Wireless Networks
https://doi.org/10.1007/s11276-019-02225-x

Design and application of adaptive PID controller based on asynchronous advantage actor–critic learning method

Qifeng Sun[1] · Chengze Du[1] · Youxiang Duan[1] · Hui Ren[1] · Hongqiang Li[2]

Abstract
To address the problems of the slow convergence and inefficiency in the existing adaptive PID controllers, we propose a new adaptive PID controller using the asynchronous advantage actor–critic (A3C) algorithm. Firstly, the controller can train the multiple agents of the actor–critic structures in parallel exploiting the multi-thread asynchronous learning characteristics of the A3C structure. Secondly, in order to achieve the best control effect, each agent uses a multilayer neural network to approach the strategy function and value function to search the best parameter-tuning strategy in continuous action space. The simulation results indicate that our proposed controller can achieve the fast convergence and strong adaptability compared with conventional controllers.

Keywords Reinforcement learning · Asynchronous advantage actor–critic · Adaptive PID control · Stepping motor

1 Introduction

The PID controller is a control loop feedback mechanism which is widely used in industrial control system [1]. Based on the investigation of conventional PID controller, the adaptive PID controller adopts online parameter adjustment method according to the state of the system, therefore it has better system adaptability. The fuzzy PID controller [2] adopts the ideology of matrix estimations [3, 4]. In order to satisfy the requirement of the self-tuning PID parameters, the method adjusts the parameters by querying fuzzy matrix table. The limitation of this method is that it needs much more prior knowledge. Moreover, this method has a large number of parameters that is needed to be optimized [5].

The adaptive PID controller [6, 7] approximates nonlinear structure by neural networks, which can achieve effective control without identifying the complex nonlinear controlled object. But, it is difficult to obtain the teacher signals in the supervised learning process. The evolutionary adaptive PID controller [8] has difficulty in achieving real-time control due to the fact that it requires less prior knowledge [9]. The adaptive PID controller, which is based on reinforcement learning [10], solves the problem by obtaining the teacher's signal in unsupervised learning process. And the optimization of the control parameters is simple. The actor–critic (AC) adaptive PID [11, 12] is the most widely used reinforcement learning controller. However, the convergence speed of the controller is affected by the correlation of the learning data in the AC algorithm [13].

Google's DeepMind team proposed the asynchronous advantage actor–critic (A3C) learning algorithm [14, 15]. This algorithm adopts multi-strategies to train multiple agents in parallel, each agent will experience different learning state. So the correlation of the learning sample is broken while improving the computational efficiency [16]. This algorithm has been applied in many fields [17, 18].

The proposed method aims to improve the convergence and adaptive ability of the PID controller. To achieve this purpose, we use the A3C algorithm that enhance the learning rate to train agent in the parallel threads. And two BP neural networks are used to approach policy function and value function separately. The experiments show that the proposed algorithm outperforms the conventional PID controlling algorithms. The rest of paper is arranged as

✉ Qifeng Sun
Sunqf@upc.edu.cn

[1] China University of Petroleum, Qingdao 266580, China

[2] China Petrochemical Group Victory Drilling Technology Research Institute, Dongying 257000, China

Published online: 31 December 2019

fellows. Starting from a brief description of PID controller in Sect. 2 and 3, we introduce our new approach in Sect. 4 and show experimental results in Sect. 5. We conclude the paper in Sect. 6.

2 Related work

The conventional PID controlling algorithms can be roughly classified into three categories: the fuzzy PID controller, neural network PID controller and reinforcement learning PID controller.

2.1 Fuzzy PID controller

Tang [19] proposed a method that combined the fuzzy math with the PID control. However, this method still had some limitations such as that it required a lot of manual experience to establish the rule table. Besides, the rule table was often only adapted to a specific application scenario. To address these issues, Sun [20] developed a fuzzy PID controller based on improved genetic algorithm, which used multiple fuzzy control rules to adjust parameters by genetic algorithm. The controller abandoned the plenty of manual work and set up an exclusive rule under the environment. Spired by the idea of the work, Zhu [21] added the normalized velocity parameter reflecting the response of the system based on the adjusting factor of fuzzy rules. The method aimed to change the mapping between input and output variables with the fuzzy subsets so that it made the controller be able to divide the error and the rate of error into multiple control stages.

2.2 Adaptive controller based on neural network

Liao [22] proposed a method utilizing the neural network to reinforce the performance of PID controller for the nonlinear system. Although the initial parameters of neural network could be determined by artificial test, it could not ensure the reliability of the manual result. Based on this, Li [23] adopted the genetic algorithm to obtain the optimal initial parameters of the network. However, the genetic algorithm was easily to fall into local optimum. In order to solve the problem, Patel [24] appended the immigration mechanism, 10% of the elite population and the inferior population were selected as the variant population, to the neural network adaptive PID controller (MN-PID). In addition, Nie [25] presented an adaptive chaos particles swarm optimization for tuning parameters of PID controller (CSP-PID) to avoid the local minima.

2.3 Reinforcement learning adaptive controller

Aziz Khater [26] proposed a PID controller that combining the ASN reinforcement-learning network with fuzzy math. Despite this method did not need too much accurate training samples compared the neural network PID, its structure was too complex to guarantee the real-time performance. In view of this point, Adel [10] designed an adaptive PID controller based on AC algorithm. This controller had simple structure with one RBF network. However, its speed of convergence was slow owing to the relevance in learning sample of AC algorithm.

3 Basic structure of PID controller

Incremental PID is an algorithm of PID control by increment of control volume. The typical control system structure is shown in Fig. 1. Besides, its formula is as follows:

$$\begin{aligned} u(t) &= u(t-1) + \Delta u(t) \\ &= u(t-1) + K_i(t)e(t) + K_p(t)\Delta e(t) \\ &\quad + K_d(t)\Delta^2 e(t) \end{aligned} \tag{1}$$

where

$$\begin{aligned} e(t) &= y'(t) - y(t) \\ \Delta e(t) &= e(t) - e(t-1) \\ \Delta^2 e(t) &= e(t) - 2 * e(t-1) + e(t-2) \end{aligned}$$

$y'(t)$, $y(t)$, $e(t)$, $\Delta e(t)$, $\Delta^2 e(t)$ represents the current actual signal value, the output value of the current system, the system output error, the first-order difference of error and the second-order difference of error respectively. In the form 1, incremental PID is cancelled the integral summation, so it saves the time of calculation. Besides, it influence the system lightly when the system is broken. In the synthesis the factor, the incremental PID is optimum choice for the practical application.

