Tracking Control for Motion Constrained Robotic System via Dynamic Event-Sampled Intelligent Learning Method

Zhinan Peng†, Jiangping Hu‡, Senior Member, IEEE, Hong Cheng†, Senior Member, IEEE, Rui Huang†, Rui Luo†, Pengbo Zhao§, and Bijoy Kumar Ghosh†, Life Fellow, IEEE

Abstract—This paper introduces a new event-sampled intelligent learning method to solve tracking control for motion constrained robotic system. In many applications, motion constraints on joint movements of a robot are needed for its safe operation or to accommodate limitations arising from its mechanical structures. To handle the control design problem, a state-transformation method is employed to guarantee motion constraints of the robot and obtain a corresponding unconstrained tracking error system. Next, the constrained control problem is transformed into a general optimal tracking control problem for the unconstrained error system. Finally, we implement neural networks based learning structures to obtain the optimal solution. An important highlight of our proposed algorithm is that we have integrated a dynamic event-sampled approach to the learning-based controller design, thus reducing the system state sampling times. Using the proposed event-sampled learning control method, the stability of the closed-loop system and the convergence of the weights of neural networks are both established. Lastly, the effectiveness of the proposed intelligent control method is verified via simulation.

I. INTRODUCTION

In the past decade, robots have been widely used in factory and industry for increasingly many different services [1], [2], [3]. Since the component dynamics of a typical robotic system are nonlinear and strongly coupled, a basic problem is to design the required control that allows the robot to perform a complex task following a specific human instruction. Naturally, motion control problems for robots have attracted extensive attention in recent years [4].

To address the robotic motion control problem, many researchers have designed controllers for achieving good control performances [5], [6], [7]. Such methods include, PID control [8], model predictive control [9], neural network (NN) control [10], decentralized fuzzy control [11], etc. Although these controller design methods have made significant progress in the field of robotic control, they also have drawbacks when it comes to practical implementation. For example, PID-based methods rely on manual selection of parameters, which are difficult to optimally choose during various application scenarios. Although NNs have capabilities of adaptation and the ability to adapt to uncertainties, it is unclear how the weights are to be chosen optimally.

In a typical real-life application of robots, due to special requirements arising from some application scenarios or dictated by the limitations of mechanical structure, the joint motion space of robot is always restricted [12], [13]. As a result, it is interesting to address the robot control design problem as a motion constrained control problem. Note that the tracking control problem for an uncertain state-constrained robotic manipulator has been investigated earlier in [14], wherein the results mainly focused on control performance. However, the optimization of the controller’s ability/performance, in energy consumption and cost, for example, has not been considered. In this paper, we put our focus on incorporating intelligent approaches for optimizing the performance of a robotic system with motion constraints.

Recently, learning-based methods have been incorporated with a basic control design framework to solve optimal control problems. We design control systems that can achieve control goals with certain optimal guarantees [15], [16], [17]. Specifically, adaptive dynamic programming (ADP) and reinforcement learning (RL) have shown powerful abilities to deal with optimal control for nonlinear or linear single-agent system [18], [19], [20], [21], [22] and multi-agent systems [23], [24], [25], [26]. More recently, ADP/RL has been applied to control robots as well [27], [28]. The optimal control problems for robot manipulators with state-constraints have been addressed via critic-NN-based ADP method in [29].

Nevertheless, note that the designed ADP-based robotic control depends on a time-triggered scheme. In other words, the system state is continuously sampled, which leads to an abundance of information transmission and increased computational cost. In this paper, we propose to combine ADP and event triggered scheme to address the motion control problem of a robotic system under prescribed motion constraints. A new event-sampled intelligent learning method is proposed for the motion tracking control of robotic systems with motion constraints. The main contributions are summarized as follows: 1) In order to deal with the motion constraints, a state-transformation method is employed. The original motion constrained tracking control problem is
transformed into an optimal control problem for a new unconstrained tracking error system. 2) An event-sampled intelligent learning framework, in which two event-triggered mechanisms with triggering conditions are proposed for computational efficiency, is proposed to compute the optimal solutions in real time. 3) In the implementation of NN-based learning algorithm, a stable term is added to the weight training law, which relaxes the condition on the initial stabilizing controller. It is shown that the tracking error system is asymptotically stable, and the estimated weights of the NN are convergent in the sense of ‘ultimately uniformly bounded’.

