INVERTED PENDULUM CONTROL OF DOUBLE Q-LEARNING REINFORCEMENT LEARNING ALGORITHM BASED ON NEURAL NETWORK

Daode ZHANG[1]*, Xiaolong WANG[1], Xuesheng LI[1], Dong WANG[1]

Aiming at the problem that the stability control time is too long in the control of the inverted pendulum, this paper combines the neural network with the double Q-learning reinforcement learning algorithm on the basis of analyzing the reasons of the delay caused by the over-estimation of the traditional Q-learning reinforcement learning algorithm. By using the generalization ability of the neural network, the continuous state space input and continuous action space output of inverted pendulum can be realized at the same time. Double Q-learning algorithm is used to solve the problem that Q-learning algorithm takes a long time to stabilize the swing caused by overestimation. The inverted pendulum simulation model is established in MATLAB. The Double Q-learning algorithm and the Q-learning algorithm are compared in the inverted pendulum control effect. The results show that the proposed method can shorten the learning time of the inverted pendulum into the inverted state. At the same time, the swing angle of the inverted pendulum can be reduced, and the control effect is better.

Keywords: Inverted Pendulum, Reinforcement Learning, Double Q-learning, Neural Network

Introductions

Inverted pendulum is a high-order, unstable, nonlinear continuous control system [1]. Its control method is widely used in industrial control fields such as military and robotics. [2][3] Therefore, the research on the control object of inverted pendulum is theoretical and methodical. It has important guiding significance for practice in industrial control. At present, in the research on the control of inverted pendulum, there are mainly control methods based on traditional PID [4] and control methods based on reinforcement learning. Intensive learning mainly includes AHC (Adaptive Heuristic Critic) method [5] and Q-learning method in controlling inverted pendulum. The control effect based on Q-learning control method has better effect than other control methods.

Although the traditional PID-based inverted pendulum control can realize the control of the inverted pendulum, the determination of the PID parameters

[1] School of Mechanical Engineering, Hubei University of Technology, Wuhan, China.
*E-mail: zhangdaode012@yeah.net.

relies too much on human experience, and multiple adjustments are needed to determine the optimal parameters. [4][14] However, using the reinforcement learning algorithm to control the inverted pendulum can effectively avoid the problem that the traditional PID control method relies too much on human experience (Jiang Guofei and Wu Yupu)[6]. In the paper, BP neural network was combined with Q learning to successfully solve the inverted pendulum control in continuous state space. [9][13] The control method realized the inverted pendulum control after 1000 steps. On the basis of this, Zhang Tao and Wu Hansheng[7] combined the neural network with the Q-learning algorithm to solve the continuous problem of a certain range of motion output space based on the continuous input of state space.[11] The control method is implemented after 1500 steps with control of the inverted pendulum. However, when these inversion pendulums are controlled based on the Q-learning intensive learning method, there is a problem that it takes a long time to achieve the inversion due to overestimation.[8] Based on the predecessors, this paper proposes a reinforcement learning algorithm based on neural network for Double Q-learning, which realizes continuous state space input and continuous action space output of inverted pendulum while avoiding overestimation in a single Q-learning algorithm. It solves the problem that the learning time is too long to control the pendulum upside down in the traditional method and has a good application value.

1. The inverted pendulum system and its control algorithm

1.1 Mathematical model of inverted pendulum

The inverted pendulum system of the control object is shown in fig1. The car ***M*** moves freely on the track and the end of the inverted pendulum ***m*** is hinged on the top of the car for free rotation.

