Event-Based Optimal Control for Cooperative Adaptive Cruise Control

Jingwei Lu, Qinglai Wei, and Xiang Cheng

***Abstract*—This paper develops a new event-based optimal control (EBOC) method for cooperative adaptive cruise control (CACC) systems using parallel control and adaptive dynamic programming (ADP). First, based on the framework of optimal control and event-based control, CACC of connected vehicles is modeled as optimal control, and a triggering condition is designed to achieve the event-based CACC. Furthermore, the control stability of the closed-loop system under the developed triggering condition is shown theoretically. Then, iterative ADP and parallel control are utilized to implement the EBOC method. Finally, the correctness of the proposed EBOC method is verified by the application to a CACC system.**

I. INTRODUCTION

In the last decade, autonomous driving has gained prominence because of the development of artificial intelligence (AI) [1–4]. Depending on the number of vehicles and driving tasks, common control problems for autonomous driving include path or velocity tracking control of a single vehicle [5–6], adaptive cruise control (ACC) for ego and leader vehicles [7–8], and cooperative ACC (CACC) for multiple vehicles [9]. Path or speed tracking control generally refers to the design of a control policy allowing an autonomous vehicle to track given path trajectories or speeds. ACC, usually containing ego and leader vehicles, involves designing a control policy based on headway and velocity to adjust the dynamics of the ego vehicle so that it tracks the leader vehicle. CACC is no longer limited to ego and leader vehicles, but rather cooperative driving control for multiple vehicles, which is an advanced version of ACC. In CACC, control policies implement control for vehicles according to real-time feedback information from different vehicles.

Optimal control is an influential branch of modern control theory, which tightly connects AI and control technology [10]. In nonlinear optimal control, a complex equation, i.e., the Hamilton-Jacobi-Bellman (HJB) equation, usually needs to be addressed, and thus adaptive dynamic programming (ADP), also known as reinforcement learning (RL), approximate dynamic programming, neuro-dynamic programming, etc., is utilized to derive the numerical solution of the HJB equation [11]. Besides ADP, parallel control is an effective data-driven control approach, which is based on the ACP methodology (ACP: artificial systems, computational experiments, and parallel execution) . Nowadays, ADP and parallel control have been utilized to CACC of connected vehicles [12–13].

Event-based control is an effective scheme to save communication and computational resources for control systems [14]. Over the last few years, event-based control has been employed to nonlinear optimal control to save communication resources and optimize the performance simultaneously [15]. In event-based optimal control (EBOC), the event-based HJB equation may need to be solved, which is usually more difficult to solve than the time-based one [16]. EBOC has now also been widely deployed in intelligent transportation systems, for example, tracking control [17], ACC [18], CACC [19], decision-making of autonomous driving [3], etc.

As mentioned earlier, event-based control is an effective solution for saving communication and computational resources, which makes it suitable for controlling large and complex systems. Connected vehicles are a typical large-scale complex system that involves the control and communication of multiple autonomous vehicles. To provide a practical method for implementing event-based CACC of connected vehicles, we introduce a novel EBOC method using parallel control and iterative ADP. The main contributions are as follows:

(1) A triggering condition is developed for CACC of connected vehicles using the time-based optimal value function and control policy. Therefore, implementing the triggering condition does not require solving the event-based HJB equation, providing a simpler implementation of event-based CACC.

(2) Because of the developed triggering condition, offline iterative ADP algorithms can be utilized to achieve the developed triggering condition. Compared with the existing EBOC methods for CACC or ACC [18–19], the proposed method provides a practical way for event-based CACC.

(3) In the implementation scheme, we utilize parallel control to recover the CACC system and predict the CACC system behavior. Therefore, the prior knowledge of the CACC system is not required.

The rest of this paper is arranged as follows. Section II concisely introduces CACC and EBOC. Section III presents the developed EBOC method, including a triggering condition

Manuscript received: 7 February 2025; revised: 21 April 2025; accepted: 12 May 2025. *(Corresponding author: Jingwei Lu.)*

Citation: J. Lu, Q. Wei, and X. Cheng, Event-based optimal control for cooperative adaptive cruise control, *Int. J. Intell. Control Syst.*, 2025, 30(2), 176–181.

Jingwei Lu is with Department of Industrial Engineering, Tsinghua University, Beijing 100084, China (e-mail: lujingweihh@gmail.com).

Qinglai Wei and Xiang Cheng are with State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

Qinglai Wei is also with Institute of Systems Engineering, Macau University of Science and Technology, Macao 999078, China (e-mail: qinglai.wei@ia.ac.cn, xiang.cheng@ia.ac.cn).

Digital Object Identifier 10.62678/IJICS202506.10179

and theoretical analysis. Section IV shows an implementation scheme for the EBOC method. Section V conducts a numerical study for applying our EBOC method to a CACC system. Section VI draws some conclusions.

