SCIENCE CHINA
Information Sciences

May 2020, Vol. 63 150210:1–150210:15
https://doi.org/10.1007/s11432-019-2663-y

• RESEARCH PAPER •
Special Focus on Advanced Techniques for Event-Triggered Control and Estimation

Event-triggered receding horizon control via actor-critic design

Lu DONG[1,2,3], Xin YUAN[1,3] & Changyin SUN[1,3*]

[1]*School of Automation, Southeast University, Nanjing* 210096, *China;*
[2]*College of Electronics and Information Engineering, Tongji University, Shanghai* 201804, *China;*
[3]*Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing* 210096, *China*

Received 15 May 2019/Revised 20 July 2019/Accepted 16 September 2019/Published online 30 March 2020

Abstract In this paper, we propose a novel event-triggered near-optimal control for nonlinear continuous-time systems. The receding horizon principle is utilized to improve the system robustness and obtain better dynamic control performance. In the proposed structure, we first decompose the infinite horizon optimal control into a series of finite horizon optimal problems. Then a learning strategy is adopted, in which an actor network is employed to approximate the cost function and an critic network is used to learn the optimal control law in each finite horizon. Furthermore, in order to reduce the computational cost and transmission cost, an event-triggered strategy is applied. We design an adaptive trigger condition, so that the signal transmissions and controller updates are conducted in an aperiodic way. Detailed stability analysis shows that the nonlinear system with the developed event-triggered optimal control policy is asymptotically stable. Simulation results on a single-link robot arm with different noise types have demonstrated the effectiveness of the proposed method.

Keywords event-triggered control, receding horizon control, actor-critic design, optimal control, nonlinear systems, neural networks

Citation Dong L, Yuan X, Sun C Y. Event-triggered receding horizon control via actor-critic design. Sci China Inf Sci, 2020, 63(5): 150210, https://doi.org/10.1007/s11432-019-2663-y

1 Introduction

Research on optimal control has lasted for many decades, and various methods and strategies have emerged [1–3]. Receding horizon control (RHC), also called model predictive control (MPC), is a promising optimal control strategy and has had a tremendous impact on industrial applications [4, 5]. The idea of conventional MPC is to online solve an open-loop optimal control problem in a receding horizon so as to obtain an optimal control sequence and apply the first control in the optimal control sequence to the plant [6]. Based on this type of control methodology, MPC shows significant advantages in terms of performance and robustness and is often used to handle the optimal control with constraints, uncertainties or disturbances.

For linear optimal control problems, MPC has been well developed and established a solid theoretical foundation. Many existing studies have focused on searching the solution to the constrained optimization problem. For the uncertain continuous-time systems, Hu et al. [7] proposed a robust MPC method, in which the constrained optimal control problem is solving through linear matrix inequalities (LMIs).

* Corresponding author (email: cysun@seu.edu.cn)

Evans et al. [8] developed a distributed MPC algorithm for linear systems with persistent bounded disturbances. It is claimed to be able to achieve the coupled constraint satisfaction. Nonlinear MPC (NMPC) shares similar control principles with the details discussed above, except that the system model is nonlinear [9–11]. The difficulty is that, however, determining the optimization problem of the feedback MPC requires to find a solution to the nonlinear Hamilton-Jacobi-Bellman (HJB) equation [6]. In addition, the effectiveness of MPC is sensitive to the model accuracy and the availability of sufficiently fast computational resources [12], which limits the application of MPC.

Bertsekas [13] surveyed some studies on adaptive dynamic programming (ADP) and MPC. ADP, as an advanced formulation of reinforcement learning (RL), is a powerful tool for the nonlinear optimal control problems [14–16]. As discussed in [13], it emphasizes the relationship between MPC and ADP. After that, the relevance between MPC and RL has also been investigated [17, 18]. The above discussion stimulates the research on RL-based MPC for nonlinear systems, and some studies have been developed [19–21]. The main idea of this type of RL-based MPC approach is to conduct an actor-critic scheme to find the optimal solution for the nonlinear systems in each finite horizon. However, it is worth pointing out that the above algorithms are all implemented based on a time-triggered mechanism, and the heavy computation load remains a significant obstacle for the practical use of NMPC. Hence, it is essential to develop an event-triggered control strategy instead of the periodic time-triggered method for reduction in resource utilization. Event-triggered control (ETC) is very popular in recent years, because it can reduce the communication load and improve the computational efficiency while maintaining competitive control performance [22–25]. Recently, some new results on ETC for networked control systems (NCSs) have emerged [26–28], which show that the event-triggered mechanism can effectively solve the problem of limited network bandwidth. Therefore, considering the enormous computational burden in MPC, especially in NMPC, the event-based MPC structure attracts increasing attention [29, 30].

In recent decades, MPC is very popular in practical applications. Ding et al. [31] surveyed related work in industrial cyber-physical systems (CPSs). However, as is known, MPC may cause huge consumption of computer resources. Especially for some large-scale systems, such as power systems and cooperative robot systems, the situation is even worse. Therefore, in this paper, an event-triggered receding horizon actor-critic (RHAC) approach for nonlinear continuous-time systems is proposed. Through decomposing the infinite horizon optimal control problem into a series of finite optimal control problems, an online finite horizon ADP algorithm is adopted in each horizon so as to obtain an optimal control strategy. Furthermore, in order to reduce the computational cost and transmission cost, an event-triggered mechanism is developed for the RHAC, and a novel adaptive trigger threshold is designed. In summary, the contributions of this paper are as follows: (1) We propose a novel RHAC method, in which the optimal control problem for nonlinear systems is solved in a receding horizon manner. (2) An actor-critic scheme is adopted to approximate the optimal control law online in each prediction horizon. (3) An adaptive trigger threshold is developed for the proposed approach to reduce computational resources.

The rest of the paper is organized as follows. Problem formulation is given in Section 2. The ETC strategy is presented in Section 3, including the event-triggered optimization and detailed neural networks implementation. In Section 4, a novel trigger condition is designed, and the stability analysis is presented. The simulation results are shown in Section 5. Finally, the conclusion is drawn in Section 6 and the future work is also discussed.

2 Problem formulation

Consider a nonlinear continuous-time system described by

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the system state with $x(0) = x_0$, and $u(t) \in \mathbb{R}^m$ is the control input. The smooth functions $f(x(t)) \in \mathbb{R}^n$ and $g(x(t)) \in \mathbb{R}^{n\times m}$ are the system dynamics.

Assumption 1. The considered system (1) is controllable and observable. State $x = 0$ is an equilibrium of system (1), and $f + gu$ is Lipschitz continuous on a compact set $\mathcal{Z} \in \mathbb{R}^n$ containing the origin.

In order to seek the optimal control policy for system (1) in the RHC scheme, we consider the following cost function in each finite horizon:

$$J(x(t)) = \Psi(x_{kT}) + \int_t^{t_k+T} r(x(\tau), u(\tau))\mathrm{d}\tau, \quad \forall t \in [t_k, t_k + T), \tag{2}$$

where T is the prediction horizon in RHC. $\{t_k | k = 0, 1, 2, \ldots\}$ is a time sequence, and each finite horizon starts at time t_k and ends at $t_k + T$. Hence, $J(x(t))$ represents the cost function at time t, $t \in [t_k, t_k + T]$. $\Psi(x_{kT}) > 0$ is the terminal cost which penalizes the terminal state x_{kT} with $x_{kT} = x(t_k + T) = x(t_{k+1})$. $r(x, u)$ denotes the utility function with $r(x, u) \geqslant 0$ and is generally defined in quadratic forms. Set $r(x, u) = x^{\mathrm{T}}Qx + u^{\mathrm{T}}u$, where Q is a positive definite symmetric matrix with appropriate dimension.

Definition 1. If a control policy $u(t) \in \Lambda(\mathcal{Z})$ satisfies the following conditions: (i) $u(t)$ is continuous on $\mathcal{Z}$, (ii) $u(0) = 0$, (iii) $u(t)$ stabilizes (1) on $\mathcal{Z}$, and (iv) $\forall x_0 \in \mathcal{Z}$, $J(x_0)$ is finite, then $u(t)$ is said to be admissible, where $\Lambda(\mathcal{Z})$ denotes the set of admissible policies.