II. PRELIMINARIES AND PROBLEM FORMULATION

A. Model Description

We consider a $n$-degree-of-freedom ($n$-DOF) robotic system [6] whose dynamics is modeled as

$$M(q)q + C(q, \dot{q})\dot{q} + G(q) = u,$$  \hspace{1cm} (1)

where $q \in \mathbb{R}^n$, $\dot{q} \in \mathbb{R}^n$ denote the state vector and the velocity vector of joint space, respectively. $M(q) \in \mathbb{R}^{n \times n}$ represents the inertial matrix, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the Coriolis and centrifugal matrix, $G(q) \in \mathbb{R}^n$ denotes the gravity vector, and $u \in \mathbb{R}^n$ is the control input of the system. The reference state is set by vectors $q_d$ and $\dot{q}_d$.

We define the motion tracking errors towards $q_d$ and $\dot{q}_d$ as $e_q = q - q_d$ and $e_{\dot{q}} = \dot{q} - \dot{q}_d$. Let $x = [q^T, \dot{q}_d^T]^T$, $x_h = [q^T, \dot{q}_h^T]^T$ be the state vector and desired state vector, respectively, then the motion tracking error vector is designed as $e = x - x_d = [e_q, e_{\dot{q}}]^T$. The dynamics of the motion tracking error is

$$\dot{e} = r(e) + c(e)u, \hspace{1cm} (2)$$

where $r(e) = \begin{bmatrix} 0 & I \\ -M^{-1}C & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ -M^{-1}G \end{bmatrix}$, $c(e) = \begin{bmatrix} 0 \\ (M^{-1})^T \end{bmatrix} x$.

The goal of this paper is to design a control policy $u$ to make the joint state of the robotic system track the desired trajectory $x_d$. The motion constraints can be described as $x^i < x_i < \bar{x}^i$, $i = 1, 2, ..., 2n$, where $\bar{x}^i$ is the upper bound of the $i$th element of state $x$, and $x^i$ represents the lower bound of the $i$th element of state $x$.

B. Motion Constraints Transformation

The original robotic system with prescribed constraints is transformed into an unconstrained system by implementing a state transformation. Define $x^i = [x^i_1, x^i_2, ..., x^i_{2n}]^T$, and $x_h^i = [\hat{x}^i_1, \hat{x}^i_2, ..., x^i_{2n}]^T$. Let $\lambda^h = x_h - x_d$ and $\lambda^l = x - x_d$. Therefore, the prescribed motion tracking error is obtained as $\lambda^l < e_i < \lambda^h$, where $\lambda^l$ and $\lambda^h$ are the $i$th elements of $\lambda^l$ and $\lambda^h$, $\lambda^l < 0$ and $\lambda^h > 0$ are the lower and upper bounds of the motion tracking error $e$, respectively.

In order to meet the prescribed motion constraints, a barrier function transformation

$$\bar{e}_i = \mathcal{B_i}(e_i, \lambda^l_i, \lambda^h_i) = \text{log}_a \left\{ \frac{\lambda^h_i (\lambda^l_i - e_i)}{\lambda^l_i (\lambda^h_i - e_i)} \right\}, \hspace{1cm} (3)$$

is introduced, where $\mathcal{B}_i(\cdot): \mathbb{R} \rightarrow \mathbb{R}$ is an invertible smooth function and $a > 1$ is a positive constant.

Remark 1: The barrier function $\mathcal{B}_i(\cdot)$ has the following desired properties: (1) $\lim_{e_i \rightarrow \lambda^l_i} \mathcal{B}_i = -\infty$, (2) $\lim_{e_i \rightarrow \lambda^h_i} \mathcal{B}_i = +\infty$, (3) $\mathcal{B}_i(0, \lambda^l_i, \lambda^h_i) = 0$.

According to (3), the inverse barrier function is

$$e_i = \mathcal{B}_i^{-1}(\xi_i, \lambda^l_i, \lambda^h_i) = \frac{\lambda^h_i (\xi_i - a^{-1}\lambda^l_i)}{\lambda^l_i (\xi_i - a^{-1}\lambda^h_i)}, \hspace{1cm} (4)$$

Using the barrier function based state-transformation, a one-to-one mapping relationship can be established between $e_i$ and $\xi_i$. From (4), one can claim that if $\xi_i \in (-\infty, +\infty)$, the original tracking error $e_i$ can be constrained in an interval $(\lambda^l_i, \lambda^h_i)$. Thus, it is concluded that once $\lim_{t \rightarrow +\infty} \xi_i = 0$, then $\lim_{t \rightarrow +\infty} e_i = 0$ satisfying the motion constraints.

