Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Mixed Driven Iterative Adaptive Critic Control Design Towards Nonaffine Discrete-Time Plants ⋆

Ding Wang *,**, **Mingming Ha** ***

* *Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: dingwang@bjut.edu.cn)*
** *Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing University of Technology, Beijing 100124, China*
*** *School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China (e-mail: hamingming_0705@foxmail.com)*

Abstract: In this paper, an effective mixed driven framework is constructed involving both data and event considerations. The primary purpose lies in that the mixed driven iterative adaptive critic method is established to address approximate optimal control towards discrete-time nonlinear dynamics. The neural dynamic programming technique is inventively integrated with the mixed driven architecture, such that the knowledge of the controlled plant is needless and the number for updating control inputs is prominently reduced. A triggering threshold is also designed with theoretical guarantee, which renders that the control signals can be updated conditionally. Through carrying out simulation studies with comparisons, the superiority of the present near-optimal regulation approach is confirmed at last.

Keywords: Discrete-time nonlinear dynamics, iterative adaptive critic algorithm, mixed driven design, neural optimal control, triggering condition.

1. INTRODUCTION

Optimal feedback design is always an important portion of the modern control community. However, unlike the common linear case, the main difficulty in nonlinear optimal control design is addressing the complex Hamilton-Jacobi-Bellman (HJB) equations. Considering the rarity of analytical methods, adaptive critic algorithms combining with neural networks were developed to obtain approximate solutions (Werbos, 1992). Heuristic dynamic programming (HDP), dual HDP (DHP), and globalized DHP (GDHP) were basic implementation tools of the adaptive critic field as described in Werbos (1992). After this pioneering work, the online learning optimal control design was paid great attention particularly in Si and Wang (2001). It is worth noting that the iterative adaptive critic algorithms in discrete-time domain were developed to solve approximate optimal control problems, by adopting HDP (Al-Tamimi, Lewis, & Abu-Khalaf, 2008; Zhang, Liu, Xiao, & Jiang, 2020), DHP (Zhang, Luo, & Liu, 2009) and GDHP (Wang, Liu, Wei, Zhao, & Jin, 2012) techniques, respectively. Note that the idea of data-driven is frequently emphasized in many existing results of the approximate optimal control synthesis. Effective learning from the big data information is an important property of data-driven adaptive critic algorithms. Though possessing excellent self-leaning and adaptivity performances, the resource utilization rate is rarely considered in these traditional time-based adaptive critic algorithms.

Within the event-based control framework, the point of view on how information could be sampled is offered for minimal triggering and actuation. The event-based design is also a hot topic under the network environment. In the last decade, a series of event-driven approaches have been acquired considerable attention within advanced control communities, including robust optimal regulation (Wang & Liu, 2018) and adaptive fault-tolerant control (Fan & Yang, 2018). Unlike the continuous-time case (Wang & Liu, 2018; Fan & Yang, 2018), the event-driven approaches in discrete-time domain were motivated a lot by the work of Eqtami, Dimarogonas, and Kyriakopoulos (2010) and were closely related to control of networked systems (Garcia & Antsaklis, 2014). Recently, event-triggered communication and control of networked systems for multi-agent consensus was surveyed by Nowzari, Garcia, and Cortes (2019). In addition, the event-based algorithms for normal discrete-time systems (Dong, Zhong, Sun, & He, 2017) and input-constrained plants (Ha, Wang, & Liu, 2020) were respectively constructed. However, there are currently no results on data and event driven iterative adaptive critic control design for discrete-time nonaffine systems, which is a difficult problem shown in literatures (Bian, Jiang, & Jiang, 2014; Kiumarsi, Kang, & Lewis, 2016; Wang & Zhong, 2019).

⋆ This work was supported in part by Beijing Natural Science Foundation under Grant JQ19013, in part by the National Natural Science Foundation of China under Grant 61773373 and Grant 61890930-5, and in part by the National Key Research and Development Project under Grant 2018YFC1900800-5.

Under these backgrounds, in this paper, a mixed data and event driven iterative adaptive critic strategy is developed for near-optimal control of discrete-time nonlinear dynamics, where data-driven learning for the complex dynamics and event-based triggering under the network environment are naturally included. This is a novel mixed framework in discrete-time domain along with the idea of iterative adaptive critic algorithms. Combining the neural dynamic programming technique with the mixed driven framework, the knowledge of the controlled plant is needless and the number for updating control inputs is signally reduced when dealing with the near-optimal regulation. In summary, the data and communication resources are both optimized during the control design process.

