Int. J. Appl. Math. Comput. Sci., 2025, Vol. 35, No. 4, 563–575
DOI: 10.61822/amcs-2025-0040

AN OPTIMAL CONTROL METHOD FOR TRAJECTORY TRACKING ERROR DETECTION OF AUTONOMOUS VEHICLES

PENG-FEI FENG [a], BINGYI JIA [b,*], HUI-QING JIN [a,c], GUANG WANG [d]

[a] Anhui Provincial Key Laboratory of Traffic Information and Safety
Anhui Sanlian University
Hefei, Anhui, 230601, China

[b] College of Mechanical and Electronic Engineering
Shandong University of Science and Technology
Qingdao, Shandong, 266590, China
e-mail: bingyi.jia@sdust.edu.cn

[c] National Center of Engineering and Technology for Vehicle Driving Safety
Anhui Sanlian University
Hefei, Anhui, 230601, China

[d] Guohua (Qingdao) Intelligent Equipment Co. Ltd.
Qingdao, Shandong, 266000, China

This paper presents a trajectory tracking error detection method for autonomous vehicles (AVs) via an optimal control scheme, where two online learning algorithms are designed: (i) an adaptive learning algorithm (ALA) and (ii) a finite-time adaptive learning algorithm (FTALA). We first construct an error dynamic system by combining the kinematic equation of AVs and an ideal tracking trajectory. To realize the adaptive optimal control for AVs, the ALA is designed, which provides us with an online solution by resolving the derived Hamilton–Jacobi–Bellman (HJB) equation. Then, the FTALA is presented, which further relaxes the requirement of the system dynamics, i.e., the system drift is not required. The adaptive critic and control action in both online learning algorithms continuously and simultaneously interact, eliminating the need for iterative steps. This approach also avoids the use of an actor neural network (NN) and an initial stabilizing control policy. Moreover, the finite-time convergence can be ensured via adopting a sliding mode technique in the FTALA. Finally, both online learning algorithms are applied to control AVs, and the simulations show their effectiveness and practicality feasibility. The FTALA reduces the time-accumulated tracking error and the cumulative control effort by about 39% and 57% in lane-keeping, and by about 38% and 57% in the S-curve, respectively.

Keywords: adaptive dynamic programming, autonomous vehicles, finite-time convergence, optimal control.

1. Introduction

In recent years, with the rapid development of intelligent transportation and artificial intelligence technology, automatic driving technology has been widely discussed by scholars (Wang *et al.*, 2023). The automatic driving trajectory control method, as the key technology connecting previous and subsequent stages in automatic driving technology, is the safety guarantee of automatic driving. The control problems pertaining to this subject have garnered greater attention from researchers (Yan *et al.*, 2018; Hu *et al.*, 2016; Ding *et al.*, 2022). Nevertheless, due to their nonlinear dynamics, classical techniques (e.g., PID control) cannot achieve optimal performance. For this purpose, some nonlinear automatic driving control strategies are further suggested. The study of control theory has paved the way for the development of advanced strategies in vehicle control. Model predictive control (Elsisi and Ebrahim, 2021), robust control (Hu *et al.*, 2016), adaptive control

*Corresponding author

(Peng *et al*., 2019), and sliding-mode control (Chen *et al*., 2020) are among notable examples in this field. Moreover, artificial intelligence-based approaches like neural networks (Faryadi and Mohammadpour Velni, 2021) and fuzzy logic control (Lakhekar *et al*., 2019) have also been proposed for autonomous vehicle control. Nevertheless, it is important to acknowledge that, while adaptive neural networks or fuzzy logic control schemes can handle nonlinearity, they may come with the drawbacks of approximation errors and increased computational workload, leading to only semi-global stability. The energy consumption during the operation of an autonomous vehicle significantly impacts its overall performance. To enhance it, one approach is to employ the optimal control method. It focuses on identifying the most suitable control actions for the given system, aiming to achieve specific performance objectives.

Solving the algebraic Riccati equation (ARE) or the Hamilton–Jacobi–Bellman (HJB) one is essential for obtaining the optimal control solution. These equations capture the dynamics of the system and provide a mathematical framework for optimal control design. However, the majority of existing algorithms are designed for offline optimal control, requiring a priori knowledge of the system dynamics and the optimal control solution (Lewis and Vrabie, 2009; Gahinet *et al*., 1994). This limitation hinders the ability to achieve real-time control and adapt to dynamic and uncertain environments. To address this issue and achieve online solutions, a recent approach called adaptive dynamic programming (ADP) has been developed, leveraging the principles of reinforcement learning (RL) (Werbos, 1992; Wang, 2019; Xu *et al*., 2021; Wang and Ye, 2022). Unlike traditional offline methods, ADP enables the control system to learn and adapt from online data, making it well-suited for dynamic and uncertain environments. One of the key techniques in ADP is policy iteration (Lee *et al*., 2014; Abu-Khalaf and Lewis, 2005; Vamvoudakis and Lewis, 2010; Bhasin *et al*., 2013), which is frequently utilized to approximate the optimal equation solution. The iteration involves two main steps: evaluating an initial stabilizing control policy and then improving the policy iteratively. By evaluating the initial policy, the system gains an understanding of the current dynamics and learns to stabilize the control inputs. This is followed by policy improvement, which involves updating the control policy based on the current knowledge of the system dynamics. The ADP approach combines the benefits of RL and optimal control, allowing for online learning and adaptation while still achieving optimal control performance (Abu-Khalaf and Lewis, 2005). Because of the strong learning ability of ADP, it has been used to resolve decentralized control of large-scale nonlinear systems with uncertainties (Tan, 2018; Tan, 2019). This makes it particularly useful in applications where the system dynamics are uncertain or subject to change, such as in AVs or robotics (Zhang *et al*., 2021).

However, canonical actor–critic ADP (e.g., Vamvoudakis and Lewis, 2010; Bhasin *et al*., 2013) typically (i) requires an admissible stabilizing baseline policy to start the policy-evaluation step, and (ii) employs two networks (actor and critic) with iterative policy evaluation/improvement. These designs often guarantee only the UUB of critic/actor weights and incur extra computation/tuning for the actor and for policy-iteration scheduling. In contrast, the present work develops a critic-only, non-iterative scheme: the control is obtained analytically from the critic via the HJB stationary condition, and the critic is learned from data by an integral Bellman residual (Section 3), thereby removing both the actor NN and the need for an initial stabilizer.

This paper aims to establish an optimal control strategy for AVs to online detect the trajectory tracking error by using two online learning algorithms. Specifically, we propose an adaptive learning algorithm (ALA) and a finite-time adaptive learning algorithm (FTALA) that are actor-free and do not rely on an initial stabilizing policy. The policy is synthesized directly from the learned value function gradient, while the critic is updated from measured transitions without policy iteration. The proposed strategy eliminates the need for an actor NN and an initial stabilizing control policy. We first construct an error dynamic system by combining the kinematic equation of AVs and an ideal tracking trajectory. We design an adaptive learning algorithm that computes the control by solving the derived HJB equation online. To further relax the model requirement in learning, the ALA/FTALA employ an integral Bellman residual so that the learning step does not require the explicit drift $f_s(\cdot)$; moreover, a sliding-mode injection in the weight-update law ensures finite-time convergence of critic-weight error under PE while avoiding input chattering. To further relax the requirement of the system dynamics, a finite-time adaptive learning algorithm is designed, where the finite-time convergence can be ensured via adopting a sliding mode technique. The adaptive critic and control action in both online learning algorithms continuously and simultaneously update each other, eliminating the need for any iterative steps.

Compared with traditional actor–critic ADP (e.g., Vamvoudakis and Lewis, 2010), the main improvements are as follows: (a) no initial stabilizer is required because learning is non-iterative and critic-only; (b) *no actor NN* and hence fewer parameters/tuning and no policy-iteration overhead; (c) stronger convergence of the critic (exponential in the ALA and finite time in the FTALA under PE) versus the UUB-type guarantees commonly reported; and (d) the sliding action is injected in the parameter space (learning law) rather than in the plant input, avoiding chattering while retaining

robustness.

Consider the aforementioned facts, we highlight the major contributions of this paper below.

(i) This work proposes a pure critic learning scheme based on the HJB stationarity condition, which synthesizes control policies directly from critic gradients, completely eliminating the need for the actor NN in traditional ADP. The method employs a non-iterative learning mechanism that avoids policy evaluation/improvement iterations while requiring no initial stabilizing control policy, significantly simplifying the control architecture and reducing parameter tuning overhead.

(ii) An FTALA incorporating sliding mode techniques is designed, where the sliding action is injected into the weight update law rather than the plant input, ensuring finite-time convergence while avoiding input chattering. The algorithm is driven by integral Bellman residuals, eliminating the explicit requirement for drift function $f(s)$, further relaxing system dynamics dependencies and achieving enhanced robustness with faster convergence rates.

