International Journal of Robotics and Automation, Vol. 31, No. 6, 2016

LATERAL CONTROL FOR AUTONOMOUS LAND VEHICLES VIA DUAL HEURISTIC PROGRAMMING

Zhenhua Huang,* Chuanqiang Lian,* Xin Xu,* and Jian Wang*

Abstract

In this paper, to improve the lateral control performance of autonomous land vehicles (ALVs) under different longitudinal velocities, a novel learning-based lateral control method based on the dual heuristic programming (DHP) algorithm is presented. A new way to calculate the lateral control errors based on the geometric relationship between the vehicle and the path is utilized. To minimize the lateral control errors of ALV, the lateral control problem is modelled as a Markov decision problem (MDP). To approximate the optimal control policy of the MDP, a learning controller based on the DHP algorithm is designed, where the critic is used to approximate the derivative of the value function and the actor is utilized to improve the control policy based on the output of the critic. The inputs of the critic and actor are comprised of the lateral control error and the vehicle's longitudinal velocity, which makes the proposed method be effectively adaptive to different longitudinal velocities. Simulation results demonstrate the proposed lateral control scheme has advantages over widely used lateral control methods, such as the pure pursuit, PD control and Stanley methods.

Key Words

Autonomous land vehicle, lateral control, dual heuristic programming, learning control

1. Introduction

Autonomous land vehicles (ALVs) have attracted enormous attention in the past decades since the key technologies of ALVs, such as intelligent sensing, planning and control [1]–[5], can be widely applied in intelligent transportation systems, driver assistance systems, military tasks and other fields. Motion control, which mainly fulfils the tracking task of driving the vehicle to achieve the desired path and the desired speed, is an important part among the major technologies in ALVs. Challenges still exist in designing a high-precision motion controller under different longitudinal velocities. When designing a motion controller of ALVs, the control problem is commonly decoupled into the lateral control problem and the longitudinal control problem, which play important roles in guaranteeing the ALVs' stability and safety.

Lateral control methods can be classified into two major categories: geometric methods and model-based methods. In geometric methods, control rules are usually designed by studying the geometric relationship between the vehicle and the reference path. Geometric methods mainly comprise the pure pursuit, the vector pursuit and the Stanley method [1], [6]–[8]. The pure pursuit method and its variations have been widely studied and employed due to their easy implementation and robustness to larger errors. In the pure pursuit method and its variations, a look-ahead distance to measure error ahead of the vehicle is often utilized. In [1] and [6], fixed look-ahead distances were used and in [2] and [9], look-ahead distances were dynamically tuned with the vehicle's speeds. The vector pursuit method makes use of the screw theory and is similar to the pure pursuit method [7]. The Stanley method was presented in [8] and successfully used in the DARPA Grand Challenge [10]. The Stanley controller is a nonlinear feedback function of the cross track error. However, some issues still exist in these geometric methods. For instance, in the pure pursuit method and its variations, the look-ahead distance is usually tuned empirically to adapt different speeds. Moreover, the cutting corner problem easily occurs at a high speed for the reason that the look-ahead distance also increases with the vehicle's velocities. The Stanley method is more suitable for lower speeds and may lead to overshoots. In addition, the desired tracking path needs to have continuous curvatures and there is also an empirical parameter to be adjusted for precise control.

Model-based lateral control methods use kinematic or dynamics models of the vehicle to design lateral controllers. In [11], a lateral control approach was proposed for Ackerman-like vehicles. By empirically adjusting the look-ahead distance, the control strategy can adapt to both low and high driving speeds. In [8], a simple kinematic vehicle model was used to design the lateral controller, which can track a path precisely under normal driving scenarios. In addition, there are some other lateral control methods

* College of Mechatronics and Automation, National University of Defense Technology, Changsha 410073, PR China, e-mail: xinxu@nudt.edu.cn, zhenhuahuang10@gmail.com, wzdslcq@163.com, jian_wang@nudt.edu.cn

Recommended by Prof. Chaomin Luo
(DOI: 10.2316/Journal.206.2016.6.206-4878)

such as proportion-integral-derivative (PID) control [4], [12], fuzzy control [13], [14], sliding-mode control [15], [16] and model predictive control [17], [18]. Similar to geometric methods, most of aforementioned methods depend on handcrafted parameters to be adaptive for various driving speeds, such as in [8], [11], [19]–[20]. Therefore, designing an adaptive controller for various driving speeds is significant for the lateral control of an autonomous vehicle.

In recent years, many researchers focus on the design of lateral controllers based on learning methods for ALV. Some supervised learning methods were also used for solving the lateral control problem and the longitudinal problem [21], [22], whereas supervised learning requires teacher signals which to some extent limits its further applications on the lateral control of ALV. Reinforcement learning (RL) [23]–[26] is a class of machine learning methods for solving Markov decision problems (MDPs) without utilizing teacher signals. In recent years, the research works on RL have been integrated with the approximate dynamic programming (ADP) community [27]–[29], which is to solve MDPs with large or continuous state and action spaces. RL and ADP have been shown to be very effective techniques to solve optimal control problems with unknown or partly known model information. There have been some works about employing RL and ADP to design lateral controllers for ALVs [30]–[33]. Oh *et al.* presented a dynamic control approach based on adaptive heuristic critic (AHC) [30] which can work with high driving speeds and large curvature paths. In [31], neural fitted Q (NFQ) iteration was used to design a lateral controller with a discrete and empirical action space which can make a toy car track the desired road. In most works mentioned above, the cost function was directly approximated. However, a potential problem exists that the approximated cost function is not smooth because the derivative of the cost function is not approximated [34].

Dual heuristic programming (DHP), as a classical type of ADP methods, has attracted much attention in the last decades [35]. The DHP algorithm utilizes a critic network to approximate the derivative of the cost function, which has been shown to be more beneficial to learn an optimal or near-optimal control policy than approximating the cost function itself [34]. In this paper, a DHP-based lateral control method for ALV is proposed. First, the lateral posture error as the input of the DHP learning algorithm is calculated based on the geometric relationship between the vehicle and the desired path. Second, an MDP model, including the definition of the state space, action space and the cost function for the path tracking task, is established. Third, in DHP, a critic module is used to approximate the derivative of the value function and an actor module is utilized to obtain the near-optimal lateral control policy based on the outputs from the critic. Simulation results demonstrate the proposed scheme can well adapt to various vehicle velocities and has advantages over other popular methods, such as the pure pursuit, proportion-derivative (PD) control and Stanley methods.