The notations used in the paper are described as follows. $\mathbb{R}$ is the set of all real numbers. $\mathbb{R}^n$ is the Euclidean space of all n-dimensional real vectors. Let Ω be a compact subset of $\mathbb{R}^n$ and $\Psi(\Omega)$ be the set of admissible control laws (Al-Tamimi, Lewis, & Abu-Khalaf, 2008) on Ω. $\mathbb{R}^{n\times m}$ is the space of all $n \times m$ real matrices. $\|\cdot\|$ derives the vector or matrix norms. $\mathbb{N}$ denotes the set $\{0, 1, 2, \dots\}$. I_n is the $n \times n$ identity matrix and "T" is the transpose operation. Note the symbols k, i, and j are used to represent the time step, the iteration index, and the sampling instant, respectively.

2. PROBLEM FORMULATION

In this paper, we consider nonlinear dynamics with the discrete-time formulation

$$x(k+1) = F(x(k), u(k)), k \in \mathbb{N}, \tag{1}$$

where $F(\cdot,\cdot)$ is a continuous system function, $x(k) \in \mathbb{R}^n$ is the state variable, and $u(k) \in \mathbb{R}^m$ is the control input. We let $x(0)$ be the initial state and assume it is a unique equilibrium point of system (1) under $u = 0$, i.e., $F(0, 0) = 0$. Generally, we also assume that system (1) can be stabilized on the set $\Omega \subset \mathbb{R}^n$ by a state feedback controller $u(x(k))$.

We consider the optimal control problem and want to find a feedback control law $u \in \Psi(\Omega)$ to minimize

$$J(x(k)) = \sum_{h=k}^{\infty} U(x(h), u(x(h))), \tag{2}$$

where $U(x, u) \geq 0, \forall x, u$ is the utility function ensuring $U(0, 0) = 0$. Normally, the utility function is selected as $U(x, u) = x^{\mathsf{T}}Qx + u^{\mathsf{T}}Ru$, where Q and R are both positive definite. Here, the term $x^{\mathsf{T}}Qx$ is called the state utility while $u^{\mathsf{T}}Ru$ is the control utility.

According to the optimality principle, the optimal cost function defined as

$$J^*(x(k)) = \min_{\{u(\cdot)\}} \sum_{h=k}^{\infty} U(x(h), u(x(h))) \tag{3}$$

satisfies the discrete-time HJB equation

$$J^*(x(k)) = \min_{u(x(k))} \big\{U(x(k), u(x(k))) + J^*(x(k+1))\big\}. \tag{4}$$

It is hard to solve the above HJB equation with the traditional manners, because the value of $J^*(x(k+1))$ is unknown in advance and the controlled plant is nonaffine. This difficulty also exists when deriving the exact optimal control by using

$$u^*(x(k)) = \arg \min_{u(x(k))} \big\{U(x(k), u(x(k))) + J^*(x(k+1))\big\}. \tag{5}$$

Therefore, it is necessary to pursue the near-optimal control design in nonlinear discrete-time domain.

For solving the discrete-time HJB equation approximately, the effective iterative form of data-driven adaptive critic design can be employed. Moreover, an event-driven condition should be designed with a positive threshold, so as to reduce the updating times of the control input. In this paper, the mixed driven structure is constituted by the data and event considerations during these two steps. The mixed data and event driven iterative adaptive critic control design will be presented detailedly in the next section.

3. MIXED DRIVEN ITERATIVE ADAPTIVE CRITIC CONTROL DESIGN WITHIN A GENERAL NONAFFINE DISCRETE-TIME DOMAIN

The data-driven iterative adaptive critic method with neural dynamic programming implementation and the event-driven control system design are included in this section.

3.1 Data-Driven Iterative Algorithm With Neural Dynamic Programming Implementation

Here, we describe the iterative learning algorithm step by step. Before carrying out the main iteration process, we should set a small positive number ε and construct two sequences $\{J^{(i)}(x(k))\}$ and $\{u^{(i)}(x(k))\}$, where i denotes the iteration index and $i \in \mathbb{N}$. The iteration process starts with $i = 0$ and the initial cost function is chosen as $J^{(0)}(\cdot) = 0$.

According to the current value of the cost function, the iterative control function is solved by

$$u^{(i)}(x(k)) = \arg \min_{u(x(k))} \big\{U(x(k), u(x(k))) + J^{(i)}(x(k+1))\big\}, \tag{6}$$

where the involved state vector $x(k+1) = F(x(k), u(x(k)))$ can be approximated by using a neural-network-based learning module. Incidentally, note that since $J^{(0)}(\cdot) = 0$, we can easily compute that $u^{(0)}(\cdot) = 0$.