It is assumed that $J(x(t))$ is continuously differentiable in each prediction horizon, and for an admissible control policy $u \in \Lambda(\mathcal{Z})$, the corresponding infinitesimal version of (2) is given by

$$r(x(t), u(t)) + (\nabla J(x(t)))^{\mathrm{T}} [f(x(t)) + g(x(t))u(t)] = 0, \quad J(0) = 0, \quad \forall t \in [t_k,\ t_{k+1}), \tag{3}$$

where $\nabla J(x(t)) = \partial J(x(t))/\partial x(t)$ is the partial derivatives of the cost function $J(x(t))$ with respect to $x(t)$. Further, the Hamiltonian function of system (1) is described as

$$\begin{aligned} H(x(t), u(t), \nabla J(x(t))) =& x^{\mathrm{T}}(t)Qx(t) + u^{\mathrm{T}}(t)u(t) + (\nabla J(x(t)))^{\mathrm{T}} [f(x(t)) + g(x(t))u(t)], \\ & \forall t \in [t_k,\ t_{k+1}). \end{aligned} \tag{4}$$

Letting $J^*(x(t))$ denote the optimal cost function, with Bellman's optimality principle, the HJB equation arrives at

$$\min_u H(x, u, \nabla J^*(x)) = 0, \quad \forall t \in [t_k,\ t_{k+1}). \tag{5}$$

Using the stationarity condition, the optimal control input $u^*(t)$ is given by

$$u^*(t) = \arg\min_u H(x, u, \nabla J^*(x)) = -\frac{1}{2} g^{\mathrm{T}}(x(t)) \nabla J^*(x(t)), \quad \forall t \in [t_k,\ t_{k+1}). \tag{6}$$

Substituting (6) into (5), the HJB equation in time-triggered case can be written as

$$\begin{aligned} H(x(t), u^*(t), \nabla J^*(x(t))) =& x^{\mathrm{T}}(t)Qx(t) + \frac{1}{4} (\nabla J^*(x(t)))^{\mathrm{T}} g(x(t)) g^{\mathrm{T}}(x(t)) \nabla J^*(x(t)) \\ & + (\nabla J^*(x(t)))^{\mathrm{T}} \left(f(x(t)) - \frac{1}{2} g(x(t)) g^{\mathrm{T}}(x(t)) \nabla J^*(x(t)) \right) = 0, \quad \forall t \in [t_k,\ t_{k+1}). \end{aligned} \tag{7}$$

At this stage, the optimal control problem for system (1) is transformed into solving (7) in a receding horizon philosophy. However, the RHC strategy may cause a high computational burden. What is worse, owing to the nonlinear nature, it is hardly possible to solve the HJB equation. In this circumstance, the finite horizon optimal control problem will be further formulated in an event-triggered scenario, and the corresponding adaptive control algorithm will be proposed to search the approximate solution to the HJB equation.

3 Event-triggered control strategy

3.1 Event-triggered optimization

The goal of the ETC is to provide a trigger mechanism, in which data transmission and controller updates are conducted in an aperiodic manner. Consider a monotonically increasing sequence $\{\delta_j\}_{j=0}^{\infty}$ with $\delta_0 = 0$

as time instants when the event is triggered. The corresponding trigger condition is expressed as

$$||e_j(t)|| \leqslant e_T, \tag{8}$$

where $||\cdot||$ denotes the Euclidean norm. e_T is the trigger threshold, and $e_j(t)$ is the trigger error defined as

$$e_j(t) = \hat{x}_j - x(t), \tag{9}$$

where $\hat{x}_j$ is the sampled state, and $x(t)$ is the current measured state. Once the trigger condition (8) is violated, that is, an event has occurred, the current state and time instant are stamped as sampled state and trigger instant, respectively. The event-based controller is only updated with the sampled state. Otherwise, the control signal remains unchanged and is maintained by zero-order hold (ZOH) until the next event arrives. With the ZOH, the event-based control input becomes a continuous signal. Therefore, we have

$$x(t) = \hat{x}_j, \quad u(t) = \mu(\hat{x}_j), \quad \forall t \in [\delta_j, \delta_{j+1}). \tag{10}$$

Therefore, the event-based system dynamics is given by

$$\dot{x}(t) = f(x(t)) + g(x(t))\mu(\hat{x}_j). \tag{11}$$

Recalling the time-triggered optimal control policy (6), we can straightforwardly derive the event-triggered optimal control policy as

$$\mu^*(\hat{x}_j) = -\frac{1}{2}g^{\mathrm{T}}(\hat{x}_j)\nabla J^*(\hat{x}_j), \tag{12}$$

and the corresponding event-triggered HJB equation arrives at

$$\begin{aligned} H(x(t), \mu^*(\hat{x}_j), \nabla J^*(x(t))) =& x^{\mathrm{T}}(t)Qx(t) + \frac{1}{4}\left(\nabla J^*(\hat{x}_j)\right)^{\mathrm{T}} g(\hat{x}_j)g^{\mathrm{T}}(\hat{x}_j)\nabla J^*(\hat{x}_j) \\ &+ \left(\nabla J^*(x(t))\right)^{\mathrm{T}} \left(f(x(t)) - \frac{1}{2}g(x(t))g^{\mathrm{T}}(\hat{x}_j)\nabla J^*(\hat{x}_j)\right). \end{aligned} \tag{13}$$

Assumption 2. The optimal control $u^*(t)$ is Lipschitz continuous with respect to the trigger error

$$||u^*(t) - \mu^*(\hat{x}_j)|| \leqslant \kappa||e_j(t)||, \tag{14}$$

where κ is a positive Lipschitz constant.

Different from the time-triggered case (5), Eq. (13) is equal to zero only at the trigger instants $\delta_j,\ j = 0, 1, 2, \ldots$, and therefore define the HJB error during the event-interval as follows:

$$\begin{aligned} e_{\mathrm{HJB},t} &= H(x(t), \mu^*(\hat{x}_j), \nabla J^*(x(t))) - H(x(t), u^*(t), \nabla J^*(x(t))) \\ &= \frac{1}{4}\left(\nabla J^*(\hat{x}_j)\right)^{\mathrm{T}} g(\hat{x}_j)g^{\mathrm{T}}(\hat{x}_j)\nabla J^*(\hat{x}_j) - \frac{1}{2}\left(\nabla J^*(x(t))\right)^{\mathrm{T}} g(x(t))g^{\mathrm{T}}(\hat{x}_j)\nabla J^*(\hat{x}_j) \\ &\quad + \frac{1}{4}\left(\nabla J^*(x(t))\right)^{\mathrm{T}} g(x(t))g^{\mathrm{T}}(x(t))\nabla J^*(x(t)) \\ &= \left(u^*(t) - \mu^*(\hat{x}_j)\right)^{\mathrm{T}} \left(u^*(t) - \mu^*(\hat{x}_j)\right). \end{aligned} \tag{15}$$

Suppose that Assumption 2 holds. Then, the HJB error (15) arrives at

$$||e_{\mathrm{HJB},t}|| \leqslant \kappa^2||e_j(t)||^2. \tag{16}$$

Remark 1. It is a common assumption that the control input is Lipschitz continuous (see, e.g., Lemmon [23], Vamvoudakis [24], Krichman et al. [32]). It is satisfied in many applications, especially when the controller is affine with respect to the gap signal, as Lemmon [23] pointed out.

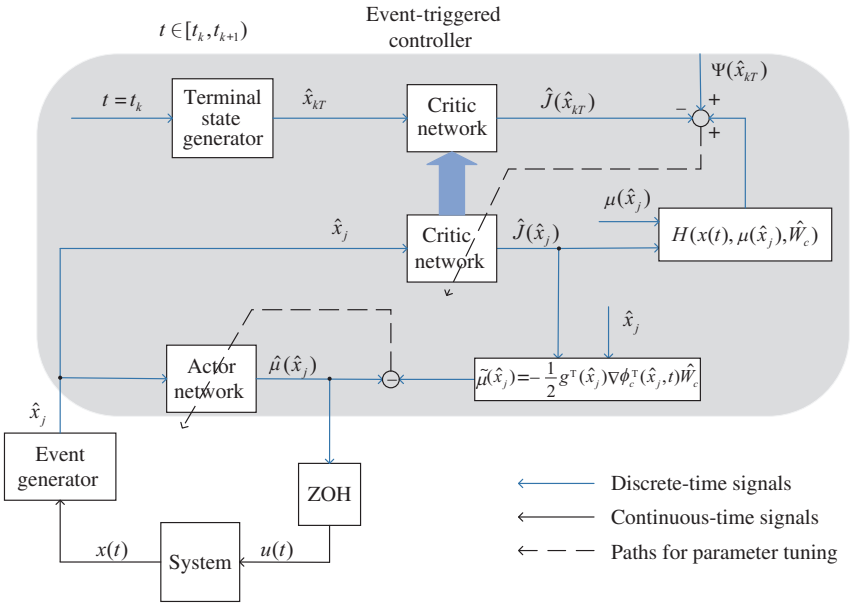

Figure 1 (Color online) The proposed event-triggered RHAC structure.

