IEEJ TRANSACTIONS ON ELECTRICAL AND ELECTRONIC ENGINEERING
IEEJ Trans 2021; **16**: 1354–1360
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI:10.1002/tee.23430

Paper

Design of a Reinforcement Learning PID Controller

Zhe Guan*[a], Member
Toru Yamamoto**, Fellow

This paper addresses a design scheme of a proportional-integral-derivative (PID) controller with a new adaptive updating rule based on reinforcement learning (RL) approach for nonlinear systems. A new design scheme that RL can be used to complement the conventional PID control technology is presented. In the proposed scheme, a single radial basis function (RBF) network is considered to calculate the control policy function of Actor and the value function of Critic simultaneously. Regarding the PID controller structure, the inputs of RBF network are system errors, the difference of output as well as the second-order difference of output, and they are composed of system states. The temporal difference (TD) error in the proposed scheme involves the reinforcement signal, the current and the previous stored value of the value function. The gradient descent method is adopted based on the TD error performance index, then, the updating rules can be yielded. Therefore, the network weights and the kernel function can be calculated in an adaptive way. Finally, the numerical simulations are conducted in nonlinear systems to illustrate the efficiency and robustness of the proposed scheme. © 2021 Institute of Electrical Engineers of Japan. Published by Wiley Periodicals LLC.

Keywords: reinforcement learning; PID control; Actor-Critic learning; RBF network; nonlinear system

Received 6 May 2020; Revised 9 September 2020

1. Introduction

Proportional-integral-derivative (PID) control is one of the most common control schemes and has been dominated the majority of industrial processes and mechanical systems, since it is of versatility, high reliability and ease of operation [1]. PID controllers can be manually tuned appropriately by the operators and control engineers based on the empirical knowledge when the mathematical model of the controlled plant is unknown. Some classical tuning methods, such as Ziegler–Nichols method [2] and Chien–Hrones–Reswich method [3], are applied to the process control and the performance then is significantly outperformed compared to the one that is manually tuned. However, those methods work well for simple controlled plants, but for complex systems with nonlinearity, the performance can not be guaranteed in the presence of uncertainty and unknown dynamics. Therefore, the adaptive PID control has been received considerable attention in recent years in order to deal with varying systems.

Several adaptive PID control strategies that include model-based adaptive PID control in [4–6], adaptive PID control based on the neural network [7,8]. It has been clarified that model-based adaptive PID control needs an assumption that the established model could represent the true plant dynamics exactly [9]. However, modeling complex systems are time-consuming and lack of accuracy, hence, the PID parameters may not be adjusted in a proper way. On the other hand, the adaptive PID control based on the neural network adopts the supervised learning to optimize the network parameters. Therefore, there are some limitations in the application of those methods, such as the teaching signal is hard to be obtained, and it is difficult to predict values for unlabeled data. As a result, the adaptive PID control based on various more advanced machine learning technologies has been discussed with the rapid development of computer science.

Bishop *et al.* [10] have clarified that machine learning is divided into three classes of algorithms: supervised learning, unsupervised learning and reinforcement learning (RL). RL differs significantly from both supervised and unsupervised learning. A definition of RL from [11] is expressed as: an RL agent has the goal of learning the best way to accomplish a task through repeated interactions with its environment. Alternatively, from the control perspective, RL refers to an agent (controller) that interacts with its environment (controlled system) and then modifies its actions (control signal) [12]. It has strong potential to combine the RL technology with the adaptive PID control to have an impact on process control applications, and it has been investigated in studies [13–15]. However, these studies adopted the same updating rule which compacted in one equation for three parameters, which leads to unknown mechanism of each parameter update. Moreover, the PID parameters trajectories were not provided. In addition, some existing methods [16–18] consider a system model to predict the future system states which are used to obtain the predictive value function for the next time step. Nevertheless, it is impractical to solve real problems, especially for process control problems.

Based on the observations above, this paper considers a PID controller with a new adaptive updating rule based on RL technology for nonlinear systems. It has been investigated that the Actor-Critic structure is the most general and successful

[a] Correspondence to: Zhe Guan, E-mail: guanzhe@hiroshima-u.ac.jp

*KOBELCO Construction Machinery, Dream-Driven Co-Creation Research Center, Hiroshima University, 1-4-1 Kagamiyama, Higashi-Hiroshima, 739-8527, Japan

**Academy of Science and Technology, Hiroshima University, 1-4-1 Kagamiyama, Higashi-Hiroshima, 739-8527, Japan

implementation to date [19]. Actor-Critic structure [20] has two separate parametric structures, one is for optimal control policy termed as Actor, the other structure is for the value function, namely the Critic. The Actor component applies control signal to a system, and a Critic component evaluates the performance using a temporal difference (TD) method. The idea of realization of the Actor-Critic is to use the RBF network. Under the Actor-Critic structure based on RBF network, the new adaptive updating rule can be designed. Furthermore, model-free RL removes the requirement for identifying a system model and becomes a powerful tool in dealing with process control. The TD method is applied to implement the model-free RL technique. In the proposed scheme, the TD error is defined without using predictive system states.