Based on the new control law, the iterative cost function is updated according to

$$J^{(i+1)}(x(k)) = \min_{u(x(k))} \big\{U(x(k), u(x(k))) + J^{(i)}(x(k+1))\big\}. \tag{7}$$

Once the newest cost function is updated, we should check the stopping criterion related to ε and decide whether the next iteration is necessary.

In case that $|J^{(i+1)}(x(k)) - J^{(i)}(x(k))| \leq \varepsilon$, we stop the iteration process and derive the near-optimal control law. Otherwise, we increase the iteration index as $i = i + 1$ and continue to implement the above steps as (6) and (7). In a word, the whole iterative process is carried out according to the sequence $J^{(0)} \to u^{(0)} \to J^{(1)} \to u^{(1)} \to \cdots$. The convergence of this iterative algorithm is reflected by considering two aspects, namely boundedness and monotonicity.

In the sequel, the detailed learning procedure of the above iterative algorithm is presented via the neural dynamic programming technique. This is a data-driven learning control process containing the approximate state $\hat{x}(k+1)$, the approximate cost $\hat{J}^{(i)}(x(k))$, and the approximate control $\hat{u}^{(i)}(x(k))$, which are just the outputs of three neural networks.

With a neural identifier, a critic network, and an action network, the implementation diagram of the data-driven iterative algorithm in discrete-time domain is displayed in Fig. 1. Note that the blue dashed line represents the backpropagating path of the involved neural networks.

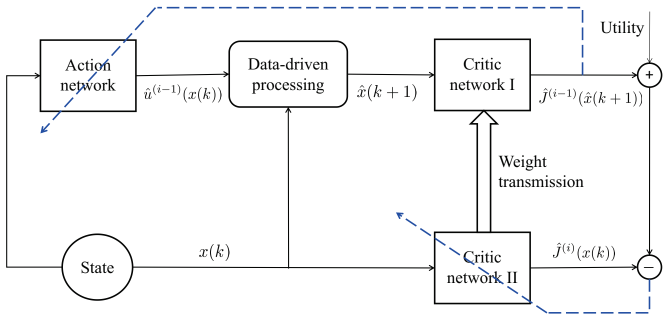

Fig. 1. Implementation diagram of the data-driven iterative algorithm.

For learning the nonlinear system dynamics, a neural network identifier is first constructed via data-driven processing. By inputting the state $x(k)$ and the control $\hat{u}^{(i-1)}(x(k))$, we can express the output of the neural identifier as

$$\hat{x}(k+1) = \omega_1^{\mathsf{T}}\sigma\big(\nu_1^{\mathsf{T}}\big[x^{\mathsf{T}}(k), \hat{u}^{(i-1)\mathsf{T}}(x(k))\big]^{\mathsf{T}}\big), \tag{8}$$

where ω_1 and ν_1 are the involved weight variables and $\sigma(\cdot)$ is the activation function. Combining with the updated state $x(k+1)$, the training performance measure of the neural identifier is defined as

$$\mathcal{E}_1(k) = \frac{1}{2}\big[\hat{x}(k+1) - x(k+1)\big]^{\mathsf{T}}\big[\hat{x}(k+1) - x(k+1)\big]. \tag{9}$$

Establishing the neural identifier is a pre-training procedure that should be conducted before the main iteration of critic and action networks.

The critic network approximates the iterative cost function with weight matrices ω_2 and ν_2 and the formulation

$$\hat{J}^{(i)}(x(k)) = \omega_2^{(i)\mathsf{T}}\sigma\big(\nu_2^{(i)\mathsf{T}}x(k)\big). \tag{10}$$

Combining with (7) of the i-th iteration, the training performance measure is

$$\mathcal{E}_2(k) = \frac{1}{2}\big[\hat{J}^{(i)}(x(k)) - J^{(i)}(x(k))\big]^2. \tag{11}$$

Note that the iteration index i is omitted in $\mathcal{E}_2(k)$ for simplicity. Actually, this performance measure is varied along with different iteration numbers.

