Adaptive Learning and Sampled-Control for Nonlinear Game Systems Using Dynamic Event-Triggering Strategy

Chaoxu Mu, *Senior Member, IEEE*, Ke Wang, *Graduate Student Member, IEEE*, and Zhen Ni, *Member, IEEE*

***Abstract*— Static event-triggering-based control problems have been investigated when implementing adaptive dynamic programming algorithms. The related triggering rules are only current state-dependent without considering previous values. This motivates our improvements. This article aims to provide an explicit formulation for dynamic event-triggering that guarantees asymptotic stability of the event-sampled nonzero-sum differential game system and desirable approximation of critic neural networks. This article first deduces the static triggering rule by processing the coupling terms of Hamilton–Jacobi equations, and then, Zeno-free behavior is realized by devising an exponential term. Subsequently, a novel dynamic-triggering rule is devised into the adaptive learning stage by defining a dynamic variable, which is mathematically characterized by a first-order filter. Moreover, mathematical proofs illustrate the system stability and the weight convergence. Theoretical analysis reveals the characteristics of dynamic rule and its relations with the static rules. Finally, a numerical example is presented to substantiate the established claims. The comparative simulation results confirm that both static and dynamic strategies can reduce the communication that arises in the control loops, while the latter undertakes less communication burden due to fewer triggered events.**

***Index Terms*— Adaptive dynamic programming (ADP), dynamic event-triggering, dynamic variable, neural networks (NNs), nonzero-sum differential game (NZSDG).**

I. Introduction

With the increasing complexity of control tasks, cyber–physical systems with multiple control inputs have attracted great attention and received extensive studies, such as multiagent systems and multiplayer differential games [1]–[3]. This class of systems is characterized by several control loops (or information networks) in which communication and computational resources are shared; in addition, for real-world applications, the controller is usually implemented digitally by means of microprocessors [2]. Therefore, it makes significant sense to achieve fewer control actions and infrequent communication while guaranteeing system performance, and hence, the aperiodic event-triggering strategy (ETS) is proposed to substitute the traditional time-triggering strategy (TTS). This article concentrates on developing a novel *dynamic event-triggering strategy* (DETS) for the multiplayer nonzero-sum differential game (NZSDG) learning-based systems. A new dynamic variable is introduced into the triggering rule. In order to clearly elaborate the necessity of this study and the innovative work, literature survey and related studies are first introduced in three aspects: the studied system, learning control, and triggering strategies.

A. Studied System

NZSDG is used to model a system where exists multiple players (or controllers) with different optimization goals that are usually named as cost functions. Basar and Olsder [4] made a comprehensive formulation of this problem and gave some detailed theoretical analysis. It states that every player has the equal privilege to share system information and mutual policies and then unilaterally minimizes its cost subject to the system dynamics. This article follows the description in [4], saying the control input is the control policy. Early studies focused on the mathematical formulation, such as the literature [5] and the references therein. Recent studies pay more attention to effectively solving NZS, namely, obtaining Nash equilibrium, seeing [6]–[12]. It also can be known that, for a nonlinear NZSDG, its optimal costs correspond to the coupled Hamilton–Jacobi (HJ) equations. It may not always be feasible to analytically derive optimal solutions to these equations, and thus, researchers have begun to adopt a learning method named adaptive dynamic programming (ADP) to approximately obtain the ideal equilibrium results.

B. Learning Control

ADP is a branch of reinforcement learning and is increasingly prevalent in the field of optimal control, which adopts approximation structures, such as neural networks (NNs), to approximate a nonlinear function by iterative learning [13]–[17]. Prokhorov *et al.* [18] and Si and Wang [19] have illustrated the core idea of this learning technique: a critic

Manuscript received 31 October 2019; revised 22 May 2020; accepted 24 January 2021. Date of publication 23 February 2021; date of current version 2 September 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62022061 and Grant 61773284 and in part by the National Science Foundation under Grant 1949921. *(Corresponding author: Chaoxu Mu.)*

Chaoxu Mu and Ke Wang are with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: cxmu@tju.edu.cn; walker_wang@tju.edu.cn).

Zhen Ni is with the Department of Computer, Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431 USA (e-mail: zhenni@fau.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3057438.

Digital Object Identifier 10.1109/TNNLS.2021.3057438

signal evaluates the current control policy and improves it in the next cycle, which is easy to implement. Thus, ADP-based control algorithms and related applications have been heavily investigated [20]–[26]. For example, Wei *et al.* [20] devised the value iterative algorithm for discrete-time systems, and Esfandiari *et al.* [23] studied the optimal neurocontroller for nonaffine systems with constrained inputs. For the optimal switching problem of the DC–DC buck circuit, the ADP-based controller was applied to generate the desired voltage in [25]. In addition, recent algorithm design and specific implementation have been comprehensively surveyed in [1]. The focus of this article is not on the exploration of new algorithms but on the computational improvement of adaptive critic algorithm for NZSDG systems. To obtain efficient computation and transmission, we feel that dynamic event-triggering rules must be explored. This motivates the following work of our article.

C. Triggering Strategies

In the abovementioned literature, data sampling and control updating are both performed in a periodic form, namely traditional TTS. However, it is desired to keep the controller inactive when the system tends to be stable or the performance is satisfied [27]–[29]. Tabuada [30] first proposed a static event-triggering strategy (SETS) to manage the sampling and ensured the system stability in the sense of input-to-state. Specifically, the author designed a triggering rule (condition), which latently determined a threshold. Then, an event occurs when the current error signal exceeds this predesigned threshold. This idea inspired most following studies [31]–[34]. The triggering conditions designed in [31]–[34] have greatly reduced the total number of sampling while excluding the instantaneous Zeno behavior. On the other hand, due to aperiodic communication and intermittent control, ETS is also gradually attracting attention in learning-based systems. Recently, some researchers have explored how to introduce an event generator (EG) when implementing ADP, so as to ameliorate learning efficiency [35]–[39]. These event-based ADP algorithms have improved computational performance without seriously influencing the approximation accuracy, and NN function approximators have also been presented with a Zeno-free behavior. In detail, Szanto *et al.* [35] proposed the event communication strategy for the strict-feedback system. The authors [36]–[39] investigated the event-based control policies for affine nonlinear systems, and their triggering conditions were derived with the aid of Lipschitz assumptions for control law or Hamiltonian. It is worth stressing that the above-discussed ETS are considered as *static* mainly because the corresponding triggering rules are only related to the values of the current state and error signal, without concerning the previous values.

Admittedly, SETS reduces the redundant transmissions and updatings compared to TTS, but its rule is still worth improving. Under this motivation, in [40], by defining an internal dynamic variable, Girard proposed a new triggering method termed DETS that generated a larger triggering interval (the time span between two adjacent events). Moreover, it relaxed the stability requirements in [30]. Subsequent studies also confirmed this conclusion, some representatives, such as [41]–[44]. For distributed linear multiagent systems, Ge and Han [41] and Hu *et al.* [42] achieved formation control and consensus control based on dynamic triggering communication, respectively. Their dynamic variables were affected by the previous values of agents' states. Moreover, Antunes and Khashooei [44] explored the application of DETS in linear quadratic control and checked its advantages by comparison. The above is another motivation that we study adaptive learning from the viewpoint of dynamic triggering.

From the literature review and discussions, it can be concluded that there still exist some questions in current studies: 1) existing algorithms for multiplayer NZSDG systems are all implemented by TTS [6]–[12], where every player's policy must be updated in a time-triggering manner; 2) existing methods in [30], [40], and so on need to theoretically calculate the minimal triggering interval to avoid the trap of Zeno behavior, which brings design difficulties for triggering rules; and 3) existing event-based ADP methods use SETS to improve the computational effectiveness. In comparison, DETS is theoretically valuable and has salient advantages, and thus, it deserves to explore in learning control. This article addresses these issues by specializing in a nonlinear two-player NZSDG. The contributions are condensed as follows.

1) *Significantly, by extending our previous work [45] and [30], static triggering rules are developed for players reducing their control computation.* Because existing triggering rules may not scale well for an NZSDG system due to its complicated coupling relations, using the rules, two players update their control policies simultaneously.
2) *Delicate theoretical analysis is delivered to explain characteristics of triggering strategies.* Theoretical results certify that, due to fewer events and guaranteed approximations, the dynamic triggering is superior to static triggering. In addition, the Zeno behavior is effectively avoided by adding an adjustable term in the triggering rule.
3) *Novel dynamic triggering strategy for approximating optimal costs and an event communication protocol will be designed.* A mathematically equivalent filter structure is adopted to generate the dynamic variable, which is restricted by an exponential decaying signal to keep nonnegative and can dynamically adjust the threshold. This is the first treatment that DETS is applied to adaptive critic learning.

The remainder of this article is arranged as follows. Section II introduces the background of a two-player NZSDG nonlinear system and the formulation of event-based control. Section III shows the critic approximation learning mechanism with event-based control policies. Section IV designs static triggering rules and proposes the dynamic triggering rule with guaranteed stability and convergence. Afterward, Section V provides the simulation results and comparative analysis. Finally, Section VI summarizes this article.

II. Background and Problem Formulation

After giving notations, some backgrounds of two-player NZSDG are briefly recalled. Then, an associated event formulation is presented.

A. Necessary Notations

In the subsequent descriptions, one lets $\Re$, $\Re^n$, and $\Re^{n\times m}$ be the set of real numbers, the n-dimensional Euclidean space, and the set of real $n \times m$ matrices, respectively. $\Re_0^+$ is all nonnegative real numbers, while $\mathbb{N}_0^+$ means the positive integers. Besides, "I_n" denotes the $n\times n$ identity matrix. As for mathematical operations, "$\top$" means the transpose, and "$\|\cdot\|$" means the norm; "∇" is the gradient operator; "$\cup$" and "$\cap$" are the *union* and *intersection*; and "$\lambda_{\min}(\cdot)$" and "$\lambda_{\max}(\cdot)$" represent the minimal eigenvalue and maximal eigenvalue, respectively.

Define the class-$\mathcal{K}$: a continuous function $\alpha : \Re_0^+ \to \Re_0^+$ will be of class-$\mathcal{K}$ if it is strictly increasing with initial value being $\alpha(0)=0$; in addition, a class-$\mathcal{K}$ function α is seen to be the class-$\mathcal{K}_\infty$ if it satisfies $\alpha(r)\to\infty$ as $r\to\infty$. Define $\beta(\zeta^-)$ as the left limit of a function $\beta(r)$ when $r\to\zeta$ from the left, i.e., $\beta(\zeta^-)=\lim_{\epsilon\to 0}\beta(\zeta-\epsilon)$; similarly, $\beta(\zeta^+)$ denote the right limit. A function $f(x):\Re^n\to\Re^m$ is Lipschitz continuous on compact set $\Omega\in\Re^n$ if the relation $\|f(x_1)-f(x_2)\|\le\mathscr{L}\|x_1-x_2\|$ exists for all $x_1, x_2\in\Omega$ with the constant $\mathscr{L}>0$.

B. Nonlinear Nonzero-Sum Differential Games

Consider a class of two-player nonlinear NZSDG in input-affine form modeled by

$$\dot{x}(t)=f(x(t))+g_1(x(t))u_1(t)+g_2(x(t))u_2(t),\ \ x_0=x(0) \quad (1)$$

where $x\in\Re^n$ denotes the system state; $f(x)\in\Re^n$ is the drift dynamics; and $g_1(x):\Re^n\to\Re^{n\times q}$ and $g_2(x):\Re^n\to\Re^{n\times l}$ are control input dynamics.[1] In addition, $u_1(t)\in\Re^q$ and $u_2(t)\in\Re^l$ denote control policies made by P_1 and P_2, respectively.[2] Let sets $\mathfrak{U}_1$ and $\mathfrak{U}_2$ be the admissible policy spaces of P_1 and P_2, i.e., $u_1\in\mathfrak{U}_1$ and $u_2\in\mathfrak{U}_2$ [15]. It is assumed that $f(x)+g_1u_1+g_2u_2$ is locally Lipschitz, and the system can be stabilized and each player has access to the feedback state information. Moreover, some standard assumptions are given by Assumption 1.

Assumption 1 (see [7], [9]): For system (1), it has the following properties.

1) $f(x)$ satisfies $f(0)=0$ and $\|f(x)\|\le\mathscr{L}_f\|x\|$.
2) $g_1(x)$ and $g_2(x)$ satisfy $\|g_1(x)\|\le\mathcal{G}_1$ and $\|g_2(x)\|\le\mathcal{G}_2$.

