Iranian Journal of Science and Technology, Transactions of Electrical Engineering (2020) 44:1059–1068
https://doi.org/10.1007/s40998-019-00286-4

RESEARCH PAPER

# Trajectory Tracking Control for Mobile Robots Using Reinforcement Learning and PID

Shuti Wang[1] · Xunhe Yin[1] · Peng Li[1] · Mingzhi Zhang[2] · Xin Wang[1]

Received: 1 March 2019 / Accepted: 27 October 2019 / Published online: 8 November 2019

**Abstract**

In this paper, a novel algorithm of trajectory tracking control for mobile robots using the reinforcement learning and PID is proposed. The Q-learning and PID are adopted for tracking the desired trajectory of the mobile robot. The proposed method can reduce the computational complexity of reward function for Q-learning and improve the tracking accuracy of mobile robot. The effectiveness of the proposed algorithm is demonstrated via simulation tests.

**Keywords** Trajectory tracking control · Reinforcement learning · Q-learning · PID

## 1 Introduction

Mobile robots have been widely applied in many fields like signpost detection, service occupations, outer space exploring, etc. (Leena and Saju 2016; Klancar and Skrjanc 2007; Simba et al. 2016). Trajectory tracking is an effective way to control mobile robots. Although many existing studies are related to path tracking control for mobile robots, it is difficult for mobile robots to ensure the desired position in space or track the reference trajectory precisely (Klancar and Skrjanc 2007; Suruz Miah and Gueaieb 2014; Xiao et al. 2017). To address this issue, a novel approach is proposed for trajectory tracking control of mobile robots using the reinforcement learning (RL) and PID.

Reinforcement learning is a novel machine learning method based on animal learning psychology, which can get the optimal policy of an agent using the responses to environment (Jiang et al. 2018a; Fernandez-Gauna et al. 2018). Thus, RL can deal with complicated sequential decision-making issues in many control applications (Jiang et al. 2018b). In Doya (2000), a RL control used in continuous time and space is presented. In Li et al. (2015), a RL control for multi-robots coordination is presented using Lyapunov stability analysis. In Li et al. (2018a), a tracking algorithm based on neural network and RL is proposed for wheeled mobile robots, in case that the system is nonlinear and discrete. In Günther et al. (2016), a RL is proposed for laser beam welding of laser welding control. In Shah et al. (2016), a predictive control algorithm using RL is investigated. To facilitate the robotic operators, a RL-based control method is developed using the neural network (Miljkovic et al. 2013), whose outputs are set as the value function for RL.

It is known that reinforcement learning usually uses action mappings and value function via function approximator for continuous states, but it is complicated to design the function approximator. To solve this problem, a new RL control method is proposed to directly achieve the action on the basis of value function (Kubalik et al. 2017). To control the dosage for curing cancer, RL is investigated and applied as well (Padmanabhan et al. 2017). In Görges (2017), two neural networks are used to approximate the value function of RL, and the output of one neural network is designed as the critic structure of RL, while the other neural network's output is considered as the actor structure. Considering the facts that traditional robust control can hardly perform well when the control system is uncertain and nonlinear, a novel robust control algorithm based on reinforcement learning has been presented in Jiang et al. (2018a).

✉ Xunhe Yin
xhyin@bjtu.edu.cn

[1] School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China

[2] School of Electrical Engineering, Beijing Jiaotong University, Beijing 100044, China

The above results show that the effect of reinforcement learning is good. However, the design of the reward function of RL is obviously complicated, and this inevitably increases the computational complexity and the system cost. In this paper, to address these problems, the form of the reward function of reinforcement learning is simplified while simultaneously ensuring the control performance of the system. The tracking control results of the improved reinforcement learning are analyzed and compared with traditional methods to show its superiorities.

In many other applications like Shi et al. (2018), where the robot soccer system is a complicated time series decision system, there are more serious uncertain problems. A new control way based on reinforcement learning and fuzzy theory has been proposed for robot soccer system, but the prior knowledge should be provided in advance, which reduces the ability of reinforcement learning to find the optimal strategy. To seek the optimal power point of the photovoltaic source, RL control is applied in Kofinas et al. (2017), to obtain the best action by trial and error, but simulation results of control are not satisfactory. In the medical field, the RL has been used to design narcosis controller like (Padmanabhan et al. 2015). To cut down traffic congestion, the RL controller is investigated with neural networks in Genders and Razavi (2018) with reduced system cost. A novel control method is designed based on fuzzy reinforcement learning and Lyapunov theory in Kumar and Sharma (2018). In Lopez-Guede et al. (2018), a control method based on RL is presented to control the robot.

Note that the Q-learning method is a vital algorithm of RL, which can get the optimal action by trial and error (Anderlini et al. 2018), but it takes a long time for the value function of Q-learning to converge. For example, to reduce the green house effect, RL is studied in Beghi et al. (2017) to control the refrigeration system, but the value function of RL is still under oscillation. In Yang et al. (2015), a new reinforcement learning control is used to control the PV array to reduce the energy consumption, but the value function of RL is not convergent. Besides, reinforcement learning has been studied for military applications like (Mendonça et al. 2018). An improved RL is presented to control the multitasks system in Zhan et al. (2017). Intelligent controller is studied based on RL, to control the liquid level of tank systems in Ramanathan et al. (2018).

These studies show that the control performance of RL still needs improvement. Thus, in this paper, a novel trajectory tracking control method named Q-learning–PID is presented by blending the Q-learning with PID. We take the advantages of RL and overcome the disadvantages for RL. It has been widely regarded that compared with robust smooth approaches, neural networks, feedback control and optimization methods (Li et al. 2017a, 2018b; Görges 2017; Liu and Song 2011), PID algorithm is simple but effective, which can reduce the complexity of the control system design and facilitate the real engineering applications. Motivated by these observations, in this paper the value function of the RL is designed for the control system, and the RL controller of the system is developed using the value function. Then, the other controller for the system is designed on the basis of PID control. The proposed RL–PID controller is applied for controlling the mobile robot.

The rest of this paper is organized as follows. In Sect. 2, the kinematics of mobile robot are introduced. In Sect. 3, the reinforcement learning (RL) is described and the design for Q-learning controller and PID controller is provided as well. The presented Q-learning–PID control algorithm is validated via simulations in Sect. 4. Finally, the conclusions of this paper are given in Sect. 5.