The main contributions in this paper include: (i) a novel learning-based lateral control method based on DHP is presented to improve the lateral control performance for ALVs under different longitudinal velocities; (ii) the performance evaluation and comparisons with other popular lateral control methods illustrate the effectiveness and advantages of the proposed method.

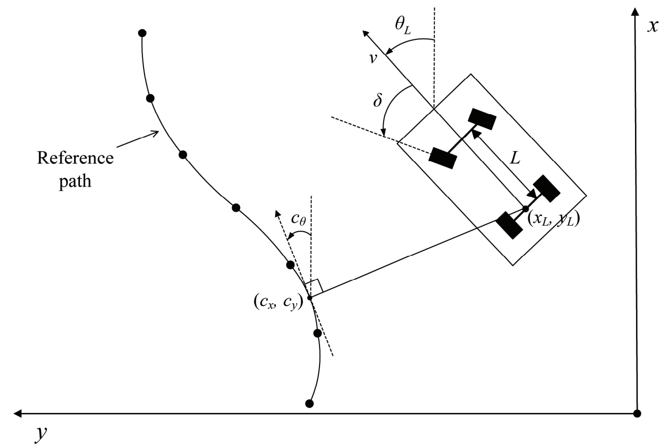

Figure 1. The kinematic model of ALVs and the local reference path in the local reference frame.

The rest of this paper is organized as follows. In Section 2, the vehicle kinematic model, geometric model of the local reference path and lateral posture error model are given. Section 3 establishes an MDP model and designs a lateral controller based on the DHP algorithm. In Section 4, simulation results are provided to demonstrate the validity of the proposed scheme. Section 5 draws conclusions.

2. Problem Formulation and Lateral Error Modelling

For an ALV, the local reference path is always determined by the perception and decision-making system in the local reference frame. Assume that path planning is implemented at time k_0, and the local reference frame is the current vehicle-body reference frame. Its origin is the centre of the vehicle's rear wheels. The positive direction of x-axis is the forward direction of the vehicle, and the positive direction of y-axis is towards to the left of the positive direction of x. Define Δk as the planning time, and then at time $(k_0 + \Delta k)$ the position and heading angle have changed compared with the posture at time k_0. Figure 1 shows the kinematic model of ALV and the local reference path in the local reference frame at time $(k_0 + \Delta k)$. In Fig. 1, (x_L, y_L) indicates the coordinates of the vehicle in the local reference frame, θ_L denotes the heading angle, the black dots on the reference path denote the planned points, (c_x, x_y) stands for the point on the reference path nearest to the vehicle, and c_θ is the angle between the path tangent at (c_x, c_y) and the x-axis. The kinematic model of the vehicle in the local reference frame is:

$$\dot{q_L} = [\dot{x}_L \ \dot{y}_L \ \dot{\theta}_L]^T = [v\cos\theta_L \ \ v\sin\theta_L \ \ v\tan\delta/L]^T \qquad (1)$$

To obtain a continuous and smooth reference path model, the quadratic polynomial fitting method is employed and the geometric model of the reference path is given as follows:

$$\begin{cases} y_p(k) = a_2 x_p^2(k) + a_1 x_p(k) + a_0 \\ \theta_p(k) = \arctan[2a_2 x_p(k) + a_1] \end{cases} \tag{2}$$

where $(x_p(k), y_p(k))$ is the coordinates of the point on the reference path and $\theta_p(k)$ is the path tangent at $(x_p(k), y_p(k))$. a_0, a_1 and a_2 are the coefficients of the constant term, the simple term and the quadratic term, respectively. Define the lateral control error as:

$$q_e = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta_L & \sin\theta_L & 0 \\ -\sin\theta_L & \cos\theta_L & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_x - x_L \\ c_y - y_L \\ c_\theta - \theta_L \end{bmatrix} \tag{3}$$

where (c_x, c_y) denotes the point on the reference path nearest to the vehicle, and c_θ is the angle between the path tangent at (c_x, c_y) and the x-axis. Thus we have:

$$(c_x, c_y) = \arg \min_{(x_p, y_p) \in P} \left\{ \sqrt{(x_p - x_L)^2 + (y_p - y_L)^2} \right\} \tag{4}$$

where P is the set of all points on the reference path. Equation (4) indicates that the vector $(c_x - x_L, c_y - y_L)$ and the path tangent at (c_x, c_y) are orthogonal. Then we can get:

$$(c_x - x_L) + (c_y - y_L)\frac{dc_y}{dc_x} = 0 \tag{5}$$

Combining with (2), (5) can be rewritten as:

$$\begin{aligned} &(c_x - x_L) + (a_2 c_x^2 + a_1 c_x + a_0 - y_L)(2a_2 c_x + a_1) \\ &= 2a_2 c_x^3 + 3a_1 a_2 c_x^2 + (a_1^2 + 2a_2(a_0 - y_L) + 1)c_x \\ &+ a_1(a_0 - y_L) - x_L = 0 \end{aligned} \tag{6}$$

The solution of the cubic equation (6) is obtained and we set:

$$\begin{cases} c_x = f_1(x_L, y_L) \\ c_y = a_2 c_x^2 + a_1 c_x + a_0 = f_2(x_L, y_L) \\ c_\theta = \arctan[2a_2 c_x + a_1] = f_3(x_L, y_L) \end{cases} \tag{7}$$

where the functions f_1, f_2, f_3 can be determined *via* (2) and (6).