4 A3C adaptive PID control

A3C algorithm is a deep reinforcement learning algorithm. It introduces an asynchronous training method on the basis of AC framework. The A3C learning framework consists of a central network (Global Net) and multiple AC

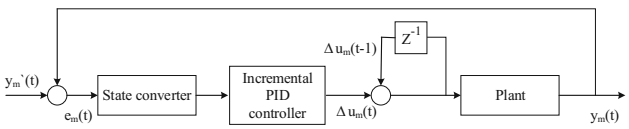

Fig. 1 PID control structure

structures, which are executed and learned in parallel by creating multiple agent in same environmental instances. The central network is responsible for updating and storing AC network parameters. One agent has its own AC structure. Different agent will transfer learning data to central network to update their parameters of AC network. Further the Actor network is responsible for policy learning, while critic network is responsible for estimating value function.

4.1 Structure of A3C-PID controller

The design of A3C adaptive PID controller is to combine the asynchronous learning structure of A3C with the incremental PID controller. Its structure is shown in Fig. 2. The whole process is as follow:

Step 1: For each thread, the initial error $e_m(t)$ enters the state converter to calculate $\Delta e_{\mathrm{m}}(t)\Delta^2 e_{\mathrm{m}}(t)$ and output the state vector $S_m(t) = \left[e_m(t), \Delta e_m(t), \Delta^2 e_m(t)\right]^{\mathrm{T}}$.

Step 2: The Actor (m) maps the state vector $S_m(t)$ to three parameters, K_p K_i and K_d, of PID controller.

Step 3: The updated controller acts on the environment to receive the reward $r_m(t)$.

After n times, Critic (m) receives $S_m(t+n)$ which is the state vector of the system. Then it produces the value function estimation $V(S_{t+n}, W_v')$ and n-step TD error δ_{TD}, which are the important basis for updating parameters. The formula of the reward function is shown as Formula (2)

$$
\begin{aligned}
r_m(t) &= \alpha_1 r_1(t) + \alpha_2 r_2(t) \\
r_1(t) &= \begin{cases} 0, & |e_m(t)| < \varepsilon \\ \varepsilon - e_m(t), & other \end{cases} \\
r_2(t) &= \begin{cases} 0, & |e_m(t)| \le |e_m(t-1)| \\ |e_m(t)| - |e_m(t-1)|, & other \end{cases}
\end{aligned}
\tag{2}
$$

In the next step, the Actor (m) and the Critic (m) send their own parameters W_{am}', W_{vm}' and the generated δ_{TD} into the Global Net to update W_a and W_v with the policy gradient and the descend gradient. Accordingly, the Global Net passes their W_a and W_v to Actor (m) and Critic (m), making them continue to learn new parameters.

4.2 A3C learning with neural networks

Multilayer feed-forward neural network [27, 28], also known as BP neural network, is a back-propagation algorithm for multilayer feed-forward networks. It has strong ability for nonlinear mapping and is suitable for solving problems with complex internal mechanism. Therefore, the method uses two BP neural networks respectively to realize the learning of policy function and value function. The network structure is as follows.

As shown in Fig. 3, the Actor network has three layers:

The first level is the input layer. The input vector $S = \left[e_m(t), \Delta e_m(t), \Delta^2 e_m(t)\right]^{\mathrm{T}}$ represents the state vector. The second layer is the hidden layer. The input of the hidden layer as follows:

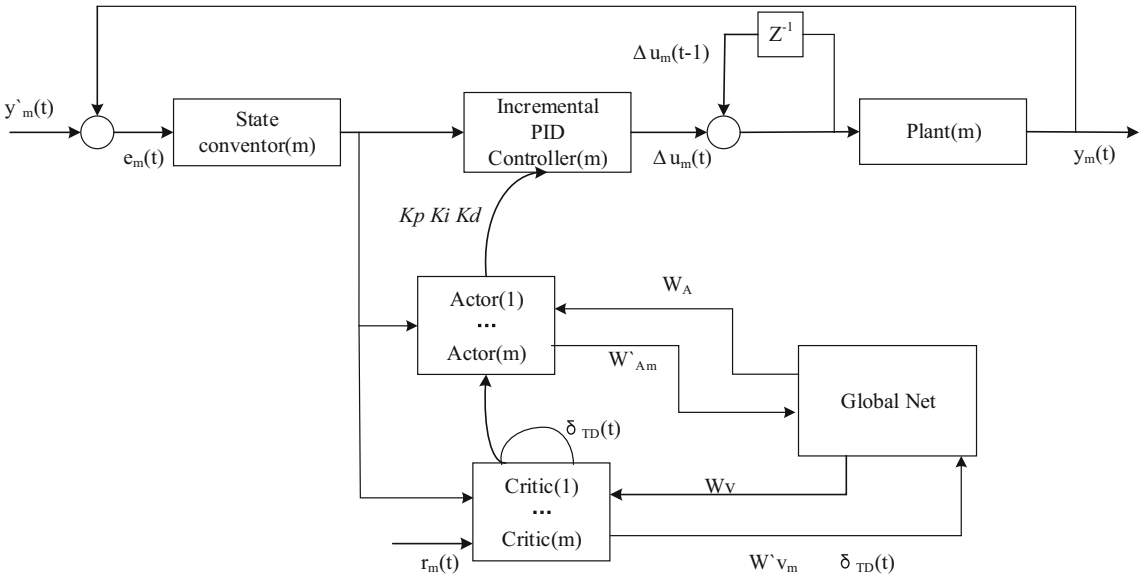

Fig. 2 Adaptive PID control diagram based on A3C learning

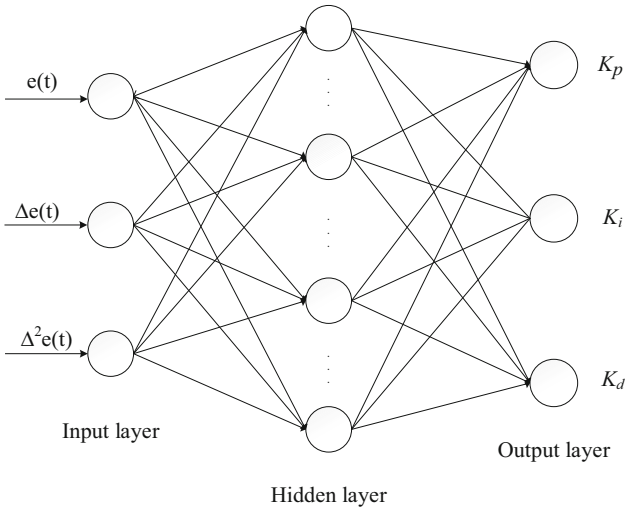

Fig. 3 Actor network structure of actor–critic

$$hi_k(t) = \sum_{i=1}^{n} w_{ik} x_i(t) - b_k \quad k = 1, 2, 3 \ldots 20 \tag{3}$$

where k represents the number of neurons in the hidden layer, w_{ik} is the weights connected the input layer and the hidden layer, b_k is the bias of the k neuron. The output of the hidden layer as follows:

$$ho_k(t) = \min(\max(hi_k(t), 0), 6) \quad k = 1, 2, 3 \ldots 20 \tag{4}$$

The third layer is the output layer. The input of the output layer as follows:

$$yi_o(t) = \sum_{j=1}^{k} w_{ho} ho_j - b_o \quad o = 1, 2, 3 \tag{5}$$

where o represents the number of neurons in the output layer, w_{ho} is the weights connected the hidden layer and the output layer, b_o is the bias of the k neuron.