## 2 Kinematics Model for the Mobile Robot

In this paper, the two-wheel mobile robot is studied, which has been applied in many fields and is shown in Fig. 1 (Simba et al. 2016; Huang et al. 2014; Li et al. 2018a). Here, $G$ represents the actual mobile robot location, while $G_\mathrm{r}$ denotes the reference mobile robot location. $v$ and $v_\mathrm{r}$ denote the linear velocities for the actual mobile robot and reference mobile robot, respectively. Similarly, $\omega$ and $\omega_\mathrm{r}$ denote the angular velocities, while $\phi$ and $\phi_\mathrm{r}$ are the corresponding angles. Besides, we define $X = [x, y, \phi]^\mathrm{T}$ and $X_\mathrm{r} = [x_\mathrm{r}, y_\mathrm{r}, \phi_\mathrm{r}]^\mathrm{T}$ as the orientations.

According to Fig. 1, the kinematics model of the mobile robot (Simba et al. 2016; Huang et al. 2014; Li et al. 2018a) can be described as

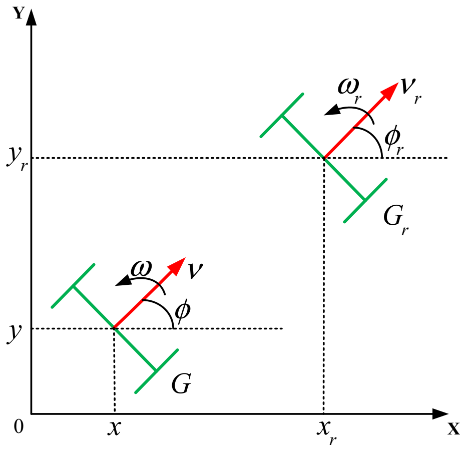

**Fig. 1** Model of mobile robots

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{1}$$

where $u = [v, \omega]^{\mathrm{T}}$ is the control input and the path tracking error model (Simba et al. 2016; Huang et al. 2014, 2016; Li et al. 2017b, 2018a) between reference mobile robot and actual mobile robot can be found by

$$\begin{bmatrix} x_{\mathrm{e}} \\ y_{\mathrm{e}} \\ \phi_{\mathrm{e}} \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\mathrm{r}} - x \\ y_{\mathrm{r}} - y \\ \phi_{\mathrm{r}} - \phi \end{bmatrix} \tag{2}$$

In formula (2), $x_{\mathrm{e}}$ is the transverse position error, $y_{\mathrm{e}}$ is the longitudinal position error, and $\phi_{\mathrm{e}}$ denotes the angle error. Then, (2) can be differentiated as

$$\begin{bmatrix} \dot{x}_{\mathrm{e}} \\ \dot{y}_{\mathrm{e}} \\ \dot{\phi}_{\mathrm{e}} \end{bmatrix} = \begin{bmatrix} \omega y_{\mathrm{e}} - v + v_{\mathrm{r}} \cos\phi_{\mathrm{e}} \\ -\omega x_{\mathrm{e}} + v_{\mathrm{r}} \sin\phi_{\mathrm{e}} \\ \omega_{\mathrm{r}} - \omega \end{bmatrix} \tag{3}$$

## 3 Trajectory Tracking Control for the Mobile Robot

The schematic diagram of the mobile robot control can be described in Fig. 2. Here, the error signal $e(t)$ between the reference $X_{\mathrm{r}}$ and the actual $X$ is defined as $X_{\mathrm{r}} - X$. The control input $u1$ is designed using the reinforcement learning, $u2$ is the output of PID controller, so the control input $u$ for mobile robot is the sum of reinforcement learning and PID control input, as

$$u = u1 + u2 \tag{4}$$

To facilitate the novel controller design, a brief review of RL and Q-learning as well as PID control theory is given as follows. Then, the design method for the trajectory tracking controller based on Q-learning and PID will be proposed (Carlucho et al. 2017; Klancar and Skrjanc 2007).

### 3.1 Description of Reinforcement Learning

Reinforcement learning is an important part of machine learning and learning control systems. The structure block for RL (Padmanabhan et al. 2015) can be described by Fig. 3.

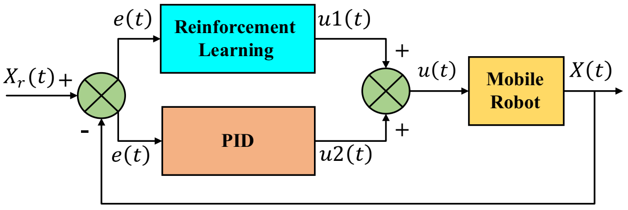

Fig. 2 Mobile robot control schematic diagram

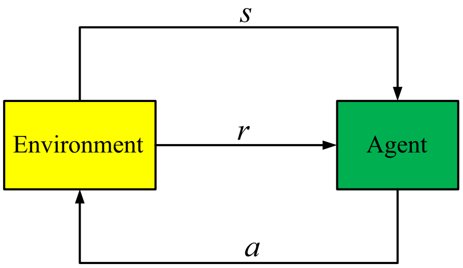

Fig. 3 Structure block of reinforcement learning

In Fig. 3, $S$ is the state of environment, $r$ is the reward value, and $a$ denotes the policy. From Fig. 3, we can see that the learning principle of RL comes from the interaction between the agent and environment (Hernández-del-Olmo et al. 2018). The agent yields the environment policy $a$ using the state $S$ and then immediately gets the reward value $r$ from environment.

The RL works based on finite Markov decision process (MDP). Usually, a finite MDP is made up of $(S, A, P, R)$, where $S$ denotes the state space of environment, $A$ represents the action space, $P$ is the state transition probability and $R$ is the immediate reward function. At each step $t$, according to the state $s_t$ of environment, the policy $a_t$ given by the agent will act on the environment, while the agent will get an immediate reward value $r_t$ from the environment. At step $t+1$, the state $s_t$ will be transited to the state $s_{t+1}$ with the transition probability $P(s_{t+1}|s_t, a_t)$. That is, the goal of agent is to get an optimal strategy $\pi^*(s) : S \to A$, which maximizes the anticipant reward value through state $s$, so the value function (Padmanabhan et al. 2015) for state $s$ can be shown as

$$V^{\pi}(s) = E^{\pi}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | s_t = s\right] \tag{5}$$

In formula (5), $\gamma \in (0, 1]$ is the discount rate, and the strategy $\pi$ can be estimated on the basis of the value function (Fernandez-Gauna et al. 2018). Thus, the goal of RL is to get the optimal policy to maximize the value function by interacting with environment.