Then the lateral error model is expressed as follows:

$$\dot{q}_e = \begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \dot{\theta}_L \begin{bmatrix} -\sin\theta_L & \cos\theta_L & 0 \\ -\cos\theta_L & -\sin\theta_L & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_x - x_L \\ c_y - y_L \\ c_\theta - \theta_L \end{bmatrix} + \begin{bmatrix} \cos\theta_L & \sin\theta_L & 0 \\ -\sin\theta_L & \cos\theta_L & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{c}_x - \dot{x}_L \\ \dot{c}_y - \dot{y}_L \\ \dot{c}_\theta - \dot{\theta}_L \end{bmatrix} \tag{8}$$

Substituting (1) and (3) into (8), yielding:

$$\dot{q}_e = \begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \frac{v\tan\delta}{L} \begin{bmatrix} e_y \\ -e_x \\ 0 \end{bmatrix} + \begin{bmatrix} \cos\theta_L \dot{c}_x + \sin\theta_L \dot{c}_y - v \\ -\sin\theta_L \dot{c}_x + \cos\theta_L \dot{c}_y \\ \dot{c}_\theta - v\frac{\tan\delta}{L} \end{bmatrix} \tag{9}$$

where $\dot{c}_x$, $\dot{c}_y$ and $\dot{c}_\theta$ can be calculated by (1) and (7), as follows:

$$\begin{cases} \dot{c}_x = \frac{\partial f_1}{\partial x_L} v\cos\theta_L + \frac{\partial f_1}{\partial y_L} v\sin\theta_L \\ \dot{c}_y = \frac{\partial f_2}{\partial x_L} v\cos\theta_L + \frac{\partial f_2}{\partial y_L} v\sin\theta_L \\ \dot{c}_\theta = \frac{\partial f_3}{\partial x_L} v\cos\theta_L + \frac{\partial f_3}{\partial y_L} v\sin\theta_L \end{cases} \tag{10}$$

Substituting (10) into (9), the lateral error model can be expressed by:

$$\dot{q}_e = \begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} \frac{\tan\delta}{L} e_y + g_1(x_L, y_L, \theta_L) - 1 \\ -\frac{\tan\delta}{L} e_x + g_2(x_L, y_L, \theta_L) \\ -\frac{\tan\delta}{L} + g_3(x_L, y_L, \theta_L) \end{bmatrix} v \tag{11}$$

where

$$\begin{cases} g_1(x_L, y_L, \theta_L) = \frac{\partial f_1}{\partial x_L}\cos^2\theta_L \\ \quad + \frac{1}{2}\frac{\partial f_1}{\partial y_L}\sin 2\theta_L + \frac{1}{2}\frac{\partial f_2}{\partial x_L}\sin 2\theta_L + \frac{\partial f_2}{\partial y_L}\sin^2\theta_L \\ g_2(x_L, y_L, \theta_L) = -\frac{1}{2}\frac{\partial f_1}{\partial x_L}\sin 2\theta_L - \frac{\partial f_1}{\partial y_L}\sin^2\theta_L \\ \quad + \frac{\partial f_2}{\partial x_L}\cos^2\theta_L + \frac{1}{2}\frac{\partial f_2}{\partial y_L}\sin 2\theta_L \\ g_3(x_L, y_L, \theta_L) = \frac{\partial f_3}{\partial x_L}\cos\theta_L + \frac{\partial f_3}{\partial y_L}\sin\theta_L \end{cases} \tag{12}$$

The motion control of the autonomous vehicle is always decoupled into two parts [6], [8]: longitudinal control and lateral control. In this paper, only the lateral control is considered based on the assumption that the vehicle speed was invariable during each learning episode. We also assume the speed of the autonomous vehicle can be changed randomly among different learning episodes so as to make the learned control policy suitable for different reference paths under various speeds.

3. Design of the DHP-based Lateral Controller

In this section, an MDP model for the lateral control of an ALV is first established, which is the basis for employing the DHP algorithm. Then the procedures of designing a DHP-based lateral controller are presented.

3.1 MDP Model for the Lateral Control of an ALV

The DHP algorithm can be utilized to solve an optimal control problem when the control problem is modelled as an MDP. In this section, the details about how to establish an MDP model for the lateral control of an ALV are introduced.

An MDP model is consisted of a quadruple $\{S, U, f, r\}$, where S is the state space, U is the action space, $f : S \times U \mapsto X$ is the state transition function, and $r : X \times U \times X \mapsto \mathbb{R}$ is the reward function.

To obtain smaller lateral control error q_e for different reference paths and vehicle speeds, the state space S is chosen as $[q_e, v]^T$, where v is the vehicle speed.

Define the action u as $\frac{\tan\delta}{L}$. Then for a constant longitudinal velocity, combining with (11), the following state transition model is obtained:

$$s(k+1) = \begin{bmatrix} e_x(k) \\ e_y(k) \\ e_\theta(k) \\ v(k) \end{bmatrix} + T_s v(k) \begin{bmatrix} g_1(k) - 1 \\ g_2(k) \\ g_3(k) \\ 0 \end{bmatrix} + T_s v(k) \begin{bmatrix} e_y(k) \\ -e_x(k) \\ -1 \\ 0 \end{bmatrix} u(k) \quad (13)$$

where T_s is the interval and $g_i(k)$ denotes $g_i(x_L(k), y_L(k), \theta_L(k))$ with $i = 1, 2, 3$.

As the control objective to control the vehicle to track the reference path as closely as possible, the reward function is set to be:

$$r(s(k)) = Q_1 e_x^2(k) + Q_2 e_y^2(k) + Q_3 e_\theta^2(k) \quad (14)$$

where Q_1, Q_2 and Q_3 are positive constants.

3.2 Lateral Controller Design Based on DHP

After establishing the MDP model, the DHP algorithm is employed to design a lateral controller. The cost function is defined as:

$$V(s(k)) = \sum_{k=0}^{\infty} r(s(k)) \quad (15)$$

where $r(s(k))$ is the reward function defined in (14). According to the Bellman's optimality principle, the optimal cost function $V^*(s(k))$ satisfies the following Hamilton–Jacobi–Bellman (HJB) equation:

$$V^*(s(k)) = \min_{u(k)}\{r(s(k)) + V^*(s(k+1))\} \quad (16)$$

The optimal control $u^*(k)$ satisfies:

$$u^*(k) = \arg\min_{u(k)}\{r(s(k)) + V^*(s(k+1))\} \quad (17)$$

According to (17), the optimal control $u^*(k)$ should satisfy the first-order necessary condition as follows:

$$\frac{\partial(r(s(k)) + V^*(s(k+1)))}{\partial u^*(k)} = \left\{\frac{\partial s(k+1)}{\partial u^*(k)}\right\}^T \frac{\partial V^*(s(k+1))}{\partial s(k+1)} = 0 \quad (18)$$