The output of the output layer as follows:

$$yo_o(t) = \log\left(1 + e^{yi_o(t)}\right) \quad o = 1, 2, 3 \tag{6}$$

Actor network does not output the value of K_p K_i and K_d directly, but output the mean and variance of the three parameters. Finally, the actual value of K_p, K_i and K_d is estimated by the Gauss distribution.

The Critic network structure is similar to the Actor network structure. As shown in Fig. 4, the Critic network also uses BP neural networks with three layers structure. The first two layers are the same as the layers in the Actor network. The output layer of the Critic network has only one node to output the value function $V(S_t, W_v')$ of the state.

In the A3C structure, Actor and Critic networks use n-step TD error method [29, 30] to learn action probability function and value function. In the learning method of this algorithm, the calculation of the n-step TD error δ_{TD} is realized by the difference between the state estimation

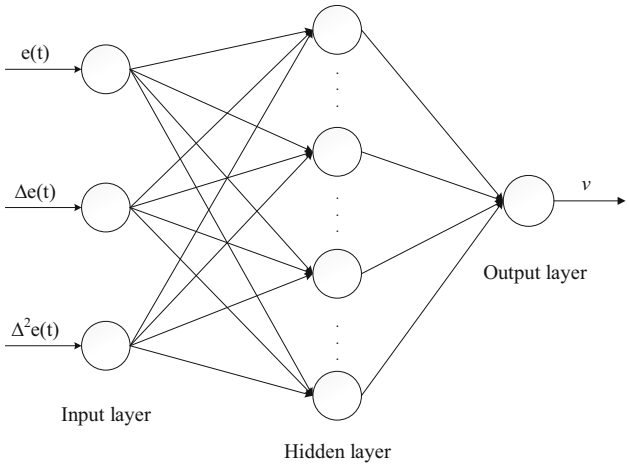

Fig. 4 Critic network structure of actor–critic

value $V(S_t, W_v')$ of the initial state and the estimation value after n-step, as followed:

$$\begin{aligned} \delta_{TD} &= q_t - V\left(S_t, W_v'\right) \\ q_t &= r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n V\left(S_{t+n}, W_v'\right) \end{aligned} \tag{7}$$

The $0 < \gamma < 1$, represents the discount factor, is used to determine the ratio of the delayed returns and the immediate returns. W_v' is the weight of the Critic network. The TD error δ_{TD} reflects the quality of the selected actions in the Actor network. The performance of the system learning is:

$$E(t) = \frac{1}{2} \delta_{TD}^2(t) \tag{8}$$

After calculating the TD error, each AC network in the A3C structure does not update its network weight directly, but updates the AC network parameters of the central network (Global-Net) with its own gradient. The update formula is as follows:

$$W_a = W_a + \alpha_a \left(dW_a + \nabla_{w'a} \log \pi\left(a \middle| s; W_a'\right) \delta_{TD}\right) \tag{9}$$

$$W_v = W_v + \alpha_c \left(dW_v + \partial \delta_{TD}^2 / W_v'\right) \tag{10}$$

where W_a is the weight of Actor network stored by the central network, and W_a' represents the weights of Actor network in AC structure, W_v is the weight of Critic network in the central network, and W_v' represents the Critic network weights for each AC structure. α_a is the learning rate of Actor, and α_c is the learning rate of Critic.

4.3 The network initialization of A3C-PID controller

The initial parameters of the network directly affect the stability of the closed loop control system. However, the

PID controller of the neural network is difficult to obtain the teacher's signal. Therefore it is necessary to determine the network parameters by experience or manual trial. The unsupervised learning characteristics of reinforcement learning enable the controller to obtain the optimal initial parameters of the network through iterative learning. However, the AC-PID controller has a slow convergence speed due to the correlation between the learning samples obtained by the AC algorithm. A3C-PID Controller learns network parameters in multi-threading asynchronously, which can break the relevance of samples and improve the convergence rate. The learning process of A3C-PID network parameter is similar to that described in the 3.1 section, but the difference is that A3C-PID sets the m for the number of computer CPU cores in iterative learning, then the value of m is set to one when online controlling.

4.4 Working process of A3C-PID controller

Based on the architecture of asynchronous learning and the learning mode with taking n-step TD error as the performance, the working process of A3C-PID controller is as follows:

(a) Setting the sampling period ts, the number of threads of the A3C algorithm m, update the period n, and initialize the network parameters of each AC structure through iteration learning on K times;
(b) Calculating errors of system and constructing state vectors as inputs to Actor(m) and Critic(m);
(c) Critic(m) outputs $V(S_t, W'_v)$;
(d) Actor(m) outputs the value of K_p, K_i and K_d. Then the system observes the error $e_m(t+1)$ when next sampling time and calculate the reward $r_m(t)$ according the Eq. (2);
(e) Determining whether to update the parameters of Actor(m) and Critic(m). The Critic outputs the state value $V(S_{t+n}, W'_v)$ then the system updates the parameters of Global Net which is W_a and W_c according to Eqs. (9) and (10), if it has meet update time n. Otherwise, returning step d);
(f) Global Net transmits the new parameters W'_{am} and W'_{cm} to each Actor(m) and Critic(m);
(g) Determining whether the end condition is satisfied, if that exiting the controlling, otherwise updating $S_m(t)$ and returning step (c).

5 Experiments

5.1 Simulation experiment of nonlinear signal

In order to verify the effectiveness and superiority of this algorithm, the nonlinear objects are simulated and analyzed based on PID, CSP-PID, MN-PID, AC-PID and A3C-PID respectively. The discrete model of the object is as follows:

$$y(k+1) = f(y(k), y(k-1), y(k-2), u(k), u(k-1)) \tag{11}$$

where $f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_3^2 + x_2^2}$.