Remark 2 (Zeno behavior)**.** In real-time control, Zeno behavior is undesirable. In this remark, we will show if Assumptions 1 and 2 hold, Zeno behavior can be avoided in the proposed design.

Define $T_j = \delta_{j+1} - \delta_j$ as the intersample time interval for all $j \in \mathbb{N}$. It can be derived that (see [23] for details)

$$T_j \geqslant \frac{1}{\bar{\kappa} + \bar{\kappa} P}, \tag{17}$$

where P is a positive constant satisfying $1/P \leqslant ||e_j(t)||/||x(t)||$. $\bar{\kappa} > 0$ is the Lipschitz constant for the closed-loop system (11). The right-hand side of inequality (17) is the lower bound of the intersample time. It is obvious that if $\bar{\kappa}$ is bounded, i.e., system (11) is Lipschitz continuous on compacts, the bound of the intersample time is non-zero. Therefore, one can ensure that Zeno behavior is excluded during the learning process. On the contrary, this bound goes to zero if the event-based system fails to be Lipschitz. More detailed derivations can be found in [23, 33].

In this part, the optimization problem for system (1) is further formulated under the event-triggered mechanism. From (16), it is evident that a small HJB error results in frequent updates of the controller and a large value means sacrificing more control performance. Therefore, it is crucial to design a suitable trigger threshold to maintain the tradeoff between resource utilization and stabilization performance.

3.2 Event-based neural network implementation

The framework of the event-triggered RHAC controller is depicted in Figure 1. In each prediction horizon, an estimated terminal state is obtained, and the actor-critic learning method is applied to approach the optimal cost function J^* with the sampled states. Unlike the time-triggered strategy, an event generator is included in the proposed method. The trigger condition (8) is evaluated as soon as a new state is received. If inequality (8) is violated, it is regarded that an event is triggered.

Meanwhile, the current state is sampled and transmitted to the actor-critic frame. Accordingly, the ETC policy is updated, and the trigger error (9) is reset to zero. Furthermore, ZOH is updated with the current value. Otherwise, the controller remains unchanged and the system is updated with the value held by the ZOH. Next, the specific neural network implementation for solving the event-triggered HJB equation (13) is presented.

3.2.1 *Critic network*

Considering the universal approximation property of the neural network, in each finite horizon, the optimal cost function $J^*(x(t))$ is formulated as

$$J^*(x(t)) = W_c^{\mathrm{T}}\phi_c(x(t),t) + \varepsilon_c(x(t)), \tag{18}$$

where W_c is the unknown ideal weights, $\phi_c(x(t),t)$ is the time-varying activation function, and $\varepsilon_c(x(t))$ is the reconstruction error of the critic network. Accordingly, the partial derivative of $J^*(x(t))$ can be written as

$$\nabla J^*(x(t)) = \left(\frac{\partial \phi_c^{\mathrm{T}}(x(t),t)}{\partial x}\right)^{\mathrm{T}} W_c + \frac{\partial \varepsilon_c(x(t))}{\partial x} = \nabla\phi_c^{\mathrm{T}}(x(t),t)W_c + \nabla\varepsilon_c(x(t)). \tag{19}$$

Using the NN approximation, the optimal ETC policy is described as follows:

$$\mu(\hat{x}_j) = -\frac{1}{2}g^{\mathrm{T}}(\hat{x}_j)\left(\nabla\phi_c^{\mathrm{T}}(\hat{x}_j,t)W_c + \nabla\varepsilon_c(\hat{x}_j)\right). \tag{20}$$

Correspondingly, the event-based HJB equation is given by

$$\begin{aligned} H(x,\mu(\hat{x}_j),W_c) &= W_c^{\mathrm{T}}\nabla\phi_c(f(x(t)) + g(x(t))\mu(\hat{x}_j)) + r(x(t),\mu(\hat{x}_j)) \\ &\triangleq \varepsilon_H, \end{aligned} \tag{21}$$

where ε_H is the residual error caused by the neural network approximation.

Assumption 3. ε_c, $\nabla\varepsilon_c$, $\nabla\phi_c$, and ε_H are bounded locally, i.e., $||\varepsilon_c|| \leqslant \varepsilon_{cM}$, $||\nabla\varepsilon_c|| \leqslant \nabla\varepsilon_{cM}$, $||\nabla\phi_c|| \leqslant \nabla\phi_{cM}$, and $||\varepsilon_H|| \leqslant \varepsilon_{HM}$, where ε_{cM}, $\nabla\varepsilon_{cM}$, $\nabla\phi_{cM}$, and ε_{HM} are positive constants.

For the unknown terminal state in each predict horizon $x_{kT},\ k \in \mathbb{N}$, it can be replaced by an estimated value $\hat{x}_{kT}$ [34]. Therefore, using the neural network, the terminal constraint of the cost function in each predict horizon can be described by

$$\begin{aligned} J(\hat{x}_{kT}) &= W_c^{\mathrm{T}}\phi_c(\hat{x}_{kT},kT) + \varepsilon_c(\hat{x}_{kT}) \\ &= \Psi(\hat{x}_{kT}), \qquad k = 1,2,\ldots. \end{aligned} \tag{22}$$

Because W_c is unknown, the actual output of the critic network under the event-triggered mechanism is given by

$$\hat{J}(\hat{x}_j) = \hat{W}_c^{\mathrm{T}}\phi_c(\hat{x}_j,t), \tag{23}$$

and the terminal constraint is

$$\hat{J}(\hat{x}_{kT}) = \hat{W}_c^{\mathrm{T}}\phi_c(\hat{x}_{kT},kT), \tag{24}$$

where $\hat{W}_c$ is the estimation of the target weight matrix, and $\phi_c(\hat{x}_j,t)$ is the event-based activation function. We select $||\phi_c(0,t)|| = 0$ with $||x(0)|| = 0$ in order to ensure $\hat{J}(0) = 0$.

Applying the estimated cost function, the event-based HJB equation (21) can be further rewritten as

$$\begin{aligned} H(x(t),\mu(\hat{x}_j),\hat{W}_c) &= \hat{W}_c^{\mathrm{T}}\nabla\phi_c(f(x(t)) + g(x(t))\mu(\hat{x}_j)) + r(x(t),\mu(\hat{x}_j)) \\ &\triangleq e_{cH}(t). \end{aligned} \tag{25}$$

Combing (22) and (24), the terminal cost error becomes

$$e_{cT}(t) = \Psi(\hat{x}_{kT}) - \hat{W}_c^{\mathrm{T}}\phi_c(\hat{x}_{kT},kT), \tag{26}$$

where $\Psi(\hat{x}_{kT})$ is a function of the terminal state $\hat{x}_{kT}$. In general, it can be defined in quadratic form $\Psi(\hat{x}_{kT}) = \hat{x}_{kT}^{\mathrm{T}}Q_f\hat{x}_{kT}$, where Q_f is a positive definite weighting matrix [19]. Hence, $||\Psi(\hat{x}_{kT})||$ is bounded by a positive constant Ψ_M.