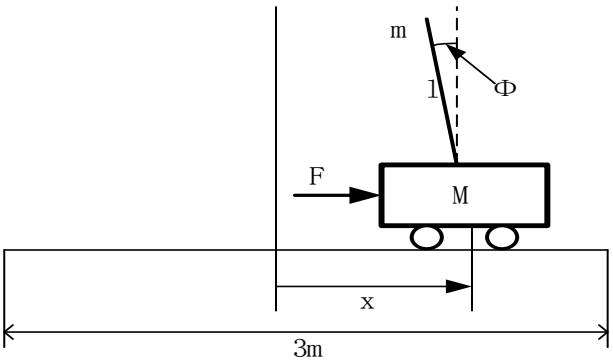

Fig.1.Diagram of inverted pendulum system

The purpose of the control is to push the left and right movement of the

trolley to keep the inverted pendulum upside down and not collide with both ends of the track. The system input is the force F acting on the trolley, and the output is four state variables:

χ -the displacement of the trolley;

ϕ -inverted pendulum deviated from the angle of the vertical direction;

$\dot{\chi}$ -the speed of the car;

$\dot{\phi}$ -Angular velocity of an inverted pendulum.

Parameters of linear first-order inverted pendulum are shown in Table 1:

Table 1

Parameters of linear first-order inverted pendulum

Inverted pendulum trolley quality	0.618kg
Quality of pendulum rod of inverted pendulum	0.0737kg
Inverted pendulum length	0.35m
Pendulum center of mass to shaft distance	0.1225m
Gravity acceleration	9.8m/s^2

Using the Lagrange equation to derive the state equation of the linear inverted pendulum state space:

$$\begin{bmatrix} \dot{\chi} \\ \dot{\phi} \\ \ddot{\chi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-4}{(4M+m)} & \frac{3mg}{(4M+m)} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-3}{(4M+m)l} & \frac{3(M+m)g}{(4M+m)l} & 1 \end{bmatrix} \begin{bmatrix} \chi \\ \dot{\chi} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{4}{(4M+m)} \\ 0 \\ \frac{3}{(4M+m)l} \end{bmatrix} F \quad (1)$$

$$y = \begin{bmatrix} \chi \\ \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \chi \\ \dot{\chi} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} F \quad (2)$$

Where $\ddot{\chi}$ is the acceleration of the car's motion and $\ddot{\phi}$ is the angular acceleration of the swing of the pendulum.

1.2 Inverted pendulum control algorithms

For this control model of inverted pendulum, many classical control methods are applied to the control of this model. The traditional PID-based inverted pendulum control method can perform well, but the PID-controlled inverted pendulum has the fatal weaknesses such as difficult adjustment,

over-reliance on human experience, debugging PID parameters for a long time to achieve pendulum upside down.[2][14] Therefore, the inverted pendulum control method based on PID is quickly replaced by other control methods. The control method of reinforcement learning is applied to the control of the inverted pendulum because of its high degree of automation, good stability, and short control time. As another method of reinforcement learning, Q-learning is also applied to the control of inerted pendulum. [4-5]The predecessors realized the control of the inverted pendulum in a short time based on the continuity of the state space and the action space [9], but the methods of this reinforcement learning still have room for improvement in the learning time. In this paper, a double Q-learning reinforcement learning algorithm based on neural network is proposed to solve the problem of long learning time in reinforcement learning.

2. Double Q-learning reinforcement learning algorithm

2.1 Q-learning algorithm

Reinforcement learning is essentially a Markov decision process (MDP). The Markov feature indicates that state transitions are only related to the current state and current actions, and are independent of previous actions and states.[10] Given a Markov process, the mapping between state space and action space is defined as a strategy π. Reinforcement learning is to obtain the optimal control strategy π^* only by interacting with the environment, learning and exploring without knowing the state transition equation or the return function of the system. At the same time, the best strategy π^* for defining MDP is the maximum expected total return from the environment under the current strategy, the maximum value $E\left\{\sum_{i=0}^{\infty}\gamma^i r_{k+i}\right\}$, of which $0<\gamma<1$ is the discount factor. Q-learning algorithm learning process is that at each time step $k=1,2,\cdots$, the controller observes the state of the system, chooses the decision, receives the immediate return, and transfers the system to the next state with probability. By seeking the optimal strategy, it maximizes the return and expectation of each time step in the future. Given a policy π, the Q value is defined as the following formula (3):

$$Q^{\pi}(X,A)=R(X,A)+\gamma\sum_{k}P_{X_kX_{k+1}}(A_k)V^{\pi}(X_{k+1}) \tag{3}$$

