\boldsymbol{g}(t_k)$, $\boldsymbol{M}(t_k)$, $\boldsymbol{V}(t_k)$, $\boldsymbol{\varrho}(t_k)$, $\boldsymbol{V}(t_k)$, $\sigma(t_k)$, $\zeta(t_k)$;
  if $k = 0, 1$ then
     Compute $\boldsymbol{\chi}(t_{k+1})$ via the adaptive Euler-type DRNN algorithm:
        $\boldsymbol{\chi}(t_{k+1}) \doteq \boldsymbol{\chi}(t_k) + \sigma(t_k)\dot{\boldsymbol{\chi}}(t_k)$;
else
  Compute $\boldsymbol{\chi}(t_{k+1})$ via the ATT-DRNN algorithm:
        $\boldsymbol{\chi}(t_{k+1}) \doteq \frac{6a}{2a-1}\boldsymbol{\chi}(t_k) - \frac{6a+1}{2a-1}\boldsymbol{\chi}(t_{k-1}) + \frac{2a}{2a-1}\boldsymbol{\chi}(t_{k-2}) - \frac{2}{2a-1}\sigma(t_k)\dot{\boldsymbol{\chi}}(t_k)$;
Obtain $\mho_{\mathcal{L}}(t_{k+1})$ from the first $n$ elements of $\boldsymbol{\chi}(t_{k+1})$ and $\mho_{\mathcal{R}}(t_{k+1})$ from the $n+1$ to $2n$ elements of $\boldsymbol{\chi}(t_{k+1})$;
if $k = 0$ then
  Compute $\mathfrak{z}_{\mathcal{L}}(t_{k+1}) = \mathfrak{z}_{\mathcal{L}}(t_k) + \sigma(t_k)\mho_{\mathcal{L}}(t_{k+1})$, $\mathfrak{z}_{\mathcal{R}}(t_{k+1}) = \mathfrak{z}_{\mathcal{R}}(t_k) + \sigma(t_k)\mho_{\mathcal{R}}(t_{k+1})$;
else
  Compute $\mathfrak{z}_{\mathcal{L}}(t_{k+1}) = \frac{4}{3}\mathfrak{z}_{\mathcal{L}}(t_k) - \frac{1}{3}\mathfrak{z}_{\mathcal{L}}(t_{k-1}) + \frac{2}{3}\sigma(t_k)\mho_{\mathcal{L}}(t_{k+1})$, $\mathfrak{z}_{\mathcal{R}}(t_{k+1}) = \frac{4}{3}\mathfrak{z}_{\mathcal{R}}(t_k) - \frac{1}{3}\mathfrak{z}_{\mathcal{R}}(t_{k-1}) + \frac{2}{3}\sigma(t_k)\mho_{\mathcal{R}}(t_{k+1})$;
   Update $t_{k+1} = t_k + \sigma(t_k)$;
end
```

Algorithm 1. Pseudo-code of the proposed ATT-DRNN algorithm (43) for addressing the DAPCMC scheme (27)–(29).

4.1 Numerical simulation verification

A specific TVQP problem with equality and inequality constraints is provided, the details of which are outlined below:

$$
\begin{aligned}
\min_{\mho(t)}.&[\cos(3t)/6+1]\mho_1^2(t)+\sin(t)\mho_1(t)\mho_2(t)\\
&+\cos(3t)\mho_1(t)\mho_3(t)+2\sin(t)\mho_1(t)\\
&+[\sin(3t)/6+1]\mho_2^2(t)+[\sin(2t)+1]\mho_2(t)\mho_4(t)\\
&+2\cos(t)\mho_2(t)-[\cos(t)/2+1/2]\mho_3^2(t)\\
&+[\cos(2t)\mho_3(t)\mho_4(t)]/2+\sin(2t)\mho_3(t)\\
&-[\sin(t)/2+1/2]\mho_4^2(t)+\cos(2t)\mho_4(t)
\end{aligned}
$$

$$
\begin{aligned}
\text{s.t.}\; &-t\sin(t/2)\mho_1(t)+\{[4\cos(t/2)]/5+1/5\}\mho_2(t)=t\sin(2t+1)\\
&-[3\cos(9t/10)]/2\mho_3(t)+\sin(9t/10)\mho_4(t)=-\cos(3t/2)/2-3/10\\
&-0.2\sin(3t)-1.2\le\mho_1(t),\mho_2(t),\mho_3(t),\mho_4(t)\le 0.2\sin(3t)+1.2
\end{aligned}
\tag{54}
$$

where $\mho(t)=[\mho_1(t),\mho_2(t),\mho_3(t),\mho_4(t)]^{\mathrm{T}}$. By referring to the standard form of the TVQP problem (27)–(29), the corresponding coefficients are as follows:

$$
\boldsymbol{U}(t)=\begin{bmatrix}\cos(3t)/3+2 & \sin(t) & \cos(3t) & 0\\ \sin(t) & \sin(3t)/3+2 & 0 & \sin(2t)+1\\ \cos(3t) & 0 & -\cos(t)-1 & \cos(2t)/2\\ 0 & \sin(2t)+1 & \cos(2t)/2 & -\sin(t)-1\end{bmatrix}\in\mathbb{R}^{4\times4}
$$

$$
\boldsymbol{\varphi}(t)=\begin{bmatrix}\sin(t) & \cos(t) & \sin(2t) & \cos(2t)\end{bmatrix}\in\mathbb{R}^4
$$

$$
\boldsymbol{A}(t)=\begin{bmatrix}-t\sin(t/2) & [4\cos(t/2)]/5+1/5 & 0 & 0\\ 0 & 0 & -[3\cos(9t/10)]/2 & \sin(9t/10)\end{bmatrix}\in\mathbb{R}^{2\times4}
$$

$$
\begin{aligned}
\boldsymbol{B}(t)&=\begin{bmatrix}\boldsymbol{I}_{4\times4}\\ -\boldsymbol{I}_{4\times4}\end{bmatrix}\in\mathbb{R}^{8\times4}\\
\boldsymbol{c}(t)&=\begin{bmatrix}t\sin(2t+1)\\ -\cos(3t/2)/2-3/10\end{bmatrix}\in\mathbb{R}^2\\
\boldsymbol{d}(t)&=\begin{bmatrix}0.2\sin(3t)+1.2\end{bmatrix}_{8\times1}\in\mathbb{R}^8.
\end{aligned}
$$

To successfully address the above TVQP problem (54) using the ATT-DRNN algorithm (42), we set parameters $h=0.1$, $p=5$, $q=0.05$, and $\gamma=0.001$; the initial values of $\mho(0)$, $\boldsymbol{\lambda}(0)$, and $\boldsymbol{\mu}(0)$ are set to random values at the range of $(0, 0.001)$. Then, the entire simulation calculation time in the program is uniformly set to $T_{\mathrm{e}}=4$ s. Besides, we set the fixed sampling period $\sigma=0.01$ s for the CTT-DRNN algorithm (36) and the CET-DRNN algorithm in Wu and Zhang (2023).

