Neural Computing and Applications (2020) 32:9761–9775
https://doi.org/10.1007/s00521-019-04509-x

ORIGINAL ARTICLE

Knowledge-based reinforcement learning controller with fuzzy-rule network: experimental validation

Chidentree Treesatayapun[1]

Received: 8 March 2019 / Accepted: 21 September 2019 / Published online: 3 October 2019

Abstract
A model-free controller for a general class of output feedback nonlinear discrete-time systems is established by action-critic networks and reinforcement learning with human knowledge based on IF–THEN rules. The action network is designed by a single input fuzzy-rules emulated network with the set of IF–THEN rules utilized by the relation between control effort and plant's output such as IF *the output is high* THEN *the control effort should be reduced*. The critic network is constructed by a multi-input FREN (MiFREN) for estimating an unknown long-term cost function. The set of IF–THEN rules for MiFREN is defined by the general knowledge of optimization such that IF *the quadratic values of control effort and tracking error are high* THEN *the cost function should be high*. The convergence of tracking error and bounded external signals can be guaranteed by Lyapunov direct method under general assumptions which are reasonable for practical plants. A computer simulation system is firstly provided to demonstrate the design method and the performance of the proposed controller. Furthermore, an experimental system with the prototype of DC-motor current control is conducted to show the effectiveness of the control scheme.

Keywords Model-free adaptive control · Reinforcement learning · Nonlinear discrete-time systems · Fuzzy neural network · DC-motor current control

1 Introduction

Mathematical models of practical plants, in general, are hardly determined with appropriate accuracy. To design the controller without a mathematical model of a controlled plant in discrete-time domain, the model-free adaptive control schemes have been proposed by using only the set of input–output data [1–3]. In general, the full-state feedback has been required to gain enough information such that the works of [4] for the linear plant and [5, 6] for nonlinear systems. On the other hand, the output feedback control schemes have been less studied than the state feedback schemes because output feedback controllers have been much more difficult in many cases [7, 8]. In order to handle the applications with unknown nonlinear discrete-time systems and lacking state measurement, model-free adaptive controllers based on output feedback have been developed with the closed-loop stability guarantee [9–12]. Nevertheless, the stability analysis is only a bare minimum requirement for controller designs, but the optimization of a prescribed cost function is preferred for several control applications [13–15].

The optimal control schemes based on the concept of action-critic networks have been proposed to determine the estimated solution of the Hamilton–Jacobi–Bellman (HJB) equation [16] within the manner of reinforcement learning (RL) algorithms [17, 18]. In general, both action and critic networks have been established by artificial neural networks (ANN) when the unknown cost function has been approximated by a critic-ANN and the solution of control effort has been obtained by an action-ANN [19, 20]. The architectures and learning schemes of action-critic networks have been proposed such that "neuro dynamic programming" [21], "adaptive critic design" [22] and "adaptive dynamic programming" for discrete-time systems [23] and continuous-time systems [24]. In [25], the controlled plant has been considered as a gray-box system

✉ Chidentree Treesatayapun
treesatayapun@gmail.com; chidentree@cinvestav.edu.mx

[1] Department of Robotic and Advanced Manufacturing, CINVESTAV, 25903 Ramos Arizpe, Coah., Mexico

and the action-critic structure has been proposed to design the adaptive controller with nearly optimization manner based on RL algorithm. Consideration of approximation errors, the generalized policy iteration has been developed in [26]. Both value and policy iterations play an importance role for solving optimal control problems, but both iterations seem inconvenient for implementation with practical plants. That motivates us to design the learning algorithm for both critic and action networks without inner iteration.

Currently, they have a few works for the implementation of practical systems with action-critic networks and RL learning because the standard algorithms cannot be directly applied for time-varying conditions and uncertainties which are common for application plants [27]. Furthermore, the measurement of full-state variables is generally required to design controllers and learning algorithms [28, 29]. Together with the economic reason, output feedback control schemes are strongly desired for a large class of practical plants. Recently, the output feedback controllers based on RL algorithms have been proposed with the condition of persistent excitation (PE) [30]. The PE condition is generally required to be satisfied for adaptive algorithms with stability analysis. In [31], the PE condition can be relaxed with the ANN control scheme for nearly optimal regulation scheme, but the controller is limited for a class of affine nonlinear discrete-time systems. For a class of non-affine systems, the Q-learning algorithm based on critic-action networks has been proposed in [32–34], but it has been emphasized on state-feedback scheme and regulation problem. For practical perspective, the output feedback controller will be developed by the action-critic structure and the online learning algorithm only.

Fuzzy systems have been successfully utilized for the presence of robustness and uncertainties of optimal controllers when mathematical models of controlled plants have been considered as unknown [35]. In [36], fuzzy hyperbolic model has been developed as an action network tuned by the internal reinforcement signal for a class of unknown discrete-time systems, but only the regulation problem has been discussed. Based on the back-stepping adaptive control, the uncertainties and unknown systems have been handled [37, 38], but the full-state feedback has been required to design controllers. The design of output feedback controller based on fuzzy systems has been proposed by [39], but this controller has been conducted by a class of continuous-time systems with unity control gain. Recently, the controller based on a recurrent-fuzzy neural network with RL has been proposed by [40] for a class of nonlinear discrete-time systems, but only the tracking error has been selected for the reward function of the critic network.

In this article, the controlled plant is considered as a class of non-affine discrete-time systems when the mathematical model is unknown. To design the controller without any model, the model-free adaptive control scheme is established by an action-critic networks architecture with RL algorithm. The control signal is generated via an action network constructed by a single input fuzzy-rules emulated network (FREN) [41]. The set of IF–THEN rules for FREN is created by the human knowledge according to the relation between the control signal and the plant's output [42] such that

Action IF *Higher output is desired,* THEN *Larger control signal is requested.*

Within the manner of optimization between the tracking error and the energy of control signal, a critic network is established to estimate the long-term cost function. A multi-input fuzzy-rules emulated network (MiFREN) is implemented to create a critic network with the set of IF–THEN rules as

Critic IF *Error is big* and *Control energy is large,* THEN *Reward should be low*.

This reward can lead to the cost function generated by MiFREN with the relation such that the lower cost function can be obtained when the tracking error and the control energy are tiny. The main contributions of this article are shortly listed as the followings:

- Unlike other works such that [17, 25, 29, 30, 34], action-critic schemes have been designed by ANNs with random weight parameters; in this work, both action and critic networks are designed by IF–THEN rules utilized by human knowledge of the controlled plant and the controller's actuator that allows the engineer to design the structure and adjustable parameters in the sense of engineering not in the random aspect.
- The online learning algorithm is developed without inner policy and value iterations while the convergence of tracking error and internal signals can be guaranteed. Unlike a case of event-trigger and sampling time systems such that [23, 27, 33, 43], the proposed controller can be utilized for more extensive discrete-time systems.
- The tracking controller is designed without the transformation of the original systems to be the augmented system dynamic that allows the proposed controller be able to be implemented directly for a large class of practical plants such as the prototype of DC-motor current control in this work.

The rest of this article is organized as follows. A class on nonlinear discrete-time systems and problem formulation is mentioned in Sect. 2. Section 3 introduces the design of action and critic networks with the concept of IF–THEN

rules related on the controlled plant's characteristic. The learning algorithm is developed in Sect. 4 with convergence analysis for tracking error and internal signals. The computer simulation system is firstly utilized to demonstrate the design procedure and the performance of the proposed controller with a selected nonlinear plant in Sect. 5.1. Secondly, in Sect. 5.2, the experimental system with a DC-motor current control is constructed to demonstrate the effectiveness and the online learning ability against the nonlinearity and uncertainty terms of practical systems. Section 6 draws the conclusions.

2 Problem statement: a class of nonlinear discrete-time systems

The block diagram in Fig. 1 presents our prototyping DC-motor current control system which has input terminal as control effort $u(k) \in \mathbb{R}$ and output terminal as measured current $y(k+1) \in \mathbb{R}$ when k denotes as k^{th} sampling time index. The control signal $u(k)$ is a driving voltage generated by a data-acquisition card (CONTEC® AIO-160802L-LPE). The motor current $y(k+1)$ is measured by the instrument circuit connected with analog input of AIO-160802L-LPE. This plant is considered as an unknown nonlinear system with input $u(k)$ and output $y(k+1)$. The mathematical model of this system will not be required to design our controller and stability analysis. The nonlinear behavior of this DC-motor driving system can be demonstrated in Fig. 2 as a *V–I* curve when input voltage and motor current are denoted as control effort $u(k)$ and current output $y(k+1)$, respectively. Without any information about system's mathematical model, this controlled plant can be considered as a class of non-affine discrete-time system and the system dynamic can be formulated as

$$y(k+1) = f_o(u(k), \ldots, u(k-l_u), y(k), \ldots, y(k-l_y)) + d(k), \tag{1}$$

when $f_o(-)$ is an unknown nonlinear function, l_u and l_y are unknown system orders and $d(k)$ is a bounded disturbance as $|d(k)| \le d_M^o$. Let us define $\chi_i(k) = [u(k-1) \ldots u(k-l_u)\, y(k) \ldots y(k-l_y)]^T$, thus the system dynamic (1) can be rewritten as

$$y(k+1) = f_o(u(k), \chi_i(k)) + d(k). \tag{2}$$

Without loss of generality, the following assumptions are stated for the nonlinear function $f_o(-)$.

Assumption 1 The nonlinear function $f_o(-)$ is continuous with respect to the first argument $u(k)$ or $\frac{\partial f_o(u(k),\chi_i(k))}{\partial u(k)}$ is existed.