### 3.2 The Q-Learning Controller Design

Q-learning is a kind of reinforcement learning and model-free mechanism, whose value function is based on the state action, and the iteration formula (Anderlini et al. 2016) for Q-learning can be described as

$$Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \alpha[r_{t+1} + \gamma \max Q_t(s_{t+1}, a) - Q_t(s_t, a_t)] \tag{6}$$

where $\alpha \in (0, 1]$ is the learning rate, i.e., the learning speed of Q-learning. $\gamma \in (0, 1]$ is the discount parameter, describing the weighting importance between the immediate reward and future long-term reward (Anderlini et al. 2016). $s$ is the state for system or environment. $a$ is the policy given by the agent. At each step $t$, the state-action value function is $Q_t(s_t, a_t)$, while the state $s$ will transfer from $s_t$ to $s_{t+1}$. Then, max $Q_t(s_{t+1}, a_t)$ is the maximal value of all predicted values at the step time $t+1$ and can be decided by the policy $a$ based on $\varepsilon$ named as the greedy policy. $r_{t+1}$ is the reward at each step $t+1$, and the state-action value function for Q-learning is updated according to formula (6).

In this paper, the Q-learning controller performs as part of the path tracking control system for mobile robot, and will be designed based on the Q-learning value function, which is another representation of the $Q$ matrix and can clearly show the change of $Q$ matrix with the iteration of learning repetitions. The Q-learning value function for mobile robot can be shown as

$$Q_{t+1}(X_t, u1_t) \leftarrow Q_t(X_t, u1_t) + \alpha[r_{t+1} + \gamma \min Q_t(X_{t+1}, u1) - Q_t(X_t, u1_t)] \tag{7}$$

In formula (7), $X$ is the state of mobile robot, $X$ contains the lateral position $x$, longitudinal direction position $y$ and angle $\phi$. The Q-learning controller $u1$ of mobile robot can be described as

$$u1 = \arg * \min Q(X_t, u1_t) \tag{8}$$

The reward function $r_t$ is designed based on the tracking errors as

$$r_t = \frac{1}{2}\left((X_r)_t - X_t\right)^{\mathrm{T}} \mathrm{M}\left((X_r)_t - X_t\right) \tag{9}$$

where the $X_{\mathrm{r}}$ is the state of the reference mobile robot and $M$ is the positive semidefinite matrix (Wang et al. 2015). The design of reward function $r_t$ is then a straightforward process. From formula (7) to (9), we can see that the Q-learning controller of mobile robot will rely on its own value function, so the goal of value function is to reduce the tracking errors of mobile robot.

The tracking control algorithm for mobile robot based on Q-learning can be depicted as follows:

Step 1: initialize $Q(X, u1)$
Step 2: take an action $u1_t$ according to the state $X_t$ and calculate the reward function $r_t$
Step 3: observe the next state $X_{t+1}$, and get $\min Q_t(X_{t+1}, u1)$
Step 4: update $Q(X, u1)$,$X_t \leftarrow X_{t+1}$
Step 5: go back to step 2 until the value of $Q(X, u1)$ is convergent.

### 3.3 The PID Controller Design

PID method has been widely used in many fields because of its briefness and high availability. In this paper, Fig. 4 shows the block diagram of PID control, and the formula (Carlucho et al. 2017; Mahmoodabadi et al. 2017) of PID controller is shown in Eq. (10).

$$u2(t) = K_{\mathrm{p}} e(t) + K_{\mathrm{i}} \int_0^t e(\tau)\mathrm{d}\tau + K_{\mathrm{d}} \frac{\mathrm{d}e(t)}{\mathrm{d}t} \tag{10}$$

In Fig. 4 and formula (10), the error signal $e(t)$ is defined as $X_{\mathrm{r}} - X$, $K_{\mathrm{p}}$ is proportional parameter, $K_{\mathrm{i}}$ delegates integral gain, $K_{\mathrm{d}}$ is named as derivative constant, and $u2(t)$ is control signal. Actions of the proportional, integral and derivative components are derived from the error signal $e(t)$, and the control input $u2(t)$ is then generated using these actions.

## 4 Simulation

Note that the control inputs and outputs of mobile robot are discrete in this work. The control inputs consist of two variables, that is, the linear speed $v$ and angular velocity $\omega$. The outputs correspond to three variables, i.e., the transverse position $x$, longitudinal position $y$ and angle $\phi$. To verify the effectiveness of the proposed algorithms, the concrete simulation example is given by using the MATLAB software.

### 4.1 Simulation Based on Q-Learning Controller

First, the single Q-learning controller is used to track the desired path for mobile robot.

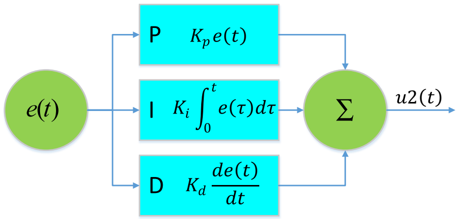

**Fig. 4** Block diagram of PID control

In formula (9), $M = [122; 012; 001]$, the initial states of mobile robot are $[2.5 \quad -2 \quad 2]^{\mathrm{T}}$. Thus, the reference trajectory can be given as

$$\begin{cases} X_r = \cos(\pi t) \\ Y_r = \sin(\pi t) \end{cases} \tag{11}$$

To observe the effect of learning rate $\alpha$ on Q-learning, the discount factor $\gamma$ is set the constant 0.9, while $\alpha$ varies as 0.08, 0.4 and 0.9, respectively. The simulation results can be shown in Figs. 5, 6, 7 and 8. In Fig. 5, the value function for Q-learning changes along with the iteration steps. At the beginning, the value function is not steady, which indicates that the Q-learning cannot find the optimal policy. The value function converges to 3.1699 after about 360 learning steps when the learning rate $\alpha$ is 0.9. When $\alpha$ is set as 0.4, the value function starts to converge to 3.1699 after about 485 learning steps. After about 2000 learning steps, the value function converges to 3.1699 when $\alpha$ is 0.08. So we can conclude that when the discount factor $\gamma$ is fixed, the convergence rate of value function will become fast with increasing learning rate $\alpha$, but the learning rate $\alpha$ basically does not affect the size of the value function.