II. Problem Formulation

Before presenting the developed EBOC method, this section briefly introduces CACC and EBOC.

A. Cooperative Adaptive Cruise Control

First, the goal of CACC is introduced. The schematic of CACC of connected vehicles is given in Fig. 1. Based on Fig. 1, we have

$$\begin{cases} e_i^d = d_{i-1} - d_i - w_{i-1} - (w_d + t_d v_i), \\ e_i^v = v_{i-1} - v_i, \ i = 1,2 \end{cases} \tag{1}$$

where d_i, w_i, and v_i are the displacement, length, and velocity of the vehicle i, respectively, with $i = 0,1,2$. $w_d > 0$ is a standstill distance guaranteeing the safety. $t_d > 0$ is the time gap.

Subsequently, denoting accelerations $a_j \triangleq \dot{v}_j$, $j = 0,1,2$, and modeling the acceleration dynamics as $\dot{a}_i = -(1/\tau_i)a_i + (1/\tau_i)u_i$ with τ_i being time constants, $i = 1,2,...,N$, one can get

$$\begin{cases} \dot{e}_i^d = e_i^v - t_d a_i, \\ \dot{e}_i^v = a_{i-1} - a_i, \\ \dot{a}_i = -\dfrac{1}{\tau_i}a_i + \dfrac{1}{\tau_i}u_i, \ i = 1,2,...,N \end{cases} \tag{2}$$

The goal of CACC is to ensure a constant time gap, and we set $t_d = 1$ s, $a_0 = 0$ m/s^2, and $\tau_i = 0.1$ s in this paper.

Defining the state vector $e \triangleq [e_1^d, e_1^v, a_1, e_2^d, e_2^v, a_2, ..., e_N^d, e_N^v, a_N]^{\mathrm{T}} \in \mathbb{R}^{3N}$, and the control vector $u \triangleq [u_1, u_2, ..., u_N]^{\mathrm{T}} \in \mathbb{R}^N$, we can obtain the following vehicular platoon dynamics (VPD)

$$\dot{e} = f(e) + gu \tag{3}$$

where the CACC system dynamics $f(e)$ and g are determined by the dynamics in Eq. (2).

In real-world deployments, feedback and control signals are necessarily in the discrete-time (DT) form. Thus, in this paper, the derivation of control policy and simulations are discussed based on the DT VPD. Discretization of Eq. (3) using the sampling time T_s [20], one can get

$$e(k+1) = f_d(e(k)) + g_d u(k) \tag{4}$$

where $e(k) \triangleq [e_1^d(k), e_1^v(k), a_1(k), e_2^d(k), e_2^v(k), a_2(k), ..., e_N^d(k), e_N^v(k), a_N(k)]^{\mathrm{T}} \in \mathbb{R}^{3N}$ and $u(k) \triangleq [u_1(k), u_2(k), ..., u_N(k)] \in \mathbb{R}^N$ are separately the discretized tracking errors and feedback control inputs, and the discretized system functions are $f_d(e(k)) = e(k) + T_s f(e(k))$ and $g_d = T_s g$. For the ease of analysis, Assumption 1 is made before our design.

Assumption 1 On a compact set Ω including the origin, the VPD in Eq. (4) is stabilized, and $f_d + g_d u$ is Lipschitz continuous with $f_d(0) = 0$.

B. Event-Based Optimal Control

Infinite-time optimal control seeks to obtain a control policy to stabilize the VPD in Eq. (4) and simultaneously minimize

$$J(e(0), u(0)) = \sum_{k=0}^{\infty} U(e(k), u(k)) \tag{5}$$

with $U(e(k), u(k)) > 0$ being the utility function.

Without considering EBOC, the value function associated with an admissible control policy μ is [11]

$$V(e(k)) = \sum_{\tau=k}^{\infty} U(e(\tau), \mu(\tau)) \tag{6}$$

The optimal value function satisfies

$$V^*(e(k)) = \min_{\mu_e(k)} \{U(e(k), \mu(k)) + V^*(e(k+1))\} \tag{7}$$

At the same time, the optimal control input can be derived by

$$u^*(k) = \arg\min_{\mu(k)} \{U(e(k), \mu(k)) + V^*(e(k+1))\} \tag{8}$$

Besides that, it is shown in Ref. [11] that

$$J^*(e(k)) = V^*(e(k)) \tag{9}$$

In conventional optimal control, we customarily develop state-feedback control policies to achieve optimal control, for example $u(k) = \mu(e(k))$. Therefore, the control $u(k)$ depends on the state $e(k)$, and the control policy mandates that the control input varies according to changes in the state vector. Instead, based on state-feedback control, event-based control is expressed by

$$u(k) = \nu(e(k_i)), \ k \in [k_i, k_{i+1}) \tag{10}$$

where $\{k_i\}$ is the set of triggering instants with $k_0 = 0$, $k_i < k_{i+1}$, $i \in \mathbb{N}$ and $\mathbb{N}$ being the set of non-negative integers, and $\nu_e(e(k))$ is a feedback control policy only working at triggering instants. In EBOC, we should not only derive an event-based control policy in Eq. (10) to stabilize the VPD in

