Knowledge-Based Systems 254 (2022) 109624

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Adaptive dynamic event-triggered control for constrained modular reconfigurable robot

Ruizhuo Song [a,*], Lu Liu [a], Zhen Xu [b]

[a] Beijing Engineering Research Center of Industrial Spectrum Imaging, School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China
[b] Research Institute of Urbanization and Urban Safety, School of Civil and Resource Engineering, University of Science and Technology Beijing, Beijing 100083, China

ARTICLE INFO

Article history:
Received 17 May 2022
Received in revised form 27 July 2022
Accepted 4 August 2022
Available online 13 August 2022

Keywords:
Adaptive dynamic programming
Constrained input
Dynamic event-triggered control
Modular reconfigurable robot

ABSTRACT

Compared with traditional robot, modular reconfigurable robot (MRR) has the advantages of strong environmental adaptability and flexible task completion. According to the optimal tracking control problem (OTCP) of MRR under some restricted conditions, this paper puts forward a constrained dynamic event-triggered control (DETC) for MRR system with disturbance through adaptive dynamic programming (ADP), which can minimize the information interaction quantity under the premise of system stability and expected control effect. In view of the uncertainty of model coupling part, the identification network is used to estimate the dynamics of MRR and the estimation error is proved to be uniformly ultimate bounded (UUB). The other three groups of critic, action and disturbance neural networks (NNs) are established by the approximation principle of ADP. The optimal control pair is obtained through policy iteration (PI) with DETC, and the triggering condition is designed based on the asymptotic stability of MRR system. At last, the strengths of the algorithm in this paper are validated through simulation experiments.

1. Introduction

Robot is the crystallization of human intelligence, which is a machine controlled by human to imitate human behavior to work, with a wide range of applications in service industry, intelligent medical care, aerospace and other fields. With the rapid development of global technology, working environment of robots becomes more and more complex and accompanied by some perturbance, which requires robots to be able to execute tedious tasks flexibly [1–3]. However, it is very expensive to research and develop the corresponding robots according to different types of operations. Therefore, it is of great value to study the modular reconfigurable robot (MRR) which can change its shape and posture in line with different requirements. Joint modules can be reconfigured for different tasks so that the robot has different functions and stronger environmental adaptability [4–6]. B. Ma et al. transformed the control of MR in an uncertain environment into a non-zero-sum differential equation and obtained the optimal solution in [7]. The policy iteration algorithm based on single network structure for MR was researched in [8]. H. Xia et al. studied the optimal sliding mode tracking control for MRR in [9]. Moreover, in practical work, robots are often interfered by unavoidable constraints such as physical conditions, accordingly, it is essential to study how to find the solution of optimal tracking control problem (OTCP) for constrained MRR systems [10–12].

As a nonlinear system, some uncertainties exist in the MRR system due to the coupling of various joints, and it is very difficult to solve the relevant Hamilton–Jacobi–Bellman (HJB) equation. As one of optimal control solutions for nonlinear systems by neural networks (NNs) approximation, adaptive dynamic programming (ADP) can effectively reduce the influence of unknown and nonlinear terms by estimation, which has been widely used in intelligent control field [13–15]. The advantages of ADP lie in that it can timely respond to internal dynamics changes and external feedback signals, adjust the network structure parameters, and finally complete the approximation of control sequences. In recent years, in order to achieve the ideal performance while avoiding complex operations, some scholars have applied ADP to OTCP of robots [16–18]. For instance, S. Li et al. proposed an online ADP tracking control method for wheeled mobile robots [19], and Y. Wen et al. first applied ADP to human clinical experiment by automatically tuning the robotic knee prosthesis [20]. In general, the approximation structure and adaptive ability of ADP can efficiently deal with the control problems of nonlinear robot systems with unknown coupling.

For modern control systems, performance and cost need to be considered simultaneously, the control strategy updating mode

[*] Corresponding author.
E-mail address: ruizhuosong@ustb.edu.cn (R. Song).

https://doi.org/10.1016/j.knosys.2022.109624
0950-7051/© 2022 Elsevier B.V. All rights reserved.

based on time-triggered control (TTC) which is extensively available in traditional control may do not perform well for a type of systems with limited communication resources. As a mechanism which updates the control law only under some certain conditions, event-triggered control (ETC) can balance the relationship between depletable resources and performance stability of the system in order to avoid communication bandwidth congestion and mechanical loss caused by the actuator being triggered frequently [21–23]. Some scholars have combined ETC with ADP, and this associative mode can achieve excellent control results in some nonlinear systems. D. Yang et al. researched a hybrid event-triggered integral sliding mode control, and applied robust ADP in a single NNs to approximate the control strategy for a constrained system in [24]. In terms of the dual heuristic dynamic programming (DHDP) structure of ADP, the optimal saturation actuator was designed under ETC within system stability for a non-affine system in [25]. Based on the triggering condition in conventional ETC, a dynamic variable is inserted to design a new triggering condition, that means adding another restriction on the basis of the original threshold, and this triggered mechanism is called as dynamic event-triggered control (DETC). Compared with ETC, DETC has a larger minimum triggering interval, which can further reduce the system sampling frequency and achieve the purpose of saving system resources, thus improving the performance of ETC [26–28].

Inspired by the mentioned literature, this paper studies a DET PI solution for OTCP of the constrained-input partly-unknown MRR with disturbance, the contribution are illustrated in the following:

(1) In order to ensure the feasibility of the reconstructed shape of MRR for the given task, the control input vector of the module subsystem is constrained in this paper. The control sequence is solved by designing a value function with a non-quadratic utility function containing a bounded function instead of the conventional quadratic form.

(2) On account of the reconfigurable nature of MRR, the unknown interconnected dynamic coupling part is approximated through building up an identification network and the resulting estimation error is proved to be uniformly ultimate bounded (UUB).

(3) Based on the ET condition, DET condition is designed to extend the sampling interval of system by inserting the dynamic variable into the triggering threshold. Moreover, the stability of MRR system is proved and the Zeno phenomenon is analyzed by the minimum trigger interval and simulation experiments.

Section 2 describes the MRR model as the research object and builds the system identification network. Section 3 obtains the optimal control sequence through ADP. The ET and DET conditions are acquired in Section 4. The simulation results are demonstrated and analyzed in Section 5, and brief conclusions are given in Section 6.

2. MRR knowledge

This section introduces the dynamic model of MRR. Taking the modularized robot that adjusts shape according to a prescribed task into consideration, the configuration change of MRR may be unknown. Hence, the research object in this paper is partially unknown, and the unknown part of MRR is simulated by establishing an identification network.

2.1. Dynamics description

The MRR consists of a series of joints with different functions as shown in Fig. 1. The dynamic system model based on the Newton-Euler iterative algorithm is described as

$$M(p)\ddot{p} + C(p, \dot{p}) + G(p) + D(p) = u \tag{1}$$

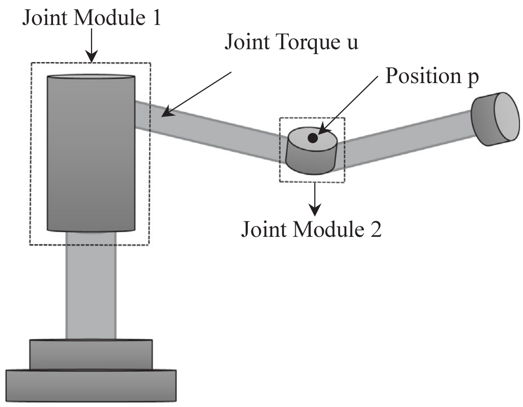

Fig. 1. The structure of MRR.

where p is the position of the joint, M expresses inertia matrix, C is the centripetal Coriolis matrix, G implies the earth gravity and D represents the disturbance torque. As the control input, u is the constrained joint torque applied to the robot.

Each module can be considered as a subsystem and the ith dynamic model is

$$M_i(p_i)\ddot{p}_i + C_i(p_i, \dot{p}_i)\dot{p}_i + G(p_i) + d_i(p) + T_i(p, \dot{p}_i, \ddot{p}_i) = u_i \tag{2}$$

where T_i is the interconnected dynamic coupling term of the module and expressed as

$$\begin{aligned} T_i = & \left\{ \sum_{m=1, m\neq i}^{n} M_{im}(p)\ddot{p}_m + [M_{ii}(p) - M_i(p_i)]\ddot{p}_i \right\} \\ & + \left\{ \sum_{m=1, m\neq i}^{n} C_{im}(p, \dot{p})\dot{p}_m + [C_{ii}(p, \dot{p}) - C_i(p_i, \dot{p}_i)]\dot{p}_i \right\} \\ & + \bar{G}_i(p) - G_i(p_i) \end{aligned} \tag{3}$$

where p_i, $\dot{p}_i$, $\ddot{p}_i$, $\bar{G}_i$, d_i and u_i are the ith element vector of p, $\dot{p}$, $\ddot{\text{p}}$, G, D and u. $M_{im}(p)$ and $C_{im}(p, \dot{p})$ are the im-th element of matrices $M(p)$ and $C(p, \dot{p})$.