Figures 6, 7 and 8 show the path tracking errors of mobile robot based on the Q-learning controller. Here, the trajectory tracking errors are not steady in the initial learning stages of Q-learning. Figure 6 shows the transverse position error, the longitudinal position error can be seen in Fig. 7, and the angle error is shown in Fig. 8. It can be seen that the learning rate $\alpha$ affects the convergence time of the trajectory tracking errors when the discount factor $\gamma$ is fixed. The convergence time of tracking errors will decrease with the increasing learning rate $\alpha$. In Fig. 6, when $\alpha$ is 0.08, the transverse position error converges to 0.2983 after about 2000 iteration steps, and the transverse position error inclines to 0.2982 after about 500 iteration

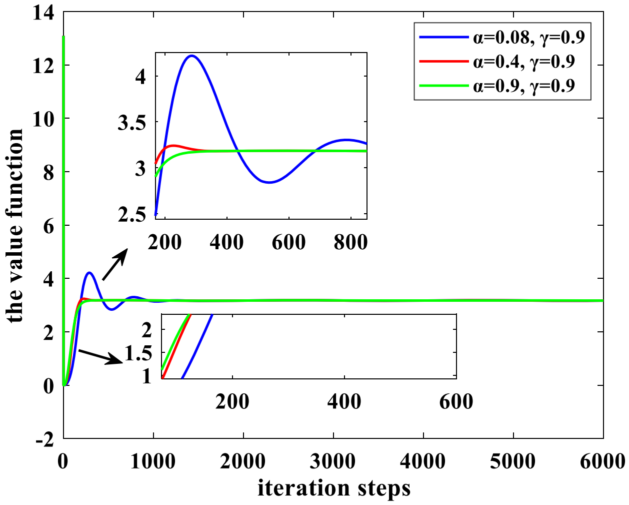

**Fig. 5** Value function

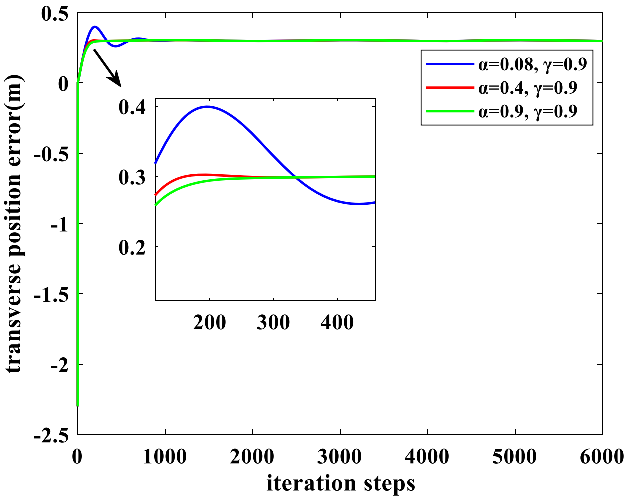

**Fig. 6** Transverse position error of mobile robot with Q-learning

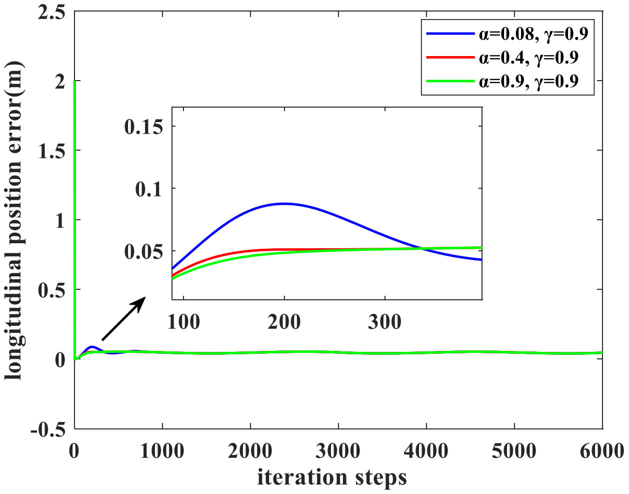

**Fig. 7** Longitudinal position error of mobile robot with Q-learning

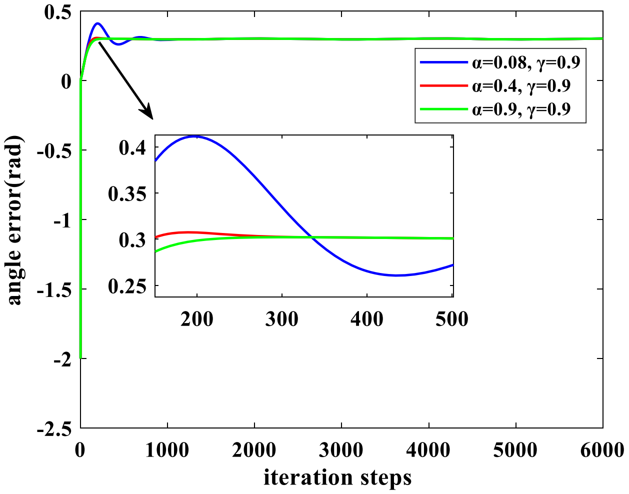

**Fig. 8** Angle error of mobile robot with Q-learning

steps when $\alpha$ is 0.4. It takes near 360 iteration steps for transverse position error to converge to 0.2982 when $\alpha$ is 0.9.

In Fig. 7, the longitudinal position error starts to converge to 0.0459 at about 1500 iteration steps when $\alpha$ is 0.08, and it takes about 400 iteration steps for longitudinal position error to converge to 0.0459 when $\alpha$ is 0.4, but the longitudinal position error needs about 260 iteration steps to incline to 0.0458 when $\alpha$ is 0.9.

In Fig. 8, when the learning rate $\alpha$ is, respectively, set for 0.08, 0.4 and 0.9, the corresponding angle error needs about 1800 iteration steps, 440 iteration steps and 300 iteration steps, respectively, to reach the stable value, and their values are basically the same, with the angle errors nearly 0.3029. The results for Figs. 5, 6, 7 and 8 are discrete, but look continuous when the discrete points are concentrated.

To test the effect of the discount factor $\gamma$ on the Q-learning controller, $\alpha$ is set for 0.9, while $\gamma$ is set for 0.09, 0.39 and 0.6, respectively. At the same time, the learning rate $\alpha$ and discount factor $\gamma$ are set for 1 value to make the Q-learning get the optimal learning parameters. The simulation results are shown in Figs. 9, 10, 11 and 12.