The current return $R(X,A)$ is the formula (4), and the system output action $V^{\pi}(X_{k+1})$ is selected according to the strategy (5).

$$R(X,A)=E\{r\mid X,A\} \tag{4}$$

$$V^{\pi}(X_{k+1})=\max_{a}Q^{\pi}(X_{k+1},a) \tag{5}$$

Q-learning learning algorithm is implemented as follows: at each time step *k*, observe the current state X_k, select and execute action A_k, and then observe the subsequent state X_{k+1}, receive immediate return r_k. Then adjust the value of Q_{k-1} according to formula

$$Q_k(X, A) = \begin{cases} (1 - \beta_k)Q_{k-1}(X, A) + \beta_k[r_k + \gamma V_{k-1}(X_{k+1})], & (X, A) = (X_k, A_k) \\ Q_{k-1}(X, A), & \forall(X, A) \neq (X_k, A_k) \end{cases} \tag{6}$$

where β_k is a learning factor. Ben[13] proved that the learning factor sequence $\{\beta_k\}$ satisfies certain conditions, if any (X, A) binary group can be performed with equation (6). Poor multiple iterations, then when $k \to \infty$, $Q_k(X, A)$ converges to $\mathrm{Q}_{\mathrm{k}}^{\pi^*}$ with probability 1. Where $\mathrm{Q}_{\mathrm{k}}^{\pi^*}$ is the expectation of the return obtained by the control A in the state X under the optimal strategy π^*.

2.2 Double Q-learning algorithm

The Double Q-learning algorithm is based on Q-learning. Since the action is selected according to the strategy in the Q-learning algorithm, the action is always maximized according to the formula (5), so that when the inverted pendulum is controlled, the action value is overestimated, thereby slowing down the learning speed. In order to avoid the overestimation of the action selection of the Q-learning algorithm in the inverted pendulum control process, the Double Q-learning algorithm is proposed. In the Double Q-learning algorithm, by training two independent value functions for action selection and action evaluation, the update procedure of Double Q-learning's reinforcement learning algorithm is as follows (7):

$$\begin{cases} Q_{k+1}^A(X_k, A_k) = Q_k^A(X_k, A_k) + \beta_k\left(r_k + \gamma Q_k^B\left(X_{k+1}, \underset{a}{\operatorname{argmax}}\, Q_k^A(X_{k+1}, A)\right) - Q_k^A(X_k, A_k)\right) \\ Q_{k+1}^B(X_k, A_k) = Q_k^B(X_k, A_k) + \beta_k\left(r_k + \gamma Q_k^A\left(X_{k+1}, \underset{a}{\operatorname{argmax}}\, Q_k^B(X_{k+1}, A)\right) - Q_k^B(X_k, A_k)\right) \end{cases} \tag{7}$$

Where r_k is the immediate return obtained after the action is performed in the current state. $0 < \gamma < 1$ is a discount factor, which is used to control whether our evaluation of motion is short-sighted. The larger γ is, the more attention is paid to the learning process, and the long-term impact of the current action on the subsequent learning process is ignored. When updating the traditional Q-learning algorithm, the current network is used to evaluate the impact of the current action on the follow-up. In this way, when maximizing the

action, it is easy to make the action choice too large because of the excessive pursuit of the maximum reward. The double Q-learning algorithm proposed in this paper chooses different value functions for action evaluation and action selection. When adjusting, it no longer uses formula (6) but adds different value functions to the algorithm, so that when updating Q value, different value functions for action evaluation and action selection will not lead to excessive action selection and overestimation due to the pursuit of maximum reward.

In addition, the model free reinforcement learning algorithm of double Q-learning only needs to optimize the action of the inverted pendulum directly through the interaction with the environment, and does not need to learn the environment model, thus avoiding the model divergence caused by the deviation of the environment model. The model-free double Q-learning algorithm can gradually converge to the optimal solution through a certain period of learning.