Therefore, the total error for the critic network is represented as

$$e_c(t) = e_{cH}(t) + e_{cT}(t)$$

$$= \hat{W}_c^{\mathrm{T}} \nabla\phi_c(f(x(t)) + g(x(t))\mu(\hat{x}_j)) + r(x(t), \mu(\hat{x}_j)) + \Psi(\hat{x}_{kT}) - \hat{W}_c^{\mathrm{T}}\phi_c(\hat{x}_{kT}, kT)$$
$$= -\tilde{W}_c^{\mathrm{T}} \left(\nabla\phi_c(f(x(t)) + g(x(t))\mu(\hat{x}_j)) - \phi_c(\hat{x}_{kT}, kT)\right) + \varepsilon_H + \varepsilon_c(\hat{x}_{kT}), \tag{27}$$

and the objective of the critic network is to minimize the following squared residual error:

$$E_c(t) = \frac{1}{2} e_c^{\mathrm{T}}(t) e_c(t). \tag{28}$$

Note that, owing to the ETC strategy, the critic network weights only update at the trigger instants $t = \delta_j, j \in \mathbb{N}$ and remain unchanged during the event interval $t \in (\delta_j, \delta_{j+1})$. Therefore, applying the gradient descent, the tuning law for the critic network can be written as

$$\begin{cases} \dot{\hat{W}}_c = 0, \quad t \in (\delta_j, \delta_{j+1}), \\ \hat{W}_c^+ = \hat{W}_c - \Delta\hat{W}_c, \quad t = \delta_j \end{cases} \tag{29}$$

with

$$\Delta\hat{W}_c = \frac{l_c}{(1 + \chi^{\mathrm{T}}\chi)^2} \frac{\partial E_c(t)}{\partial \hat{W}_c} = l_c \frac{\chi}{(1 + \chi^{\mathrm{T}}\chi)^2} e_c^{\mathrm{T}}(t), \tag{30}$$

where $\hat{W}_c^+ = \hat{W}_c(\delta_j^+)$ and δ_j^+ is the time instant just after δ_j. $(1 + \chi^{\mathrm{T}}\chi)^2$ is the normalized term with $\chi = \nabla\phi_c(f(x(t)) + g(x(t))\mu(\hat{x}_j)) - \phi_c(\hat{x}_{kT}, kT)$. l_c denotes the learning rate of the critic network. Then Eq. (29) can be rewritten as

$$\begin{cases} \dot{\hat{W}}_c = 0, \quad t \in (\delta_j, \delta_{j+1}), \\ \hat{W}_c^+ = \hat{W}_c - l_c \frac{\chi}{(1+\chi^{\mathrm{T}}\chi)^2} e_c^{\mathrm{T}}(t), \quad t = \delta_j. \end{cases} \tag{31}$$

Let the critic weight estimation error be $\tilde{W}_c = W_c - \hat{W}_c$. The error dynamics for the critic network weight is expressed as follows:

$$\tilde{W}_c^+ = \tilde{W}_c - l_c \frac{\chi}{(1 + \chi^{\mathrm{T}}\chi)^2} \left(\tilde{W}_c^{\mathrm{T}}\chi - \varepsilon_H - \varepsilon_c(\hat{x}_{kT})\right), \quad t = \delta_j. \tag{32}$$

3.2.2 *Actor network*

The optimal control policy can be obtained by the approximation property of neural network as follows:

$$u^*(t) = W_a^{\mathrm{T}}\phi_a(x(t), t) + \varepsilon_a(x(t)), \tag{33}$$

where W_a is the target weight matrix, $\phi_a(x(t), t)$ is the time-varying activation function, and $\varepsilon_a(x(t))$ is the reconstruction error of the actor network.

Recalling the optimal control (6) defined by the gradient form of the optimal cost function and combining the definition of partial derivative of $J^*(x(t))$ (19), $u^*(t)$ can also be expressed by

$$u^*(t) = -\frac{1}{2} g^{\mathrm{T}}(x) \left(\nabla\phi_c^{\mathrm{T}}(x(t), t) W_c + \nabla\varepsilon_c(x(t))\right). \tag{34}$$

Obviously, we have

$$W_a^{\mathrm{T}}\phi_a(x(t), t) + \varepsilon_a(x(t)) + \frac{1}{2} g^{\mathrm{T}}(x)\nabla\phi_c^{\mathrm{T}}(x(t), t) W_c + \frac{1}{2} g^{\mathrm{T}}(x)\nabla\varepsilon_c(x(t)) = 0. \tag{35}$$

Considering the event-based control strategy, Eq. (33) can be approximated by the output of the actor network as

$$\hat{\mu}(\hat{x}_j) = \hat{W}_a^{\mathrm{T}}\phi_a(\hat{x}_j, t), \tag{36}$$

where $\hat{W}_a$ is the estimation of the target weights and $\phi_a(\hat{x}_j, t)$ is the event-based activation function.

Assumption 4. $\phi_a(x(t), t)$ and $\varepsilon_a(x(t))$ are bounded by $||\phi_a(x(t), t)|| \leqslant \phi_{aM}$ and $||\varepsilon_a(x(t))|| \leqslant \varepsilon_{aM}$, respectively, where ϕ_{aM} and ε_{aM} are positive constants.

Assumption 5. The activation function of the actor network is Lipschitz continuous such that $||\phi_a(x(t),t)-\phi_a(\hat{x}_j,t)|| \leqslant \kappa_\phi||x(t)-\hat{x}_j|| = \kappa_\phi||e_j(t)||$, where κ_ϕ is a positive Lipschitz constant.

With the estimated cost function (23), Eq. (34) can be approximated as

$$\tilde{\mu}(\hat{x}_j) = -\frac{1}{2}g^{\mathrm{T}}(\hat{x}_j)\nabla\phi_c^{\mathrm{T}}(\hat{x}_j,t)\hat{W}_c. \tag{37}$$

However, with the control policies (36) and (37), Eq. (35) is no longer satisfied. Therefore, the error function of the actor network is defined as the gap of (36) and (37), and given by

$$e_a(t) = \hat{W}_a^{\mathrm{T}}\phi_a(\hat{x}_j,t) + \frac{1}{2}g^{\mathrm{T}}(\hat{x}_j)\nabla\phi_c^{\mathrm{T}}(\hat{x}_j,t)\hat{W}_c, \qquad E_a(t) = \frac{1}{2}e_a^{\mathrm{T}}(t)e_a(t). \tag{38}$$

With the proposed event-triggered mechanism, the actor network weights are updated aperiodicly. That is, the control input is updated with the sampled state at the trigger instants $t=\delta_j,\ j\in\mathbb{N}$ and held by the ZOH during the event interval $t\in(\delta_j,\delta_{j+1})$. Applying the gradient descent rule, we have

$$\begin{cases} \dot{\hat{W}}_a = 0, \quad t\in(\delta_j,\delta_{j+1}), \\ \hat{W}_a^+ = \hat{W}_a - l_a\phi_a(\hat{x}_j,t)e_a^{\mathrm{T}}(t), \quad t=\delta_j, \end{cases} \tag{39}$$

where $\hat{W}_a^+$ are the updated weights just after the trigger instants, and l_a is the learning rate of the actor network. With the weight estimation error $\tilde{W}_a = W_a - \hat{W}_a$, at the trigger instants, the error dynamics for the actor network weight is given by

$$\tilde{W}_a^+ = \tilde{W}_a - l_a\phi_a(x(t),t)\left(\tilde{W}_a^{\mathrm{T}}\phi_a(x(t),t) + \frac{1}{2}g^{\mathrm{T}}(x)\nabla\phi_c^{\mathrm{T}}\tilde{W}_c + \varsigma\right)^{\mathrm{T}}, \quad t=\delta_j, \tag{40}$$

with $\varsigma = -(1/2)g^{\mathrm{T}}(x)\nabla\varepsilon_c(x(t)) - \varepsilon_a(x(t))$. It holds that $||\varsigma|| \leqslant \varsigma_M = (1/2)g_M\nabla\varepsilon_{cM} + \varepsilon_{aM}$, where ς_M is a positive constant.

4 Trigger condition design and stability analysis

In this section, an effective trigger condition is designed. With the proposed trigger strategy, the stability of the closed-loop system can be guaranteed. See the following theorem.

Theorem 1. Consider the nonlinear system with event-based control input (11). The weight tuning laws for the critic and actor networks are selected as (29) and (39), respectively. Let Assumptions 1–5 hold. Then the closed-loop system (11) is asymptotically stable, and the weight estimation errors are guaranteed to be uniformly ultimate boundedness (UUB) through the following trigger condition:

$$D(||e_j(t)||) \leqslant \sqrt{\frac{(1-\beta^2)\underline{\lambda}(Q)}{2\kappa_\phi^2||\hat{W}_a||^2}||x(t)||^2 + ||\hat{\mu}(\hat{x}_j)||^2} \triangleq e_T, \tag{41}$$

where $0<\beta<1$. $\underline{\lambda}(Q)$ is the minimal eigenvalue of matrix Q. $D(\cdot)$ denotes the dead-zone operator and is described by

$$D(||e_j(t)||) = \begin{cases} ||e_j(t)||, & ||x(t)|| > b_x, \\ 0, & \text{otherwise}, \end{cases} \tag{42}$$

where b_x denotes the ultimate bound for the state.