Note that the multiplayer case of (1) is $f(x)+\sum_{j=1}^N g_j(x)u_j$, which can be employed to describe such a multicontroller dynamical system with every player trying to optimize individual performance index. In real-world scenarios, multiplayer NZSDG can be used to design optimal motion planning for multiple robots with different goals, coordinate the charging of autonomous electric vehicles, and design load frequency controllers for power systems. For example, in a multiarea power system, the states are usually defined to describe deviations of certain variables from their target or equilibrium values (such as frequency deviation and power deviation); besides, frequency regulation controllers of different areas are considered as independent players. For clear presentations, the subsequent designs are still based on the two-player system.

Next, for two players, define the scalar infinite-horizon cost function $J_i:\mathfrak{U}_1\times\mathfrak{U}_2\to\Re$ as

$$J_1\triangleq J_1(x_0,u_1,u_2)=\int_t^\infty r_1(x(\nu),u_1(\nu),u_2(\nu))d\nu \quad (2a)$$

$$J_2\triangleq J_2(x_0,u_1,u_2)=\int_t^\infty r_2(x(\nu),u_1(\nu),u_2(\nu))d\nu \quad (2b)$$

with the immediate cost $r_i:\Re^{n+q+l}\to\Re,\ i=1,2$

$$r_i(x,u_1,u_2)=x^\top Q_i x+u_1^\top R_{i1}u_1+u_2^\top R_{i2}u_2$$

where $x^\top Q_i x$, $Q_i\ge 0$ denotes the state penalty and $u_1^\top R_{i1}u_1$ represents the control cost caused by P_1. Besides, the corresponding matrices satisfy $R_{11}=R_{11}^\top>0$, $R_{22}=R_{22}^\top>0$, $R_{12}=R_{12}^\top\ge 0$, and $R_{21}=R_{21}^\top\ge 0$.

Together, this game can be formally defined by

$$\mathcal{G}_N=\{(P_1\cup P_2),\{u_1,u_2\},\{J_1,J_2\}\},\quad u_1\in\mathfrak{U}_1,u_2\in\mathfrak{U}_2.$$

In $\mathcal{G}_N$, every player wants to find its optimal control policy in the process of optimizing the predefined cost, and the final ideal gaming result can be described by Definition 1.

Definition 1 (see [4]): For an N-player NZSDG, such a control policy set $\{u_1^*,\ldots,u_i^*,\ldots,u_N^*\}$, $i\in\mathbb{N}_0^+$ can be regarded to achieve a Nash equilibrium if the following cost relations are satisfied

$$J_i^*\triangleq J_i\big(u_1^*,\ldots,u_i^*,\ldots,u_N^*\big)\le J_i\big(u_1^*,\ldots,u_i,\ldots,u_N^*\big)\quad\forall i. \quad (3)$$

The associated optimal costs $J_1^*,\ldots,J_i^*,\ldots,J_N^*$ are usually called as Nash equilibrium solutions.

Therefore, the desired equilibrium result is expressed by

$$\mathcal{G}_N^*=\big\{(P_1\cup P_2),\big\{u_1^*,u_2^*\big\},\big\{J_1^*,J_2^*\big\}\big\},\quad u_1^*\in\mathfrak{U}_1,u_2^*\in\mathfrak{U}_2.$$

According to previous studies [6]–[12], $\mathcal{G}_N^*$ can be computed by the following standard process.

1) First, by denoting $\nabla J_i\triangleq\partial J_i/\partial x$, $i=1,2$, the Hamilton function of player i is defined by

$$H_i(x,\nabla J_i,u_1,u_2)=r_i(x,u_1,u_2)+\nabla J_i^\top(f(x)+g_1(x)u_1+g_2(x)u_2). \quad (4)$$

2) Next, optimal control policies can be derived by

$$\frac{\partial H_1}{\partial u_1}=0\ \Rightarrow\ u_1^*(x)=-\frac{1}{2}R_{11}^{-1}g_1^\top(x)\nabla J_1^* \quad (5a)$$

$$\frac{\partial H_2}{\partial u_2}=0\ \Rightarrow\ u_2^*(x)=-\frac{1}{2}R_{22}^{-1}g_2^\top(x)\nabla J_2^*. \quad (5b)$$

[1]The time variable t will be subsequently omitted for brevity.
[2]In the following, player 1 and player 2 are also referred to P_1 and P_2.

3) Finally, the Nash equilibrium is transformed into solving the following two coupled HJ equations:

$$0 = x^\top Q_1 x + \left(\nabla J_1^*\right)^\top f(x) - \frac{1}{4}\left(\nabla J_1^*\right)^\top g_1(x) R_{11}^{-1} g_1^\top(x) \nabla J_1^* + \frac{1}{4}\left(\nabla J_2^*\right)^\top g_2(x) R_{22}^{-1} R_{12} R_{22}^{-1} g_2^\top(x) \nabla J_2^* - \frac{1}{2}\left(\nabla J_1^*\right)^\top g_2(x) R_{22}^{-1} g_2^\top(x) \nabla J_2^* \tag{6a}$$

$$0 = x^\top Q_2 x + \left(\nabla J_2^*\right)^\top f(x) - \frac{1}{4}\left(\nabla J_2^*\right)^\top g_2(x) R_{22}^{-1} g_2^\top(x) \nabla J_2^* + \frac{1}{4}\left(\nabla J_1^*\right)^\top g_1(x) R_{11}^{-1} R_{21} R_{11}^{-1} g_1^\top(x) \nabla J_1^* - \frac{1}{2}\left(\nabla J_2^*\right)^\top g_1(x) R_{11}^{-1} g_1^\top(x) \nabla J_1^*. \tag{6b}$$

It is obvious that the coupling terms (such as $(\nabla J_1^*)^\top g_2(x)$) make it more difficult to solve (6) than general nonlinear equations, which are also the biggest difficulties in designing the triggering rule for an NZSDG system. This is why current ETS-based studies devote to the relatively simple zero-sum and single-player cases (such as [36]–[38] and references therein). These motivate us to develop an ADP-based learning scheme with event-based control policies to obtain $\mathcal{G}_N^*$.

C. Event Sampled-Control Method

In the above description, the state $x(t)$ needs to be sampled continuously. To save communication and reduce computation, an ETS-based control method is employed such that the sampling and updating are driven by defined events.

It is assumed that the controller is implemented on a digital platform. An increasing sequence $\{\tau_s\}_{s=0}^{\infty}$, $\tau_0 = 0$, $\tau_s \in \Re_0^+$, $s \in \mathbb{N}_0^+$ is defined to represent the triggering instants of events, which are generated by an EG. Then, the measurement error of (1) is computed by

$$e_s(t) = x_s - x(t); \quad t \in \left[\tau_s, \tau_{s+1}\right) \tag{7}$$

where $x_s \triangleq x(\tau_s)$ is the sampled state at $t = \tau_s$ and $x(t)$ is the continuous state. A triggering interval $[\tau_s, \tau_{s+1})$ can be thought of as a cycle, in which the error signal $e_s(t)$ is varied randomly and reset to 0 at each τ_s.

Due to the state-feedback structure, the system state will influence the updating of control policies, which can be, thus, calculated by

$$u_1^*(x_s) = -\frac{1}{2} R_{11}^{-1} g_1^\top(x_s) \nabla J_1^*(x_s) \tag{8a}$$

$$u_2^*(x_s) = -\frac{1}{2} R_{22}^{-1} g_2^\top(x_s) \nabla J_2^*(x_s) \tag{8b}$$

with $\nabla J_i^*(x_s) = (\partial J_i^*/\partial x)\,|_{x=x_s}$, $i = 1, 2$. Evidently, one sampling means one transmission and one calculation, which is the main reason why ETS are favored compared to real-time sampling or periodic TTS.

Using a zero-order holder (ZOH) obtains the piecewise continuous control signals

$$u_i^*(t) = \begin{cases} u_i^*(x_s), & t \in \left[\tau_s, \tau_{s+1}\right) \\ -\frac{1}{2} R_{ii}^{-1} g_i^\top(x_{s+1}) \nabla J_i^*(x_{s+1}), & t = \tau_{s+1}. \end{cases} \tag{9}$$

To summarize, the specific event control procedure is that an EG identifies whether the current error signal reaches the triggering rules and then transmits the corresponding state to all players; using new state information, players update their control policies, which will be passed through ZOHs and eventually applied to the system. Therefore, in the event-sampled context, the main design tasks can be given more precisely as follows.

1) How to design an adaptive learning structure using event control signals to approximate the solutions of equation (6)?
2) How to design static triggering rules that can naturally avoid the Zeno behavior to help EG generate the required events?
3) How to design a more efficient dynamic triggering rule with equivalent stability and convergence?

III. Event-Sampled Approximator Design and Corresponding Weight Tuning Laws

This section is dedicated to complete the first task proposed in Section II-C.

A. Implementation of Critic Neural Networks

According to [46], there exists a three-layer feedforward NN that can approximate the optimal solution $J_i^*(x)$ and its gradient $\nabla J_i^*(x)$, $i = 1, 2$ on Ω by

$$J_i^*(x) = w_i^\top \varphi_i(x) + \varsigma_i(x) \tag{10}$$

$$\nabla J_i^* = \nabla \varphi_i^\top w_i + \nabla \varsigma_i. \tag{11}$$

This NN is usually termed "critic" NN, and its ideal weight is $w_i \in \Re^{l_c}$; besides, $\varphi_i(x) : \Re^n \to \Re^{l_c}$ is the activation function with l_c denoting the neuron number of hidden layer and $\varsigma_i(x) \in \Re$ is the reconstruction error. In addition, for any $x \in \Omega$, $\nabla \varphi_i \triangleq \partial \varphi_i(x)/\partial x$ and $\nabla \varsigma_i \triangleq \partial \varsigma_i(x)/\partial x$.

Then, referring to (8), the optimal ETS-based control policies can be derived as

$$u_1^*(x_s) = -\frac{1}{2} R_{11}^{-1} g_1^\top(x_s)\left[\nabla \varphi_1^\top(x_s) w_1 + \nabla \varsigma_1(x_s)\right] \tag{12a}$$

$$u_2^*(x_s) = -\frac{1}{2} R_{22}^{-1} g_2^\top(x_s)\left[\nabla \varphi_2^\top(x_s) w_2 + \nabla \varsigma_2(x_s)\right]. \tag{12b}$$

One can let the practical weight $\hat{w}_i$ estimate w_i, and thus, the approximated ETS-based control policy pair is deduced as

$$\hat{u}_1(x_s) = -\frac{1}{2} R_{11}^{-1} g_1^\top(x_s) \nabla \varphi_1^\top(x_s) \hat{w}_1 \tag{13a}$$

$$\hat{u}_2(x_s) = -\frac{1}{2} R_{22}^{-1} g_2^\top(x_s) \nabla \varphi_2^\top(x_s) \hat{w}_2. \tag{13b}$$

These two control laws will be employed to update control signals when the sampled state x_s is transmitted to players. Furthermore, similar to (9), one can get $\hat{u}_i(t)$, which is the actual control signal applied to the system when learning goes. Note that the critic signal is actually continuous, i.e., $\hat{w}_i(t)$, and t is omitted in case of no ambiguity. When an event is triggered, the current value $\hat{w}_i(\tau_s)\,|_{t=\tau_s}$ will be transmitted for players to update the control policies.

Substituting ideal cost (11) and actual control (13) into (4) gives the event-version Hamilton function

$$H_i(x,w_i,\hat{u}_1,\hat{u}_2) = x^\top Q_1 x+\hat{u}_1^\top R_{i1}\hat{u}_1+\hat{u}_2^\top R_{i2}\hat{u}_2 + w_i^\top \nabla\varphi_i(f(x)+g_1(x)\hat{u}_1+g_2(x)\hat{u}_2)\triangleq \mathscr{E}_{H_i} \tag{14}$$

where $\mathscr{E}_{H_i} = -\nabla\varsigma_i^\top(f(x)+g_1(x)\hat{u}_1+g_2(x)\hat{u}_2)$ denote the unavoidable residual error. In addition, for compact display, $\hat{u}_1\triangleq\hat{u}_1(x_s)$ and $\hat{u}_2\triangleq\hat{u}_2(x_s)$ are also adopted in the follow-up expressions. Next, we present how to achieve $\hat{w}_i \to w_i$.