Therefore, according to (2), (3), and the fact that $\dot{\xi}_i = \partial J^s(\xi_i)$, the motion tracking control problem of the original robotic system is transformed to a new unconstrained error system related to $\xi_i$, which is derived as

$$\dot{\xi}_i = f(\xi_i) + g(\xi_i)u, \hspace{1cm} (5)$$

where $f(\xi_i) = \partial J^l(\xi_i) = \partial J^l(\xi_i)\mathcal{B}_i^{-1}(\xi_i)$, $g(\xi_i) = \partial J^l(\xi_i)\mathcal{B}_i^{-1}(\xi_i)$, and $\partial J^l(\xi_i) = diag\left\{ \frac{dJ^l}{dx_1}, \frac{dJ^l}{dx_2}, ..., \frac{dJ^l}{dx_{2n}} \right\}$, where $i = 1, 2, ..., 2n$.

It would follow that the motion tracking control problem of the robotic system (1) with motion constraints would be solved by stabilizing the new unconstrained transformation system (5).

C. Problem Description

For the new error system (5), an infinite-horizon cost function is defined as

$$J(\xi_i, u) = \int_0^\infty Q(\xi_i(\sigma), u(\sigma))d\sigma, \hspace{1cm} (6)$$

where $Q(\xi_i, u) = \xi^T A \xi_i + u^T B u$ is the local utility function and $A \in \mathbb{R}^{2n \times 2n}$, $B \in \mathbb{R}^{n \times n}$ are positive-definite symmetric constant matrices.

The goal of this paper is to design an optimal control to solve the motion tracking problem as well as minimizing the cost function (6). The optimal cost function is expressed by

$$J^*(\xi_i) = \min_u \int_0^\infty Q(\xi_i(\sigma), u(\sigma))d\sigma. \hspace{1cm} (7)$$

By differentiating $J(\xi_i, u)$ along the error system (5), one obtains the following Bellman equation (Lyapunov equation)

$$Q(\xi_i, u) + \nabla J^T(\xi_i) (f(\xi_i) + g(\xi_i)u(\xi_i)) = 0, \hspace{0.5cm} (7)$$

where $\nabla J(\xi_i)$ denotes the partial derivative of the cost function $J$ with respect to $\xi_i$, that is, $\nabla J(\xi_i) = \partial J/\partial \xi_i$. Then, the Hamiltonian function is described as

$$H(\xi_i, u(\xi_i), \nabla J^T(\xi_i)) = Q(\xi_i, u) + \nabla J^T(\xi_i) (f(\xi_i) + g(\xi_i)u(\xi_i)). \hspace{1cm} (8)$$

The optimal cost function $J^*(\xi_i)$ satisfies the following Hamilton-Jacobi-Bellman (HJB) equation

$$\min_u [H(\xi_i, u(\xi_i), \nabla J^T(\xi_i))] = 0. \hspace{1cm} (9)$$
It follows that, the optimal control policy is expressed by

$$u^*(\xi) = -\frac{1}{2}B^{-1}g^T(\xi)\nabla J^*(\xi).$$  \hspace{1cm} (10)

### III. Event-Triggered Learning Control Design

In this section, the basic principle of event-sampled mechanism is recalled. Thereafter, an event-sampled learning control is designed for solving an optimal control problem.

#### A. The Basic Formulation of an Event-Triggered Mechanism

The motion tracking error state is sampled at the triggering instant, where a set of triggering instants is \(\{\psi_j\}_{j=0}^\infty\); \(\psi_0 = 0, \psi_j \in \mathbb{R}_+^+, j \in \mathbb{N}_0^+\). The occurrence of the event and the triggering instants are determined by the triggering condition which depends on a triggering error. Using the difference between the sampled state and the true state, the triggering error can be expressed mathematically as follows

$$\delta_j(t) = \hat{E}_j - \xi(t), \quad \psi_j \leq t < \psi_{j+1},$$  \hspace{1cm} (11)

where \(\hat{E}_j = \hat{E}(\psi_j)\) is the sampled error state, \(\psi_j\) is the triggered instant and \(\xi(t)\) is the current state. If the event is triggered, \(\delta_j(t)\) is reset to zero.

The interval between \(\psi_j\) and \(\psi_{j+1}\) is called the triggering interval. When an event is triggered at \(t = \psi_j\), then \(\hat{E}_j\) is sampled and applied to calculate the control policy. A basic event-triggered (Et) control policy is expressed by

$$u^*(\hat{E}_j) = -\frac{1}{2}B^{-1}g^T(\hat{E}_j)\nabla J^*(\hat{E}_j).$$  \hspace{1cm} (12)

#### B. Event-Sampled NN-Learning Control Design

An event-sampled NN-based intelligent learning control framework is now proposed.

