J China Univ Mining & Technol 2007, 17(1): 0040－0044

A Proposal of Adaptive PID Controller Based on Reinforcement Learning

WANG Xue-song, CHENG Yu-hu, SUN Wei

School of Information and Electrical Engineering, China University of Mining & Technology, Xuzhou, Jiangsu 221008, *China*

Abstract: Aimed at the lack of self-tuning PID parameters in conventional PID controllers, the structure and learning algorithm of an adaptive PID controller based on reinforcement learning were proposed. Actor-Critic learning was used to tune PID parameters in an adaptive way by taking advantage of the model-free and on-line learning properties of reinforcement learning effectively. In order to reduce the demand of storage space and to improve the learning efficiency, a single RBF neural network was used to approximate the policy function of Actor and the value function of Critic simultaneously. The inputs of RBF network are the system error, as well as the first and the second-order differences of error. The Actor can realize the mapping from the system state to PID parameters, while the Critic evaluates the outputs of the Actor and produces TD error. Based on TD error performance index and gradient descent method, the updating rules of RBF kernel function and network weights were given. Simulation results show that the proposed controller is efficient for complex nonlinear systems and it is perfectly adaptable and strongly robust, which is better than that of a conventional PID controller.

Key words: reinforcement learning; Actor-Critic learning; adaptive PID control; RBF network

CLC number: TP 18

1 Introduction

Among various controller design methods, PID (Proportional, Integral and Derivative) control is a kind of practical control technique because of its versatility, typical structure, high reliability and ease of operation. Moreover, PID control has a prominent advantage in that we can obtain a definite control performance by choosing appropriate PID parameters based on experience when the mathematical model of the controlled plant is not exact. Therefore, PID controllers are widely used in many engineering fields[1]. But PID parameters cannot be tuned on-line to adapt to the changes of system parameters once they are settled. It is difficult to obtain satisfactory control effects when we apply PID control to time-varying or time-lag systems.

In order to solve this problem, the adaptive PID controller design has received wide attention. The common design idea of adaptive PID controller is to adjust PID parameters according to varying system states to obtain better control effects. There are many kinds of adaptive PID control methods that have been proposed recently, such as fuzzy adaptive PID control[2], adaptive PID control based on neural network[3] and adaptive PID control based on evolution algorithm[4]. Fuzzy adaptive PID control design needs much prior knowledge and has a problem of parameter optimization. Because adaptive PID control based on neural network generally adopts supervised learning to optimize network parameters, its applications are limited due to some factors, such as teaching signals which are difficult to obtain. Although adaptive PID control based on evolution algorithm needs little prior knowledge, it is unable to realize real-time and on-line optimization because of its slow computing speed.

Unlike supervised learning of neural network, reinforcement learning adopts a ‘trial and error’ mechanism existing in human and animal learning. It emphasizes that an agent can learn to achieve a goal from interactions with the environment. At first, a reinforcement learning agent exploits the environment actively and then evaluates the exploitation re-

Received 06 September 2006; accepted 10 October 2006

Projects 0601033B supported by the Science Foundation for Post-doctoral Scientists of Jiangsu Province, 0C4466 and 0C060093 by the Scientific and Technological Foundation for Youth of China University of Mining & Technology

Corresponding author. Tel: +86-516-83995312; E-mail address: wangxuesongcumt@163.com

sults, based on which controller is modified. It can realize unsupervised on-line learning without a system model[5–6]. Actor-Critic learning proposed by Barto et al is one of the most important reinforcement learning methods, which provides a working method of finding the optimal action and the expected value simultaneously[7]. Actor-Critic learning is widely used in artificial intelligence, robot planning and control, optimization and scheduling fields. Based on this analysis, a new adaptive PID controller based on reinforcement learning is proposed in this paper. PID parameters are tuned on-line and adaptively by using the Actor-Critic learning method, which can solve the deficiency of realizing effective control for complex and time-varying systems by conventional PID controllers.