Using the state variable $x(k)$ and the weight variables ω_3 and ν_3, the action neural network approximates the iterative control law as follows:

$$\hat{u}^{(i-1)}(x(k)) = \omega_3^{(i-1)\mathsf{T}}\sigma\big(\nu_3^{(i-1)\mathsf{T}}x(k)\big). \tag{12}$$

The performance measure for tuning action parameters is

$$\mathcal{E}_3(k) = \frac{1}{2}[\hat{J}^{(i-1)}(\hat{x}(k+1)) - \mathcal{V}_0]^2, \tag{13}$$

where $\mathcal{V}_0$ is always set as zero in light of Si and Wang (2001) and the iteration index i is also omitted in $\mathcal{E}_3(k)$. It is important to note that the action training error is defined as $\hat{J}^{(i-1)}(\hat{x}(k+1)) - \mathcal{V}_0$ within the neural dynamic programming formulation. This is quite different from the training strategy given in (Zhang, Luo, & Liu, 2009; Wang, Liu, Wei, Zhao, & Jin, 2012), where the vector error between the iterative controller and the optimal controller is considered. By virtue of this manner, the direct dependence on the control matrix is removed, which is very significant to address the optimal feedback control design of nonaffine dynamics.

By adopting the gradient-based adaptation rule, the weight matrices of the neural identifier, the critic network, and the action network can be updated with a unified criterion

$$\Delta\omega_l = -\alpha_l\left(\frac{\partial\mathcal{E}_l(k)}{\partial\omega_l}\right), \tag{14a}$$

$$\Delta\nu_l = -\alpha_l\left(\frac{\partial\mathcal{E}_l(k)}{\partial\nu_l}\right), l = 1, 2, 3, \tag{14b}$$

where $\alpha_l > 0, l = 1, 2, 3$ are the learning rates of the three networks and $\Delta\omega_l$ and $\Delta\nu_l$ with $l = 1, 2, 3$ are the difference values of two orderly updating steps.

After carrying out the data-driven algorithm with sufficient iteration steps, the practical near-optimal control law written as $\hat{u}^*(x(k))$ is derived. If we directly apply the near-optimal control law to the original plant, the control input may update frequently. In other words, relying on the time-based manner always brings in an obvious waste of the communication resource. In the sequel, we focus on designing an effective triggering condition and study how the feedback control system becomes under the event-driven manner.

3.2 Event-Driven Control System Design With A Proper Triggering Condition

The background of the event-driven design is provided here by defining a monotonically increasing sequence $\{s_j\}_{j=0}^{\infty}$, where $j \in \mathbb{N}$. The event-driven control signal is only updated at the sampling instants $s_0, s_1, s_2, \ldots$. Then, the general feedback control law $u(x(k))$ can be denoted as $u(x(s_j))$ in this part, where $x(s_j)$ is the sampled state at the time instant $k = s_j, k \in [s_j, s_{j+1}), j \in \mathbb{N}$. In this paper, after the data-driven learning stage, the practical control law, i.e., $\hat{u}^*(x(k))$, with event-driven consideration can be written as $\hat{u}^*(x(s_j))$. A zero-order-hold is always introduced to keep the event-driven control input at the instant $k = s_j$, until the next event occurs. The event-driven error is defined as

$$e(k) = x(s_j) - x(k), k \in [s_j, s_{j+1}), j \in \mathbb{N}, \tag{15}$$

which means $x(s_j) = x(k) + e(k)$, so that the feedback control is rewritten as $\hat{u}^*(x(s_j)) = \hat{u}^*(x(k) + e(k))$. Via the function of this control law, we have

$$x(k+1) = F(x(k), \hat{u}^*(x(k) + e(k))), k \in \mathbb{N}, \tag{16}$$

which is the closed-loop form of system (1) with event-based consideration.

The following statements proposed in Eqtami, Dimarogonas, and Kyriakopoulos (2010) are basic conditions used for coping with the discrete-time event-driven design. The formulas $\|e(k)\| \leq \|x(k)\|$ and $\|x(k+1)\| \leq \pi\|x(k)\| + \pi\|e(k)\|$ are assumed to be true, where $x(k+1)$ is given by (16) and the positive constant $\pi \in (0, 0.5)$.

We now analyze how to derive an applicable triggering threshold. Note the triggering error formula implies that

$$\begin{aligned}\|x(k-1)\| &= \|x(s_j) - e(k-1)\| \\ &\leq \|x(s_j)\| + \|e(k-1)\|. \end{aligned} \tag{17}$$

Observing the proposed assumption and the inequality (17), we consider $\|e(k)\| \leq \|x(k)\|$ and further find that

$$\begin{aligned}\|e(k)\| &\leq \pi\|x(k-1)\| + \pi\|e(k-1)\| \\ &\leq \pi(\|x(s_j)\| + \|e(k-1)\|) + \pi\|e(k-1)\| \\ &= 2\pi\|e(k-1)\| + \pi\|x(s_j)\|. \end{aligned} \tag{18}$$