1) **Critc NN Design:** First, the cost function can be approximated as follows

$$J^*(\xi) = W_c^T \phi(\xi) + \epsilon(\xi),$$  \hspace{1cm} (13)

$$\nabla J^*(\xi) = \nabla \phi^T(\xi)W_c + \nabla \epsilon(\xi),$$

where \(W_c \in \mathbb{R}^{L \times L}\) is the ideal hidden-to-output weight, \(L\) represents the number of neural nodes in the hidden layer, \(\phi(\cdot) : \mathbb{R}^{2n} \to \mathbb{R}^L\) is the activation function, \(\epsilon(\xi)\) is the reconstruction error, \(\nabla \phi = \partial \phi / \partial x\), and \(\nabla \epsilon = \partial \epsilon / \partial x\).

In fact, the ideal/optimal weight is unknown in advance for the learning process. Thus the estimated weight \(\hat{W}_c\) is used to approach the ideal weight, then the real cost function is estimated by the following critic approximator

$$\hat{J}^*(\xi) = \hat{W}_c^T \phi(\xi), \nabla \hat{J}^*(\xi) = \nabla \phi^T(\xi)\hat{W}_c.$$

The current error \(\hat{e}_B\) of the critic NN can be expressed as

$$\hat{e}_B = H(\xi, u(\xi), \nabla \hat{J}^*(\xi)) - H(\xi, \hat{u}(\xi), \nabla J^*(\xi))$$

$$= \hat{W}_c^T \nabla \phi(\xi)(f(\xi) + g(\xi)\hat{u}^*(\xi)) + Q(\xi, \hat{u}^*(\xi)).$$  \hspace{1cm} (15)

The gradient-descent (GD) method is applied to online adjust the critic weights for minimizing the objective function \(E_c = \frac{1}{2}\|\hat{e}_B\|^2\). The tuning law of the critic NN is described as follows

$$\hat{W}_c = -\alpha_t \frac{1}{(1 + \phi^T \phi)^2} \frac{\partial E_c}{\partial \hat{W}_c} = -\alpha_t \frac{1}{(1 + \phi^T \phi)^2} \phi \hat{e}_B,$$  \hspace{1cm} (16)

where \(\phi = \nabla \phi(\xi)(f(\xi) + g(\xi)\hat{u}^*(\xi)), \quad \hat{e}_B = \hat{W}_c^T \nabla \phi(\xi)(f(\xi) + g(\xi)\hat{u}^*(\xi)) + Q(\xi, \hat{u}^*(\xi)),\) and \(\alpha_t\) represents the learning rate, \((1 + \phi^T \phi)^2\) is utilized for normalization.

**Assumption 1:** [18], [29] Let us consider the error system (5) with cost function (6) and the optimal control policy (10). Let \(\hat{V}(\xi)\) be a radially unbounded and continuously differentiable Lyapunov function candidate satisfying \(\hat{V}(\xi) = \hat{V}^T(\xi)\hat{V}(\xi) = \hat{V}^T(\xi)(f(\xi) + g(\xi)\hat{u}^*(\xi))^2 < 0\). Then, it is concluded that the following inequality is satisfied \(\hat{V}(\xi) = \hat{V}^T(\xi)(f(\xi) + g(\xi)\hat{u}^*(\xi)) = -\hat{V}^T(\xi)Q\hat{V}(\xi) \leq -\sigma_{\text{min}}(\Omega)\|\hat{V}(\xi)\|^2\), where \(\Omega\) is a positive-definite matrix.

**Remark 2:** Note that the above tuning law designed by minimizing \(E_c\) may not ensure the stability of the transformed system (5) in the entire online learning procedure. We would have to ensure that the situation \(\hat{V}(\xi) = \hat{V}^T(\xi)(f(\xi) + g(\xi)\hat{u}^*(\xi)) > 0\) is avoided.

To surmount this difficulty, a new tuning law is designed as follows by adding an auxiliary term for stabilization,

$$\hat{W}_c = \hat{W}_{hc} + \hat{W}_{ac},$$  \hspace{1cm} (17)

where \(\hat{W}_{hc} = -\alpha_t \frac{1}{(1 + \phi^T \phi)^2} \phi \left(\hat{W}_c^T \phi - 0.25\hat{W}_c^T \nabla \phi(\xi)E \nabla \phi^T(\xi)\hat{W}_c \right.$$  

$$+ \epsilon_H \right) - 0.5\alpha_c \chi(\hat{E}, \hat{u}^*) \nabla \phi(\xi)E \nabla \phi^T(\xi),$$  \hspace{1cm} (18)

where \(\epsilon_H = \nabla \hat{V}^T(f(\xi) + g(\xi)\hat{u}^*) + \frac{1}{2} \nabla \hat{V}^T \nabla \hat{V}\).