3. Neural Network Based Double Q-learning Reinforcement Learning Algorithm for Inverted Pendulum

In the reinforcement learning method, the only feedback signal that the controller can get from the environment is the reward and punishment value when the inverted pendulum deviates from the vertical direction angle $\pm 3^{o}$ or the trolley exceeds the effective range ± 1.5 m of the track in the current state $X = (\chi, \dot{\chi}, \phi, \dot{\phi})$ after the implementation of action A. The defined reward and punishment rules are of formula (8):

$$r = \begin{cases} -1, & |\chi| > 1.5 \quad or \quad |\phi| > 3^{o} \\ l\cos\phi, & else \end{cases} \tag{8}$$

Considering that the current state acquired by the system is of 4 parameters, it is also necessary to avoid the large number of operations caused by the complicated network structure, and slow down the time when the inverted pendulum reaches a stable state. At the same time, the continuous state space and continuous motion output space are approximated. [11-12] This paper uses a simple BP neural network to generalize the Q value of the untrained state-decision binary group. Therefore, the neural network based Double Q-learning reinforcement learning inverted pendulum control system structure diagram is shown in Fig 2:

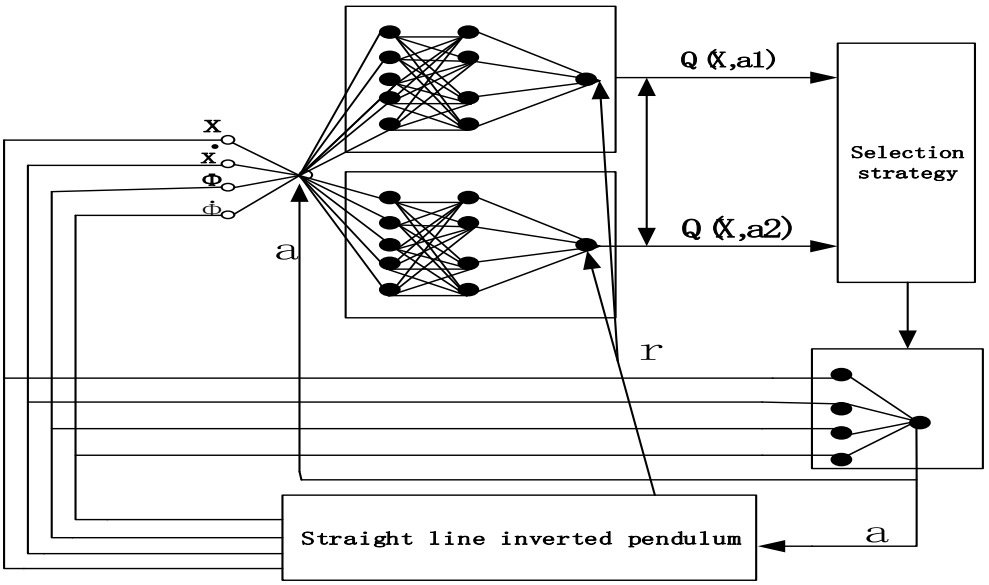

Fig.2. Inverted pendulum control system structure

State of the system $X = \left(\chi,\ \dot{\chi},\ \phi,\ \dot{\phi}\right)$ as input to two BP neural networks, the two BP neural networks correspond to different network weights W,D. The control system needs to output the next action according to the reward and evaluation of the current action. The reward and punishment rules are defined by (8), which shows that the smaller the reward and punishment value, the more likely the control failure is. However, we should not only pursue higher rewards and penalties, but also choose the action that makes the reward the biggest. This will cause overestimation and slow down the learning speed and unstable control. In fact, the double Q-learning reinforcement learning algorithm is to retransmit the failure signal in the update iteration of Q value at each time step, and judge the current decision-making according to Q value. It is only based on formula (7) when calculating the updated Q value. Different value functions are used to calculate action selection and action evaluation. In this way, the problem of overestimation caused by using the same value function for action selection and action evaluation can be effectively avoided, and the action output is relatively conservative. Therefore, using double Q-learning algorithm to control the inverted pendulum, the change of pendulum angle is moderate in theory, and it is easier to achieve stable state than Q-learning algorithm.