In order to facilitate the subsequent design of value function, we define the state as $x_i = [x_{i1}, x_{i2}]^T = [p_i, \dot{p}_i]^T$, and the module is described as

$$\begin{aligned} \dot{x}_{i1} &= x_{i2} \\ \dot{x}_{i2} &= f_i(x_i) + g_i(x_i)u_i + d_i(x_i) + h_i(x_i) \end{aligned} \tag{4}$$

where

$$\begin{aligned} f_i(x_i) &= [-G_i(p_i) - C_i(p_i, \dot{p}_i)\dot{p}_i]M_i^{-1}(p_i) \\ g_i(x_i) &= M_i^{-1}(p_i) \\ h_i(x_i) &= -T_i(p, \dot{p}_i, \ddot{p}_i)M_i^{-1}(p_i). \end{aligned} \tag{5}$$

Considering that the joint torque force is finite in practical control, so the joint torque control input of the research object is constrained, and satisfies $|u_i| \leq u_{ib}$, where u_{ib} is the boundary of the control input vector component and $u_{ib} > 1$.

Definition 1. *If the control input u_i is continuous on Ω_u and $u_i(0) = 0$, the admissible control policy is defined as $u_i \in \Psi_\Omega$, u_i stabilizes Eq. (4) on Ω_u.*

2.2. System identification

MRR needs to change its shape according to different tasks, however, system dynamics f_i, g_i and the interconnected dynamic coupling part h_i may be unknown in practical application. Thus, the unknown terms are depicted by the identification NN.

According to Eq. (4), the system can be constructed as

$$
\begin{aligned}
\dot{x}_i =& f_i(x_i) + g_i(x_i)u_i + h_i(x_i) + d_i(x_i) \\
=& ({w_f}^T\phi_f(x_i) + \varepsilon_f) + ({w_g}^T\phi_g(x_i) + \varepsilon_g)u_i \\
&+ ({w_h}^T\phi_h(x_i) + \varepsilon_h) + d_i(x_i)
\end{aligned} \tag{6}
$$

where $x_i = [x_{i1}, x_{i2}]^T$, w_f, w_g and w_h are ideal networks weights, ϕ_f, ϕ_g and ϕ_h represent the activation functions, ε_f, ε_g and ε_h are on behalf of the reconstruction errors.

The estimation form of the above equation is obtained as

$$
\dot{\hat{x}}_i = \hat{w}_f^T\phi_f(\hat{x}_i) + \hat{w}_g^T\phi_g(\hat{x}_i)u_i + \hat{w}_h^T\phi_h(\hat{x}_i) + d_i(x_i) \tag{7}
$$

where $\hat{x}_i = [\hat{x}_{i1}, \hat{x}_{i2}]^T$ is the estimation vector of system state, $\hat{w}_f$, $\hat{w}_g$, $\hat{w}_h$, $\phi_f(\hat{x}_i)$, $\phi_g(\hat{x}_i)$ and $\phi_h(\hat{x}_i)$ are estimation forms of w_f, w_g, w_h, $\phi_f(x_i)$, $\phi_g(x_i)$ and $\phi_h(x_i)$.

The estimation error is defined as

$$
\begin{aligned}
\tilde{x}_i &= x_i - \hat{x}_i \\
\tilde{w}_f &= w_f - \hat{w}_f \\
\tilde{w}_h &= w_h - \hat{w}_h \\
\tilde{w}_g &= w_g - \hat{w}_g .
\end{aligned} \tag{8}
$$

Then, the update rates of $\hat{w}_f$, $\hat{w}_g$ and $\hat{w}_h$ are designed as

$$
\begin{aligned}
\dot{\hat{w}}_f &= A_f\phi_f(\hat{x}_i)\tilde{x}_i^T \\
\dot{\hat{w}}_g &= A_g\phi_g(\hat{x}_i)\tilde{x}_i^T \\
\dot{\hat{w}}_h &= A_h\phi_h(\hat{x}_i)\tilde{x}_i^T
\end{aligned} \tag{9}
$$

where A_f, A_g and A_h are designed matrices.

Based on Eqs. (7) and (8), we have

$$
\begin{aligned}
\dot{\tilde{x}}_i =& \dot{x}_i - \dot{\hat{x}}_i \\
=& \tilde{w}_f^T\phi_f(\hat{x}_i) + w_f^T(\phi_f(x_i) - \phi_f(\hat{x}_i)) + \varepsilon_f \\
&+ \tilde{w}_g^T\phi_g(\hat{x}_i)u_i + w_g^T(\phi_g(x_i) - \phi_g(\hat{x}_i))u_i + \varepsilon_g u_i \\
&+ \tilde{w}_h^T\phi_h(\hat{x}_i) + w_h^T(\phi_h(x_i) - \phi_h(\hat{x}_i)) + \varepsilon_h .
\end{aligned} \tag{10}
$$

Assumption 1. The activation function, reconstruction error and network weights of identification NN have upper bound, they are bounded by $\|\phi_f\| \le \phi_{fm}$, $\|\phi_g\| \le \phi_{gm}$, $\|\phi_h\| \le \phi_{hm}$, $\|\varepsilon_f\| \le \varepsilon_{fm}$, $\|\varepsilon_g\| \le \varepsilon_{gm}$, $\|\varepsilon_h\| \le \varepsilon_{hm}$, $\|w_f\| \le w_{fm}$, $\|w_g\| \le w_{gm}$ and $\|w_h\| \le w_{hm}$, where ϕ_{fm}, ϕ_{gm}, ϕ_{hm}, ε_{fm}, ε_{gm}, ε_{hm}, w_{fm}, w_{gm} and w_{hm} are positive constants.

Assumption 2. The activation functions are Lipschitz continuous and satisfy the following inequalities:

$$
\begin{aligned}
&\|\phi_f(x_i) - \phi_f(\hat{x}_i)\| \le B_1\|\tilde{x}_i\| \\
&\|\phi_g(x_i) - \phi_g(\hat{x}_i)\| \le B_2\|\tilde{x}_i\| \\
&\|\phi_h(x_i) - \phi_h(\hat{x}_i)\| \le B_3\|\tilde{x}_i\|
\end{aligned}
$$

where B_1, B_2 and B_3 are positive constants.

Theorem 1. *Based on the above assumptions, the update rules in Eq. (9) for estimated weights of identification NN can make the estimation error be uniformly ultimate bounded (UUB).*

Proof. The Lyapunov function is defined as

$$
\begin{aligned}
L =& L_x + L_w \\
=& \frac{1}{2}\tilde{x}_i^T\tilde{x}_i + \frac{1}{2}\mathrm{tr}(\tilde{w}_f^T{A_f}^{-1}\tilde{w}_f) + \frac{1}{2}\mathrm{tr}(\tilde{w}_g^T{A_g}^{-1}\tilde{w}_g) + \frac{1}{2}\mathrm{tr}(\tilde{w}_h^T{A_h}^{-1}\tilde{w}_h)
\end{aligned} \tag{11}
$$

where

$$
\begin{aligned}
L_x &= \frac{1}{2}\tilde{x}_i^T\tilde{x}_i \\
L_w &= \frac{1}{2}\mathrm{tr}(\tilde{w}_f^T{A_f}^{-1}\tilde{w}_f) + \frac{1}{2}\mathrm{tr}(\tilde{w}_g^T{A_g}^{-1}\tilde{w}_g) + \frac{1}{2}\mathrm{tr}(\tilde{w}_h^T{A_h}^{-1}\tilde{w}_h).
\end{aligned}
$$

According to Assumptions 1 and 2, the identifier error has the following upper bound as

$$
\begin{aligned}
\dot{\tilde{x}}_i \le& \tilde{w}_f^T\phi_f(\hat{x}_i) + \tilde{w}_g^T\phi_f(\hat{x}_i)u_i + \tilde{w}_h^T\phi_h(\hat{x}_i) + w_{fm}^TB_1\tilde{x}_i \\
&+ w_{gm}^TB_2u_{ib}\tilde{x}_i + w_{hm}^TB_3\tilde{x}_i + \varepsilon_{fm} + \varepsilon_{gm}u_{ib} + \varepsilon_{hm}.
\end{aligned} \tag{12}
$$

Thus, $\dot{L}_x$ satisfies

$$
\begin{aligned}
\dot{L}_x \le& \tilde{x}_i^T[\tilde{w}_f^T\phi_f(\hat{x}_i) + \tilde{w}_g^T\phi_f(\hat{x}_i)u_{ib} + \tilde{w}_h^T\phi_h(\hat{x}_i) + w_{fm}^TB_1\tilde{x}_i \\
&+ w_{gm}^TB_2u_{ib}\tilde{x}_i + w_{hm}^TB_3\tilde{x}_i + \varepsilon_{fm} + \varepsilon_{gm}u_{ib} + \varepsilon_{hm}] \\
\le& \tilde{x}_i^T[\tilde{w}_f^T\phi_f(\hat{x}_i) + \tilde{w}_g^T\phi_f(\hat{x}_i)u_{ib} + \tilde{w}_h^T\phi_h(\hat{x}_i)] \\
&+ \tilde{x}_i^T[w_{fm}^TB_1 + w_{gm}^TB_2u_{ib} + w_{hm}^TB_3]\tilde{x}_i + \tilde{x}_i^T\varepsilon_m \\
\le& \tilde{x}_i^T[\tilde{w}_f^T\phi_f(\hat{x}_i) + \tilde{w}_g^T\phi_g(\hat{x}_i)u_{ib} + \tilde{w}_h^T\phi_h(\hat{x}_i)] + \tilde{x}_i^T(-A_x)\tilde{x}_i + \tilde{x}_i^T\varepsilon_m
\end{aligned} \tag{13}
$$

where $\varepsilon_m = \varepsilon_{fm} + \varepsilon_{gm}u_{ib} + \varepsilon_{hm}$ and $-A_x = w_{fm}^TB_1 + w_{gm}^TB_2u_{ib} + w_{hm}^TB_3$ is a positive definite matrix.