2 Adaptive PID Controller Based on Reinforcement Learning

2.1 Controller structure

The structure of an adaptive PID controller based on Actor-Critic learning is sketched in Fig. 1 based on the design idea of the incremental PID controller described by Eq. (1).

$$\begin{aligned} u(t) = u(t-1) + \Delta u(t) = u(t-1) + \boldsymbol{K}(t)\boldsymbol{x}(t) = \\ u(t-1) + k_{\mathrm{I}}(t)x_1(t) + k_{\mathrm{P}}(t)x_2(t) + k_{\mathrm{D}}(t)x_3(t) = \\ u(t-1) + k_{\mathrm{I}}(t)e(t) + k_{\mathrm{P}}(t)\Delta e(t) + k_{\mathrm{D}}(t)\Delta^2 e(t) \end{aligned} \tag{1}$$

where $\boldsymbol{x}(t) = \left[x_1(t), x_2(t), x_3(t)\right]^{\mathrm{T}} = \left[e(t), \Delta e(t), \Delta^2 e(t)\right]^{\mathrm{T}}$; $e(t) = y_{\mathrm{d}}(t) - y(t)$, $\Delta e(t) = e(t) - e(t-1)$, and $\Delta^2 e(t) = e(t) - 2e(t-1) + e(t-2)$ represent the system output error, the first-order difference of error and the second-order difference of error respectively; $\boldsymbol{K}(t) = \left[k_{\mathrm{I}}(t), k_{\mathrm{P}}(t), k_{\mathrm{D}}(t)\right]$ is a vector of PID parameters.

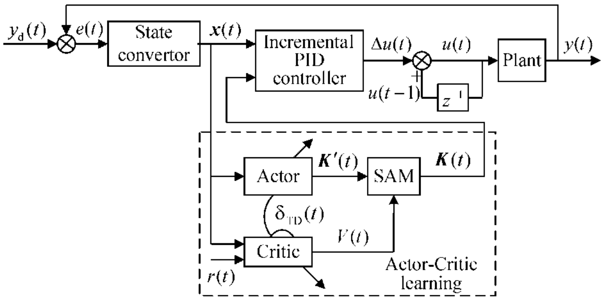


Fig. 1 Self-adaptive PID controller based on reinforcement learning

In Fig. 1, thick lines denote vectors, while thin lines denote scalars. $y_{\mathrm{d}}(t)$ and $y(t)$ are the given and the actual system outputs respectively. The error $e(t)$ is converted into a system state vector $\boldsymbol{x}(t)$ through a state convertor, which is needed by the Actor-Critic learning part shown inside the broken curve in Fig. 1. There are three essential components of an Actor-Critic learning architecture, including an Actor, a Critic and a stochastic action modifier (SAM). The Actor is used to estimate a policy function and realizes the mapping from the current system state vector to the recommended PID parameters $\boldsymbol{K}'(t) = [k'_{\mathrm{I}}(t), k'_{\mathrm{P}}(t), k'_{\mathrm{D}}(t)]$ that will not participate in the design of the PID controller directly. The SAM is used to generate stochastically the actual PID parameters $\boldsymbol{K}(t)$ according to the recommended PID parameters $\boldsymbol{K}'(t)$ suggested by the Actor and the estimated signal $V(t)$ from the Critic. The Critic receives a system state vector and an external reinforcement signal (i.e., immediate reward) $r(t)$ from the environment and produces a TD error (i.e., internal reinforcement signal) $\delta_{\mathrm{TD}}(t)$ and an estimated value function $V(t)$. $\delta_{\mathrm{TD}}(t)$ is provided for the Actor and the Critic directly and is viewed as an important basis for updating parameters of the Actor and the Critic. $V(t)$ is send to the SAM and is used to modify the output of the Actor.

The effect of the system error and the change rate of error on control performance must be considered simultaneously during the design of the external reinforcement signal $r(t)$. Therefore, $r(t)$ is defined as

$$r(t) = \alpha r_{\mathrm{e}}(t) + \beta r_{\mathrm{ec}}(t) \tag{2}$$

where α and β are weighted coefficients, $r_{\mathrm{e}}(t) = \begin{cases} 0 & |e(t)| \le \varepsilon \\ -0.5 & \text{otherwise} \end{cases}$, $r_{\mathrm{ec}}(t) = \begin{cases} 0 & |e(t)| \le |e(t-1)| \\ -0.5 & \text{otherwise} \end{cases}$ and ε is a tolerant error band.