Proof. Consider the following Lyapunov function candidate:

$$L(t) = L_1(t) + L_2(t) + L_3(t) + L_4(t) \tag{43}$$

with $L_1(t) = (1/l_c)\mathrm{tr}(\tilde{W}_c^{\mathrm{T}}\tilde{W}_c)$, $L_2(t) = (1/l_a)\mathrm{tr}(\tilde{W}_a^{\mathrm{T}}\tilde{W}_a)$, $L_3(t) = J^*(x(t))$, and $L_4(t) = J^*(\hat{x}_j)$. Because the Lyapunov function is continuous during the event intervals and discrete at the trigger instants, we need consider two cases to proceed the stability analysis.

Case 1. Event is not triggered, i.e., $\delta_j < t < \delta_{j+1},\ j \in \mathbb{N}$.

In this case, the fourth term of Lyapunov function $L_4(t)$ has a zero derivative. In addition, during the event interval, the actor-critic controller is not updated. Then one has

$$\dot{L}_1(t) = 0, \quad \dot{L}_2(t) = 0. \tag{44}$$

Taking the time derivative of the third term in (43) with respect to the sampled-data system (11), we have

$$\begin{aligned}\dot{L}_3(t) =& \dot{J}^*(x) = \left(\nabla J^*(x)\right)^{\mathrm{T}} \dot{x} \\ =& \left(\nabla J^*(x)\right)^{\mathrm{T}} \left(f(x) + g(x)\hat{\mu}(\hat{x}_j)\right) \\ =& \left(\nabla J^*(x)\right)^{\mathrm{T}} \left(f(x) + g(x)u^*(t)\right) - \left(\nabla J^*(x)\right)^{\mathrm{T}} g(x)(u^*(t) - \hat{\mu}(\hat{x}_j)).\end{aligned} \tag{45}$$

According to the time-triggered Hamiltonian function (3) and optimal control policy (6), one has

$$\left(\nabla J^*(x)\right)^{\mathrm{T}} \left(f(x) + g(x)u^*(t)\right) = -x^{\mathrm{T}}(t)Qx(t) - u^{*\mathrm{T}}(t)u^*(t), \tag{46}$$

$$g^{\mathrm{T}}(x)\nabla J^*(x) = -2u^*(t). \tag{47}$$

Therefore, $\dot{L}_3(t)$ becomes

$$\dot{L}_3(t) = -x^{\mathrm{T}}(t)Qx(t) + u^{*\mathrm{T}}(t)u^*(t) - 2u^{*\mathrm{T}}(t)\hat{\mu}(\hat{x}_j), \tag{48}$$

where

$$\begin{aligned}&u^{*\mathrm{T}}(t)u^*(t) - 2u^{*\mathrm{T}}(t)\hat{\mu}(\hat{x}_j) \\ &\quad = ||u^*(x) - \hat{\mu}(\hat{x}_j)||^2 - ||\hat{\mu}(\hat{x}_j)||^2 \\ &\quad = ||W_a^{\mathrm{T}}\phi_a(x(t),t) + \varepsilon_a(x(t)) - \hat{W}_a^{\mathrm{T}}\phi_a(\hat{x}_j,t)||^2 - ||\hat{\mu}(\hat{x}_j)||^2 \\ &\quad \leqslant 2||\hat{W}_a^{\mathrm{T}}(\phi_a(x(t),t) - \phi_a(\hat{x}_j,t))||^2 + 4||\tilde{W}_a^{\mathrm{T}}\phi_a(x(t),t)||^2 + 4||\varepsilon_a(x(t))||^2 - ||\hat{\mu}(\hat{x}_j)||^2 \\ &\quad \leqslant 2||\hat{W}_a||^2\kappa_\phi^2||e_j(t)||^2 + 4\phi_{aM}^2||\tilde{W}_a||^2 + 4\varepsilon_{aM}^2 - ||\hat{\mu}(\hat{x}_j)||^2.\end{aligned} \tag{49}$$

Then, substituting (41) and (49) into (48), we have the time derivative of the Lyapunov function candidate during the event interval as follows:

$$\dot{L}(t) = \dot{L}_3(t) \leqslant -\beta^2\underline{\lambda}(Q)||x(t)||^2 + \xi_{\mathrm{error}}, \tag{50}$$

where $\xi_{\mathrm{error}} = 4\phi_{aM}^2||\tilde{W}_a||^2 + 4\varepsilon_{aM}^2$. It is obvious that during the event interval $\dot{L}(t) < 0$ as long as

$$||x(t)|| \geqslant \sqrt{\frac{\xi_{\mathrm{error}}}{\beta^2\underline{\lambda}(Q)}} \triangleq b_x. \tag{51}$$

Therefore, it indicates that the event-based system is asymptotically stable. Because the neural network weights are constant during the event interval, thus the critic and actor weight estimation errors are UUB.

Case 2. Event is triggered, i.e., $t = \delta_j,\ j \in \mathbb{N}$.

In this situation, the system state is sampled and the control input jumps to a new value. Therefore, the Lyapunov function candidate (43) is discrete at the trigger instants, and the corresponding first difference can be written as

$$\Delta L(t) = \underbrace{\Delta L_1(t) + \Delta L_2(t)}_{\Delta L_N(t)} + \underbrace{\Delta L_3(t) + \Delta L_4(t)}_{\Delta L_V(t)}. \tag{52}$$

Considering the first term in (52), we have

$$\Delta L_1(t) = \frac{1}{l_c}\mathrm{tr}\left(\tilde{W}_c^{+\mathrm{T}}\tilde{W}_c^+\right) - \frac{1}{l_c}\mathrm{tr}\left(\tilde{W}_c^{\mathrm{T}}\tilde{W}_c\right)$$

$$\begin{aligned}
&\leqslant -2\frac{\chi\chi^{\mathrm{T}}}{(1+\chi^{\mathrm{T}}\chi)^2}||\tilde{W}_c||^2+2\mathrm{tr}\left(\frac{\chi(\varepsilon_H+\varepsilon_c(\hat{x}_{kT}))}{(1+\chi^{\mathrm{T}}\chi)^2}\tilde{W}_c^{\mathrm{T}}\right)\\
&\quad+2l_c\frac{\chi\chi^{\mathrm{T}}}{(1+\chi^{\mathrm{T}}\chi)^2}||\tilde{W}_c||^2+l_c(\varepsilon_{HM}^2+\varepsilon_{cM}^2)\\
&\leqslant -(1-2l_c)\frac{\chi\chi^{\mathrm{T}}}{(1+\chi^{\mathrm{T}}\chi)^2}||\tilde{W}_c||^2+(2+l_c)(\varepsilon_{HM}^2+\varepsilon_{cM}^2)\\
&\leqslant -(1-2l_c)\underline{\lambda}(\chi\chi^{\mathrm{T}})||\tilde{W}_c||^2+(2+l_c)(\varepsilon_{HM}^2+\varepsilon_{cM}^2),
\end{aligned}\tag{53}$$

where $\underline{\lambda}(\chi\chi^{\mathrm{T}})$ is the minimal eigenvalue of matrix $\chi\chi^{\mathrm{T}}$.