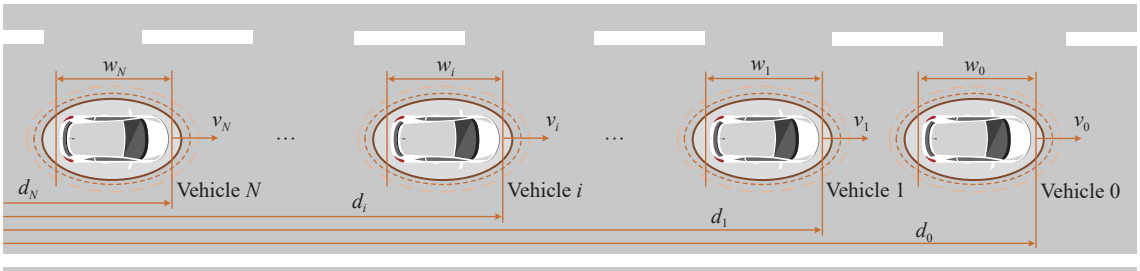

Figure 1 Schcmatic of a vchicular platoon.

Eq. (4), but also consider optimizing the performance index in Eq. (5).

III. Event-Based Optimal Control Design

This section presents our EBOC method to achieve CACC. First, a triggering condition is developed. Then, theoretical analysis is given to demonstrate the correctness of the EBOC method.

A. Triggering Condition

To achieve event-based CACC for connected vehicles, we first present a triggering condition. According to Eqs. (8) and (10), the following event-based control policy is developed

$$u(k) = \pi(e(k)) = u^*(e(k_i)),\ k \in [k_i, k_{i+1}) \tag{11}$$

On the basis of Eq. (11), the closed-loop VPD (CVPD) is

$$e(k+1) = f_d(e(k)) + g_d u^*(e(k_i)),\ k \in [k_i, k_{i+1}) \tag{12}$$

Now, inspired by Ref. [15], the following triggering condition is developed

$$I_\alpha(e(k+1), e(k), e(k_i)) \leqslant 0 \tag{13}$$

where

$$I_\alpha(e(k+1), e(k), e(k_i)) = \alpha(V^*(e(k+1)) - V^*(e(k))) + U(e(k), u^*(e(k_i)))$$

with $\alpha \geqslant 1$. In the meantime, the triggering instant is decided by

$$k_{i+1} = \inf\{k | I_\alpha(e(k+1), e(k), e(k_i)) > 0, k > k_i\} \tag{14}$$

Remark 1 Notice that the triggering condition in Formula (13) is constructed based on the conventional optimal value function in Eq. (7) and control input in Eq. (8) without event-based control, and thus the existing offline iterative ADP methods are effective in obtaining Eqs. (7) and (8).

B. Theoretical Analysis

To theoretically prove the correctness of the developed EBOC method and guarantee the safety of EBOC of connect vehicles, based on the Lyapunov approach [21], Theorem 1 shows the control stability of the CVPD in Eq. (12) determined by our EBOC method.

Theorem 1 Consider the CVPD in Eq. (12), the event-based control policy in Eq. (11), and the triggering condition in Formula (13). Letting Assumption 1 be met, and denoting $J(e(0), \pi)$ as the actual performance index under the event-based control policy in Eq. (11) with the triggering condition in Formula (13), we can obtain the following conclusions.

(1) The CVPD in Eq. (11) is asymptotically stable.

(2) The actual performance index $J(e(0), \pi)$ satisfies $J(e(0), \pi) \leqslant \alpha J^*(e(0))$.

Proof Choose the optimal value function as the Lyapunov function at first, i.e., $L(k) = V^*(e(k))$. Depending on whether k is a triggering instant or not, the discussion for stability of the CVPD in Eq. (12) can be divided into two cases.

Case 1 The instant k is not a triggering instant, i.e., $k \notin \{k_i\}$. Thus, one can derive

$$\begin{aligned} \triangle L(k) &= V^*(e(k+1)) - V^*(e(k)) = \\ &\frac{1}{\alpha}(I_\alpha(e(k+1), e(k)) - U(e(k), u^*(e(k_i)))) \leqslant \\ &-\frac{1}{\alpha} U(e(k), u^*(e(k_i))) \leqslant 0 \end{aligned} \tag{15}$$

Case 2 The instant k is a triggering instant, i.e., $k \in \{k_i\}$. Then, one has

$$\begin{aligned} \triangle L(k) &= V^*(e(k+1)) - V^*(e(k)) = \\ &- U(e(k), u_e^*(e(k))) \leqslant 0 \end{aligned} \tag{16}$$

Thus, according to the Lyapunov approach [21], the CVPD in Eq. (11) is asymptotically stable.