Based on Eqs. (8) and (9), the derivative form of weight estimation errors are

$$
\begin{aligned}
\dot{\tilde{w}}_f &= -\dot{\hat{w}}_f = -A_f\phi_f(\hat{x}_i)\tilde{x}_i^T \\
\dot{\tilde{w}}_g &= -\dot{\hat{w}}_g = -A_g\phi_g(\hat{x}_i)\tilde{x}_i^T \\
\dot{\tilde{w}}_h &= -\dot{\hat{w}}_h = -A_h\phi_h(\hat{x}_i)\tilde{x}_i^T .
\end{aligned} \tag{14}
$$

As a result, the derivative of the second term of Lyapunov function can be achieved as

$$
\dot{L}_w = -\tilde{w}_f^T\phi_f(\hat{x}_i)\tilde{x}_i^T - \tilde{w}_g^T\phi_g(\hat{x}_i)\tilde{x}_i^T - \tilde{w}_h^T\phi_h(\hat{x}_i)\tilde{x}_i^T . \tag{15}
$$

To summarize them together, we have

$$
\begin{aligned}
\dot{L} \le& \tilde{x}_i^T(-A_x)\tilde{x}_i + \tilde{x}_i^T\varepsilon_m + \tilde{x}_i^T\tilde{w}_g^T\phi_g(\hat{x}_i)(u_{ib} - 1) \\
\le& -\|\tilde{x}_i\|^2\lambda_{\min}(A_x) + \|\tilde{x}_i\|\varepsilon_m + \|\tilde{x}_i\|w_{gm}\phi_{gm}(u_{ib} - 1)
\end{aligned} \tag{16}
$$

The inequality above implies that when $\|\tilde{x}_i\| > \dfrac{\varepsilon_m + w_{gm}\phi_{gm}(u_{ib} - 1)}{\lambda_{\min}(A_x)}$, the derivative of the Lyapunov function has $\dot{L} < 0$, thus the identifier error is UUB.

3. Optimal tracking control via ADP

This section deals with OTCP for MRR through approximation principle of ADP. The optimal control sequence can be obtained by solving the designed value function, after that, the subsystem can track the given trajectory with the optimal control law.

3.1. Optimal control pair

Before solving the HJB equation to get the time-triggered optimal control pair, an auxiliary system is set up firstly.

The tracking trajectory is defined as

$$
\dot{x}_{ip} = f(x_{ip}). \tag{17}
$$

The tracking error can be calculated as

$$
\wp_e = x_i - x_{ip}, \tag{18}
$$

and the dynamic equation of tracking error is

$$
\dot{\wp}_e = f_i(x_i) + g_i(x_i)u_i + d_i(x_i) + h_i(x_i) - f(x_{ip}). \tag{19}
$$

Next, the auxiliary dynamic system in this OTCP is devised as

$$
\dot{\wp} = f(\wp) + g(\wp)u_i + d_i(\wp) + h(\wp) \tag{20}
$$

where the auxiliary system state is $\wp = [{\wp_e}^T, {x_{ip}}^T]^T$, accordingly, $f(\wp) = [(f_i(x_i) - f(x_{ip}))^T, f(x_{ip})^T]^T$, $g(\wp) = [g_i(\wp_e + x_{ip}), 0]^T$, $d_i(\wp) = [d_i(\wp_e + x_{ip}), 0]^T$ and $h(\wp) = [h_i(\wp_e + x_{ip}), 0]^T$.

The approximate value function about the tracking error is designed as

$$V(\wp) = \int_0^\infty P(\wp, u_i, d_i) d\tau \tag{21}$$

and the utility function $P(\wp, u_i, d_i)$ is

$$\begin{aligned} P(\wp, u_i, d_i) =& \wp^T Q \wp + P_u(u_i) - \gamma^2 d_i^T d_i \\ =& \wp^T Q \wp + \int_0^{u_i} 2u_{ib} \tanh^{-1}\left(\frac{\nu}{u_{ib}}\right)^T R d\nu - \gamma^2 d_i^T d_i \end{aligned} \tag{22}$$

where R is the positive definite diagonal matrix, $\gamma > 0$.

Remark 1. In order to describe the actual MRR system more accurately and ensure the satisfactory closed-loop performance, this section considers input limitation when designing the value function. The smooth and differentiable hyperbolic tangent function is selected to approximate the constraint of the control input ($-u_{ib} \leq u_i \leq u_{ib}$). The energy consumption of control input P_u is in the form of a bounded function $tanh(\cdot)$ rather than the regular quadratic form as $u_i^T R u_i$.

$P_u(u_i)$ can be deduced further as

$$\begin{aligned} P_u(u_i) =& \int_0^{u_i} 2u_{ib} \tanh^{-1}\left(\frac{\nu}{u_{ib}}\right)^T R d\nu \\ =& 2u_{ib} {u_i}^T R \tanh^{-1}\left(\frac{u_i}{u_{ib}}\right) + {u_{ib}}^2 \bar{R} \ln\left(\mathbf{1} - \frac{u_i^2}{{u_{ib}}^2}\right) \end{aligned} \tag{23}$$

where $\bar{R}$ is a positive definite horizontal vector, **1** is a all 1 column vector.

The Hamilton function can be achieved as

$$\begin{aligned} & H(\wp, u_i, d_i, \nabla V) \\ =& \wp^T Q \wp - \gamma^2 d_i^T d_i + P_u(u_i) + \nabla V^T(\wp)[f(\wp) \\ & + g(\wp) u_i + d_i(\wp) + h(\wp)]. \end{aligned} \tag{24}$$

where $\nabla V = \dfrac{\partial V(\wp)}{\partial \wp}$ is the derivative form of $V(\wp)$ with respect to the auxiliary state $\wp$ and $V(0) = 0$.

The above equation satisfies the following HJB equation

$$\min_{u_i} \max_{d_i} H(\wp, u_i, d_i, \nabla V^*) = 0. \tag{25}$$

By solving the HJB equation, the Nash equilibrium $({u_i}^*, {d_i}^*)$ can be obtained as

$${u_i}^* = -u_{ib} \tanh\left(\frac{1}{2u_{ib}} R^{-1} g^T(\wp) \nabla V^*(\wp)\right) \tag{26}$$

$${d_i}^* = \frac{1}{2\gamma^2} \nabla V^*(\wp). \tag{27}$$

Inserting Eqs. (26) and (27) into Eq. (24), the according Hamilton function is

$$\begin{aligned} & H(\wp, {u_i}^*, {d_i}^*, \nabla V^*) \\ =& \wp^T Q \wp + \frac{1}{4\gamma^2} \nabla {V^*}^T \nabla V^* + {u_{ib}}^2 \bar{R} \ln\left(1 - \tanh^2(N^*(\wp))\right) \\ & + \nabla {V^*}^T f(\wp) + \nabla {V^*}^T h(\wp) \end{aligned} \tag{28}$$

where $N^*(\wp) = \left(\dfrac{1}{2u_{ib}} R^{-1} g^T \nabla V^*(\wp)\right)$.

3.2. Network approximation

The neural network structure in this paper is identification, critic, action and disturbance NNs, which are used to approximate the coupling part of MRR subsystem, value function, control law and disturbance law respectively.

Based on the critic NN, the optimal value function is established as

$$V^*(\wp) = {W_{ic}}^T \varphi_{ic}(\wp) + \varepsilon_{ic} \tag{29}$$

where $W_{ic} \in R^{n_1}$ is the weight matrix, n_1 is the number of hidden neurons, φ_{ic} is the activation function, ε_{ic} is the approximation error.

The differential form of the above equation is

$$\nabla V^*(\wp) = \nabla {\varphi_{ic}}^T(\wp) W_{ic} + \nabla \varepsilon_{ic} \tag{30}$$

where $\nabla \varphi_{ic}$ and $\nabla \varepsilon_{ic}$ are the differential form of φ_{ic} and ε_{ic}.