The remainder of this paper is organized in the following way. The problem formulation is discussed in Section 2, where an assumption is introduced as well. In Section 3, the adaptive PID controller based on Actor-Critic algorithm is proposed. Numerical simulations and comparative studies are conducted to illustrate the efficiency and feasibility in Section 4. Finally, Section 5 concludes this paper.

2. Problem Statement

2.1. System description Consider the following discrete-time systems described by nonlinear dynamics in the affine state-space difference equation form

$$\begin{aligned} x(t+1) &= f(x(t)) + g(x(t))u(t) \\ y(t) &= h(x(t), u(t-1)) \end{aligned} \tag{1}$$

with state $x(\cdot) \in \mathbb{R}$, control input $u(\cdot) \in \mathbb{R}$ and output $y(\cdot) \in \mathbb{R}$, respectively. $f(\cdot)$, $g(\cdot)$ and $h(\cdot)$ are assumed to be unknown in the proposed scheme.

It is required to provide two assumptions on the above system to capture the idea about RL technology.

Assumption 2.1. The above system satisfies the 1-step Markov property since the state at time $t+1$ only depends on the state and inputs at the previous time t, independent with the historical data.

This assumption is under the framework of Markov decision processes (MDP), whose objective is to achieve a specified goal through a satisfactory control policy. It is defined in a similar way with RL technology.

Assumption 2.2. The sign of partial derivatives of $h(\cdot)$ with respect to all arguments are known, and it is also regarded as the sign of system Jacobian.

2.2. Controller structure It is well recognized that a PID controller is applied to process systems, therefore, the derivative kick sometimes has an impact on the performance of the closed-loop system. As a result, this paper introduces the following velocity-type PID controller:

$$u(t) = u(t-1) + K_I(t)e(t) - K_P(t)\Delta y(t) - K_D(t)\Delta^2 y(t) \tag{2}$$

that is

$$\Delta u(t) = \boldsymbol{K}(t)\boldsymbol{\Theta}(t) \tag{3}$$

where $\boldsymbol{\Theta}(t) := [e(t), -\Delta y(t), -\Delta^2 y(t)]^T$ is composed of system states. Δ denotes the difference operator defined by $\Delta := 1 - z^{-1}$.

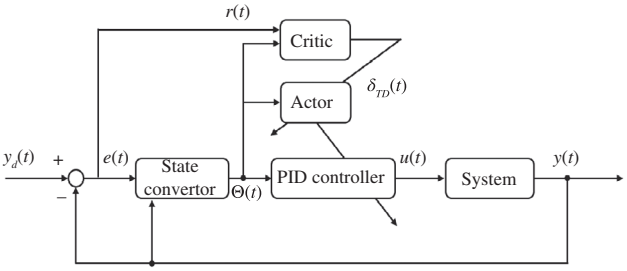


Fig. 1. The block diagram of the proposed scheme

The $\Delta^2 y(t)$ then becomes $\Delta^2 y(t) = y(t) - 2y(t-1) + y(t-2)$. And $\boldsymbol{K}(t) := [K_I(t), K_P(t), K_D(t)]$ is a vector of control parameters. $e(t)$ is the control error and is defined by the difference between reference signal y_d and system output y as follows:

$$e(t) = y_d(t) - y(t) \tag{4}$$

2.3. Objective The schematic diagram of the proposed method is shown in Fig. 1, in which the system state $\boldsymbol{\Theta}(t)$ is constructed based on $e(t)$ and current system output firstly, and then they will be used as inputs to the Actor-Critic structure. The Actor tunes the controller online using the observed system states along the system trajectory, while the Critic, which receives the system states, evaluates the control performance. The reinforcement signal $r(t)$ and the Critic are involved in the TD error $\delta_{TD}(t)$. The TD error is viewed as a crucial basis, and is applied to update the parameters' weights. As a result, the objective of this paper is to design a PID controller with new adaptive updating rule under the Actor-Critic structure.

3. Adaptive Controller Design

The proposed algorithm will be explained in detail in this section.