Now, according to the asymptotic stability of the CVPD in Eq. (11), we can derive that $\lim_{k\to\infty} e(k) = 0$ and $\lim_{k\to\infty} V^*(e(k)) = 0$.

Meanwhile, on the basis of Formula (15), one can derive that $U(e(k), u^*(e(k_i))) \leqslant \alpha(V^*(e(k)) - V^*(e(k+1)))$ for $k \notin \{k_i\}$. Then, according to Formula (16), we can obtain that $U(e(k), u_e^*(e(k))) = V^*(e(k)) - V^*(e(k+1)) \leqslant \alpha(V^*(e(k)) - V^*(e(k+1)))$ for $k \in \{k_i\}$. Overall, we can derive

$$U(e(k), \pi(e(k))) \leqslant \alpha\left(V^*(e(k)) - V^*(e(k+1))\right) \tag{17}$$

Then, it follows from Formula (17) that

$$\begin{aligned} J(e(0), \pi) &= \sum_{k=0}^{\infty} U(e(k), \pi(e(k))) \leqslant \\ &\sum_{k=0}^{\infty} \alpha\left(V^*(e(k)) - V^*(e(k+1))\right) = \\ &\alpha V^*(e(0)) - \alpha \lim_{k\to\infty} V^*(e(k)) = \\ &\alpha V^*(e(0)) = \alpha J^*(e(0)) \end{aligned} \tag{18}$$

Therefore, we have $J(e(0), \pi) \leqslant \alpha J^*(e(0))$. ■

Remark 2 Based on the Bellman principle of optimality, the optimal performance index in Eq. (9) is the minimum value for all the control modes. The introduction of event-based control will inevitably degrade this performance index. The conclusion (2) in Theorem 1 preliminarily analyzes changes in the actual performance index under the developed EBOC method and provides an upper bound for the actual performance index.

IV. Implementation Scheme

To deploy our EBOC method, we have to obtain $e(k+1)$ at instant k, which is almost impossible to realize with the real-world systems alone. Therefore, this section adopts ADP and parallel control to fulfill the EBOC method.

Based on the parallel control, our implementation scheme consists of the following three steps.

(1) Artificial systems (A): A model neural network (NN) is employed to recover the VPD in Eq. (4) and then predict $\hat{e}(k+1)$ at k.

(2) Computational experiments (C): ADP, including the critic and action NNs, is utilized to derive the optimal value function in Eq. (7) and control policy in Eq. (8).

(3) Parallel execution (P): Based on the actual state $e(k)$ and the predicted state $\hat{e}(k+1)$, the triggering condition in Formula (13) is achieved.

Then, we present the NNs designed for EBOC of connected vehicles.

Model NN. The model NN is used for representation of the VPD in Eq. (4) and is

$$e(k+1) = \theta_m^{*\mathrm{T}}\phi_m(\beta(k)) + \epsilon_m(k) \tag{19}$$

where $\beta(k) = [e^{\mathrm{T}}(k), u^{\mathrm{T}}(k)]^{\mathrm{T}}$, $\theta_m^* \in \mathbb{R}^{\eta_m \times 4N}$ is the ideal model NN weight, $\phi_m(\beta(k))$: $\mathbb{R}^{4N} \to \mathbb{R}^{\eta_m}$ is the model NN activation function, and $\epsilon_m(k)$ is the approximation error.

Due to that θ_m^* is unknown, the model NN is

$$\hat{e}(k+1) = \theta_m^{\mathrm{T}}\phi_m(\beta(k)) \tag{20}$$

with θ_m being the approximation of θ_m^*. In the training, we tune the model NN weights by minimizing the difference

$$\xi_m(k) = \frac{1}{2}\|\tilde{e}(k+1)\|^2 \tag{21}$$

where $\tilde{e}(k+1) = \hat{e}(k+1) - e(k+1)$.

Critic NN. The critic NN is utilized for approximating of Eq. (7) and is given by

$$V^*(e(k)) = \theta_c^{*\mathrm{T}}\phi_c(e(k)) + \epsilon_c(k) \tag{22}$$

where $\theta_c^* \in \mathbb{R}^{\eta_c}$ is the ideal critic NN weight, $\phi_c(e(k))$: $\mathbb{R}^{3N} \to \mathbb{R}^{\eta_c}$ is the critic NN activation function, and $\epsilon_c(k)$ is the approximation error.