Then, along with the actor network weight estimation error dynamics (40), the second term in (52) becomes

$$\begin{aligned}
\Delta L_2(t)=&\frac{1}{l_a}\mathrm{tr}\left(\tilde{W}_a^{+\mathrm{T}}\tilde{W}_a^{+}\right)-\frac{1}{l_a}\mathrm{tr}\left(\tilde{W}_a^{\mathrm{T}}\tilde{W}_a\right)\\
=&-2\mathrm{tr}\left(\tilde{W}_a^{\mathrm{T}}\phi_a(x(t),t)\left(\tilde{W}_a^{\mathrm{T}}\phi_a(x(t),t)+\frac{1}{2}g^{\mathrm{T}}(x)\nabla\phi_c^{\mathrm{T}}\tilde{W}_c+\varsigma\right)^{\mathrm{T}}\right)\\
&+l_a\left\|\phi_a(x(t),t)\left(\tilde{W}_a^{\mathrm{T}}\phi_a(x(t),t)+\frac{1}{2}g^{\mathrm{T}}(x)\nabla\phi_c^{\mathrm{T}}\tilde{W}_c+\varsigma\right)^{\mathrm{T}}\right\|^2\\
\leqslant&-\left(\frac{1}{2}\phi_{am}^2-3l_a\phi_{aM}^4\right)||\tilde{W}_a||^2+\left(1+3l_a\phi_{aM}^2\right)\varsigma_M^2\\
&+\frac{1}{2}g_M^2\nabla\phi_{cM}^2\left(1+\frac{3}{2}l_a\phi_{aM}^2\right)||\tilde{W}_c||^2,
\end{aligned}\tag{54}$$

where $0<\phi_{am}\leqslant||\phi_a(x(t),t)||\leqslant\phi_{aM}$ is ensured by the persistence of excitation (PE) condition. Combining (53) and (54), $\Delta L_N(t)$ arrives at

$$\begin{aligned}
\Delta L_N(t)=&\Delta L_1(t)+\Delta L_2(t)\\
\leqslant&-\left((1-2l_c)\underline{\lambda}(\chi\chi^{\mathrm{T}})-\frac{1}{2}g_M^2\nabla\phi_{cM}^2\left(1+\frac{3}{2}l_a\phi_{aM}^2\right)\right)||\tilde{W}_c||^2\\
&-\left(\frac{1}{2}\phi_{am}^2-3l_a\phi_{aM}^4\right)||\tilde{W}_a||^2+\varepsilon_{\mathrm{total}},
\end{aligned}\tag{55}$$

where $\varepsilon_{\mathrm{total}}=\left(1+3l_a\phi_{aM}^2\right)\varsigma_M^2+(2+l_c)(\varepsilon_{HM}^2+\varepsilon_{cM}^2)$. Hence, for $\forall t=\delta_j$, $\Delta L_N(t)<0$ as long as

$$||\tilde{W}_c||>\sqrt{\varepsilon_{\mathrm{total}}\Big/\left((1-2l_c)\underline{\lambda}(\chi\chi^{\mathrm{T}})-\frac{1}{2}g_M^2\nabla\phi_{cM}^2\left(1+\frac{3}{2}l_a\phi_{aM}^2\right)\right)}\triangleq b_{\tilde{W}_c}$$

or

$$||\tilde{W}_a||>\sqrt{\varepsilon_{\mathrm{total}}\Big/\left(\frac{1}{2}\phi_{am}^2-3l_a\phi_{aM}^4\right)}\triangleq b_{\tilde{W}_a},$$

where $b_{\tilde{W}_c}$ and $b_{\tilde{W}_a}$ are ultimate bounds for $\tilde{W}_c$ and $\tilde{W}_a$, respectively. This indicates the critic and actor weight estimation errors are UUB at the trigger instants.

Next, $\Delta L_V(t)$ can be expressed by

$$\Delta L_V(t)=J^*(x^+)-J^*(x(t))+J^*(\hat{x}_{j+1})-J^*(\hat{x}_j).\tag{56}$$

Because $\dot{L}(t)<0$ for all $t\in[\delta_j,\ \delta_{j+1})$ and the cost function is continuous, there exists $J^*(x^+)\leqslant J^*(x(t))$. Then, we have

$$\Delta L_V(t)<J^*(\hat{x}_{j+1})-J^*(\hat{x}_j)\leqslant-k\left(||e_{j+1}(\delta_j)||\right),\tag{57}$$

where $e_{j+1}(\delta_j)=\hat{x}_{j+1}-\hat{x}_j$, and $k(\cdot)$ is a class-$\mathcal{K}$ function [35].

Combining (55) and (57), one has $\Delta L(t) < 0$. Furthermore, we can conclude that the system state is asymptotically stable and the neural network estimation errors are UUB at the trigger instants $t = \delta_j,\ j \in \mathbb{N}$.

In summary, the sampled-date system is asymptotically stable and the neural network estimation errors are UUB as long as the trigger condition is defined as (41). The proof is completed.

Remark 3. The dead-zone operator can help avoid unnecessary triggers owing to the reconstruction error of the neural network. Through the dead-zone operator, controller update is stopped if the system state is within the ultimate bound.

Remark 4. The defined Lyapunov function candidate (43) consists of four parts. $L_1(t)$ and $L_2(t)$ are for the critic error dynamics given by (32) and the actor error dynamics given by (40), respectively. Eq. (44) implies that the neural network weight estimation errors are constant during the event interval, and Eq. (55) implies that the neural network weight estimation errors remain bounded at the trigger instants. Then we can conclude that the neural network weight estimation errors are UUB. The last two terms are the optimal value function with respect to the system state and sampled state, which means we need to make sure that the system state is UUB, as well as the sampled state.

Remark 5. Theoretical analysis of Theorem 1 shows that the proposed trigger condition can ensure the stability of the closed-loop system. To the best of the authors' knowledge, it is the first time to develop the event-triggered architecture for such an RHAC controller. Compared with [36], an advantage of the proposed trigger threshold is that the designed trigger threshold e_T is a function of the current system state and event-based control input, which makes the designed trigger threshold more adaptive. In comparison with [24], we take use of the dead-zone operator, which can further reduce unnecessary triggers.

5 Simulation results

The proposed event-triggered RHAC approach is tested on a single-link robot arm system [37], and the system model is given by

$$\ddot{\theta}(t) = -\frac{MgH}{G}\sin(\theta(t)) - \frac{D}{G}\dot{\theta}(t) + \frac{1}{G}u(t) \tag{58}$$

with parameters as follows.

$g = 9.81$ m/s^2: the acceleration of gravity.

$H = 0.5$ m: the length of the arm.

$D = 2$ N·m·s/rad: the viscous friction coefficient.

$M = 5$ kg: the mass of the payload.

$G = 5$ kg·m^2: the moment of inertia.

$\theta(t)$: the angle position of the arm.

$u(t)$: the control torque of the arm.

The system state is defined as $x(t) = [x_1(t)\ x_2(t)]^{\mathrm{T}} = [\theta(t)\ \dot{\theta}(t)]^{\mathrm{T}}$ and $u(t)$ is the control input. Then the dynamic function (58) can be rewritten as

$$\begin{cases} \dot{x}_1(t) = x_2(t); \\ \dot{x}_2(t) = -4.905\sin(x_1(t)) - 0.4x_2(t) + 0.2u(t). \end{cases} \tag{59}$$

The initial state is set to $x_0 = [0.5\ \ -0.5]^{\mathrm{T}}$. Q in the utility function is selected as $Q = I_{2\times 2}$, where I is the identity matrix, and Q_f in the terminal constraint is set to $Q_f = 0.5Q$. The critic network weights are initialized to zero, while the actor network weights are initialized with random values within $[-1\ 1]$. The time-varying activation functions for both actor and critic networks are polynomial form and set to $\phi_c(x(t), t) = \phi_a(x(t), t) = [x_1, x_1\exp(-\tau), x_2, x_2\exp(-\tau), x_1^2, x_2^2, x_1x_2\tau]$, where $\tau = (kT - t)/T$ is the normalized time-to-go for $t \in [t_k, t_{k+1}), k \in \mathbb{N}$. The sample interval for system discretization is selected as $\Delta t = 0.02$ s.

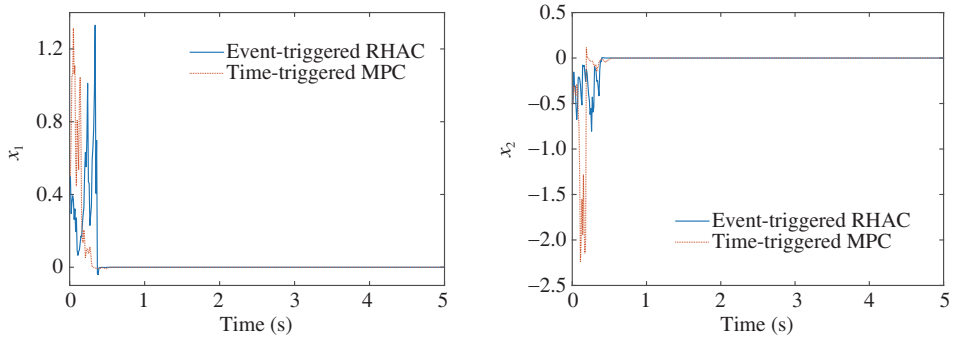

Figure 2 (Color online) Comparison of the state trajectories for the single-link robot arm system with Gaussian sensor noise.