3.1. TD error We will first introduce a value function $V(\boldsymbol{\Theta}(t))$, which involves the information of system states, and the value function is updated at each time step. It is noteworthy that the predictive system states are not used in the implementation of the Critic design. In other words, the system model is not considered to predict system states. Alternatively, we store the previous value of the value function, namely, $V(t-1)$, therefore, the proposed scheme requires no prior knowledge about system model compared to other existing methods.

As a result, the TD error is defined shown as:

$$\delta_{TD}(t) = r(t) + \gamma V(\boldsymbol{\Theta}(t)) - V(\boldsymbol{\Theta}(t-1)) \tag{5}$$

with $0 < \gamma \leq 1$ a discount factor, which indicates the decay of current value function in the absence of reinforcement signal.

The TD error reveals that the learning based on the immediate reward, namely, the reinforcement signal, and the value function. Note that the definition of reinforcement signal $r(t)$ and value function $V(\boldsymbol{\Theta}(t))$ will be given in next subsection.

3.2. Actor-Critic learning based on RBF network The RBF network has been used as a technique to identify parameters by performing function mappings. The simple structure,

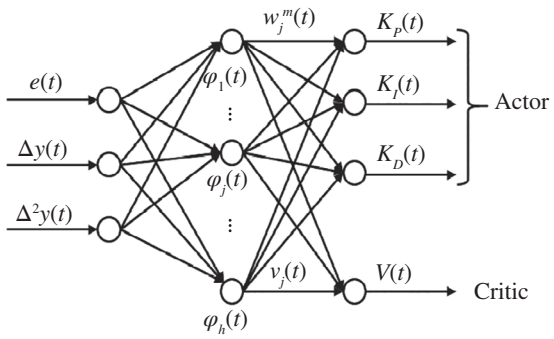


Fig. 2. RBF network topology with Actor-Critic structure

parameters convergence and adequate learning are recognized as merits of RBF network and are discussed in Ref. [21]. As a consequence, the implementation of Actor-Critic is used by RBF network, and the network topology is shown in Fig. 2. It consists of three-layer neural networks.

The input layer consists the available process measurements and system states are constructed. On the basis of the RBF network topology, it allows to pass the system states to the hidden layers which are shared by the Actor and the Critic directly. The control signal $u(t)$ and value function are generated by means of a simpler way that is the weighted sum of the function value associated with units in the hidden layer [22]. The detail of each layer is described as follows.

The input layer includes a vector which is $\boldsymbol{\Theta}(t) \in \mathbb{R}^3$, and it is passed to the hidden layer and is used to calculate the output of hidden unit.

In the hidden layer, the Gaussian function is selected as a kernel function of the hidden unit of RBF network, therefore, the output $\varphi(t)$ is shown as following:

$$\varphi_j(t) = \exp\left(-\frac{\|\boldsymbol{\Theta}(t) - \boldsymbol{\mu}_j(t)\|^2}{2\sigma_j^2(t)}\right), j = 1, 2, 3, \ldots, h \quad (6)$$

where $\boldsymbol{\mu}_j$ and σ_j are the center vector and width scalar of the unit, respectively, h is the number of the hidden units. The center vector is defined as follows:

$$\boldsymbol{\mu}_j(t) := [\mu_{1j}, \mu_{2j}, \mu_{3j}]^T$$

The third layer is called output layer where the outputs of the Actor and the Critic are involved. It should be noted that as mentioned previously the outputs are calculated in a simple and direct way. Therefore, it can yield the PID parameters $\boldsymbol{K}(t)$ in the following:

$$K_{P,I,D}(t) = \sum_{j=1}^{h} w_j^{P,I,D}(t)\varphi_j(t) \quad (7)$$

with the weights $w_j^m(t)$ between the jth hidden unit and output layer of the Actor, and m refers to the weights that are utilized to calculate the assigned parameters $K_P(t), K_I(t)$ and $K_D(t)$. The value function of critic part can be obtained as follows:

$$V(t) = \sum_{j=1}^{h} v_j(t)\varphi_j(t) \quad (8)$$

where $v_j(t)$ denotes the weight between the jth hidden unit and output layer of the Critic.