Similarly, critic network in the implementation is

$$\hat{V}(e(k)) = \hat{\theta}_c^{\mathrm{T}}\phi_c(e(k)) \tag{23}$$

with $\hat{\theta}_c$ being the approximation of θ_c^*. According to the Bellman equation [11], we tune the critic NN weights by minimizing the difference

$$\xi_c(k) = \frac{1}{2}\|\varepsilon_c(k)\|^2 \tag{24}$$

where $\varepsilon_c(k) = U(e(k), \hat{u}(e(k))) + \hat{V}(e(k)) - \hat{V}(\hat{e}(k+1))$ with $\hat{u}(e(k))$ being the action NN.

Action NN. The action NN is utilized for approximating of Eq. (8) and is defined as

$$u^*(e(k)) = \theta_a^{*\mathrm{T}}\phi_a(e(k)) + \epsilon_a(k) \tag{25}$$

where $\theta_a^* \in \mathbb{R}^{\eta_a}$ is the ideal action NN weight, $\phi_a(e(k))$: $\mathbb{R}^N \to \mathbb{R}^{\eta_a}$ is the action NN activation function, and $\epsilon_a(k)$ is the approximation error.

During training, the action network is rewritten as

$$\hat{u}(e(k)) = \hat{\theta}_a^{\mathrm{T}}\phi_a(e(k)) \tag{26}$$

with $\hat{\theta}_a$ being the approximation of θ_a^*. In the training, we update the action NN weights by minimizing the difference

$$\xi_a(k) = \frac{1}{2}\|\varepsilon_a(k)\|^2 \tag{27}$$

where $\varepsilon_a(k) = \hat{V}(\hat{e}(k+1)) + U(e(k), \hat{u}(e(k)))$.

Now, based on the model, critic NN, and action NN, the triggering condition in Formula (13) is

$$I_\alpha(\hat{e}(k+1), e(k), e(k_i)) \leqslant 0 \tag{28}$$

where $I_\alpha(e(k+1), e(k), e(k_i)) = \alpha(\hat{V}(e(k+1)) - \hat{V}(e(k))) + U(e(k), \hat{u}(e(k)))$.

Remark 3 Notice that Eqs. (7) and (8) are the conventional optimal value function and control input without event-based control, and they can be obtained by the iterative or online ADP. Thus, considering convergence and so forth, we employ offline iterative ADP.

V. Application to Cooperative Adaptive Cruise Control

This section conducts a numerical study for CACC of the VPD in Eq. (4) under the proposed EBOC method, including obtaining the the optimal value function in Eq. (7) and control policy in Eq. (8) based on the iterative ADP, and applying the developed EBOC method to the VPD in Eq. (4).

We first set $N = 3$ in the numerical study. The learning parameters are as follows: $U(e(k), u(k)) = e^{\mathrm{T}}(k)e(k) + u^{\mathrm{T}}(k)u(k)$, $\phi_c(e(k)) = [e_1^2(k), e_1(k)e_2(k), e_1(k)e_3(k), e_1(k)e_4(k), e_1(k)e_5(k), e_1(k)e_6(k), e_2^2(k), e_2(k)e_3(k), e_2(k)e_4(k), e_2(k)e_5(k), e_2(k)e_6(k), e_3^2(k), e_3(k)e_4(k), e_3(k)e_5(k), e_3(k)e_6(k), e_4^2(k), e_4(k)e_5(k), e_4(k)e_6(k), e_5^2(k), e_5(k)e_6(k), e_6^2(k)]^{\mathrm{T}}$, $\phi_a(e(k)) = [e_1(k), e_2(k), e_3(k), e_4(k), e_5(k), e_6(k)]^{\mathrm{T}}$, and $x(0) = [5,5,0,5,5,0]^{\mathrm{T}}$. Then, by utilizing the value iteration algorithm [11], the learning results are shown in Figs. 2 and 3, including iterative value functions and the evolution of the weights, and the final weights are

$$\begin{aligned}\theta_c = [&14.9882, 10.9817, -2.9722, 0.7333, 6.5114, -0.7171,\\ &24.5021, -4.1276, -4.4924, 15.1627, -1.1120, 1.3829,\\ &0.8341, 0.9380, -0.1689, 13.5499, 7.0190, -2.9609,\\ &16.5524, -3.7422, 1.3203]^{\mathrm{T}},\end{aligned}$$

and

$$\theta_a = \begin{bmatrix} 0.6300, & 0.9387, & -0.1674, & -0.1526, & -0.1837, & 0.0321 \\ 0.1775, & 0.2892, & 0.0351, & 0.6325, & 0.8635, & -0.1433 \end{bmatrix}^{\mathrm{T}}.$$

The learning results indicate that value iteration guarantees the convergence of the NN weights for CACC of the VPD in Eq. (4).