Figure 2A shows the element trajectories of the status vector $\mho(t)$ generated by the ATT-DRNN algorithm (43) with $a=-0.3$ and $\delta=2$, which are strictly confined to the ranges of inequality constraints. Meanwhile, it can be seen from Figure 2B that the equality constraint $\boldsymbol{A}(t)\mho(t)$ of the TVQP problem (54) can be promptly satisfied and can consistently maintain this state. To save space, the solving states of the proposed algorithm (43) with different a and δ, as well as similar figures of other algorithms, are omitted.

In order to research the impact of δ on the solving results of the ATT-DRNN algorithm (43), the variation trajectory of the RE $\left\|\boldsymbol{e}(t)\right\|_2$ when taking different δ with $a=-0.3$ is exhibited in Figure 2C. As we can see that, when entering the steady state, the RE $\left\|\boldsymbol{e}(t)\right\|_2$ maintains at around 10^{-4} with $\delta=1$, 10^{-6} with $\delta=2$, and 10^{-8} with $\delta=3$. In other words, as the setting value of δ increases, the convergence speed of the ATT-DRNN algorithm (43) is accelerated, and the solution precision is higher.

To demonstrate the excellent performance of the proposed ATT-DRNN algorithm (43) compared with other conventional algorithms, we further investigate the REs $\left\|\boldsymbol{e}(t)\right\|_2$ generated by the algorithms of ATT-DRNN (43) with $\delta=2$, CTT-DRNN (36) and CET-DRNN in Wu and Zhang (2023) by figuring out the TVQP problem (54), respectively. The REs $\left\|\boldsymbol{e}(t)\right\|_2$ synthesized by these three algorithms with different a are displayed in Figure 2D. It can be seen that the RE $\left\|\boldsymbol{e}(t)\right\|_2$ generated by the ATT-DRNN algorithm (43) with $\delta=2$ and different a values can converge as small as 10^{-6} in approximately 0.3 s. The REs $\left\|\boldsymbol{e}(t)\right\|_2$ generated by the CTT-DRNN algorithm (36) with different a values can converge to roughly 10^{-4} in 1 s. The RE $\left\|\boldsymbol{e}(t)\right\|_2$ generated by the CET-DRNN algorithm in Wu and Zhang (2023) merely converges to around 10^{-2} in 0.5 s. Overall, the solution accuracy and convergence rate of the ATT-DRNN algorithm (43) are superior to the other two conventional algorithms. Besides, it can be concluded that the computing precision of the ATT-DRNN algorithm (43) is higher with the smaller absolute value of a chosen. In addition, the variation curves of the adaptive sampling period $\sigma(t)$ and the adaptive convergence factor $\zeta(t)$ with different δ values are portrayed in Figures 2E, F, separately, which indicate that $\sigma(t)$ and $\zeta(t)$ can converge and stabilize to their corresponding values in an extremely short time. The greater the δ chosen, the smaller the final stable value of $\sigma(t)$ and the larger the final stable value of $\zeta(t)$. Furthermore, during the solving process, as the RE $\left\|\boldsymbol{e}(t)\right\|_2$ rapidly converges and decreases at the beginning stage, the change of $\sigma(t)$ is inversely proportional to it, and the change of $\zeta(t)$ is directly proportional to it, which is consistent with our previous analysis conclusions from *Remark 2.5* to *Remark 2.7*.

In summary, the several situations above confirm that the ATT-DRNN algorithm (43) has an excellent ability to solve the TVQP problem. Compared with other conventional algorithms, the proposed algorithm has faster convergence speed and higher precision.

4.2 Control case I of dual-arm robot

For this fraction, we establish a DAPCMC scheme consisting of two UR5 robotic arms placed on the contralateral side, controlled by the ATT-DRNN algorithm (43) for dual heart-shaped trajectory tracking. In addition, the UR5 robotic arm is a sensitive lightweight 6-DOF robot, which has a small footprint and can be directly installed in a narrow workspace to complete tasks with high sensitivity requirements (Vivas and Sabater, 2021).

According to the design form of the DAPCMC scheme (27), (28), we establish a particular TVQP problem for a UR5 dual-arm robot consisting of two 6-DOF (with $n=6$) UR5 robotic arms. In this scheme, the LA and RA's initial joint-angle vectors are set

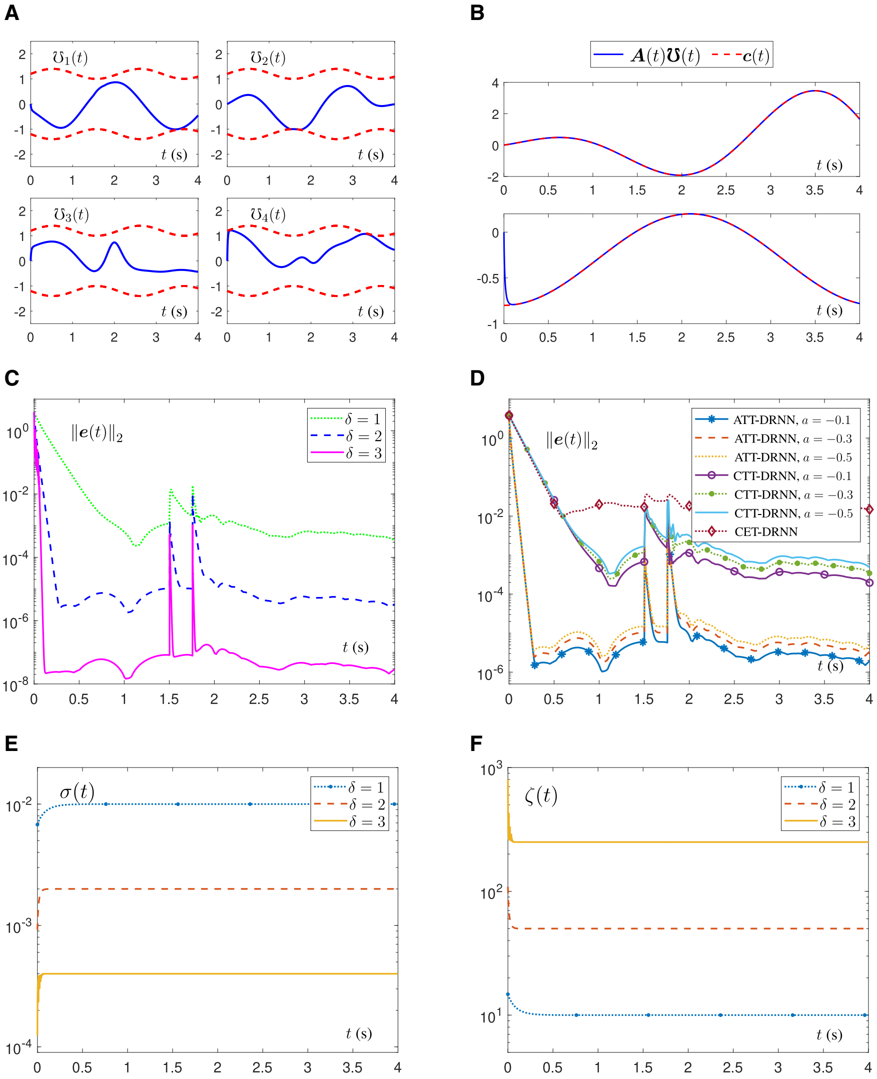

FIGURE 2
Numerical simulation results of the ATT-DRNN (43), CTT-DRNN (36), and CET-DRNN in Wu and Zhang (2023) algorithms for addressing the TVQP problem (53), separately. **(A)** Status vector $\mho(t)$ under inequality constraint generated by the ATT-DRNN algorithm (43) with $a = -0.3$ and $\delta = 2$. **(B)** $A(t)\mho(t)$ under equality constraint $c(t)$ obtained by the ATT-DRNN algorithm (43) with $a = -0.3$ and $\delta = 2$. **(C)** RE $\|e(t)\|_2$ generated by the ATT-DRNN algorithm (43) with $a = -0.3$ and different δ. **(D)** RE $\|e(t)\|_2$ by three algorithms with different a and $\delta = 2$. **(E)** Adaptive sampling period $\sigma(t)$ with different δ. **(F)** Adaptive convergence factor $\zeta(t)$ with different δ.