Those various output weights can be trained by gradient-based learning algorithm. Therefore, we can obtain the adaptive updating rule under user-specified parameters. Recall the [5], the reinforcement signal is defined as

$$r(t) := \frac{1}{2}\{y_d(t) - y(t)\}^2 \quad (9)$$

which indicates that the current response of the system is used, such that a system model is not required. The TD error then becomes

$$\delta_{TD}(t) = \frac{1}{2}\{y_d(t) - y(t)\}^2 + \gamma V(t) - V(t-1) \quad (10)$$

As a result, the cost function in this study is denoted in the following:

$$J(t) = \frac{1}{2}\delta_{TD}^2(t) \quad (11)$$

Thus, the partial differential equations with respect to each output weight of the Actor are developed as

$$w_j^P(t+1) = w_j^P(t) - \alpha_w \frac{\partial J(t)}{\partial w_j^P(t)} \quad (12)$$

where α_w is a learning rate, and

$$\begin{aligned}\frac{\partial J(t)}{\partial w_j^P(t)} &= \frac{\partial J(t)}{\partial \delta_{TD}(t)}\frac{\partial \delta_{TD}(t)}{\partial y(t)}\frac{\partial y(t)}{\partial u(t)}\frac{\partial u(t)}{\partial K_P(t)}\frac{\partial K_P(t)}{\partial w_j^P(t)} \\ &= \delta_{TD}(t)e(t)\{y(t) - y(t-1)\}\varphi_j(t)\frac{\partial y(t)}{\partial u(t)}\end{aligned} \quad (13)$$

$$\begin{aligned}\frac{\partial J(t)}{\partial w_j^I(t)} &= \frac{\partial J(t)}{\partial \delta_{TD}(t)}\frac{\partial \delta_{TD}(t)}{\partial y(t)}\frac{\partial y(t)}{\partial u(t)}\frac{\partial u(t)}{\partial K_I(t)}\frac{\partial K_I(t)}{\partial w_j^I(t)} \\ &= -\delta_{TD}(t)e^2(t)\varphi_j(t)\frac{\partial y(t)}{\partial u(t)}.\end{aligned} \quad (14)$$

$$\begin{aligned}\frac{\partial J(t)}{\partial w_j^D(t)} &= \frac{\partial J(t)}{\partial \delta_{TD}(t)}\frac{\partial \delta_{TD}(t)}{\partial y(t)}\frac{\partial y(t)}{\partial u(t)}\frac{\partial u(t)}{\partial K_D(t)}\frac{\partial K_D(t)}{\partial w_j^D(t)} \\ &= \delta_{TD}(t)e(t) \times \{y(t) - 2y(t-1) + y(t-2)\}\varphi_j(t)\frac{\partial y(t)}{\partial u(t)}\end{aligned} \quad (15)$$

It should be noted that a prior information about the system Jacobian $\partial y(t)/\partial u(t)$ is required to calculate the above equations. Here, we consider a relation $\epsilon = |\epsilon|\text{sign}(\epsilon)$, therefore, the system Jacobian is obtained by the following equation:

$$\frac{\partial y(t)}{\partial u(t)} = \left|\frac{\partial y(t)}{\partial u(t)}\right| \text{sign}\left(\frac{\partial y(t)}{\partial u(t)}\right) \quad (16)$$

with sign $(\epsilon) = 1(\epsilon > 0), -1(\epsilon < 0)$. Based on the above assumption, the sign of the system Jacobian can be obtained [23]. The updating rule for output weight of the Critic is

$$\begin{aligned}v_j(t+1) &= v_j(t) - \alpha_v \frac{\partial J(t)}{\partial v_j(t)} \\ &= v_j(t) + \alpha_v \delta_{TD}(t)\varphi_t(t),\end{aligned} \quad (17)$$

with a learning rate α_v.

The centers and the widths of hidden units in the hidden layer are considered to be updated in the following ways:

$$\mu_{ij}(t+1) = \mu_{ij}(t) - \alpha_\mu \frac{\partial J(t)}{\partial \mu_{ij}(t)} = \mu_{ij} + \alpha_\mu \delta_{TD}(t) v_j(t) \varphi_j(t) \frac{\boldsymbol{\Theta}_i(t) - \mu_{ij}(t)}{\sigma_j^2(t)} \quad (18)$$

with $\boldsymbol{\Theta}_i(t)$ denotes the element in $\boldsymbol{\Theta}(t) \in \mathbb{R}^3$, while,

$$\sigma_j(t+1) = \sigma_j(t) - \alpha_\sigma \frac{\partial J(t)}{\partial \sigma_j(t)} = \sigma_j + \alpha_\sigma \delta_{TD}(t) v_j(t) \varphi_j(t) \frac{\|\boldsymbol{\Theta}(t) - \mu_j(t)\|^2}{\sigma_j^3(t)} \quad (19)$$

where α_μ and α_σ are learning rates of center and width, respectively.

3.3. Algorithm summary The every design step of the proposed adaptive PID controller under Actor-Critic structure based on RBF network is presented in Algorithm 1.