The inputs *rin* is that:

$$rin(t) = \begin{cases} 0.5\sin(\pi k/25) & k < 250 \\ 0.5 & 250 \le k < 460 \\ -0.5 & 460 \le k < 660 \\ 0.5 & 660 \le k < 870 \\ 0.3\sin(\pi k/25) + 0.4\sin(\pi k/32) + 0.3\sin(\pi k/40) & 870 \le k < 1000 \end{cases} \tag{12}$$

The parameters of nonlinear signal simulation are set as follows: the sampling period is 1 s, $m = 4$, $\alpha_a = 0.001$, $\alpha_c = 0.01$, $\varepsilon = 0.001$, $\gamma = 0.9$, $n = 30$, $K = 3000$. The root mean square error (RMSE) and the mean absolute error (MAE) are used to describe the accuracy of the controller. The simulation results are shown in Figs. 5, 6, 7, 8 and 9 and Table 1.

The simulation results show that the A3C-PID controller reaches the minimum for the root mean square error (RMSE) and the mean absolute error (MAE) value. Compared with the other three controllers, the control accuracy of A3C-PID is higher. It not only proves that our design of a new PID controller is reasonable but the controller has the better control performance for the nonlinear system.

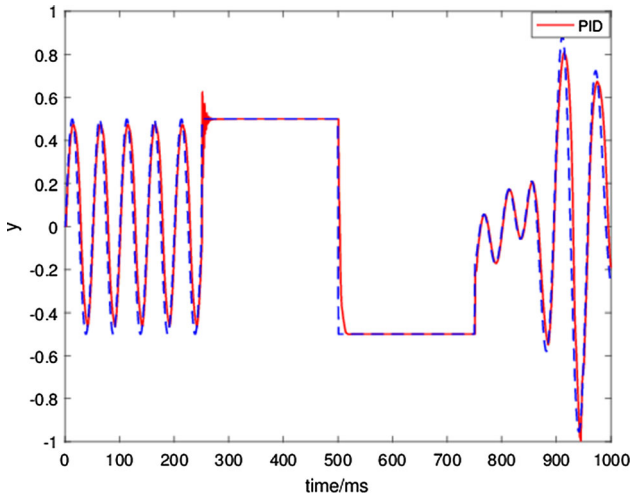

Fig. 5 Position tracking of PID

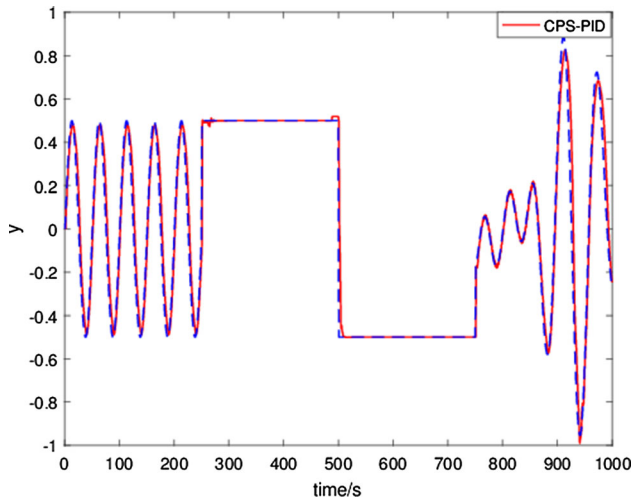

Fig. 6 Position tracking of CPS-PID

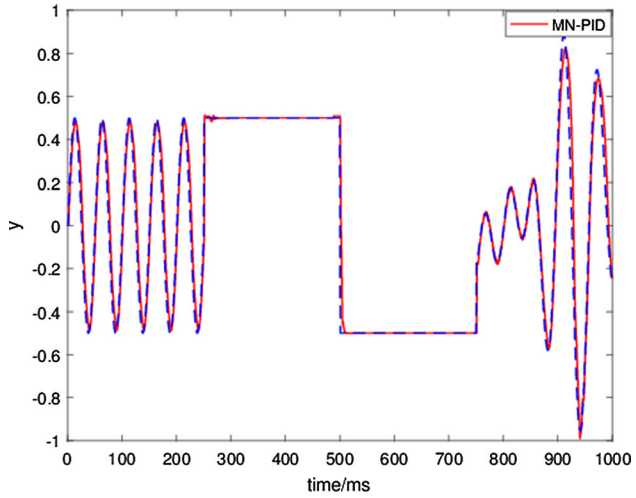

Fig. 7 Position tracking of MN-PID

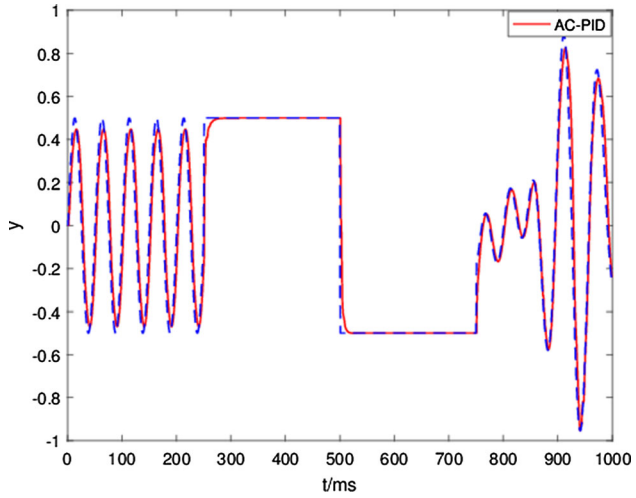

Fig. 8 Position tracking of AC-PID

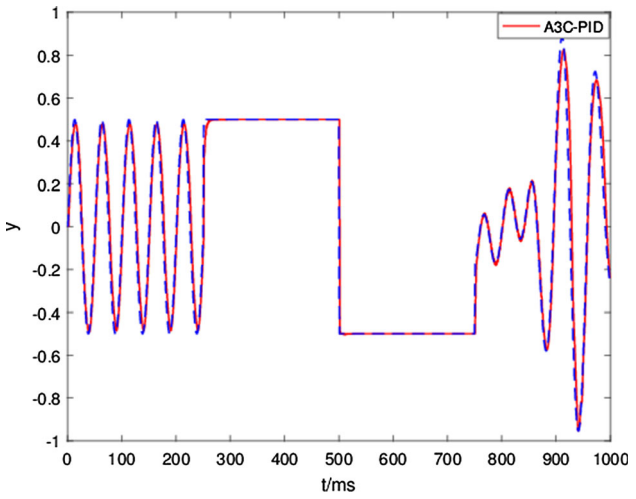

Fig. 9 Position tracking of A3C-PID

Table 1 The comparison of controller performance

Kinds of controllers	RMSE	MAE
PID	0.1547	0. 0705
CPS-PID	0.1201	0.0620
MN-PID	0.1203	0.0628
AC-PID	0.1196	0.0621
A3C-PID	0.0884	0.0326

5.2 Simulation experiment of inverted pendulum

The control of single inverted pendulum is a classic problem in the control study. The control process is to apply the force F to the bottom of the car to make the car stay in the setting position and make the angle between the rod and the vertical line in a deviation range.