Assumption 2 Two constants g_m and g_M are existed where

$$0 < g_m \le \left| \frac{\partial f_o(u(k), \chi_i(k))}{\partial u(k)} \right| \le g_M. \tag{3}$$

Those assumptions are standard requirements for several nonlinear discrete-time control schemes. In this work, the

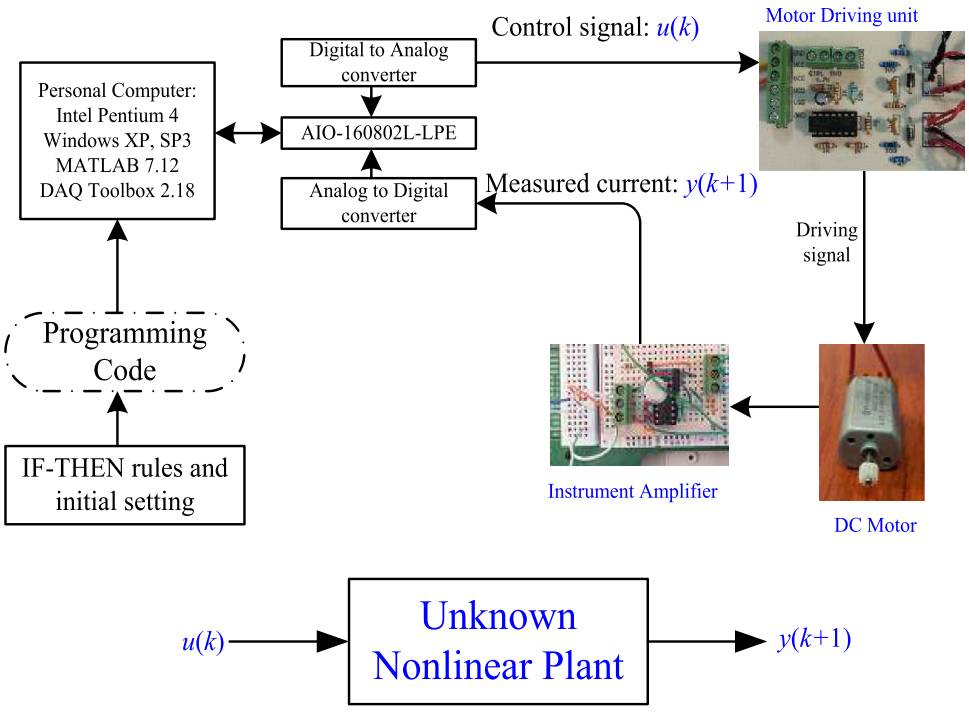

Fig. 1 DC-motor current control configuration

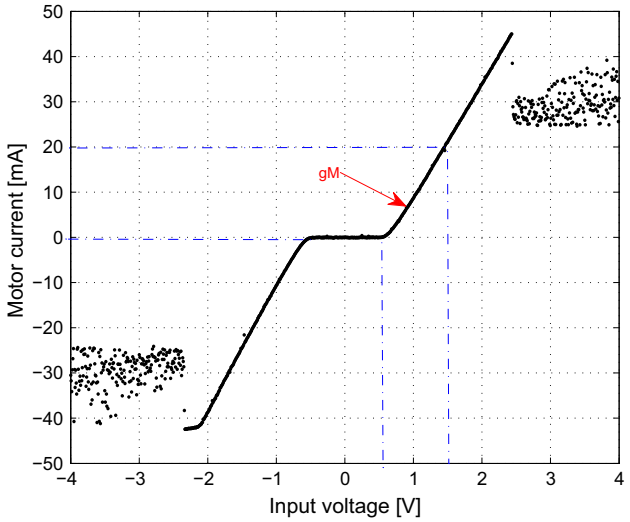

Fig. 2 *V–I* characteristic of DC-motor driver system

proposed control scheme will be designed under the conditions that the nonlinear function $f_o(-)$ and the boundaries in (3) are completely unknown. The boundaries in (3) can be estimated by *V–I* curve or experimental data. For example, in this application the estimated value of (3) can be obtained by the estimated tangent of the curve in Fig. 2 as

$$g_M = \frac{20-0}{1.5-0.5} = 20. \tag{4}$$

The proposed control scheme will be developed to handle the tracking problem for a class of system in (1) by adaptive networks and stability analysis in the next section.

3 Action and critic architecture based on FRENs

In this work, the control scheme is proposed by the concept of action and critic networks presented by Fig. 3 when an action network is established by FRENaction or FRENa and a critic network is created by MiFRENcritic or MiFRENc. The action network or FRENa is designed to generate the control effort for the controlled plant, and parameters inside this network are tuned to minimize the estimated cost function obtained by the critic network or MiFRENc. The reword function for MiFRENc is established by IF–THEN rules according to the relation of tracking error and control effort. Two sets of IF–THEN rules and network architectures will be introduced for both FRENa and MiFRENc in the followings subsections.

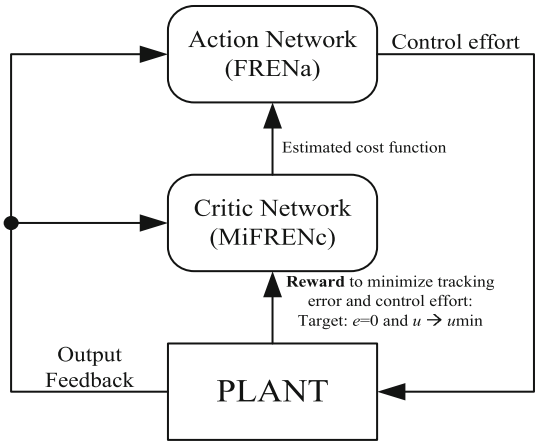

Fig. 3 FREN: action and critic networks architecture

3.1 Action network: FRENa

According to the human knowledge related on the controlled plant, the IF–THEN rules can be defined as

"IF$e(k)$ is Positive Large THEN $u(k)$ is Negative Large",

when $e(k)$ denotes as the tracking error given by

$$e(k) = y(k) - r(k), \tag{5}$$

where $r(k)$ is the desired trajectory. That means the error determined by (5) is large in positive thus the output $y(k)$ should be reduced by the large in negative of control effort $u(k)$. In this work, the set of IF–THEN rules can be defined as

$$\begin{aligned}
&\text{IF } e(k) \text{ is NL} && \text{THEN } u_1(k) = \beta_{\text{PL}}(k)\mu_{\text{NL}}(e_k),\\
&\text{IF } e(k) \text{ is NM} && \text{THEN } u_2(k) = \beta_{\text{PM}}(k)\mu_{\text{NM}}(e_k),\\
&\text{IF } e(k) \text{ is NS} && \text{THEN } u_3(k) = \beta_{\text{PS}}(k)\mu_{\text{NS}}(e_k),\\
&\text{IF } e(k) \text{ is Z} && \text{THEN } u_4(k) = \beta_{\text{Z}}(k)\mu_{\text{Z}}(e_k),\\
&\text{IF } e(k) \text{ is PS} && \text{THEN } u_5(k) = \beta_{\text{NS}}(k)\mu_{\text{PS}}(e_k),\\
&\text{IF } e(k) \text{ is PM} && \text{THEN } u_6(k) = \beta_{\text{NM}}(k)\mu_{\text{PM}}(e_k),\\
&\text{IF } e(k) \text{ is PL} && \text{THEN } u_7(k) = \beta_{\text{NL}}(k)\mu_{\text{PL}}(e_k),
\end{aligned}$$

The notations of linguistic variables N, P, L, M, S and Z denote as negative, positive, large, medium, small and zero, respectively. The nonlinear function $\mu_{\square}(e_k)$ is a membership function and $\beta_{\square}(k)$ is an adjustable parameter for linguistic value $\square$, where $\square$ denotes as linguistic values such that Negative Large (NL), Negative Medium(NM),..., Zero(Z), ..., Positive Large(PL) for all using membership functions. Regarding to the relation of FREN's computation [41], the control effort can be obtained by

$$u(k) = \sum_{i=1}^{7} u_i(k). \tag{6}$$

To simplify, the control effort can be rewritten as

$$u(k) = \beta_a^T(k)\phi_a(k), \tag{7}$$

when

$$\beta_a(k) = [\beta_{\mathrm{PL}}(k) \quad \beta_{\mathrm{PM}}(k) \quad \cdots \quad \beta_{\mathrm{NL}}(k)]^T, \tag{8}$$

and

$$\phi_a(k) = [\mu_{\mathrm{NL}}(e_k) \quad \mu_{\mathrm{NM}}(e_k) \quad \cdots \quad \mu_{\mathrm{PL}}(e_k)]^T. \tag{9}$$

The network architecture of FRENa is depicted in Fig. 4. According to the universal function approximation of FREN [41], it exists the ideal parameter β_a^* that leads to

$$u^*(k) = \beta_a^{*T}\phi_a(k) + \varepsilon_a(k), \tag{10}$$

when $\varepsilon_a(k)$ is the approximation error of FRENa. By using (2), the error dynamic can be obtained as

$$e(k+1) = f_o(u(k), \chi_i(k)) + d(k) - r(k+1). \tag{11}$$

Adding and subtracting $f_o(u^*(k), \chi_i(k))$ into (11), thus, the error dynamic can be rewritten as

$$e(k+1) = f_o(u(k), \chi_i(k)) - f_o(u^*(k), \chi_i(k)) + d(k). \tag{12}$$

By using mean value theorem and Assumption 1, the error dynamic (12) can be obtained as

$$e(k+1) = g(u^i(k), \chi_i(k))[u(k) - u^*(k)] + d(k), \tag{13}$$

where

$$g(u^i(k), \chi_i(k)) = \frac{\partial f_o(u^i(k), \chi_i(k))}{\partial u^i(k)}, \tag{14}$$

when $u^i(k) \in [\min\{u_k^*, u_k\}, \max\{u_k^*, u_k\}]$. Substituting $u^*(k)$ with (10) and $u(k)$ with (7) and defining $g(u^i(k), \chi_i(k)) = g(k)$, this, the error dynamic (13) can be rewritten as

$$e(k+1) = g(k)[\beta_a(k) - \beta_a^*]^T\phi_a(k) - g(k)\varepsilon_a(k) + d(k). \tag{15}$$

Let us define $\tilde{\beta}_a(k) = \beta_a(k) - \beta_a^*$, $d_a(k) = d(k) - g(k)\varepsilon_a(k)$ and $\Lambda_a(k) = \tilde{\beta}_a^T(k)\phi_a(k)$, thus, we obtain

$$e(k+1) = g(k)\Lambda_a(k) + d_a(k). \tag{16}$$

The error dynamic obtained in (16) indicates the relation with the difference of ideal and adjustable parameters of action network FRENa and its approximation error.

