The Adaptive PID Controlling Algorithm Using Asynchronous Advantage Actor-Critic Learning Method

Qifeng Sun(✉), Hui Ren, Youxiang Duan, and Yanan Yan

University of Petroleum, Qingdao 266580, China
sunqf@upc.edu.cn

Abstract. To address the problems of the slow convergence and inefficiency in the existing adaptive PID controllers, we proposed a new adaptive PID controller using the Asynchronous Advantage Actor-Critic (A3C) algorithm. Firstly, the controller can parallel train the multiple agents of the Actor-Critic (AC) structures 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 indicated that our proposed controller can achieve the fast convergence and strong adaptability compared with conventional controllers.

Keywords: Deep Reinforcement Learning · Asynchronous Advantage Actor-Critic · Adaptive PID control

1 Introduction

The PID controller is a control mechanism of loop feedback, which is widely used in industrial control system [1]. Based on the investigation of conventional PID controller, the adaptive PID controller adjusts parameters online according to the state of the system. Therefore, it has better system adaptability. At present, the majority of the adaptive PID controllers are as follows: The fuzzy PID controller [2], which adopts the ideology of matrix estimations like [3, 4]. It takes the error and the error rate as the input and adjusts the parameters by querying fuzzy matrix table in order to satisfy the requirement of the self-tuning PID parameters. The limitation of this method is that it needs much more prior knowledge. Moreover, this method has a large number of parameters, so that it needed to be optimized [5].

The adaptive PID controller [6, 7] can achieve effective control without identifying the complex nonlinear controlled object using the good approximation ability of the neural network to nonlinear structure. 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 because it requires less prior knowledge [9]. The adaptive PID controller based on reinforcement learning [10] solves the problem that the teacher's signal is difficult to obtain by unsupervised learning process. What is

H. Song et al. (Eds.): SIMUtools 2019, LNICST 295, pp. 498–507, 2019.
https://doi.org/10.1007/978-3-030-32216-8_48

more, the optimization of the control parameters is simple. The Actor-Critic (AC) adaptive PID [11, 12] is the most widely used in the 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]. This algorithm adopts multi strategies such as [15] 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 domain [17, 18].

Under the several problems in view of discovery, the contributions of this paper are as follows:

1. To address the problem of data relevance and the teacher signal, we draw lessons from the A3C algorithm that enhancing the learning rate with an aim to train agent in the parallel threads.
2. In order to improve the precision and adaptive ability of the controller, we use two BP neural network to approach policy function and value function separately.
3. Extensive simulation results and discussions demonstrated that our proposed adaptive PID controlling algorithm outperforms the conventional PID controlling algorithms.

In Sect. 2, we present the related work about the adaptive controller. In Sect. 3, we present our design of A3C-PID controller. Section 4 describes the result that we apply A3C-PID to the position control of stepper motor. Section 5 discusses the results achieved so far and presents some directions for further work.

2 Related Work

The conventional PID controlling algorithms can be roughly classified into two categories including the neural network PID controllers and reinforcement learning PID controllers.

2.1 Related Works with Adaptive Controller Based on Neural Network

The paper [19] 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 can be determined by artificial test, it is not enough to ensure the reliability of the manual result. Based on this, the author of [20] adopt the genetic algorithm to obtain the optimal initial parameters of the network. However, the genetic algorithm is easily to fall into local optimum. In order to solve the problem, author of [21] 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.

2.2 Related Works with Reinforcement Learning Adaptive Controller

The authors of [10] proposed a PID controller that combining the ASN reinforcement learning network with fuzzy math. Despite this method does not need too much accurate training samples compared the neural network PID, its structure is too complex to guarantee the real-time performance for itself. In view of this point, literature [22] designed an adaptive PID controller based on Actor-Critic algorithm. The controller has simple structure that formed just one RBF network. However, it convergences slowly owing to the learning sample of Actor-Critic algorithm is relevance.

3 A3C Adaptive PID Control

3.1 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 as shown in Fig. 1. The whole process is as follow: for each thread, the initial error $e_{\mathrm{m}}(t) = y'(t) - y(t)$ enters the state converter to calculate $\Delta e_m(t) = e_{\mathrm{m}}(t) - e_m(t-1)$ $\Delta^2 e_{\mathrm{m}}(t) = e_m(t) - 2 * e_m(t-1) + e_m(t-2)$ and output the state vector $S_m(t) = [e_m(t), \Delta e_m(t),\ \Delta^2 e_m(t)]^{\mathrm{T}}$. Then the Actor (m) maps the state vector $S_m(t)$ to three parameters, $\mathrm{K_p}$ $\mathrm{K_i}$ and $\mathrm{K_d}$, of PID controller. 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. Finally it produces the value function estimation $V(S_{t+n}, W'_v)$ and n-step TD error δ_{TD}, which are viewed as the important basis for updating parameters. The formula of the reward function is shown as Formula (1)

$$r_{\mathrm{m}}(t) = \alpha_1 r_1(t) + \alpha_2 r_2(t) \tag{1}$$

$$r_1(t) = \begin{cases} 0, & |e_m(t)| < \varepsilon \\ \varepsilon - e_m(t), & other \end{cases} \quad r_2(t) = \begin{cases} 0, & |e_m(t)| \le |e_m(t-1)| \\ |e_m(t)| - |e_m(t-1)|, & other \end{cases}$$