B. Critic Weight Tuning Laws

Analogously, substituting estimated weight $\hat{w}_i, i=1,2$ and actual control (13) into (4) gives the approximated event-version Hamilton function

$$\hat{H}_i(x,\hat{w}_i,\hat{u}_1,\hat{u}_2) = x^\top Q_1 x+\hat{u}_1^\top R_{i1}\hat{u}_1+\hat{u}_2^\top R_{i2}\hat{u}_2 + \hat{w}_i^\top \nabla\varphi_i(f(x)+g_1(x)\hat{u}_1+g_2(x)\hat{u}_2)\triangleq e_i \tag{15}$$

where e_i is the real-time learning error, and it will be fed back to the approximator. Therefore, one can minimize the square error $E=(1/2)e_1^\top e_1+(1/2)e_2^\top e_2$ to minimize these two learning errors. The associated weight tuning laws can, thus, be derived by using the gradient-descent adaptation

$$\dot{\hat{w}}_1 = -\alpha_1\frac{\partial E}{\partial e_1}\frac{\partial e_1}{\partial \hat{w}_1} = -\alpha_1\frac{\sigma_1}{\left(\sigma_1^\top\sigma_1+1\right)^2}e_1 \tag{16}$$

$$\dot{\hat{w}}_2 = -\alpha_2\frac{\partial E}{\partial e_2}\frac{\partial e_2}{\partial \hat{w}_2} = -\alpha_2\frac{\sigma_2}{\left(\sigma_2^\top\sigma_2+1\right)^2}e_2 \tag{17}$$

with the variable $\sigma_i, i=1,2$ being defined by

$$\frac{\partial e_i}{\partial \hat{w}_i} = \nabla\varphi_i(f(x)+g_1(x)\hat{u}_1+g_2(x)\hat{u}_2)\triangleq\sigma_i. \tag{18}$$

Besides, the term $(\sigma_i^\top\sigma_i+1)^2$ in the denominator is functioned as normalization [7], and $\alpha_1>0$ and $\alpha_2>0$ are learning rates of two critic NNs.

Define the weight error as $\tilde{w}_i = w_i-\hat{w}_i$, and the following relation can be derived by combining (14) and (15):

$$e_i = -\tilde{w}_i^\top\nabla\varphi_i(f(x)+g_1(x)\hat{u}_1+g_2(x)\hat{u}_2)+\mathscr{E}_{H_i}. \tag{19}$$

Integrating (16), (17), and (19), the weight error dynamics are

$$\dot{\tilde{w}}_1 = -\alpha_1\tilde{\sigma}_1\tilde{\sigma}_1^\top\tilde{w}_1+\alpha_1\frac{\tilde{\sigma}_1}{\bar{\sigma}_1}\mathscr{E}_{H_1} \tag{20}$$

$$\dot{\tilde{w}}_2 = -\alpha_2\tilde{\sigma}_2\tilde{\sigma}_2^\top\tilde{w}_2+\alpha_2\frac{\tilde{\sigma}_2}{\bar{\sigma}_2}\mathscr{E}_{H_2} \tag{21}$$

where $\tilde{\sigma}_i$ and $\bar{\sigma}_i, i=1,2$ signify

$$\bar{\sigma}_i = \sigma_i^\top\sigma_i+1,\qquad \tilde{\sigma}_i = \sigma_i/\left(\sigma_i^\top\sigma_i+1\right). \tag{22}$$

It is worth noting that this critic learning method with fewer parameters does not need actors, which excludes some interferences and, hence, is effective for analyzing event-triggering strategies. At the same time, this learning is carried out online and requires persistency-of-excitation (PE) noises. Up until now, the overall learning framework of this event-sampled NZSDG system has been completed, and it is depicted by Fig. 1. As can be seen, this event system is composed of 2 ZOHs, two critic NNs, two players (controllers), and one EG. By computing the error in real time, EG determines whether an event is triggered and then passes the sampled state to two players. Subsequently, two players simultaneously update their control policies, which become continuous control inputs after passing through ZOHs. Also, note that the weight signal is continuous and will be transmitted for players after receiving event signals. In this way, the communication of two loops is reduced, and control computation is also cut down.

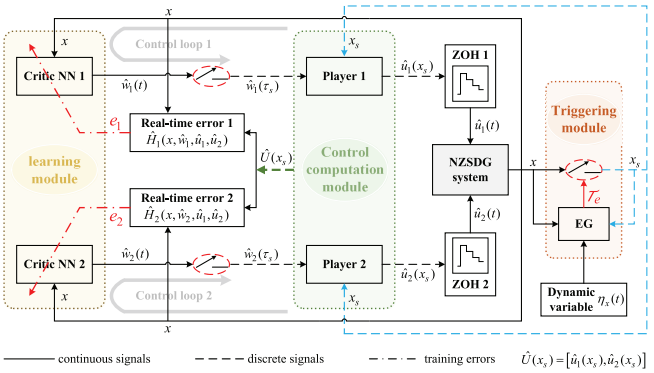


Fig. 1. Schematic.

IV. Improved Event Triggering Rules and Dynamic Triggering Strategy

This section concentrates on tasks 2 and 3. The dynamic triggering rule is proposed, and some theoretical results are provided.

A. "Dynamic Triggering" Versus "Static Triggering"

We begin our design by illustrating the differences between DETS and SETS.

1) Mathematically, the dynamic triggering rule usually involves a differential equation. Such a problem related to differential equations, whose current solutions depend on previous values, can be seen as "*dynamic*."
2) Formally, static triggering rules only include the current values of x and $e_s(t)$, while previous values are not considered.
3) Fundamentally, the static triggering threshold is adaptively adjusted and only time-varying. In contrast, the dynamic triggering threshold includes a dynamic variable.

B. Static Event-Triggering Strategy

1) Normal SETS (NSETS): In the ADP event research, the current work [36]–[39] is devoted to devising SETS rules for single player (ordinary optimal control) and zero-sum cases, and we have also studied the zero-sum problem in recent work [45]. However, these problems are relatively simpler than that of the NZSDG system with complicated coupling relations. Therefore, based on the recent progress and the work of [30], we first devise an NSETS triggering rule.

The first event is thought of to occur at $t=\tau_0=0$ (i.e., give initial values in the algorithm), and subsequent event instants are generated by the following triggering rule:

$$\tau_{s+1}=\inf\{t\in\Re_0^+ \mid t>\tau_s\cap[(1-\theta)x^\top\mathcal{Q}x+\mathcal{U}(x_s) -\mathcal{L}_\Sigma\|e_s(t)\|^2\le 0]\} \quad (23)$$

where $\theta\in(0,1)$ is a conservative design, and $Q_1+Q_2=\mathcal{Q}$ and $\mathcal{U}(x_s)$, $\mathcal{L}_\Sigma$ satisfy

$$\mathcal{U}(x_s)=r_1^2\|\hat{u}_1(x_s)\|^2+r_2^2\|\hat{u}_2(x_s)\|^2 \quad (24)$$

$$\mathcal{L}_\Sigma=\mathcal{L}_1\|\hat{w}_1\|^2+\mathcal{L}_2\|\hat{w}_2\|^2. \quad (25)$$

The rule in (23) indicates that an event is generated whenever $(1-\theta)x^\top\mathcal{Q}x+\mathcal{U}(x_s)=\mathcal{L}_\Sigma\|e_s(t)\|^2$. Therefore, if we use this NSETS, the value of $(1-\theta)x^\top\mathcal{Q}x+\mathcal{U}(x_s)-\mathcal{L}_\Sigma\|e_s(t)\|^2$ remains nonnegative for all $t\in[0,\tau_\infty)$, which means that this rule is the counterpart to the following triggering condition:

$$\|e_s(t)\|^2\le\mathcal{T}_e \quad (26)$$

and the triggering threshold of NSETS can be calculated by

$$\mathcal{T}_e^N=\frac{(1-\theta)x^\top\mathcal{Q}x+\mathcal{U}(x_s)}{\mathcal{L}_\Sigma}. \quad (27)$$

It is easy to understand that the larger this threshold, the larger the triggering interval, which also means fewer events. Next, Theorem 1 illustrates that the rule (23) can guarantee the system's asymptotic stability when critic NNs are trained. Before this, the following assumptions are commonly required (see [7], [9], [36], [39]).

Assumption 2: For system (1), each player satisfies the following conditions.

1) The residual error and ideal critic weight are constrained by $\|\mathscr{E}_{H_i}\|\le\mathcal{E}_i$ and $\|w_i\|\le\mathcal{W}_i$, respectively.
2) $\nabla\varsigma_i$ is upper bounded by $\|\nabla\varsigma_i\|\le\mathcal{B}_{\varsigma_i}$.
3) The gradient $\nabla\varphi_i(x)$ is Lipschitz on $x\in\Omega$ by conforming $\|\nabla\varphi_i(x)-\nabla\varphi_i(x_s)\|\le\mathscr{L}_{\varphi_i}\|e_s(t)\|$, and it is also limited by $\|\nabla\varphi_i\|\le\mathcal{B}_{\varphi_i}$.
4) Control input dynamics $g_i(x)$ is also considered to be Lipschitz on $x\in\Omega$ by $\|g_i(x)-g_i(x_s)\|\le\ell_i\|e_s(t)\|$.

Theorem 1: For the two-player NZSDG system (1), two costs are defined by (2) and are approximated using (10). Let two critic NN train their weights according to (16)–(17); let EG trigger events in the light with rule (23), and two players update their policies using (13). Then, this closed-loop event system will be asymptotically stable, and the weight error $\tilde{w}_i$ will be uniformly ultimately bounded (UUB) if it satisfies

$$\|\tilde{w}_i\|>\sqrt{\frac{\mathfrak{B}_\Sigma}{\alpha_i\lambda_{\min}(\tilde{\sigma}_i\tilde{\sigma}_i^\top)-2\mathfrak{B}_i}},\quad i=1,2. \quad (28)$$

Proof: By considering the requirements of stability and learning, one can select the Lyapunov candidate as

$$L_1(t)=L_{11}(t)+L_{12}(t)+L_{13}(t)+L_{14}(t) \quad (29)$$

with

$$L_{11}(t)=J_1^*(x)+J_2^*(x),\quad L_{12}(t)=J_1^*(x_s)+J_2^*(x_s)$$
$$L_{13}(t)=\frac{1}{2}\tilde{w}_1^\top(t)\tilde{w}_1(t),\quad L_{14}(t)=\frac{1}{2}\tilde{w}_2^\top(t)\tilde{w}_2(t).$$

After making some operations, it yields

$$\dot{L}_1(t)\le-\theta x^\top\mathcal{Q}x<0. \quad (30)$$

For detailed mathematical operations and the related variables $(r_1,r_2,\mathcal{L}_1,\mathcal{L}_2,\mathfrak{B}_\Sigma,\mathfrak{B}_i)$, please see the Appendix.

Remark 1: Note that $(1-\theta)x^\top\mathcal{Q}x+\mathcal{U}(x_s)$ and $\mathcal{L}_\Sigma\|e_s(t)\|^2$ both can satisfy characteristics of class-$\mathcal{K}_\infty$, so if we let $\alpha(\|x\|)=(1-\theta)x^\top\mathcal{Q}x+\mathcal{U}(x_s)$ and $\gamma(\|e_s\|)=\mathcal{L}_\Sigma\|e_s(t)\|^2$, then we can think that the rule (23) is equivalent to (8) of [30]. The difference is that this design does not rely on the input-to-state stable (ISS) [47] analysis.

Remark 2: The above design still needs to address an important issue, namely, how to exclude Zeno behavior. This is an indispensable theoretical guarantee, and the later proof is relatively difficult. Of course, the rule (23) is Zeno-free (see [45] for a similar proof). Noticeably, the following improved rule can avoid it from the design perspective.

2) Improved SETS (ISETS): By introducing an exponential signal $\delta e^{-\xi t}$, where the parameter $\delta\in\Re_0^+$ affects the amplitude, while $\xi\in\Re_0^+$ determines the decaying rate, the triggering rule of ISETS is presented as

$$\tau_{s+1}=\inf\{t\in\Re_0^+ \mid t>\tau_s\cap[(1-\theta)x^\top\mathcal{Q}x+\mathcal{U}(x_s) +\delta e^{-\xi t}-\mathcal{L}_\Sigma\|e_s(t)\|^2\le 0]\}. \quad (31)$$

The following Lemma 1 reveals that the design of this improved triggering rule is also reasonable.