TABLE 2 The D-H parameters of UR5 robotic arm and its joint-angle and joint-velocity physical limits in the DAPCMC scheme (27)–(29).

Joint	$\widetilde{\alpha}$ (rad)	$\widetilde{a}$ (m)	$\widetilde{d}$ (m)	$\mathfrak{z}$ (rad)	$\mathfrak{z}^+$ (rad)	$\mathfrak{z}^-$ (rad)	$\dot{\mathfrak{z}}^+$ (rad/s)	$\dot{\mathfrak{z}}^-$ (rad/s)
1	1.5708	0	0.0892	$\mathfrak{z}_1$	1.5708	−1.5708	0.285	−0.285
2	0	−0.4250	0	$\mathfrak{z}_2$	0	−3.1416	0.285	−0.285
3	0	−0.3923	0	$\mathfrak{z}_3$	0	−3.1416	0.285	−0.285
4	1.5708	0	0.1092	$\mathfrak{z}_4$	1.5708	−1.5708	0.285	−0.285
5	−1.5708	0	0.0948	$\mathfrak{z}_5$	3.1416	0	0.285	−0.285
6	0	0	0.0825	$\mathfrak{z}_6$	1.5708	−1.5708	0.285	−0.285

as $\mathfrak{z}_{\mathcal{L}}(0) = [0, \pi/12, -4\pi/9, 13\pi/36, \pi/2, 0]^{\mathrm{T}}$ rad and $\mathfrak{z}_{\mathcal{R}}(0) = [0, -\pi/12, 4\pi/9, -13\pi/36, -\pi/2, 0]^{\mathrm{T}}$ rad, respectively; the LA and RA's initial joint-velocity vectors are set as $\dot{\mathfrak{z}}_{\mathcal{L}}(0) = \dot{\mathfrak{z}}_{\mathcal{R}}(0) = [\mathbf{0}]_{6\times 1}$ rad/s. We set the design parameters as $h = 0.2$, $p = 10$, $q = 0.2$, and $\delta = 2$. Then, the other correlative parameters are taken as $a = -0.1$, $\xi = 5$, and $\gamma = 0.001$; λ and μ are set to random values at the range of $(0, 0.001)$; α and β for two robotic arms are uniformly set as 0.8 and 0.1. Besides, we set the fixed

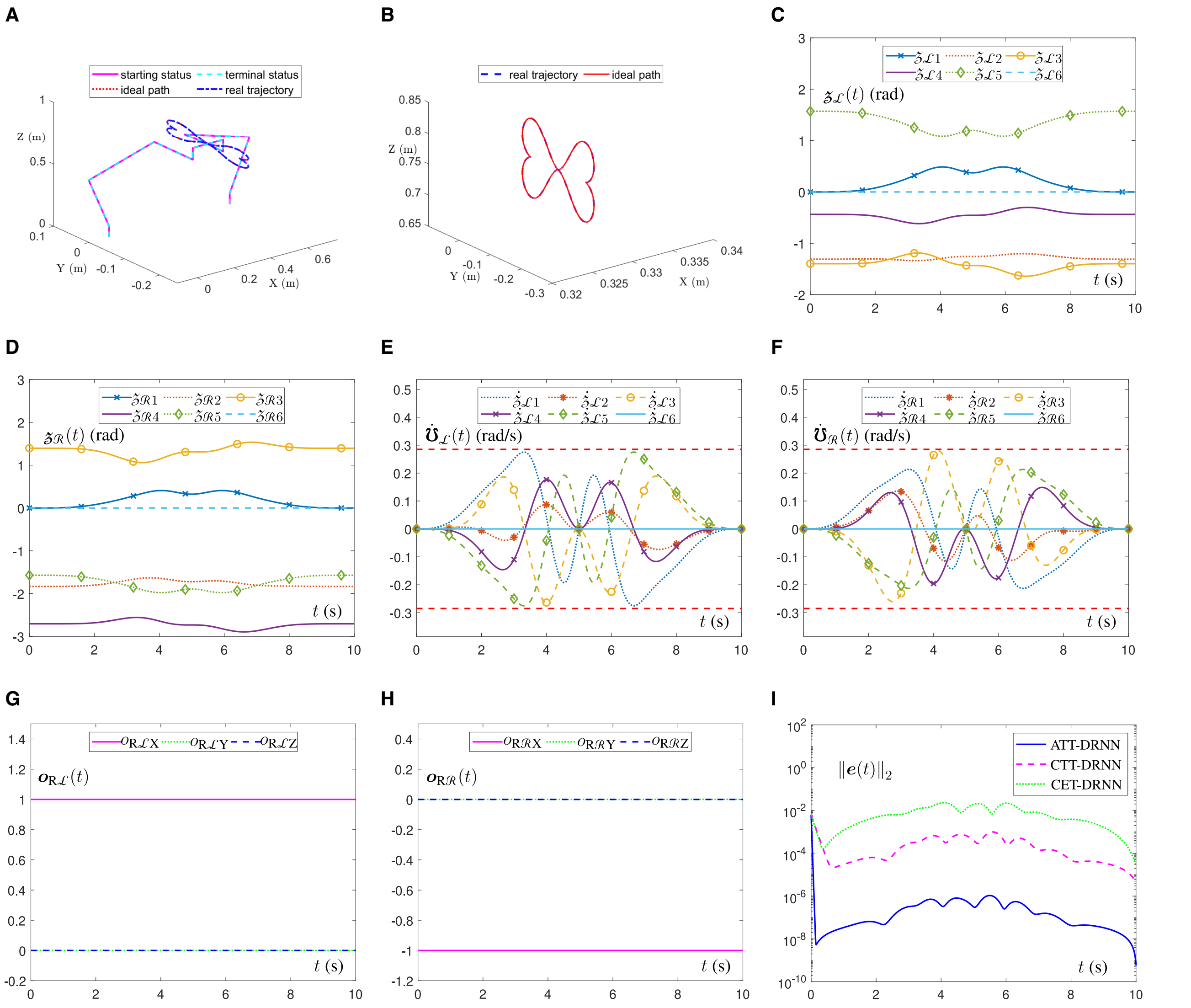

FIGURE 3
The ATT-DRNN algorithm (43) controls the posture collaboration motion of the UR5 dual-arm robot with two arms placed on the contralateral side for the dual heart-shaped trajectory tracking. **(A)** Starting and terminal statuses of the dual-arm robot and the end-executor real and ideal trajectories in 3D space. **(B)** Outlines of real trajectory and ideal path in a 3D space. **(C)** Variations of the LA joint angles $\mathfrak{z}_{\mathcal{L}}(t)$. **(D)** Variations of the RA joint angles $\mathfrak{z}_{\mathcal{R}}(t)$. **(E)** Variations of the LA joint velocities $\dot{\mho}_{\mathcal{L}}(t)$. **(F)** Variations of the RA joint velocities $\dot{\mho}_{\mathcal{R}}(t)$. **(G)** Variations of the LA end-executor orientation $\boldsymbol{o}_{\mathrm{R}\mathcal{L}}(t)$. **(H)** Variations of the RA end-executor orientation $\boldsymbol{o}_{\mathrm{R}\mathcal{R}}(t)$. **(I)** The RE $\|\boldsymbol{e}(t)\|_2$ generated by three different algorithms.