The estimation form of Eq. (29) is

$$\hat{V}(\wp) = \hat{W}_{ic}^T \varphi_{ic}(\wp) \tag{31}$$

where $\hat{W}_{ic}$ is the estimation form of W_{ic}.

The derivative of $\hat{V}(\wp)$ is

$$\nabla \hat{V}(\wp) = \nabla {\varphi_{ic}}^T(\wp) \hat{W}_{ic}. \tag{32}$$

The approximate Hamilton function is

$$\begin{aligned} & H(\wp, u_i, d_i, \hat{W}_{ic}) \\ =& \wp^T Q \wp - \gamma^2 d_i^T d_i + P_u(u_i) + \hat{W}_{ic}^T \nabla \varphi_{ic}(\wp)[f(\wp) \\ & + g(\wp) u_i + d_i(\wp) + h(\wp)] \\ =& e_c. \end{aligned} \tag{33}$$

The update rule of critic NN is

$$\dot{\hat{W}}_{ic} = -l_c \kappa_1 ({\kappa_1}^T \hat{W}_{ic} + \psi(\wp, d_i, u_i)) \tag{34}$$

where $l_c > 0$ is the learning rate, $\kappa_1 = \dfrac{\kappa}{\kappa^T \kappa + 1}$, $\kappa = \nabla \varphi_{ic}(\wp)[f(\wp) + g(\wp) u_i + d_i(\wp) + h(\wp)]$,

$\psi(\wp, u_i, d_i) = \wp^T Q \wp + P_u(u_i) - \gamma^2 d_i^T d_i$.

Similar with above process, the optimal control law is approximated as

$$u_i^* = -u_{ib} \tanh\left(\frac{1}{2u_{ib}} R^{-1} g^T \left(\nabla {\varphi_{ic}}^T(\wp) W_{ic} + \nabla \varepsilon_{ic}\right)\right). \tag{35}$$

The control strategy is estimated as

$$\hat{u}_i = \hat{W}_{ia}^T \varphi_{ia}(\wp) \tag{36}$$

where $W_{ia} \in R^{n_2}$ is the unknown action NN weight matrix, n_2 is the number of hidden neurons, $\hat{W}_{ia}$ is the estimation expression of W_{ia} and φ_{ia} denotes the activation function.

The error generated in the continuous updating process is

$$e_a = \hat{W}_{ia}^T \varphi_{ia}(\wp) + u_{ib} \tanh\left(\frac{1}{2u_{ib}} R^{-1} g^T \left(\nabla {\varphi_{ic}}^T(\wp) W_{ic} + \nabla \varepsilon_{ic}\right)\right). \tag{37}$$

The update law of action NN is

$$\dot{\hat{W}}_{ia} = - l_a \varphi_{ia}(\wp) e_a^T \tag{38}$$

where $l_a > 0$ is the learning rate.

At last, we design the disturbance network.

The optimal disturbance law can be derived through disturbance NN:

$$d_i^* = \frac{1}{2\gamma^2} \left(\nabla {\varphi_{ic}}^T(\wp) W_{ic} + \nabla \varepsilon_{ic}\right). \tag{39}$$

Then, the disturbance strategy is approximated as

$$\hat{d}_i = \hat{W}_{id}^T \varphi_{id}(\wp) \tag{40}$$

where $W_{id} \in R^{n_3}$ is the weight matrix, n_3 is the number of hidden neurons in disturbance NN, $\hat{W}_{id}$ is the estimation value of W_{id} and φ_{id} represents the activation function.

The error caused by disturbance NN for disturbance law estimation is

$$e_d = \hat{W}_{id}^T \varphi_{id}(\wp) - \frac{1}{2\gamma^2}\left(\nabla {\varphi_{ic}}^T(\wp) W_{ic} + \nabla \varepsilon_{ic}\right). \tag{41}$$

The update rule of the disturbance NN is

$$\dot{\hat{W}}_{id} = -l_d \varphi_{id}(\wp) e_d^T \tag{42}$$

where $l_d > 0$ is the learning rate.

Next, we make some assumptions about the weights and activation functions of NNs.

Assumption 3. The weights and activation functions are assumed to have upper bounds, expressed as $\|\hat{W}_{ic}\| \le W_{cm}$, $\|\hat{W}_{ia}\| \le W_{am}$, $\|\hat{W}_{id}\| \le W_{dm}$, $\|\varphi_{ic}\| \le \varphi_{cm}$, $\|\nabla\varphi_{ic}\| \le \varphi_{cb}$, $\|\varphi_{ia}\| \le \varphi_{am}$, $\|\nabla\varphi_{ia}\| \le \varphi_{ab}$, $\|\varphi_{id}\| \le \varphi_{dm}$ and $\|\nabla\varphi_{id}\| \le \varphi_{db}$.

4. Dynamic event-triggered policy iteration

This section introduces the fundamental principle of DETC, which will apply to the iterative process of control strategy to deal with OTCP of MRR. The triggering condition is designed based on the system stability and the dynamic threshold is achieved through the ET condition. After that, the Zeno phenomenon is analyzed.

4.1. Event-triggered control

The main idea of ETC is that the control policy is updated only when the system sampling error exceeds the threshold that can be designed according to the actual situation.

We set $\{t_j\}_{j=1}^{\infty}$ and $j \in \mathbb{N}$ as the triggering moments, the auxiliary system state is sampled aperiodically, therefore, the state is described as

$$\wp = \begin{cases} \wp, & t \in [t_j, t_{j+1}) \\ \wp_j, & t = t_j \end{cases} \tag{43}$$

where $\wp$ is the current state, $\wp_j$ is the sampling state.

The sampling error is defined as the difference between current state and sampling state:

$$e_j = \wp_j - \wp. \tag{44}$$

The event-triggered system can be written as

$$\dot{\wp} = f(\wp) + g(\wp)u_i(e_j + \wp) + d(e_j + \wp) + h(\wp). \tag{45}$$

The triggering condition, also regarded as the event generator, is described as

$$\|e_j\| > e_T \tag{46}$$

where e_T is the threshold.

Based on the basic principle of ETC, when the sampling error exceeds the threshold value, the system is triggered and the controller updates the control policy. At other times, the controller maintains the control strategy of the previous moment.

At the sampling times, the control pair becomes

$$\check{u}_i^* = -u_{ib}\tanh\left(\frac{1}{2u_{ib}} R^{-1} g^T(\wp_j) \nabla V^*(\wp_j)\right) \tag{47}$$

$$\check{d}_i^* = \frac{1}{2\gamma^2}\nabla V^*(\wp_j). \tag{48}$$

The event-triggered Hamilton function is

$$\begin{aligned} & H(\wp, \check{u}_i^*, {d_i}^*, \nabla V^*) \\ =& \wp^T Q \wp + \nabla {V^*}^T(\wp) f(\wp) + \nabla {V^*}^T(\wp) h(\wp) \\ & + \frac{1}{2\gamma^2}\nabla {V^*}^T(\wp_j)\nabla V^*(\wp) - \frac{1}{4\gamma^2}\nabla {V^*}^T(\wp_j)\nabla V^*(\wp_j) \\ & - \nabla {V^*}^T(\wp) g(\wp) u_{ib}\tanh(N^*(\wp_j)) \\ & + \nabla {V^*}^T(\wp_j) g(\wp_j) u_{ib}\tanh(N^*(\wp_j)) \\ & + {u_{ib}}^2 \bar{R}\ln\left(1 - \tanh^2(N^*(\wp_j))\right) \\ =& 0. \end{aligned} \tag{49}$$

Assumption 4. N^* is Lipschitz continuous on a compact set Ω according to the sampling error and has

$$\|N^*(\wp) - N^*(\wp_j)\| \le \ell_n \|e_j\| \tag{50}$$

where $\ell_n > 0$.

4.2. Dynamic event-triggered condition

Theorem 2. *In view of MRR system Eq. (1), corresponding subsystem Eq. (4), sampling error Eq. (44), triggering condition Eq. (46), control policy pair Eq. (47), Eq. (48) and Assumption 3, the asymptotic stability of the closed-system is ensured with the threshold as*

$$e_T = \sqrt{\frac{(1-\beta)^2 \lambda_{\min}(Q)\|\wp_j\|^2 + P_u(\check{u}_i) - \frac{1}{4\gamma^2}\|W_{cm}\|^2\|\varphi_{cb}\|^2}{{u_{ib}}^2 {\ell_n}^2 \lambda_{\max}(R) - (1 - \frac{1}{\beta})\lambda_{\max}(Q)}} \tag{51}$$

where $0 < \beta < 1$.