(iii) A learning method based on integral Bellman residuals computed from measured transition data is developed, enabling critic weights to achieve exponential convergence (ALA) or finite-time convergence (FTALA) under persistent excitation conditions, surpassing the uniform ultimate boundedness (UUB) limitations of traditional approaches. This mechanism is purely data-driven, requiring no complete system dynamics model, providing a more practical online optimization solution for autonomous vehicle trajectory tracking control.

The problem formulation is stated in Section 2. Section 3 shows the optimal controller design via the ALA and FTALA. The simulations are given in Section 4, followed by Section 5, which summarizes the paper.

2. Problem description

2.1. Dynamic model construction. This paper considers the kinematic equation of AVs as (Zhang *et al.*, 2021; Ma *et al.*, 2017)

$$\begin{cases} \dot{x}(t) = v_x(t)\cos(\theta(t)) - d_r w_x(t)\sin(\theta(t)), \\ \dot{y}(t) = v_x(t)\sin(\theta(t)) + d_r w_x(t)\cos(\theta(t)), \\ \dot{\theta}(t) = w_x(t). \end{cases} \tag{1}$$

For the kinematic equation (1), the variable $x(t)$ represents the horizontal position of the vehicle's mass center while $y(t)$ corresponds to the vertical position. The orientation of the vehicle is denoted by $\theta(t)$. Additionally, the longitudinal velocity of the mass center in the body-fixed frame is denoted as $v_x(t)$, while the yaw angular velocity about the Z-axis, which is orthogonal to the X-Y plane, is represented by $w_x(t)$. The straight-line distance from the mass center to the vehicle's rear axle is defined as d_r.

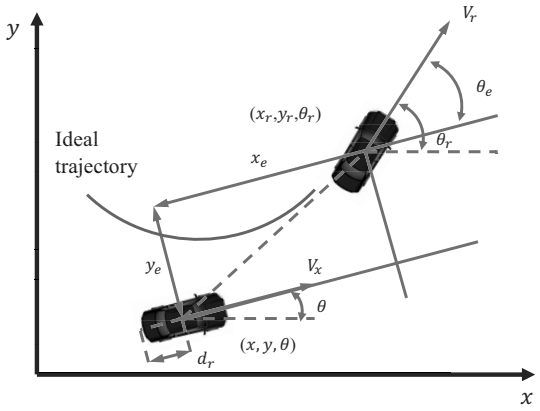

Fig. 1. AV's tracking trajectory.

The aim of this paper is to enable autonomous vehicles (AVs) to follow the desired trajectory. To achieve this, we assume that the ideal trajectory is defined by the fixed longitudinal velocity $v_r(t)$ and yaw angular velocity $w_r(t)$ of the vehicle. Consequently, the ideal trajectory dynamics can be expressed as

$$\begin{cases} \dot{x}_r(t) = v_r(t)\cos(\theta_r(t)) - d_r w_r(t)\sin(\theta_r(t)), \\ \dot{y}_r(t) = v_r(t)\sin(\theta_r(t)) + d_r w_r(t)\cos(\theta_r(t)), \\ \dot{\theta}_r(t) = w_r(t), \end{cases} \tag{2}$$

with $x_r(t)$, $y_r(t)$, and $\theta_r(t)$ representing the ideal horizontal and vertical positions, and vehicle orientation, respectively.

In this vehicle system, the tracking errors for the horizontal and vertical position as well as orientation are defined as $x_e(t)$, $y_e(t)$, and $\theta_e(t)$, respectively. Therefore, we derive the error vector as

$$\begin{bmatrix} x_e(t) \\ y_e(t) \\ \theta_e(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) & 0 \\ -\sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r(t) - x(t) \\ y_r(t) - y(t) \\ \theta_r(t) - \theta(t) \end{bmatrix}. \tag{3}$$

In this paper, we display the AV's free-body diagram and ideal trajectory as Fig. 1.

The primary goal is to seek the control actions (v_x and w_x) for AVs that have the ability to stabilize tracking errors.

Based on Fig. 1 and the tracking error (3), we can obtain the tracking error dynamics of the AV.

Lemma 1. *The tracking error dynamics are calculated as*

$$\dot{s}(t) = f(s(t)) + g(s(t))u(t), \tag{4}$$

where

$$f(t) = \begin{bmatrix} v_r(t)\cos(\theta_t(t)) \\ v_r(t)\sin(\theta_t(t)) \\ w_r(t) \end{bmatrix}, \quad s(t) = \begin{bmatrix} x_e(t) \\ y_e(t) \\ \theta_e(t) \end{bmatrix},$$

$$g(t) = \begin{bmatrix} y_e(t) & -1 \\ -x_e(t) - d_r & 0 \\ -1 & 0 \end{bmatrix}$$

and

$$u(t) = \begin{bmatrix} w_x(t) \\ v_x(t) \end{bmatrix}.$$

Proof.
Step 1. According to the tracking error (3), based on (1) and (2) we have

$$\begin{aligned} \dot{x}_e =& \cos(\theta)(\dot{x}_r - \dot{x}) + \sin(\theta)(\dot{y}_r - \dot{y}) \\ & - \sin(\theta)\dot{\theta}(x_r - x) + \cos(\theta)\dot{\theta}(y_r - y) \\ =& \cos(\theta_r - \theta_e)\dot{x}_r - \cos(\theta)[v_x\cos(\theta) - d_r w_x \sin(\theta)] \\ & + \sin(\theta_r - \theta_e)\dot{y}_r - \sin(\theta)[v_x\sin(\theta) \\ & + d_r w_x\cos(\theta)] - \sin(\theta)w_x(x_r - x) \\ & + \cos(\theta)w_x(y_r - y) \\ =& \dot{x}_r[\cos(\theta_r)\cos(\theta_e) + \sin(\theta_e)] + y_e w_x \\ & + \dot{y}_r[\sin(\theta_r)\cos(\theta_e) - \cos(\theta_r)\sin(\theta_e)] - v_x. \end{aligned} \tag{5}$$

From Fig. 1 and the modeling process, we have $\dot{x}_r\sin(\theta_r) = \dot{y}_r cos(\theta_e)$; then (5) becomes

$$\begin{aligned} \dot{x}_e =& y_e w_x + [v_r\cos(\theta_r) - d_r w_r\sin(\theta_r)]\cos(\theta_r)\cos(\theta_e) \\ & - v_x + [v_r\sin(\theta_r) + d_r w_r\cos(\theta_r)]\sin_(\theta)\cos(\theta_e) \\ =& y_e w_x - v_x + v_r cos(\theta_e). \end{aligned} \tag{6}$$

Step 2. Then, focusing on y_e, we have

$$\begin{aligned} \dot{y}_e =& -\sin(\theta)(\dot{x}_r - \dot{x}) + \cos(\theta)(\dot{y}_r - \dot{y}) \\ & - \cos(\theta)\dot{\theta}(x_r - x) - \sin(\theta)\dot{\theta}(y_r - y) \\ =& -\sin(\theta)(\theta_r - \theta_e)\dot{x}_r + \sin(\theta) \\ & \times (v_x\cos(\theta) - d_r w_x\sin(\theta)) \\ & + \cos(\theta)(\theta_r - \theta_e)\dot{y}_r - \cos(\theta) \\ & \times (v_x\sin(\theta) + d_r w_x\cos(\theta)) \\ & - \cos(\theta)w_x(x_r - x) - \sin(\theta)w_x(y_r - y) \\ =& -d_r w_x - \dot{x}_r[\sin(\theta_r)\cos(\theta_e) - \cos(\theta_r)\sin(\theta_e)] \\ & - x_e w_x + \dot{y}_r[\cos(\theta_r)\cos(\theta_e) + \sin(\theta_r)\sin(\theta_e)] \\ =& -d_r w_x - x_e w_x + v_r\sin(\theta_e). \end{aligned} \tag{7}$$

Step 3. According to (1) and (2), the tracking error dynamic can be calculated as

$$\dot{\theta}_e(t) = \dot{\theta}_r(t) - \dot{\theta}(t) = w_r(t) - w_x(t). \tag{8}$$

Hence, we have

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} v_r\cos(\theta_e) + y_e w_x - v_x \\ v_r\sin(\theta_e) - x_e w_x - d_r w_x \\ w_r - w_x \end{bmatrix}. \tag{9}$$

This completes the proof. ■

Remark 1. This study targets the learning/control architecture and its convergence-optimality properties for lane-keeping and path-following at low to moderate speeds. In this regime, the kinematic bicycle model with small slip angles and quasi-constant longitudinal speed captures the geometry of the tracking error (x_e, y_e, θ_e) that the ALA/FTALA regulate. Unmodeled dynamics (tire relaxation, actuator/drivetrain lags, load variations) act as bounded matched/unmatched disturbances in the error system; these are injected in simulations and are handled by critic-only, integral Bellman residual learning and by the finite-time weight update, without introducing input chattering. Importantly, our theoretical guarantees rely on persistent excitation of measured regressors rather than on a specific high-fidelity dynamic parameterization. Extending the approach to full vehicle dynamics is part of future work, but is not required to establish the present claims or the reported performance within the specified speed/curvature envelopes.