**Remark 3:** The redesigned tuning law (17), the estimated weight error dynamics of the critic NN is deduced as follows

$$\hat{W}_c = -\alpha_t \frac{1}{(1 + \phi^T \phi)^2} \phi \left(\hat{W}_c^T \phi - 0.25\hat{W}_c^T \nabla \phi(\xi)E \nabla \phi^T(\xi)\hat{W}_c \right.$$  

$$+ \epsilon_H \right) - 0.5\alpha_c \chi(\hat{E}, \hat{u}^*) \nabla \phi(\xi)E \nabla \phi^T(\xi),$$  \hspace{1cm} (18)

where \(\hat{W}_c \in \mathbb{R}^{L^d}\) is the current value of the actor weight and also the estimated value of ideal weight of the critic \(W_c\).

The objective function of the actor NN is defined as \(\alpha_a = -\frac{1}{2}B^{-1}g^T(\xi)\nabla \phi^T(\xi)\hat{w}_a - W_c\). The event-triggered based weight tuning law of the actor NN is redesigned as

$$\hat{w}_a = \hat{w}_a + \psi_j < t < \psi_{j+1} \{ \hat{w}_a + Pr(\alpha_a L(\hat{w}_a - W_c)) \}, \quad t = \psi_{j+1},$$  \hspace{1cm} (20)

in which \(\alpha_a\) is the learning rate, \(L\) is an appropriate positive-definite matrix, \(Pr(\cdot)\) is a projection operator which restricts the actor weight \(\hat{w}_a\) in a compact set indicating that \(\hat{w}_a\) remains bounded [30]. We define \(\hat{w}_a = W_c - \hat{w}_a\) as the actor
weight error. The actor weight error dynamics is expressed by $\dot{\hat{w}_a} = Pr\{-\alpha_3 L(\hat{w}_a - \hat{W}_c)\}$.

IV. EVENT-SAMPLED SCHEME DESIGNS

In this section, two event-sampled schemes are proposed to obtain triggering conditions. The main theoretical results are also given for stability and the NN convergence. Before moving forward, a boundedness assumption [31] and a Lipschitz assumption [32], [33] are commonly imposed.

Assumption 2: Boundedness assumption: The ideal critic weight $|\hat{W}_c| \leq \omega_{cc}$. The activation function and its gradient $|\phi(\xi)| \leq \phi_{max} |\nabla \phi(\xi)| \leq \phi_{max}$. The reconstruction error and its gradient $|\hat{\phi}(\xi)| \leq \epsilon_{R}, |\nabla \hat{\phi}(\xi)| \leq \epsilon_{Rm}$. The Bellman error $|\epsilon_{B}(\xi)| \leq \epsilon_{BM}$.

Assumption 3: $f(\cdot)$ is Lipschitz continuous with $|f(\xi)| \leq L_f |\xi|$. $g(\cdot)$ is Lipschitz continuous by satisfying $|g(\xi) - g(\xi')| \leq L_g |\delta(t)|$ and also bounded by $|g(\xi)| \leq g_M$. $\nabla \phi(\cdot)$ is also Lipschitz continuous with $|\nabla \phi(\xi) - \nabla \phi(\xi')| \leq L_{\phi} |\delta(t)|$.

A. Static Event Triggering Scheme

The static event-sampled scheme (s-ESS) is designed for determining the triggering instants. We assume that $t = \psi_0$ is the first triggering instant, then the subsequent triggering time can be described as follows,

$$
\psi_{j+1} = \inf \{ \tau \in \mathbb{R}^+ | \psi_j + (1 - \theta)\lambda_{\min}(A)(|\xi|^2 - 0.5g_M^2) - \|B^{-1}\|^2(\phi_{ma}^2 + \phi_{ma}^2L_a^2)|\hat{w}_a|^2|\delta_j(t)|^2 < 0 \},
$$

where $\theta \in (0, 1)$ is a conservative design, $\lambda_{\min}(A)$ stands for the minimal eigenvalue of $A$. To describe the triggering rules more compactly, the s-ESS can be expressed by the static triggering condition (STC) form

$$
|\delta_j(t)|^2 \leq \frac{2(1 - \theta)\lambda_{\min}(A)|\xi|^2}{\lambda_{\min}(A) - |\xi|^2} \leq \frac{4}{\lambda_{\max}(A)} \leq \frac{15}{\lambda_{\max}(A)} = \psi_0^3,
$$

where $\psi_0^3$ is the triggering threshold of the s-ESS.