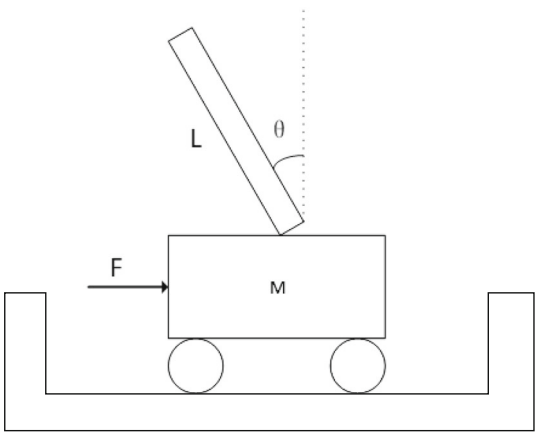

Fig. 10 The structure of single inverted pendulum

Figure 10 shows a single inverted pendulum. As shown in Fig. 10, the quality of the car is M, the quality of the pendulum is m, the position of the car is x, the angle of the pendulum is, the equation of the single inverted pendulum is obtained as Eqs. (13) and (14).

$$\ddot{\theta} = \frac{m(m+M)gl}{(m+M)I + mMl^2}\theta - \frac{ml}{(m+M)l + mMl^2}F \tag{13}$$

$$\ddot{x} = \frac{m^2gl^2}{(m+M)I + mMl^2}\theta - \frac{I+ml}{(m+M)l + mMl^2}F \tag{14}$$

where $I{=}\frac{1}{12}\mathrm{m}L^2, l = \frac{1}{2}L$, F represents the force acting on the car, and take continuous value on [− 10, 10]. Sampling period is 20 ms. Single inverted pendulum has 4 control indexes: pendulum angle, swing speed, position of trolley and speed of car. There initial conditions are as follow:

$$\theta(0) = -10^\circ, \dot{\theta}(0) = 0, x(0) = 0.2, \dot{x}(0) = 0 \tag{15}$$

The final state of expectation is:

$$\theta(0) = 0^\circ, \dot{\theta}(0) = 0, x(0) = 0, \dot{x}(0) = 0 \tag{16}$$

In the simulation, the parameters of the inverted pendulum are as follows:

$$g = 9.8\,\mathrm{m/s^2}, M = 10\,\mathrm{kg}, m = 0.1\,\mathrm{kg}, L = 0.5\,\mathrm{m},$$
$$\mu_c = 0.005, \mu_p = 2 \cdot 10^{-5}$$

The μ_c presents the friction coefficient of the car relative to the guide rail indicates. The μ_p presents the friction coefficient of the rod to the car. The parameters of the A3C-PID controller are set to as follow:

$$\mathrm{m} = 4, \alpha_a = 0.002, \alpha_c = 0.01, \varepsilon = 0.001, \gamma = 0.9, n = 50$$

The results of the simulation are shown in Figs. 11 and 12. Figure 11 is the response of the four controlling indicators of the inverted pendulum in 10 s. From Fig. 11, it

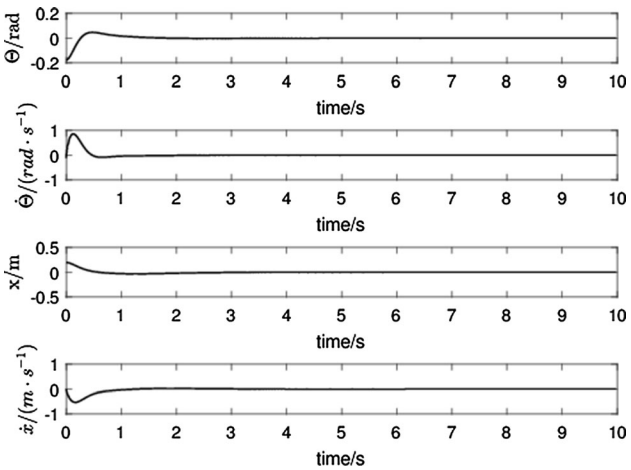

Fig. 11 The response of four index with A3C-PID

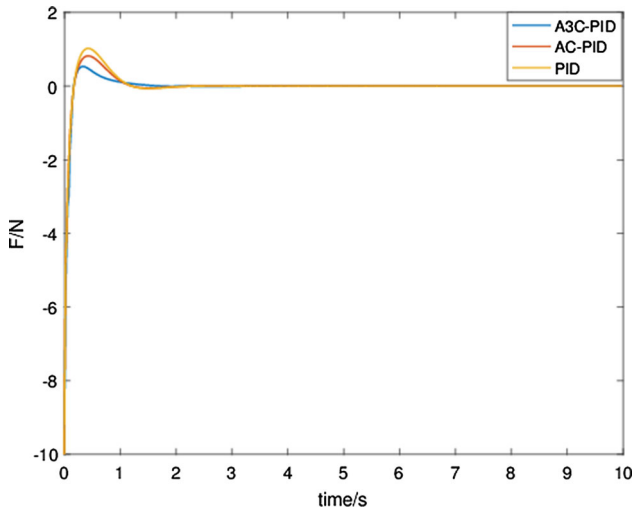

Fig. 12 The output of controller among the PID, AC-PID and A3C-PID

can be seen that under the A3C-PID controlling, the inverted pendulum can quickly reach the stable state of 4 control indicators. Figure 12 is the output of A3C-PID, AC-PID and traditional PID control. It can be seen that A3C-PID controller has better system tracking performance than traditional PID and AC-PID.

5.3 Position control of two phase hybrid stepping motor

5.3.1 Closed loop control structure of stepping motor

The stepper motor is a low speed permanent magnet synchronous motor. It is not used as the input of the pulse sequence. But used in the digital control system by changing the excitation state to realize the angle actuating element. The stepper motor usually adds a photoelectric encoder, a rotating transformer or other measuring feedback elements to achieve high precision positioning control in the closed loop control. The block diagram of the closed-loop servo control system is shown in Fig. 13. From the Fig. 13, the inner loop includes the current loop and the speed loop. The current loop is used to track the current of the two phase hybrid stepping motor, so that the dual phase hybrid stepping motor can output the torque smoothly under the micro step. The speed loop control enables the load electricity to track the setting speed and achieve the effect of speed control. The outer loop is the position loop, which loads the output to track a given position. The position loop controller usually adopts PID control. Therefore, we added the A3C-PID to the position loop to test the validity of the controller.