Lemma 1: On the basis of Theorem 1, one uses the triggering rule (31), and then, the system (1) also has asymptotic stability, while the weight error is UUB if (28) holds.

Proof: Consider the following Lyapunov candidate:

$$L_2(t)=L_1(t)+\frac{\delta}{\xi}e^{-\xi t}. \quad (32)$$

It is clear that $L_2(t)$ is positive definite and radially unbounded. Taking its time derivative obtains:

$$\dot{L}_2(t)=\dot{L}_1(t)-\delta e^{-\xi t}. \quad (33)$$

Evidently, if $(1-\theta)x^\top\mathcal{Q}x+\mathcal{U}(x_s)+\delta e^{-\xi t}-\mathcal{L}_\Sigma\|e_s(t)\|^2$ keeps nonnegative for all $t\in[0,\tau_\infty)$, then it derives

$$\dot{L}_2(t)\le-\theta x^\top\mathcal{Q}x<0. \quad (34)$$

This result is consistent with Theorem 1, and this lemma is, thus, established.

Next, we explain why this rule can naturally preclude the Zeno behavior without giving a minimal triggering interval.

Proposition 1: For the proposed ISETS, its triggering rule (31) can avoid the trap of the Zeno behavior.

Proof: Note that, in (23), the variable $(1-\theta)x^\top\mathcal{Q}x+\mathcal{U}(x_s)$ may become 0 during the regulation, which means that the triggering condition may become

$$-\mathcal{L}_\Sigma\|e_s(t)\|^2\le 0. \quad (35)$$

It is obvious that it is always satisfied, which may cause the accumulations of events in a very short interval, also known as the Zeno behavior. In this case, it is necessary to theoretically prove that there exists a explicit minimal triggering interval $\tau_{\min}=\min[\tau_{s+1}-\tau_s]>0$.

Therefore, we introduce the exponential term $\delta e^{-\xi t} > 0$ to deal with this problem. It can be seen that the relationship (35) now becomes $\delta e^{-\xi t} - \mathcal{L}_\Sigma \|e_s(t)\|^2 \le 0$, which avoids the infinite loop, which accomplishes this proof.

Remark 3: It is worth emphasizing that this exponential term just plays an adjusting role, so its magnitude should be controlled by δ to be relatively small; in addition, the decaying rate ξ should be selected to be asymptotically convergent according to the system operation.

C. Dynamic Event-Triggering Strategy

As can be seen from (23) and (31), SETS rules only depend on $x(t)$ and $e_s(t)$ without considering previous values. This design leads to the fact that the SETS rule fails to adjust its triggering threshold according to the system's overall circumstances, and hence, ISETS can be further improved. Using the dynamic idea in [40], one defines an internal dynamic variable η_x described by the following differential equation:

$$\dot{\eta}_x = -\lambda \eta_x + \left\{ \underbrace{(1-\theta)x^\top \mathcal{Q} x + \mathcal{U}(x_s) + \delta e^{-\xi t} - \mathcal{L}_\Sigma \|e_s(t)\|^2}_{\Gamma(x, e_s)} \right\}$$
$$\eta_x^0 = \eta_x(0) \ge 0. \tag{36}$$

Intuitively, (36) can be regarded as a first-order filter, where η_x is the filtered value of $\Gamma(x, e_s)$. In other words, this dynamic variable is actually a processed signal. The parameter $\lambda > 0$ represents the filtering coefficient that characterizes the extent to which $\Gamma(x, e_s)$ affects η_x.

A straightforward advantage behind the proposed DETS is that it relaxes the stability requirement, that is to say, it is unnecessary to keep $\Gamma(x, e_s)$ always nonnegative. This can be achieved by ensuring that η_x is nonnegative for all time. Therefore, events can be triggered using such a DETS defined by the following rule:

$$\tau_{s+1} = \inf\Big\{ t \in \mathfrak{R}_0^+ \mid t > \tau_s \cap \Big[\eta_x(t) + \beta\big((1-\theta)x^\top \mathcal{Q} x + \mathcal{U}(x_s) + \delta e^{-\xi t} - \mathcal{L}_\Sigma \|e_s(t)\|^2\big) \le 0 \Big] \Big\} \tag{37}$$

where $\beta \in \mathfrak{R}_0^+$ is an adjustment parameter that provides a bridge between SETS and DETS; if $\beta \to \infty$, then the rule (37) becomes (31). Refer to Remark 6 for a detailed analysis. Proposition 2 states why $\eta_x(t)$ can remain nonnegative throughout the control stage.

Proposition 2: Let $e_s(t)$ and $\eta_x(t)$ be computed by (7) and (36). If the system adopts (37) to generate events, then $\eta_x(t) \ge 0$ always holds.

Proof: Note that when EG triggers events with the dynamic rule (37), for any $t \in [0, \tau_\infty)$, this inequality

$$\eta_x + \beta\big((1-\theta)x^\top \mathcal{Q} x + \mathcal{U}(x_s) + \delta e^{-\xi t} - \mathcal{L}_\Sigma \|e_s(t)\|^2\big) \ge 0 \tag{38}$$

is obvious. First, if $\beta = 0$, then $\eta_x(t) \ge 0$ is true.

Second, if $\beta \ne 0$, by combining (36) and (38) and considering $\beta \in \mathfrak{R}_0^+$, it can ensure the following relation:

$$\dot{\eta}_x(t) + \lambda \eta_x(t) \ge -\frac{\eta_x(t)}{\beta}, \quad \eta_x^0 \ge 0. \tag{39}$$

Then, by means of the comparison lemma [47], it follows:

$$\eta_x(t) \ge \eta_x^0 e^{-\left(\lambda + \frac{1}{\beta}\right)t}, \quad t \in [0, \tau_\infty) \tag{40}$$

which means that $\eta_x(t)$ is restricted by a positive exponential signal, so one can obtain $\eta_x(t) \ge 0$.

On this basis, we give Theorem 2 to illustrate the dynamic variable (36), and the rule (37) can also ensure the performance of the learning-based system.

Theorem 2: Based on Lemma 1, let EG generate events by (37), the system variables $x(t)$ and $\eta_x(t)$ are asymptotically stable, the error $e_s(t)$ is asymptotically convergent, and the weight error $\tilde{\omega}_i(t)$ is convergent in the sense of UUB if (28) holds.

Proof: Select the following Lyapunov candidate function $L_3(t) : \mathfrak{R}_0^+ \times \mathfrak{R}_0^+ \to \mathfrak{R}_0^+$:

$$L_3(t) = L_2(t) + \eta_x(t). \tag{41}$$

Considering related derivations of Theorem 1 and Lemma 1, $\dot{L}_3(t)$ can be developed as

$$\begin{aligned} \dot{L}_3(t) \le & -(1-\theta)x^\top \mathcal{Q} x - \mathcal{U}(x_s) - \delta e^{-\xi t} + \mathcal{L}_\Sigma \|e_s(t)\|^2 + \mathfrak{B}_\Sigma \\ & - \sum_{i=1}^{2} \left[\left(\frac{1}{2} \alpha_i \lambda_{\min}\left(\tilde{\sigma}_i \tilde{\sigma}_i^\top\right) - \mathfrak{B}_i \right) \|\tilde{\omega}_i\|^2 \right] - \theta x^\top \mathcal{Q} x - \lambda \eta_x \\ & + (1-\theta)x^\top \mathcal{Q} x + \mathcal{U}(x_s) + \delta e^{-\xi t} - \mathcal{L}_\Sigma \|e_s(t)\|^2. \end{aligned} \tag{42}$$

It is clear that, as long as (28) is satisfied, one can obtain

$$\dot{L}_3(t) \le -\theta x^\top \mathcal{Q} x - \lambda \eta_x < 0. \tag{43}$$

According to the Lyapunov theory, it can be concluded that $L_3(t)$ decreases along with $x(t)$ and $\eta_x(t)$ asymptotically converging to the origin; in addition, note that $\dot{x}(t) = -\dot{e}_s(t)$, and thus, $e_s(t)$ is also asymptotically convergent.

This proof is, thus, completed.

Remark 4: It can be seen from (36) that the calculation of η_x needs previous system information, which exactly explains that the rule (37) does not only focus on the current triggering information. In addition, compared to Lemma 1, Theorem 2 relaxes the requirement for signal $\Gamma(x, e_s)$.

Remark 5: So far, the asymptotically stable issue of two-player NZSDG has been investigated based on dynamic triggering. An important motivation behind this is that the current research efforts are mainly focused on ultimately bounded stability analysis, minimal triggering interval, and static triggering design. This scheme can be easily extended to the multiplayer case because the coupling terms are similar. As for further finite-time stability, time-varying HJ equations and some constraint conditions must be handled [48]. If it comes to exponential stability, the triggering rule is required to improve and maybe controllers need to be redesigned.

Another important motivation that the DETS is devised into the NN learning is to obtain fewer events than SETS and, of course, fewer than TTS. As mentioned earlier, fewer events mean that the triggering interval $T_s = \tau_{s+1} - \tau_s$ is greater. The triggering intervals of three strategies (NSETS, ISETS, and DETS) are denoted by T_s^N, T_s^I, and T_s^D, respectively. Their relationship is explained by Proposition 3.

Proposition 3: Let (23), (31), and (37) determine the arrival of next event τ_{s+1}, respectively; let $\delta, \xi, \beta, \eta_x \in \mathfrak{R}_0^+$ and $\theta \in (0, 1)$, and then, it has $T_s^D \geq T_s^I \geq T_s^N$.

Proof: By referring to the triggering condition (26) and the expression (27), it can deduce the triggering thresholds for the three strategies

$$\mathcal{T}_e^N = \frac{(1-\theta)x^\top \mathcal{Q}x + \mathcal{U}(x_s)}{\mathcal{L}_\Sigma} \tag{44}$$

$$\mathcal{T}_e^I = \frac{(1-\theta)x^\top \mathcal{Q}x + \mathcal{U}(x_s)}{\mathcal{L}_\Sigma} + \frac{\delta e^{-\xi t}}{\mathcal{L}_\Sigma} \tag{45}$$

$$\mathcal{T}_e^D = \frac{(1-\theta)x^\top \mathcal{Q}x + \mathcal{U}(x_s)}{\mathcal{L}_\Sigma} + \frac{\delta e^{-\xi t}}{\mathcal{L}_\Sigma} + \frac{\eta_x}{\mathcal{L}_\Sigma \beta}. \tag{46}$$

It is obvious that, for a given state x_s and error $e_s(t)$, the relation $\mathcal{T}_e^D \geq \mathcal{T}_e^I \geq \mathcal{T}_e^N$ establishes. An intuitive interpretation is that a smaller threshold determines that the next triggering instant will come earlier, so it further derives $\tau_{s+1}^N \leq \tau_{s+1}^I \leq \tau_{s+1}^D$, which means $\tau_{s+1}^D - \tau_s \geq \tau_{s+1}^I - \tau_s \geq \tau_{s+1}^N - \tau_s$, and thus, this proposition is confirmed.

Next, some discussions are attached to provide some insight on how to tune parameters in order to realize conversion between different strategies and how to adjust the controller.

Remark 6: By comparatively analyzing (44)–(46), one can conclude that: 1) when $\delta \to 0$, (45)→(44), and thus, ISETS becomes NSETS; 2) when $\xi \to \infty$, (45)→(44), and thus, ISETS becomes NSETS; 3) when $\beta \to \infty$, (46)→(45), and thus, DETS becomes ISETS. Therefore, DETS will achieve the smallest cumulative number of events and obtain more sophisticated controllers. Furthermore, just for NSETS only, it indicates in (44) that smaller values of $\mathcal{L}_i$ and θ result in fewer events.

Remark 7: Relying on event triggering and critic learning, the controller is parameterized in (13), and the control performance is mainly determined by two parameters: $\hat{w}_i$ and x_s. For the former, it is desired to get a convergence value that is very close to w_i; for the latter, the sampling can be flexibly adjusted (see Remark 6). Frequent sampling benefits the learning accuracy while bringing more communication burden; a good compromise can be obtained by selections.

By incorporating the dynamic triggering strategy, the entire flow of the learning algorithm is presented as Algorithm 1. In the end, the main advantages are summarized as follows.