For the s-ESS, once the triggering condition (22) is violated, the event is triggered and a new state is sampled immediately at this triggering instant. Next, the stability of the unconstrained error system is analyzed under the s-ESS based learning control method.

Theorem 1: Let us consider the transformed system (5) together with the defined cost function (6) and assume that the critic and actor are approximated by (14) and (19), respectively. Furthermore assume that the weight tuning laws of the NN are computed by (17) and (20) and the events are triggered through the static triggering condition (22). If the following inequalities

$$
||\hat{W}_c|| > \rho_w, ||\nabla J(\xi)|| > \rho_j, ||\hat{w}_a|| > \sqrt{(\sigma_1 + \sigma_2)/(2rf - 1)}
$$

hold, then the unconstrained error system is asymptotically stable and weight errors are ultimately uniformly bounded (UUB).

Proof: The Lyapunov candidate function is selected as

$$
\mathcal{L}_i(t) = \mathcal{L}_{i1}(t) + \mathcal{L}_{i2}(t) + \mathcal{L}_{i3}(t) + \mathcal{L}_{i4}(t),
$$

where $\mathcal{L}_{i1}(t) = J^T(\hat{\xi}_i) \hat{J}(\hat{\xi}_i) + \mathcal{L}_{i2}(t) = 0.5\tilde{W}_c^T \tilde{W}_c + \alpha_3 \hat{V}(\tilde{\xi}_i), \mathcal{L}_{i4}(t) = \tilde{w}_a^T \alpha_3 \tilde{w}_a$.

Case i: When the event is not triggered, $\mathcal{L}_{i2}, \mathcal{L}_{i4}$ are constants. Therefore, we have $\mathcal{L}_{i2}(0) = 0, \mathcal{L}_{i4}(0) = 0$.

Then, we can obtain

$$
\dot{\mathcal{L}}_{i1}(t) = -(\nabla J^T(\xi) + g(\xi) u(\xi))\nabla J(\xi) - (\nabla J^T(\xi) + g(\xi) u(\xi) - \hat{u}(\xi)) \dot{\hat{u}}(\xi).
$$

For (25), let $h = u(\xi) - \hat{u}(\xi)$ and it can be rewritten as

$$
\dot{\mathcal{L}}_{i1}(t) = -(\nabla J^T(\xi) + g(\xi) u(\xi) - \hat{u}(\xi)) \dot{\hat{u}}(\xi).
$$

With the help of the Young’s inequality and Cauchy-Schwarz inequality, we have

$$
J^*(\xi) \leq -\xi^T Q_\xi^2 u^T B^T u \leq -\xi^T Q_\xi^2 u^T B^T u + \frac{1}{2} |\xi|^2 + \frac{1}{2} |\xi|^2 |\hat{u}(\xi)|^2.
$$

(27)

For the first term of (28), it further evolves into

$$
|\xi|^2 \leq |\xi|^2 |\hat{u}(\xi)|^2 + \frac{1}{2} |\xi|^2 + \frac{1}{2} |\xi|^2 |\hat{u}(\xi)|^2.
$$

(28)

Based on (27), (28), $J^*(\xi)$ can be simplified as follows

$$
J^*(\xi) \leq -\xi^T Q_\xi^2 u^T B^T u + \frac{1}{2} |\xi|^2 + \frac{1}{2} |\xi|^2 |\hat{u}(\xi)|^2 + \omega_{alt}.
$$

(30)

with $\omega_{alt} = \phi_{ma}^2 \hat{w}_a^2 + \epsilon_{Rm}^2 + 0.5|B^{-1}|^2 \frac{1}{2} \left( \phi_{ma}^2 + \phi_{ma}^2L_a^2 \right)|\hat{w}_a|^2$.

According to [29], it can be proven that $\mathcal{L}_{i3}(t) < 0$, as long as $\|\hat{W}_c\| > \rho_w$, $||\nabla J(\xi)|| > \rho_j$ hold, where $\rho_w = \max\left(\frac{\phi_{ma}^2 \phi_{ma}^2 \phi_{ma}^2}{4\xi^2}, \frac{\phi_{ma}^2 \phi_{ma}^2 \phi_{ma}^2}{4\xi^2}\right), \rho_j = \max\left(\frac{\phi_{ma}^2 \phi_{ma}^2 \phi_{ma}^2}{4\xi^2}, \frac{\phi_{ma}^2 \phi_{ma}^2 \phi_{ma}^2}{4\xi^2}\right)$, $\mu_1 = 0.5 \frac{1}{4\xi^2}$, $\mu_2 = 0.5 \frac{1}{4\xi^2}$.