Combining (18) with the fact that $e(s_j) = 0$, we derive that

$$\begin{aligned}\|e(k)\| &\leq 2\pi(2\pi\|e(k-2)\| + \pi\|x(s_j)\|) + \pi\|x(s_j)\| \\ &\leq \cdots \\ &\leq \pi\|x(s_j)\|\big[1 + 2\pi + \cdots + (2\pi)^{k-s_j-1}\big]. \end{aligned} \tag{19}$$

Hence, we can develop a triggering condition formed as $\|e(k)\| \leq \bar{e}(\pi)$, where the threshold is

$$\bar{e}(\pi) = \frac{1-(2\pi)^{k-s_j}}{1-2\pi}\pi\|x(s_j)\|, \pi \in (0, 0.5). \tag{20}$$

The above inequality is a practical triggering condition of the event-driven design. Note that here, the threshold $\bar{e}$ is a function of the constant π. How the triggering control performance is affected by the choice of the constant is a key issue that should be verified, where a special attention should be paid in terms of reducing control updating times.

The design diagram of the above event-driven control system is given in Fig. 2, where the blue dashed line denotes the implementation path in the next time step. The triggering condition therein is actually worked as a switch and the case of control updating is depicted. When the threshold is not violated, the control signal is kept unchanged until the violation occurs. Hence, there always exists a stair-stepping control curve during the event-driven design.

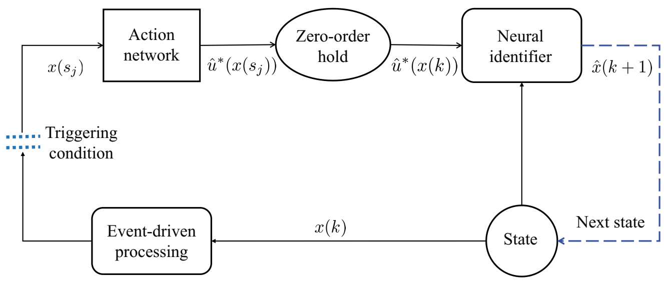

Fig. 2. Design diagram of the event-driven control system.

It is worth mentioning that in Fig. 2, the weight variables of the action network and the neural identifier are determined after the previous iterative learning activity. Therefore, the data and event driven processes are closely related to each other in the whole mixed driven framework.

4. SIMULATION STUDIES

In this section, we apply the mixed driven iterative adaptive critic approach to a specified nonaffine system, in order to verify the near-optimal control performance. We consider a discrete-time nonlinear plant formulated as

$$\begin{aligned} x(k+1) = &\begin{bmatrix} 0.3x_2(k) - 0.5\cos(x_2(k))\sin(0.6x_1(k)) \\ -0.1x_1(k) + x_2(k) + 0.1x_2^2(k) \end{bmatrix} \\ &+ \begin{bmatrix} x_1(k)\tanh(u(k)) \\ 0.8u^3(k) \end{bmatrix}, \end{aligned} \tag{21}$$

where the involved state vector is $x(k) = [x_1(k), x_2(k)]^\mathsf{T}$ and the control variable is $u(k)$. In order to handle the approximate optimal regulation, the utility term of the cost function is selected with $Q = 0.2I_2, R = I$ and then three neural networks are constructed within the proposed mixed driven framework.

We first train the neural identifier with an architecture of 3–8–2 (number of the input, hidden, and output layers) by choosing the learning rate as $\alpha_1 = 0.2$ and employing the rule (14). After a data-driven learning stage with a randomly initial choice in $[-0.1, 0.1]$, the weight variables finally converge to two constant matrices, which can be employed to update the system states of subsequent time steps. Hence, this replaces the usage of the controlled plant during the following main training process.

Then, we determine the structures of the critic and action networks as 2–8–1 and 2–8–1, respectively and train them according to (14). During the learning process, we choose the initial state as $x(0) = [1, -1]^\mathsf{T}$ and set the learning rates as $\alpha_2 = \alpha_3 = 0.2$. Similar as the neural identifier, the initial weights of critic and action networks are both chosen randomly in $[-0.1, 0.1]$. In this situation, we employ the iterative HDP algorithm for 28 iterations and then the prespecified accuracy $\varepsilon = 10^{-6}$ is reached, where 2000 training times are involved for each iteration. Here, the convergence curve of the iterative cost function with the specified initial state is shown in Fig. 3. Besides, the convergence trends of the 2-norm calculation of critic and action weights are presented in Fig. 4, where subgraphs (a) and (b) are for the critic network while (c) and (d) are for the action network. Since the given norms of these weight matrices are convergent, we know that all the weight elements are convergent during the iteration process.