1) By freeing players from frequent computation, the proposed DETS algorithms can be more efficiently executed than those in [10]–[12]. It is of great sense for multiplayer games with multiple control loops.
2) Unlike previously discussed ADP event designs [35]–[39], the proposed triggering rules overcome the difficulty brought by coupling terms while maintaining the desired stability.
3) Compared with the static triggering in [36]–[39], the designed DETS rule can further reduce control computation, certainly premised on accurate approximate learning. This shows that it is feasible to apply dynamic triggering to adaptive learning.
4) In contrast to the static rules [37]–[39] or dynamic rules [40]–[42], an additional exponential term is incorporated to help the triggering rule naturally avoid the Zeno behavior, which does not impair the asymptotic stability.
5) Different from the classic dynamic triggering [41]–[43], our rule design does not need to rely on ISS analysis; besides, we give an explicit lower bound of the dynamic variable η_x [see (40)], and the simulation will verify this property.

Algorithm 1 Dynamic Event-Triggering-Based Adaptive Critic Learning Algorithm

Input: η_x^0, $e_s(0) = 0$, $\hat{w}_i(0)$ and T (the running time).
Output: $w_i \approx \hat{w}_i(T)$ and S (total number of samples).
1: **Initialization:**
system settings: x_0 and R_{ij}, Q_i, α_i; $i, j = 1, 2$.
triggering parameters: $\lambda, \beta, \theta, \mathcal{L}_1, \mathcal{L}_2, \xi, \delta$.
2: **while** $t \leq T$ **do**
3: Calculate the state error $e_s(t)$ adopting (7);
4: Train the critic weights from (16)–(17) using $\hat{u}_i(x_s)$;
5: Compute the dynamic variable $\eta_x(t)$ by (36).
6: **for** $i = 1, 2$ **do**
7: **if** the triggering rule (37) is satisfied, **then**
8: Record the number $s + 1$ and update S;
9: Update $\tau_{s+1} = t, x_{s+1} = x(t), e_{s+1}(t) = 0$;
10: Compute new policies $\hat{u}_i(x_{s+1})$ by (13).
11: **end if**
12: **end for**
13: **end while**

V. Simulation and Analysis

In this section, a numerical two-player system is simulated using TTS, NSETS, ISETS, and DETS, respectively; some results are provided for substantiating the claimed theoretical achievements. Note that TTS is run by the equidistant period of 0.01 s.

Consider a widely adopted differential game (similar games can refer to [9]–[12]) with nonlinear dynamics

$$\dot{x} = f(x) + g_1(x)u_1 + g_2(x)u_2 \tag{47}$$

with

$$f(x) = \begin{bmatrix} x_2 - 2x_1 \\ -x_2 - 0.5x_1 + 0.25x_2\cos^2(x_1) + 0.25x_2\sin^2(2x_1+1) \end{bmatrix}$$

$$g_1(x) = \begin{bmatrix} 0 \\ \cos(x_1) \end{bmatrix}, \quad g_2(x) = \begin{bmatrix} 0 \\ \sin(2x_1 + 1) \end{bmatrix}.$$

It can be known that when the system parameters are configured according to $R_{11} = R_{12} = 2I_1$, $R_{21} = R_{22} = I_1$, $Q_1 = 2I_2$, $Q_2 = I_2$, the optimal Nash costs are $J_1^*(x) = 1/2x_1^2 + x_2^2$ and $J_2^*(x) = 1/4x_1^2 + 1/2x_2^2$. The entire learning stage lasts for $T = 150$ s starting with $x_0 = [0.1, -0.5]^\top$.

In addition, two critic NNs adopt such settings: activation functions are $\varphi_1(x) = \varphi_2(x) = [x_1^2, x_1x_2, x_2^2]^\top$, learning rates are $\alpha_1 = \alpha_2 = 0.1$, two sets of estimated weight are denote by $\hat{w}_1 = [\hat{w}_{11}, \hat{w}_{12}, \hat{w}_{13}]^\top$ and $\hat{w}_2 = [\hat{w}_{21}, \hat{w}_{22}, \hat{w}_{23}]^\top$, and finally, probing noises are injected in the first 140 s.

TABLE I
COMPARATIVE RESULTS OF THREE STRATEGIES

Strategies	Samples	Average interval	Minimal interval	Structure	System variables	Converged weights	Errors:$\|\tilde{w}_1\|$,$\|\tilde{w}_2\|$
TTS	15000	0.01s	0.01s	2NN	$x(t)$	w_1^T, w_2^T	0.0027, 0.0012
NSETS	2551	0.0588s	0.02s	2NN+2ZOH	$x(t), e_s(t)$	w_1^N, w_2^N	0.0037, 0.0015
ISETS	2391	0.0606s	0.03s	2NN+2ZOH	$x(t), e_s(t)$	w_1^I, w_2^I	0.0037, 0.0016
DETS	1380	0.1079s	0.03s	2NN+2ZOH	$x(t), e_s(t), \eta_x(t)$	w_1^D, w_2^D	0.0102, 0.0041

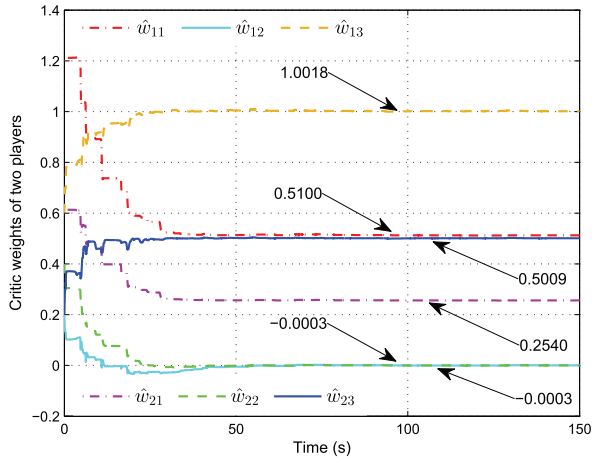


Fig. 2. Two sets of critic weight signals obtained by DETS.

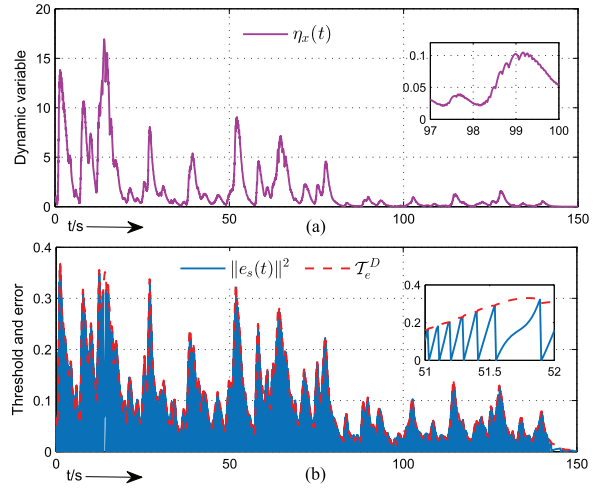


Fig. 3. Responses of dynamic variable and triggering process.

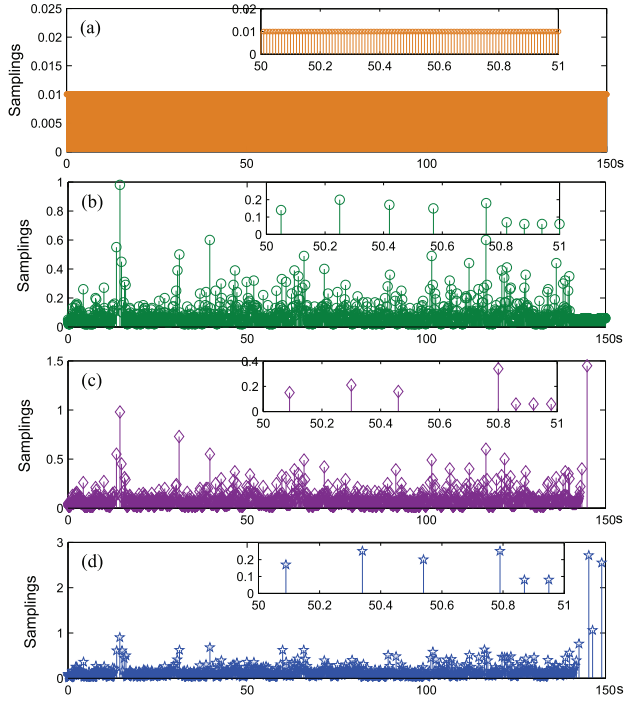


Fig. 4. Triggering comparison of four strategies. (a) TTS. (b) NSETS. (c) ISETS. (d) DETS.

A. Triggering Results During the Learning Stage

1) Triggering Results of DETS: First, by comparing with TTS, which is run by the equidistant period of 0.01 s, the triggering effect of DETS is analyzed by executing (36) and (37). *Basic triggering parameters* are selected as $\eta_x^0 = 1, \lambda = 0.3, \beta = 0.5, \theta = 0.5, \mathcal{L}_1 = \mathcal{L}_2 = 15, \xi = 0.015$, and $\delta = 0.005$, which is a benchmark for later comparison. Run Algorithm 1 to acquire the learning results, as displayed in Figs. 2 and 3. It can be seen that this learning finishes with the converged weights $w_1^D \approx \hat{w}_1 = [0.5100, -0.0003, 1.0018]^\top$ and $w_2^D \approx \hat{w}_2 = [0.2540, -0.0003, 0.5009]^\top$, and these values of DETS closely approximate the ideal weights. The evolving trajectory of η_x is drawn by the magenta line in Fig. 3(a), and obviously, $\eta_x(t) \geq 0$. At the same time, the close-up of triggering logic relation is given in Fig. 3(b), which indicates that $e_s(t)$ and $\mathcal{T}_e^D$ comply with the condition (26).

Next, in order to objectively appraise the proposed DETS, its features and advantages will be illustrated by comparing with TTS, NSETS, and ISETS. Their parameters are exactly consistent with DETS. The total number of samples (events) and the average triggering interval, as well as other comparisons, are listed in Table I, where converged weights are

$$w_1^T = [0.5025, -0.0008, 1.0004]^\top, w_2^T = [0.2510, -0.0007, 0.5002]^\top$$
$$w_1^N = [0.5030, -0.0014, 1.0017]^\top, w_2^N = [0.2512, -0.0005, 0.5008]^\top$$
$$w_1^I = [0.5029, -0.0014, 1.0018]^\top, w_2^I = [0.2512, -0.0005, 0.5009]^\top.$$

The interevent periods of four strategies are recorded in Fig. 4. It is noticeable that the largest average interval and smallest samples are achieved by the DETS in all strategies, and hence, the DETS algorithm is capable of further reducing

TABLE II
ISETS-BASED TRS

TR (%) δ / ξ	0.005	0.015	0.15	0.3
0.005	15.79	15.35	13.26	12.13
0.015	15.94	15.70	14.22	13.44
0.15	16.77	16.72	16.61	16.55
0.3	17.01	17.00	16.99	16.97

TABLE III
TRS UNDER DIFFERENT PARAMETERS. (A) PARAMETERS AND TRS OF DETS. (B) PARAMETERS AND TRS OF NSETS

(a)

β	0.1	0.5	3	7	10	20
TR (%)	10.97	9.20	10.21	11.47	12.03	13.04
λ	0.1	0.3	5	9	12	20
TR (%)	8.43	9.20	14.79	15.41	15.58	15.72

(b)

$\mathcal{L}_i, i=1,2$	15	16	17	18	19	20
TR (%)	17.01	17.67	18.32	19.00	15.95	20.09
θ	0.5	0.6	0.7	0.8	0.9	1
TR (%)	17.01	17.85	18.87	20.21	22.01	24.86

the computational burden. Therefore, the salient advantage of the dynamic rule (37) is that it finds a compromise between the allowed triggered intervals and the approximate performance, which is also corroborated by Fig. 2 and weight errors in Table I. On the other hand, for a time span from 50 to 51 s, starting with TTS, one can see that there are many events that are triggered periodically. Moving on to DETS, one can find that only a few events are triggered.

2) Analysis of the Exponential Term Under ISETS: Under the ISETS-based communication protocol, we analyze the influences of the designed exponential term on the triggering result, which is evaluated by triggering rate (TR). This rate is calculated based on the total number of TTS samples S^T. For example, for basic ISETS, it has

$$\text{TR} = S^I/S^T = 2391/15000 \times 100\% = 15.94\%.$$

Combined with (45) and Table II, one can conclude that the faster the exponential signal decays ($\xi \uparrow$), the shorter it lasts, and then, the rate increases; the larger its amplitude ($\delta \uparrow$), the greater the impact on the threshold, and the rate consequently decreases. These results experimentally confirm the theoretical analysis in Remark 6.