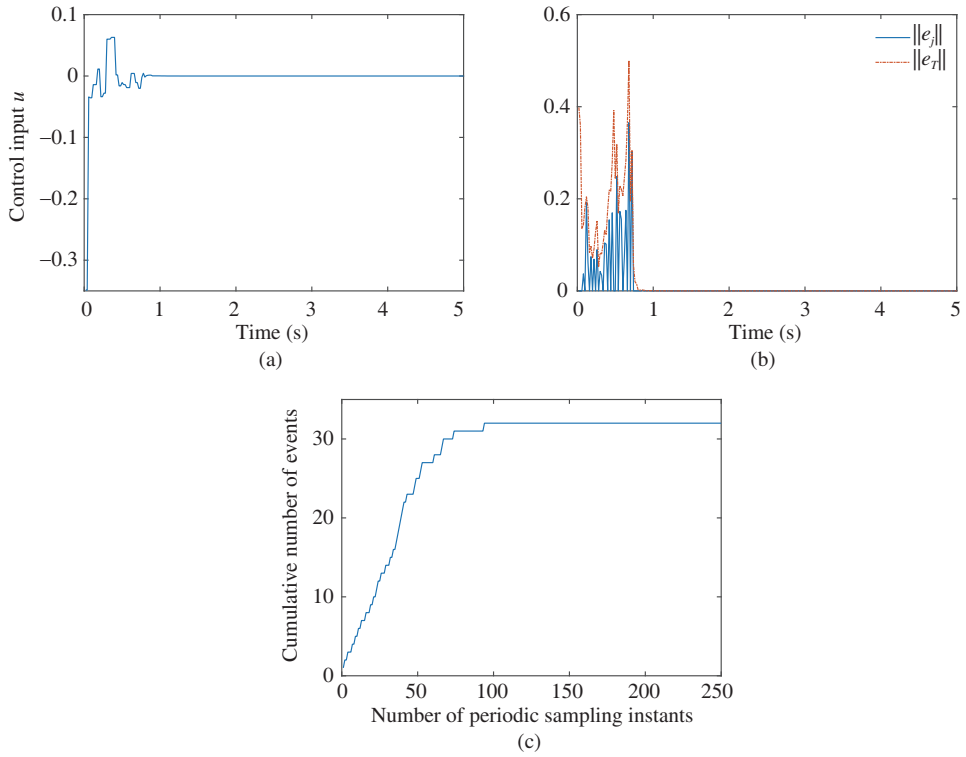

Figure 3 (Color online) Event-triggered RHAC algorithm for the single-link robot arm system with Gaussian sensor noise. (a) Evolution of the control input; (b) comparison of trigger error $||e_j||$ and trigger threshold $||e_T||$; (c) cumulative number of events.

In order to demonstrate the robustness and effectiveness of the proposed event-triggered architecture, we add Gaussian noise and uniform noise to sensor and actuator, respectively. In addition, a traditional time-triggered MPC method [38] is also conducted on the single-link robot arm system to make a contrast.

Case 1. Gaussian sensor noise. In this case, a Gaussian sensor noise with zero mean and variance $\sigma^2 = 0.2$ is added to the system. A comparative study is conducted in this case. For the proposed event-triggered RHAC method, parameters in the trigger condition are selected as $\beta = 0.6$ and $\kappa_\phi = 1.5$. The learning rates for the neural networks are $l_c = l_a = 0.1$. The prediction horizon is set as $T = 20$, and the total simulation time is 5 s. For the time-triggered MPC method, we choose $T = 50$.

Figure 2 shows the comparison of the system states with the proposed event-triggered RHAC method and the time-triggered MPC method. We can see that all the curves converge to zero and two methods can obtain competitive results. Simulation results with the proposed event-triggered RHAC method are shown in Figure 3. Figure 3(a) clearly shows that the control input is a piecewise signal. The event-based controller is updated at the trigger instants and keeps unchanged during the event interval. The

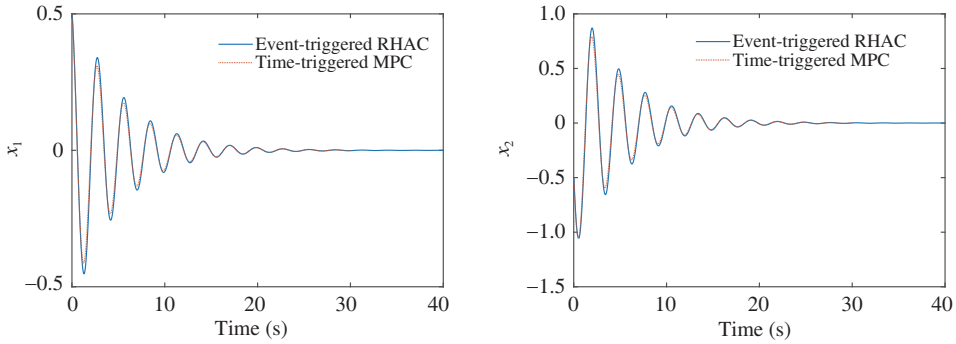

Figure 4 (Color online) Comparison of the state trajectories for the single-link robot arm system with uniform actuator noise.

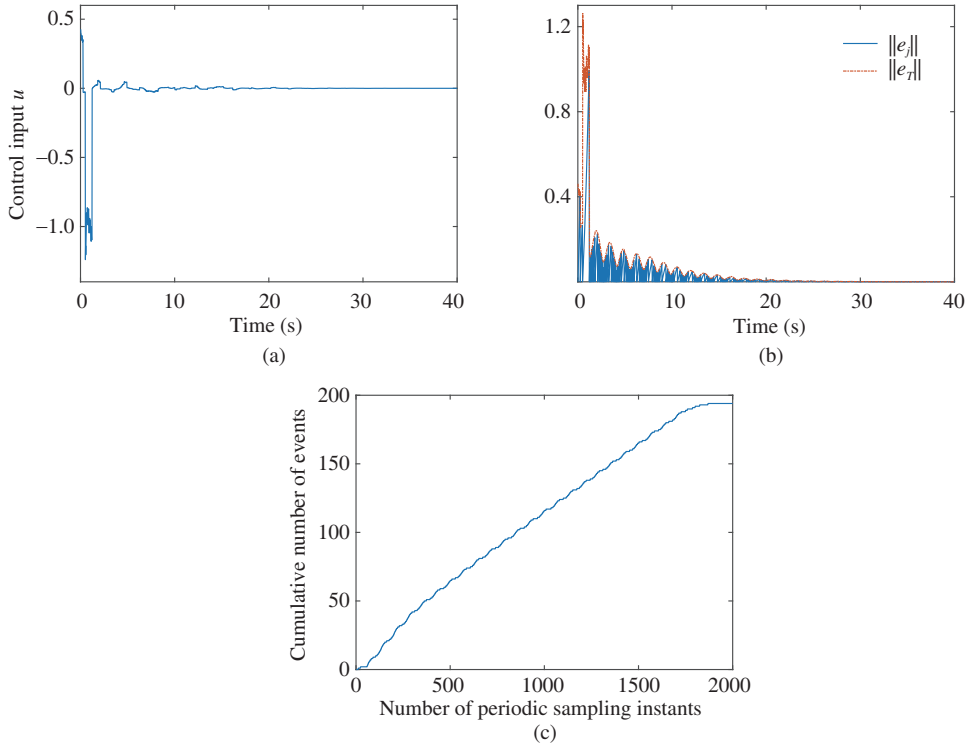

Figure 5 (Color online) Event-triggered RHAC algorithm for the single-link robot arm system with uniform actuator noise. (a) Evolution of the control input; (b) comparison of trigger error $||e_j||$ and trigger threshold $||e_T||$; (c) cumulative number of events.

relationship between the trigger error $||e_j||$ and trigger threshold $||e_T||$ is presented in Figure 3(b). We can see that the trigger threshold converges to zero as the system states and event-triggered control policy converge to zero. In addition, in any event interval, the trigger error starts from zero and continues to increase until it is greater than the trigger threshold. Then e_j is reset to zero immediately. In Figure 3(c), the curve shows the cumulative number of events under the event-triggered mechanism against the total sampling instants. It is obvious that with $\Delta t = 0.02$ s, for the time-triggered MPC, there are 250 sampling instants included during the simulation time of 5 s. While for the proposed event-triggered RHAC controller, it only updates 32 times in total, which indicates that the computational cost has been reduced by about 87%. It is worth pointing out that the cumulative number of events becomes constant after 94 sampling instants owing to the dead-zone operator.