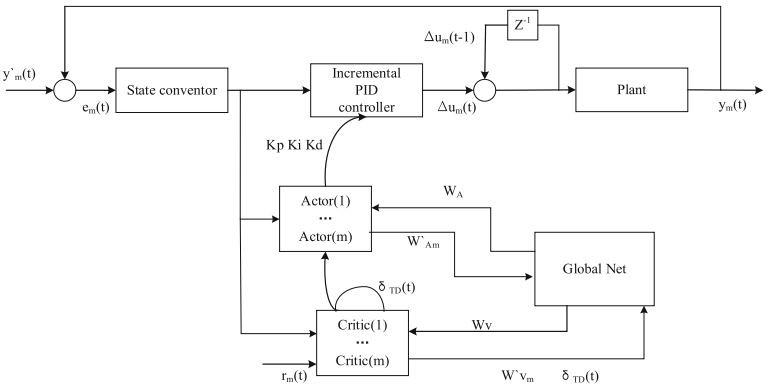

Fig. 1. Adaptive PID control diagram based on A3C learning

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.

3.2 A3C Learning with Neural Networks

Multilayer feed-forward neural network [23], 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:

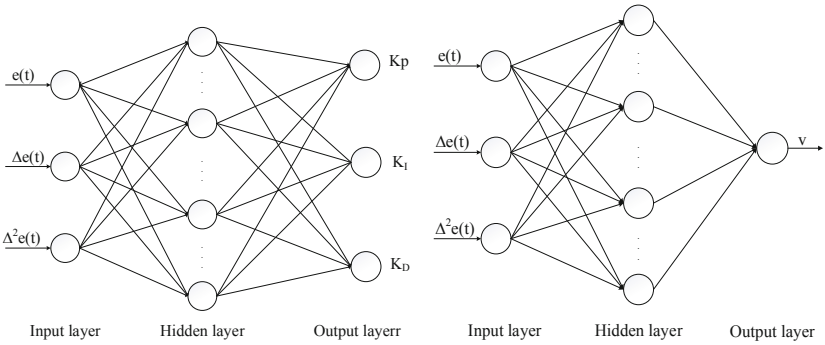

Fig. 2. Network structure of Actor-Critic

As shown in Fig. 2, the Actor network has 3 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:

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

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:

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

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

$$yi_o(t) = \sum_{j=1}^{k} w_{ho} ho_j - b_o \quad o = 1, 2, 3 \tag{4}$$

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:

$$yo_o(t) = \log\left(1 + e^{yi_o(t)}\right) \quad o = 1, 2, 3 \tag{5}$$

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. 3, 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. Obviously, the difference lies in the output layer of the Critic network which 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 [24] 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 value $V(S_t, W'_v)$ of the initial state and the estimation value after n-step, as followed:

$$\delta_{TD} = q_t - V\left(S_t, W'_v\right) \tag{6}$$

$$q_t = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n V\left(S_{t+n}, W'_v\right)$$

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:

$$E(t) = \frac{1}{2}\delta_{TD}^2(t) \tag{7}$$

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

$$W_a = W_a + \alpha_a\left(dW_a + \nabla_{w'a} \log \pi\left(a \middle| s; W'_a\right)\delta_{TD}\right) \tag{8}$$

$$W_v = W_v + \alpha_c\left(dW_v + \partial\delta_{TD}^2 / W'_v\right) \tag{9}$$

Where W_a, which is stored by the central network, is the weight of Actor network, W'_a represents the weights of Actor network in AC structure, W_v is the weight of Critic

network in the central network, 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 Position Control of Two Phase Hybrid Stepping Motor

4.1 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 that it is difficult to accurately describe it. 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 while 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. 10–14.

$$u_a = L\frac{di_a}{dt} + Ri_a - k_e\omega \sin(N_r\theta) \tag{10}$$

$$u_{\mathrm{b}} = L\frac{di_b}{dt} + Ri_b - k_e\omega \sin(N_r\theta) \tag{11}$$

$$T_e = -k_e i_a \sin(N_r\theta) + k_e i_e \cos(N_r\theta) \tag{12}$$

$$J\frac{d\omega}{dt} + B\omega + T_L = T_e \tag{13}$$

$$\frac{d\theta}{dt} = \omega \tag{14}$$

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 Simulink in Matlab. The simulation is shown in Fig. 3.