In Fig. 9, when $\alpha$ is identical, although the discount factor $\gamma$ increases, the value function is basically unchanged. If $\gamma$ gets bigger, the value function needs less iteration steps to get the steady value. But when $\alpha$ and $\gamma$ are both constant 1, it takes more iteration steps for the value function to converge, although their values for the value function are convergent.

In Figs. 10, 11 and 12, we can see that when $\alpha$ is the same, with increasing $\gamma$, the transverse position error, longitudinal position error and angle error decrease. When $\alpha$ and $\gamma$ are both set for 1, the tracking errors of mobile robot are minimum, the transverse position error converges

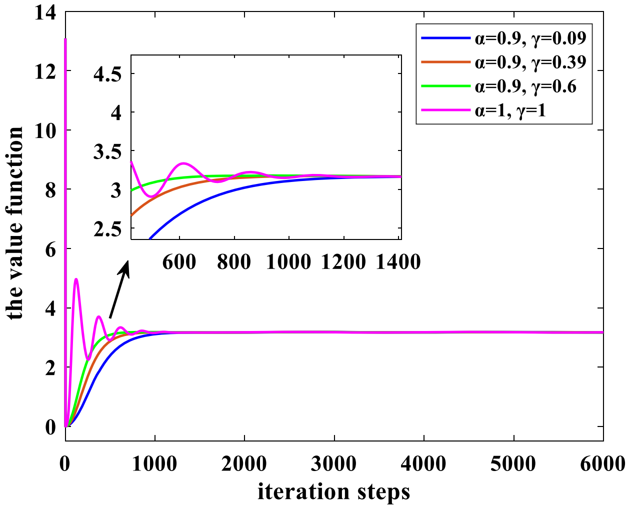

**Fig. 9** Value function

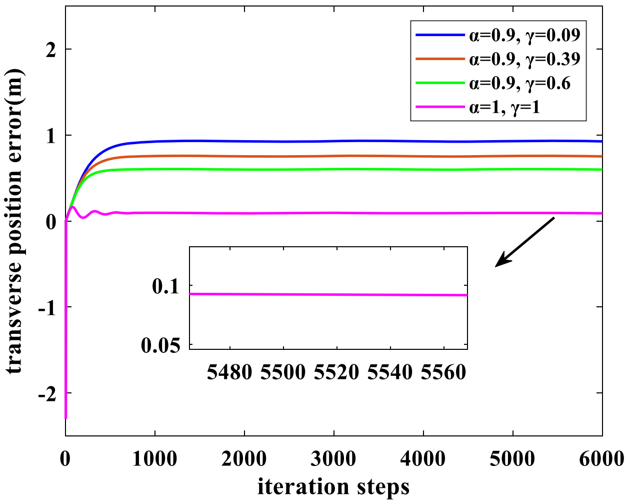

**Fig. 10** Transverse position error of mobile robot with Q-learning

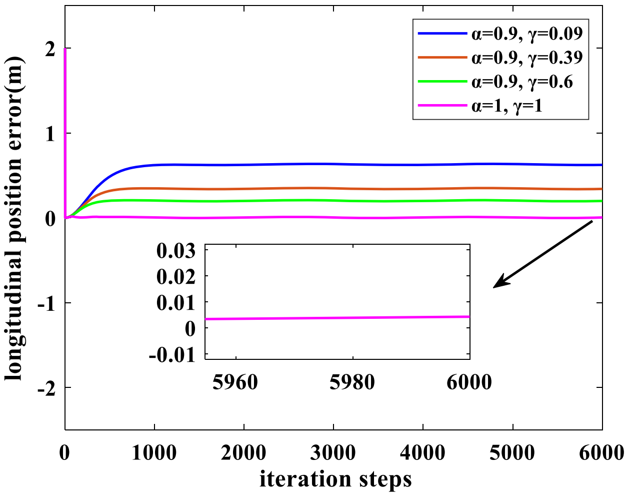

**Fig. 11** Longitudinal position error of mobile robot with Q-learning

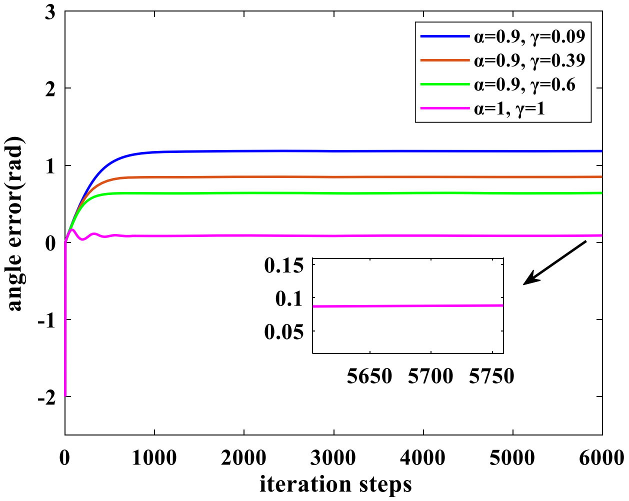

**Fig. 12** Angle error of mobile robot with Q-learning

to 0.0892, the convergent value of longitudinal position error is about 0.0041, and the angle error tends to stable value 0.0894.

Based on the above simulation results, it is concluded that the value of Q-learning can converge to the steady value when the learning rate $\alpha$ and discount factor $\gamma$ are set as appropriate values. To make the Q-learning converge, the learning rate $\alpha$ and discount factor $\gamma$ will be constantly adjusted. Figures 6, 7, 8, 10, 11 and 12 show that the path tracking errors based on Q-learning are larger, the control effect of using the single Q-learning is poor as well, so an improved method named Q-learning–PID is applied that blends the Q-learning with PID for tracking control of mobile robot, as shown in the coming section.

## 4.2 Simulation Based on Q-Learning–PID

To address the shortcomings of Q-learning, the hybrid Q-learning–PID is designed for the path tracking control of mobile robot. The reference path is formula (11), while the learning rate $\alpha$ and the discount parameter $\gamma$ are both set as 1. The adjustment process for parameters of PID is the same as that with or without Q-learning. That is, the proportional parameters, the integral gains and the derivative constants of PID are obtained by an empirical trial-and-error method. To verify the effect of the proposed method, the simulation diagrams are exhibited in Figs. 13, 14, 15 and 16 for comparisons of PID, Q-learning and Q-learning–PID.