Uncertain attacking signals pose a challenge to the system (4) of AVs during autonomous driving and operation. Specifically, these signals take the form of denial of service (DoS) attacks, which are deliberately introduced into the tracking dynamic via communication networks. Thus, the system (4) of AVs can be rewritten as

$$\dot{s}(t) = f(s(t)) + g(s(t))u(t) + a_1(t), \tag{10}$$

with $u(t) = u_1(t) + u_2(t) + a_2(t)$ representing the actual system input acting on the vehicle, where $u_1(t)$ corresponds to the resilient tracking control policy and $u_2(t) = \begin{bmatrix} w_r(t) & v_r(t) \end{bmatrix}^T$ denotes the reference policy, $a_1(t)$ is sensor-side additive attack/disturbance in the error dynamics, and $a_2(t)$ is actuator-side attack on the input. The uncertain sensor attack signal is given by $a(t) = \begin{bmatrix} a_x(t) & a_y(t) & a_z(t) \end{bmatrix}^T$, while the actuator attack signal is expressed as $a_2(t) = \begin{bmatrix} a_{21}(t) & a_{22}(t) \end{bmatrix}^T$.

During the tracking driving process, the reference policy u_2 plays a crucial role in ensuring the adherence to the desired trajectory. It serves as a predefined reference for the vehicles. On the other hand, the resilient tracking control policy $u_1(t)$ is designed to minimize tracking errors and mitigate uncertainties.

Assumption 1. This paper assumes that the sensor and actuator attack signals are state-dependent and can be parameterized as $a_1(t) = \varpi(t)s(t)$ and $a_2(t) = \kappa\mu(s(t))$ for all $t \geq 0$. Following Ma *et al.* (2017), $\varpi(t)$ and $\mu(t)$ are unknown time-varying matrices satisfying $\|\varpi(t)\|_2 \leq c$ and $\|\mu(t)\|_2 \leq d$, where c and d are positive constants. Additionally, $\mu(\cdot)$ is an unknown but bounded nonlinear function.

To account for uncertain attacking signals in communication networks, we need to analyze the tracking error dynamics of AVs. For this purpose, we rewrite the autonomous driving system (10) as (Zhang *et al.*, 2021)

$$\begin{aligned}\dot{s}(t) =& f(s(t)) + g(s(t))(u_1(t) + u_2(t)) + \Lambda(t)\\ =& f_s(s(t)) + g(s(t))u_1(t) + \Lambda(t),\end{aligned} \tag{11}$$

with $f_s(s(t)) = f(s(t)) + g(s(t))u_2(t)$ being the desired dynamic part and $\Lambda = a_1(t) + g(t)a_2(t)$ the overall attacking signal. The reference policy $u_2(t)$ is determined based on the desired reference, which is typically predetermined in the driving system. In addition, $u_1(t)$ is the control which needs to be designed in this paper.

2.2. Optimal control. The purpose of the paper is to construct an optimal controller u in such a way that the investigated system state $s(t) \to 0, t \geq 0$, and it minimizes the following value function in a near optimal way:

$$V(s) = \int_t^\infty \left[s(\tau)^T Q s(\tau) + u_1(\tau)^T R u_1(\tau)\right] \mathrm{d}\tau, \tag{12}$$

with $Q > 0 \in \mathbb{R}^{n\times n}$ and $R > 0 \in \mathbb{R}^{m\times m}$ being the weight matrices of appropriate dimensions.

Then we can obtain the derivative of (12), i.e., $\dot{V}$, as

$$\dot{V}(s) = -s^T Q s - u_1^T R u_1. \tag{13}$$

To get the optimal control for the system (4) with (13), one defines the Hamiltonian function as

$$\begin{aligned}&H(s, u_1, V_s)\\ &= s^T Q s + u_1^T R u_1 + V_s^T \left[f_s(s) + g(s)u_1 + \Lambda\right],\end{aligned} \tag{14}$$

with $V_s = \partial V/\partial s$.

Then, we give the optimal value function as

$$V^*(s(t)) = \min_{u_1} V(s(t)). \tag{15}$$

Thus, the HJB equation is computed as

$$0 = \min_{u_1} H(s, u_1^*, V_s^*). \tag{16}$$

We set the equation $\partial H(s, u_1^*, V_s^*)/\partial u_1^*$; then we have the optimal control action,

$$u_1^*(s) = \frac{1}{2} R^{-1} g^T(s) V_s^*(s). \tag{17}$$

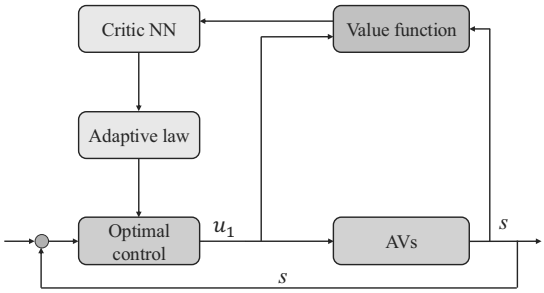

Fig. 2. Schematic of the proposed control system.

From (17) we have that, if the function $V_s(s)$ is known, the control u_1^* can be resolved directly. It is unfortunate that the function $V_s(s)$ is unknown, so the next task is how to solve the function $V_s(s)$.

3. Optimal controller design by using two online learning algorithms

In this section, we first use the single critic NN to reconstruct the value function (12); then the optimal control action (17) can be obtained based on the reconstructed value function. To get the optimal control solution, two online learning methods are presented. The first one is used to resolve the HJB equation via designing a novel adaptive law, where the full system dynamics should be known. To relax the requirement of the system dynamics, we then propose an FTALA. Both of these online learning methods avoid the use of an initial stabilising control policy and the actor NN. The control logic diagram based on ADP is given in Fig. 2.

The basic idea of the ADP scheme is to use a critic NN to approximate the optimal value function $V^*(s)$. In this case, we consider $V^*(s)$ as a continuous function (Vamvoudakis and Lewis, 2010; Chen and Herrmann, 2019). This can be approximated using a critic NN as

$$V^*(s) = W^T B(s) + \varepsilon(s), \tag{18}$$

with $W \in \mathbb{R}^l$ being the ideal critic NN weights, $B(s) \in \mathbb{R}^l$ the activation function, l the number of neurons, and ε the NN approximation error. Thus, we can obtain its derivative with respect to s as

$$V_s^*(s) = (\nabla B(s))^T W + \nabla\varepsilon(s), \tag{19}$$

with $\nabla B(s) = \partial B(s)/\partial s \in \mathbb{R}^{l\times 2n}$ and $\nabla\varepsilon(s) = \partial\varepsilon(s)/\partial s \in \mathbb{R}^{2n}$ being the derivative of the NN error with respect to s.

Then, we have the following assumptions (Lv *et al.*, 2019; Vamvoudakis and Lewis, 2010).

Assumption 2. The NN weights W, the activation function $B(s)$ and the NN error $\varepsilon(s)$ are bounded, i.e.,

$\|W\| \leq W_M$, $\|B(s)\| \leq B_M$, $\|\varepsilon(s)\| \leq \varepsilon_M$ for positive constants $W_M > 0$, $B_M > 0$, $\varepsilon_M > 0$. The derivative $\nabla B(s)$ of $B(s)$ and the derivative $\nabla\varepsilon(s)$ of $\varepsilon(s)$ are bounded, i.e., $\|\nabla B(s) \leq B_{Md}\|$, $\nabla\varepsilon(s) \leq \varepsilon_{Md}$ for positive constants $W_{Md} > 0$, $B_{Md} > 0$.

To estimate the unknown weight W, we can obtain the practical value function as

$$\hat{V}(s) = \hat{W}^T B(s), \tag{20}$$

with $\hat{W}$ being the estimate of W. Then, we can get the approximated solution of the HJB equation (16) as

$$\hat{V}_s(s) = (\nabla B(s))^T \hat{W}. \tag{21}$$

Considering the critic NN (18) and (19), the ideal optimal control can be given as

$$u_1^*(s) = -\frac{1}{2} R^{-1} g^T(s)[(\nabla B(s))^T W + \nabla\varepsilon(s)], \tag{22}$$

and its practical optimal control action as

$$\hat{u}_1(s) = -\frac{1}{2} R^{-1} g^T(s)(\nabla B(s))^T \hat{W}. \tag{23}$$

Next, we will design two online learning methods to learn online the unknown critic NN weights $\hat{W}$, whose convergence can guarantee the system state $s \to 0$. Because the proposed methods have strong convergence, the actor NN used in the existing results is removed.