Next, we apply the proposed EBOC method to the VPD in Eq. (4), and the triggering parameter is $\alpha = 0.1$. The states of vehicles 1 and 2, i.e., e_i^d, e_i^v, and a_i, $i = 1,2$, are displayed in Fig. 4, and the controls of vehicles 1 and 2, i.e., u_i, $i = 1,2$ are

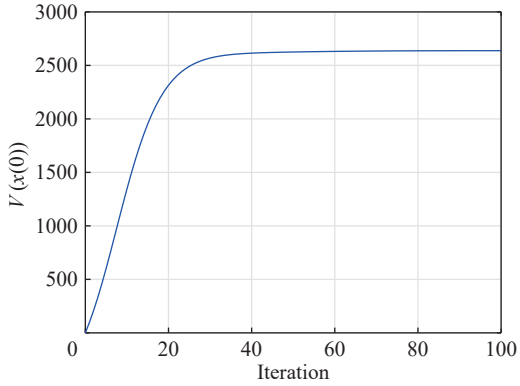

Figure 2 Iterative value function.

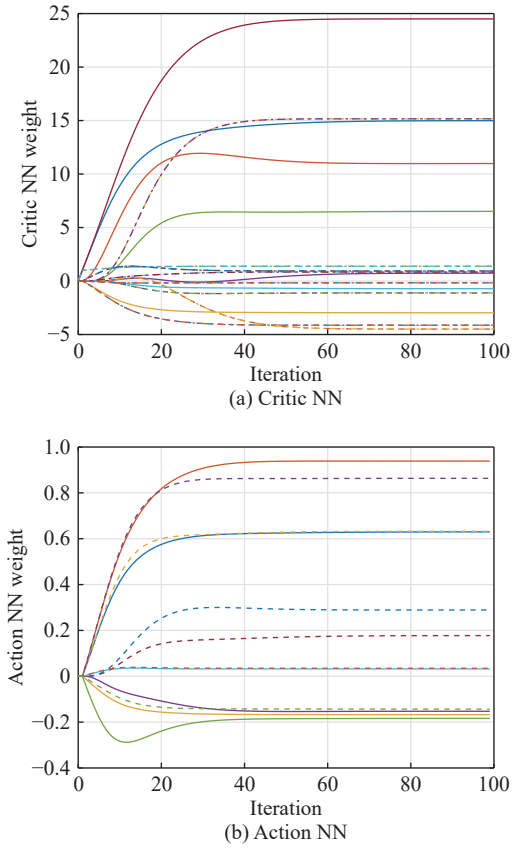

Figure 3 Evolution of NN weight.

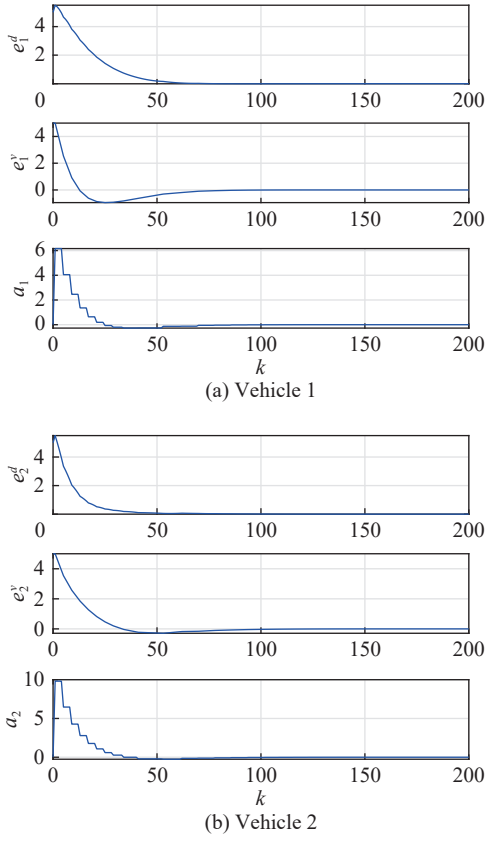

Figure 4 Evolution of state.

presented in Fig. 5. From Figs. 4 and 5, we obtain that the proposed EBOC method achieves CACC of the VPD in Eq. (4). That is, the tracking errors e_i^d and e_i^v converge to a small neighborhood of 0. In the meantime, the root mean squared errors (RMSEs) of e_i^d and e_i^v, $i = 1, 2$, are utilized to evaluate the control performance. We define $\text{RMSE}_i^d = \sqrt{\sum_{i=1}^{N}(e_i^d)^2/N}$ and $\text{RMSE}_i^v = \sqrt{\sum_{i=1}^{N}(e_i^v)^2/N}$, where N is the maximum time step. Then, one has $\text{RMSE}_1^d = 1.3077$, $\text{RMSE}_2^d = 0.9264$, $\text{RMSE}_1^v = 0.8095$, and $\text{RMSE}_2^v = 0.9673$. In the meantime, the number of control input updates is 28, which means that the proposed method saves 86% of the communication resources. Overall, the application demonstrates the effectiveness of the proposed EBOC method in the CACC of vehicular platoons.