Algorithm 1 Adaptive PID controller under Actor-Critic based on RBF network

1: Initialize instant $t = 0$, control input signal $u(0)$ and reference signal $y_d(t)$.
2: Initialize the parameters $w_j^{P,I,D}(0)$, $v_j(0)$, $\mu_{ij}(0)$, $\sigma_j(0)$ and set the values for the use-specified learning rates α_w, α_v, α_μ, α_σ.
3: **for** $t = 0 : EndTime$
4: Observe the system output $y(t)$ and then the system error $e(t)$ can be obtained.
5: Compute the kernel function (6) in hidden layer.
6: Calculate the output of Actor, that is the current PID parameters from (7), and the output of Critic value function $V(t)$ from (8) at time t.
7: Obtain the TD error $\delta_{TD}(t)$ from (10) together with stored value $V(t-1)$.
8: Update the weights of the PID parameters by (13)–(15) and the weights of the value function according to (17).
9: Update the centers and the widths of RBF kernel functions by (18)–(19).
10: **end for**

4. Numerical Simulations

The proposed scheme has been implemented on two numerical simulations and comparative studies in this section to evaluate the efficiency and feasibility.

4.1. Case one: System with a hysteresis Consider the following nonlinear system from [24]:

$$y(t+1) = \frac{y(t)y(t-1)[y(t)+2.5]}{1+y(t)^2+y(t-1)^2} + u(t) + \xi(t) \quad (20)$$

where $\xi(t)$ denotes the Gaussian noise with zero mean and variance of 0.01^2. The reference signal values are set as follows:

$$y_d(t) = \begin{cases} 2.5(0 \le t < 100) \\ 3.5(100 \le t < 200) \\ 1(200 \le t < 300) \\ 3(300 \le t < 400) \end{cases} \quad (21)$$

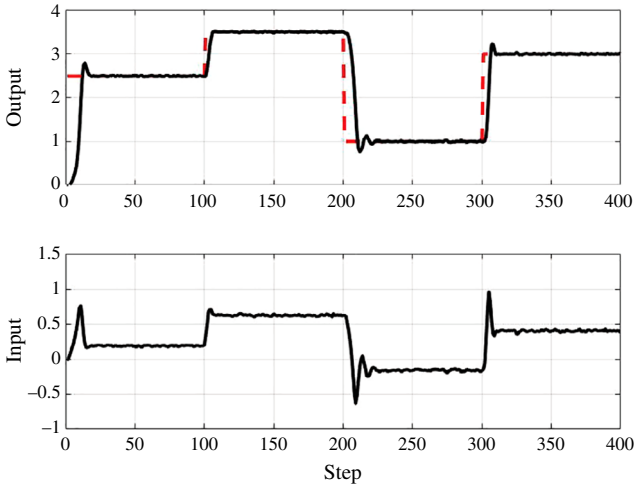


Fig. 3. Control result obtained by the proposed scheme for case one

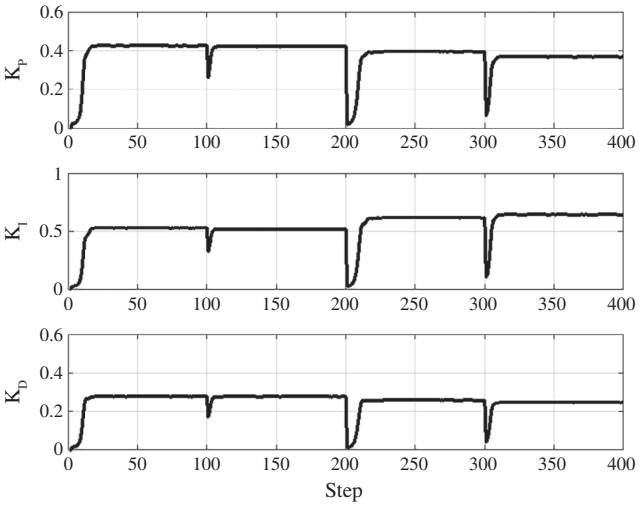


Fig. 4. Trajectories of adaptive PID parameters for case one

The user-specified learning rates included in the proposed are summarized as follows:

$$\alpha_w = 0.011, \alpha_v = 0.05, \alpha_\mu = 0.001, \alpha_\sigma = 0.005$$

and the coefficient γ equals to 0.98. The hidden units in topology RBF network are decided as 3. The initial PID parameters in the proposed scheme are set as:

$$\boldsymbol{K}(0) = [0,\ 0,\ 0]^T$$

The simulation results are presented in Fig. 3, where the output signal can track the reference signal by employing the proposed scheme. Regardless of the strong nonlinearity, the proposed scheme can work well when the reference signal is changed. Moreover, the PID parameters are depicted in Fig. 4, where they can be updated based on the updated weights. Furthermore, they ultimately tended to reach constant values.