Case 2. Uniform actuator noise. In this case, a 10% uniform noise is added to the control input. For the proposed method, we choose $\beta = 0.6$ and $\kappa_\phi = 1.5$ for the trigger condition. The prediction is $T = 100$ and the total simulation time is 40 s. The initial learning rates for the neural networks are

selected as $l_c = 0.2$ and $l_a = 0.8$. They are decreased by 0.005 every five time steps until they reach 0.005. For the time-triggered MPC method, set $T = 50$.

Figure 4 shows the trajectories of the system states under two different methods. The simulation results are very similar. Figure 5(a) shows the convergence of the control input with uniform noise, which is implemented through $u(\hat{x}_j) = u(\hat{x}_j) \cdot (1 + \text{NoisePercentage} \cdot \rho)$, where ρ is a uniformly distributed random variable. The relationship of trigger error $||e_j||$ and trigger threshold $||e_T||$ is depicted in Figure 5(b). From Figure 5(c), it can be seen that with the event-triggered strategy, the controller updates 194 times while the time-triggered controller uses 2000 samples.

6 Conclusion and future work

In this paper, an event-triggered RHAC approach was proposed for a class of nonlinear continuous-time systems. In order to deal with the disturbances, the RHAC strategy was adopted. First, the infinite horizon optimal control problem was decomposed into a series of finite horizon optimal control problems. Then, the actor-critic algorithm was applied to solve the finite horizon optimal control problem in each prediction horizon. To further reduce the computational cost and transmission cost, an event-triggered methodology was developed, and a novel trigger threshold was designed based on the system states and event-triggered control law to ensure closed-loop stability. The simulation results demonstrated that the proposed method can significantly reduce resource costs without sacrificing control performance.

In practical applications, especially for NCSs, delays and packet losses are inevitable, which can seriously degrade the control performance and even cause system instability. Therefore, event-triggered RHAC design for nonlinear NCSs will be carried out. In addition, at present the length of prediction horizon is selected by empirical value. It is possible to develop learning-based strategy to adaptively determine the length in the future.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 61803085, 61921004, 61931020) and National Key R&D Program of China (Grant No. 2018AAA0101400).

References

1 Song R Z, Xiao W D, Sun C Y. A new self-learning optimal control laws for a class of discrete-time nonlinear systems based on ESN architecture. Sci China Inf Sci, 2014, 57: 068202
2 Li C H, Zhang E C, Jiu L, et al. Optimal control on special Euclidean group via natural gradient algorithm. Sci China Inf Sci, 2016, 59: 112203
3 Wei W N. Maximum principle for optimal control of neutral stochastic functional differential systems. Sci China Math, 2015, 58: 1265–1284
4 Lee J H. Model predictive control: review of the three decades of development. Int J Control Autom Syst, 2011, 9: 415–424
5 Bequette B W. Nonlinear control of chemical processes: a review. Ind Eng Chem Res, 1991, 30: 1391–1413
6 Mayne D Q, Rawlings J B, Rao C V, et al. Constrained model predictive control: stability and optimality. Automatica, 2000, 36: 789–814
7 Hu L S, Huang B, Cao Y Y. Robust digital model predictive control for linear uncertain systems with saturations. IEEE Trans Automat Contr, 2004, 49: 792–796
8 Evans M, Cannon M, Kouvaritakis B. Robust and stochastic linear MPC for systems subject to multiplicative uncertainty. IFAC Proc Vol, 2012, 45: 335–341
9 Fang H, Shang C S, Chen J. An optimization-based shared control framework with applications in multi-robot systems. Sci China Inf Sci, 2018, 61: 014201
10 Zhou L M, Jia L, Wang Y L. A robust integrated model predictive iterative learning control strategy for batch processes. Sci China Inf Sci, 2019, 62: 219202
11 Aguilera R P, Lezana P, Quevedo D E. Switched model predictive control for improved transient and steady-state performance. IEEE Trans Ind Inf, 2015, 11: 968–977
12 Venkat A N, Hiskens I A, Rawlings J B, et al. Distributed MPC strategies with application to power system automatic generation control. IEEE Trans Contr Syst Technol, 2008, 16: 1192–1206
13 Bertsekas D P. Dynamic programming and suboptimal control: a survey from ADP to MPC*. Eur J Control, 2005, 11: 310–334
14 Wei Q L, Liu D R. A novel policy iteration based deterministic Q-learning for discrete-time nonlinear systems. Sci China Inf Sci, 2015, 58: 122203

15 Wang D, Mu C X. Developing nonlinear adaptive optimal regulators through an improved neural learning mechanism. Sci China Inf Sci, 2017, 60: 058201
16 Ding D R, Wang Z D, Han Q L, et al. Neural-network-based output-feedback control under round-robin scheduling protocols. IEEE Trans Cybern, 2019, 49: 2372–2384
17 Ernst D, Glavic M, Capitanescu F, et al. Reinforcement learning versus model predictive control: a comparison on a power system problem. IEEE Trans Syst Man Cybern B, 2009, 39: 517–529
18 Görges D. Relations between model predictive control and reinforcement learning. IFAC-PapersOnLine, 2017, 50: 4920–4928
19 Lian C Q, Xu X, Chen H, et al. Near-optimal tracking control of mobile robots via receding-horizon dual heuristic programming. IEEE Trans Cybern, 2016, 46: 2484–2496
20 Xu X, Chen H, Lian C Q, et al. Learning-based predictive control for discrete-time nonlinear systems with stochastic disturbances. IEEE Trans Neural Netw Learn Syst, 2018, 29: 6202–6213
21 Dong L, Yan J, Yuan X, et al. Functional nonlinear model predictive control based on adaptive dynamic programming. IEEE Trans Cybern, 2019, 49: 4206–4218
22 Anta A, Tabuada P. To sample or not to sample: self-triggered control for nonlinear systems. IEEE Trans Automat Contr, 2010, 55: 2030–2042
23 Lemmon M D. Event-triggered feedback in control, estimation, and optimization. In: Networked Control Systems. London: Springer, 2010. 293–358
24 Vamvoudakis K G. Event-triggered optimal adaptive control algorithm for continuous-time nonlinear systems. IEEE/CAA J Autom Sin, 2014, 1: 282–293
25 Duan G P, Xiao F, Wang L. Hybrid event- and time-triggered control for double-integrator heterogeneous networks. Sci China Inf Sci, 2019, 62: 022203
26 Zhang X M, Han Q L. Event-triggered dynamic output feedback control for networked control systems. IET Control Theor Appl, 2014, 8: 226–234
27 Zhang B L, Han Q L, Zhang X M. Event-triggered H_∞ reliable control for offshore structures in network environments. J Sound Vib, 2016, 368: 1–21
28 Zhang X M, Han Q L. A decentralized event-triggered dissipative control scheme for systems with multiple sensors to sample the system outputs. IEEE Trans Cybern, 2016, 46: 2745–2757
29 Liu C X, Gao J, Li H P, et al. Aperiodic robust model predictive control for constrained continuous-time nonlinear systems: an event-triggered approach. IEEE Trans Cybern, 2018, 48: 1397–1405
30 Li H P, Shi Y. Event-triggered robust model predictive control of continuous-time nonlinear systems. Automatica, 2014, 50: 1507–1513
31 Ding D R, Han Q L, Wang Z D, et al. A survey on model-based distributed control and filtering for industrial cyber-physical systems. IEEE Trans Ind Inf, 2019, 15: 2483–2499
32 Krichman M, Sontag E D, Wang Y. Input-output-to-state stability. SIAM J Control Opt, 2001, 39: 1874–1928
33 Tabuada P. Event-triggered real-time scheduling of stabilizing control tasks. IEEE Trans Automat Contr, 2007, 52: 1680–1685
34 Heydari A, Balakrishnan S N. Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics. IEEE Trans Neural Netw Learn Syst, 2013, 24: 145–157
35 Khalil H K. Nonlinear Systems. New Jersey: Prentice Hall, 2002
36 Dong L, Zhong X N, Sun C Y, et al. Event-triggered adaptive dynamic programming for continuous-time systems with control constraints. IEEE Trans Neural Netw Learn Syst, 2017, 28: 1941–1952
37 Zhong X N, He H B. An event-triggered ADP control approach for continuous-time system with unknown internal states. IEEE Trans Cybern, 2017, 47: 683–694
38 Chen H. Model Predictive Control. Beijing: Science Press, 2013