2.2 Actor-Critic learning based on RBF network

The RBF network is a kind of multi-layer feedforward neural network. It has the characteristics of a simple structure, strong global approximation ability and a quick and easy training algorithm[8]. On the other hand, the inputs of the Actor and the Critic are both the same state vector derived from the environment and their small difference is the difference in their outputs. Therefore, there is only one RBF network, as shown in Fig. 2. It is used to implement the policy function learning of the Actor and the value function learning of the Critic simultaneously. That is, the Actor and the Critic can share the input and the hidden layers of the RBF network. This working manner can decrease the demand for storage space and avoid the repeated computation for the outputs of the hidden units in order to improve the learning efficiency. The definite meaning of each layer is described as follows.

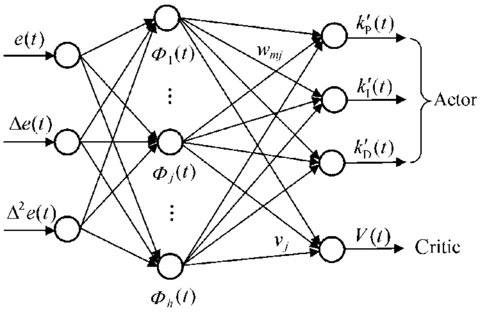


Fig. 2 Actor-Critic learning based on RBF network

Layer 1: input layer. Each unit in this layer denotes a system state variable x_i where i is an input variable index. Input vector $\boldsymbol{x}(t) \in R^3$ is transmitted to the next layer directly.

Layer 2: hidden layer. The kernel function of the hidden unit of RBF network is adopted as a Gaussian function. The output of the *j*th hidden unit is

$$\Phi_j(t) = \exp\left(-\frac{\left\|\boldsymbol{x}(t)-\boldsymbol{\mu}_j(t)\right\|^2}{2\sigma_j^2(t)}\right), \quad j = 1, 2, \cdots, h \qquad (3)$$

where $\boldsymbol{\mu}_j = \left[\mu_{1j}, \mu_{2j}, \mu_{3j}\right]^{\mathrm{T}}$ and σ_j are the center vector and the width scalar of the *j*th hidden unit respectively, h the number of hidden units.

Layer 3: output layer. The layer is made up of an Actor part and a Critic part. The *m*th output of the Actor part, $K'_m(t)$ and the value function of the Critic part, $V(t)$ are calculated as

$$K'_m(t) = \sum_{j=1}^{h} w_{mj}(t)\Phi_j(t), \quad m = 1,\ 2,\ 3 \qquad (4)$$

$$V(t) = \sum_{j=1}^{h} v_j(t)\Phi_j(t) \qquad (5)$$

where w_{mj} denotes the weight between the *j*th hidden unit and the *m*th Actor unit, and v_j denotes the weight between the *j*th hidden unit and the single Critic unit.

In order to solve the dilemma of ‘exploration’ and ‘exploitation’, the output of the Actor part does not pass to the PID controller directly. A Gaussian noise term n_k is added to the recommended PID parameters $\boldsymbol{K}'(t)$ coming from the Actor[9], consequently the actual PID parameters $\boldsymbol{K}(t)$ are modified as Eq. (6). The magnitude of the Gaussian noise depends on $V(t)$. If $V(t)$ is large, n_k is small, and vice versa.

$$\boldsymbol{K}(t) = \boldsymbol{K}'(t) + n_k(0, \sigma_V(t)) \qquad (6)$$

where $\sigma_V(t) = \dfrac{1}{1+\exp\left(2V(t)\right)}$.

The feature of Actor-Critic learning is that the Actor learns the policy function and the Critic learns the value function using the TD method simultaneously[10]. The TD error $\delta_{\mathrm{TD}}(t)$ is calculated by the temporal difference of the value function between successive states in the state transition.

$$\delta_{\mathrm{TD}}(t) = r(t) + \gamma V(t+1) - V(t) \qquad (7)$$

where $r(t)$ is the external reinforcement reward signal, $0 < \gamma < 1$ denotes the discount factor that is used to determine the proportion of the delay to the future rewards. The TD error indicates, in fact, the goodness of the actual action, therefore, the performance index function of system learning can be defined as follows.