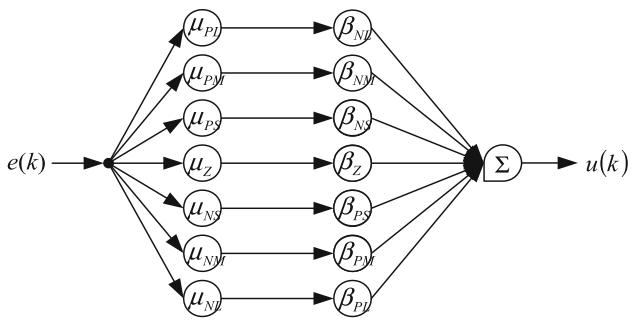

Fig. 4 FRENa network architecture

3.2 Critic network: MiFRENc

In order to minimize for both tracking error and control energy, an infinite-horizon cost function is defined as

$$L(k) = \sum_{i=k}^{\infty} \gamma_L^{i-k}[pe^2(i) + qu^2(i)], \tag{17}$$

when p and q are positive constants and $0 < \gamma_L \leq 1$ as a discount factor. Let us rearrange (17) as

$$\begin{aligned} L(k) &= pe^2(k) + qu^2(k) \\ &\quad + \gamma_L \sum_{i=k+1}^{\infty} \gamma_L^{i-(k+1)}[pe^2(i) + qu^2(i)], \\ &= l(k) + \gamma_L L(k+1), \end{aligned} \tag{18}$$

when $l(k)$ is the local cost function defined by

$$l(k) = pe^2(k) + qu^2(k). \tag{19}$$

Let us define $\xi_k = [e^2(k) : u^2(k)]$ as the current states including the tracking error and the control effort, thus we have

$$L(k) = l(\xi_k) + \gamma_L L(k+1). \tag{20}$$

For the closed-loop system with output feedback, it is clear that the next time index of tracking error is the function of current control effort and the current control effort is the function of current tracking error that leads to

$$\xi_{k+1} = [e^2(k+1) : u^2(k+1)] = \mathfrak{f}_{\xi}(\xi_{\mathfrak{k}}), \tag{21}$$

when $\mathfrak{f}_{\xi}(-)$ is an unknown analytic function. According to composition of functions, we have

$$\xi_{k+2} = \mathfrak{f}_{\xi} \circ \mathfrak{f}_{\xi}(\xi_{\mathfrak{k}}) \triangleq \mathfrak{f}_{\xi}^{\mathfrak{i}}(\xi_{\mathfrak{k}}). \tag{22}$$

Combination (20–22) and all future steps, it leads us to

$$L(k) = l(\xi_k) + \gamma_L l(F_{\xi}(\xi_k)), \tag{23}$$

where $F_{\xi}(\xi_k) = \mathfrak{f}_{\xi}^{\mathfrak{j}}(\xi_{\mathfrak{k}})$ for $j = 1 \rightarrow \infty$. Regarding (23), the cost function in (17) can be estimated by MiFRENc as $\hat{L}(k)$. This network has two inputs $e^2(k)$ and $u^2(k)$ and one output $\hat{L}(k)$ as Fig. 5. The relation between inputs and estimated cost function can be established by the set of IF–THEN rules such that

''IF $e^2(k)$ is Large and $u^2(k)$ is Large THEN $\hat{L}(k)$ should be Large value.'' (24)

This is a strange forward IF–THEN rule to indicate that the good reward can be obtained when the control system has less tracking error with lower control effort. Thus, the set of IF–THEN rules can be defined as

IF $e^2(k)$ is L and $u^2(k)$ is L THEN $\hat{L}_1(k) = \beta_{L1}(k)\phi_1(k)$,
IF $e^2(k)$ is L and $u^2(k)$ is S THEN $\hat{L}_2(k) = \beta_{L2}(k)\phi_2(k)$,
IF $e^2(k)$ is L and $u^2(k)$ is Z THEN $\hat{L}_3(k) = \beta_{L3}(k)\phi_3(k)$,
IF $e^2(k)$ is L and $u^2(k)$ is L THEN $\hat{L}_4(k) = \beta_{S1}(k)\phi_4(k)$,
IF $e^2(k)$ is L and $u^2(k)$ is S THEN $\hat{L}_5(k) = \beta_{S2}(k)\phi_5(k)$,
IF $e^2(k)$ is L and $u^2(k)$ is Z THEN $\hat{L}_6(k) = \beta_{S3}(k)\phi_6(k)$,
IF $e^2(k)$ is L and $u^2(k)$ is L THEN $\hat{L}_7(k) = \beta_{Z1}(k)\phi_7(k)$,
IF $e^2(k)$ is L and $u^2(k)$ is S THEN $\hat{L}_8(k) = \beta_{Z2}(k)\phi_8(k)$,
IF $e^2(k)$ is L and $u^2(k)$ is Z THEN $\hat{L}_9(k) = \beta_{Z3}(k)\phi_9(k)$,

when $\phi_1(k) = \mu_L(e_k^2)\mu_L(u_k^2)$, $\phi_2(k) = \mu_L(e_k^2)\mu_S(u_k^2)$ and so on. The estimated cost function can be obtained as

$$\hat{L}(k) = \sum_{i=1}^{9} \hat{L}_1(k). \tag{25}$$

To simplify, the relation in (25) can be rewritten as

$$\hat{L}(k) = \beta_c^T(k)\phi_c(k), \tag{26}$$

when

$$\beta_c(k) = [\beta_{L1}(k) \quad \beta_{L1}(k) \quad \cdots \quad \beta_{Z3}(k)]^T, \tag{27}$$

and

$$\phi_c(k) = [\phi_1(k) \quad \phi_2(k) \quad \cdots \quad \phi_9(k)]^T. \tag{28}$$

The network architecture of MiFRENc is depicted in Fig. 5. Regarding the universal function approximation of MiFREN, it exists β_c^* such that

$$L(k) = \beta_c^{*T}\phi_c(k) + \varepsilon_c(k), \tag{29}$$

when $\varepsilon_c(k)$ is the approximation error of MiFRENc. By adding and subtracting $\beta_c^{*T}\phi_c(k)$ on the left hand side of (26), thus we obtain

$$\hat{L}(k) = \tilde{\beta}_c^T(k)\phi_c(k) + \beta_c^{*T}\phi_c(k), \tag{30}$$

when $\tilde{\beta}_c(k) = \beta_c^T(k) - \beta_c^*$. Let us define $\Lambda_c(k) = \tilde{\beta}_c^T(k)\phi_c(k)$, thus, the estimated cost function (30) can be rewritten as

$$\hat{L}(k) = \Lambda_c(k) + \beta_c^{*T}\phi_c(k). \tag{31}$$

It us clear that the accuracy of estimated cost function relates on the learning algorithm of weight parameters β. The proposed learning algorithms will be developed in the next section to tune all adjustable parameters inside FRENa and MiFRENc with convergence analysis.

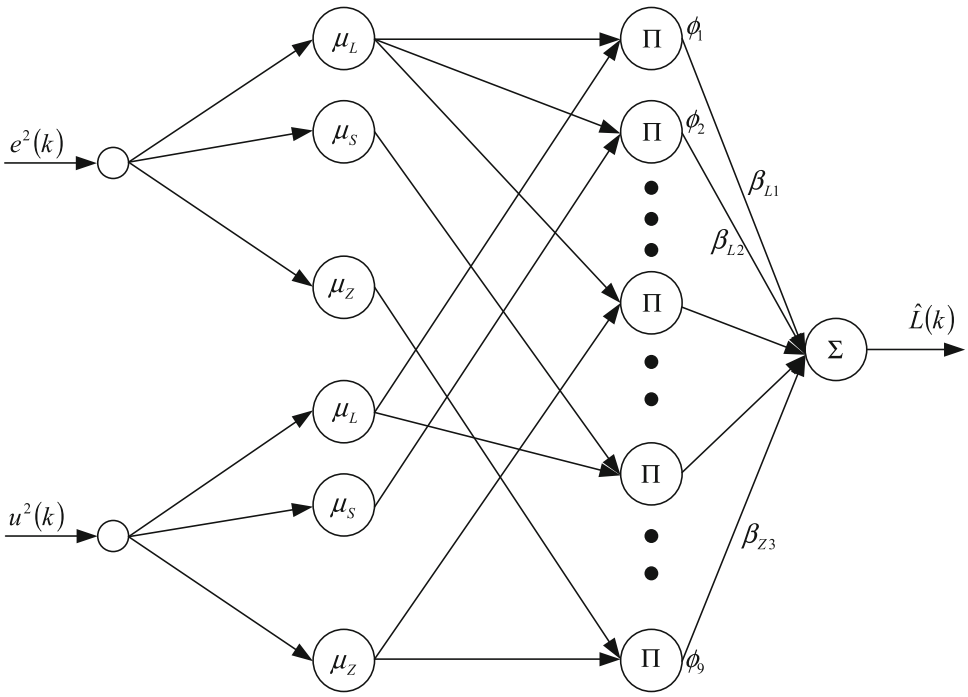

Fig. 5 MiFRENc network architecture

4 Learning algorithms and performance analysis

The learning algorithms are developed for both FRENa and MiFRENc. To improve the computation complexity according to the practical systems point of view, in this work, only the parameters $\beta(k)$ have been tuned by the proposed learning laws. The performance analysis beside of the tracking error and external signals is established by Lyapunov direct method.

4.1 Learning algorithm for FRENa

In this subsection, the learning algorithm is developed for adjustable parameters of FRENa. To avoid the causality problem of $e(k+1)$ in (16), the error function of FRENa is given by $\Lambda_a(k)$ and the estimated function $\hat{L}(k)$ as

$$e_a(k) = \sqrt{g(k)}\Lambda_a(k) + \frac{1}{\sqrt{g(k)}}\hat{L}(k). \tag{32}$$

The cost function of FRENs is given as

$$E_a(k) = \frac{1}{2}e_a^2(k). \tag{33}$$

Based on the gradient reach, the tuning law for β_a is established as

$$\beta_a(k+1) = \beta_a(k) - \eta_a \frac{\partial E_a(k)}{\partial \beta_a(k)}, \tag{34}$$

when η_a denotes as the selected learning rate which will be given next by the main theorem. By using the chain rule, the partial derivative term can be determined as

$$\begin{aligned}\frac{\partial E_a(k)}{\partial \beta_a(k)} &= \frac{\partial E_a(k)}{\partial e_a(k)}\frac{\partial e_a(k)}{\partial \Lambda_a(k)}\frac{\partial \Lambda_a(k)}{\partial \beta_a(k)}, \\ &= e_a(k)\sqrt{g(k)}\phi_a(k).\end{aligned} \tag{35}$$

Substituting (35) into (34) and using $e_a(k)$ in (32), we obtain

$$\begin{aligned}\beta_a(k+1) &= \beta_a(k) - \eta_a[\sqrt{g(k)}\Lambda_a(k) \\ &\quad + \frac{1}{\sqrt{g(k)}}\hat{L}(k)]\sqrt{g(k)}\phi_a(k), \\ &= \beta_a(k) - \eta_a[g(k)\Lambda_a(k) + \hat{L}(k)]\phi_a(k).\end{aligned} \tag{36}$$

Let us recall the error dynamic (16) and consider to neglect the disturbance or $d_a(k) = 0$, thus, we obtain

$$g(k)\Lambda_a(k) = e(k+1). \tag{37}$$

Substituting (37) into (36), the learning law of β_a can be rewritten as

$$\beta_a(k+1) = \beta_a(k) - \eta_a[e(k+1) + \hat{L}(k)]\phi_a(k). \tag{38}$$

The unknown nonlinear function $g(k)$ is completely disappeared in the learning law (38), that allows this algorithm is capable for online learning phase of FRENa with unknown plant's dynamic equations.