Proof. The optimal value function is chosen as the Lyapunov function, and the derivative form is

$$\dot{L} = \nabla {V^*}^T(\wp)[f(\wp) + g(\wp)\check{u}_i^* + \check{d}_i^* + h(\wp)]. \tag{52}$$

In terms of Eq. (28), we have

$$\begin{aligned} & \nabla {V^*}^T(\wp) f(\wp) + \nabla {V^*}^T(\wp) h(\wp) \\ =& -\wp^T Q \wp - \frac{1}{4\gamma^2}\nabla {V^*}^T(\wp)\nabla V^*(\wp) \\ & - {u_{ib}}^2 \bar{R}\ln\left(1 - \tanh^2(N^*(\wp))\right). \end{aligned} \tag{53}$$

In the light of Eqs. (49) and (53), $\nabla {V^*}^T g \check{u}_i^*$ can be written as

$$\begin{aligned} & \nabla {V^*}^T g \check{u}_i^* \\ =& \frac{1}{4\gamma^2}\nabla {V^*}^T(\wp)\nabla V^*(\wp) + {u_{ib}}^2 \bar{R}\ln\left(1 - \tanh^2(N^*(\wp))\right) \\ & - \frac{1}{2\gamma^2}\nabla {V^*}^T(\wp_j)\nabla V^*(\wp) + \frac{1}{4\gamma^2}\nabla {V^*}^T(\wp_j)\nabla V^*(\wp_j) \\ & - \nabla {V^*}^T(\wp_j) g(\wp_j) u_{ib}\tanh(N^*(\wp_j)) \\ & - {u_{ib}}^2 \bar{R}\ln\left(1 - \tanh^2(N^*(\wp_j))\right). \end{aligned} \tag{54}$$

According to Eq. (23), it implies that

$$\begin{aligned} & {u_{ib}}^2 \bar{R}\ln\left(1 - \tanh^2(N^*(\wp))\right) - {u_{ib}}^2 \bar{R}\ln\left(1 - \tanh^2(N^*(\wp_j))\right) \\ =& \int_{u_i^*}^{\check{u}_i^*} 2u_{ib} {N^*}^T(\wp) R d\nu - u_{ib}\nabla {V^*}^T(\wp) g(\wp)\tanh(N^*(\wp)) - P_u(\check{u}_i^*) \\ & + u_{ib}\nabla {V^*}^T(\wp_j) g(\wp_j)\tanh(N^*(\wp_j)) \\ & + \int_0^{\check{u}_i^*} 2u_{ib}\tanh^{-1}\left(\frac{\nu}{u_{ib}}\right)^T R d\nu. \end{aligned} \tag{55}$$

Based on Eq. (48), what can be obtained is

$$\nabla V^{*T}(\wp)\check{d}_i^* = \frac{1}{2\gamma^2}\nabla V^T(\wp)\nabla V^*(\wp_j). \tag{56}$$

Together with Eqs. (54)–(56), Eq. (52) becomes

$$\begin{aligned}\dot{L} =& -\wp^T Q\wp + \frac{1}{4\gamma^2}\nabla V^{*T}(\wp_j)\nabla V^*(\wp_j) - P_u(\check{u}_i^*)\\ &+ \int_{u_i^*}^{\check{u}_i^*} 2u_{ib}\left(N^{*T}(\wp) + \tanh^{-1}\left(\frac{\nu}{u_{ib}}\right)^T\right)Rd\nu.\end{aligned} \tag{57}$$

Applying the Young's inequality, the first term of the above equation can be written as

$$\begin{aligned}\wp^T Q\wp =& (\wp_j - e_j)^T Q(\wp_j - e_j)\\ &\geq (1-\beta)\wp_j^T Q\wp_j + (1-\frac{1}{\beta})e_j{}^T Qe_j.\end{aligned} \tag{58}$$

Letting $\nu = -u_{ib}\tanh(\vartheta)$, the last term of Eq. (57) can be deduced as

$$\begin{aligned}&\int_{u_i^*}^{\check{u}_i^*} 2u_{ib}\left(N^{*T}(\wp) + \tanh^{-1}\left(\frac{\nu}{u_{ib}}\right)^T\right)Rd\nu\\ \leq& \int_{N^*(\wp)}^{N^*(\wp_j)} 2u_{ib}{}^2(\vartheta - N^*(\wp))^T Rd\vartheta\\ =& u_{ib}{}^2(N^*(\wp) - N^*(\wp_j))^T R(N^*(\wp) - N^*(\wp_j))\\ \leq& u_{ib}{}^2\ell_n{}^2\|R\|\|e_j\|^2.\end{aligned} \tag{59}$$

After inserting Eq. (59) into Eq. (57), the derivative of Lyapunov function satisfies

$$\begin{aligned}\dot{L} \leq& -\wp^T Q\wp + \frac{1}{4\gamma^2}\nabla V^{*T}(\wp_j)\nabla V^*(\wp_j) - P_u(\check{u}_i) + u_{ib}{}^2\ell_n{}^2\|R\|\|e_j\|^2\\ \leq& -(1-\beta)\lambda_{\min}(Q)\|\wp_j\|^2 - (1-\frac{1}{\beta})\lambda_{\max}(Q)\|e_j\|^2\\ &+ \frac{1}{4\gamma^2}\|\nabla V^*(\wp_j)\|^2 - P_u(\check{u}_i) + u_{ib}{}^2\ell_n{}^2\|R\|\|e_j\|^2\\ \leq& [u_{ib}{}^2\ell_n{}^2\lambda_{\max}(R) - (1-\frac{1}{\beta})\lambda_{\max}(Q)]\|e_j\|^2\\ &- (1-\beta)\lambda_{\min}(Q)\|\wp_j\|^2 + \frac{1}{4\gamma^2}\|W_{cm}\|^2\|\varphi_{cb}\|^2 - P_u(\check{u}_i).\end{aligned} \tag{60}$$

Note that with the triggering condition Eq. (46) and the threshold Eq. (51), if $\wp$ satisfies $\wp \neq 0$, then $\dot{L}$ has $\dot{L} < \beta(\beta - 1)\lambda_{\min}(Q)\|\wp_j\|^2 < 0$. Thus, the proof of the system asymptotic stability is finished.

Under the premise of system control effect, in order to further reduce the sampling frequency, an internal dynamic variable is added into the threshold Eq. (51) to obtain a larger triggering interval time.

The dynamic equation including the dynamic variable ϱ is described as

$$\begin{aligned}\dot{\varrho}(t) =& -K(\varrho(t)) - \|e_j\| + e_T\\ \varrho(0) =& \varrho_0 > 0\end{aligned} \tag{61}$$

where K is $K-$class function, ϱ_0 is a parameter to be designed.

Remark 2. If $K(\cdot)$ is $K-$class function, it has the following characteristics: the function is continuous and strictly increasing in $\mathbb{R}$; If $\alpha \to +\infty$, $K(\cdot)$ will satisfy: $K(0) = 0$ and $K(\alpha) \to +\infty$.

The dynamic triggering condition is defined as

$$\varrho(t) + \hbar_d\{-\|e_j\| + e_T\} < 0 \tag{62}$$

where $\hbar_d$ is the devised parameter. It is obviously that Eq. (46) is the special case of Eq. (62).

Similar to the proof of Theorem 2, if the Lyapunov function is chosen as $D(t) = L(t) + \varrho(t)$, we can get $\dot{D}(t) = \dot{L}(t) + \dot{\varrho}(t) < 0$, hence, the asymptotic stability of the system can be guaranteed under the designed dynamic event-triggered mechanism [29].

Remark 3. The weights of three NNs are only updated when the system is triggered, and the control strategy pair output at the last triggered moment is maintained when the system is not triggered.

4.3. Zeno phenomenon

Zeno phenomenon refers to the fact that the sampling time interval of the system has no lower bound, that is, the system is triggered for an infinite number of times and the control policy pair is updated all the time, which is likely to bring communication congestion to the system.

The following theorem will discuss the sampling interval of the system has a lower bound.

Theorem 3. *The sampling interval of the system has the lower bound as*

$$\tau_j \geq \frac{1}{w_{fm}}\ln\left(\frac{w_{fm\|e_j\|}}{l_e} + 1\right) \tag{63}$$

where $\tau_j = t_{j+1} - t_j$ is the sampling interval.

Proof. On the basis of the definition of the event-triggered error in Eq. (44), the derivative form is

$$\|\dot{e}_j\| = \| - \dot{\wp}\|. \tag{64}$$

Based on Eq. (6), Assumptions 1 and 3, $\|\dot{e}_j\|$ becomes

$$\begin{aligned}&\|\dot{e}_j\|\\ \leq& w_{fm}\|e_j\| + \| w_{fm}^T\phi_{fm} + w_{gm}^T\phi_{gm}\hat{W}_{ia}^T\varphi_{ia}(\wp_j)\\ &+ w_{hm}^T\phi_{hm} + \hat{W}_{id}^T\varphi_{id}(\wp_j) + \varepsilon_m \|\\ \leq& w_{fm}\|e_j\| + \| w_{fm}{}^T\phi_{fm} + w_{gm}{}^T\phi_{gm}W_{am}^T\varphi_{am}\\ &+ w_{hm}{}^T\phi_{hm} + W_{dm}^T\varphi_{dm} + \varepsilon_m \|\\ \leq& w_{fm}\|e_j\| + l_e\end{aligned} \tag{65}$$

where $l_e = \|w_{fm}{}^T\phi_{fm} + w_{gm}{}^T\phi_{gm}W_{am}^T\varphi_{am} + w_{hm}{}^T\phi_{hm} + W_{dm}^T\varphi_{dm} + \varepsilon_m\|$.