Then, the mean square error (MSE) is adopted to evaluate the control effect of the proposed control approach. It is defined as

$$\mathrm{MSE} = \frac{\sum_{i=1}^{K}(e_i)^2}{K} \qquad (12)$$

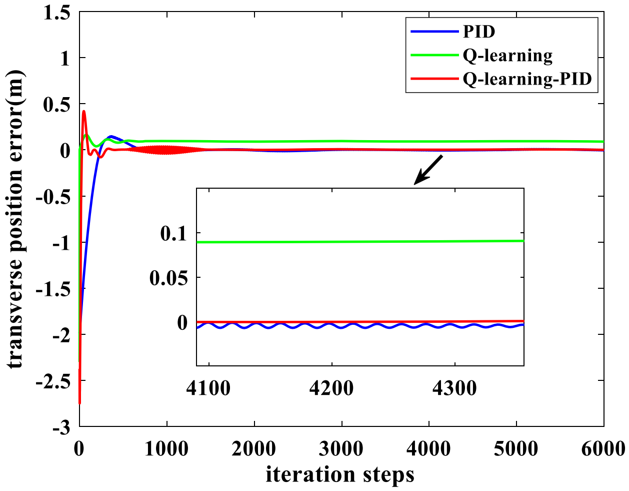

Fig. 13 Transverse position error of mobile robot

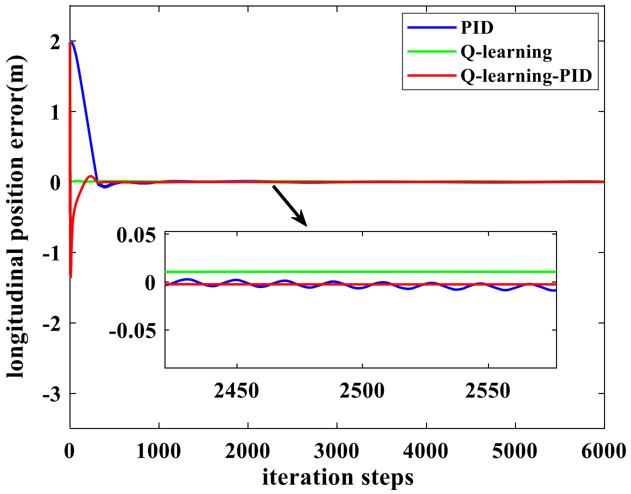

Fig. 14 Longitudinal position error of mobile robot

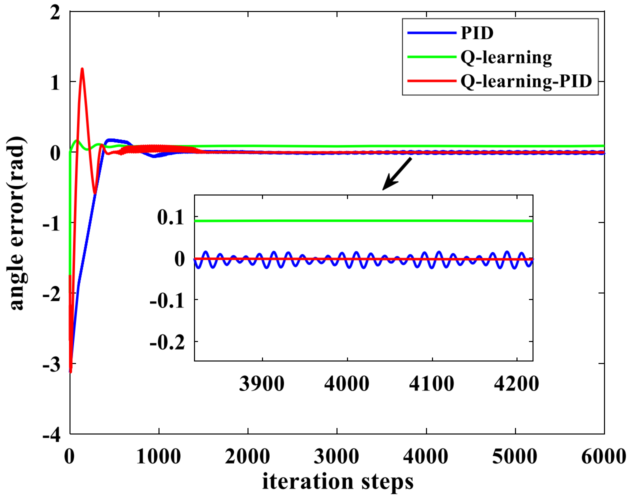

Fig. 15 Angle error of mobile robot

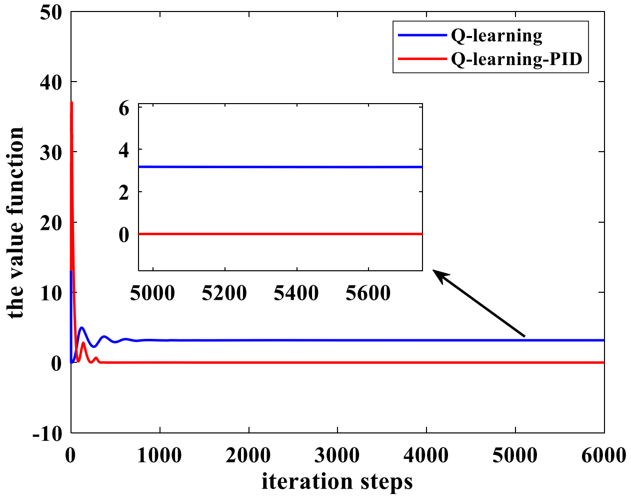

Fig. 16 Value function

where $i \in \{1, 2, \ldots, 6000\}$ denotes the ith error; $e_i$ is the tracking error of mobile robot; the MSEs of transverse position, longitudinal position and angle are listed in Table 1.

Compared with PID and Q-learning, the Q-learning–PID still has overshoots and chatters at the beginning of training but has the smallest MSE, as shown in Table 1. There are peaks in simulation results, but the system response speed is faster and its convergent time is shorter. To study the reinforcement learning more conveniently, we use a kinematics model for our mobile robot in this work to validate the reinforcement learning for tracking control. From Figs. 13, 14, 15 and Table 1, we see that the transverse position error, the longitudinal position error and the angle error for mobile robot path tracking based on Q-learning–PID are convergent and near to zero. The value function of Q-learning based on the Q-learning–PID algorithm is shown in Fig. 16, indicating that the value function based on Q-learning–PID converges to zero, but the value function based on Q-learning converges to 3.17. That is, results in this work show that the value function of Q-learning converges to the steady state after iterations. From formula (7)–(9), it is found that the value function is designed to ensure the minimum error, so the value function tending to be zero indicates the tracking error tends to zero as well.

Consequently, we can conclude that the control performance of Q-learning–PID is superior to the PID and Q-learning. Q-learning has the ability of self-learning and can find the optimal strategy by trial and error with environment, but the control effect of a single Q-learning is not sufficient. PID is simple and robust, but PID lacks the self-learning ability. Thus, adding Q-learning to the PID helps the original PID improve the learning ability and thus reduce the tracking error, while adding PID to the Q-learning can improve the control performance of the single Q-learning. Consequently, the proposed Q-learning–PID has the superior advantages over traditional Q-learning and PID.