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

Based on the TD error performance index, the weights of Actor and Critic are updated according to the following equations through a gradient descent method and a chain rule.

$$w_{mj}(t+1) = w_{mj}(t) + \alpha_{\mathrm{A}}\delta_{\mathrm{TD}}(t)\frac{K_m(t)-K'_m(t)}{\sigma_V(t)}\Phi_j(t) \qquad (9)$$

$$v_j(t+1) = v_j(t) + \alpha_{\mathrm{C}}\delta_{\mathrm{TD}}(t)\Phi_j(t) \qquad (10)$$

where α_{A} and α_{C} are learning rates of Actor and Critic respectively.

Because the Actor and the Critic share the input and the hidden layers of RBF network, the centers and the widths of hidden units need to be updated only once according to the following rules.

$$\mu_{ij}(t+1) = \mu_{ij}(t) + \eta_\mu \delta_{\mathrm{TD}}(t) v_j(t)\Phi_j(t)\frac{x_i(t)-\mu_{ij}(t)}{\sigma_j^2(t)} \qquad (11)$$

$$\sigma_j(t+1) = \sigma_j(t) + \eta_\sigma \delta_{\mathrm{TD}}(t) v_j(t)\Phi_j(t)\frac{\left\|x(t)-\mu_j(t)\right\|^2}{\sigma_j^3(t)} \qquad (12)$$

where η_μ and η_σ are learning rates of center and width respectively.

3 Controller Design Steps

The whole design steps of the proposed adaptive PID controller can be described as follows.

Step 1. Initializing parameters of Actor-Critic learning controller, including $w_{mj}(0)$, $v_j(0)$, $\mu_{ij}(0)$, $\sigma_j(0)$, η_μ, η_σ, α_{C}, α_{A}, γ, ε, α and β.

Step 2. Detecting the actual system output $y(t)$, calculating the system error $e(t)$, constituting system state variables $e(t)$, $\Delta e(t)$ and $\Delta^2 e(t)$.

Step 3. Receiving an immediate reward $r(t)$ from Eq. (2).

Step 4. Calculating the Actor output $\boldsymbol{K}'(t)$ and the Critic value function $V(t)$ from Eq. (4) and Eq. (5) at time t respectively.

Step 5. Calculating the actual PID parameters $\boldsymbol{K}(t)$ from Eq. (6), and consequently calculating the control output of PID controller $u(t)$ from Eq. (1).

Step 6. Applying $u(t)$ to the controlled plant and observing the system output $y(t+1)$ and the immediate reward $r(t+1)$ at the next sampling time.

Step 7. Calculating the Actor output $\boldsymbol{K}'(t+1)$ and the Critic value function $V(t+1)$ from Eq. (4) and Eq. (5) at time $(t+1)$ respectively.

Step 8. Calculating the TD error $\delta_{\text{TD}}(t)$ from Eq. (7).

Step 9. Updating the weights of the Actor and the Critic from Eq. (9) and Eq. (10) respectively.

Step 10. Updating the centers and the widths of RBF kernel functions according to Eq. (11) and Eq. (12) respectively.

Step 11. Judging whether the control process is finished or not. If not, then $t \leftarrow t+1$ and turn to Step 2.

4 Simulation Research

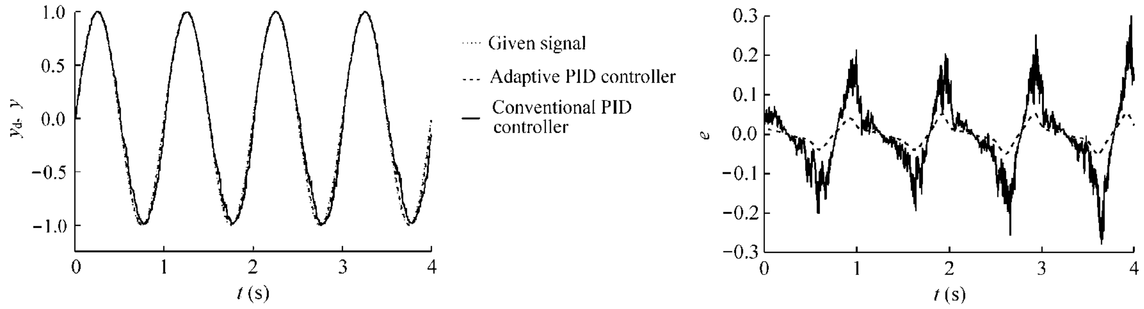
Using the following complex nonlinear system to test the validity of the proposed adaptive PID controller.