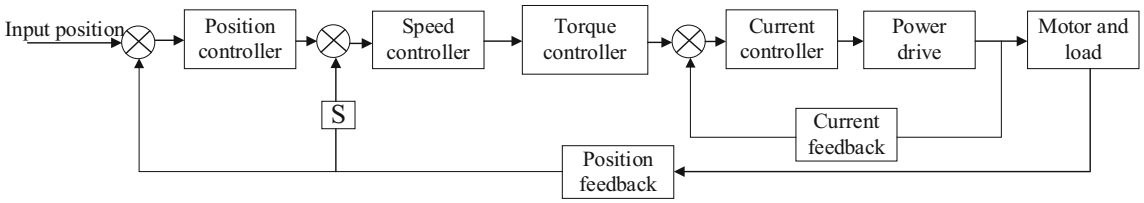

Fig. 13 The closed-loop servo control system of hybrid stepping motor

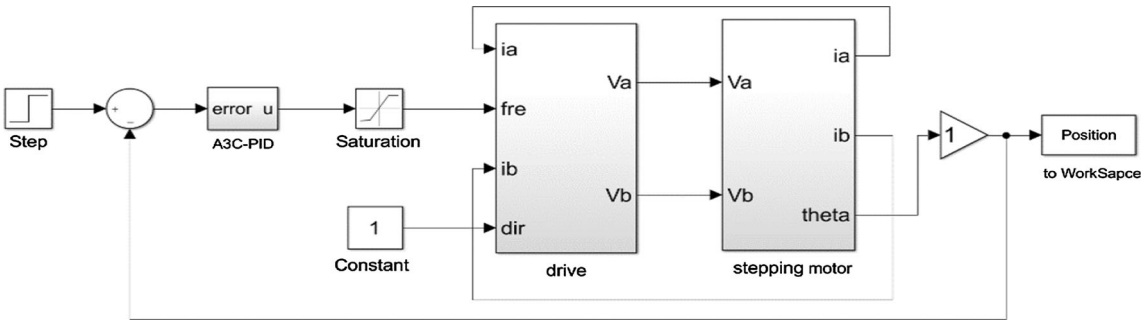

Fig. 14 The simulation of servo system

5.3.2 Modeling and simulation of two-phase hybrid stepping motor

In this paper, a two-phase hybrid stepping motor is used to control in the simulation experiment. Firstly, we need to establish a mathematical model. However, the two-phase hybrid stepping motor is a highly nonlinear mechanical and electrical device, so it is difficult to describe it accurately. Therefore, the mathematical model of a two phase hybrid stepping motor is studied in this paper. It is simplified and assumed to be as follows: The magnetic chain in the phase winding of the permanent magnet varies with the rotor position according to the sinusoidal law. The magnetic hysteresis and the eddy current effect are not considered, only the mean and fundamental components of the air gap magnetic conductance are considered. The mutual inductance between the two phase windings is ignored. On the basis of the above limit, the mathematical model of the two phase hybrid stepping motor can be described by the Eqs. 17–21.

$$u_a = L\frac{di_a}{dt} + Ri_a - k_e\omega\sin(N_r\theta) \tag{17}$$

$$u_b = L\frac{di_b}{dt} + Ri_b - k_e\omega\sin(N_r\theta) \tag{18}$$

$$T_e = -k_e i_a \sin(N_r\theta) + k_e i_b \cos(N_r\theta) \tag{19}$$

$$J\frac{d\omega}{dt} + B\omega + T_L = T_e \tag{20}$$

$$\frac{d\theta}{dt} = \omega \tag{21}$$

In above formulas, u_a and u_b are two-phase voltage and current respectively of A and B. R is winding resistance. L is winding inductance. k_e is torque coefficient. θ and ω are rotation angle and angular velocity of motor respectively. N_r is the number of rotor teeth. T_e is electromagnetic torque of hybrid stepping motor. T_L is Load torque. J and B are the load moment of inertia and the viscous friction coefficient respectively. It can be seen from the mathematical model of a two phase hybrid stepping motor that the two phase hybrid stepping motor is still a highly nonlinear and coupled system under a series of simplified conditions.

The simulation model of two phase hybrid stepping motor servo control system is built by using Simulink in Matlab. The simulation is shown in Fig. 14. The parameters of the motor are as follows: L = 0.5H, N_r = 50, R=8Ω, J = 2 g cm^2, B = 0 N m s/rad, N = 100, T_L = 0, k_e = 17.5 N m/A. The N is the reduction ratio of the harmonic reducer. The parameters of A3C-PID controller are set as follows: m = 4, $\alpha_a = 0.001$, ts = 0.001 s, $\alpha_c = 0.01$, $\varepsilon = 0.001$, $\gamma = 0.9$, n = 30, K = 3000. The results are shown in Figs. 15 and 16 and Table 2.

Dynamic performance of the A3C, BP, and AC adaptive PID controller are shown on Fig. 15. In the time of early simulation (20 cycles), the BP-PID controller has a faster response speed and a shorter rise time (12 ms), but it has a higher overshoot of 2.1705%. On the contrary, both the AC-PID and the A3C-PID controller have smaller overshoot as 0.1571% and 0.1021%. But the adjustment time of AC-PID is long (48 ms), and the rise time is 21 ms. In contrast, A3C-PID controller has better stability and rapidity.