The above equation is induced further as

$$\begin{aligned}\|e_j\| \leq& \int_{t_j}^{t} l_e e^{w_{fm}(t-s)}ds\\ =& \frac{l_e}{w_{fm}}\left(e^{w_{fm}(t-t_j)} - 1\right)\end{aligned} \tag{66}$$

thus, the sampling interval meets

$$\tau_j \geq \frac{1}{w_{fm}}\ln\left(\frac{w_{fm\|e_j\|}}{l_e} + 1\right) > 0 \tag{67}$$

Up till now, the analysis of sampling interval is finished.

4.4. Policy iteration process

Compared with the value iteration algorithm, PI approach has been proved to converge faster [30], so this paper adopts PI to obtain the optimal control strategy pair. In order to facilitate comparison, ETC and DETC are successively applied into the iterative process of control policy sequences, and these two triggering mechanisms can receive the optimal solution with the relatively little iterative cost.

The two algorithms designed in this paper are described in Algorithm 1 chart. It is important to note that if ETC is in operation, Eq. (46) is used to determine whether the event occurs, and if DETC is in motion, Eq. (62) is the event generator.

Algorithm 1 The (dynamic) event-triggered policy iteration

```
Initialization:
Define the initial sampling time $t_j = 0$;
Assign an initial value to the initial state $\wp(0)$;
Initialize the optimal control policy pair $u_i(0)$, $d_i(0)$;
Initialize the neural networks;
Main loop:
Compute the event-triggered error $e_j$ Eq. (44);
Compute the threshold Eq. (51);
if event occurs by Eq. (46) (Eq. (62)) then
    Set the current state as the sampled state;
    Update the control policy pair Eqs. (47) and (48);
else
    Keep values of $u_i(\wp_j)$ and $d_i(\wp_j)$;
Update the system state Eq. (45).
return
```

5. Simulation analysis

The simulation experiment in this section uses the same MRR model to compare TTC, ETC and DETC. The effectiveness of the algorithm in this paper is verified by analyzing the experimental results. Simulation A applies ETC in PI process, whereas simulation B obtains the optimal control pair through DETC. Both two experiments include the comparison between TTC and ETC.

This section takes a 2-DOF MRR dynamic model in [31] as the simulation object. The state of MRR system will track the ideal trajectory under the control of the proposed algorithm, and the tracking task of MRR is that x_1 follows the given trajectory $0.3\sin(t) + 0.1\cos(2t)$ and x_2 follows the given trajectory $0.2\cos(3t) - 0.3\sin(2t)$. In order to ensure that the comparison experiment has reference significance, the parameters of the two experiments are basically consistent except the threshold. Some specific parameters are as follows: MRR system state is set as $x = [x_1, x_2, x_3, x_4]^T = [p_1, p_2, \dot{p}_1, \dot{p}_2]^T$, and the initial state value is $x = [1, 1, 0, 0]^T$, the control input is $u = [u_1, u_2]^T$ and the boundary of input is $u_{ib} = 10$, the initial input value is set as $u = [-5.3, -5]^T$, and the sampling interval is 0.01 s.

5.1. Simulation A

Simulation A applies ADP based on ETC to obtain the optimal control law so as to make the MRR complete the tracking task. The analysis of experimental results is carried out around control torque, system state and sampling times.

Fig. 2 describes the control torque of MRR system. It can be seen from the figure that two components of the control input vector are within the boundary, which indicates that the utility function designed for the constrained system is effective. In addition, the lines of control strategies are not smooth but rather like stair shaped, which means that the control input law is not constantly updated.

Under the action of the control strategy in Fig. 2, the four state components of MRR system including position and velocity can respectively complete the tracking task for the ideal trajectory in Figs. 3 and 4. Taking these three figures together, it is not difficult to find that although the control strategy is not updated in real time, the system can still be controlled to complete the given tracking mission, which means that PI algorithm combined with ETC can achieve optimal performance indicators of MRR system.

In order to visualize the characteristics of ETC, Figs. 5 and 6 respectively describe the sampling tracking situation of the MRR for the given trajectory and velocity when the system is triggered. Compared with Figs. 3 and 4, the four state components in Figs. 5 and 6 update only at sampling moments. This trigger scheme can certainly reduce more data transfers, but at the expense of performance metrics. Therefore, it can be seen from Figs. 5 and

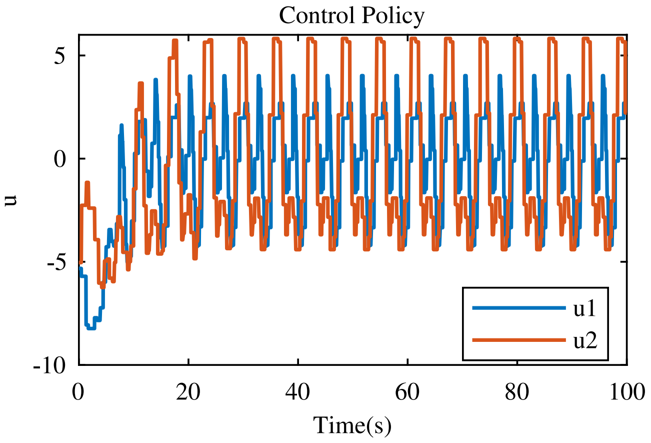

Fig. 2. Control input with ET.

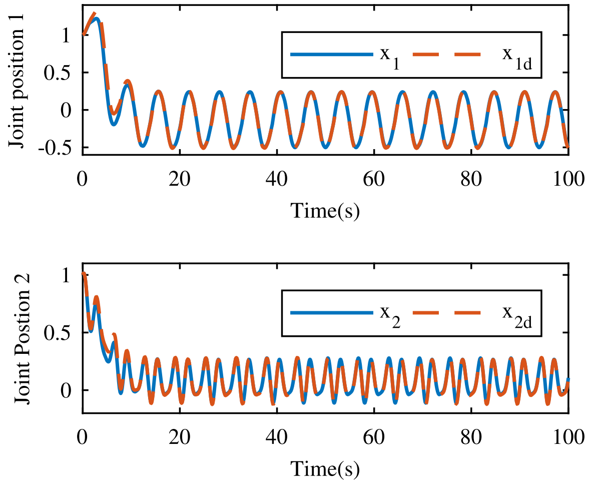

Fig. 3. Trajectory tracking performance with ET.

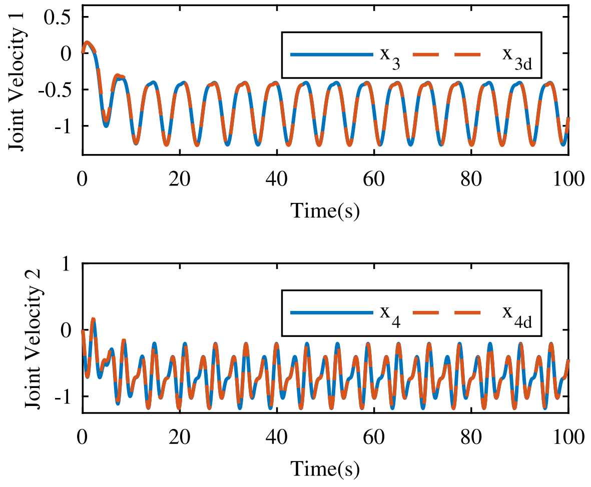

Fig. 4. Velocity tracking performance with ET.

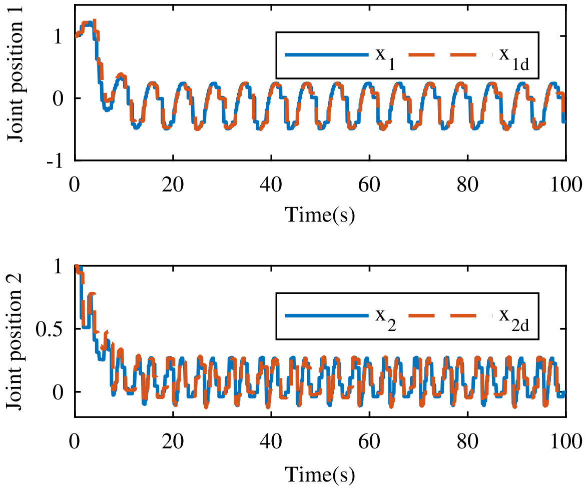

Fig. 5. Sampling trajectory tracking performance with ET.