It is also worth noting that 0.3 is seemingly the limit of ξ when $\delta = 0.005$, with the current TR being equal to NSETS, i.e., $2551/15000 \times 100\% = 17.01\%$.

3) Partition Analysis: Note that three triggering rules (23), (31), and (37) are progressively derived, so there should be some specific switching relationships among them. Think of

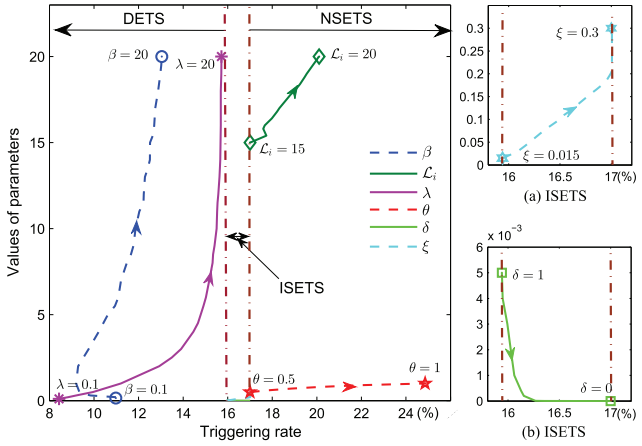


Fig. 5. Strategy partition based on basic triggering parameters.

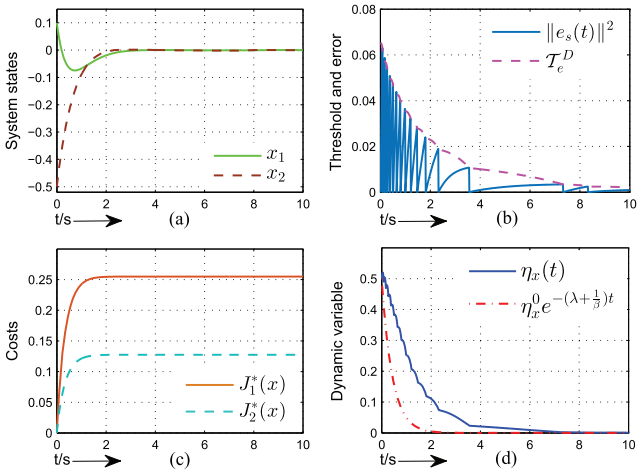


Fig. 6. Event control performance during the testing stage obtained by DETS. (a) Optimal states. (b) Triggering process. (c) Optimal costs. (d) Dynamic variable.

the *basic triggering parameters* in Section V-A1 as a benchmark, by varying related parameters; the resulting strategy partition is depicted by Fig. 5 (so different basic parameters definitely mean different partitions). Fig. 5(a) and (b) shows two close-ups because the zone of ISETS is relatively small. Some representative data are provided in Table III additionally. According to total samples in Table I, it can be known that three zones are partitioned through 15.94% and 17.01%.

In Fig. 5, the blue curve corresponds to the given value of λ with parameter β varying from 0.1 to 20, and other curves are similarly obtained by fixing other parameters. In Table III, the increase caused by β (or $\mathcal{L}_i, \theta$) is a result of the threshold decrease shown in (46). Also, looking at the filtering coefficient λ, a bigger value maybe weakens the filtering effect and, thus, render DETS more static. What needs to noticed is that it may be problematic to give a too small value for $\mathcal{L}_i$ (where $\mathcal{L}_i = \varrho_i(\mathscr{L}_{\varphi_i}^2\mathcal{G}_i^2 + \ell_i^2\mathcal{B}_{\varphi_i}^2)$; see the Appendix).

This partition is an intuitive description for the fact that these three strategies have certain schedulability. Besides,

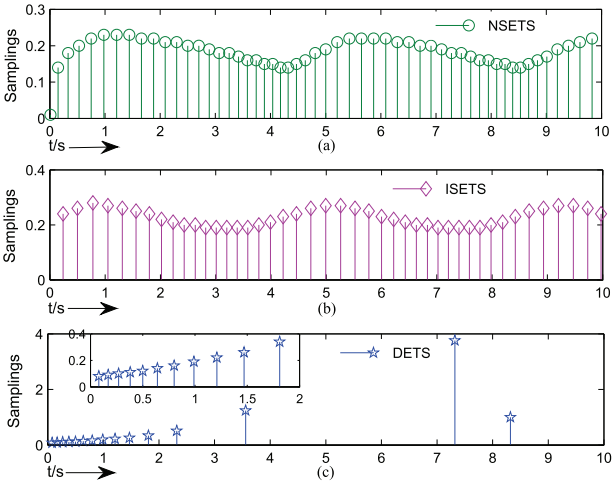


Fig. 7. Triggering comparison of three event strategies.

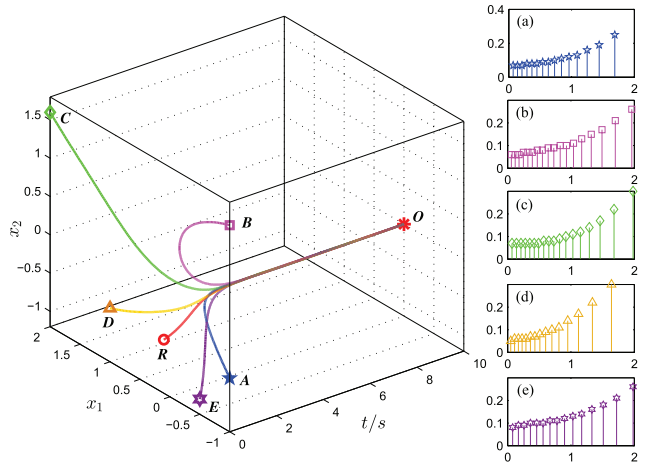


Fig. 8. Stabilizing processes with different initial points.

using this information, practitioners can tell where the parameter selections are concentrated and how they evolve.

B. Triggering Results During the Testing Stage

1) Testing Results of DETS: With the converged critic weights, we transfer to the testing phase, and the DETS-based control performance is given in Fig. 6. The triggering comparison for three event strategies is given in Fig. 7. It is obvious that convergence processes presented by four subgraphs are consistent. It is emphasized that the dynamic signal $\eta_x(t)$ is indeed restricted by an exponential signal $\eta_x^0 e^{-(\lambda+(1/\beta))t}$ [see Fig. 6(d)]. Also, note that the triggering process in Fig. 6(b) is consistent with the sampling recorded in Fig. 7(c).

Next, looking at Fig. 7, for the same control task, NSETS requires 54 samples, and ISETS requires 44 samples, while DETS only requires 15, which means that the controller only needs to calculate 15 times and the control loop conducts 15 transmissions. In addition, observing Fig. 7(c) again, it is noted that control actions are concentrated in the initial phase, and control updatings also reduce as the system gradually stabilizes. In contrast, NSETS and ISETS cannot achieve this. This exactly shows that DETS can adjust its triggering more efficiently.

2) Testing Results of Different Initial Points: At the end, we investigate the DETS-based control performances with different starting points. Our initial point is $R=(0.1,-0.5)$, and the common ending point is $O=(0,0)$. The five comparative points are randomly selected as

$$A=(-1,-0.5),\ B=(-1,1.5),\ C=(2,1.6)$$
$$D=(1,-0.5),\ E=(-0.5,-1).$$

The stabilization process for these five points is exhibited by Fig. 8, and its five subgraphs show respective triggering interevent periods. It can be observed that DETS can achieve stable event control for all starting points, and meanwhile, more events are concentrated in the initial control stage. These results to some extent show that this dynamic design does not depend on specific initial states.

Moreover, the proposed DETS learning method also have advantages over other reported methods, such as identifier-based ADP (IbADP) [11] and online iterative learning (OIL) [12]. When the learning process is implemented under the same settings, with regard to learning accuracy, DETS and OIL are very close. In contrast, IbADP performs worse due to the extra identifying error. It is, therefore, concluded that the proposed DETS can maintain appreciable learning performance with lower communication and computation.

VI. Conclusion

For the two-player NZSDG problem, this article studied event-based control methods when players learn their optimal cost functions. By introducing an EG and two ZOHs, an event learning-based system was built, in which control policies were intermittently updated. Three triggering rules were sequentially devised: NSETS, ISETS, and DETS, where DETS was a novel dynamic triggering strategy. This strategy was termed *dynamic* because of a defined dynamic variable, and it can naturally achieve Zeno-free by incorporating an exponential term. Comparative simulation results demonstrated that these three strategies closely learned the ideal critic weights while ensuring the asymptotic stability, and DETS was apparently favored because of the least samples and lowest computational burden. In the future, we may be consider unknown dynamics or other learning algorithms.

Appendix
Proof Of Theorem 1

Here, we elaborate how to achieve the transition from (29)–(30). Considering that the system dynamics will jump when an event occurs, this proof will be provided in two aspects.

1) Events are not triggered, namely, $\forall t \in [\tau_s, \tau_{s+1})$.

Calculating the time derivative for every component of (29) obtains

$$\dot{L}_{11}(t) = \left(\nabla J_1^*\right)^\top \left(f(x)+g_1(x)\hat{u}_1(x_s)+g_2(x)\hat{u}_2(x_s)\right) + \left(\nabla J_2^*\right)^\top \left(f(x)+g_1(x)\hat{u}_1(x_s)+g_2(x)\hat{u}_2(x_s)\right) \tag{48}$$

$$\dot{L}_{12}(t) = 0 \tag{49}$$

$$\dot{L}_{13}(t) = -\alpha_1 \tilde{w}_1^\top \tilde{\sigma}_1 \tilde{\sigma}_1^\top \tilde{w}_1 + \alpha_1 \frac{\tilde{w}_1^\top \tilde{\sigma}_1}{\bar{\sigma}_1} \mathscr{E}_{H_1} \tag{50}$$

$$\dot{L}_{14}(t) = -\alpha_2 \tilde{w}_2^\top \tilde{\sigma}_2 \tilde{\sigma}_2^\top \tilde{w}_2 + \alpha_2 \frac{\tilde{w}_2^\top \tilde{\sigma}_2}{\bar{\sigma}_2} \mathscr{E}_{H_2}. \tag{51}$$

First, for the term (48), one makes the transformation with (5), and it yields

$$\begin{aligned}\dot{L}_{11}(t) &= \underbrace{\left(\nabla J_1^*\right)^\top f(x) - 2u_1^*(x) R_{11} \hat{u}_1(x_s) + \left(\nabla J_1^*\right)^\top g_2(x) \hat{u}_2(x_s)}_{\dot{L}_{11}^1(t)} \\ &\quad + \underbrace{\left(\nabla J_2^*\right)^\top f(x) - 2u_2^*(x) R_{22} \hat{u}_2(x_s) + \left(\nabla J_2^*\right)^\top g_1(x) \hat{u}_1(x_s)}_{\dot{L}_{11}^2(t)}.\end{aligned} \tag{52}$$

For the sake of clarity, we analyze $\dot{L}_{11}^1(t)$ individually, and the transformation of $\dot{L}_{11}^2(t)$ is analogous. With (5a) and (6a), $(\nabla J_1^*)^\top f(x)$ can be easily derived. After plugging it into $\dot{L}_{11}^1(t)$, then one can get

$$\begin{aligned}\dot{L}_{11}^1(t) &= -x^\top Q_1 x + u_1^{*\top}(x) R_{11} u_1^*(x) - 2u_1^*(x) R_{11} \hat{u}_1(x_s) \\ &\quad - u_2^{*\top}(x) R_{12} u_2^*(x) - \left(\nabla J_1^*\right)^\top g_2(x) \left(u_2^*(x) - \hat{u}_2(\hat{x}_s)\right) \\ &\le -x^\top Q_1 x + \left(u_1^*(x) - u_1(x_s)\right)^\top R_{11} \left(u_1^*(x) - \hat{u}_1(x_s)\right) \\ &\quad - \hat{u}_1^\top(x_s) R_{11} \hat{u}_1(x_s) - \left(\nabla J_1^*\right)^\top g_2(x) \left(u_2^*(x) - \hat{u}_2(x_s)\right).\end{aligned} \tag{53}$$