Let $\lambda^*(s(k)) = \partial V^*(s(k))/\partial s(k)$, we have:

$$\begin{aligned} \lambda^*(s(k)) &= \frac{\partial(r(s(k)) + V^*(s(k+1)))}{\partial s(k)} \\ &= \frac{\partial r(s(k))}{\partial s(k)} + \left\{\frac{\partial s(k+1)}{\partial s(k)}\right\}^T \frac{\partial V^*(s(k+1))}{\partial s(k+1)} \\ &\quad + \left\{\frac{\partial u^*(k)}{\partial s(k)}\right\}\left\{\frac{\partial s(k+1)}{\partial u^*(k)}\right\}^T \frac{\partial V^*(s(k+1))}{\partial s(k+1)} \end{aligned} \quad (19)$$

Substituting (18) into (19), yielding:

$$\lambda^*(s(k)) = \frac{\partial r(s(k))}{\partial s(k)} + \left\{\frac{\partial s(k+1)}{\partial s(k)}\right\}^T \lambda^*(s(k+1)) \quad (20)$$

Figure 2 shows the architecture of the DHP-based lateral control system. The actor is to generate suitable control u. The critic is to evaluate the control performance under current control signals. In this paper, Critic #1 has the same structure and parameters as Critic #2 and the error model is defined in (13). The critic and actor modules are designed with three-layer back propagation (BP) neural networks.

3.2.1 The Critic

The task of the critic network is to approximate the derivative of the cost function defined in (19). The structure of the critic network is given as:

$$\hat{\lambda}(s(k)) = w_c^T \sigma(\nu_c^T s(k)) \quad (21)$$

where ν_c denotes the weight from the input layer to the hidden layer, w_c denotes the weight from the hidden layer to the output layer and $\sigma(\cdot)$ is the sigmoid function.

According to (20), the temporal difference (TD) error is defined as:

$$\delta_{TD}(k) = \hat{\lambda}(s(k)) - \left\{\frac{\partial r(s(k))}{\partial s(k)} + \left\{\frac{\partial \hat{s}(k+1)}{\partial s(k)}\right\}^T \hat{\lambda}(\hat{s}(k+1))\right\} \quad (22)$$

where $\partial\hat{s}(k+1)/\partial s(k)$ can be calculated *via* (13).

Then the weight update rules for the critic network are given by:

$$\begin{aligned} w_c^{[i+1]} &= w_c^{[i]} - \alpha\delta_{TD}(k)\frac{\partial\hat{\lambda}(s(k))}{\partial w_c^{[i]}} \\ \nu_c^{[i+1]} &= \nu_c^{[i]} - \alpha\delta_{TD}(k)\frac{\partial\hat{\lambda}(s(k))}{\partial \nu_c^{[i]}} \end{aligned} \quad (23)$$

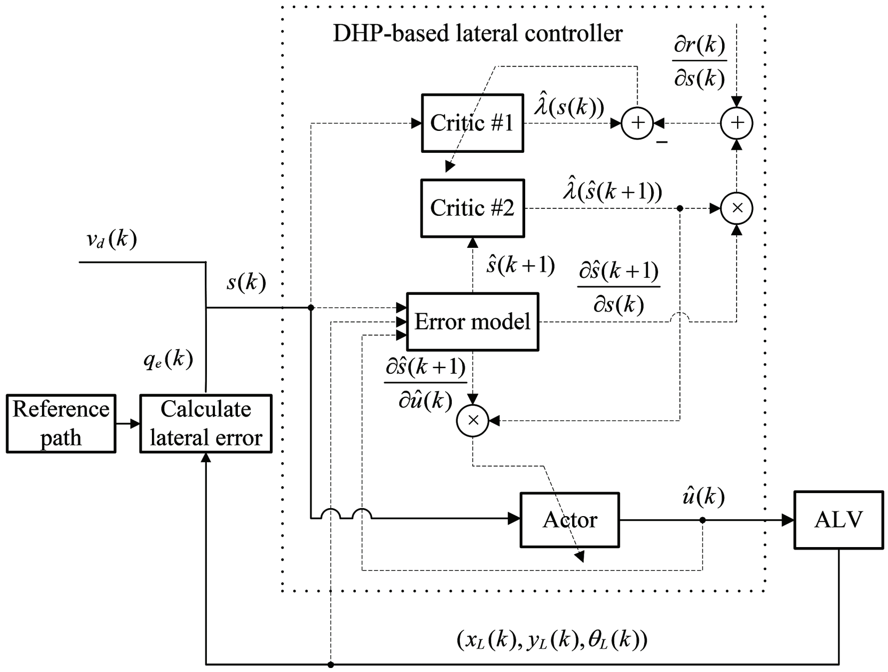

Figure 2. Architecture of the DHP-based lateral control system. The solid lines denote signal paths and the dashed lines denote data paths.

where i is the iteration number, $0<\alpha\leq 1$ is the step size and $\partial\hat{\lambda}(s(k))/\partial w_c^{[i]}, \partial\hat{\lambda}(s(k))/\partial \nu_c^{[i]}$ can be calculated by (21).

3.2.2 The Actor

The actor is to approximate the optimal control u^*. Suppose the control signal subjects to the following constraint:

$$\|u(k)\| \leq \bar{U} \tag{24}$$

where $\bar{U}$ is a positive constant. The approximation structure of the actor is given as:

$$\hat{u}(k) = \bar{U}\Phi(w_a^T\sigma(\nu_a^T s(k))) \tag{25}$$

where the term $w_a^T\sigma(\nu_a^T s(k))$ is implemented *via* a three-layer BP neural network. ν_a is the weight from the input layer to the hidden layer, w_a is the weight from the hidden layer to the output layer and $\sigma(\cdot)$ is the sigmoid function. $\Phi(\cdot)$ is a monotonic odd function and subjects to $\|\Phi(\cdot)\| \leq 1$. In addition, its first derivative is bounded by a constant B.

Then according to (18), we define:

$$\varepsilon(k) = \left\{\frac{\partial \hat{s}(k+1)}{\partial \hat{u}(k)}\right\}^T \hat{\lambda}(\hat{s}(k+1)) \tag{26}$$

where $\partial\hat{s}(k+1)/\partial\hat{u}(k)$ can be computed *via* (13).

Then the actor's weight update rules are:

$$\begin{aligned} w_a^{[i+1]} &= w_a^{[i]} - \beta\varepsilon(k)\frac{\partial\hat{u}(k)}{\partial w_a^{[i]}} \\ \nu_a^{[i+1]} &= \nu_a^{[i]} - \beta\varepsilon(k)\frac{\partial\hat{u}(k)}{\partial \nu_a^{[i]}} \end{aligned} \tag{27}$$

where i is the iteration number, $0<\beta\leq 1$ is the step size and $\partial\hat{u}(k)/\partial w_a^{[i]}, \partial\hat{u}(k)/\partial\nu_a^{[i]}$ can be calculated by (25).