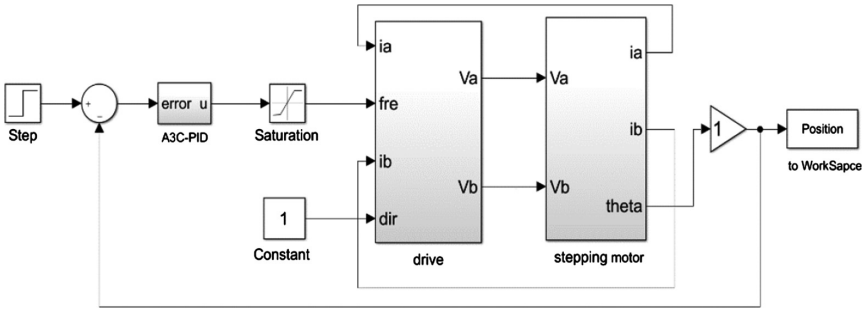

Fig. 3. The simulation of servo system

The parameters of the motor are as follows: $L = 0.5\,\mathrm{H}$, $N_r = 50$, $R = 8\,\Omega$, $J = 2\,\mathrm{g.cm^2}$, $B = 0\,\mathrm{N\,m\,s/rad}$, $N = 100$, $T_L = 0$, $k_e = 17.5\,\mathrm{N\,m/A}$. The N is the reduction ratio of the harmonic reducer. The A3C-PID controller parameters are set as follows: $m = 4$, $\alpha_a = 0.001$, $ts = 0.001\,\mathrm{s}$, $\alpha_c = 0.01$, $\varepsilon = 0.001$, $\gamma = 0.9$, $n = 30$, $K = 3000$. The simulation results are shown in Figs. 4, 5 and Table 1.

Table 1. 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

Dynamic performance of the A3C, BP, and AC adaptive PID controller are shown on Fig. 4. 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.

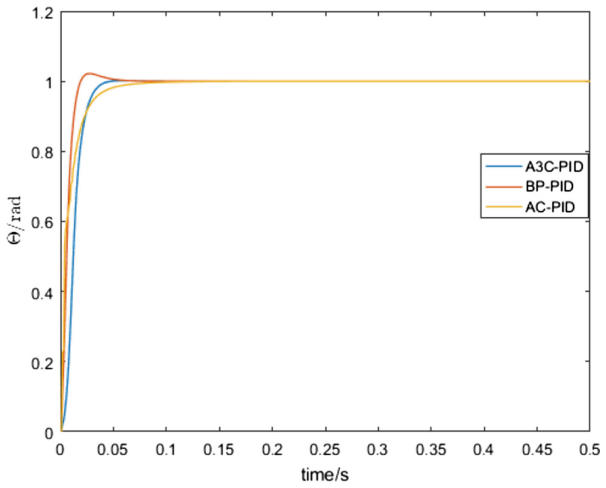

Fig. 4. Position tracking

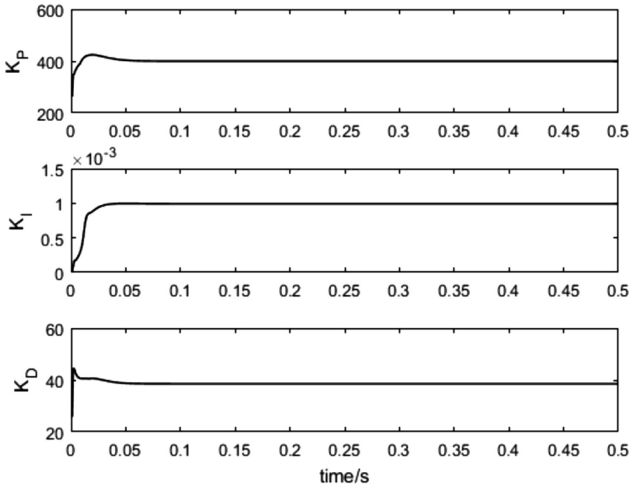

Fig. 5. The result of controller parameter turning

Figure 5 shows the process of adaptive transformation of A3C-PID controller parameters. As be seen from Fig. 5, 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. Meanwhile, the value of K_i is gradually increased to

eliminate the cumulative error. However, a small amount of overshoot is caused. K_d tends to be stable at this stage because it has a large influence on the system. When the final tracking error comes to 0, K_P, K_i and K_d reach a steady state. Simulation results show that the A3C-PID controller has an excellent adaptive capability.