The comparative study for this case is discussed, the result of which is shown in Fig. 5. In this case, the conventional PID tuning

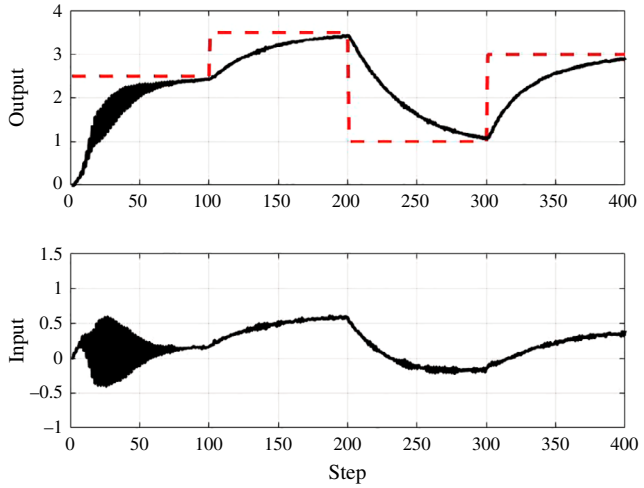


Fig. 5. Control result obtained by the conventional scheme for case one

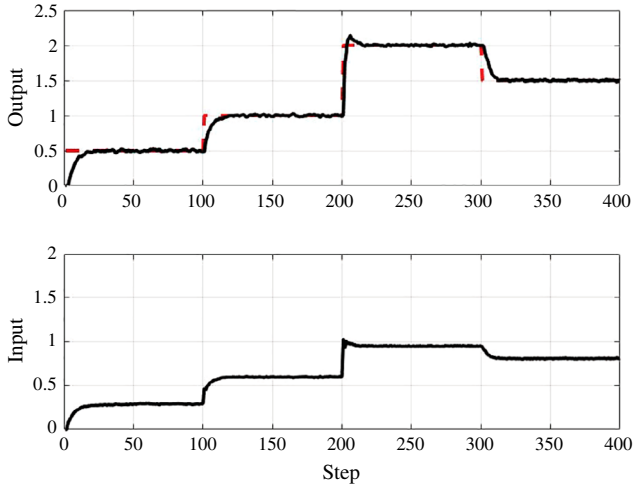


Fig. 6. Control result obtained by the proposed scheme for case two

method is employed and the parameters are calculated based on Chien–Hrones–Reswich method [3]:

$$K_P = 0.645, K_I = 0.028, K_D = 0.327$$

Figure 5 shows unsatisfactory performance in terms of tracking property based on the conventional scheme, since it can not deal with the nonlinearity in the system.

4.2. Case two: Hammerstein model Consider the following non-linear Hammerstein models from Ref. [25].

$$\left.\begin{aligned} y(t) &= 0.6y(t-1) - 0.1y(t-2) \\ &\quad + 1.2x(t-1) - 0.1x(t-2) + \xi(t) \\ x(t) &= u(t) - u^2(t) + u^3(t) \end{aligned}\right\} \tag{22}$$

where $\xi(t)$ is the white Gaussian noise with zero mean and variance of 0.01^2. Besides, the reference signal values are set as follows:

$$y_d(k) = \begin{cases} 0.5(0 \le t < 100) \\ 1(100 \le t < 200) \\ 2(200 \le t < 300) \\ 1.5(300 \le t < 400) \end{cases} \tag{23}$$

The user-specified learning rates are set as the same values as the case one.

The proposed scheme is employed, and the control results are shown in Fig. 6, where it is apparent that desirable control results can be obtained. The trajectories of PID parameters for this case are depicted in Fig. 7, where the parameters are adjusted suitably according to the change of reference signal. They reach to constant values along with time goes.

The comparative study for this case is discussed by employing a conventional PID tuning method, whose parameters are shown below.

$$K_P = 0.486, K_I = 0.227, K_D = 0.122$$

These PID parameters are calculated by Chien–Hrones–Reswich method [3]. The result is shown in Fig. 8, where the oscillation is shown because of nonlinearity.

As a result, the effectiveness and feasibility are finally confirmed through these two case studies.