4. Simulation Results

In this section, two driving courses, *i.e.*, the lane change course and the figure eight course shown in Fig. 3, are utilized to evaluate the performance of the proposed scheme. In Fig. 3(a), the lane change course requires the vehicle to perform a single lane change on a two-lane road. It is a procedure to evaluate the capability of tracking a straight path and changing lanes. The eight course shown in Fig. 3(b) is to evaluate the capability of tracking a varying curvature path.

In the design of the DHP-based controller, the critic and actor neural networks are constructed in (21) and (25) with structures 4-12-4 and 4-12-1, respectively. The sigmoid function is $\sigma(x)=1/(1+\exp(-x))$ and the monotonic odd function in (25) is chosen as $\Phi(x)=\tanh(x)$. The control boundary $\bar{U}$ in (24) is set to be 0.2 based on the real vehicle dynamics. The initial weights $w_c^{[0]}$, $\nu_c^{[0]}$, $w_a^{[0]}$ and $\nu_a^{[0]}$ are all selected randomly from -0.5 to 0.5. The step sizes α and β are set to be 0.6 and 0.4, respectively. The boundary of the posture error *Ebound* is chosen as $[-3,3;-3,3;-\pi/2,\pi/2;0,70]$, which means the states e_x, e_y, e_θ and v are constrained within the following intervals: $[-3,3]$ (m), $[-3,3]$ (m), $[-\pi/2,\pi/2]$ (rad), $[0,70]$ (km/h). The maximum failed number $N_{\max}$ is 200. The number M of the points on the local path is set to be 5. The interval T_s is 0.05 s and the wheelbase L is 2.85 m. The reward function r is defined in (14) with $Q_1=0.2$, $Q_2=1.6$ and $Q_3=0.2$.

In the simulation, the proposed method is compared with pure pursuit, PD control as well as the Stanley

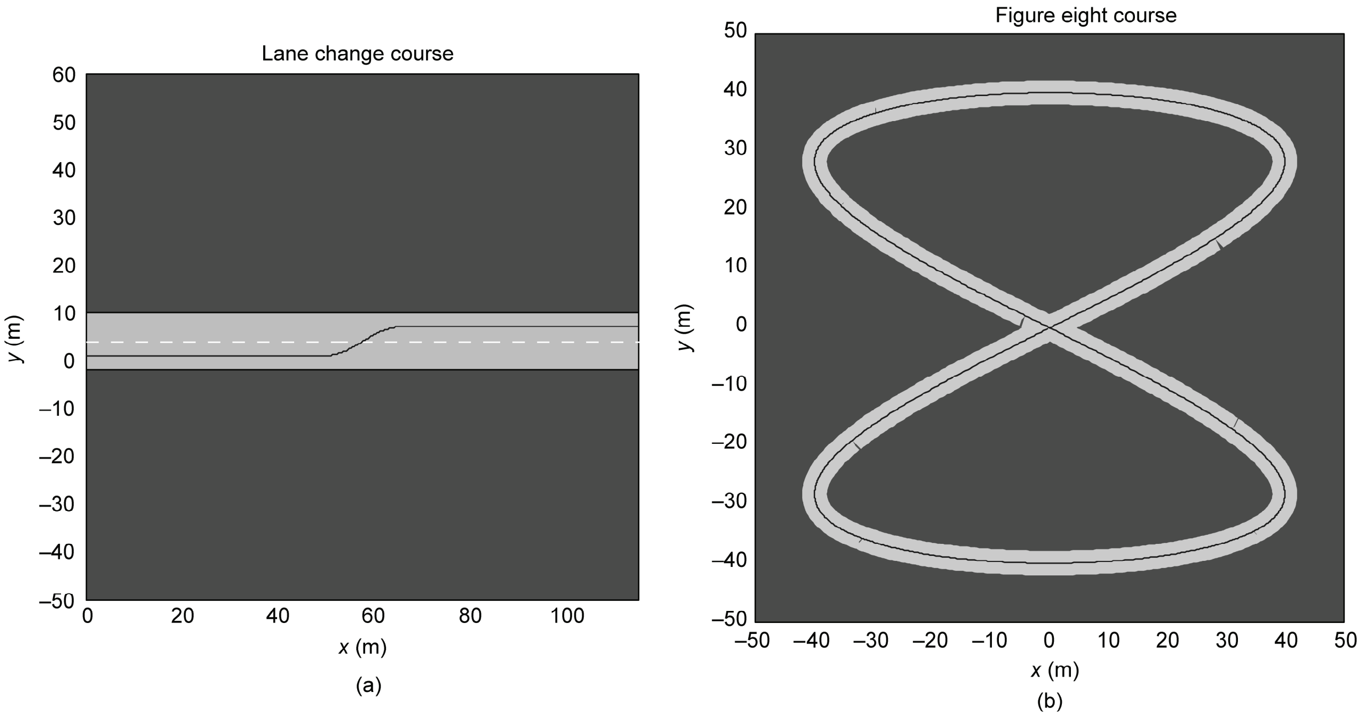

Figure 3. Driving courses: (a) Lane change course and (b) figure eight course.

method. The control law of pure pursuit is given as follows [8]: $\delta(k) = \arctan\left(\frac{2L\sin(\alpha)}{K_1 v(k)}\right)$, where L is the vehicle's wheelbase, α is the angle between the vehicle's orientation and the look-ahead vector, $v(k)$ is the vehicle speed and K_1 is the gain parameter. In PD control, the control law is $\delta(k) = K_p e(k) + K_d \dot{e}(k)$, where $e(k)$ is the lateral posture error defined in (11) and K_p and K_d are adjustable parameters. In the Stanley method [8], the steering control signal is generated *via* $\delta(k) = \theta_e(k) + \arctan\left(\frac{\mathcal{K} e_{fa}(k)}{v_x(k)}\right)$, where $\theta_e(k)$ and $e_{fa}(k)$ are the heading error and the distance between the centre of front wheels and its nearest path point, respectively. $\mathcal{K}$ is a parameter.