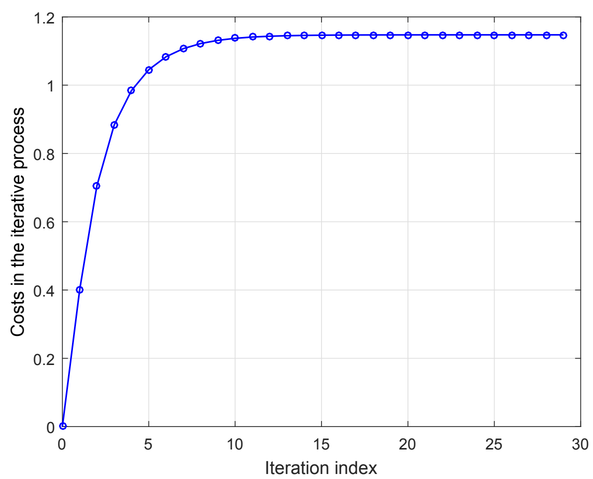

Fig. 3. Convergence curve of the iterative cost function.

Next, for turning to the event-driven design, we severally let $\pi = 0.2$, $\pi = 0.1$, and $\pi = 0.3$, so that specify the

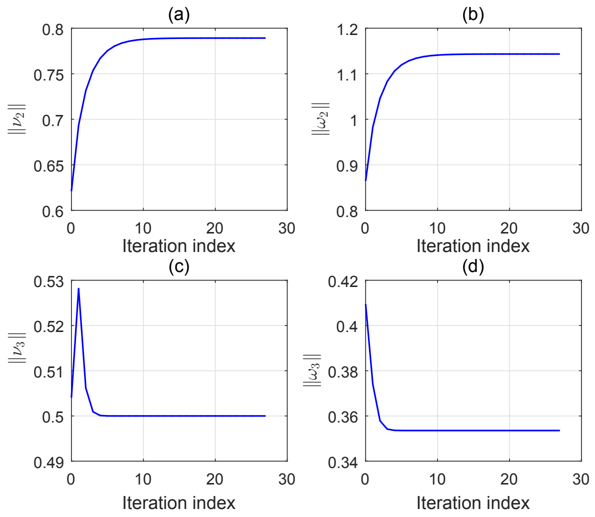

Fig. 4. Convergence trends of the 2-norm calculation of weight matrices. (a) Input-hidden weight of the critic network. (b) Hidden-output weight of the critic network. (c) Input-hidden weight of the action network. (d) Hidden-output weight of the action network.

triggering threshold (20) as the following cases:

$$\bar{e}(0.2) = \frac{1 - 0.4^{k-s_j}}{3}\|x(s_j)\|, \tag{22a}$$

$$\bar{e}(0.1) = \frac{1 - 0.2^{k-s_j}}{8}\|x(s_j)\|, \tag{22b}$$

$$\bar{e}(0.3) = \frac{3(1 - 0.6^{k-s_j})}{4}\|x(s_j)\|. \tag{22c}$$

In the sequel, several case studies are implemented based on the learnt weights of the iterative HDP algorithm for 300 time steps. In Case 1A, we apply the adaptive critic controller $\hat{u}^*(x(k))$ with $\pi = 0.2$ and the event-driven threshold being selected as (22a). In Case 2, we revisit the traditional time-based controller design method like in Zhang, Luo, and Liu (2009). Using the corresponding controllers, the state trajectories of two case studies are given in Fig. 5. Note the state responses therein are almost the same. Besides, the triggering threshold related to Case 1A is depicted in Fig. 6. Remarkably, the control trajectories of the two cases are illustrated in Fig. 7, where a stair-stepping curve is clearly observed, just affected by the event-based mechanism.

At last, for checking the event-driven control performance and updating times along with the variation of the threshold, we severally set $\pi = 0.1$ (Case 1B) and $\pi = 0.3$ (Case 1C) and conduct other two case studies. To this end, we change the corresponding thresholds to (22b) and (22c) and compare the obtained control curves with the time-based case of Case 2 via Figs. 8 and 9, respectively. It is observed from Figs. 7, 8, and 9 that the stair-stepping phenomenon of the event-driven control curves become more and more obvious as the enlargement of the constant π. Additionally, we let the control updating times of event-based and time-based formulations be denoted as $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively. In this example, we apply the traditional algorithm for 300 time steps, so that $\mathcal{T}_2 = 300$. However, with the involvement of the event-based mechanism, the related updating times of control signals are overtly re-