4.2 Learning algorithm for MiFRENc

The learning algorithm to tune parameters inside MiFRENc is developed in this subsection. Let us define the error function of MiFRENc as

$$e_c(k) = \delta\hat{L}(k) - \hat{L}(k-1) + l(k), \tag{39}$$

when δ is a positive constant which will be discussed next for the performance analysis. The cost function to be minimized for tuning β_c is given as

$$E_c(k) = \frac{1}{2}e_c^2(k). \tag{40}$$

The learning dynamic of β_a is obtained as

$$\beta_c(k+1) = \beta_c(k) - \eta_c \frac{\partial E_c(k)}{\partial \beta_c(k)}, \tag{41}$$

when η_c denotes as the selected learning rate. By using the chain rule with $E_c(k)$ in (40), $e_c(k)$ in (39) and $\hat{L}(k)$ in (31), the partial derivative term can be obtained as

$$\begin{aligned}\frac{\partial E_c(k)}{\partial \beta_c(k)} &= \frac{\partial E_c(k)}{\partial e_c(k)}\frac{\partial e_c(k)}{\partial \hat{L}(k)}\frac{\partial \hat{L}(k)}{\partial \beta_c(k)}, \\ &= e_c(k)\delta\phi_c(k).\end{aligned} \tag{42}$$

The learning dynamic (41) can be obtained as

$$\beta_c(k+1) = \beta_c(k) - \eta_c e_c(k)\delta\phi_c(k). \tag{43}$$

Recalling $e_c(k)$ in (39) with (43), thus, the learning algorithm for MiFRENc can be rewritten as

$$\beta_c(k+1) = \beta_c(k) - \eta_c\delta[l(k) - \hat{L}(k-1) + \delta\hat{L}(k)]\phi_c(k). \tag{44}$$

This is a practical tuning law which will be used to adjust the parameter β_c as online learning phase.

4.3 Performance analysis

The main theorem is proposed to demonstrate the setting of controller's parameters and learning rates to ensure the closed-loop performance when the tracking error and internal signals are bounded within defined compact sets.

Theorem 4.1 *Consider the nonlinear discrete-time system described by* (1) *and let Assumptions* 1 *and* 2 *be held. Let* d_M, g_M, ε_{cM}, β_{aM} *and* L_M*be existed. Under the control law in* (7) *and learning algorithms in* (38) *and* (44), *it guarantees that the functions* $\Lambda_a(k)$ *and* $\Lambda_c(k)$ *and the tracking*

error $e(k)$ are bounded when designed parameters are appropriately chosen as the followings:

$$\frac{1}{2} < \delta \leq 1, \tag{45}$$

$$0 < \eta_a \leq \frac{g_m}{N_a^2 g_M^2}, \tag{46}$$

and

$$0 < \eta_c \leq \frac{1}{\delta^2 N_c^2}, \tag{47}$$

where N_a *and* N_c *are number of IF–THEN rules of FRENa and MiFRENc, respectively. The boundaries of* $e(k)$, $\Lambda_a(k)$ *and* $\Lambda_c(k)$ *are obtained as* Ω_e, Ω_a *and* Ω_c *when*

$$\Omega_e \doteq \sqrt{\frac{\Xi_M}{\frac{\rho_1}{3} - \frac{\rho_3}{4}p}}. \tag{48}$$

$$\Omega_a \doteq \sqrt{\frac{\Xi_M}{\rho_2 g_m - \rho_1 g_M^2 - \frac{\rho_3}{8}q}}, \tag{49}$$

and

$$\Omega_c \doteq \sqrt{\frac{\Xi_M}{\rho_3 \delta^2 - \rho_4}}, \tag{50}$$

where

$$\Xi_M \doteq \rho_1 d_m^2 + \rho_3 \varepsilon_{cM}^2 + \frac{\rho_3}{8}\beta_{aM}^2 + \left[\frac{\rho_3}{8}(\gamma - 1)^2 + \frac{\rho_2}{g_o}\right] L_M^2. \tag{51}$$

All constants $\rho_1, \rho_2, \ldots, \rho_4$ *are given as*

$$\rho_1 > \frac{3}{4}p\rho_3, \tag{52}$$

$$\rho_2 > \frac{\rho_1 g_M^2 + \frac{\rho_3}{8}q}{g_m}\rho_3, \tag{53}$$

$$\rho_3 > \frac{\rho_4}{\delta^2}, \tag{54}$$

and

$$\rho_4 > \frac{\rho_3}{4}. \tag{55}$$

Remark In this work, the number of IF–THEN rules is given as 7 and 9 rules for FRENa and MiFRENc, respectively. The design of the number of IF–THEN rules is conducted by the computation complexity, and the results of simulation and experimental systems will be discussed by the next section.

Proof By using the Lyapunov direct method, in this work, the candidate function is given as

$$V(k) = \rho_1 e^2(k) + \frac{\rho_2}{\eta_a}\tilde{\beta}_a^T(k)\tilde{\beta}_a(k) + \frac{\rho_3}{\eta_c}\tilde{\beta}_c^T(k)\tilde{\beta}_c(k) + \rho_4 \Lambda_c^2(k-1), \tag{56}$$

or

$$V(k) = V_1(k) + V_2(k) + V_3(k) + V_4(k), \tag{57}$$

when

$$V_1 = \rho_1 e^2(k), \tag{58}$$

$$V_2 = \frac{\rho_2}{\eta_a}\tilde{\beta}_a^T(k)\tilde{\beta}_a(k), \tag{59}$$

$$V_3 = \frac{\rho_3}{\eta_c}\tilde{\beta}_c^T(k)\tilde{\beta}_c(k), \tag{60}$$

and

$$V_4 = \rho_4 \Lambda_c^2(k-1). \tag{61}$$

According to the error dynamic in (16), the change of Lyapunov candidate function $V_1(k)$ can be obtained by

$$\begin{aligned} \Delta V_1(k) &= \rho_1\left[e^2(k+1) - e^2(k)\right], \\ &= \rho_1\left[[g(k)\Lambda_a(k) + d_a(k)]^2 - e^2(k)\right], \\ &\leq \rho_1\left[2g^2(k)\Lambda_a^2(k) + 2d_a^2(k) - e^2(k)\right]. \end{aligned} \tag{62}$$

Applying Assumption 2 and the upper bound of the disturbance and the estimation error as d_m when $|d_a(k)| \leq d_M$: $\forall k = 1, 2, \ldots$, the relation in (62) can be rewritten as

$$\Delta V_1(k) \leq -\rho_1 e^2(k) + 2\rho_1 g_M^2 \Lambda_a^2(k) + 2\rho_1 d_M^2. \tag{63}$$

By using the tuning law in (36), the change of $V_2(k)$ can be expressed as

$$\begin{aligned} \Delta V_2(k) &= \frac{\rho_2}{\eta_a}\left[\tilde{\beta}_a^T(k+1)\tilde{\beta}_a(k+1) - \tilde{\beta}_a^T(k)\tilde{\beta}_a(k)\right], \\ &= \frac{\rho_2}{\eta_a}\Big[\left[\tilde{\beta}_a(k) - \eta_a[g(k)\Lambda_a(k) + \hat{L}(k)]\phi_a(k)\right]^T\left[\tilde{\beta}_a(k) - \eta_a[g(k)\Lambda_a(k) + \hat{L}(k)]\phi(k)\right] - \tilde{\beta}_a^T(k)\tilde{\beta}_a(k)\Big], \\ &= -2\rho_2[g(k)\Lambda_a(k) + \hat{L}(k)]\tilde{\beta}_a^T(k)\phi(k) + \rho_2\eta_a[g(k)\Lambda_a(k) + \hat{L}(k)]^2\phi_a^T(k)\phi(k), \\ &= -2\rho_2\Lambda_a(k)[g(k)\Lambda_a(k)] - 2\rho_2\Lambda_a(k)\hat{L}(k) + \rho_2\eta_a||\phi_a(k)||^2[g(k)\Lambda_a(k) + \hat{L}(k)]^2. \end{aligned} \tag{64}$$

With the lower bound and upper bound of $g(k)$ in (3), the change of $V_2(k)$ (64) can be rewritten as

$$\begin{aligned}\Delta V_2(k) \le & -2\rho_2 g_m \Lambda_a^2(k) - 2\rho_2 \Lambda_a(k)\hat{L}(k) \\ & + \rho_2 \eta_a ||\phi_a(k)||^2 g_M^2 \Lambda_a^2(k) \\ & + \rho_2 \eta_a ||\phi_a(k)||^2 [\hat{L}^2(k) + 2g(k)\Lambda_a(k)\hat{L}(k)], \\ = & \rho_2 \Big[-g_m \Lambda_a^2(k) - (g_m - \eta_a ||\phi_a(k)||^2 g_M^2)\Lambda_a^2(k) \\ & - 2\Lambda_a(k)[I - \eta_a ||\phi_a(k)||^2 g(k)]\hat{L}(k) \\ & + \eta_a ||\phi_a(k)||^2 \hat{L}^2(k)\Big], \\ = & \rho_2 \Big[-g_m \Lambda_a^2(k) - (g_m - \eta_a ||\phi_a(k)||^2 g_M^2)\Big[\Lambda_a^2(k) \\ & + \frac{2\Lambda_a(k)[I - \eta_a ||\phi_a(k)||^2 g(k)]\hat{L}(k)}{g_m - \eta_a ||\phi_a(k)||^2 g_M^2}\Big] \\ & + \eta_a ||\phi_a(k)||^2 \hat{L}^2(k)\Big], \\ = & \rho_2 \Big[-g_m \Lambda_a^2(k) - (g_m - \eta_a ||\phi_a(k)||^2 g_M^2) \\ & \times \Big|\Big|\Lambda_a(k) + \frac{[1 - \eta_a ||\phi_a(k)||^2 g(k)]\hat{L}(k)}{g_m - \eta_a ||\phi_a(k)||^2 g_M^2}\Big|\Big|^2 \\ & + \frac{[1 - \eta_a ||\phi_a(k)||^2 g(k)]^2 \hat{L}^2(k)}{g_m - \eta_a ||\phi_a(k)||^2 g_M^2} \\ & + \eta_a ||\phi_a(k)||^2 \hat{L}^2(k)\Big], \\ = & -\rho_2 g_m \Lambda_a^2(k) - \rho_2 (g_m - \eta_a ||\phi_a(k)||^2 g_M^2) \\ & \times \Big|\Big|\Lambda_a(k) + \frac{[1 - \eta_a ||\phi_a(k)||^2 g(k)]\hat{L}(k)}{g_m - \eta_a ||\phi_a(k)||^2 g_M^2}\Big|\Big|^2 \\ & + \rho_2 \frac{1 - \eta_a ||\phi_a(k)||^2 g_m}{g_m - \eta_a ||\phi_a(k)||^2 g_M^2} \hat{L}^2(k).\end{aligned} \tag{65}$$