VI. Conclusion

In this paper, a new EBOC method is developed for the CACC of vehicular platoons using ADP. First, the CACC of connected vehicles is modeled as optimal control. Then, a new triggering condition is proposed using the time-based optimal value function and control policy. Furthermore, the control stability of the CACC system under the triggering condition is shown theoretically. Subsequently, an implementation scheme is presented for the proposed EBOC method based on the parallel control and iterative ADP. Finally, the correctness of the proposed EBOC method is verified by the CACC of connected vehicles.

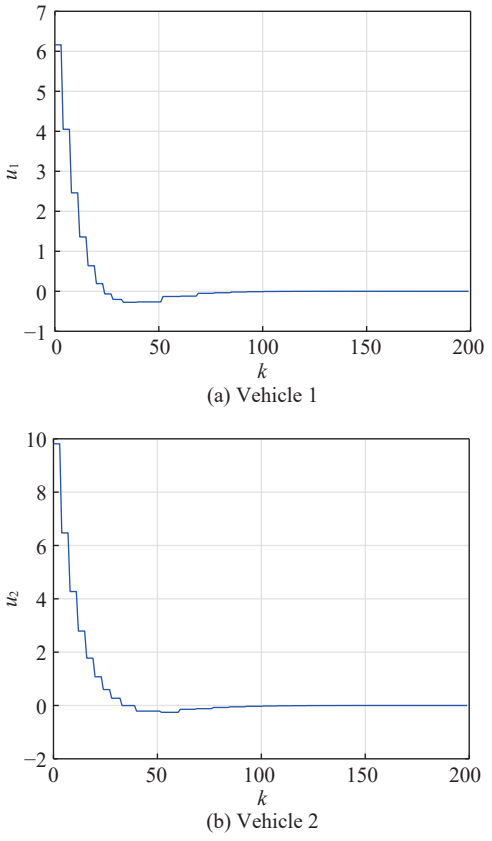

Figure 5 Evolution of control.

ACKNOWLEDGMENT

This work was supported by the China Postdoctoral Science Foundation (No. 2025M770781) and the Postdoctoral Fellowship Program of CPSF (No. GZC20240839).

REFERENCES

[1] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, A survey of deep learning techniques for autonomous driving, *J. Field Robot.*, 2020, 37(3), 362–386.

[2] D. Omeiza, H. Webb, M. Jirotka, and L. Kunze, Explanations in autonomous driving: A survey, *IEEE Trans. Intell. Transport. Syst.*, 2022, 23(8), 10142–10162.

[3] J. Lu, L. Han, Q. Wei, X. Wang, X. Dai, and F.-Y. Wang, Event-triggered deep reinforcement learning using parallel control: A case study in autonomous driving, *IEEE Trans. Intell. Veh.*, 2023, 8(4), 2821–2831.

[4] J. Zhao, W. Zhao, B. Deng, Z. Wang, F. Zhang, W. Zheng, W. Cao, J. Nan, Y. Lian, and A. F. Burke, Autonomous driving system: A comprehensive survey, *Expert Syst. Appl.*, 2024, 242, 122836.

[5] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker, A predictive controller for autonomous vehicle path tracking, *IEEE Trans. Intell. Transport. Syst.*, 2009, 10(1), 92-102.

[6] F. Dang, D. Chen, J. Chen, and Z. Li, Event-triggered model predictive control with deep reinforcement learning for autonomous driving, *IEEE Trans. Intell. Veh.*, 2024, 9(1), 459-468.

[7] S. Li, K. Li, R. Rajamani, and J. Wang, Model predictive multi-objective vehicular adaptive cruise control, *IEEE Trans. Contr. Syst. Technol.*, 2011, 19(3), 556–566.

[8] J. Zhao, Z. Wang, Y. Lv, J. Na, C. Liu, and Z. Zhao, Data-driven learning for H_∞ control of adaptive cruise control systems, *IEEE Trans. Veh. Technol.*, 2024, 73(12), 18348–18362.

[9] V. Milanes, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, Cooperative adaptive cruise control in real traffic situations, *IEEE Trans. Intell. Transport. Syst.*, 2014, 15(1), 296–305.

[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[11] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof, *IEEE Trans. Syst., Man, Cybern. Part B: Cybern.*, 2008, 38(4), 943–949.

[12] J. Li, Y. Liu, A. Fotouhi, X. Wang, Z. Chen, Y. Zhang, and L. Li, Cooperative ecological adaptive cruise control for plug-in hybrid electric vehicle based on approximate dynamic programming, *IEEE Trans. Veh. Technol.*, 2023, 72(3), 3132–3145.

[13] J. Lu, Q. Wei, and F.-Y. Wang, Parallel control for nonzero-sum games with completely unknown nonlinear dynamics via reinforcement learning, *IEEE Trans. Syst., Man, Cybern.: Syst.*, 2025, 55(4), 2884–2896.