3.1. Adaptive learning for optimal control. We write the tracking-error dynamics in input-affine form $\dot{s} = f_s(s) + g(s)u$, where $f_s(\cdot)$ denotes the *drift* (autonomous part) and $g(\cdot)$ the *input map*. In Section 3.1 (ALA) we explicitly use $f_s(\cdot)$ and $g(\cdot)$ to form the regressor Ξ; in Section 3.2 (FTALA) the *learning* step will rely on the integral Bellman residual and therefore *does not require an explicit* $f_s(\cdot)$.

Substituting (19) into (16), we have the following HJBE:

$$s^T Q s + \hat{W}\{\nabla B[f_s(s) + g(s)u_1 + \Lambda]\} + \varepsilon_{\text{HJBE}} = 0, \tag{24}$$

where $\varepsilon_{\text{HJBE}} = \nabla\varepsilon^T[f_s(s) + g(s)u_1 + \Lambda]$ is the residual error.

To realize the online solution of the HJB equation (24), we rewrite it as

$$\begin{cases} \Xi = \nabla B(s)[f_s(s) + g(s)u_1 + \Lambda], \\ \Phi = s^T Q s + u_1^T R u_1. \end{cases} \tag{25}$$

Then, the HJBE (24) can be rewritten as

$$\Phi = -W^T \Xi - \varepsilon_{\text{HJBE}}. \tag{26}$$

From (26) we have that only the critic NN weight W is unknown, thus we can develop an online learning algorithm to learn the unknown weight W. To this end, we define the auxiliary variables $P_1 \in \mathbb{R}^{l\times l}$ and $Q_1 \in \mathbb{R}^l$ as

$$\begin{cases} \dot{P}_1 &= -\ell P_1 + \Xi\Xi^T, \qquad P_1(0) = 0, \\ \dot{Q}_1 &= -\ell Q_1 + \Xi\Phi^T, \qquad Q_1(0) = 0, \end{cases} \tag{27}$$

with ℓ being the positive constant. Then, we have the solution of (27):

$$\begin{cases} P_1 = \displaystyle\int_0^t e^{-\ell(t-\tau)}\Xi(\tau)\Xi(\tau)^T, \\ Q_1 = \displaystyle\int_0^t e^{-\ell(t-\tau)}\Xi(\tau)\Phi(\tau)^T. \end{cases} \tag{28}$$

For P_1 and Q_1 defined in (28), we define the auxiliary variable $M_1 \in \mathbb{R}^l$ as

$$M_1 = P_1\hat{W} + Q_1. \tag{29}$$

Considering (26) and (28), we know $Q_1 = -P_1 W + \nu_1$, with $\nu_1 = -\int_0^t e^{-\ell(t-\tau)}\varepsilon_{\text{HJBE}}(\tau)\Xi(\tau)\,\mathrm{d}\tau$ being a bounded variable, i.e., $\|\nu_1\| \leq \varepsilon_{\nu 1}$ for $\varepsilon_{\nu 1} > 0$. Thus, from (27)–(29) we have that

$$M_1 = -P_1\tilde{W} + \nu_1, \tag{30}$$

where $\tilde{W} = W - \hat{W}$ is the estimation error of the NN weight.

Then, an adaptive law can be designed to learn $\hat{W}$ as

$$\dot{\hat{W}} = \Gamma_1 M_1, \tag{31}$$

where $\Gamma_1 > 0$ is the learning gain.

The convergence of the learning algorithm (31) can be explained by verifying the positive definiteness of matrix P_1, as specified in (28).

Lemma 2. (Lv *et al.*, 2019) *The variable P_1 provided in (28) is positive definite for the persistent excitation variable Ξ given in (26).*

Theorem 1. *Consider the critic NN with the learning law (31) when the variable Ξ in (26) is persistent excitation. Then the critic NN weight error $\tilde{W}$ exponentially converges to 0.*

Proof. According to Lemma 2, we have that P_1 is positive definite for the persistent excitation variable Ξ, i.e., the minimum eigenvalue $\lambda_{\min}(P_1) > \sigma > 0$. Then, we choose a Lyapunov function as $V_1 = -\frac{1}{2}\tilde{W}^T\Gamma_1\tilde{W}$; thus, we can obtain its derivative as

$$\dot{V}_1 = \tilde{W}\Gamma_1\tilde{W} = -\tilde{W}^T P_1\tilde{W} + \tilde{W}\nu_1. \tag{32}$$

This further shows that

$$\dot{V}_1 = -\tilde{W}^T P_1\tilde{W} + \tilde{W}^T\nu_1 \leq -\|\tilde{W}\|(\sigma\|\tilde{W}\| - \varepsilon_{\nu 1}). \tag{33}$$

Therefore, we have that the estimated error $\tilde{W}$ is convergent to the small set given by $\Omega : \{\tilde{W} \mid \|\tilde{W}\| \leq \varepsilon_{\nu 1}/\sigma\}$, whose size is determined by the NN error ε and the excitation level σ. In this ideal case, i.e., $\varepsilon_{\text{HJBE}} = 0$, the estimation error $\tilde{W}$ exponentially converges to 0. ■

3.2. Finite-time adaptive learning for optimal control. Although the learning algorithm proposed in Section 3.1 can obtain the online optimal control action, the full system dynamics are required. Moreover, the convergence rate of the critic NN weight is not fast. Hence, this subsection will develop an FTALA. To this end, we define a Bellman function (Chen and Herrmann, 2019) as

$$V(s(t-T)) = \int_{t-T}^{t} r(s(\tau), u_1(\tau))\,\mathrm{d}\tau + V(s(t)), \quad (34)$$

with $r(s(\tau), u_1(\tau)) = s^T Q s + u_1^T R u_1$, $T > 0$ being the sample time. The system dynamics $f_s(s)$ and $g(s)$ are not taken into account in the Bellman function (34).

By the chain rule, $V(s(t)) - V(s(t-T)) = \int_{t-T}^{t} \nabla_s V(s;\hat{W})^{\top}\big(f_s(s) + g(s)u_1\big)\,\mathrm{d}\tau$, while (34) computes the *same* quantity from measured transitions and the applied input. Hence, the *learning/update law* in the FTALA is driven by a *data-driven* Bellman residual and does not require an explicit drift $f_s(\cdot)$. For control synthesis, we still use the HJB stationary condition $u_1(s) = -\frac{1}{2}R^{-1}g(s)^{\top}\nabla_s V(s;\hat{W})$ (plus u_r if applicable); thus, $g(s)$ appears only in the analytic policy, not in the learning step.

The proposed ALA and FTALA are critic-only, by using the HJB stationary condition. The policy is obtained directly from the critic, so no actor NN is needed. Learning is non-iterative, driven by the integral Bellman residual (34) computed from data, so no initial stabilizer is required. In the FTALA, a sliding term in the weight update ensures finite-time weight-error convergence under PE and avoids input chattering, unlike typical actor–critic ADP.

In order to propose a design method of value function approximation, our approach involves reconstructing the function using a critic NN as

$$V(s) = W^T B_1(s) + \varepsilon_1(s), \quad (35)$$

where $B_1(s) : \mathbb{R}^n \to \mathbb{R}^N$ is the activation function and N denotes the number of neurons, $W \in \mathbb{R}^N$ stands for the critic NN weights, and $\varepsilon_1(s) \in \mathbb{R}$ is the critic NN estimation error. Based on the results of Chen and Herrmann (2019), we assume the errors $\varepsilon_1(s)$ and $\nabla\varepsilon_1(s)$ of $\varepsilon_1(s)$ are bounded.

The update of the critic is achieved by inserting the value function (35) into the Bellman function (34), which yields

$$\underbrace{\int_{t-T}^{t} r(s(\tau), u_1(\tau))\,\mathrm{d}\tau}_{k(s,u_1)} + \underbrace{W^T B_1(s(t)) - W^T B_1(s(t-T))}_{W^T \Delta B_1(t)} = -\varepsilon_s, \quad (36)$$

where $k(s, u_1)$ is the integral term, $\Delta B_1(t) = B_1(s(t)) - B_1(s(t-T))$, and the error $\varepsilon_s = \varepsilon_1(s(t)) - \varepsilon_1(s(t-T))$ of (36) is bounded. Then, a finite-time adaptive law will be designed to estimate online the unknown weight W in (36).