By integrating (30) and (31), the derivative of (24) is

$$
\dot{\mathcal{L}}_{i1}(t) \leq -[(1 - \theta)\xi^T A_\xi - 0.5 \phi_{ma}^2 \phi_{ma}^2 |\delta_j(t)|^2 |\hat{w}_a|^2 - \theta \xi^T A_\xi + \omega_{alt}.
$$

(32)

Obviously, if all the conditions in Theorem 1 are satisfied, we can obtain $\mathcal{L}_{i1}(t) \leq 0$. 

2691
Case ii: To prove the stability at triggered instant \( t = \psi_j + 1 \), the difference of Lyapunov function is defined by
\[
\Delta L_1 = J^*(\xi^+) - J^*(\xi_j^+) + J^*(\xi_j^+) - J^*(\xi_j) + \Delta L_{13} + \Delta L_{14},
\]
(33)
where
\[
\Delta L_{13} = 0.5\dot{w}_a^T \alpha_a^{-1} \dot{w}_a - 0.5\dot{w}_a^T \alpha_a^{-1} \dot{w}_a,
\]
\[
\Delta L_{14} = 0.5\dot{W}_c^T \dot{W}_c^+ - 0.5\dot{W}_c^T \dot{W}_c + V(\xi^+) - V(\xi_j).
\]
A similar proof procedure can now be inferred from [33]. Based on the two cases, it is concluded that the system stability and the weight convergence can be ensured under the triggering rule (22). The proof is completed.

B. Dynamic Event Triggering Scheme

Following the procedure in [34], a dynamic event-sampled scheme (d-ESS) is designed based on the above learning control.

First, a filter-based dynamic equation with respect to additional dynamic variable \( \eta_\xi(t) \) is introduced
\[
\dot{\eta}_\xi(t) = -\kappa \eta_\xi(t) + \left\{ (1 - \theta) \lambda_{\min}(A)||\xi(t)||^2 - 0.5g_{W}^2L_A||\dot{w}_a||^2||\delta(t)||^2 \right\},
\]
where \( \eta_\xi^0 = \eta_\xi(0) \geq 0 \), and \( \kappa \) is a positive filter constant. Therefore, with the variable \( \eta_\xi \), the dynamic triggering condition is designed as
\[
||\delta(t)||^2 \leq \frac{2(1 - \theta) \lambda_{\min}(A)||\xi(t)||^2}{g_{W}^2 ||B^{-1}||^2 (g_{W}^2 L_A^2 + \phi_{E}^2 L_A^2)||\dot{w}_a||^2} + \frac{2 \eta_\xi}{\beta g_{W}^2 L_A||\dot{w}_a||^2},
\]
where the right side of (34) is denoted as \( \Upsilon_\Omega^D \) representing triggering threshold, \( \beta > 0 \) denotes the adjustable parameter. Note that, if \( \beta \to \infty \), two triggering schemes will be the same.

Remark 4: In contrast to the s-ESS, the main advantage of the d-ESS is that it reduces the conservativeness of the stability condition. That is, \( (1 - \theta) \lambda_{\min}(A)||\xi(t)||^2 - \frac{1}{2}g_{W}^2 L_A||\dot{w}_a||^2||\delta(t)||^2 \) does not always need to be non-negative. In addition, the dynamic variable \( \eta_\xi(t) \) is always non-negative (illustrated in simulation study). Next, the stability of the unconstrained error system is analyzed with the help of d-ESS based learning control.

Theorem 2: For the transformed system (5) with the cost function (6). The weight tuning laws of the NN are computed by (17) and (20). Let events be triggered via the dynamic triggering scheme (34). Then, the unconstraint error system is asymptotically stable and two weight errors are UUB.

Proof: First, we select the following Lyapunov function
\[
L_2(t) = L_1(t) + \eta_\xi(t).
\]
By combining Theorem 1, the derivative of \( L_2(t) \) can be written as follows
\[
\dot{L}_2(t) = -\theta \xi^T A \xi - \lambda \eta_\xi(t) + \omega_b + \dot{L}_{13}(t).
\]
It is obvious that, as long as (23) is satisfied, one has \( \dot{L}_2(t) < 0 \). Hence, both \( \eta_\xi(t) \) and \( \xi(t) \) will converge to zero. This completes the proof.

V. Numerical Simulation

In this section, a simulation example is provided based on a 2-DOF robotic system [29] to verify the effectiveness and performance of the proposed theoretical results.

Fig. 1. Evolution of the weight parameters: (a) critic and (b) actor NN.