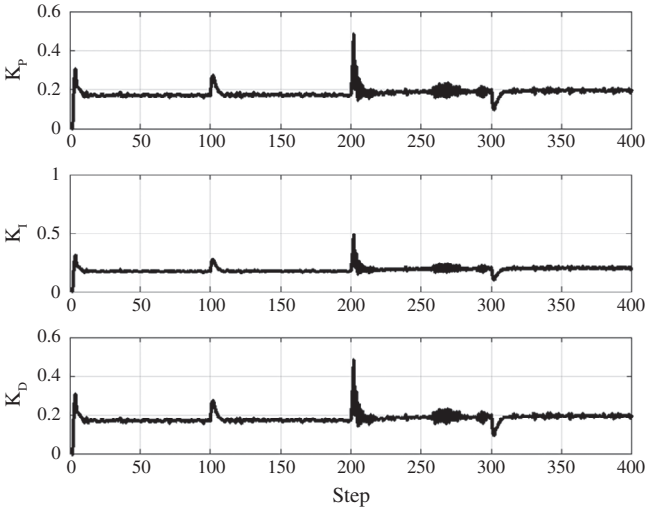


Fig. 7. Trajectories of adaptive PID parameters for case two

5. Conclusions

This paper has studied a novel adaptive PID controller under the Actor-Critic structure based on RBF network for nonlinear systems. A new adaptive updating rule was presented via weights update in the network. First, the conventional PID controller combined with the RL on the basis of RBF network, and the parameters were adapted in an on-line manner. The mechanism of the proposed scheme did not require a system model to predict the system states. The TD error was defined by considering the current and previous stored value of the value function. Then, the hidden layer of RBF network was shared by the Actor and the Critic. The storage space could be saved and the computation cost was reduced for the outputs of the hidden units. In addition, the initial PID parameters are set as zero, which means there is no need to know the *prior* knowledge on the controlled system.

Finally, numerical simulations were given to indicate the efficiency and feasibility of the proposed scheme for complex nonlinear systems. The PID parameters based on the new adaptive updating rule reached to constant values. The proposed scheme

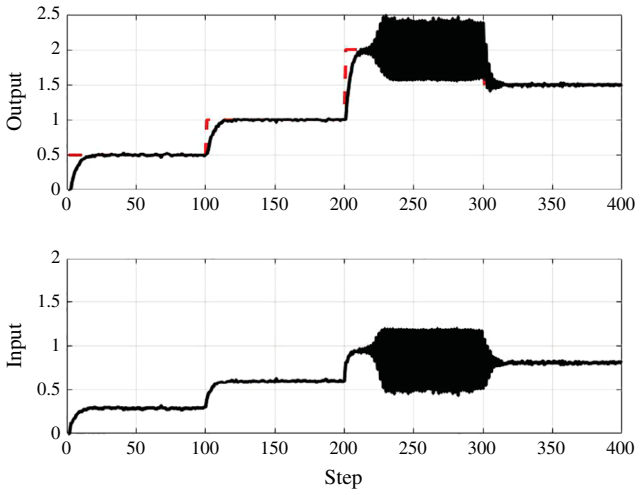


Fig. 8. Control result obtained by the conventional scheme

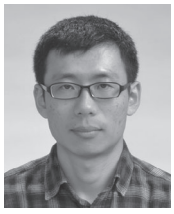
will be employed in a real system to verify the effectiveness from the practical point of view in near future.

Acknowledgment

This work was supported by 'Hiroshima Manufacturing Digital Innovation Creation Program' with the Grant from Cabinet Office, Government of Japan and Hiroshima Prefecture.