sampling period $\sigma = 0.01$ s for the CTT-DRNN algorithm (36) and the CET-DRNN algorithm in Wu and Zhang (2023). Moreover, the D-H parameters of the UR5 robotic arm and its joint-angle and joint-velocity physical limits in the DAPCMC scheme (27)–(29) are exhibited in Table 2.

Figure 3A illustrates the movement trajectory outlines of the dual-arm robot in a 3D space. It can be observed that the end-executor real trajectory is unanimous with the ideal path and that the joint-angle terminal status also perfectly overlaps with the starting status for each side of the dual robotic arms, which can be further confirmed by Figures 3C, D. Similarly, in Figure 3B, the ATT-DRNN algorithm (43) controls the dual-arm robot to achieve the posture collaboration motion and accomplish the dual heart-shaped path tracking. Finally, Figures 3E, F outline the joint-velocity variation curves for the joints of the left and right robotic arms. Clearly, all the joint-velocity values are not beyond the joint-velocity physical limits set at the beginning. Besides, the end-executor orientation variation curves are shown in Figures 3G, H, which remain constant during the task execution. Furthermore, in Figure 3I, the RE $\left\|\boldsymbol{e}(t)\right\|_2$ generated by the ATT-DRNN algorithm (43) for addressing the DAPCMC scheme (27)-(29) maintains at around 10^{-7}. By contrast, the RE $\left\|\boldsymbol{e}(t)\right\|_2$ generated by the CTT-DRNN algorithm (36) keeps at roughly 10^{-4}, and that generated by the CET-DRNN algorithm in Wu and Zhang (2023) can merely remain at about 10^{-2}.

In addition, Figure 4 shows the position error variation curves of the end-executor when the UR5 dual-arm robot tracks the dual heart-shaped trajectory under the control of different algorithms. In Figure 4A, the dual-arm robot controlled by the ATT-DRNN algorithm (43) can accomplish the trajectory following task

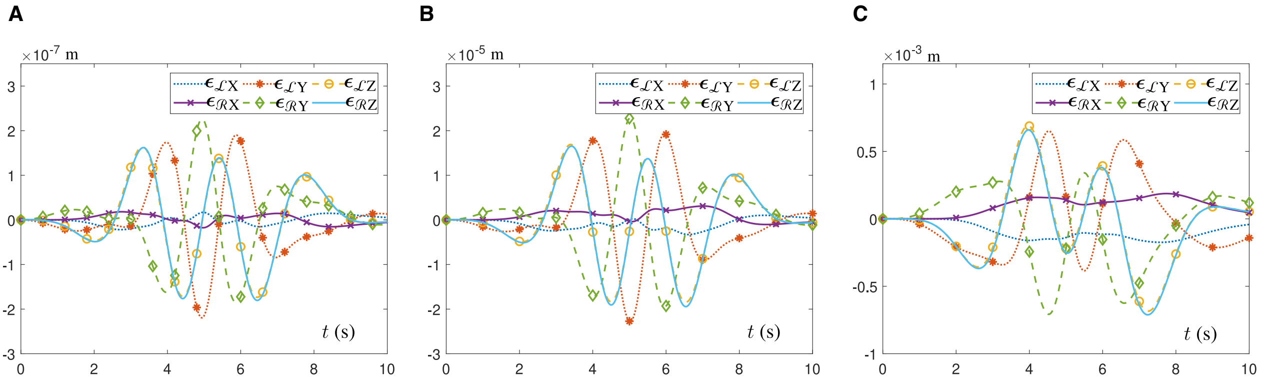

FIGURE 4
Variations of the end-executor position error when the UR5 dual-arm robot with two arms placed on the contralateral side to achieve the posture collaboration motion for the dual heart-shaped trajectory tracking. **(A)** Generated by the ATT-DRNN algorithm (43). **(B)** Generated by the CTT-DRNN algorithm (36). **(C)** Generated by the CET-DRNN algorithm in Wu and Zhang (2023).

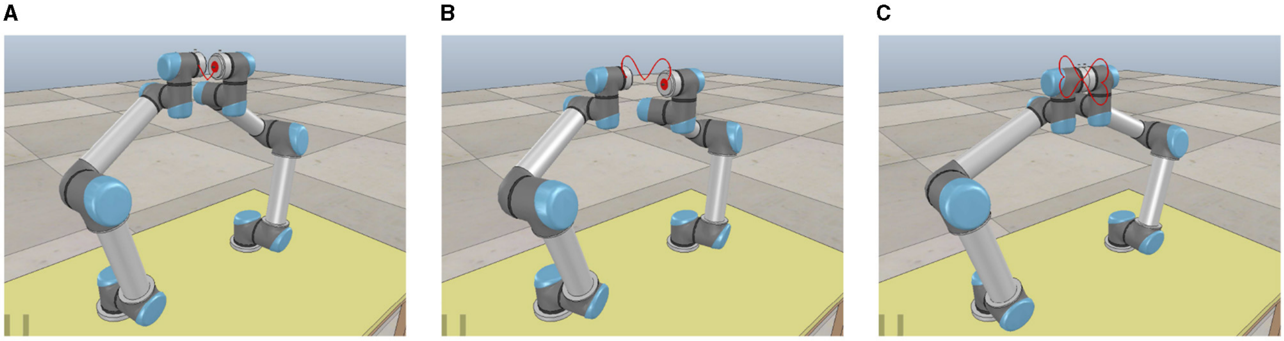

FIGURE 5
Snapshots of the UR5 dual-arm robot with two arms placed on the contralateral side during the trajectory tracking task of the dual heart-shaped path via the ATT-DRNN algorithm (43) solving the DAPCMC scheme (27)–(29). **(A)** Capturing at the starting moment. **(B)** Capturing at the intermediate moment. **(C)** Capturing at the terminal moment.