According to (36), it is indicated that the weight W of the unknown critic NN is expressed in linear parameterized form. Consequently, it can be estimated 'directly' by utilizing a learning method recently proposed by Lv *et al.* (2019), which is derived via the extracted estimation error. Most current ADP schemes focus on ensuring the uniform ultimate boundedness (UUB) of the estimated NN weights rather than achieving convergence. In this paper, we will propose a novel adaptive law that guarantees the convergence of the estimated weight $\hat{W}$ to the true weight W. This enhanced convergence property eliminates the need for an actor NN and allows the calculated optimal control based on the critic NN to approach the ideal optimal solution. To accomplish this objective, we introduce two variables, $P_2 \in \mathbb{R}^{N\times N}$ and $Q_2 \in \mathbb{R}^N$, as

$$\begin{cases} \dot{P}_2 = \ell_2 P_2 + \Delta B_1(t)\Delta B_1(t)^T,\ P_2(0) = 0, \\ \dot{Q}_2 = \ell_2 Q_2 + \Delta B_1(t) k(s, u_1),\ Q_2(0) = 0, \end{cases} \quad (37)$$

with the design parameter $\ell_2 > 0$. Then, the solution of (37) can be derived as

$$\begin{cases} P_2 = \displaystyle\int_0^t e^{-\ell_2(t-\tau)}\Delta B_1(\tau)\Delta B_1(\tau)^T, \\ Q_2 = \displaystyle\int_0^t e^{-\ell_2(t-\tau)}\Delta B_1(\tau)k(\tau)\,\mathrm{d}\tau. \end{cases} \quad (38)$$

The estimation of the value function (35) can be obtained as follows:

$$\hat{V}(s) = \hat{W}^T B_1(s), \quad (39)$$

where $\hat{V}$ and $\hat{W}$ are the estimates of V and W, respectively.

Now, a finite-time adaptive law that is based on the sliding mode technique (Na *et al.*, 2015) can be developed as

$$\dot{\hat{W}} = -\Gamma_2 P_2 \frac{M_2}{\|M_2\|}, \quad (40)$$

where $M_2 = P_2\hat{W} + Q_2 \in \mathbb{R}^N$ and $\Gamma_2 > 0$ is the learning gain.

Based on the auxiliary vector M_2, the derived adaptive law (4) is driven by the error $\tilde{W}$, which is calculated using the measurable system input u and output s. The main objective of this new algorithm is to ensure the convergence of the estimated weight $\hat{W}$ to the unknown weight W. Therefore, the proposed adaptive law (40) differs significantly from existing online ADP algorithms (see, e.g., the work of Vamvoudakis and Lewis (2010) or Wang (2019)) and the references therein) that rely on gradient based methods and only retain the UUB of $\hat{W}$. In this paper, we demonstrate that this new adaptive law can achieve rapid and guaranteed convergence. Consequently, the use of the actor NN as employed in the existing ADP literature is avoided.

Remark 2. The FTALA places the sliding action in the critic learning law (40), not in the plant input, ensuring finite-time weight-error convergence under PE without input chattering. It is critic-only, non-iterative, and driven by the integral Bellman residual (34), thus it *does not require an explicit drift* $f_s(s)$ for learning; $g(s)$ appears only in the analytic policy $u_1(s) = -\frac{1}{2}R^{-1}g(s)^\top \nabla_s V(s;\hat{W})$ (plus u_r if applicable). Unlike finite-time controllers (reaching but non-optimal) and actor–critic ADP (iterative, stabilizer-dependent), the FTALA attains cost optimality with finite-time critic learning.

Remark 3. The adaptive learning algorithms proposed in this paper offer several advantages over traditional gradient-based methods. Firstly, they eliminate the need for an actor NN and an initial stabilizing control policy, simplifying the control structure. Then, the adaptive critic is trained using a novel adaptive law, which ensures strict convergence of the critic NN weights without relying on gradient methods. This results in a more efficient and robust learning process that can adapt to changing dynamics in real-time, making it particularly suitable for autonomous vehicles operating in dynamic and uncertain environments. Moreover, the FTALA incorporates a sliding mode technique to guarantee faster convergence within a finite-time frame, enhancing the responsiveness and reliability of the control system.

Lemma 3. *If variable $\Delta_1(s)$ satisfies the PE condition, then the variable P_2 provided in (38) is positive definite, i.e., the minimum eigenvalue $\lambda_{\min}(P_2) > \sigma_2 > 0$. The following section will focus on the convergence analysis of the finite-time adaptive law (40) that we developed.*

Theorem 2. *The convergence of the estimation error $\tilde{W} = W - \hat{W}$ to 0 in a finite-time can be guaranteed by considering the satisfaction of the PE condition for the finite-time adaptive law (40) with $\Delta B_1(s)$ in (37).*

Proof. Our first step is to analyze the boundness of M_2. By considering (38) and the system states $s(t)$ and $s(t-T)$, we can establish an upper bound for the variable P_2 under the condition $\alpha_1 > 0$, ensuring that $\lambda_{\max}(P_2) \le \alpha_1$ holds. Taking $k(s,u_1)$ given in (36) into (38) results in $Q_2 = -P_2 W + \theta$, with $\theta = \int_0^t e^{-\ell_2(t-\tau)}\Delta B_1(\tau)\varepsilon_s(\tau)\,\mathrm{d}\tau$ being the bounded by constant $\alpha_2 > 0$ as the error ε_s is bounded. Then we have

$$M_2 = -P_2\tilde{W} + \theta. \tag{41}$$

Since $\Delta B_1(s)$ meets the necessary PE condition, applying Lemma 3 allows us to establish the invertibility of P_2, given its symmetric positive definiteness. Therefore, using $P_2^{-1}M_2 = -\tilde{W} + P_2^{-1}M_2\theta$ for selecting a Lyapunov function, we can directly deduce $P_2^{-1}M_2$. Consequently, the derivative of $P_2^{-1}M_2$ is calculated as

$$\frac{\partial}{\partial t}(P_2^{-1}M_2) = -\dot{\tilde{W}} + \frac{\partial P_2^{-1}}{\partial t}\theta + P_2^{-1}\dot{\theta} = \dot{\hat{W}} + \theta_s, \tag{42}$$

with $\theta_s = -P_2^{-1}\dot{P}_2P_2^{-1}\theta + P_2^{-1}\dot{\theta}$ being bounded for bounded θ, i.e., $\|\theta_s\| \le \alpha_3$ for the positive constant α_3. According to the above facts, we have that P_2^{-1} is bounded due to $\lambda_{\min}(P_2^{-1}) > \alpha_2$ and $\lambda_{\max}(P_2^{-1}) < \alpha_1$, thus the bounds of P_2^{-1} can be defined as $\lambda_{\min}(P_2) > \alpha_1$ and $\lambda_{\max}(P_2) < 1/\sigma_2$. Then, we define the time-varying Lyapunov function as

$$V_2 = \frac{R_1}{2}(P_2^{-1}M_2)^T\Gamma_2^{-1}P_2^{-1}M_2, \tag{43}$$

where $R_1 > 0$ is a design parameter. Then, we have the derivative of V_2 as

$$\begin{aligned}\dot{V}_2 &= R_1M_2^TP_2^{-1}\Gamma_2^{-1}(\dot{\hat{W}} + \theta_s)\\ &= R_1M_2^TP_2^{-1}\Gamma_2^{-1}(\Gamma P_2\frac{M}{\|M\|} + \theta_s)\\ &\le \alpha_4\sqrt{V_2},\end{aligned} \tag{44}$$

with

$$\alpha_4 = (\sigma_2 - R_1\alpha_3\lambda_{\max}(\Gamma_2^{-1}))\sqrt{2/\lambda\max(\Gamma_2^{-1})}$$

being a positive constant for $0 < R_1 < \sigma_2/(\lambda_{\max}(\Gamma_2^{-1})\alpha_3)$. Based on the results of Chen and Herrmann (2019), we have $R_1 = 0$ and $M_2 = 0$ in finite time $t_1 = 2\sqrt{R_1(0)}/\alpha_4 > 0$. Then we obtain $\varepsilon_s(s) \neq 0$, $M_2 = 0$ for $\varepsilon_1(s) \neq 0$. This indicates that $\tilde{W} = P_2^{-1}\theta_s$, and so $\|\tilde{W}\| \le \alpha_2/\sigma_2$ is bounded after finite time t_1. In the ideal case, for $\varepsilon_1(s) = 0$, we have $\varepsilon_s(s)$, $M_2 = 0$ and $\theta = \theta_s = 0$. This shows that the weight error $\tilde{W}$ will convergence to zero in finite time t_1. ∎

4. Simulations

In this section, we will apply the two learning algorithms to a simulated autonomous driving system (4) for vehicles.