The system matrices for the 2-DOF robot are given as
\[
M(q) = \begin{bmatrix}
p_1 + p_2 + 2p_3 \cos(q_2) & p_2 + p_3 \cos(q_2) \\
p_2 + p_3 \cos(q_2) & p_2
\end{bmatrix},
\]
\[
C(q, \dot{q}) = \begin{bmatrix}
-p_3 \dot{q}_2 \sin(q_2) & -p_3(q_1 + \dot{q}_2) \sin(q_2) \\
-p_3 \dot{q}_1 \sin(q_2) & 0
\end{bmatrix},
\]
\[
G(q) = \begin{bmatrix}
0 & 0
\end{bmatrix}^T,
\]
where \( p_1 = m_1 l_1^2 + m_2 l_2^2 + N_1, \quad p_2 = m_2 \tilde{l}_2^2 + N_2, \quad p_3 = m_2 l_1 l_2, \quad p_4 = m_1 l_2^2 + m_2 \tilde{l}_2^2, \quad p_5 = m_2 l_1 l_2 \) with \( l_1 = 1/2, \quad l_2 = l_2/2 \), and \( q = [q_1, q_2]^T, \quad \dot{q} = [q_1, \dot{q}_2]^T \). The motion constraints on the robotic system are set as
\[
-1.5 rad < q_1 < 1.5 rad, -1.5 rad < q_2 < 1.5 rad, \\
-2 rad/s < \dot{q}_1 < 2 rad/s, -2 rad/s < \dot{q}_2 < 2 rad/s.
\]
In simulation study, we set the desired state of the robotic system as $q_d = [0, 0]^T$. Hence, the constraints for the joint position error and velocity error are

$$-1.5\text{rad} < e_1 < 1.5\text{rad}, -1.5\text{rad} < e_2 < 1.5\text{rad},$$

$$-2\text{rad}/s < e_1 < 2\text{rad}/s, -2\text{rad}/s < e_2 < 2\text{rad}/s.$$  

The Lyapunov function candidate is chosen as $V(\xi) = \xi_1^2 + \xi_2^2 + \xi_3^2$. The activation function is selected as

$$\Phi(\xi) = [\xi_1^2, \xi_1 \xi_2, \xi_1 \xi_3, \xi_1, \xi_2^2, \xi_2 \xi_3, \xi_3^2]^T.$$  

We set the initial values of the joint states as $q(0) = [0.6, 0.6, 0.05, 0.05]^T$. The parameters are selected as $A=I_4$, $B = 0.8I_2$, $\theta = 0.2$, $g_m = 15$, $\eta_\xi(0) = 2$, $\beta = 0.3$, $\kappa = 0.5$, $\alpha_1 = \alpha_2 = \alpha_3 = 0.8$, the sampling time is set as $T = 0.01s$.

The learning processes of the critic NN and actor NN are displayed in Fig. 1(a) and Fig. 1(b), respectively. After a period of learning process, both of the weights of the two networks can converge to the optimal values. We can observe that the weight parameters of actor NN is piecewise updating, which means the weight parameters are only updated when an event is triggered according to the dynamic triggering condition (34).

Fig. 2(a) shows the evolution of states $\xi$ for the transformed error system. Fig. 2(b) gives the evaluation process of the motion tracking error $e = [e_1^T, e_2^T]^T$, which illustrates that the joint states of the original robotic system can track the desired trajectory without obeying the motion constraint conditions, i.e., $\lambda_1^l < e_1 < \lambda_2^u$. Fig. 3 shows the comparison of control policies $u = [u_1, u_2]^T$ under traditional time-sampled scheme (TSS) and d-ESS, respectively. It can be seen that, by using the proposed event-sampled learning control scheme, the control policy is only updated at event instant and stays the same at triggering interval, i.e., $\psi_j \leq t < \psi_{j+1}$. Moreover, the relationship between triggering error $\delta_j(t)$ and event triggering threshold $\Upsilon_j^D$ is provided in Fig. 4. From Fig. 4, we can verify that $\eta_\xi$ is always non-negative. In summary, these simulation results substantiate the effectiveness of the proposed event-sampled learning control method.
VI. CONCLUSION
This paper proposes an event-sampled learning control strategy for robotic system with motion constraint on joint state. To cope with the constrained problem, a state transformation method has been proposed to transfer the original control problem into an unconstrained problem for an error system. Then, an optimal control problem has been formulated for the transformed error system from optimization viewpoint. To obtain the corresponding optimal control policy in an online manner, a novel learning control approach has been proposed by co-designing NN and two event-sampled schemes, which can further decrease the sampling times for computation. The closed loop system is shown to be stable. Simulation results have shown the effectiveness of the proposed event-sampled intelligent learning method. In the future, we will test our proposed control on a real exoskeleton robot [36].

REFERENCES