It can be simplified as

$$\begin{aligned}\Delta V_2(k) \le & -\rho_2 g_m \Lambda_a^2(k) + \frac{\rho_2}{g_m}\hat{L}^2(k) - \rho_2 (g_m \\ & - \eta_a ||\phi_a(k)||^2 g_M^2)\Big|\Big|\Lambda_a(k) \\ & + \frac{[1 - \eta_a ||\phi_a(k)||^2 g(k)]\hat{L}(k)}{g_m - \eta_a ||\phi_a(k)||^2 g_M^2}\Big|\Big|^2.\end{aligned} \tag{66}$$

Referring the learning law of β_c in (43), the change of $V_3(k)$ can be expressed as

$$\begin{aligned}\Delta V_3(k) = & \frac{\rho_3}{\eta_c}\Big[\tilde{\beta}_c^T(k+1)\tilde{\beta}_c(k+1) - \tilde{\beta}_c^T(k)\tilde{\beta}_c(k)\Big], \\ = & \frac{\rho_3}{\eta_c}\Big[[\tilde{\beta}_c(k) - \eta_c \delta e_c(k)\phi_c(k)]^T[\tilde{\beta}_c(k) \\ & - \eta_c \delta e_c(k)\phi_c(k)] - \tilde{\beta}_c^T(k)\tilde{\beta}_c(k)\Big], \\ = & \frac{\rho_3}{\eta_c}\Big[-2\eta_c \delta e_c(k)\tilde{\beta}_c^T(k)\phi_c(k) \\ & + \eta_c^2 \delta^2 e_c^2(k)||\phi_c(k)||^2\Big], \\ = & -2\rho_3 \delta \Lambda_c(k) e_c(k) \\ & + \rho_3 \eta_c \delta^2 ||\phi_c(k)||^2 e_c^2(k).\end{aligned} \tag{67}$$

By adding and subtracting $\delta L(k)$ and $L(k-1)$ on the left hand side of the error function (39) for MiFRENc, we obtain

$$\begin{aligned}e_c(k) = & \delta[\hat{L}(k) - L(k)] + \delta L(k) - [\hat{L}(k-1) \\ & - L(k-1)] - L(k-1) + l(k), \\ = & \delta[\hat{\beta}_c^T(k)\phi_c(k) - \beta_c^T \phi_c(k) - \varepsilon_c(k)] + \delta L(k) \\ & - L(k-1) + l(k) - [\hat{\beta}_c^T(k-1)F_c(k-1) \\ & - \beta_c^T \phi_c(k-1) - \varepsilon_c(k-1)], \\ = & \delta[\hat{\beta}_c^T(k) - \beta_c^T]\phi_c(k) - [\hat{\beta}_c^T(k-1) \\ & - \beta_c^T]\phi_c(k-1) + \delta L(k) - L(k-1) + l(k) \\ & - \delta\varepsilon_c(k) + \varepsilon_c(k-1), \\ = & \delta\tilde{\beta}_c^T(k)\phi_c(k) - \tilde{\beta}_c^T(k-1)\phi_c(k-1) + \delta L(k) \\ & - L(k-1) + l(k) - \delta\varepsilon_c(k) + \varepsilon_c(k-1).\end{aligned} \tag{68}$$

Regarding to the definition of $\Lambda_c(k)$, the relation in (68) can be rewritten as

$$\begin{aligned}e_c(k) = & \delta\Lambda_c(k) - \Lambda_c(k-1) + \delta L(k) - L(k-1) \\ & + l(k) - \delta\varepsilon_c(k) + \varepsilon_c(k-1).\end{aligned} \tag{69}$$

Let us rearrange (69), thus, we obtain

$$\begin{aligned}\delta\Lambda_c(k) = & e_c(k) - \delta L(k) + \Lambda_c(k-1) + L(k-1) \\ & - l(k) + \delta\varepsilon_c(k) - \varepsilon_c(k-1).\end{aligned} \tag{70}$$

Substitute (70) into (67), thus, we have

$$\begin{aligned}\Delta V_3(k) = & -2\rho_3 e_c(k)\Big[e_c(k) - \delta L(k) + \Lambda_c(k-1) \\ & + L(k-1) - l(k) + \delta\varepsilon_c(k) - \varepsilon_c(k-1)\Big] \\ & + \rho_3 \eta_c \delta^2 ||\phi_c(k)||^2 e_c^2(k), \\ = & -\rho_3\Big[1 - \eta_c \delta^2 ||\phi_c(k)||^2\Big]e_c^2(k) - \rho_3 e_c^2(k) \\ & + 2\rho_3 e_c(k)\Big[\delta L(k) - \Lambda_c(k-1) - L(k-1) \\ & + l(k) - \delta\varepsilon_c(k) + \varepsilon_c(k-1)\Big], \\ = & -\rho_3\Big[1 - \eta_c \delta^2 ||\phi_c(k)||^2\Big]e_c^2(k) \\ & - \rho_3 \delta^2 \Lambda_c^2(k) + \rho_3\Big[\delta L(k) - \Lambda_c(k-1) \\ & - L(k-1) + l(k) - \delta\varepsilon_c(k) + \varepsilon_c(k-1)\Big]^2, \\ \le & -\rho_3\Big[1 - \eta_c \delta^2 ||\phi_c(k)||^2\Big]e_c^2(k) \\ & - \rho_3 \delta^2 \Lambda_c^2(k) + \frac{\rho_3}{4}\Lambda_c^2(k-1) \\ & + \frac{\rho_3}{4} l^2(k) + \frac{\rho_3}{4}[\delta L(k) - L(k-1)]^2 \\ & + \frac{\rho_3}{4}\Big[\delta\varepsilon_c(k) - \varepsilon_c(k-1)\Big]^2.\end{aligned} \tag{71}$$

Let us define the designed parameter δ as $0 < \delta \le 1$ and

recall the local cost function $l(k)$ in (19), thus, the relation in (71) can be obtained as

$$\begin{aligned}\Delta V_3(k) \leq & -\rho_3\left[1-\eta_c\delta^2||F_c(k)||^2\right]e_c^2(k)-\rho_3\delta^2\Lambda_c^2(k)\\ & +\frac{\rho_3}{4}\Lambda_c^2(k-1)+\frac{\rho_3}{4}pe^2(k)\\ & +\frac{\rho_3}{8}q\Lambda_a^2(k)+\frac{\rho_3}{8}||\beta_a^T(k)\phi_a(k)||^2\\ & +\frac{\rho_3}{4}[\delta L(k)-L(k-1)]^2+\rho_3\varepsilon_{cM}^2,\end{aligned} \tag{72}$$

where $|\varepsilon_c(k)| \leq \varepsilon_{cM}^2$. For $V_4(k)$, its first difference can be obtained as

$$\Delta V_4(k)=\rho_4\left[\Lambda_c^2(k)-\Lambda_c^2(k-1)\right]. \tag{73}$$

Finally, the change of Lyapunov function $V(k)$ is obtained as

$$\begin{aligned}\Delta V(k) \leq & -\frac{\rho_1}{3}e^2(k)+\rho_1 g_M^2\Lambda_a^2(k)+\rho_1 d_M^2\\ & -\rho_2 g_m\Lambda_a^2(k)-\rho_2(g_m-\eta_a||\phi_a(k)||^2g_M^2)\\ & \times\left|\left|\Lambda_a(k)+\frac{[1-\eta_a||\phi_a(k)||^2g(k)]L(k)}{g_m-\eta_a||\phi_a(k)||^2g_M^2}\right|\right|^2\\ & +\frac{\rho_2}{g_m}L^2(k)-\rho_3\left[1-\eta_c\delta^2||\phi_c(k)||^2\right]e_c^2(k)\\ & -\rho_3\delta^2\Lambda_c^2(k)+\frac{\rho_3}{4}\Lambda_c^2(k-1)+\frac{\rho_3}{4}pe^2(k)\\ & +\frac{\rho_3}{8}q\Lambda_a^2(k)+\frac{\rho_3}{8}||\beta_a^T\phi_a(k)||^2\\ & +\frac{\rho_3}{4}[\delta L(k)-L(k-1)]^2+\rho_3\varepsilon_{cM}^2\\ & +\rho_4\left[\Lambda_c^2(k)-\Lambda_c^2(k-1)\right],\\ \leq & -\left[\frac{\rho_1}{3}-\frac{\rho_3}{4}p\right]e^2(k)\\ & -\left[\rho_2 g_m-\rho_1 g_M^2-\frac{\rho_3}{8}q\right]\Lambda_a^2(k)\\ & -\left[\rho_3\delta^2-\rho_4\right]\Lambda_c^2(k)-\left[\rho_4-\frac{\rho_3}{4}\right]\Lambda_c^2(k-1)\\ & -\rho_2[g_m-\eta_a||\phi_a(k)||^2g_M^2]\left|\left|\Lambda_a(k)\right.\right.\\ & +\left.\left.\frac{[1-\eta_a||\phi_a(k)||^2g(k)]L(k)}{g_m-\eta_a||\phi_a(k)||^2g_M^2}\right|\right|^2\\ & -\rho_3\left[1-\eta_c\delta^2||\phi_c(k)||^2\right]e_c^2(k)+\Xi_M.\end{aligned} \tag{74}$$

The membership functions of FRENa and MiFRENc are given by (9) and (28), respectively. It is clear that $\phi_a(k)$ and $\phi_c(k)$ are satisfied as the followings

$$0<\phi_a(k)\leq N_a, \tag{75}$$

and

$$0<\phi_c(k)\leq N_c. \tag{76}$$

According to the designed parameters given by (45)–(47), constants ρ_{1-4} satisfied conditions in (52)–(55) and the relations in (75, 76), the change of Lyapunov function can be negative semi-define or $\Delta V(k)\leq 0$ when

$$|e(k)| \geq \sqrt{\frac{\Xi_M}{\frac{\rho_1}{3}-\frac{\rho_3}{4}p}} \doteq \Omega_e, \tag{77}$$

$$|\Lambda_a(k)| \geq \sqrt{\frac{\Xi_M}{\rho_2 g_m-\rho_1 g_M^2-\frac{\rho_3}{8}q}} \doteq \Omega_a, \tag{78}$$

and

$$|\Lambda_c(k)| \geq \sqrt{\frac{\Xi_M}{\rho_3\delta^2-\rho_4}} \doteq \Omega_c. \tag{79}$$

Thus, the existence of the compact sets (48), (79) can be encouraged by (77)–(79), respectively. This proof is completed by the manner of Lyapunov direct method. □

The validation of the proposed control scheme will be presented in the next section for the computer simulation system with a non-affine discrete-time system and the hardware implementation system for DC-motor current control-plant.