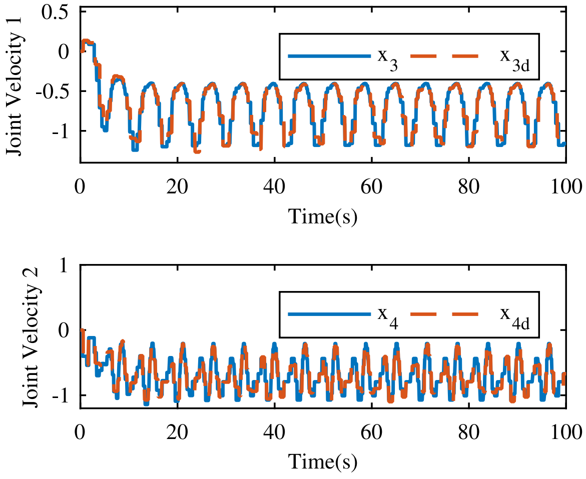

Fig. 6. Sampling velocity tracking performance with ET.

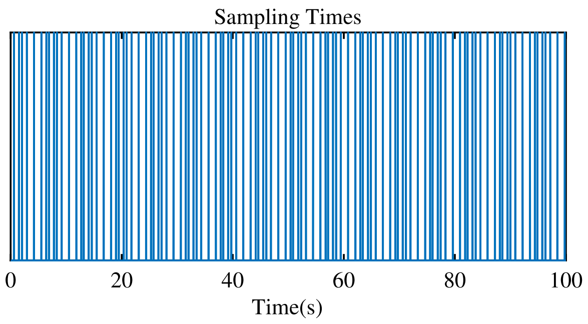

Fig. 7. Sampling times with ET.

6 that although the given tracking task can be completed, the control effect is not as good as Figs. 3 and 4.

The system sampling times within ETC is shown in Fig. 7. The control policy is counted every time when it is updated, and the blank areas represent those moments when the system is not triggered. The system sampling times, that is, the control policy update times, under the two triggering mechanisms are as follows: among the 10 000 iterations, the control policy is updated for 10 000 times under TTC and 222 times under ETC.

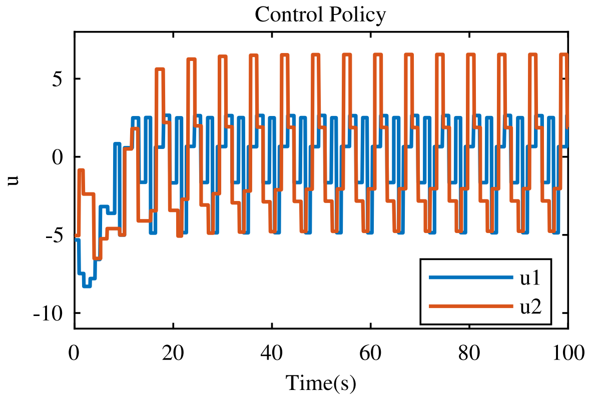

Fig. 8. Control input with DET.

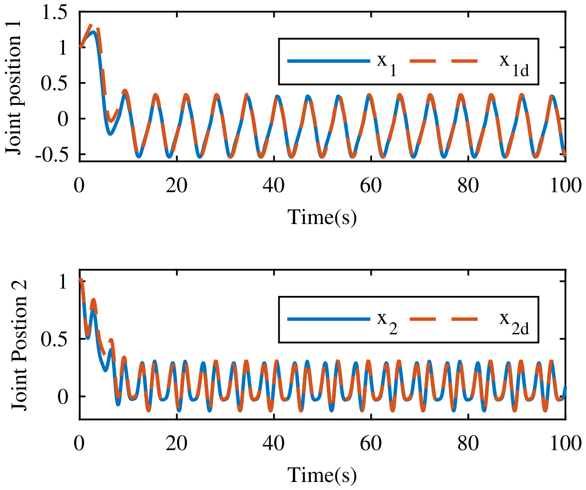

Fig. 9. Trajectory tracking performance with DET.

5.2. Simulation B

Simulation B applies DETC to deal with OTCP for MRR system through ADP. The experimental figures including control strategy, MRR system trace and trigger time statistics are as follows. Some parameters related to DETC are $K(\varrho(t)) = 0.02\varrho$ and $\hbar_d = 1$.

Fig. 8 shows the updating process of two control policy components under DETC. Compared with the control policy under ETC in Fig. 2, Fig. 8 presents a more obvious ladder shape, indicating that the moments when the control policy is not updated has increased. According to Figs. 2 and 8, the triggering interval of DETC is longer than of ETC.

The tracking performance of the four state components of the same MRR system is shown in Figs. 9 and 10. It can be clearly seen from the figures that under the effect of the control sequence in Fig. 8, the MRR system can complete the tracking task of the given position and velocity trajectories, which implies that the system can still achieve OTC even if the update interval of the control strategy becomes longer and the number of updates decreases under DETC. In other words, the dynamic event-triggered condition designed in this paper can make the system extend the sampling interval of the state and reduce the number of system triggers without producing a great sacrifice to the control performance of system.

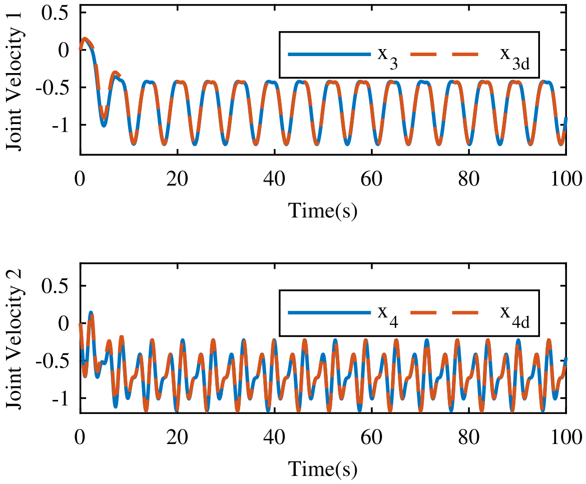

Fig. 10. Velocity tracking performance with DET.

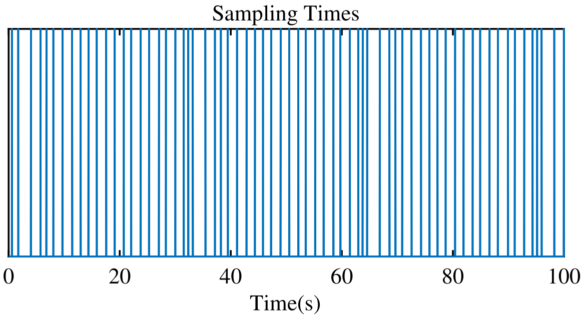

Fig. 11. Sampling times with DET.

The update times of control policy under DETC is given in Fig. 11. Combining Figs. 7 and 11 together, it is obvious that the sampling times in Fig. 11 are more sparse than in Fig. 7, which means that the sampling times in Fig. 11 are less. After statistics, the specific data of the three triggering mechanisms are as follows: the same MRR system is sampled for 10 000 times under TTC, triggered for 222 times under ETC, and triggered for 134 times under DETC. The statistical data comparison reveals that the update times of control policy in ETC are far less than that in TTC, and the control policy in DETC has the least triggering times. In addition, Figs. 7 and 11 verify the content of Theorem 3 that the two triggering conditions designed in this paper can effectively avoid the Zeno phenomenon.

Based on the above comparison of the three triggering mechanisms, it can be concluded that on the premise that the same MRR system completes the tracking task of the same given trajectory, the comparison of data transmission amount of the three mechanisms is as follows: DETC is the best, ETC is the second, and TTC has the most communication times. So far, the validity of the proposed algorithm has been manifested by the above experiments.

6. Conclusion

Aiming at OTCP of MRR, this paper proposes the PI algorithm based on DETC. The dynamic model of MRR system is established by Newton–Euler iterative method, and the identification of coupling parts between subsystems is achieved by NNs. Based on Lyapunov theory, the closed-loop MRR system can maintain asymptotic stability within two triggering conditions. The two control torques obtained by PI guide the system to meet satisfactory tracking performance. The feasibility and advantages of the algorithm in this paper are verified through a series of comparative experiments about the three schemes on OTCP of the two-degree-of-freedom MRR system. What can be obtained from simulation results is that compared with TTC, ETC can effectively reduce the number of control policy updates while satisfying the control performance, and DETC can decrease the number of trigger times more than other two mechanisms in the process of completing the preset assignment. In general, the algorithm proposed in this paper can make the MRR system successfully complete the tracking mission and realize the optimal energy consumption in the environment with disturbance.

CRediT authorship contribution statement

Ruizhuo Song: Conceptualization, Methodology, Software. **Lu Liu:** Writing – original draft, Writing – review & editing. **Zhen Xu:** Supervision, Editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61873300, 61722312, in part by the Fundamental Research Funds for the Central Universities, China under Grant FRF-MP-20-11, and in part by Interdisciplinary Research Project for Young Teachers of USTB (Fundamental Research Funds for the Central Universities), China under Grant FRF-IDRY-20-030.

References

[1] M. Biglarbegian, W.W. Melek, J.M. Mendel, Design of novel interval type-2 fuzzy controllers for modular and reconfigurable robots: theory and experiments, IEEE Trans. Ind. Electron. 58 (4) (2011) 1371–1384, http://dx.doi.org/10.1109/TIE.2010.2049718.