accurately with the maximal position error of the end-executor being less than 2.3×10^{-7} m. In Figure 4B, the dual-arm robot controlled by the CTT-DRNN algorithm (36) can realize the maximal tracking error of the end-executor no more than 2.3×10^{-5} m. In Figure 4C, the dual-arm robot controlled by the CET-DRNN algorithm in Wu and Zhang (2023) can merely ensure that the position error of the end-executor is within 7×10^{-4} m.

To further simulate the movement status of the dual-arm robot vividly and intuitively in the physical scene, we utilize a virtual robot experiment platform (i.e., CoppeliaSim 2020) to show the real-time status of the dual-arm robot following the ideal paths with the help of the ATT-DRNN algorithm (43) in solving the DAPCMC scheme (27)–(29). Snapshots describing the movement process (i.e., starting moment, intermediate moment, and terminal moment) are shown in Figure 5.

4.3 Control case II of dual–arm robot

For this fraction, we establish a DAPCMC scheme consisting of two UR5 robotic arms placed on the identical side, controlled by the ATT-DRNN algorithm (43) for the heart-shaped and auspicious cloud trajectory tracking.

According to the design form of the DAPCMC scheme (27), (28), we establish another particular TVQP problem for a UR5 dual-arm robot consisting of two 6-DOF UR5 robotic arms. In this scheme, the LA and RA's initial joint-angle vectors are set as $\mathfrak{z}_{\mathcal{L}}(0) = \mathfrak{z}_{\mathcal{R}}(0) = [0, -\pi/12, -2\pi/3, \pi/4, -\pi/2, 0]^{\mathrm{T}}$ rad; the LA and RA's initial joint-velocity vectors are set as $\dot{\mathfrak{z}}_{\mathcal{L}}(0) = \dot{\mathfrak{z}}_{\mathcal{R}}(0) = [\mathbf{0}]_{6\times 1}$ rad/s. We set the design parameters as $h = 0.2$, $p = 5$, $q = 0.05$, and $\delta = 2$. Then, the other correlative parameters are taken as $a = -0.5$, $\xi = 5$, and $\gamma = 0.001$; λ and μ are set to random values at the range of $(0, 0.001)$; α and β for two robotic arms are uniformly set as 0.8 and 0.1. Additionally, the joint-angle and joint-velocity physical limits in the DAPCMC scheme (27)-(28) are separately set as follows: $\mathfrak{z}^{+}_{\mathcal{L}} = 0.3$ rad/s, $\mathfrak{z}^{-}_{\mathcal{L}} = -0.3$ rad/s, $\mathfrak{z}^{-}_{\mathcal{R}} = 0.54$ rad/s, and $\mathfrak{z}^{-}_{\mathcal{R}} = -0.54$ rad/s.

Figure 6A shows the movement trajectory outlines of the dual-arm robot in a 3D space. It is evident that the end-executor real trajectory aligns seamlessly with the ideal path. Moreover, the terminal statuses of the joint angles for both robotic arms precisely coincide with their initial ones, as corroborated by the results presented in Figures 6C, D. Similarly, in Figure 6B,

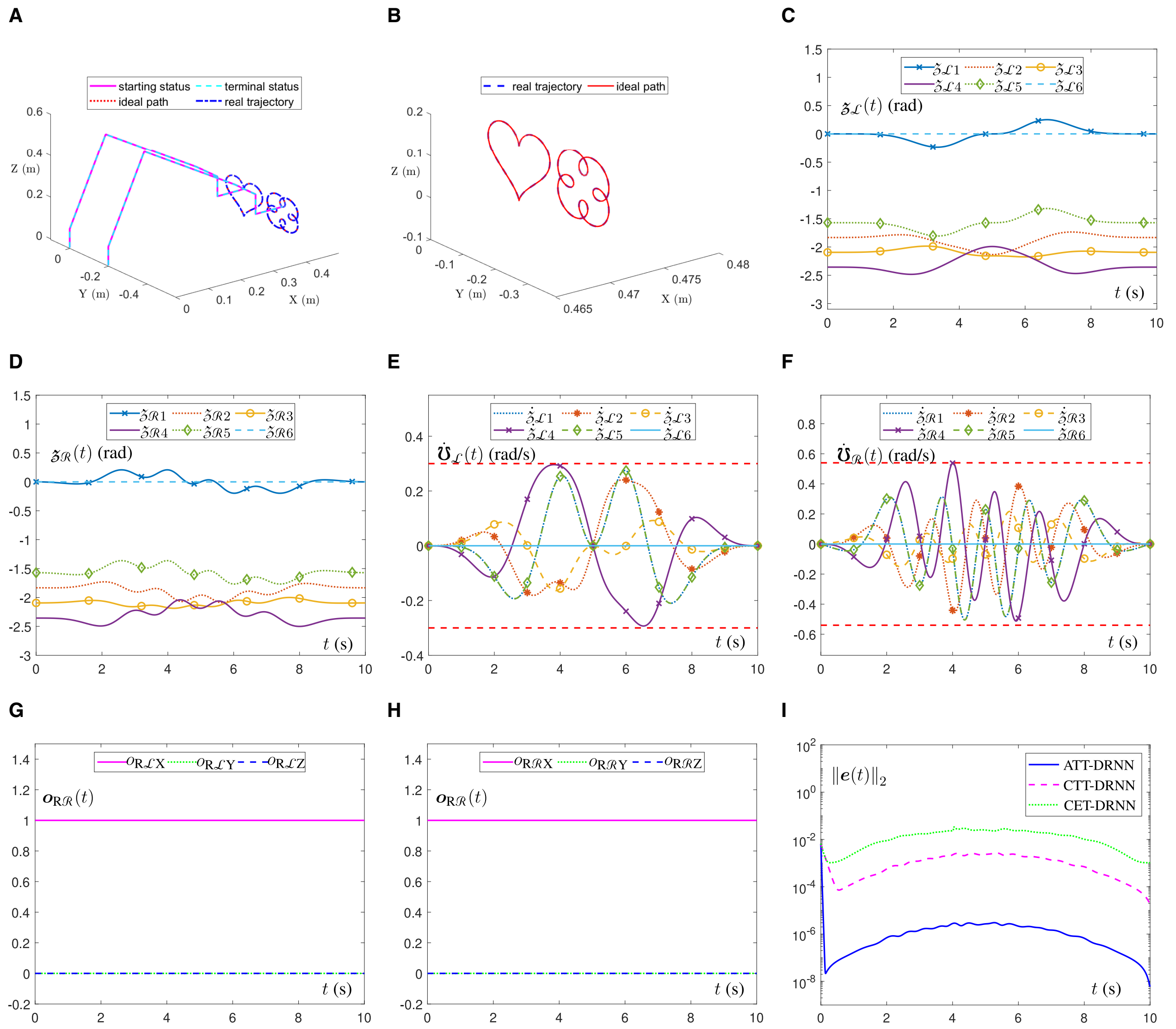

FIGURE 6

The ATT-DRNN algorithm (43) controls the posture collaboration motion of the UR5 dual-arm robot with two arms placed on the identical side for the heart-shaped and auspicious cloud trajectory tracking. **(A)** Starting and terminal statuses of the dual-arm robot and the end-executor real and ideal trajectories in a 3D space. **(B)** Outlines of real trajectory and ideal path in a 3D space. **(C)** Variations of the LA joint angles $\mathfrak{z}_{\mathcal{L}}(t)$. **(D)** Variations of the RA joint angles $\mathfrak{z}_{\mathcal{R}}(t)$. **(E)** Variations of the LA joint velocities $\dot{\mho}_{\mathcal{L}}(t)$. **(F)** Variations of the RA joint velocities $\dot{\mho}_{\mathcal{R}}(t)$. **(G)** Variations of the LA end-executor orientation $\boldsymbol{o}_{\mathrm{R}\mathcal{L}}(t)$. **(H)** Variations of the RA end-executor orientation $\boldsymbol{o}_{\mathrm{R}\mathcal{R}}(t)$. **(I)** The RE $\|\boldsymbol{e}(t)\|_2$ generated by three different algorithms.