This system allows us to examine the tracking error dynamic function as

$$\dot{s}(t) = f(s(t)) + g(s(t))(u_1(t) + u_2(t)) + \Lambda(t), \quad (45)$$

with

$$s(t) = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix}, \quad u_1(t) = \begin{bmatrix} w_x \\ y_x \end{bmatrix},$$

$$d_r = 1.2(m), \quad f(t) = \begin{bmatrix} v_r(t)\cos(\theta_e(t)) \\ v_r(t)\sin(\theta_e(t)) \\ w_r(t) \end{bmatrix},$$

$$g(t) = \begin{bmatrix} y_e(t) & -1 \\ -x_e(t) - d_r & 0 \\ -1 & 0 \end{bmatrix},$$

and $\Lambda(t) = a_1(t) + g(t)a_2(t)$.

To complete the simulation, we choose the initial system states as $\begin{bmatrix} 1.2 & -1.2 & -0.5 \end{bmatrix}^T$; the desired longitudinal velocity and yaw angular velocity can be selected as $v_r(t) = 0.5$, $w_r(t) = 0$. Then, a critic NN in (18) is introduced to learn the solution of (23). For both learning algorithms, the initial weight vector is $W(0) = \text{rands}(6, 1)$.

Case 1. $\Lambda(t) = 0$, i.e., $a_1 = 0$ and $a_2 = 0$. To benchmark the proposed methods against a modern sliding-mode controller, we additionally implement the super-twisting control (STC) method by Moreno and Osorio (2012) as well as Xiong *et al*. (2021) under the same Case 1 setting. The sliding map is

$$\sigma = S\, s, \quad S = \begin{bmatrix} 0 & 1.5 & 0 \\ 1.5 & 0 & 1.5 \end{bmatrix},$$

and the super-twisting law is

$$u_1 = u_{\text{eq}} - K_1\, |\sigma|^{1/2}\, \text{sgn}(\sigma) + z, \qquad \dot{z} = -K_2\, \text{sgn}(\sigma),$$

with $K_1 = \text{diag}(5, 5)$ and $K_2 = \text{diag}(8, 8)$. The plant parameters and the initial condition follow the learning-based cases: $d_r = 1.2$, $v_r = 0.5$, $w_r = 0$, and $s(0) = [1.2, -1.2, -0.5]^\top$. The sampling time, horizon and reference trajectory are kept identical to the ALA/FTALA for fairness. The STC results are now reported separately in Figs. 10–12 (instead of being merged into Figs. 4–6).

To this end, we select the regressor of the critic NN as $B(s) = [s_1^2, s_1 s_2, s_1 s_3, s_2^2, s_2 s_3, s_3^2]^T$ for the ALA, the positive definite matrices in the value function are chosen as $Q = 350 I_{3\times 3}$, $R = 1$, and the learning gains are set as $\ell = 60$, $\Gamma_1 = 600$. Besides, we choose the positive definite matrices in the value function as $Q = 350 I_{3\times 3}$, $R = 1$, and the learning gains are set as $\ell_2 = 100$, $\Gamma_2 = 700$ in the FTALA. We choose the regressor of the critic NN as $B_1(s) = [s_1^2, s_1 s_2, s_1 s_3, s_2^2, s_2 s_3, s_3^2]^T$.

Table 1. Integral metrics.

Scenario	Controller	IAE	IAU
Lane keeping	STC	1.14	7.89
Lane keeping	ALA	1.22	7.59
Lane keeping	FTALA	1.07	7.51

The convergence of critic NN weights under the ALA and FTALA algorithms is depicted in Fig. 3. This illustrates that the estimated weights have the capability to converge to specific values. Therefore, these learning outcomes provide empirical evidence supporting the validity of the convergence property outlined in Theorems 1 and 2. In particular, the FTALA converges faster than the ALA (Fig. 3(b) vs. 3(a)), and this faster weight learning translates into a faster tracking transient in Fig. 4(b) relative to Fig. 4(a). With the added STC benchmark (Figs. 7–9), the system also achieves successful tracking; however, STC exhibits a larger initial control effort and a slower decay of the tracking errors before settling, consistent with input-side sliding injection (see Fig. 8). Regarding control inputs (Fig. 9), the FTALA produces the smoothest and smallest-amplitude actuation, ALA is intermediate, while STC shows the largest initial actuation and higher total variation. Overall, the FTALA attains the fastest transient and highest tracking accuracy among the three, while the ALA improves over a purely learning-free baseline.

In addition to settling time, overshoot, quadratic cost J, and control-input total variation, we report two integral metrics:

$$\text{IAE} = \int_0^T \|s(t)\|\, \text{d}t, \quad \text{IAU} = \int_0^T \|u(t)\|_1\, \text{d}t,$$

where $u(t) \in \mathbb{R}^m$ is the control applied. The result can be found in Table 1. For the lane keeping case, the FTALA attains the lowest IAE (1.07) and IAU (7.51), indicating tighter tracking with the least cumulative control effort. STC achieves a moderate IAE (1.14) but the largest IAU (7.89), consistent with stronger input injection. The ALA exhibits the largest IAE (1.22) and an intermediate IAU (7.59). Overall, the ranking is FTALA $<$ STC $<$ ALA in IAE, and FTALA $<$ ALA $<$ STC in IAU.

Remark 4. STC is robust, model-free, and finite-time, whereas the FTALA additionally enforces HJB optimality to minimize the tracking error and control effort, and adapts online to time-varying uncertainties/attacks. Thus the FTALA offers an optimal robust adaptive framework; simulations show faster convergence, better tracking, and lower cumulative control effort.

Case 2. $\Lambda(t) \neq 0$, i.e., $a_1 \neq 0$ and $a_2 \neq 0$. To further show the robustness of the proposed FTALA, we consider the uncertainties introduced by Jin *et al*. (2017) as

$$a_1(t) = -(0.75 + 0.15\sin(2.5t)), \qquad t \geq 0, \quad (46)$$

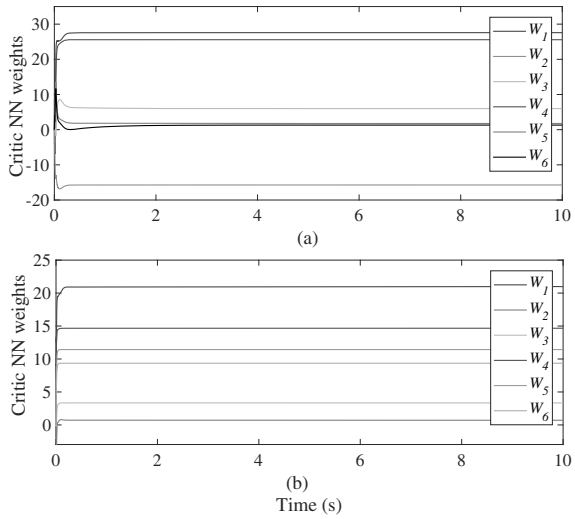

Fig. 3. Convergence of the critic NN weights: ALA (a), FTALA (b).

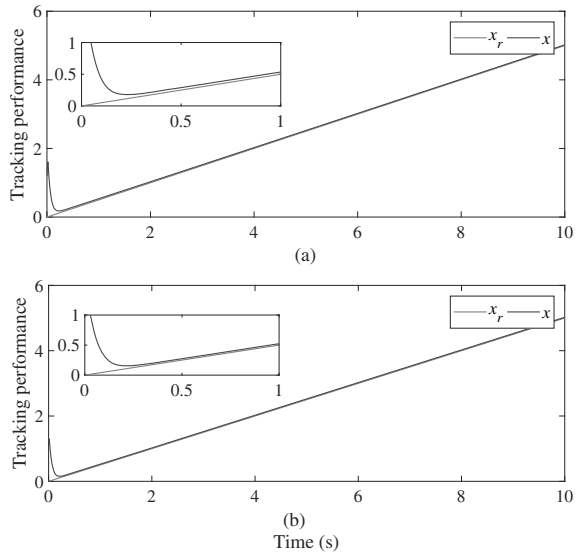

Fig. 4. Tracking performance under the proposed optimal control: ALA (a), FTALA (b).

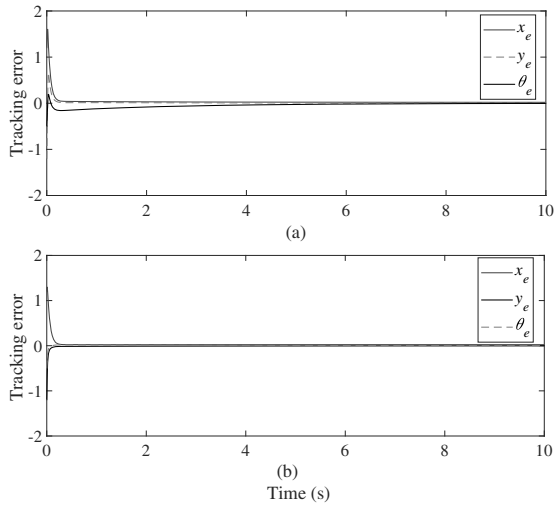

Fig. 5. Tracking error for the system (4): ALA (a), FTALA (b).