$$y(t) = \frac{ay(t-1)}{1+y^2(t-1)}u(t-1)+u(t-2) \quad (13)$$

where $a=1$. In order to examine the robustness of the system, parameter a occurs perturbation at t = 1 500 step according to Eq. (14).

$$a = 1.2\left(1-0.8\mathrm{e}^{-0.01(t-1500)}\right) \quad (14)$$

We applied the proposed adaptive PID controller and the conventional PID controller to track a given sine signal. Sampling period $T_s = 0.001$ s during the simulation. PID parameters of the conventional PID controller are set off-line as $k_{\text{P}} = 0.12$, $k_{\text{I}} = 0.32$ and $k_{\text{D}} = 0.08$ through the use of the Ziegler-Nichols tuning rule. The corresponding parameters for the adaptive PID controller are set as follows, $\alpha = 0.6$, $\beta = 0.4$, $\varepsilon = 0.01$, $\gamma = 0.98$, $\alpha_{\text{A}} = 0.013$, $\alpha_{\text{C}} = 0.01$, $\eta_{\mu} = 0.025$, $\eta_{\sigma} = 0.015$ and the topology structure of RBF network is 3-6-4. The detailed simulation results are shown in Fig. 3 where y_{d} and y are the given sine signal and the actual system output and e is the error between them. Simulation results indicate that the proposed adaptive PID controller exhibits perfect control performance and adapts to the changes of parameters of the controlled plant. Therefore, it has the characteristics of being strongly robust and adaptable.



(a) System output response (b) System error

Fig. 3 Simulation results

5 Conclusions

1) The possibility of combining reinforcement learning and the conventional PID controller was discussed. PID parameters can be tuned on-line through the use of the Actor-Critic reinforcement learning method and consequently an adaptive PID controller is formed.

2) The working manner of sharing the input and the hidden layers of RBF network by the Actor and the Critic not only decreases the demand for storage space from this learning system, but also avoids the repeated computation for the outputs of the hidden units.

3) The control precision is affected by the tolerant error band ε. The smaller ε, the higher the control precision and the slower the learning speed. We should make a tradeoff between precision and speed given the experience gained during the design of the controller.

4) Simulation results indicate that the proposed adaptive PID controller can realize stable tracking control for complex nonlinear systems. It is strongly robust for system disturbances, which is better than that of a conventional PID controller.

References

[1] Ang K H, Chong G, Li Y. PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, 2005, 13(4): 559–576.

[2] Liu K, Tang P R, Yang W M. Application of fuzzy-PID control system in full-mechanized coal face. *Journal of China University of Mining & Technology* (*English Edition*), 2005, 15(1): 48–51.

[3] Chen J H, Huang T C. Applying neural networks to on-line updated PID controllers for nonlinear process control. *Journal of Process Control*, 2004, 14(2): 211–230.

[4] Zhou K T, Zhen L X. Optimal design of PID parameters by evolution algorithm. *Journal of Huaqiao University* (*Natural Science*), 2005, 26(1): 85–88. (In Chinese)

[5] Gao Y, Chen S F, Lu X. Research on reinforcement learning technology. *ACTA Automatic Sinica*, 2004, 30(1): 86–100. (In Chinese)

[6] Wang X S, Cheng Y H, Sun W. Q learning based on self-organizing fuzzy radial basis function network. *Lecture Notes in Computer Science*, 2006, 3971: 607–615.

[7] Barto A G, Sutton R S, Anderson C W. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man and Cybernetics*, 1983, 13(5): 834–846.

[8] Wang X S, Cheng Y H, Sun W. Iterative learning controller for trajectory tracking tasks based on experience database. *Lecture Notes in Artificial Intelligence*, 2006, 3930: 780–789.

[9] Cheng Y H, Yi J Q, Zhao D B. Application of Actor-Critic Learning to adaptive state space construction. *Proceedings of the Third International Conference on Machine Learning and Cybernetics*. Shanghai: Institute of Electrical and Electronics Engineers Inc. Press, 2004: 26–29.

[10] Kondo T, Ito K. A reinforcement learning with evolutionary state recruitment strategy for autonomous mobile robots control. *Robotics and Autonomous Systems*, 2004, 46(2): 111–124.