To obtain the best control performance of each method, the parameters K_1, K_p, K_d and $\mathcal{K}$ are empirically selected as $0.28, [2, 2, 2], [0.05, 0.05, 0.05]$ and 5 based on the simulation and real-world driving, respectively. In the simulation, the tracking performance of pure pursuit, PD control, Stanley and DHP at different vehicle speeds on the tested courses are obtained and compared. In the design of the DHP-based lateral controller, the control policy is obtained based on the learning for the two courses and during DHP learning, the vehicle speeds are randomly changed from 1 km/h to 70 km/h among different learning episodes, which can make the control policy obtained *via* DHP suitable to track different reference paths under various vehicle speeds.

In the lane change course, the number of the global path points is 565 and the initial path point is set to be $z_1 = [0; 0; 0]$. The initial vehicle coordinate is set to be $[0.5; 0.5; 0.1\pi]$. Figure 4 illustrates the tracking control performance of pure pursuit, PD control, Stanley and DHP at vehicle speeds of 10 km/h and 30 km/h. Compared with

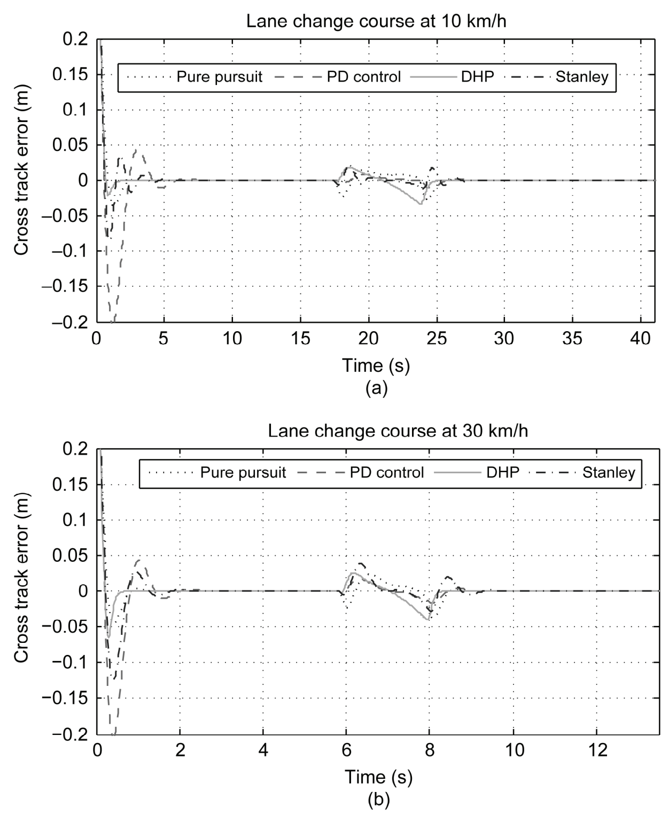

Figure 4. The tracking control performance in the lane change course at vehicle speeds of 10 km/h and 30 km/h.

the pure pursuit, PD control and Stanley methods, the DHP-based controller can make the vehicle to track the reference path faster when initial lateral error exists and has smaller overshoot. It is illustrated that at vehicle

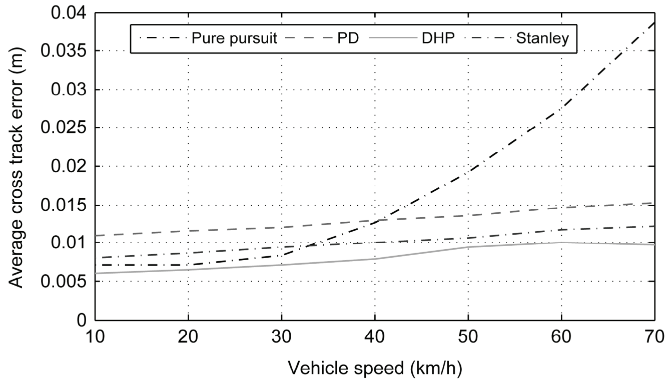

Figure 5. The average cross track error at different vehicle speeds in the lane change course.

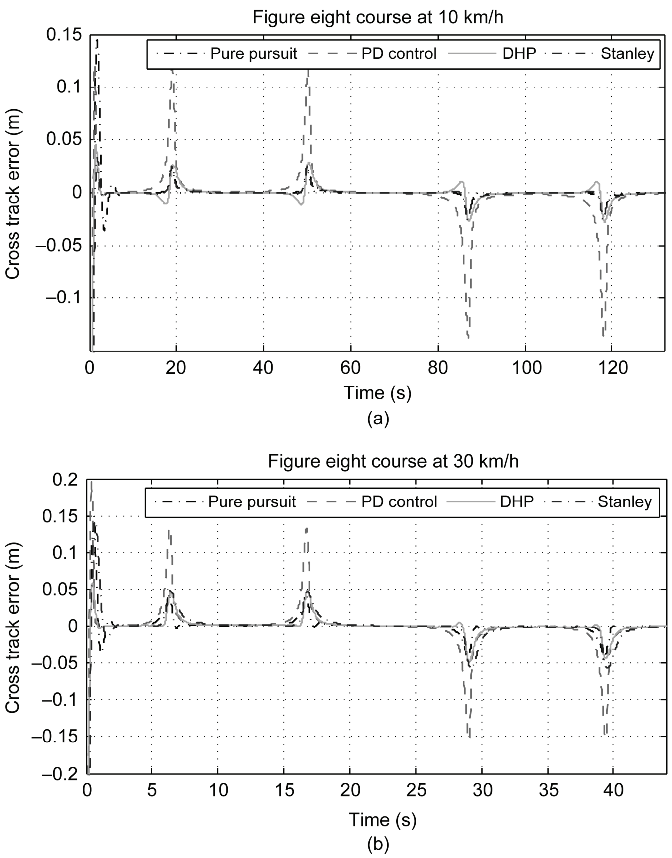

Figure 6. The tracking control performance in the figure eight course at vehicle speeds of 10 km/h and 30 km/h.

speeds of 10 km/h and 30 km/h, the DHP-based controller has better ability of changing lanes than the pure pursuit method. The abilities of tracking a straight path of the four methods are comparable under the two speeds.