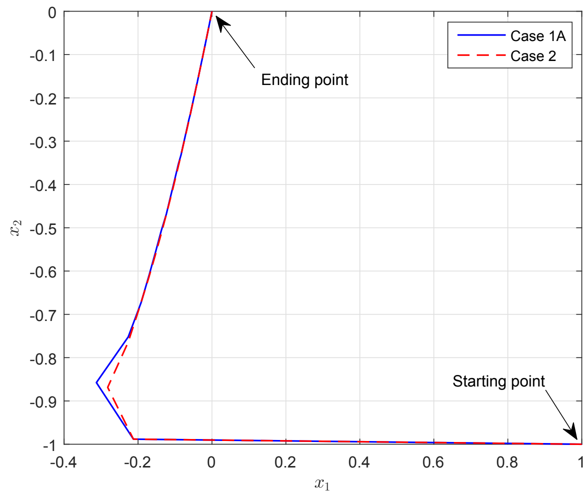

Fig. 5. State trajectories of Case 1A ($\pi = 0.2$) and Case 2.

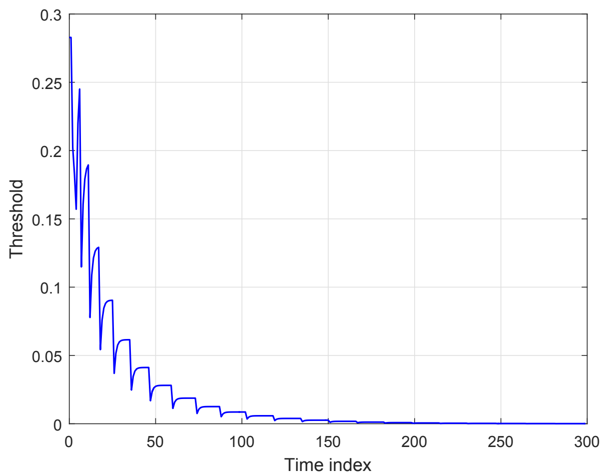

Fig. 6. Triggering threshold of Case 1A ($\pi = 0.2$).

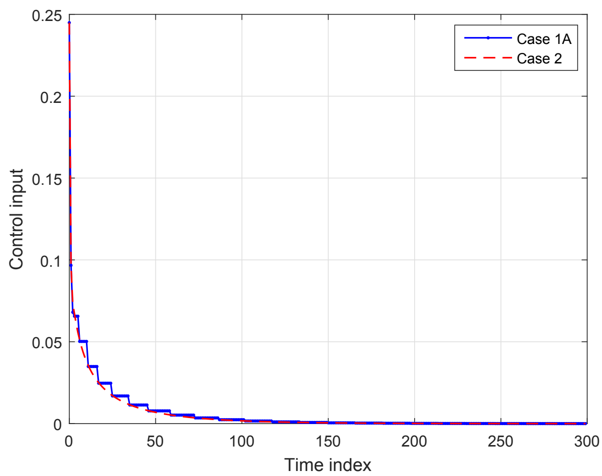

Fig. 7. Control inputs of Case 1A ($\pi = 0.2$) and Case 2.

duced. For Cases 1A, 1B, and 1C, the updating times are $\mathcal{T}_1 = 25$, $\mathcal{T}_1 = 80$, and $\mathcal{T}_1 = 8$, respectively. Such simulation results verify that the event-driven scheme can greatly lessen control updating times while still guarantee a satisfying control performance.

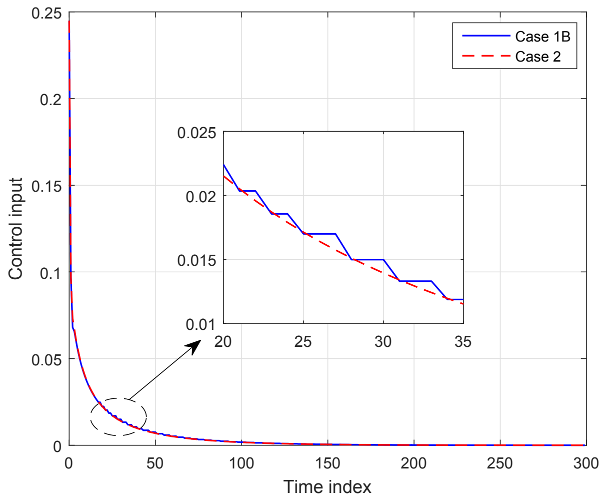

Fig. 8. Control inputs of Case 1B ($\pi = 0.1$) and Case 2.