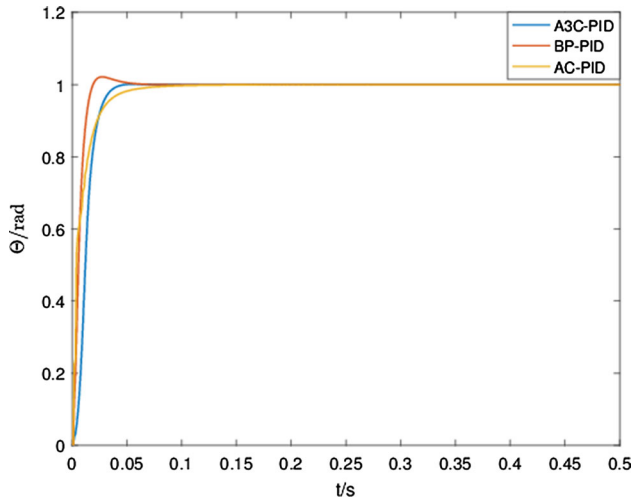

Fig. 15 Position tracking

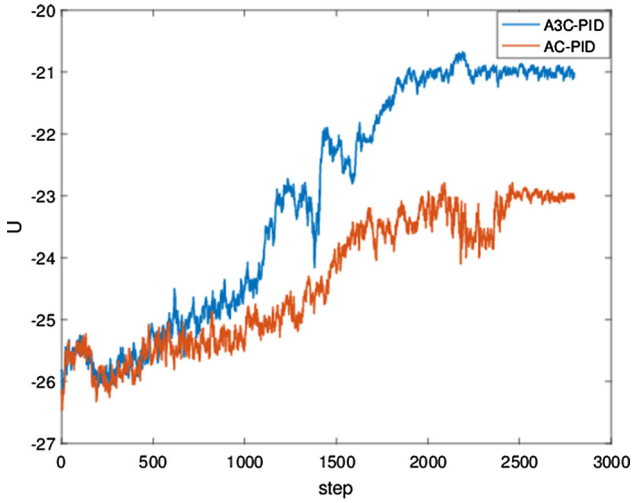

Fig. 17 Reward value curve of reinforcement learning

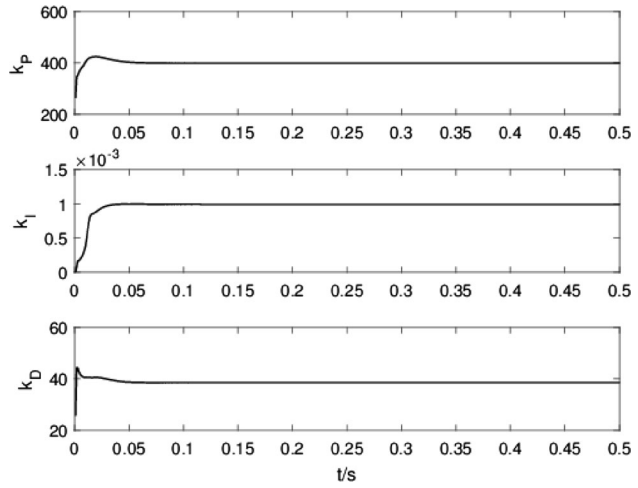

Fig. 16 The result of controller parameter turning

Figure 16 shows the process of adaptive transformation of A3C-PID controller parameters. As be seen from Fig. 16, the A3C-PID controller is able to adjust the PID parameters based on errors in different periods. At the beginning of the simulation, the tracking error of system is large. In order to ensure a fast response speed of the system, K_P is continuously increasing while K_d is reducing. Then the system is in order to prevent from having a high overshoot, which limits the increasing of K_i. With the error decreasing, K_P begins to decrease and the value of K_i is gradually increased to eliminate the cumulative error, but at the same time, a small amount of overshoot is caused. Since the K_d value at this stage has a large influence on the system, it tends to be stable. When the final tracking error comes to zero, K_P, K_i and K_d reach a steady state.

Table 2 The comparison of controller performance

Controller	Overshoot (%)	Rise time (ms)	Steady state error	Adjustment time (ms)
A3C-PID	0.1571	18	0	33
AC-PID	0.1021	21	0	48
BP-PID	2.1705	12	0	32

Simulation results show that the A3C-PID controller has good adaptive capabilities.

The AC-PID and A3C-PID reward value curves are shown in Fig. 17. The goal of reinforcement learning is to learn the best strategy to maximize reward value U. The calculation formula is as seen in Eq. (22)

$$U = \mathrm{E}\left[\sum_{t=0}^{end} \gamma^t R(S_t)\right] \tag{22}$$

We can conclude from the analysis of Fig. 17 that A3C-PID controller has a higher U value after 3000 iterations than AC-PID controller. In addition, the U value of A3C-PID has become stable after about 1800 iterations, while AC-PID converges only after the 2500 iterations. So, A3C-PID has a faster convergence rate than the AC-PID.

6 Conclusions

Machine Learning and Intelligent Algorithms have been well applied in many industrial fields [31–36]. The purpose of this paper has been to present our efforts to improve the convergence and adaptability of the adaptive PID

controller. In this paper, a new PID controller is proposed with A3C algorithm. The controller uses the BP neural network to approach the policy function and the value function. BP neural network have the strong ability in nonlinear mapping, which can enhance the adaptive ability of the controller. The learning speed of A3C PID controller is accelerated with the parallel training of CPU multi-threading. The method of asynchronous multi-thread training reduces the correlation of the training data, and makes the controller more stable and adaptable. Our experiments of nonlinear signal and inverted pendulum demonstrate that A3C-PID controller has higher control accuracy than others PID controller. The experiments about the position control of two phase hybrid stepping motor show that A3C-PID controller has a good performance on overshoot, rise time, steady state error and adjustment time. According these work, the effectiveness and application significance of the new method can be confirmed. Our aim is to make the controller apply to the multi-axis motion control and the actual industrial production.

Acknowledgements This work was supported by National Science and Technology Major Project of China (Grant Number 2017ZX05009-001).