Figure 5 shows the average cross track error at different vehicle speeds in the lane change course, where the average cross error is defined as follows:

$$E_{ace} = \frac{1}{N_e} \sum_{k=1}^{N_e} \sqrt{e_x^2(k) + e_y^2(k)} \tag{28}$$

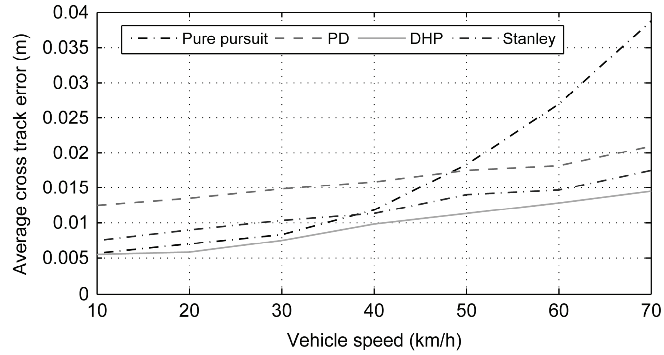

Figure 7. The average cross track error at different vehicle speeds in the figure eight course.

where N_e is the time step when the vehicle reaches the end of the reference path, $e_x(k)$ and $e_y(k)$ are defined in (11). As illustrated in Fig. 5, the DHP-based controller always has the least average error at different vehicle speeds.

In the figure eight course, the global path consists of 1,257 path points with initial path point $z_1 = [0; 0; 0.4636]$. The initial vehicle coordinate is set to be $[0.2; 1.0; 0.05\pi]$. Figure 6 shows the tracking control performance of the methods at vehicle speeds of 10 km/h and 30 km/h. DHP always has better tracking performance than PD control. DHP and pure pursuit have comparable tracking performance at vehicle speeds of 10 km/h and 30 km/h. When the vehicle speed increases, the tracking error of pure pursuit becomes larger and the Stanley method has a larger overshoot. The average cross track error of the figure eight course at different vehicle speeds is shown in Fig. 7. The average cross error is defined in (28). It is shown that the DHP-based lateral controller has smaller average cross error than the pure pursuit, PD control and Stanley methods. Meanwhile, the increasing rate of the average cross error of the DHP-based lateral controller is smaller than that of the pure pursuit method.

5. Conclusion

In this paper, a novel learning-based lateral control method is presented for ALVs. Based on the MDP model of the lateral control problem, the DHP algorithm is utilized to approximate the optimal control policy. The controller performance can be efficiently optimized in an online learning style. By using the lateral control error as well as the vehicle's longitudinal velocity as the learning controller's inputs, the proposed method can be adaptive to different longitudinal velocities. Simulation results show the proposed scheme outperforms existing popular methods, such as the pure pursuit, PD control and Stanley methods. Our ongoing work is to test the performance of the proposed method in real ALV platforms and improve the robustness of the learning controller.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 91220301 and the NSFC joint Foundation for Chinese Automobile Industry under Grant U1564214.

References

[1] X.Y. Wang, M.Y. Fu, H.B. Ma, and Y. Yang, Lateral control of autonomous vehicles based on fuzzy logic, *Control Engineering Practice*, *34*, 2015, 1–17.

[2] M.W. Park, S.W. Lee, and W.Y. Han, Development of lateral control system for autonomous vehicle based on adaptive pure pursuit algorithm, *International Conf. on Control, Automation and Systems*, KINTEX, Korea, 2014, 1443–1447.

[3] A. Zhu and S.X. Yang, Tracking control of a mobile robot with stability analysis, *International Journal of Robotics & Automation*, *28*(4), 2013, 340–348.

[4] S. Yu, T. Wang, Z. Wang, Y. Wang, C. Yao, Li, and Xiaofan, Original design of a wheelchair robot equipped with variable geometry single tracked mechanisms, *International Journal of Robotics & Automation*, *30*(1), 2015, 87–97.

[5] A. Zhu and S.X. Yang, An improved approach to dynamic task assignment of non-holonomic multi-robots, *International Journal of Robotics & Automation*, *26*(4), 2011, 362–368.

[6] O. Amidi, Integrated mobile robot control, Technical Report CMU-RI-TR-90-17, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1990.

[7] J.S. Wit, Vector pursuit path tracking for autonomous ground vehicles, Ph.D. Dissertation, University of Florida, Gainesville, FL, 2000.

[8] J.M. Snider, Automatic steering methods for autonomous automobile path tracking, Technical Report CMU-RI-TR-09-08, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 2009.

[9] J.H. Guo, P. Hu, L.H. Li, R.B. Wang, M.H. Zhang, and L. Guo, Study on lateral fuzzy control of unmanned vehicles via genetic algorithms, *Journal of Mechanical Engineering*, *48*(6), 2012, 76–82.

[10] M. Buehler, K. Iagnemma, and S. Singh, The DARPA urban challenge, *Springer Tracts in Advanced Robotics*, *56*(11), 2009, 6.

[11] M.A. Sotelo, Lateral control strategy for autonomous steering of Ackerman-like vehicles, *Robotics and Autonomous Systems*, *45*(3–4), 2003, 223–233.

[12] L. Menhour, D. Lechner, and A. Charara, Two degrees of freedom PID multi-controllers to design a mathematical driver model: Experimental validation and robustness tests, *Vehicle System Dynamics*, *49*, 2011, 595–624.

[13] J. Pérez, V. Milaneïs, and E. Onieva, Cascade architecture for lateral control in autonomous vehicles, *IEEE Transactions on Intelligence Transportation System*, *12*(1), 2011, 73–82.

[14] H. Šiljak, Inverse matching-based mobile robot following algorithm using fuzzy logic, *International Journal of Robotics & Automation*, *29*(4), 2014, 369–377.

[15] P.R. Ouyang, W.H. Yue, and V. Pano, Hybrid PD sliding mode control for robotic manipulators, *International Journal of Robotics & Automation*, *29*(4), 2014, 387-395.

[16] F. Piltan, N. Sulaiman, A. Gavahian, S. Soltani, and S. Roosta, Design mathematical tunable gain PID-like sliding mode fuzzy controller with minimum rule base, *International Journal of Robotics & Automation*, *2*(2), 2011, 146–156.

[17] 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 Transactions on Intelligence Transportation System*, *10*(1), 2009, 92–102.

[18] P. Falcone, F. Borrelli, J. Asgari, H.E. Tseng, and D. Hrovat, Predictive active steering control for autonomous vehicle systems, *IEEE Transactions on Control Systems Technology*, *15*(3), 2007, 566–580.