5 Validation results

5.1 Simulation results

The following non-affine discrete-time system with output feedback plant is used for simulation:

$$y(k+1)=\sin(y_k)+[5+\cos(y_k u_k)]u_k. \tag{80}$$

The desire trajectory is given as

$$r(k+1)=A_r\sin\left(\omega_r\pi\frac{k}{k_M}\right), \tag{81}$$

where $k_M=4000$ as the maximum time index, $Ar=1.0$, $\omega_r=16$ when $0<k\leq\frac{k_M}{2}$ and $Ar=2.0$, $\omega_r=8$ when $\frac{k_M}{2}<k\leq k_M$. The designed parameter δ is selected as $\delta=0.75$ to follow (45). The learning rate of MiFRENc is designed by (47) as

$$0<\eta_c\leq\frac{1}{\delta^2N_c^2}=\frac{1}{0.75^2 9^2}=0.0219. \tag{82}$$

Thus, we select the learning rate for MiFRENc as $\eta_c=0.02$. For designing the learning rate of FRENa, let us chose the boundaries g_m and g_M as 1 and 2, respectively.

According to (46), the learning rate of FRENa is designed as

$$0<\eta_a \le \frac{g_m}{N_a^2 g_M^2} = \frac{1}{7^2 2^2} = 0.005. \tag{83}$$

Thus, the learning rate for FRENa is given in $\eta_a = 0.0025$. The membership settings of FRENa and MiFRENc are depicted in Figs. 6 and 7, respectively. The setting of membership functions can be desired by the proper ranges of $e(k)$, $e^2(k)$ and $u^2(k)$. In this application, the ranges are given as $[-5, 5]$, [0, 10] and [0, 10] for $e(k)$, $e^2(k)$ and $u^2(k)$, respectively. The initial setting of adjustable parameters $\beta_{\square}(1)$ for FRENa and MiFRENc is given as Table 1.

The tracking performance is presented in Fig. 8 for both the motor current $y(k)$ and the tracking error $e(k)$. The maximum absolute value of tracking error is $|e(k)|_{\max} = 2.4022$ and the average absolute value of tracking error at steady state is 0.0074 when $k = 3000-4000$. Figure 9 displays the control effort $u(k)$, and Fig. 10 illustrates the estimated cost function $\hat{L}(k)$.

5.2 Experimental results

The DC-motor current control system is constructed to validate the performance of control scheme. The desired trajectory is given as

$$r(k+1) = I_r \sin\left(\omega_r \pi \frac{k}{k_M}\right), \tag{84}$$

where $k_M = 2000$ as the maximum time index, $I_r = 15[\text{mA}]$, $\omega_r = 8$ when $0<k\le\frac{k_M}{2}$ and $I_r = 30[\text{mA}]$, $\omega_r = 4$ when $\frac{k_M}{2}<k\le k_M$. The designed parameter δ is selected as $\delta = 0.75$ to follow (45). The learning rate of MiFRENc is designed by (47) as

$$0<\eta_c \le \frac{1}{\delta^2 N_c^2} = \frac{1}{0.75^2 9^2} = 0.0219. \tag{85}$$

Thus, we select the learning rate for MiFRENc as $\eta_c = 0.02$.

Remark The learning rate η_c is selected as the same as simulation case because this learning rate is related only the network architecture of MiFRENc which is same as the previous case.

Regarding to the result in (4), let us chose the boundaries g_m and g_M as 10 and 20, respectively. According to (46), the learning rate of FRENa is designed as

$$0<\eta_a \le \frac{g_m}{N_a^2 g_M^2} = \frac{10}{7^2 20^2} = 0.00051. \tag{86}$$

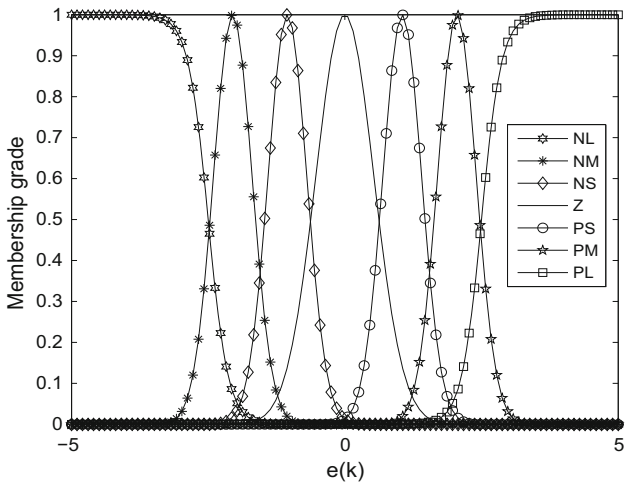

Fig. 6 FRENa membership functions: simulation case

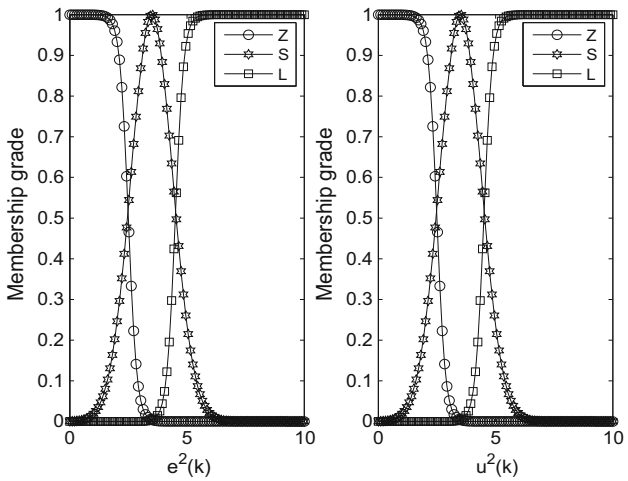

Fig. 7 MiFRENc membership functions: simulation case

Table 1 Initial setting $\beta_{\square}(1)$: simulation case

FRENa		MiFRENc	
Parameter	Value	Parameter	Value
$\beta_{NL}(1)$	− 0.5	$\beta_{L1}(1)$	1
$\beta_{NM}(1)$	− 0.25	$\beta_{L2}(1)$	0.7
$\beta_{NS}(1)$	− 0.15	$\beta_{L3}(1)$	0.6
$\beta_{Z}(1)$	0	$\beta_{S1}(1)$	0.5
$\beta_{PS}(1)$	0.15	$\beta_{S2}(1)$	0.4
$\beta_{PM}(1)$	0.25	$\beta_{S3}(1)$	0.3
$\beta_{PL}(1)$	0.5	$\beta_{Z1}(1)$	0.2
		$\beta_{Z2}(1)$	0.1
		$\beta_{Z3}(1)$	0.1

Thus, we desire to select the learning rate for FRENa as $\eta_a = 0.00025$. It is around half of computation result obtained by (86).

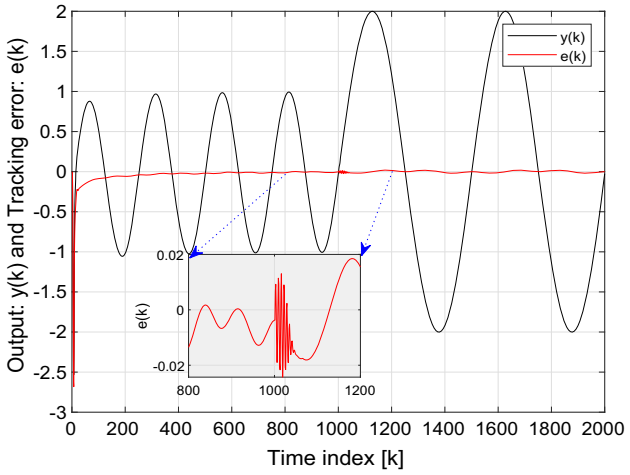

Fig. 8 Tracking performance $y(k)$ and $e(k)$: simulation case

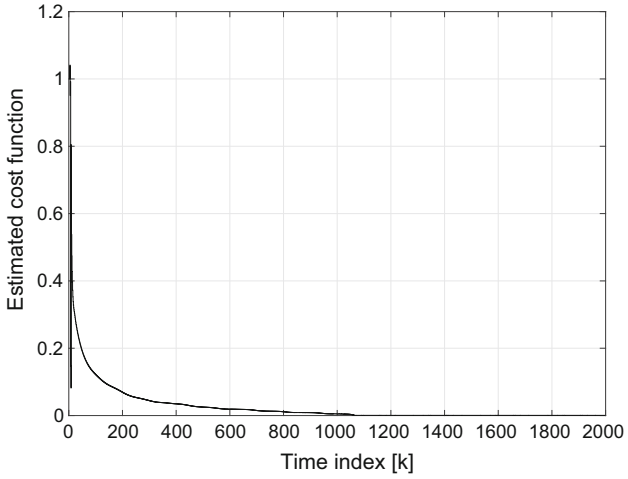

Fig. 10 Estimated cost function $\hat{L}(k)$: simulation case

Remark In this experimental system case, the constants g_m and g_M are selected as 10 times because the relation between output $(y(k) : \pm 50$ [mA]) and input $(u(k) : \pm 5$ [V]) with value ranges is around 10 times without unit.

The membership settings of FRENa and MiFRENc for this experimental system are illustrated in Figs. 11 and 12, respectively when the proper ranges are given in $[-50, 50]$mA, $[0, 10]$mA^2 and $[0, 10]$V^2 for $e(k)$, $e^2(k)$ and $u^2(k)$, respectively. The initial setting of adjustable parameters $\beta_{\square}(1)$ for FRENa and MiFRENc is given as Table 2.