References

(1) Åström KJ, Hägglund T. *PID Controllers: Theory, Design and Tuning*. 2nd ed. Research Triangle Park, NC: Instrument Society of America; 1995.
(2) Ziegler JG, Nichols NB. Optimum settings for automatic controllers. *Transactions of the ASME* 1942; **64**:759–768.
(3) Chien KL, Hrones JA, Reswick JB. On the automatic control of generalized passive systems. *Transactions of the ASME* 1952; **74**(2):175–185.
(4) Chang WD, Hwang RC, Hsieh JG. A multivariable on-line adaptive PID controller using auto-tuning neurons. *Engineering Application of Artificial Intelligence* 2003; **16**:57–63.
(5) Yamamoto T, Shah S. Design and experimental evaluation of multivariable self-tuning PID controller. *IEE Proceedings-Control Theory and Applications* 2004; **151**(5):645–652.
(6) Yu DL, Chang TK, Yu DW. A stable self-learning PID control for multivariable time varying systems. *Control Engineering Practice* 2007; **15**(12):1577–1587.
(7) Chen JH, Huang TC. Applying neural networks to on-line updated PID controllers for nonlinear process control. *Journal of Process Control* 2004; **14**(2):211–230.
(8) Liao YT, Koiwai K, Yamamoto T. Design and implementation of a hierarchical-clustering CMAC PID controller. *Asian Journal of Control* 2019; **21**(3):1077–1087.
(9) Hou ZS, Chi RH, Gao HJ. An overview of dynamic linearization based data-driven control and applications. *IEEE Transactions on Industrial Electronics* 2016; **64**(5):4076–4090.
(10) Bishop CM. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York. Inc: Secaucus, NJ; 2006.
(11) Sutton RS, Barto AG. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press; 2018.
(12) Lewis FL, Vrabie D. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits and Systems Magazine* 2009; **9**(3):32–50.
(13) Wang XS, Cheng YH, Sun W. A proposal of adaptive PID controller based on reinforcement learning. *Journal of China University Mining and Technology* 2007; **17**(1):40–44.
(14) Howell MN, Best MC. On-line PID tuning for engine idle-speed control using continuous action reinforcement learning automata. *Control Engineering Practice* 2000; **8**:147–154.
(15) Jin ZS, Li HC, Gao HM. An intelligent weld control strategy based on reinforcement learning approach. *The International Journal of Advanced Manufacturing Technology* 2019; **100**:2163–2175.
(16) Prokhorov DV, Santiago RA, Wunsch DC II. Adaptive critic designs: A case study for neuro-control. *Neural Networks* 1995; **8**:1367–1372.
(17) Morinelly JE, Ydstie BE. Dual MPC with reinforcement learning. *IFAC-PapersOnLine* 2016; **49**:266–271.
(18) Shin J, Lee JH, Realff MJ. Operational planning and optimal sizing of microgrid considering multi-scale wind uncertainty. *Applied Energy* 2017; **195**:616–633.
(19) Shin J, Badgwell TA, Liu KH, Lee JH. Reinforcement learning - overview of recent progress and implications for process control. *Computers and Chemical Engineering* 2019; **127**:282–294.
(20) Barto AG, Sutton RS, Anderson C. Neuron-like adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics* 1983; **SMC-13**:834–846.
(21) Elanayar SVT, Shin YC. Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems. *IEEE Transaction on Neural Network* 1994; **5**(4):584–603.
(22) Roger Jang JS, Sun CT. Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Transaction on Neural Network* 1993; **4**(1):156–159.
(23) Omatu S, Marzuki K, Rubiyah Y. *Neuro-Control and Its Applications*. Springer-Verlag: London, U.K.; 1995.
(24) Narendra KS, Parthasarathy K. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks* 1990; **1**(1):4/27–4/27.
(25) Zi-Qiang L. On identification of the controlled plants described by the Hammerstein system. *IEEE Transactions on Automatic Control* 1994; **Ac-39**(2):569–573.



Zhe Guan (Member) received the M.S. degree and D.Eng. degree in control system engineering from the Hiroshima University, Japan, in 2015 and 2018, respectively. He is currently working in KOBELCO Construction Machinery Dream-Driven Co-Creation Research Center, Hiroshima University, Japan. He had worked in Micron Japan as a dry etch engineer in 2018. He was a visiting postdoc with the Department of Electrical and Computer Engineering, University of Alberta from April to June in 2019. His current research interests are in area of adaptive control, data-driven control, reinforcement learning and their applications. He is a member of the Society of Instrument and Control Engineers in Japan (SICE).

Toru Yamamoto (Fellow) received the B.Eng. and M.Eng. degrees from the Tokushima University, Japan, in 1984 and 1987, respectively, and the D.Eng. degree from Osaka University, Japan, in 1994. He is currently a Professor with the Department of System Cybernetics, Graduate School of Engineering, Hiroshima University, Japan, and a leader of the National Project on Regional Industry Innovation with

support from the Cabinet Office, Government of Japan. He was a Visiting Researcher with the Department of Mathematical Engineering and Information Physics, University of Tokyo, Japan, in 1991, and an Overseas Research Fellow of the Japan Society for Promotion of Science (JSPS) with the Department of Chemical and Materials Engineering, the University of Alberta for 6 months in 2006. His current research interests are in area of self-tuning and learning control, data-driven control, and their implementation for industrial systems. Dr. Yamamoto was a National Organizing Committee Chair of 5th International Conference on Advanced Control of Industrial Processes (ADCONIP 2014) and General Chair of SICE (Society of Instrument and Control Engineers in Japan) Annual Conference 2019, both held in Hiroshima. He is also a Fellow of SICE and Japan Society of Mechanical Engineers (JSME).