[19] L. Menhour, A. Charara, and D. Lechner, Switched LQR/H_∞ steering vehicle control to detect critical driving situations, *Control Engineering Practice*, *24*, 2014, 1–14.

[20] F. Piltan, N. Sulaiman, Z. Tajpaykar, P. Ferdosali, and M. Rashidi, Design artificial nonlinear robust controller based on CTLC and FSMC with tunable gain, *International Journal of Robotics & Automation* *2*(3), 2011, 205–220.

[21] I. Markelić, T. Kulvičius, M. Tamosiunaite, and F. Wörgötter, Anticipatory driving for a robot-car based on supervised learning, *Anticipatory Behavior in Adaptive Learning Systems*, 2009, 267–282.

[22] D. Stavens, G. Hoffmann, and S. Thrun, Online speed adaptation using supervised learning for high-speed, off-road autonomous driving, *Proc. International Joint Conf. on Artificial Intelligence*, Hyderabad, India, 2007, 2218–2224.

[23] R.S. Sutton and A.G. Barto, *Reinforcement learning: An introduction* (Cambridge, MA: MIT Press, 1998).

[24] X. Xu, Z.S. Hou, C.Q. Lian, and H.B. He, Online learning control using adaptive critic designs with sparse kernel machines, *IEEE Transactions on Neural Networks and Learning System*, *24*(5), 2013, 762–775.

[25] X. Xu, D.W. Hu, and X.C. Lu,Kernel-based least-squares policy iteration for reinforcement learning, *IEEE Transactions on Neural Networks*, *18*(4), 2007, 973–992.

[26] X. Xu, C.M. Liu, S.X. Yang, and D.W. Hu, Hierarchical approximate policy iteration with binary-tree state space decomposition, *IEEE Transactions on Neural Networks*, *22*(12), 2011, 1863–1877.

[27] P.J. Werbos, Approximate dynamic programming for real-time control and neural modeling, in D.A. White and D.A. Sofge (eds.), *Handbook of intelligent control: Neural, fuzzy, and adaptive approaches* (New York: Van Nostrand, 1992), Chapter 13.

[28] F.L. Lewis and D.R. Liu, *Reinforcement learning and approximate dynamic programming for feedback control* (New York: Wiley, 2012).

[29] X. Xu, C.Q. Lian, L. Zuo, and H.B. He, Kernel-based approximate dynamic programming for real-time online learning control: An experimental study, *IEEE Transactions on Control Systems Technology*, *22*(1), 2014, 146–156.

[30] S.-Y. Oh, J.-H. Lee, and D.-H. Choi, A new reinforcement learning vehicle control architecture for vision-based road following, *IEEE Transactions on Vehicular Technology*, *49*(3), 2000, 997–1005.

[31] M.A. Riedmiller, M. Montemerlo and H. Dahlkamp, Learning to Drive a Real Car in 20 Minutes, *Frontiers in the Convergence of Bioscience and Information Technologies 2007, FBIT 2007*, Jeju Island, Korea, October 11–13, 2007, 645–650.

[32] R. Hafner and M. Riedmiller, Neural reinforcement learning controllers for a real robot application, *IEEE International Conf. on Robotics and Automation*, Roma, Italy, 2007.

[33] S.X. Yang, A. Zhu, G. Yuan, and M.Q.-H. Meng, A bioinspired neurodynamics-based approach to tracking control of mobile robots, *IEEE Transactions on Industrial Electronics*, *59*(8), 2012, 3211–3220.

[34] G.K. Venayagamoorthy, R.G. Harley, and D.C. Wunsch, Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbogenerator, *IEEE Transactions on Neural Networks*, *13*(3), 2002, 764–773.

[35] H.G. Zhang, Y.H. Luo, and D.R. Liu, 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), 2009, 1490–1503.

Biographies

Zhenhua Huang received the Bachelor's degree in vehicle engineering from Changsha University of Science and Technology in 2008 and the Master's degree from the College of Mechatronics and Automation (CMA), National University of Defense Technology (NUDT), PR China in 2010. He is currently a Ph.D. student at the Institute of Unmanned Systems, CMA, NUDT, PR China. His current research interests include machine learning, robot control and autonomous land vehicles. He has co-authored 11 papers in international journals and conferences.

Chuanqiang Lian received the Bachelor's degree from the Department of Automation, Tsinghua University, Beijing, China in 2008 and the Master's degree from the College of Mechatronics and Automation, National University of Defense Technology, Changsha, China in 2010. He is currently pursuing the Ph.D. degree with the Institute of Unmanned Systems. He has authored or co-authored more than 10 papers in international journals and conferences. His current research interests include reinforcement learning, approximate dynamic programming and autonomous vehicles.

Xin Xu received the BS degree in electrical engineering from the Department of Automatic Control, National University of Defense Technology (NUDT), Changsha, China in 1996 and the Ph.D. degree in control science and engineering from the College of Mechatronics and Automation, NUDT in 2002. He has been a visiting professor in Hong Kong Polytechnic University, University of Alberta, University of Guelph, and the University of Strathclyde, UK. He is currently a professor with the College of Mechatronics and Automation, NUDT, China. He has co-authored more than 140 papers in international journals and conferences and co-authored in four books. His research interests include intelligent control, reinforcement learning, approximate dynamic programming, machine learning, robotics and autonomous vehicles. He received the Fork Ying Tong Youth Teacher Fund of China in 2008 and the second class National Natural Science Award of China in 2012. He serves as the co-Editor-in-Chief of *Journal of Intelligent Learning Systems and Applications* and the Associate Editor-in-Chief of *CAAI Transactions on Intelligent Technology* (Elsevier). He is an Associate Editor of *Information Sciences*, *Intelligent Automation & Soft Computing* and *Acta Automatica Sinica*. He was a Guest Editor of the *International Journal of Adaptive Control and Signal Processing* and *Mathematical Problems in Engineering*. He is a Senior Member of IEEE and a member of the IEEE CIS Technical Committee on Approximate Dynamic Programming and Reinforcement Learning (ADPRL) and the IEEE RAS Technical Committee on Robot Learning.

Jian Wang received the Ph.D. degree in automatic control from the College of Mechatronics and Automation, National University of Defense Technology, Changsha, China in 2013. He has authored or co-authored more than 10 papers in international journals and conferences. His research interests include autonomous land vehicles, intelligent systems and machine learning.