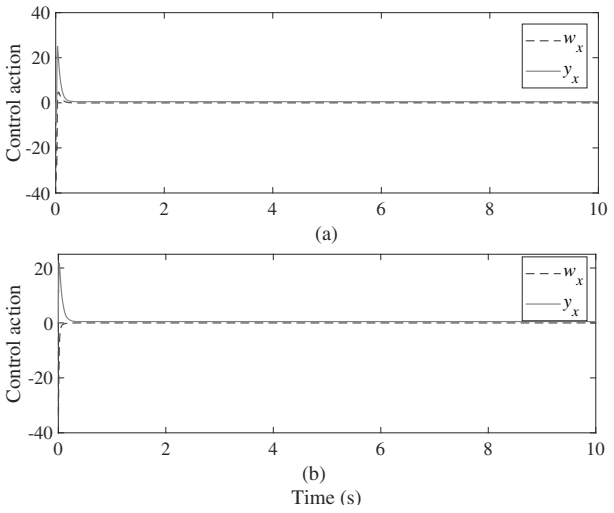

Fig. 6. Control action for the AV: ALA (a), FTALA (b).

and

$$a_2(t) = \left[1, 1\right] 0.005\cos(2.5t) + \left[0.1\cos(2t), 0.5\sin(t)\right]^T \times 0.2\sin(x_e(t))\cos(y_e(t)). \quad (47)$$

In this case, the positive definite matrices in the value function are given as $Q = 60I_{3\times3}$, $R = 1$, and the learning gains as $\ell_2 = 280$, $\Gamma_2 = 600$ in the FTALA; other parameters are the same as in the previous case.

Figure 10 shows the convergence of critic NN weights. Compared with Fig. 3(b), the critic NN weights in Fig. 10 converge faster because the system more easily satisfies the PE condition in the presence of interference. The corresponding tracking errors and control inputs are shown in Fig. 11.

In order to verify if the designed optimal controller can enable the AV to track complex trajectory signals, we set the desired longitudinal velocity and yaw angular velocity as $v_r(t) = 0.5\sin(2t)$ and $w_r(t) = 0$. In the learning, the gains can be given as $\ell_2 = 118$ and $\Gamma_2 = 600$. The learning results can be found in Fig. 12, which shows the convergence of the critic NN weights. Based on these learning results, the optimal controller for the AV can be designed, thus the AV can accurately track the desired trajectory as shown in Fig. 13(a), which presents the tracking performance of the AV. To further

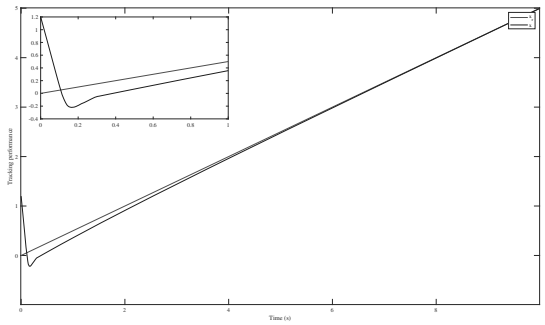

Fig. 7. Tracking performance under the STC baseline.

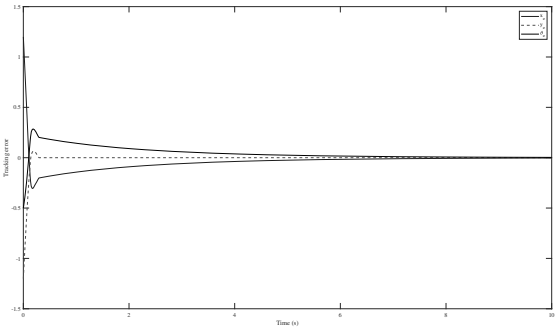

Fig. 8. Tracking errors under the STC baseline.

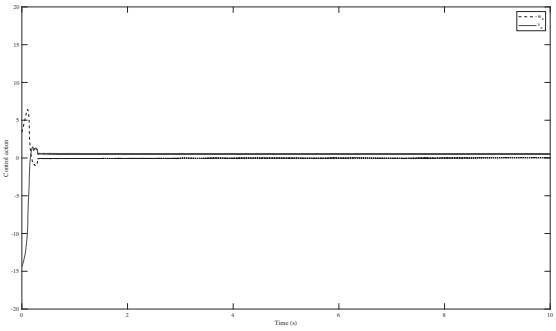

Fig. 9. Control actions under the STC baseline.

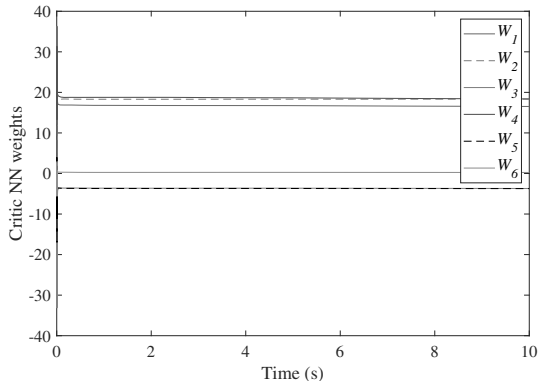

Fig. 10. Convergence of the critic NN weights.

show the tracking accuracy, we give the tracking error in Fig. 13(b). It is worth noting that, comparing the tracking results in Fig. 13 with those in Fig. 11, it can be seen that Fig. 13 shows a faster convergence speed. This is because it tracks complex signals, and the system can satisfy the PE condition in a short period of time, thus enabling a rapid response. The above results confirm the efficacy of the optimal control approach proposed in this study. By employing online learning algorithms, the designed control strategy enables the system output, represented as x, to accurately track the desired trajectory, even in the presence of uncertainties in the system model.

5. Conclusion

This paper presented a new optimal control based trajectory tracking error detection for AVs by using two learning schemes, i.e., the ALA and FTALA. In both online learning algorithms, the adaptive critic and control action interact with each other in a continuous and simultaneous manner, eliminating the need for any iterative steps. This approach also eliminates the requirement of an actor neural network and an initial stabilizing control policy. As for the ALA, it provides us withe an online solution by resolving the derived HJB equation with full system dynamic information. To further relax the requirement of the system dynamics, i.e., where the system drift is not required since it can directly online update the Bellman equation, the FTALA is presented. Moreover, the finite-time convergence can be guaranteed via adopting a sliding mode technique in the FTALA. Finally, both online learning algorithms are applied to control AVs, and the simulations show their effectiveness and practicality feasibility. Future work will focus on the optimal control of AVs with completely unknown system dynamics.

Acknowledgment

This work was supported by the Key Laboratory of Traffic Information and Safety of Anhui Higher Education Institutes (JTX202501), the Key R&D Program of Shandong Province (2025CXPT071), and the Road Traffic Safety Scientific Research and Innovation Team Project (2023AH010064).

References

Abu-Khalaf, M. and Lewis, F.L. (2005). Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach, *Automatica* **41**(5): 779–791.

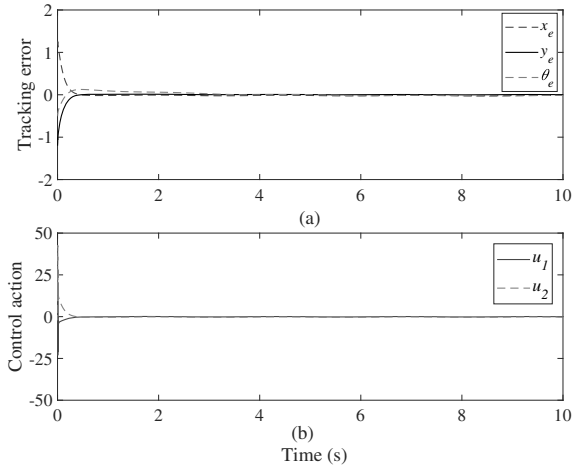

Fig. 11. Tracking error and control action for the AV.

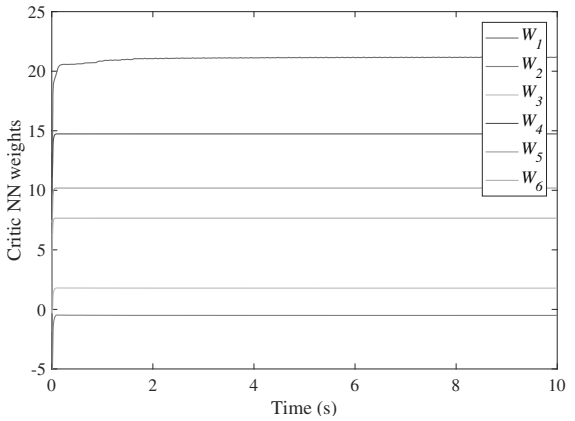

Fig. 12. Convergence of critic NN weights.