the ATT-DRNN algorithm (43) controls the dual-arm robot to realize the posture collaboration motion and accomplish the task of tracking the heart-shaped and auspicious cloud trajectories separately. Subsequently, Figures 6E, F delineate the joint-velocity profiles of the left and right robotic arms. It is apparent that none of the joint-velocity values exceed the predetermined physical limits initially determined. Besides, the end-executor orientation variation curves are shown in Figures 6G, H, which maintain a constant state during the task execution. Furthermore, in Figure 6I, the RE $\left\|\boldsymbol{e}(t)\right\|_2$ generated by the ATT-DRNN algorithm (43) maintains at approximately 10^{-6} and converges to a extremely small value of 10^{-8}. By contrast, the RE $\left\|\boldsymbol{e}(t)\right\|_2$ generated by the CTT-DRNN algorithm (36) keeps at roughly 10^{-3} and converges to about 10^{-5} and that generated by the CET-DRNN algorithm in Wu and Zhang (2023) can merely converge to about 10^{-3}.

In addition, Figure 7 shows the position error variation curves of the end-executor when the UR5 dual-arm robot tracks the heart-shaped and auspicious cloud trajectory under the control of different algorithms. In Figure 7A, the dual-arm robot controlled by the ATT-DRNN algorithm (43) can accomplish the trajectory following task accurately with the maximal position error of the end-executor being less than 1.0×10^{-6} m. In Figure 7B, the dual-arm robot controlled by the CTT-DRNN algorithm (36) can realize the maximal position error of the end-executor no more than 1.2×10^{-4} m. In Figure 7C, the dual-arm robot controlled by the CET-DRNN algorithm in Wu and Zhang (2023) can merely ensure that the position error of the end-executor is within 1.8×10^{-3} m.

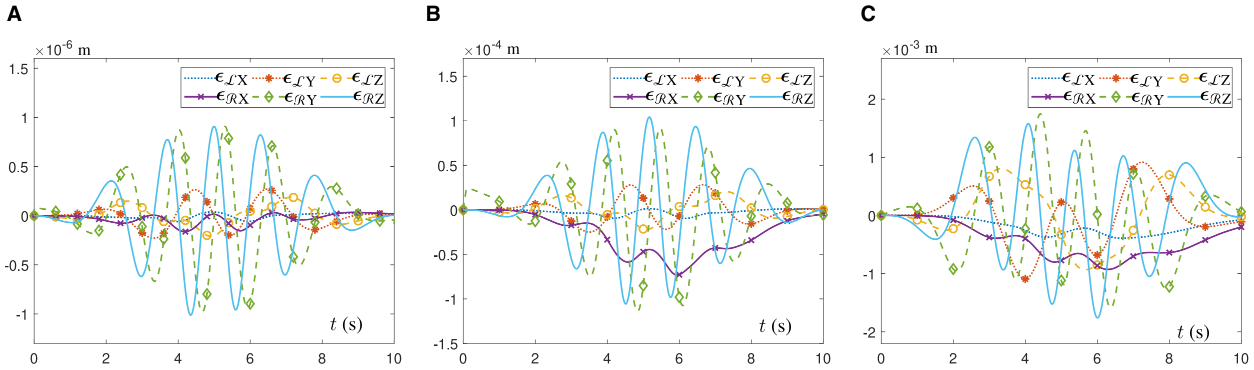

FIGURE 7
Variations of the end-executor position error when the UR5 dual-arm robot with two arms placed on the identical side to achieve the posture collaboration motion for the heart-shaped and auspicious cloud trajectory tracking. **(A)** Generated by the ATT-DRNN algorithm (43). **(B)** Generated by the CTT-DRNN algorithm (36). **(C)** Generated by the CET-DRNN algorithm (Wu and Zhang, 2023).

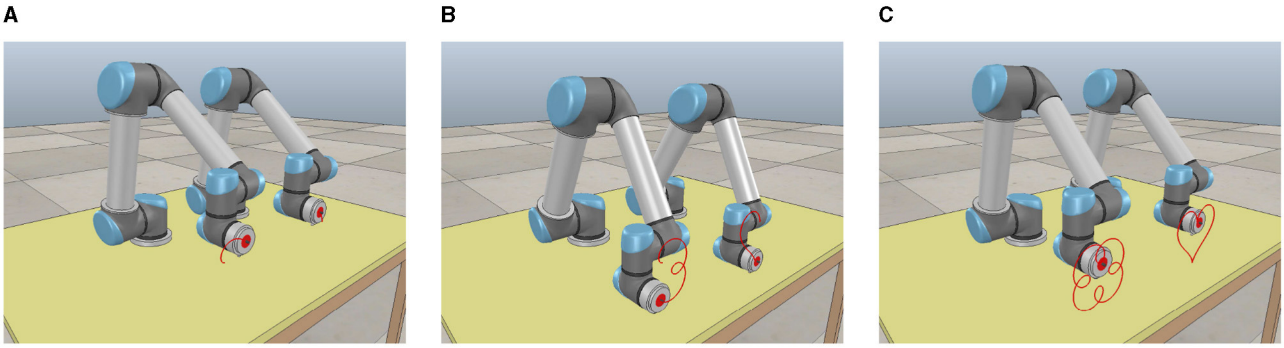

FIGURE 8
Snapshots of the UR5 dual-arm robot with two arms placed on the identical side during the trajectory tracking task of the heart-shaped and auspicious cloud path via the ATT-DRNN algorithm (43) solving the DAPCMC scheme (27)-(29). **(A)** Capturing at the starting moment. **(B)** Capturing at the intermediate moment. **(C)** Capturing at the terminal moment.