[2] C. Jiang, Z. Ni, Y. Guo, H. He, Learning human–robot interaction for robot-assisted pedestrian flow optimization, IEEE Trans. Syst. Man Cybern. Syst. 49 (4) (2019) 797–813, http://dx.doi.org/10.1109/TSMC.2017.2725300.

[3] P.K. Patchaikani, L. Behera, G. Prasad, A single network adaptive critic-based redundancy resolution scheme for robot manipulators, IEEE Trans. Ind. Electron. 59 (8) (2012) 3241–3253, http://dx.doi.org/10.1109/TIE.2011.2143372.

[4] C. Li, X. Gu, H. Ren, A cable-driven flexible robotic grasper with lego-like modular and reconfigurable joints, IEEE/ASME Trans. Mechatronics 22 (6) (2017) 2757–2767, http://dx.doi.org/10.1109/TMECH.2017.2765081.

[5] R. Barhemat, S. Mahjoubi, V.C. Li, Y. Bao, Lego-inspired reconfigurable modular blocks for automated construction of engineering structures, Autom. Constr. 139 (2022) 104323, http://dx.doi.org/10.1016/j.autcon.2022.104323.

[6] J. Yuan, G. Liu, B. Wu, Power efficiency estimation-based health monitoring and fault detection of modular and reconfigurable robot, IEEE Trans. Ind. Electron. 58 (10) (2011) 4880–4887, http://dx.doi.org/10.1109/TIE.2011.2116753.

[7] B. Ma, Y. Li, T. An, B. Dong, Compensator-critic structure-based neuro-optimal control of modular robot manipulators with uncertain environmental contacts using non-zero-sum games, Knowl.-Based Syst. 224 (2021) 107100, http://dx.doi.org/10.1016/j.knosys.2021.107100.

[8] B. Dong, T. An, X. Zhu, Y. Li, K. Liu, Zero-sum game-based neuro-optimal control of modular robot manipulators with uncertain disturbance using critic only policy iteration, Neurocomputing 450 (2021) 183–196, http://dx.doi.org/10.1016/j.neucom.2021.04.032.

[9] H. Xia, F. Xia, B. Zhao, Y. Huang, Approximate optimal sliding mode tracking control for modular reconfigurable robots based on critic-only structure, IFAC-PapersOnLine 53 (5) (2020) 554–559, http://dx.doi.org/10.1016/j.ifacol.2021.04.222.

[10] R. Song, L. Liu, Event-triggered constrained robust control for partly-unknown nonlinear systems via ADP, Neurocomputing 404 (2020) 294–303, http://dx.doi.org/10.1016/j.neucom.2020.05.012.

[11] Y. Fang, H. Xu, Q. Liu, D.T. Pham, Evolutionary optimization using epsilon method for resource-constrained multi-robotic disassembly line balancing, J. Manuf. Syst. 56 (2020) 392–413, http://dx.doi.org/10.1016/j.jmsy.2020.06.006.

[12] K. Zhang, R. Su, H. Zhang, Y. Tian, Adaptive resilient event-triggered control design of autonomous vehicles with an iterative single critic learning framework, IEEE Trans. Neural Netw. Learn. Syst. 32 (12) (2021) 5502–5511, http://dx.doi.org/10.1109/TNNLS.2021.3053269.

[13] Y. Liang, H. Zhang, J. Duan, S. Sun, Event-triggered reinforcement learning H_∞ control design for constrained-input nonlinear systems subject to actuator failures, Inform. Sci. 543 (2021) 273–295, http://dx.doi.org/10.1016/j.ins.2020.07.055.

[14] Y. Wen, J. Si, A. Brandt, X. Gao, H. Huang, Online reinforcement learning control for the personalization of a robotic knee prosthesis, IEEE Trans. Cybern. 50 (6) (2020) 2346–2356, http://dx.doi.org/10.1109/TCYB.2019.2890974.

[15] J. Yuan, G. Zhang, S. Yu, Z. Chen, Z. Li, Y. Zhang, A multi-timescale smart grid energy management system based on adaptive dynamic programming and multi-NN fusion prediction method, Knowl.-Based Syst. 241 (2022) 108284, http://dx.doi.org/10.1016/j.knosys.2022.108284.

[16] H. Fang, Y. Zhu, S. Dian, G. Xiang, R. Guo, S. Li, Robust tracking control for magnetic wheeled mobile robots using adaptive dynamic programming, ISA Trans. (2021) http://dx.doi.org/10.1016/j.isatra.2021.10.017.

[17] M. Park, K. Kalyanam, S. Darbha, P.P. Khargonekar, M. Pachter, P.R. Chandler, Performance guarantee of an approximate dynamic programming policy for robotic surveillance, IEEE Trans. Autom. Sci. Eng. 13 (2) (2016) 564–578, http://dx.doi.org/10.1109/TASE.2014.2366295.

[18] P.K. Patchaikani, L. Behera, G. Prasad, A single network adaptive critic-based redundancy resolution scheme for robot manipulators, IEEE Trans. Autom. Sci. Eng. 59 (8) (2012) 3241–3253, http://dx.doi.org/10.1109/TIE.2011.2143372.

[19] S. Li, L. Ding, H. Gao, Y. Liu, L. Huang, Z. Deng, ADP-based online tracking control of partially uncertain time-delayed nonlinear system and application to wheeled mobile robots, IEEE Trans. Cybern. 50 (7) (2020) 3182–3194, http://dx.doi.org/10.1109/TCYB.2019.2900326.

[20] Y. Wen, J. Si, A. Brandt, X. Gao, H. Huang, Online reinforcement learning control for the personalization of a robotic knee prosthesis, IEEE Trans. Cybern. 50 (6) (2020) 2346–2356, http://dx.doi.org/10.1109/TCYB.2019.2890974.

[21] S. Jiang, M. Liu, J. Lin, H. Zhong, A prediction-based online soft scheduling algorithm for the real-world steelmaking-continuous casting production, Knowl.-Based Syst. 111 (1) (2016) 159–172, http://dx.doi.org/10.1016/j.knosys.2016.08.010.

[22] Q. Li, L. Xia, R. Song, L. Liu, Output event-triggered tracking synchronization of heterogeneous systems on directed digraph via model-free reinforcement learning, Inform. Sci. 559 (2021) 171–190, http://dx.doi.org/10.1016/j.ins.2021.01.056.

[23] H. Zhang, H. Su, K. Zhang, Y. Luo, Event-triggered adaptive dynamic programming for non-zero-sum games of unknown nonlinear systems via generalized fuzzy hyperbolic models, IEEE Trans. Fuzzy Syst. 27 (11) (2019) 2202–2214, http://dx.doi.org/10.1109/TFUZZ.2019.2896544.

[24] D. Yang, T. Li, X. Xie, H. Zhang, Event-triggered integral sliding-mode control for nonlinear constrained-input systems with disturbances via adaptive dynamic programming, IEEE Trans. Syst. Man Cybern. Syst. 50 (11) (2020) 4086–4096, http://dx.doi.org/10.1109/TSMC.2019.2944404.

[25] M. Ha, D. Wang, D. Liu, Event-triggered constrained control with DHP implementation for nonaffine discrete-time dystems, Inform. Sci. 519 (2020) 110–123, http://dx.doi.org/10.1016/j.ins.2020.01.020.

[26] T. Shi, Y. Zheng, Y. Guan, Input–output finite-time control of Markov jump systems with round-robin protocol: an dynamic event-triggered approach, J. Franklin Inst. (2022) http://dx.doi.org/10.1016/j.jfranklin.2022.03.023.

[27] X. Yi, K. Liu, D.V. Dimarogonas, K.H. Johansson, Dynamic event-triggered and self-triggered control for multi-agent systems, IEEE Trans. Automat. Control 64 (8) (2019) 3300–3307, http://dx.doi.org/10.1109/TAC.2018.2874703.

[28] P.H. Coutinho, M.L. Peixoto, I. Bessa, R.M. Palhares, Dynamic event-triggered gain-scheduling control of discrete-time quasi-LPV systems, Automatica 141 (2022) 110292, http://dx.doi.org/10.1016/j.automatica.2022.110292.

[29] A. Girard, Dynamic triggering mechanisms for event-triggered control, IEEE Trans. Automat. Control 60 (7) (2015) 1992–1997, http://dx.doi.org/10.1109/TAC.2014.2366855.

[30] B. Luo, Y. Yang, D. Liu, Policy iteration Q-learning for data-based two-player zero-sum game of linear discrete-time systems, IEEE Trans. Cybern. 51 (7) (2021) 3630–3640, http://dx.doi.org/10.1109/TCYB.2020.2970969.

[31] B. Zhao, Y. Li, Model-free adaptive dynamic programming based near-optimal decentralized tracking control of reconfigurable manipulators, Int. J. Control Autom. Syst. 16 (2) (2018) 478–490, http://dx.doi.org/10.1007/s12555-016-0711-5.