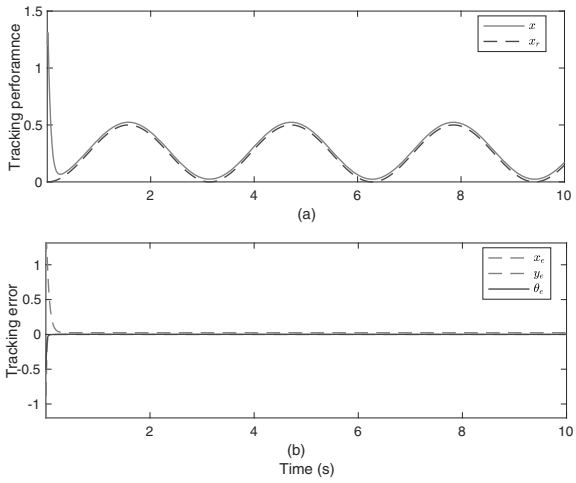

Fig. 13. Tracking performance of the AV.

Bhasin, S., Kamalapurkar, R., Johnson, M., Vamvoudakis, K.G., Lewis, F.L. and Dixon, W.E. (2013). A novel actor–critic–identifier architecture for approximate optimal control of uncertain nonlinear systems, *Automatica* **49**(1): 82–92.

Chen, A.S. and Herrmann, G. (2019). Adaptive optimal control via continuous-time q-learning for unknown nonlinear affine systems, *2019 IEEE 58th Conference on Decision and Control (CDC), Nice, France*, pp. 1007–1012.

Chen, J., Shuai, Z., Zhang, H. and Zhao, W. (2020). Path following control of autonomous four-wheel-independent-drive electric vehicles via second-order sliding mode and nonlinear disturbance observer techniques, *IEEE Transactions on Industrial Electronics* **68**(3): 2460–2469.

Ding, C., Ding, S., Wei, X. and Mei, K. (2022). Output feedback sliding mode control for path-tracking of autonomous agricultural vehicles, *Nonlinear Dynamics* **110**(3): 2429–2445.

Elsisi, M. and Ebrahim, M.A. (2021). Optimal design of low computational burden model predictive control based on SSDA towards autonomous vehicle under vision dynamics, *International Journal of Intelligent Systems* **36**(11): 6968–6987.

Faryadi, S. and Mohammadpour Velni, J. (2021). A reinforcement learning-based approach for modeling and coverage of an unknown field using a team of autonomous ground vehicles, *International Journal of Intelligent Systems* **36**(2): 1069–1084.

Gahinet, P., Nemirovskii, A., Laub, A.J. and Chilali, M. (1994). The LMI control toolbox, *Proceedings of the 1994 33rd IEEE Conference on Decision and Control, Lake Buena Vista, USA*, pp. 2038–2041.

Hu, C., Jing, H., Wang, R., Yan, F. and Chadli, M. (2016). Robust h_∞ output-feedback control for path following of autonomous ground vehicles, *Mechanical Systems and Signal Processing* **70**: 414–427.

Jin, X., Haddad, W.M. and Yucelen, T. (2017). An adaptive control architecture for mitigating sensor and actuator attacks in cyber-physical systems, *IEEE Transactions on Automatic Control* **62**(11): 6058–6064.

Lakhekar, G.V., Waghmare, L.M. and Roy, R.G. (2019). Disturbance observer-based fuzzy adapted s-surface controller for spatial trajectory tracking of autonomous underwater vehicle, *IEEE Transactions on Intelligent Vehicles* **4**(4): 622–636.

Lee, J.Y., Park, J.B. and Choi, Y.H. (2014). Integral reinforcement learning for continuous-time input-affine nonlinear systems with simultaneous invariant explorations, *IEEE Transactions on Neural Networks and Learning Systems* **26**(5): 916–932.

Lewis, F.L. and Vrabie, D. (2009). Reinforcement learning and adaptive dynamic programming for feedback control, *IEEE Circuits and Systems Magazine* **9**(3): 32–50.

Lv, Y., Ren, X. and Na, J. (2019). Adaptive optimal tracking controls of unknown multi-input systems based on

nonzero-sum game theory, *Journal of the Franklin Institute* **356**(15): 8255–8277.

Ma, G., Ghasemi, M. and Song, X. (2017). Integrated powertrain energy management and vehicle coordination for multiple connected hybrid electric vehicles, *IEEE Transactions on Vehicular Technology* **67**(4): 2893–2899.

Moreno, J.A. and Osorio, M. (2012). Strict Lyapunov functions for the super-twisting algorithm, *IEEE Transactions on Automatic Control* **57**(4): 1035–1040.

Na, J., Mahyuddin, M.N., Herrmann, G., Ren, X. and Barber, P. (2015). Robust adaptive finite-time parameter estimation and control for robotic systems, *International Journal of Robust and Nonlinear Control* **25**(16): 3045–3071.

Peng, Y., Chen, J. and Ma, Y. (2019). Observer-based estimation of velocity and tire-road friction coefficient for vehicle control systems, *Nonlinear Dynamics* **96**(1): 363–387.

Tan, L.N. (2018). Distributed h_∞ optimal tracking control for strict-feedback nonlinear large-scale systems with disturbances and saturating actuators, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **50**(11): 4719–4731.

Tan, L.N. (2019). Event-triggered distributed h_∞ constrained control of physically interconnected large-scale partially unknown strict-feedback systems, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **51**(4): 2444–2456.

Vamvoudakis, K.G. and Lewis, F.L. (2010). Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem, *Automatica* **46**(5): 878–888.

Wang, D. (2019). Robust policy learning control of nonlinear plants with case studies for a power system application, *IEEE Transactions on Industrial Informatics* **16**(3): 1733–1741.

Wang, X., Lv, B., Wang, K. and Zhang, R. (2023). ASTS: Autonomous switching of task-level strategies, *International Journal of Applied Mathematics and Computer Science* **33**(4): 553–568, DOI: 10.34768/amcs-2023-0040.

Wang, X. and Ye, X. (2022). Consciousness-driven reinforcement learning: An online learning control framework, *International Journal of Intelligent Systems* **37**(1): 770–798.

Werbos, P. (1992). Approximate dynamic programming for real-time control and neural modeling, *in* D.A. White and D.A. Sofge (Eds.), *Handbook of Intelligent Control*, Van Nostrand Reinhold, New York, Chapter 13.

Xiong, X., Kamal, S. and Jin, S. (2021). Adaptive gains to super-twisting technique for sliding mode design, *Asian Journal of Control* **23**(1): 362–373.

Xu, N., Niu, B., Wang, H., Huo, X. and Zhao, X. (2021). Single-network ADP for solving optimal event-triggered tracking control problem of completely unknown nonlinear systems, *International Journal of Intelligent Systems* **36**(9): 4795–4815.

Yan, Z., Song, B., Zhang, Y., Zhang, K., Mao, Z. and Hu, Y. (2018). A rotation-free wireless power transfer system with stable output power and efficiency for autonomous underwater vehicles, *IEEE Transactions on Power Electronics* **34**(5): 4005–4008.

Zhang, K., Su, R., Zhang, H. and Tian, Y. (2021). Adaptive resilient event-triggered control design of autonomous vehicles with an iterative single critic learning framework, *IEEE Transactions on Neural Networks and Learning Systems* **32**(12): 5502–5511.

Peng-fei Feng holds an MS degree and is currently a professor and the dean of the School of Intelligent Transportation Modern Industry, Hefei University of Technology, China. His research interests include intelligent traffic control and traffic flow control.

Bingyi Jia received his BS degree in mechanical design and manufacture and automation from the Shandong University of Science and Technology, Qingdao, China, in 2023. He is currently pursuing a PhD degree with the College of Mechanical and Electronic Engineering, Shandong University of Science and Technology. His research interests include adaptive control, data-driven and robust control.

Hui-qing Jin holds a PhD degree and is currently a professor and PhD supervisor at Anhui Sanlian University, China. His research interests include road traffic accident prevention engineering, traffic accident epidemiology, and driver psychology and behavior analysis.

Guang Wang holds an MS degree in mechanical engineering from the Beijing University of Technology, China. His research interests include the design and development of harmonic drive reducers and other key components in robotic systems.

Received: 10 April 2025
Revised: 18 August 2025
Accepted: 25 September 2025