5 Conclusions

In this paper, a new PID controller is proposed with asynchronous advantage actor-critic 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 in CPU multithreading. The method of asynchronous multi-thread training reduces the correlation of the training data and makes the controller more stable. In the simulation of nonlinear signal and inverted pendulum, the control accuracy of A3C-PID controller is higher than others PID controllers.

Current work includes that we use the controller to control the position of two phase hybrid stepping motor and analyze the performance of controller such as: overshoot, rise time, steady state error and adjustment time. According to these work, it confirmed the effectiveness and application significance of the algorithm. Finally, our aim is to make the controller apply to the multi-axis motion control and the actual industrial production.

References

1. Adel, T., Abdelkader, C.: 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 (2013)
2. Savran, A.: A multivariable predictive fuzzy PID control system. Appl. Soft Comput. **13**(5), 2658–2667 (2013)
3. Jiang, D., Wang, W., Shi, L., Song, H.: A compressive sensing-based approach to end-to-end network traffic reconstruction. IEEE Trans. Netw. Sci. Eng. (2018). https://doi.org/10.1109/tnse.2018.2877597
4. Jiang, D., Huo, L., Li, Y.: Fine-granularity inference and estimations to network traffic for SDN. PLoS One **13**(5), 1–23 (2018)
5. Zhang, X., Bao, H., Du, J., et al.: Application of a new membership function in nonlinear fuzzy PID controllers with variable gains. Inf. Control **2014**(5), 1–7 (2014)
6. Cao-Cang, L.I., Zhang, C.F.: Adaptive neuron PID control based on minimum resource allocation network. Appl. Res. Comput. **32**(1), 167–169 (2015)
7. Patel, R., Kumar, V.: Multilayer neuro PID controller based on back propagation algorithm. Procedia Comput. Sci. **54**, 207–214 (2015)
8. Wang, X.S., Cheng, Y.H., Wei, S.: A proposal of adaptive PID controller based on reinforcement learning. J. China Univ. Min. Technol. **17**(1), 40–44 (2007)

9. Su, Y., Chen, L., Tang, C., et al.: Evolutionary multi-objective optimization of PID parameters for output voltage regulation in ECPT system based on NSGA-II. Trans. China Electrotech. Soc. **31**(19), 106–114 (2016)
10. Akbarimajd, A.: Reinforcement learning adaptive PID controller for an under-actuated robot arm. Int. J. Integr. Eng. **7**(2), 20–27 (2015)
11. Chen, X.S., Yang, Y.M.: A novel adaptive PID controller based on actor-critic learning. Control Theory Appl. **28**(8), 1187–1192 (2011)
12. Bahdanau, D., Brakel, P., Xu, K., et al.: An actor-critic algorithm for sequence prediction. arXiv preprint arXiv:1607.07086 (2016)
13. Wang, Z., Bapst, V., Heess, N., et al.: Sample efficient actor-critic with experience replay. arXiv preprint arXiv:1611.01224 (2016)
14. Mnih, V., Badia, A.P., Mirza, M., et al.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning, pp. 1928–1937 (2016)
15. Jiang, D., Huo, L., Lv, Z., et al.: A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. IEEE Trans. Intell. Transp. Syst. **19**, 3305–3319 (2018)
16. Liu, Q., et al.: A survey on deep reinforcement learning. Chin. J. Comput. **41**(01), 1–27 (2018)
17. Qin, R., Zeng, S., Li, J.J., et al.: Parallel enterprises resource planning based on deep reinforcement learning. Zidonghua Xuebao/Acta Autom. Sin. **43**(9), 1588–1596 (2015)
18. Liao, F.F., Xiao, J.: Research on self-tuning of PID parameters based on BP neural networks. Acta Simulata Syst. Sin. **07**, 1711–1713 (2005)
19. Guo-Yong, L.I., Chen, X.L.: Neural network self-learning PID controller based on real-coded genetic algorithm. Micromotors Servo Tech. **1**, 43–45 (2008)
20. Sheng, X., Jiang, T., Wang, J., et al.: Speed-feed-forward PID controller design based on BP neural network. J. Comput. Appl. **35**(S2), 134–137 (2015)
21. Ma, L., Cai, Z.X.: Fuzzy adaptive controller based on reinforcement learning. Cent. South Univ. Technol. **29**(2), 172–176 (1998)
22. Liu, Z., Zeng, X., Liu, H., et al.: A heuristic two-layer reinforcement learning algorithm based on BP neural networks. J. Comput. Res. Dev. **52**(3), 579–587 (2015)
23. Xu, X., Zuo, L., Huang, Z.: Reinforcement learning algorithms with function approximation: recent advances and applications. Inf. Sci. **261**, 1–31 (2014)
24. Yang, S.Y., Xu, L.P., Wang, P.J.: Study on PID control of a single inverted pendulum system. Control Eng. China **S1**, 1711–1713 (2007)