Note that $\Phi_1(x) = (1/2)\phi_1(x)\phi_1^\top(x)$ with the coupling term $\phi_1(x)$ being $\phi_1(x) = (\nabla J_1^*)^\top g_2(x) \in \mathbb{R}^{1\times l}$; by using the Cauchy inequality, one can derive

$$\begin{aligned}&-\phi_1(x)\left(u_2^*(x) - \hat{u}_2(x_s)\right) \\ &\quad \le \Phi_1(x) + \frac{1}{2}\left(u_2^*(x) - \hat{u}_2(x_s)\right)^\top \left(u_2^*(x) - \hat{u}_2(x_s)\right).\end{aligned} \tag{54}$$

Then, by denoting $r_1 = \lambda_{\min}(R_{11}), \mathscr{R}_1 = \lambda_{\max}(R_{11})$ and combining (54), $\dot{L}_{11}^1(t)$ in (53) can be rewritten as

$$\begin{aligned}\dot{L}_{11}^1(t) \le -x^\top Q_1 x + \mathscr{R}_1^2 \|u_1^*(x) - \hat{u}_1(x_s)\|^2 - r_1^2 \|\hat{u}_1(x_s)\|^2 \\ + \frac{1}{2}\|u_2^*(x) - \hat{u}_2(x_s)\|^2 + \Phi_1(x).\end{aligned} \tag{55}$$

Similarly, let $r_2 = \lambda_{\min}(R_{22}), \mathscr{R}_2 = \lambda_{\max}(R_{22})$ and the coupling be $\phi_2(x) = (\nabla J_2^*)^\top g_1(x) \in \mathbb{R}^{1\times q}$, $\Phi_2(x) = (1/2)\phi_2(x)\phi_2^\top(x)$, and then, (52) can be initially evolved as

$$\begin{aligned}\dot{L}_{11}(t) \le -x^\top \mathcal{Q} x + \left(\mathscr{R}_1^2 + \frac{1}{2}\right)\|u_1^*(x) - \hat{u}_1(x_s)\|^2 + \Phi_1(x) \\ + \left(\mathscr{R}_2^2 + \frac{1}{2}\right)\|u_2^*(x) - \hat{u}_2(x_s)\|^2 + \Phi_2(x) \\ - r_1^2 \|\hat{u}_1(x_s)\|^2 - r_2^2 \|\hat{u}_2(x_s)\|^2.\end{aligned} \tag{56}$$

Next, every component of (56) is separately analyzed. By using the relation $\tilde{w}_1 = w_1 - \hat{w}_1$ and the inequality $\|x\text{-}y\|^2 \le 2\|x\|^2 + 2\|y\|^2$, one can derive

$$\begin{aligned}&\left(\mathscr{R}_1^2 + \frac{1}{2}\right)\|u_1^*(x) - \hat{u}_1(x_s)\|^2 \\ &\quad \le \frac{1}{2}\varrho_1 \left\|\left(g_1^\top(x_s)\nabla\varphi_1^\top(x_s) - g_1^\top(x)\nabla\varphi_1^\top(x)\right)\hat{w}_1\right\|^2 \\ &\qquad + \frac{1}{2}\varrho_1 \|g_1^\top(x)\nabla\varphi_1^\top(x)\tilde{w}_1 + g_1^\top(x)\nabla\varsigma_1(x)\|^2\end{aligned} \tag{57}$$

with $\varrho_1 = (\mathscr{R}_1^2 + 1/2)\|R_{11}^{-1}\|^2$. For the first term of (57), it can further yield

$$\begin{aligned}&\frac{1}{2}\left\|\left(g_1^\top(x_s)\nabla\varphi_1^\top(x_s) - g_1^\top(x)\nabla\varphi_1^\top(x)\right)\right\|^2 \\ &\quad = \frac{1}{2}\left\|\left(\nabla\varphi_1^\top(x_s) - \nabla\varphi_1^\top(x)\right)g_1(x_s) + \nabla\varphi_1^\top(x)\left(g_1(x_s) - g_1(x)\right)\right\|^2 \\ &\quad \le \left\|\left(\nabla\varphi_1^\top(x_s) - \nabla\varphi_1^\top(x)\right)g_1(x_s)\right\|^2 + \left\|\nabla\varphi_1^\top(x)\left(g_1(x_s) - g_1(x)\right)\right\|^2 \\ &\quad \le \left(\mathscr{L}_{\varphi_1}^2 \mathcal{G}_1^2 + \ell_1^2 \mathcal{B}_{\varphi_1}^2\right)\|e_s(t)\|^2.\end{aligned} \tag{58}$$

Then, recalling Assumption 2, (57) can be expressed as

$$\begin{aligned}\left(\mathscr{R}_1^2 + \frac{1}{2}\right)\|u_1^*(x) - \hat{u}_1(x_s)\|^2 \le \mathcal{L}_1 \|\hat{w}_1\|^2 \|e_s(t)\|^2 \\ + \varrho_1 \mathcal{G}_1^2 \mathcal{B}_{\varphi_1}^2 \|\tilde{w}_1\|^2 + \varrho_1 \mathcal{G}_1^2 \mathcal{B}_{\varsigma_1}^2\end{aligned} \tag{59}$$

where $\mathcal{L}_1 = \varrho_1(\mathscr{L}_{\varphi_1}^2 \mathcal{G}_1^2 + \ell_1^2 \mathcal{B}_{\varphi_1}^2)$. For the third term of (56), using NN expression (10), it obtains

$$\begin{aligned}\Phi_1(x) &= \frac{1}{2}\left(\nabla J_1^*\right)^\top g_2(x) g_2^\top(x) \nabla J_1^* \\ &= \frac{1}{2}\left(w_1^\top \nabla\varphi_1 + \nabla\varsigma_1^\top\right) g_2(x) g_2^\top(x) \left(\nabla\varphi_1^\top w_1 + \nabla\varsigma_1\right).\end{aligned} \tag{60}$$

At this point, note the boundedness of the relevant variables in Assumptions 1 and 2, and the result $\Phi_1(x) \le \Phi_{1m}$ is obvious. Similarly, $\Phi_2(x) \le \Phi_{2m}$ can also be obtained.

Taking similar operations for $\|u_2^*(x) - \hat{u}_2(x_s)\|^2$, (52) can be finally transformed into

$$\begin{aligned}\dot{L}_{11}(t) \le -x^\top \mathcal{Q} x - r_1^2 \|\hat{u}_1(x_s)\|^2 - r_2^2 \|\hat{u}_2(x_s)\|^2 + \mathfrak{B}_1 \|\tilde{w}_1\|^2 + \mathfrak{B}_m \\ + \left(\mathcal{L}_1 \|\hat{w}_1\|^2 + \mathcal{L}_2 \|\hat{w}_2\|^2\right)\|e_s(t)\|^2 + \mathfrak{B}_2 \|\tilde{w}_2\|^2\end{aligned} \tag{61}$$

with $\mathfrak{B}_i = \varrho_i \mathcal{G}_i^2 \mathcal{B}_{\varphi_i}^2$, $\mathfrak{B}_m = \sum_{i=1}^2 \varrho_i \mathcal{G}_i^2 \mathcal{B}_{\varsigma_i}^2 + \Phi_{1m} + \Phi_{2m}$.

Second, we continue to analyze (50) and (51). With $\bar{\sigma}_1 \ge 1$, $\dot{L}_{13}(t)$ can be deduced as

$$\dot{L}_{13}(t) \le -\alpha_1 \tilde{w}_1^\top \tilde{\sigma}_1 \tilde{\sigma}_1^\top \tilde{w}_1 + \alpha_1 \tilde{w}_1^\top \tilde{\sigma}_1 \mathscr{E}_{H_1}. \tag{62}$$

Applying Young's inequality to the term $\tilde{w}_1^\top \tilde{\sigma}_1 \mathscr{E}_{H_1}$ obtains

$$\dot{L}_{13}(t) \le -\frac{1}{2}\alpha_1 \lambda_{\min}\left(\tilde{\sigma}_1 \tilde{\sigma}_1^\top\right)\|\tilde{w}_1\|^2 + \frac{1}{2}\alpha_1 \mathcal{E}_1^2. \tag{63}$$

Similarly, for $\dot{L}_{14}(t)$, it has

$$\dot{L}_{14}(t) \le -\frac{1}{2}\alpha_2 \lambda_{\min}\left(\tilde{\sigma}_2 \tilde{\sigma}_2^\top\right)\|\tilde{w}_2\|^2 + \frac{1}{2}\alpha_2 \mathcal{E}_2^2. \tag{64}$$

Finally, by merging (61)–(64) and noting the expressions (24)–(25), the time derivative of $L_1(t)$ can be evolved into

$$\begin{aligned}\dot{L}_1(t) \le -(1-\theta)x^\top \mathcal{Q} x - \mathcal{U}(x_s) + \mathcal{L}_\Sigma \|e_s(t)\|^2 + \frac{1}{2}\mathfrak{B}_\Sigma + \frac{1}{2}\mathfrak{B}_\Sigma \\ - \sum_{i=1}^2 \left[\left(\frac{1}{2}\alpha_i \lambda_{\min}\left(\tilde{\sigma}_i \tilde{\sigma}_i^\top\right) - \mathfrak{B}_i\right)\|\tilde{w}_i\|^2\right] - \theta x^\top \mathcal{Q} x\end{aligned} \tag{65}$$

where $\mathfrak{B}_\Sigma = \mathfrak{B}_m + (1/2)\alpha_1 \mathcal{E}_1^2 + (1/2)\alpha_2 \mathcal{E}_2^2$. At this point, if we let

$$-(1-\theta)x^\top \mathcal{Q} x - \mathcal{U}(x_s) + \mathcal{L}_\Sigma \|e_s(t)\|^2 \le 0 \tag{66}$$

$$\frac{1}{2}\mathfrak{B}_\Sigma - \left(\frac{1}{2}\alpha_i \lambda_{\min}\left(\tilde{\sigma}_i \tilde{\sigma}_i^\top\right) - \mathfrak{B}_i\right)\|\tilde{w}_i\|^2 \le 0 \tag{67}$$

then the result in (30) can be obtained. It is evident that condition (66) indicates $(1-\theta)x^\top \mathcal{Q}x+\mathcal{U}(x_s)-\mathcal{L}_\Sigma \|e_s(t)\|^2$ is nonnegative. Mathematically, this condition corresponds to the triggering rule (23). As for (67), it obviously corresponds to the inequality in (28).

In addition, the parameter θ introduces the term $-\theta x^\top \mathcal{Q}x$, which can be considered as a conservative design to keep the derivative term $\dot{L}_1(t)$ negative.

2) Events are triggered, namely, $t=\tau_{s+1}$ and $s \in \mathbb{N}_0^+$.

In this case, one needs to pay attention to the time point τ_{s+1}^- because the next event has been triggered when $t=\tau_{s+1}$, and thus, the pivotal change is reflected in $\tau_{s+1}^- \to \tau_{s+1}^+$. Consequently, the difference of $L_1(t)$ can be calculated as

$$\begin{aligned}\Delta L_1(t) &= \Delta L_1\big(\tau_{s+1}^+\big)-\Delta L_1\big(\tau_{s+1}^-\big)\\ &= \Delta L_{11}(t)+\Delta L_{12}(t)+\Delta L_{13}(t)+\Delta L_{14}(t) \quad (68)\end{aligned}$$

with

$$\begin{aligned}\Delta L_{11}(t) &= J_1^*(x_{s+1})-J_1^*\big(x\big(\tau_{s+1}^-\big)\big)+J_2^*(x_{s+1})-J_2^*\big(x\big(\tau_{s+1}^-\big)\big)\\ \Delta L_{12}(t) &= J_1^*(x_{s+1})-J_1^*(x_s)+J_2^*(x_{s+1})-J_2^*(x_s)\\ &\quad \Delta L_{13}(t)+\Delta L_{14}(t)\\ &= \frac{1}{2}\big[\tilde{w}_1^\top(\tau_{s+1})\tilde{w}_1(\tau_{s+1})-\tilde{w}_1^\top\big(\tau_{s+1}^-\big)\tilde{w}_1\big(\tau_{s+1}^-\big)\big]\\ &\quad +\frac{1}{2}\big[\tilde{w}_2^\top(\tau_{s+1})\tilde{w}_2(\tau_{s+1})-\tilde{w}_2^\top\big(\tau_{s+1}^-\big)\tilde{w}_2\big(\tau_{s+1}^-\big)\big].\end{aligned}$$

From (23) and (28), it can be concluded that $\dot{L}_1(t)<0$, and the system state is asymptotically stable. Also, note that the state x and weight signals $\hat{w}_1$ and $\hat{w}_2$ are continuous during the triggering process. Therefore, for $\forall t=\tau_{s+1}$, one can, thus, derive that $J_i^*(x_{s+1}) \le J_i^*(x(\tau_{s+1}^-))$; $i=1,2$, and $(1/2)\tilde{w}_i^\top(\tau_{s+1})\tilde{w}_i(\tau_{s+1}) \le (1/2)\tilde{w}_i^\top(\tau_{s+1}^-)\tilde{w}_i(\tau_{s+1}^-)$, and accordingly, the difference ΔL_1 is further deduced as

$$\Delta L_1(t) \le \Delta L_{12}(t) = -\mathcal{K}_1(\|e_{s+1}(\tau_s)\|)-\mathcal{K}_2(\|e_{s+1}(\tau_s)\|)$$

where $\mathcal{K}_1(\cdot)$ and $\mathcal{K}_2(\cdot)$ are class-$\mathcal{K}$, and $e_{s+1}(\tau_s)=x_{s+1}-x_s$. This result indicates that the Lyapunov candidate (29) is also decreasing at every instant $\tau_{s+1}, s \in \mathbb{N}_0^+$.