To further simulate the movement status of the dual-arm robot vividly and intuitively in the physical scene, we utilize the virtual robot experiment platform to show the real-time status of the dual-arm robot following the ideal paths with the help of the ATT-DRNN algorithm (43) in solving the DAPCMC scheme (27)–(29). Snapshots describing the movement process (i.e., starting moment, intermediate moment, and terminal moment) are shown in Figure 8.

In summary, the aforementioned two control cases of dual-arm robots substantiate that the proposed DAPCMC scheme (27)–(29) and its corresponding ATT-DRNN algorithm (43) can be utilized for the posture collaboration control of the industrial robots with joint physical limits considered and further demonstrate the potential of the proposed scheme and algorithm to optimize the efficiency and precision of repetitive trajectory tracking in practical applications.

5 Conclusion

In this study, the ATT-DRNN algorithm (43) has been devised for solving the DAPCMC scheme (27)–(29) with a novel JLCS (7), (8). Additionally, theoretical analyses and results have indicated the excellent performance of the ATT-DRNN algorithm (43) and the ACRNN model (41) in terms of the convergence rate and precision. Then, three illustrative examples with comparisons have further demonstrated that the proposed DAPCMC scheme (27)–(29) in a 3D space with the innovative JLCS (7), (8) offers a new solution measure for realizing the posture collaboration motion control of constrained dual-arm robots and accomplishing repetitive trajectory following missions, and it can be worked out by the ATT-DRNN algorithm (43) efficiently and accurately.

Finally, some possible research directions in the future are put forward.

- The whole design process of the ATT-DRNN algorithm (43) is set in an ideal noiseless environment. Therefore, enhancing the ATT-DRNN algorithm (43) with relevant anti-noise technologies to make it possess strong robustness in various noise environments is an interesting future research direction.
- The ATT-DRNN algorithm (43) involves an explicit inverse operation, which is computationally expensive. Thus,

proposing an inverse-free ATT-DRNN algorithm is another future research direction.

- Two robotic arms of the same model are used to form a dual-arm robot in this study. Thus, achieving the collaboration motion control by composing a heterogenous multi-arm robot system is a meaningful future research direction.
- Popularizing the ATT-DRNN design scheme to more kinds of engineering applications (e.g., UAV flight control) is also a significant future research direction.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

YZ: Data curation, Formal analysis, Software, Validation, Visualization, Writing – original draft. YH: Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing. BQ: Conceptualization, Funding acquisition, Investigation, Methodology, Resources, Supervision, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFC3011105, in part by the National Natural Science Foundation of China under Grant 62006254, and in part by the Shenzhen Outbound Postdoctoral Program under Grant SZBH202127.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Arents, J., and Greitans, M. (2022). Smart industrial robot control trends, challenges and opportunities within manufacturing. *Appl. Sci.-Basel* 12:937. doi: 10.3390/app12020937

Bombile, M., and Billard, A. (2022). Dual-arm control for coordinated fast grabbing and tossing of an object: proposing a new approach. *IEEE Robot. Autom. Mag.* 29, 127–138. doi: 10.1109/MRA.2022.3177355

Cai, J., and Yi, C. (2023). An adaptive gradient-descent-based neural networks for the on-line solution of linear time variant equations and its applications. *Inf. Sci.* 622, 34–45. doi: 10.1016/j.ins.2022.11.157

Cheng, Z., Feng, W., Zhang, Y., Sun, L., Liu, Y., Chen, L., et al. (2023). A highly robust amphibious soft robot with imperceptibility based on a water-stable and self-healing ionic conductor. *Adv. Mater. Weinheim.* 13:2301005. doi: 10.1002/adma.202301005

Chico, A., Cruz, P. J., Vásconez, J. P., Benalcázar, M. E., Álvarez, R., Barona, L., et al. (2021). "Hand gesture recognition and tracking control for a virtual UR5 robot manipulator," in *2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM)* (Cuenca, Ecuador: IEEE), 1–6.

Dai, J., Chen, Y., Xiao, L., Jia, L., and He, Y. (2022). Design and analysis of a hybrid GNN-ZNN model with a fuzzy adaptive factor for matrix inversion. *IEEE Trans. Ind. Inform.* 18, 2434–2442. doi: 10.1109/TII.2021.3093115

Ekrem, Ö., and Aksoy, B. (2023). Trajectory planning for a 6-axis robotic arm with particle swarm optimization algorithm. *Eng. Appl. Artif. Intell.* 122:106099. doi: 10.1016/j.engappai.2023.106099

Fu, Z., Zhang, Y., and Tan, N. (2023). Gradient-feedback ZNN for unconstrained time-variant convex optimization and robot manipulator application. *IEEE Trans. Ind. Inform.* 19, 10489–10500. doi: 10.1109/TII.2023.3240737

Hu, C., Kang, X., and Zhang, Y. (2018). Three-step general discrete-time Zhang neural network design and application to time-variant matrix inversion. *Neurocomputing* 306, 108–118. doi: 10.1016/j.neucom.2018.03.053

Isidori, A. (1989). *Nonlinear Control Systems: An Introduction*. Springer-Verlag, Berlin: Germany.