[14] K. J. Aström, Event based control, in *Analysis and Design of Nonlinear Control Systems: In Honor of Alberto Isidori*. Berlin, Germany: Springer, 2008, 127–147.

[15] J. Lu, Q. Wei, T. Zhou, Z. Wang, and F.-Y. Wang, Event-triggered near-optimal control for unknown discrete-time nonlinear systems using parallel control, *IEEE Trans. Cybern.*, 2023, 53(3), 1890–1904.

[16] L. Dong, X. Zhong, C. Sun, and H. He, Adaptive event-triggered control based on heuristic dynamic programming for nonlinear discrete-time systems, *IEEE Trans. Neural Netw. Learning Syst.*, 2017, 28(7), 1594–1605.

[17] J. Lu, L. Li, and F.-Y. Wang, Event-triggered parallel control using deep reinforcement learning with application to comfortable autonomous driving, *IEEE Trans. Intell. Veh.*, 2024, 9(3), 4470–4479.

[18] Z. Wang and C. Hu, Optimal adaptive cruise control of connected vehicles with prescribed performance via event-triggered ADP, in *Proc. 2024 8th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, Chongqing, China, 2024, 1–6.

[19] F. Zhao, W. Gao, T. Liu, and Z. Jiang, Event-triggered cooperative adaptive cruise control of vehicular platoons, in *Proc. 2024 IEEE 63rd Conference on Decision and Control (CDC)*, Milan, Italy, 2024, 6421–6427.

[20] S. K. Gupta, *Numerical Methods for Engineers*. London, UK: New Academic Science, 1995.

[21] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice Hall, 2002.

Jingwei Lu received the PhD degree in computer application technology from University of Chinese Academy of Sciences, Beijing, China, in 2022. He is currently a postdoctoral research fellow at Department of Industrial Engineering, Tsinghua University, Beijing, China. He has authored or coauthored over 30 journal and conference articles. He is a recipient of the Shuimu Tsinghua Scholar Program. He was the guest editor of *IEEE Journal of Radio Frequency Identification*. He served as a peer reviewer for *IEEE Transactions on Cybernetics*, *IEEE Transactions on Industrial Electronics*, *IEEE Transactions on Intelligent Transportation Systems*, etc. His research interests include optimal control, adaptive dynamic programming, deep reinforcement learning, and autonomous driving.

Qinglai Wei received the BS degree in automation and the PhD degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2002 and 2009, respectively. From 2009 to 2011, he was a postdoctoral fellow at State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He is currently a professor at Institute of Automation, Chinese Academy of Sciences, China. He has authored 4 books and published over 80 international journal articles. His research interests include adaptive dynamic programming, neural networks based control, optimal control, nonlinear systems, and their industrial applications. He was a recipient of the IEEE/CAA Journal of Automatica Sinica Best Paper Award, the IEEE System, Man, and Cybernetics Society Andrew P. Sage Best Transactions Paper Award, the IEEE Transactions on Neural Networks and Learning Systems Outstanding Paper Award, the Outstanding Paper Award of Acta Automatica Sinica, and the Zhang Siying Outstanding Paper Award of the Chinese Control and Decision Conference (CCDC). He was also a recipient of the Shuang-Chuang Talents in Jiangsu Province, China, and the Young Researcher Award of Asia-Pacific Neural Network Society (APNNS). He has been the vice president of the IEEE Computational Intelligence Society's (CIS) Beijing Chapter since 2022. He was the general co-chair of the 2019 Symposium of Parallel Intelligence, the invited session chair of the IEEE 8th Data Driven Control and Learning Systems Conference, the special sessions chair of the 25th International Conference on Neural Information Processing, the program co-chair of the 24th International Conference on Neural Information Processing, and the registration chair of the 12th World Congress on Intelligent Control and Automation (WCICA2016) and the 2014 IEEE World Congress on Computational Intelligence (WCCI2014). He was a guest editor of several international journals. He is the deputy editor-in-chief of *IEEE/CAA Journal of Automatica Sinica*, *Neurocomputing*, and *Acta Automatica Sinica*. He is also an associate editor of *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Automation Science and Engineering*, *IEEE Transactions on Consumer Electronics*, *IEEE Transactions on Cognitive and Developmental Systems*, and *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Xiang Cheng received the BS degree in electrical engineering and the MS degree in control science and engineering from China University of Mining and Technology, Xuzhou, China, in 2016 and 2019, respectively. He is currently pursuing the PhD degree in pattern recognition and intelligent system at School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China, and also at State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He is also an engineer at Institute of Automation, Chinese Academy of Sciences, China. His research interests include adaptive dynamic programming, reinforcement learning, and modeling and optimization of complex industrial processes.