The whole control system works as follows: at each time step k, the current state X_k is obtained, and different BP values are calculated with $Q_k^A = Q\left(X_k, a_k, W_k\right), Q_k^B = Q\left(X_k, a_k, D_k\right)$ and output through two BP neural networks. According to a certain strategy, the current decision a_k is selected, and the output neural network weight V is adjusted according to the current decision.

After neural network nonlinear activation mapping, the output of the neural network is based on the formula (9).The system updates the state-decision Q-value according to equation (7), then according to the error $e_A = Q^A(X_k, a_k) - Q(X_k, a_k, W_k), e_B = Q^B(X_k, a_k) - Q(X_k, a_k, D_k)$ update the weight of the BP neural network W,D. The least square method is used to iterate continuously, so that the neural network output approaches the ideal output $Q^A(X_k, a_k), Q^B(X_k, a_k)$.

$$a = f\left(\sum_{i}^{n} V_i X_i - c\right) \tag{9}$$

4. Simulation Implementation and Analysis of Results

As mentioned in the previous section, considering the number of system state variables and the continuous action space input and the continuous action space output requirements, and also taking into account the time when the inverted pendulum reaches a steady state, the state input uses two BP neural networks and each network segment. In the third layer, the input layer and the hidden layer each has five neuron nodes, and the output layer has one neuron node. The action output adopts a BP neural network, the network is divided into two layers, the input layer has four neuron nodes, and the output layer has one neuron node. The input of the BP neural network is normalized so that it is distributed between $[-1, 1]$, the learning factor of the Q-learning algorithm is $\beta = 0.2$, the discount factor is $\gamma = 0.95$, and the reward and punishment values obtained are based on the formula (9). The BP neural network activation function of the action output uses the tanh function and the output neuron threshold $c = 0.5$, and its output range is between $(-5,5)$. The neural network-based Double Q-learning reinforcement learning algorithm is used to control the inverted pendulum control model. The simulation experiment is carried out in MATLAB. [15] The number of trials exceeds 600 times in each experiment, or the average number of steps in a trial exceeds 1000 steps. Learn to stand upside down and start another experiment. The double Q-learning algorithm based on neural network realizes the effect of inverted pendulum and the effect of inverted pendulum based on Q-learning algorithm.

(1) Comparison between Q-learning and Double Q-learning algorithm to control the swing angle

The neural network based Double Q-learning algorithm and the neural network based Q-learning algorithm realize the change of the swing angle of the inverted pendulum control with the number of learning steps as shown in Fig. 3:

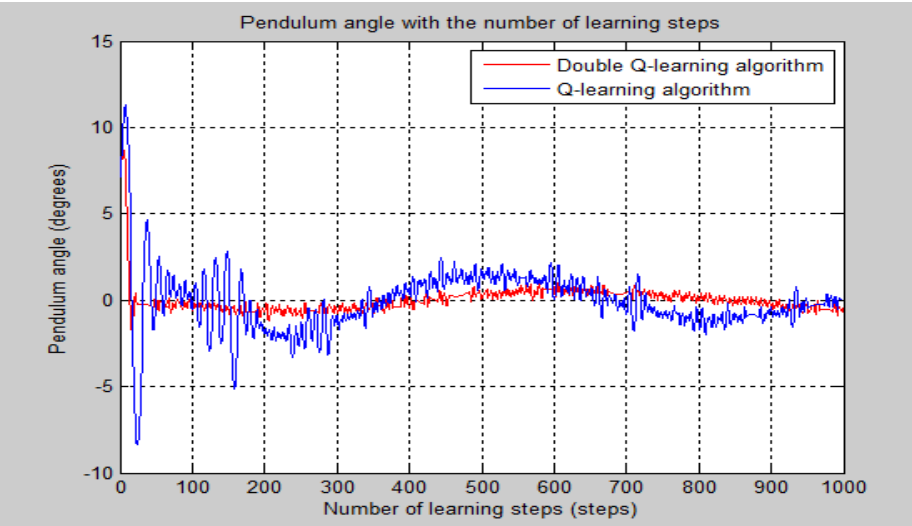

Fig.3.Pendulum angle with the number of learning steps

The simulation results of simulation Fig 3 show that: In the 1000-step learning process, the Double Q-learning reinforcement learning algorithm is quickly stabilized by the 30-step learning in the initial state until the pendulum angle is around and fluctuates within the range of $0^o - 3^o$; The basic Q-learning reinforcement learning algorithm is stable to the pendulum angle after 170 steps in the initial state, and fluctuates within the range of $0^o - 3^o$. Compared with the Q-learning algorithm, the double Q-learning algorithm based on neural network can avoid over-estimation and shorten the time needed to control the inverted pendulum to reach a stable state. After the pendulum reaches a stable state, the pendulum deviates from the vertical by a smaller angle.

(2) Comparative Analysis of Q-learning and Double Q-learning Algorithms for Controlling Car Displacement

Compared the neural network based Double Q-learning algorithm with the neural network based Q-learning algorithm, the displacement of the trolley with the number of learning steps is shown in Fig 4:

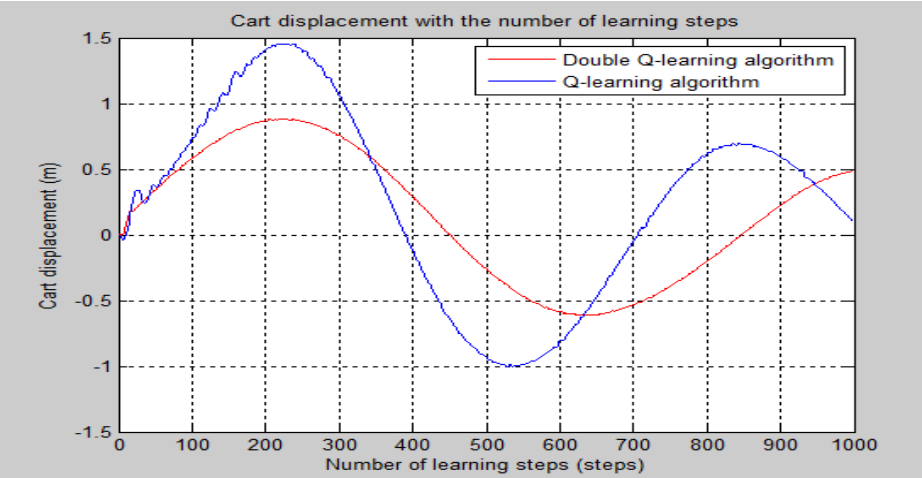

Fig.4.Pendulum angle with the number of learning steps

The simulation results in Fig.4 shows that in the 1000-step learning process, the control method based on Double Q-learning is used to control the inverted pendulum. The maximum displacement of the trolley is 0.8m, and the displacement of the trolley changes gently. The base Q-learning reinforcement learning control method has a maximum displacement of 1.48m for the inverted pendulum control process and a slight oscillation of the displacement of the trolley before 200 steps.

(3) Comparative Analysis of Learning Steps between Double Q-learning Algorithm and Other Control Methods

The learning steps of the double Q-learning reinforcement learning method for inverted pendulum control proposed in this paper are compared with those of other control methods in terms of learning time as shown in Table 2.