In the end, based on the analysis for two aspects, if (23) and (28) are satisfied, then the system is asymptotically stable, and two weight errors are UUB. This proof is, thus, completed.

REFERENCES

[1] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2042–2062, Jun. 2018.

[2] X.-M. Zhang, Q.-L. Han, and B.-L. Zhang, "An overview and deep investigation on sampled-data-based event-triggered control and filtering for networked systems," *IEEE Trans. Ind. Informat.*, vol. 13, no. 1, pp. 4–16, Feb. 2017.

[3] W. Gao, Z.-P. Jiang, F. L. Lewis, and Y. Wang, "Leader-to-formation stability of multiagent systems: An adaptive optimal control approach," *IEEE Trans. Autom. Control*, vol. 63, no. 10, pp. 3581–3587, Oct. 2018.

[4] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*. Philadelphia, PA, USA: SIAM, 1998.

[5] J. Engwerda, *LQ Dynamic Optimization and Differential Games*. New York, NY, USA: Wiley, 2005.

[6] D. Vrabie and F. Lewis, "Integral reinforcement learning for online computation of feedback Nash strategies of nonzero-sum differential games," in *Proc. 49th IEEE Conf. Decis. Control (CDC)*, Dec. 2010, pp. 3066–3071.

[7] M. Johnson, R. Kamalapurkar, S. Bhasin, and W. E. Dixon, "Approximate N-player nonzero-sum game solution for an uncertain continuous nonlinear system," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1645–1658, Aug. 2015.

[8] H. Zhang, H. Jiang, C. Luo, and G. Xiao, "Discrete-time nonzero-sum games for multiplayer using policy-iteration-based adaptive dynamic programming algorithms," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3331–3340, Oct. 2017.

[9] H. Jiang, H. Zhang, Y. Luo, and J. Han, "Neural-network-based robust control schemes for nonlinear multiplayer systems with uncertainties via adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 3, pp. 579–588, Mar. 2019.

[10] S. Yasini, M. B. Naghibi Sitani, and A. Kirampor, "Reinforcement learning and neural networks for multi-agent nonzero-sum games of nonlinear constrained-input systems," *Int. J. Mach. Learn. Cybern.*, vol. 7, no. 6, pp. 967–980, Dec. 2016.

[11] Y. Lv and X. Ren, "Approximate Nash solutions for multiplayer mixed-zero-sum game with reinforcement learning," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 12, pp. 2739–2750, Dec. 2019.

[12] Q. Zhang and D. Zhao, "Data-based reinforcement learning for nonzero-sum games with unknown drift dynamics," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2874–2885, Aug. 2019.

[13] P. J. Werbos, *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches-Approximate Dynamic Programming for Realtime Control and Neural Modeling*. New York, NY, USA: Van Nostrand, 1992.

[14] F.-Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009.

[15] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Aug. 2009.

[16] C. Mu, K. Wang, Z. Ni, and C. Sun, "Cooperative differential game-based optimal control and its application to power systems," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5169–5179, Aug. 2020.

[17] Y. Li, K. Li, and S. Tong, "Adaptive neural network finite-time control for multi-input and multi-output nonlinear systems with positive powers of odd rational numbers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2532–2543, Aug. 2020.

[18] D. V. Prokhorov, R. A. Santiago, and D. C. Wunsch, "Adaptive critic designs: A case study for neurocontrol," *Neural Netw.*, vol. 8, no. 9, pp. 1367–1372, Jan. 1995.

[19] J. Si and Y.-T. Wang, "Online learning control by association and reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 264–276, Mar. 2001.

[20] Q. Wei, D. Liu, and H. Lin, "Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 840–853, Mar. 2016.

[21] C. Mu, D. Wang, and H. He, "Novel iterative neural dynamic programming for data-based approximate optimal control design," *Automatica*, vol. 81, pp. 240–252, Jul. 2017.

[22] Z. Ni, N. Malla, and X. Zhong, "Prioritizing useful experience replay for heuristic dynamic programming-based learning systems," *IEEE Trans. Cybern.*, vol. 49, no. 11, pp. 3911–3922, Nov. 2019.

[23] K. Esfandiari, F. Abdollahi, and H. A. Talebi, "Adaptive near-optimal neuro controller for continuous-time nonaffine nonlinear systems with constrained input," *Neural Netw.*, vol. 93, pp. 195–204, Sep. 2017.

[24] J. Na and G. Herrmann, "Online adaptive approximate optimal tracking control with simplified dual approximation structure for continuous-time unknown nonlinear systems," *IEEE/CAA J. Autom. Sinica*, vol. 1, no. 4, pp. 412–422, Oct. 2014.

[25] A. Heydari, "Optimal switching of DC–DC power converters using approximate dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 586–596, Mar. 2018.

[26] H. Jiang and H. He, "Data-driven distributed output consensus control for partially observable multiagent systems," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 848–858, Mar. 2019.

[27] Y.-X. Li and G.-H. Yang, "Adaptive neural control of pure-feedback nonlinear systems with event-triggered communications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6242–6251, Dec. 2018.

[28] Y.-F. Gao, X.-M. Sun, C. Wen, and W. Wang, "Estimation of sampling period for stochastic nonlinear sampled-data systems with emulated controllers," *IEEE Trans. Autom. Control*, vol. 62, no. 9, pp. 4713–4718, Sep. 2017.

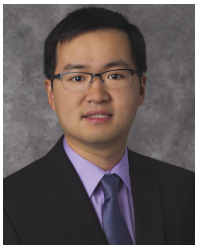
[29] Y.-X. Li and G.-H. Yang, "Observer-based fuzzy adaptive event-triggered control codesign for a class of uncertain nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 3, pp. 1589–1599, Jun. 2018.
[30] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1680–1685, Sep. 2007.
[31] A. Anta and P. Tabuada, "Exploiting isochrony in self-triggered control," *IEEE Trans. Autom. Control*, vol. 57, no. 4, pp. 950–962, Apr. 2012.
[32] M. C. F. Donkers and W. P. M. H. Heemels, "Output-based event-triggered control with guaranteed $\mathcal{L}_\infty$-gain and improved and decentralized event-triggering," *IEEE Trans. Autom. Control*, vol. 57, no. 6, pp. 1362–1376, Jun. 2012.
[33] Y. Fan, Y. Yang, and Y. Zhang, "Sampling-based event-triggered consensus for multi-agent systems," *Neurocomputing*, vol. 191, pp. 141–147, May 2016.
[34] F. Fang and Y. Xiong, "Event-driven-based water level control for nuclear steam generators," *IEEE Trans. Ind. Electron.*, vol. 61, no. 10, pp. 5480–5489, Oct. 2014.
[35] N. Szanto, V. Narayanan, and S. Jagannathan, "Event-sampled direct adaptive NN output- and state-feedback control of uncertain strict-feedback system," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1850–1863, May 2018.
[36] A. Sahoo, H. Xu, and S. Jagannathan, "Approximate optimal control of affine nonlinear continuous-time systems using event-sampled neurodynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 639–652, Mar. 2017.
[37] X. Yang and H. He, "Adaptive critic designs for event-triggered robust control of nonlinear systems with unknown dynamics," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2255–2267, Jun. 2019.
[38] X. Zhong and H. He, "An event-triggered ADP control approach for continuous-time system with unknown internal states," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 683–694, Mar. 2017.
[39] Q. Zhang, D. Zhao, and Y. Zhu, "Event-triggered H_∞ control for continuous-time nonlinear system via concurrent learning," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 7, pp. 1071–1081, Jul. 2017.
[40] A. Girard, "Dynamic triggering mechanisms for event-triggered control," *IEEE Trans. Autom. Control*, vol. 60, no. 7, pp. 1992–1997, Jul. 2015.
[41] X. Ge and Q.-L. Han, "Distributed formation control of networked multi-agent systems using a dynamic event-triggered communication mechanism," *IEEE Trans. Ind. Electron.*, vol. 64, no. 10, pp. 8118–8127, Oct. 2017.
[42] W. Hu, C. Yang, T. Huang, and W. Gui, "A distributed dynamic event-triggered control approach to consensus of linear multiagent systems with directed networks," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 869–874, Feb. 2020.
[43] Q. Li, B. Shen, Z. Wang, T. Huang, and J. Luo, "Synchronization control for a class of discrete time-delay complex dynamical networks: A dynamic event-triggered approach," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1979–1986, May 2019.
[44] D. J. Antunes and B. A. Khashooei, "Consistent dynamic event-triggered policies for linear quadratic control," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1386–1398, Sep. 2018.
[45] C. Mu and K. Wang, "Approximate-optimal control algorithm for constrained zero-sum differential games through event-triggering mechanism," *Nonlinear Dyn.*, vol. 95, no. 4, pp. 2639–2657, Mar. 2019.
[46] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Netw.*, vol. 3, no. 5, pp. 551–560, Jan. 1990.
[47] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
[48] Y. Li, T. Yang, and S. Tong, "Adaptive neural networks finite-time optimal control for a class of nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4451–4460, Nov. 2020.

Chaoxu Mu (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from the School of Automation, Southeast University, Nanjing, China, in 2012.

She was a Visiting Ph.D. Student with the Royal Melbourne Institute of Technology University, Melbourne, VIC, Australia, from 2010 to 2011. She was a Post-Doctoral Fellow with the Department of Electrical, Computer and Biomedical Engineering, The University of Rhode Island, Kingston, RI, USA, from 2014 to 2016. She is currently a Professor with the School of Electrical and Information Engineering, Tianjin University, Tianjin, China. She has authored over 100 journal and conference articles and coauthored two monographs. Her current research interests include nonlinear system control and optimization and adaptive and learning systems.

Ke Wang (Graduate Student Member, IEEE) received the B.S. degree in control technology and instruments from the Shaanxi University of Science and Technology University, Xi'an, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Electrical and Information Engineering, Tianjin University, Tianjin, China.

He took part in the Mathematics competition of Chinese College Students and received the First Prize. He received the Tianjin Research Innovation Project for Postgraduate Students in 2020. His current research interests include reinforcement learning, adaptive control, differential games, and event-triggering.



Zhen Ni (Member, IEEE) received the B.S. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2010, and the Ph.D. degree from the University of Rhode Island, Kingston, RI, USA, in 2015.

He is currently an Assistant Professor with the Department of Computer, Electrical Engineering, and Computer Science, Florida Atlantic University, Boca Raton, FL, USA. He was with the Department of Electrical Engineering and Computer Science, South Dakota State University, Brookings, SD, USA, from 2015 to 2019. His current research interests include artificial intelligence, reinforcement learning, adaptive dynamic programming, and adaptive control.

Dr. Ni is a recipient of the prestigious ORAU Ralph E. Powe Junior Faculty Enhancement Award in 2020, the IEEE Computational Intelligence Society Outstanding Ph.D. Dissertation Award in 2020, and the INNS Aharon Katzir Young Investigator Award in 2019. He is an Associate Editor of the *IEEE Computational Intelligence Magazine* from 2018 to present and the IEEE TRANSACTIONS OF NEURAL NETWORKS AND LEARNING SYSTEMS 2019 to present. He is the Chair for the 2021 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL) under the IEEE Symposium Series on Computational Intelligence (SSCI).