The tracking performance is represented in Fig. 13 for both the motor current $y(k)$ and the tracking error $e(k)$. The maximum absolute value of tracking error is $|e(k)|_{\max} = 78.1642$ [mA] and the average absolute value of tracking error at steady state is 0.4817 [mA] when $k = 1500 - 2000$. Furthermore, the control effort $u(k)$ and the estimated cost function $\hat{L}(k)$ are depicted in Figs. 14 and 15, respectively. In Fig. 13, the large variation of the tracking error is observed. It is caused by the instant back-EMF of the motor. For the compensate of this issue, the controller produces a large variation of the control effort as depicted in Fig. 14. Thus, this phenomenon leads to a second peak of $\hat{L}(k)$ in Fig. 15. The phase plan between $u(k)$ and $e(k)$ is depicted in Fig. 16 to represent the character of a large variation with a clear point of view. Moreover, when the desired trajectory $r(k)$ is changed, the controller provides a higher amplitude of the armature voltage depicted in Fig. 14 that leads to increasing of the cost function (17). Thus, in Fig. 15, the second ripple is detected because of the increasing of the control energy.

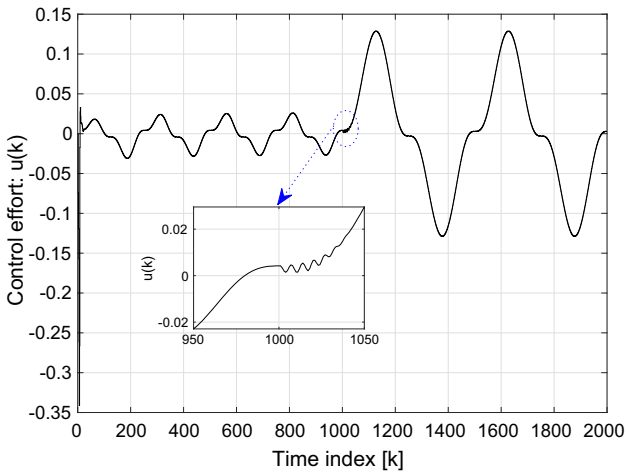

Fig. 9 Control effort $u(k)$: simulation case

To demonstrate the advantage of the proposed RL learning algorithm, the second run is tested when the initial parameters of MiFRENc and FRENa are selected as the final parameters obtained by the first run. For the second run, the large variation is compensated as the results depicted in Fig. 17. The maximum absolute value of tracking error is $|e(k)|_{\max} = 7.391$ [mA] and the average absolute value of tracking error at steady state is 0.2197 [mA] when $k = 1500$–2000. Furthermore, the plot in Fig. 18 indicates the effectiveness of the proposed controller to compensate the large variation occurred in this plant.

6 Conclusions

An adaptive controller for a class of nonlinear discrete-time systems has been proposed by action-critic networks (FRENa and MiFRENc). Practically, the controller has only required the parameter g_M, which has been directly estimated by experimental data, when the mathematical model of controlled plants has been completely omitted.

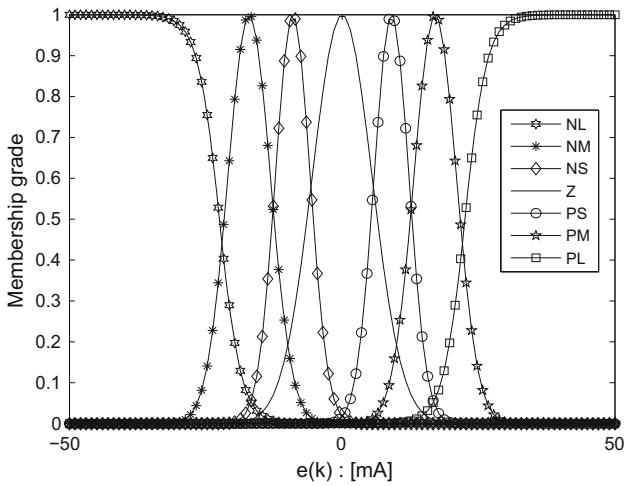

Fig. 11 FRENa membership functions: experimental system case

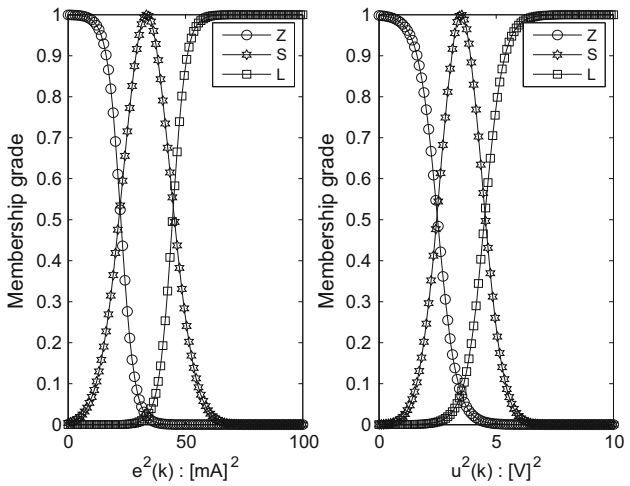

Fig. 12 MiFRENc membership functions: experimental system case

Table 2 Initial setting $\beta_{\square}(1)$: experimental system case

FRENa		MiFRENc	
Parameter	Value	Parameter	Value
$\beta_{NL}(1)$	− 3.5	$\beta_{L1}(1)$	1
$\beta_{NM}(1)$	− 2.25	$\beta_{L2}(1)$	0.7
$\beta_{NS}(1)$	− 1.15	$\beta_{L3}(1)$	0.6
$\beta_{Z}(1)$	0	$\beta_{S1}(1)$	0.5
$\beta_{PS}(1)$	1.15	$\beta_{S2}(1)$	0.4
$\beta_{PM}(1)$	2.25	$\beta_{S3}(1)$	0.3
$\beta_{PL}(1)$	3.5	$\beta_{Z1}(1)$	0.2
		$\beta_{Z2}(1)$	0.2
		$\beta_{Z3}(1)$	0.1

Two sets of IF–THEN rules have been created according to the human knowledge of controlled plant and the optimization manner of tracking error and control energy for

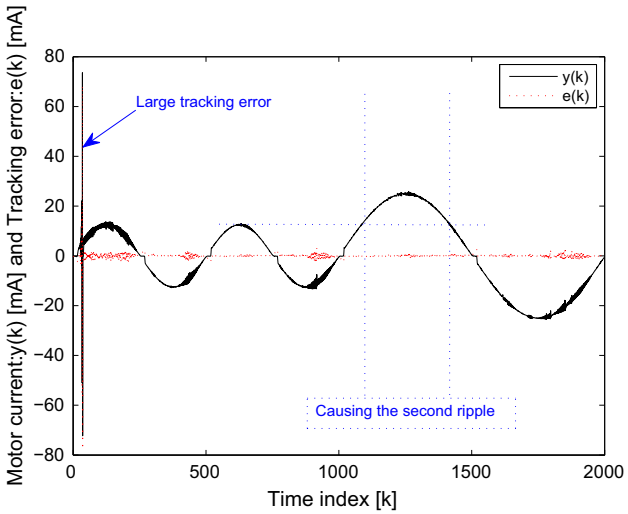

Fig. 13 Tracking performance $y(k)$ and $e(k)$: experimental system

FRENa and MiFRENc, respectively. The online learning algorithm of two networks has been developed to tune all adjustable parameters by RL manner. The theoretical analysis has been conducted by the Lyapunov method to guarantee the convergence of tracking error and internal signals. The numerical system based on computer simulation has demonstrated the effectiveness of the proposed controller and the convergence of error signal. The experimental system with DC-motor current control has been established by our prototyping product. The controller design has been conducted by using only the *V–I* characteristic curve obtained by the standard testing process. The results have represented the satisfied performance of control scheme such that a superior tracking performance and a compensation of large variation occurred by unknown nonlinear terms of controlled plant.