References

1. Adel, T., & Abdelkader, C. (2013). A particle swarm optimization approach for optimum design of PID controller for nonlinear systems. In *International conference on electrical engineering and software applications* (pp. 1–4). IEEE.
2. Savran, A. (2013). A multivariable predictive fuzzy PID control system. *Applied Soft Computing, 13*(5), 2658–2667.
3. Jiang, D., Wang, W., Shi, L., & Song, H. (2018). A compressive sensing-based approach to end-to-end network traffic reconstruction. *IEEE Transactions on Network Science and Engineering, 5*(3), 1–12.
4. Jiang, D., Huo, L., & Li, Y. (2018). Fine-granularity inference and estimations to network traffic for SDN. *PLoS ONE, 13*(5), 1–23.
5. Zhang, X., Bao, H., Du, J., & Wang, C. (2014). Application of a new membership function in nonlinear fuzzy PID controllers with variable gains. *Information and Control, 5,* 1–7.
6. Caocang, Li, & Cuifang, Zhang. (2015). Adaptive neuron PID control based on minimum resource allocation network. *Application Research of Computers, 32*(1), 167–169.
7. Sheng, X., Jiang, T., Wang, J., et al. (2015). Speed-feed-forward PID controller design based on BP neural network. *Journal of Computer Applications, 35*(S2), 134–137.
8. Wang, X. S., Cheng, Y. H., & Wei, S. (2007). A proposal of adaptive PID controller based on reinforcement learning. *Journal of China University of Mining and Technology, 17*(1), 40–44.
9. Huo, L., Jiang, D., & Lv, Z. (2018). Soft frequency reuse-based optimization algorithm for energy efficiency of multi-cell networks. *Computers and Electrical Engineering, 66*(2), 316–331.
10. Akbarimajd, A. (2015). Reinforcement learning adaptive PID controller for an under-actuated robot arm. *International Journal of Integrated Engineering, 7*(2), 20–27.
11. Chen, X. S., & Yang, Y. M. (2011). A novel adaptive PID controller based on actor–critic learning. *Control Theory and Applications, 28*(8), 1187–1192.
12. Bahdanau, D., Brakel, P., Xu, K., Goyal, A., Lowe, R., Pineau, J., et al. (2016). *An actor–critic algorithm for sequence prediction*. arXiv preprint arXiv:1607.07086.
13. Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., et al. (2016). *Sample efficient actor–critic with experience replay*. arXiv preprint arXiv:1611.01224.
14. Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., et al. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning* (pp. 1928–1937).
15. Jiang, D., Huo, L., Lv, Z., Song, H., & Qin, W. (2018). A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. *IEEE Transactions on Intelligent Transportation Systems, 19*(10), 3305–3319.
16. Liu, Q., Zhai, J. W., Zhang, Z. Z., & Zhong, S. (2018). A survey on deep reinforcement learning. *Chinese Journal of Computers, 41*(01), 1–27.
17. Qin, R., Zeng, S., Li, J. J., & Yuan, Y. (2015). Parallel enterprises resource planning based on deep reinforcement learning. *Zidonghua Xuebao/Acta Automatica Sinica, 43*(9), 1588–1596.
18. Jiang, D., Wang, Y., Lv, Z., Qi, S., & Singh, S. (2019). Big data analysis-based network behavior insight of cellular networks for industry 4.0 applications. *IEEE Transactions on Industrial Informatics*. https://doi.org/10.1109/TII.2019.2930226.
19. Tang, H. C., Li, Z. X., Wang, Z. T., et al. (2005). A fuzzy PID control system. *Electric Machines and Control, 2,* 136–138.
20. Sun, J. P., Yan, L., Li, Y., et al. (2006). Design of fuzzy PID controllers based on improved genetic algorithms. *Chinese Journal of Scientific Instrument, S3,* 1991–1992.
21. Zhu, Y. H., Xue, L. Y., & Huang, W. (2011). Design of fuzzy PID controller based on self-organizing adjustment factors. *Journal of System Simulation, 23*(12), 2732–2737.
22. Liao, F. F., & Xiao, J. (2005). Research on self-tuning of PID parameters based on BP neural networks. *Acta Simulata Systematica Sinica, 07,* 1711–1713.
23. Li, G. Y., & Chen, X. L. (2008). Neural network self-learning PID controller based on real-coded genetic algorithm. *Micromotors Servo Technique, 1,* 43–45.
24. Patel, R., & Kumar, V. (2015). Multilayer neuro PID controller based on back propagation algorithm. *Procedia Computer Science, 54,* 207–214.

25. Nie, S. K., Wang, Y. J., Xiao, S., & Liu, Z. (2017). An adaptive chaos particle swarm optimization for tuning parameters of PID controller. *Optimal Control Applications and Methods, 38*(6), 1091–1102.
26. Aziz Khater, A., El-Bardini, M., & El-Rabaie, N. M. (2015). Embedded adaptive fuzzy controller based on reinforcement learning for DC motor with flexible shaft. *Arabian Journal for Science and Engineering, 40*(8), 2389–2406.
27. Liu, Z., Zeng, X., Liu, H., & Chu, R. (2015). A heuristic two-layer reinforcement learning algorithm based on BP neural networks. *Journal of Computer Research and Development, 52*(3), 579–587.
28. Zhu, J., Song, Y., Jiang, D., & Song, H. (2018). A new deep-Q-learning-based transmission scheduling mechanism for the cognitive Internet of things. *IEEE Internet of Things Journal, 5*(4), 2375–2385.
29. Xu, X., Zuo, L., & Huang, Z. (2014). Reinforcement learning algorithms with function approximation: Recent advances and applications. *Information Sciences, 261,* 1–31.
30. Jiang, D., Huo, L., & Song, H. (2018). Rethinking behaviors and activities of base stations in mobile cellular networks based on big data analysis. *IEEE Transactions on Network Science and Engineering, 1*(1), 1–12.
31. Wang, F., Jiang, D., & Qi, S. (2019). An adaptive routing algorithm for integrated information networks. *China Communications, 7*(1), 196–207.
32. Huo, L., & Jiang, D. (2019). Stackelberg game-based energy-efficient resource allocation for 5G cellular networks. *Telecommunication System, 23*(4), 1–11.
33. Jiang, D., Zhang, P., Lv, Z., & Song, H. (2016). Energy-efficient multi-constraint routing algorithm with load balancing for smart city applications. *IEEE Internet of Things Journal, 3*(6), 1437–1447.
34. Jiang, D., Li, W., & Lv, H. (2017). An energy-efficient cooperative multicast routing in multi-hop wireless networks for smart medical applications. *Neurocomputing, 220*(2017), 160–169.
35. Wang, F., Jiang, D., Wen, H., & Song, H. (2019). Adaboost-based security level classification of mobile intelligent terminals. *The Journal of Supercomputing, 75,* 1–19.
36. Sun, M., Jiang, D., Song, H., & Liu, Y. (2017). Statistical resolution limit analysis of two closely-spaced signal sources using Rao test. *IEEE Access, 5,* 22013–22022.

Qifeng Sun was born in 1976, Ph.D. Graduated from China University of Petroleum. Now he is a lecturer of college of computer science and technology, China University of Petroleum, Qingdao, 26658, China. His research interest includes intelligent control, machine learning etc.

Chengze Du was born in 1996, Postgraduate. He is a researcher of college of computer science and technology, China University of Petroleum, Qingdao, 26658, China. His research interest includes deep learning, industrial application, etc.

Youxiang Duan was born in 1964, Ph.D. Graduated from China University of Petroleum. Now he is a professor of college of computer science and technology, China University of Petroleum, Qingdao, 26658, China. His research interest includes service computing, intelligent control, machine learning etc.

Hui Ren was born in 1993, received his B.S. degree in Electrical Engineering from China University of Petroleum, China. His research interest includes intelligent control, Actor-Critic learning, etc.

Hongqiang Li was born in 1975, Senior Engineer, Shengli Drilling Institute of Sinopec Group, Dongying, 257000, China. He has been engaged in field service of instruments while drilling. He is dedicated to the research control accuracy analysis of horizontal wells with large displacement, azimuth gamma imaging while drilling, diameter while drilling, and array imaging while drilling measurement methods.