Jiang, Y., Wang, Y., Miao, Z., Na, J., Zhao, Z., and Yang, C. (2022). Composite-learning-based adaptive neural control for dual-arm robots with relative motion. *IEEE Trans. Neural Netw. Learn. Syst.* 33, 1010–1021. doi: 10.1109/TNNLS.2020.3037795

Jin, L., Liu, L., Wang, X., Shang, M., and Wang, F.-Y. (2024). Physical-informed neural network for MPC-based trajectory tracking of vehicles with noise considered. *IEEE Trans. Intell. Veh.* 9, 4493–4503. doi: 10.1109/TIV.2024.3358229

Kastritsi, T., and Doulgeri, Z. (2021). A controller to impose a RCM for hands-on robotic-assisted minimally invasive surgery. *IEEE Trans. Med. Robot. Bionics* 3, 392–401. doi: 10.1109/TMRB.2021.3077319

Khan, Z. H., Siddique, A., and Lee, C. W. (2020). Robotics utilization for healthcare digitization in global COVID-19 management. *Int. J. Environ. Res. Public Health* 17:3819. doi: 10.3390/ijerph17113819

Li, W. (2020). Predefined-time convergent neural solution to cyclical motion planning of redundant robots under physical constraints. *IEEE Trans. Ind. Electron.* 67, 10732–10743. doi: 10.1109/TIE.2019.2960754

Li, W., Chiu, P. W. Y., and Li, Z. (2023). A novel neural approach to infinity-norm joint-velocity minimization of kinematically redundant robots under joint limits. *IEEE Trans. Neural Netw. Learn. Syst.* 34, 409–420. doi: 10.1109/TNNLS.2021.3095122

Liao, B., Zhang, Y., and Jin, L. (2016). Taylor $O(h^3)$ discretization of ZNN models for dynamic equality-constrained quadratic programming with application to manipulators. *IEEE Trans. Neural Netw. Learn. Syst.* 27, 225–237. doi: 10.1109/TNNLS.2015.2435014

Liu, M., Luo, W., Cai, Z., Du, X., Zhang, J., and Li, S. (2023a). Numerical-discrete-scheme-incorporated recurrent neural network for tasks in natural language processing. *CAAI Trans. Intell. Technol.* 8, 1415–1424. doi: 10.1049/cit2.12172

Liu, M., Wu, H., Shi, Y., and Jin, L. (2023b). High-order robust discrete-time neural dynamics for time-varying multi-linear tensor equation with *M*-tensor. *IEEE Trans. Ind. Inform.* 19, 9457–9467. doi: 10.1109/TII.2022.3228394

Liufu, Y., Jin, L., Shang, M., Wang, X., and Wang, F.-Y. (2024). ACP-incorporated perturbation-resistant neural dynamics controller for autonomous vehicles. *IEEE Trans. Intell. Veh.* doi: 10.1109/TIV.2023.3348632

McCartney, G., and McCartney, A. (2020). Rise of the machines: Towards a conceptual service-robot research framework for the hospitality and tourism industry. *Int. J. Contemp. Hosp. Manag.* 32, 3835–3851. doi: 10.1108/IJCHM-05-2020-0450

Qiu, B., Guo, J., Yang, S., Yu, P., and Tan, N. (2023). A novel discretized ZNN model for velocity layer weighted multicriteria optimization of robotic manipulators with multiple constraints. *IEEE Trans. Ind. Inform*. 19, 6717–6728. doi: 10.1109/TII.2022.3197270

Shi, Y., and Zhang, Y. (2018). Discrete time-variant nonlinear optimization and system solving via integral-type error function and twice ZND formula with noises suppressed. *Soft Comput*. 22, 7129–7141. doi: 10.1007/s00500-018-3020-5

Shi, Y., Zhao, W., Li, S., Li, B., and Sun, X. (2023). Novel discrete-time recurrent neural network for robot manipulator: A direct discretization technical route. *IEEE Trans. Neural Netw. Learn. Syst*. 34, 2781–2790. doi: 10.1109/TNNLS.2021.3108050

Song, Q., Wu, Y., and Soh, Y. C. (2008). Robust adaptive gradient-descent training algorithm for recurrent neural networks in discrete time domain. *IEEE Trans. Neural Netw*. 19, 1841–1853. doi: 10.1109/TNN.2008.2001923

Stolfi, A., Gasbarri, P., and Sabatini, M. (2017). A combined impedance-PD approach for controlling a dual-arm space manipulator in the capture of a non-cooperative target. *Acta Astronaut*. 139, 243–253. doi: 10.1016/j.actaastro.2017.07.014

Tanyıldızı, A. K. (2023). Design, control and stabilization of a transformable wheeled fire fighting robot with a fire-extinguishing, ball-shooting turret. *Machines* 11:492. doi: 10.3390/machines11040492

Vivas, A., and Sabater, J. M. (2021). "UR5 robot manipulation using Matlab/Simulink and ROS," in *2021 IEEE International Conference on Mechatronics and Automation (ICMA)* (Takamatsu: IEEE), 338–343.

Wang, S., Jin, L., Du, X., and Stanimirovi, P. S. (2021). Accelerated convergent zeroing neurodynamics models for solving multi-linear systems with M-tensors. *Neurocomputing* 458, 271–283. doi: 10.1016/j.neucom.2021.06.005

Wang, Y., Li, H., Zhao, Y., Chen, X., Huang, X., and Jiang, Z. (2023). A fast coordinated motion planning method for dual-arm robot based on parallel constrained DDP. *IEEE-ASME Trans. Mechatron*. doi: 10.1109/TMECH.2023.3323798

Wei, L., Jin, L., and Luo, X. (2022). Noise-suppressing neural dynamics for time-dependent constrained nonlinear optimization with applications. *IEEE Trans. Syst., Man, Cybern., Syst*., 52, 6139–6150. doi: 10.1109/TSMC.2021.3138550

Wu, W., and Zhang, Y. (2023). Novel adaptive zeroing neural dynamics schemes for temporally-varying linear equation handling applied to arm path following and target motion positioning. *Neural Netw*. 165, 435–450. doi: 10.1016/j.neunet.2023.05.056

Xiao, L., Dai, J., Jin, L., Li, W., Li, S., and Hou, J. (2021). A noise-enduring and finite-time zeroing neural network for equality-constrained time-varying nonlinear optimization. *IEEE Trans. Syst. Man Cybern. Syst*. 51, 4729–4740. doi: 10.1109/TSMC.2019.2944152

Yan, J., Jin, L., Yuan, Z., and Liu, Z. (2022). RNN for receding horizon control of redundant robot manipulators. *IEEE Trans. Ind. Electron*. 69, 1608–1619. doi: 10.1109/TIE.2021.3062257

Yang, M., Zhang, Y., and Hu, H. (2021). Posture coordination control of two-manipulator system using projection neural network. *Neurocomputing* 427, 179–190. doi: 10.1016/j.neucom.2020.11.012

Yang, M., Zhang, Y., Zhang, Z., and Hu, H. (2020). Adaptive discrete ZND models for tracking control of redundant manipulator. *IEEE Trans. Ind. Inform*. 16, 7360–7368. doi: 10.1109/TII.2020.2976844

Yang, Q., Du, X., Wang, Z., Meng, Z., Ma, Z., and Zhang, Q. (2023). A review of core agricultural robot technologies for crop productions. *Comput. Electron. Agric*. 206:107701. doi: 10.1016/j.compag.2023.107701

Yang, S., Wen, H., Hu, Y., and Jin, D. (2020). Coordinated motion control of a dual-arm space robot for assembling modular parts. *Acta Astronaut*. 177, 627–638. doi: 10.1016/j.actaastro.2020.08.006

Zhang, Y., and Zhang, Z. (2013). *Repetitive Motion Planning and Control of Redundant Robot Manipulators*. New York: Springer Science & Business Media, Springer-Verlag.

Zhang, Z., Lin, Y., Li, S., Li, Y., Yu, Z., and Luo, Y. (2018). Tricriteria optimization-coordination motion of dual-redundant-robot manipulators for complex path planning. *IEEE Trans. Control Syst. Technol.* 26, 1345–1357. doi: 10.1109/TCST.2017.2709276