The responses of transverse position and longitudinal position with their reference positions are shown in Figs. 17 and 18, respectively. The tracking curves of the mobile robot using PID, Q-learning and Q-learning–PID, are, respectively, shown in Fig. 19.

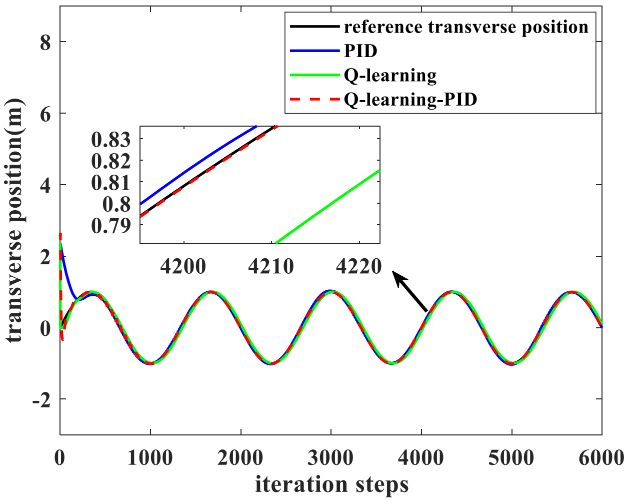

Fig. 17 Response of transverse position

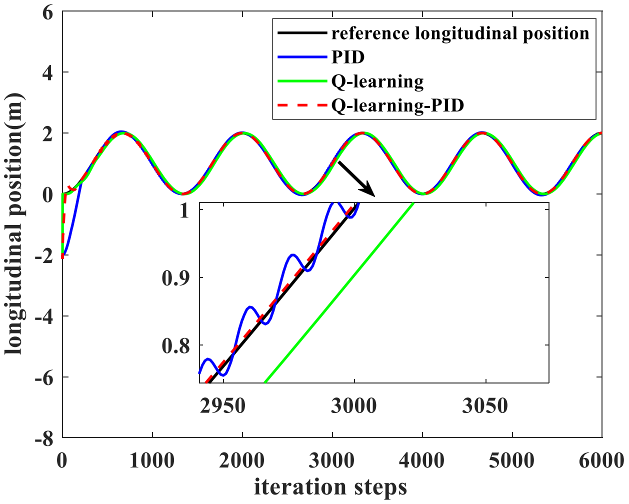

Fig. 18 Response of longitudinal position

In Figs. 17, 18 and 19, the reference signals are still provided using formula (11). Figure 17 denotes the tracking response of transverse position; Fig. 18 shows the tracking response of longitudinal position; Fig. 19 expresses the trajectories of mobile robot with different control methods. From Figs. 17 and 18, it can be seen that using the Q-learning–PID, the mobile robot can converge to the reference signals with the best performance, compared to the PID and Q-learning. The PID and Q-learning

Table 1 MSE of mobile robot with different control methods

| MSE | Control methods | | |
|---|---|---|---|
| | PID | Q-learning | Q-learning–PID |
| MSE of transverse position | $2.6715 \times 10^{-5}$ | 0.0085 | $1.1577 \times 10^{-5}$ |
| MSE of longitudinal position | $2.9617 \times 10^{-5}$ | $3.5701 \times 10^{-5}$ | $3.0629 \times 10^{-6}$ |
| MSE of angle | $1.2181 \times 10^{-4}$ | 0.0074 | $1.0074 \times 10^{-5}$ |

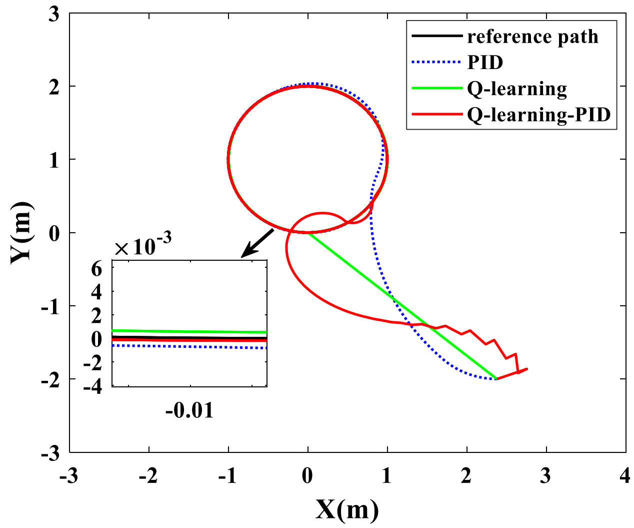

**Fig. 19** Trajectory tracking of mobile robot

controllers have tracking residual in the transverse position and even oscillation in the longitudinal position tracking. That is, the tracking results of Q-learning–PID are better compared to Q-learning and PID.

As shown in Fig. 19, at the beginning stage of tracking, the Q-learning selects the optimal strategy by trial and error and combines itself with PID after some iterations. In this simulation test, the mobile robot follows the reference trajectory after about 500 iteration steps, which means the Q-learning gains the best combination with PID.

## 5 Conclusion

Q-learning has ability of learning, and achieves the best action by trial and error, but performs poorly in complicated conditions and is prone to dimension disaster, so its control ability needs to be improved. PID control is effective and robust, but lacks in learning ability. This paper presents a novel algorithm named Q-learning–PID to overcome the above deficits. Experiment simulations illustrate that the Q-learning–PID can ensure the value function converges to zero, indicating that the path tracking of Q-learning–PID is better than the single Q-learning or PID. This novel controller might be applied in more engineering areas and deserves further studies.

**Acknowledgements** This work was supported by the Fundamental Research Funds for the Central Universities under Grant 2019JBM004.