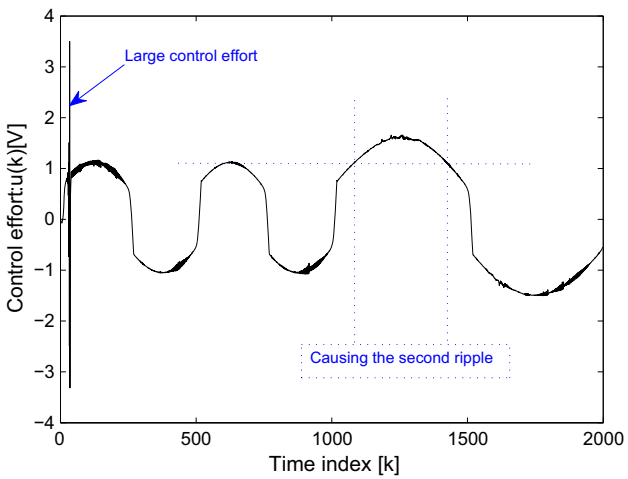

Fig. 14 Control effort $u(k)$: experimental system

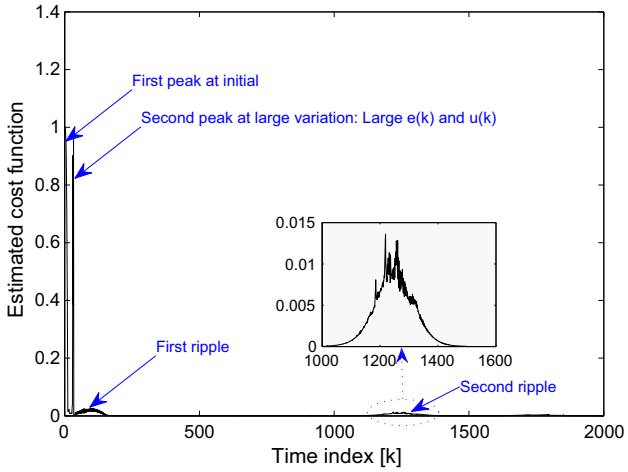

Fig. 15 Estimated cost function $\hat{L}(k)$: experimental system

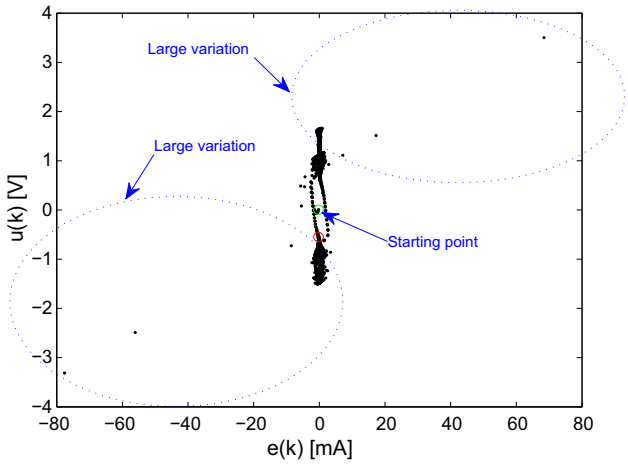

Fig. 16 $u(k)$ and $e(k)$: experimental system

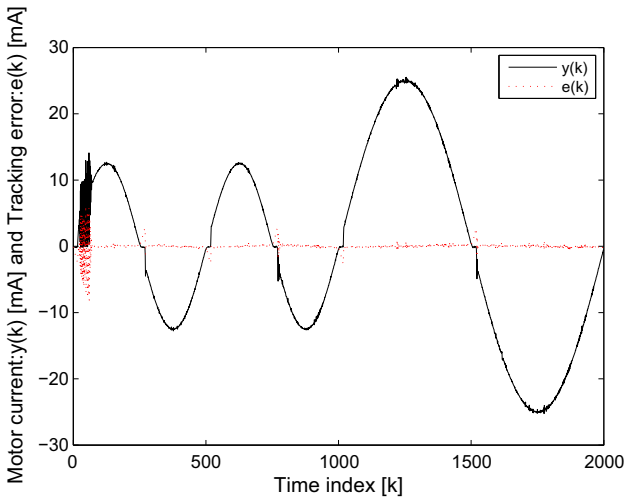

Fig. 17 Tracking performance $y(k)$ and $e(k)$: second run

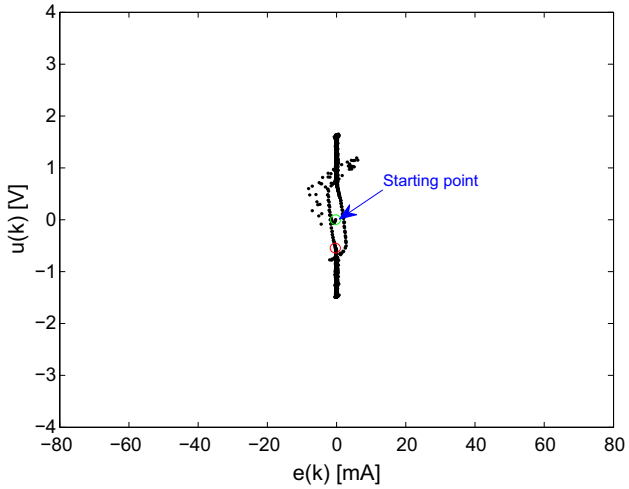

Fig. 18 $u(k)$ and $e(k)$: second run

Unlike other RL controllers, in this work, the critic network has been designed directly by using the set of IF–THEN rules from the human knowledge of the controlled plant. To emphasize this advantage, the research based on nonholonomic systems with this proposed scheme is our future investigating theme.

Acknowledgements This research was supported by Fundamental Research Funds for CINVESTAV-IPN 2017 and Mexican Research Organization CONACyT Grant # 257253.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Hou ZS, Wang Z (2013) From model-based control to data-driven control: survey, classification and perspective. Inf Sci 235:3–35
2. Zhu Y, Hou ZS (2014) Data-driven MFAC for a class of discrete-time nonlinear systems with RBFNN. IEEE Trans Neural Netw Learn Syst 25(5):1013–2014
3. Wang X, Li X, Wang J, Fang X, Zhu X (2016) Data-driven model-free adaptive sliding mode control for the multi degree-of-freedom robotic exoskeleton. Inf Sci 327:246–257
4. Mu C, Zhao Q, Gao Z, Sun C (2019) Q-learning solution for optimal consensus control of discrete-time multiagent systems using reinforcement learning. J Franklin Inst 356:6946–6967
5. He S, Zhang M, Fang1 H, Liu F, Luan X, Ding Z (2019) Reinforcement learning and adaptive optimization of a class of Markov jump systems with completely unknown dynamic information. Neural Comput Appl, pp 1–10. https://doi.org/10.1007/s00521-019-04180-2
6. Kaldmae A, Kotta U (2014) Input output linearization of discrete-time systems by dynamic output feedback. Eur J Control 20:73–78

7. Treesatayapun C (2018) Discrete-time adaptive controller for unfixed and unknown control direction. IEEE Trans Ind Electron 65(7):5367–5375
8. Wang HP, Ghazally IYM, Tian Y (2018) Model-free fractional-order sliding mode control for an active vehicle suspension system. Adv Eng Softw 115:452–461
9. Treesatayapun C (2015) Data input-output adaptive controller based on IF-THEN rules for a class of non-affine discrete-time systems: the robotic plant. J Intell Fuzzy Syst 28:661–668
10. Liu YJ, Tong S (2015) Adaptive NN tracking control of uncertain nonlinear discrete-time systems with nonaffine dead-zone input. IEEE Trans Cybernet 45(3):497–505
11. Zhang CL, Li JM (2015) Adaptive iterative learning control of non-uniform trajectory tracking for strict feedback nonlinear time-varying systems with unknown control direction. Appl Math Model 39:2942–2950
12. Precup RE, Radac MB, Roman RC, Petriu EM (2017) Model-free sliding mode control of nonlinear systems: algorithms and experiments. Inf Sci 381:176–192
13. Zhou Y, Kampen EJ, Chu QP (2018) Incremental model based online dual heuristic programming for nonlinear adaptive control. Control Eng Pract 73:13–25
14. Dong B, Zhou F, Liu K, Li-in Y (2018) Decentralized robust optimal control for modular robot manipulators via critic-identifier structure-based adaptive dynamic programming. Neural Comput Appl, pp 1–18
15. Radac MB, Precup RE (2018) Data-driven model-free slip control of anti-lock braking systems using reinforcement Q-learning. Neurocomputing 275:317–329
16. Yang Q, Jagannathan S (2012) Reinforcement learning controller design for affine nonlinear discrete-time systems using online approximators. IEEE Trans Syst Man Cybern B Cybern 42(2):377–390
17. Wang D, Liu D, Zhao D, Huang Y (2013) A neural-network-based iterative GDHP approach for solving a class of nonlinear optimal control problems with control constraints. Neural Comput Appl 22(2):219–227
18. Kiumarsi B, Lewis FL, Modares H, Karimpour A, Sistani MBN (2014) Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. Automatica 50(4):1167–1175
19. Liu D, Yang X, Li H (2013) Adaptive optimal control for a class of continuous-time affine nonlinear systems with unknown internal dynamics. Neural Comput Appl 23(7–8):1843–1850
20. Lin YC, Chen DD, Chen MS, Chen X, Jia L (2018) A precise BP neural network-based online model predictive control strategy for die forging hydraulic press machine. Neural Comput Appl 29(9):585–596
21. Bertsekas DP, Tsitsiklis JN (1996) Neuro-dynamic programming. Athena Scientific, Cambridge, MA
22. Prokhorov DV, Wunsch DC (1997) Adaptive critic designs. IEEE Trans Neural Netw 8(5):997–1007
23. Liu D, Wang D, Yang X (2013) An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs. Inf Sci 220(20):331–342
24. Zhao B, Liu D, Li Y (2017) Observer based adaptive dynamic programming for fault tolerant control of a class of nonlinear systems. Inf Sci 384:21–33
25. Adhyaru MD, Kar IN, Gopal M (2011) Bounded robust control of nonlinear systems using neural network? Based HJB solution. Neural Comput Appl 20(1):91–103
26. Wei Q, Li B, Song R (2018) Discrete-time stable generalized self-learning optimal control with approximation errors. IEEE Trans Neural Netw Learn Syst 29(4):1226–1238
27. Wei Q, Liu D (2014) Stable iterative adaptive dynamic programming algorithm with approximation errors for discrete-time nonlinear sys. Neural Comput Appl 24:1355–1367
28. Alibekov E, Kubalik J, Babuska R (2016) Policy derivation methods for critic-only reinforcement learning in continuous action spaces. IFAC-PapersOnLine 49:285–290
29. Luo Y, Sun Q, Zhang H, Cui L (2015) Adaptive critic design-based robust neural network control for nonlinear distributed parameter systems with unknown dynamics. Neurocomputing 148:200–208
30. Liang Y, Zhang H, Xiao G, Jiang H (2018) Reinforcement learning-based online adaptive controller design for a class of unknown nonlinear discrete-time systems with time delays. Neural Comput Appl 30:1733–1745
31. Xu H, Zhao Q, Jagannathan S (2015) Finite-horizon near-optimal output feedback neural network control of quantized nonlinear discrete-time systems with input constraint. IEEE Trans Neural Netw Learn Syst 26(8):1776–1788
32. Wei Q, Lewis FL, Sun Q, Yan P, Song R (2017) Discrete-time deterministic Q-learning: a novel convergence analysis. IEEE Trans Cybernet 47(5):1224–1237
33. Wei Q, Song R, Li B, Lin X (2018) A novel policy iteration-based deterministic Q-learning for discrete-time nonlinear systems. In: Self-learning optimal control of nonlinear systems, pp 85–109
34. Liu C (2018) Optimal power management based on Q-learning and neuro-dynamic programming for plug-in hybrid electric vehicles. Ph.D. thesis dissertation, Information Systems Engineering, University of Michigan-Dearborn
35. Navin NK, Sharma R (2017) A fuzzy reinforcement learning approach to thermal unit commitment problem. Neural Comput Appl 31:737–750
36. Tang Y, He H, Ni Z, Zhong X, Zhao D, Xu X (2016) Fuzzy-based goal representation adaptive dynamic programming. IEEE Trans Fuzzy Syst 24(5):1159–1175
37. Sui S, Tong S, Sun K (2018) Adaptive-dynamic-programming-based fuzzy control for triangular structure nonlinear uncertain systems with unknown time delay. Opt Control Appl Methods 39(2):819–834
38. Wang T, Zhang Y, Gao J (2015) Adaptive fuzzy backstepping control for a class of nonlinear systems with sampled and delayed measurements. IEEE Trans Fuzzy Syst 23(2):302–312
39. Chang EC, Wu RC, Zhu K, Chen GY (2018) Adaptive neuro-fuzzy inference system-based grey time-varying sliding mode control for power conditioning applications. Neural Comput Appl 30(3):699–707
40. Khater AA, El-Nagar AM, El-Bardini M, El-Rabaie NM (2019) Online learning based on adaptive learning rate for a class of recurrent fuzzy neural network. Neural Comput Appl, pp 1–20. https://doi.org/10.1007/s00521-019-04372-w
41. Treesatayapun C, Uatrongjit S (2005) Adaptive controller with fuzzy rules emulated structure and its applications. Eng Appl Artif Intell 18:603–615
42. Treesatayapun C (2014) Adaptive control based on IF–THEN rules for grasping force regulation with unknown contact mechanism. Robot Comput Integr Manuf 30:11–18
43. Sahoo A, Xu H, Jagannathan S (2016) Near optimal event-triggered control of nonlinear discrete-time systems using neurodynamic programming. IEEE Trans Neural Netw Learn Syst 27(9):1801–1815