Table 2

Comparisons of learning steps with different methods

Control Method	Number of learning steps (steps)
AHC	6000
Q-learning(Continuous state space)	1000
Q-learning(State and motion space are continuous)	170
Double Q-learning(State and motion space are continuous)	30

Among them, each learning step takes 0.02 seconds. The comparison of the results in Table 2 shows that the learning method based on the neural network-based Double Q-learning algorithm improves the learning efficiency by 99% compared with the AHC algorithm; Learning efficiency is improved by 97% compared to the Q-learning algorithm with continuous state space; Learning efficiency increased by 82% compared to the Q-learning algorithm with state and motion continuous. In this paper, a double Q-learning reinforcement learning control method based on neural network is proposed. After the state space and action space are continuous by using neural network, the double Q-learning algorithm is used to solve the problems of long swing time, large swing angle fluctuation and large car displacement caused by over-estimation in traditional Q-learning algorithm. The learning efficiency has been greatly improved.

5. Conclusion

This paper analyzes the reason why the Q-learning reinforcement learning control method causes the inverted pendulum to achieve the upside down time long, proposes neural network based double Q-learning reinforcement learning inverted pendulum control method. As the same time, the mathematical model of the inverted pendulum is established, and the simulation is carried out in MATLAB. The inverted pendulum control method of double Q-learning reinforcement learning based on neural network proposed in this paper effectively avoids the problem of long learning time caused by over estimation of Q-learning reinforcement learning. The swing angle fluctuates in a small range, the control effect is better. In addition, this paper also combines the double Q-learning and neural network to fit the continuous motion space output, and can also be extended to large-scale function approximation, which has good application and research value.

Acknowledgement

This work is supported by Wuhan science and technology support program (No. 2017010201010137) and the National Natural Science Foundation of China (No.61976083).

REFERENCES

[1]. *Wenjie Mao.* Application of reinforcement learning in inverted pendulum and balance control. Beijing University of Technology. 2018.

[2]. *Jiulong Jiang, Xueren Li, Jun Du.* Research on Active Disturbance Rejection Decoupling Control Method for Inverted Pendulum. Measurement and control technology, 32(6): 87-91. 2013

[3]. *Han Fujian.* Development and significance of inverted pendulum system. Shandong industrial technology. (17).2014
[4]. *Zhi Yang*. Summary of Key Technologies of Industrial Tuning PID Regulator. Chemical automation and instrumentation. 2000.
[5]. *C.W Anderson*. Learning to control an inverted pendulum using neural networks. IEEE Control System Magazine, 9(3): 31-37. 1989
[6]. *Guofei Jiang, Cangpu Wu*. Inverted pendulum control based on Q-learning algorithm and BP neural network. Journal of Automation, 24(5): 622-666. 1998
[7]. *Tao Zhang, Hansheng Zhang.* Intensive Learning Algorithms Based on Neural Networks for Implementing Inverted Pendulum. Computer simulation, 2006.
[8]. *Hado van Hasselt.* Double Q-learning. Advances in Neural Information Processing Systems, 23:2613–2621, 2010.
[9]. *Borja Fernandez-Gauna, Juan Luis Osa, Manuel Graña.* Experiments of conditioned reinforcement learning in continuous space control tasks. Neurocomputing, 271. 2018
[10]. *M. Riedmiller.* Concepts and facilities of a neural reinforcement learning control architecture for technical process control. Journal of Neural Computing and Application, 8:323–338, 2000.
[11]. *A.G Barto, R S Sutton, C.W Anderson.* Neuron like adaptive elements that can solve difficult learning control problems. IEEE Trans on SMC, 13(5): 834-846. 1983
[12]. *M. Lagoudakis and R. Parr.* Least-squares policy Iteration. Journal of Machine Learning Research, 4:1107–1149, 2003.
[13]. *Ben J.A. Kröse.* Learning from delayed rewards. Robotics and Autonomous Systems, 1995,15(4).
[14]. *Lixin Wei, Hao wang*. Fractional PID parameter optimization of inverted pendulum based on particle swarm optimization. Control engineering, 26(02):196-201. 2019
[15]. *Zhenyuan Wu, Yanying Guo*. Modeling and control of inverted pendulum system. Science and technology communication, 10(17):131-134. 2008