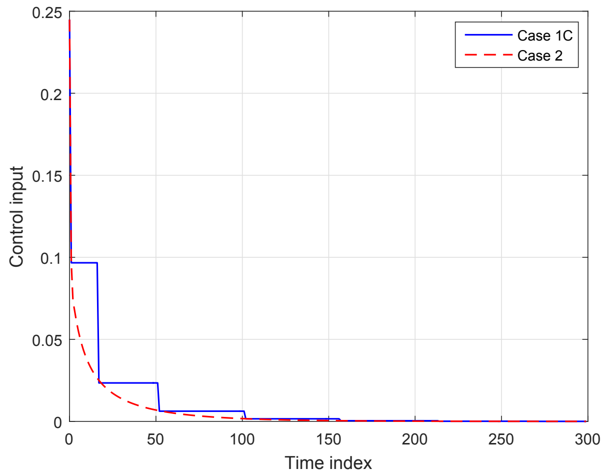

Fig. 9. Control inputs of Case 1C ($\pi = 0.3$) and Case 2.

5. CONCLUSION

In this paper, a mixed driven framework including data and event manners is developed for addressing discrete-time nonlinear optimal regulation. The mixed driven iterative adaptive critic algorithm is presented with neural dynamic programming implementation and proper triggering threshold design. For clarifying the effectiveness, simulation studies are illustrated to display the neural near-optimal control performance with an emphasis on nonaffine dynamics. Practical applications based on the proposed formulation, such as for complex wastewater treatment processes, will be paid attention to in the future.

REFERENCES

Al-Tamimi, A., Lewis, F. L., & Abu-Khalaf, M. (2008). Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, 38(4), 943–949.

Bian, T., Jiang, Y., & Jiang, Z. P. (2014). Adaptive dynamic programming and optimal control of nonlinear nonaffine systems. *Automatica*, 50(10), 2624–2632.

Dong, L., Zhong, X., Sun, C., & He, H. (2017). Adaptive event-triggered control based on heuristic dynamic programming for nonlinear discrete-time systems. *IEEE Transactions on Neural Networks and Learning Systems*, 28(7), 1594–1605.

Eqtami, A., Dimarogonas, D. V., & Kyriakopoulos, K. J. (2010). Event-triggered control for discrete-time systems. In *Proceedings of American Control Conference*, Baltimore, MD, USA, June 2010, pp. 4719–4724.

Fan, Q. Y., & Yang, G. H. (2018). Event-based fuzzy adaptive fault-tolerant control for a class of nonlinear systems. *IEEE Transactions on Fuzzy Systems*, 26(5), 2686–2698.

Garcia, E., & Antsaklis, P. J. (2014). Optimal model-based control with limited communication. In *Proceedings of 19th IFAC World Congress*, Cape Town, South Africa, August 2014, pp. 10908–10913.

Ha, M., Wang, D., & Liu, D. (2020). Event-triggered adaptive critic control design for discrete-time constrained nonlinear systems. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, in press.

Kiumarsi, B., Kang, W., & Lewis, F. L. (2016). H_∞ control of nonaffine aerial systems using off-policy reinforcement learning. *Unmanned Systems*, 4(1), 51–60.

Nowzari, C., Garcia, E., Cortes, J. (2019). Event-triggered communication and control of networked systems for multi-agent consensus. *Automatica*, 105, 1–27.

Si, J., & Wang, Y. T. (2001). On-line learning control by association and reinforcement. *IEEE Transactions on Neural Networks*, 12(2), 264–276.

Wang, D., & Liu, D. (2018). Learning and guaranteed cost control with event-based adaptive critic implementation. *IEEE Transactions on Neural Networks and Learning Systems*, 29(12), 6004–6014.

Wang, D., Liu, D., Wei, Q., Zhao, D., & Jin, N. (2012). Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming. *Automatica*, 48(8), 1825–1832.

Wang, D., & Zhong, X. (2019). Advanced policy learning near-optimal regulation. *IEEE/CAA Journal of Automatica Sinica*, 6(3), 743–749.

Werbos, P. J. (1992). Approximate dynamic programming for real-time control and neural modeling. In White, D. A., & Sofge, D. A. (Eds.), *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*. New York, NY: Van Nostrand Reinhold, ch. 13.

Zhang, H., Luo, Y., & Liu, D. (2009). Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints. *IEEE Transactions on Neural Networks*, 20(9), 1490–1503.

Zhang, H., Liu, Y., Xiao, G., & Jiang, H. (2020). Data-based adaptive dynamic programming for a class of discrete-time systems with multiple delays. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(2), 432–441.