## References

Anderlini E, Forehand David I M, Stansell P, Xiao Q, Abusara M (2016) Control of a point absorber using reinforcement learning. IEEE Trans Sustain Energy 7(4):1681–1690

Anderlini E, Forehand DIM, Bannon E, Xiao Q, Abusara M (2018) Reactive control of a two-body point absorber using reinforcement learning. Ocean Eng 148:650–658

Beghi A, Rampazzo M, Zorzi S (2017) Reinforcement learning control of transcritical carbon dioxide supermarket refrigeration systems. IFAC PapersOnline 50(1):13754–13759

Carlucho I, De Paula M, Villar SA, Acosta GG (2017) Incremental Q-learning strategy for adaptive PID control of mobile robots. Expert Syst Appl 80:183–199

Doya K (2000) Reinforcement learning in continuous time and space. Neural Comput 12(1):219–245

Fernandez-Gauna B, Osa JL, Graña M (2018) Experiments of conditioned reinforcement learning in continuous space control tasks. Neurocomputing 271:38–47

Genders W, Razavi S (2018) Evaluating reinforcement learning state representations for adaptive traffic signal control. Proc Comput Sci 130:26–33

Görges D (2017) Relations between model predictive control and reinforcement learning. IFAC PapersOnLine 50(1):4920–4928

Günther J, Pilarski PM, Helfrich G, Shen H, Diepold K (2016) Intelligent laser welding through representation, prediction, and control learning: an architecture with deep neural networks and reinforcement learning. Mechatronics 34:1–11

Hernández-del-Olmo F, Gaudioso E, Dormido R, Duro N (2018) Tackling the start-up of a reinforcement learning agent for the control of wastewater treatment plants. Knowl Based Syst 144:9–15

Huang J, Wen C, Wang W, Jiang Z-P (2014) Adaptive output feedback tracking control of a nonholonomic mobile robot. Automatica 50:821–831

Huang D, Zhai J, Ai W, Fei S (2016) Disturbance observer-based robust control for trajectory tracking of wheeled mobile robots. Neurocomputing 198:74–79

Jiang H, Zhang H, Cui Y, Xiao G (2018a) Robust control scheme for a class of uncertain nonlinear systems with completely unknown dynamics using data-driven reinforcement learning method. Neurocomputing 273:68–77

Jiang Z, Fan W, Liu W, Zhu B, Jinjing G (2018b) Reinforcement learning approach for coordinated passenger inflow control of urban rail transit in peak hours. Transp Res 88:1–16

Klancar G, Skrjanc I (2007) Tracking-error model-based predictive control for mobile robots in real time. Robot Auton Syst 55:460–469

Kofinas P, Doltsinis S, Dounis AI, Vouros GA (2017) A reinforcement learning approach for MPPT control method of photovoltaic sources. Renew Energy 108:461–473

Kubalik J, Alibekov E, Babuska R (2017) Optimal control via reinforcement learning with symbolic policy approximation. IFAC PapersOnLine 50(1):4162–4167

Kumar A, Sharma R (2018) Linguistic Lyapunov reinforcement learning control for robotic manipulators. Neurocomputing 272:84–95

Leena N, Saju KK (2016) Modelling and trajectory tracking of wheeled mobile robots. Proc Technol 24:538–545

Li Y, Chen L, Tee KP, Li Q (2015) Reinforcement learning control for coordinated manipulation of multi-robots. Neurocomputing 170:168–175

Li P, Dargaville R, Cao Y, Li D, Xia J (2017a) Storage aided system property enhancing and hybrid robust smoothing for large-scale PV Systems. IEEE Trans Smart Grid 8(6):2871–2879

Li R, Liwei Zhang L, Han JW (2017b) Multiple vehicle formation control based on robust adaptive control algorithm. IEEE Intell Transp Syst Mag 9(2):41–51

Li S, Ding L, Gao H, Chen C, Liu Z, Deng Z (2018a) Adaptive neural network tracking control-based reinforcement learning for wheeled mobile robots with skidding and slipping. Neurocomputing 283:20–30

Li P, Li R, Cao Y, Li D, Xie G (2018b) Multiobjective sizing optimization for island microgrids using a triangular aggregation model and the Levy–Harmony algorithm. IEEE Trans Ind Inf 14(8):3495–3505

Liu F, Song YD (2011) Stability condition for sampled data based control of linear continuous switched systems. Syst Control Lett 60(10):787–797

Lopez-Guede JM, Estevez J, Garmendia A, Graña M (2018) Making physical proofs of concept of reinforcement learning control in single robot hose transport task complete. Neurocomputing 271:95–103

Mahmoodabadi MJ, Abedzadeh Maafi R, Taherkhorsandi M (2017) An optimal adaptive robust PID controller subject to fuzzy rules and sliding modes for MIMO uncertain chaotic systems. Appl Soft Comput 52:1191–1199

Mendonça Matheus R F, Bernardino HS, Neto RF (2018) Reinforcement learning with optimized reward function for stealth applications. Entertain Comput 25:37–47

Miljkovic Z, Mitić M, Lazarevic M, Babic B (2013) Neural network reinforcement learning for visual control of robot manipulators. Expert Syst Appl 40(5):1721–1736

Padmanabhan R, Meskin N, Haddad WM (2015) Closed-loop control of anesthesia and mean arterial pressure using reinforcement learning. Biomed Signal Process Control 22:54–64

Padmanabhan R, Meskin N, Haddad WM (2017) Reinforcement learning-based control of drug dosing for cancer chemotherapy treatment. Math Biosci 293:11–20

Ramanathan P, Mangla KK, Satpathy S (2018) Smart controller for conical tank system using reinforcement learning algorithm. Measurement 116:422–428

Shah H, Gopal M (2016) Model-free predictive control of nonlinear processes based on reinforcement learning. Int Fed Autom Control 49(1):89–94

Shi H, Lin Z, Zhang S, Li X, Hwang K-S (2018) An adaptive decision-making method with fuzzy Bayesian reinforcement learning for robot soccer. Inf Sci 436–437:268–281

Simba KR, Uchiyama N, Sano S (2016) Real-time smooth trajectory generation for nonholonomic mobile robots using Bézier curves. Robot Comput Integr Manuf 41:31–42

Suruz Miah M, Gueaieb W (2014) Mobile robot trajectory tracking using noisy RSS measurements: an RFID approach. ISA Trans 53:433–443

Wang H, Fei Richard Yu, Zhu L, Tang T, Ning B (2015) A cognitive control approach to communication-based train control systems. IEEE Trans Intell Transp Syst 16(4):1676–1689

Xiao G, Zhang H, Luo Y, Qiuxia Q (2017) General value iteration based reinforcement learning for solving optimal tracking control problem of continuous-time affine nonlinear systems. Neurocomputing 245:114–123

Yang L, Nagy Z, Goffin P, Schlueter A (2015) Reinforcement learning for optimal control of low exergy buildings. Appl Energy 156:577–586

Zhan Y, Ammar HB, Taylor ME (2017) Scalable lifelong reinforcement learning. Pattern Recognit 72:407–418