Output Feedback Controller for a Class of Unknown Nonlinear Discrete Time Systems Using Fuzzy Rules Emulated Networks and Reinforcement Learning

C. Treesatayapun

Department of Robotic and Advanced Manufacturing, CINVESTAV-Saltillo, Ramos Arizpe, Mexico

ABSTRACT
A model-free adaptive control for non-affine discrete time systems is developed by utilising the output feedback and action-critic networks. Fuzzy rules emulated network (FREN) is employed as the action network and multi-input version (MiFREN) is implemented as the critic network. Both networks are constructed using human knowledge based on IF–THEN rules according to the controlled plant and the learning laws are established by reinforcement learning without any off-line learning phase. The theoretical derivation of the convergence of the tracking error and internal signal is demonstrated. The numerical simulation and the experimental system are given to validate the proposed scheme.

KEYWORDS
Model-free adaptive control; reinforcement learning; nonlinear discrete time systems; fuzzy neural network; DC motor current control

1. Introduction
Due to the complexity of controlled plants nowadays, it is commonly difficult or impossible to establish its mathematical model especially for the discrete time system [1]. By utilising only input–output data of the controlled plant, the model-free approaches have been developed [2, 3]. On the other hand, the performance of the controllers is related to data’s quality and quantity [4]. For some engineering applications, it is very difficult to access all state variables, thus the output feedback is still a preferable scheme [5, 6]. Furthermore, the close-loop analysis and stability approaches have been proposed [7, 8, 9] to guarantee the performance of controllers. From the engineering point of view, the stability analysis beside of closed-loop’s performance is only a basic minimum requirement even for the artificial intelligence controller [10]. Therefore, the optimal controllers are more desirable for modern applications [11] or by nature view [12].

To ensure the closed-loop performance with the optimisation of the predefined cost function, the schemes based on adaptive dynamic programming have been utilised but the mathematic models have been required for its iterative learning [13, 14]. With the model-free aspects, reinforcement learning (RL) algorithms have been developed to solve optimal control [15, 16] with the estimated solution of the Hamilton–Jacobi–Bellman equation [17, 18]. To mimic the RL process, the approaches based on action-critic networks have been...
derived by artificial neural networks (ANNs) under considering the controlled plant as a black box [19, 20]. Nevertheless, even the mathematic model is unknown but the engineer still has basic human knowledge of the controlled plant such that ‘IF higher output is required THEN more control effort should be supplied’. Thus, the controlled plant can be considered as a grey box.

To integrate the human knowledge as IF–THEN format into the controller, fuzzy logic systems (FLSs) have been utilised in control applications [21] also including the optimal problems [22]. By including the learning ability to FLS, the integrations between FLS and ANN have been developed such as fuzzy neural network (FNN) [23] and fuzzy rules emulated network (FREN) [24, 25]. Thereafter, the approaches of using FNN and FREN for solving the optimal problem with RL have been proposed [26, 27] when the controlled plants have been considered as a class of affine systems. On the other hand, the problem of non-affine systems has been studied in Ref. [28] by the approach of critic-action networks when the state feedback has been utilised for gaining enough information to tune ANNs.

In this work, the output feedback model-free controller is proposed when the control effort is non-affine with respect to system dynamics. The controller is designed by the action network called FRENa with the set of IF–THEN rules according to the controlled plant. Thereafter, the long-term cost function is estimated by the multi-input version of FREN called MiFRENc when IF–THEN rules are established under the general aspect for minimising both tracking error and control energy. The learning laws are derived with the RL approach to tune all adjustable parameters of FRENa and MiFRENc aiming to minimise the tracking error and the estimated cost function. Furthermore, the closed-loop analysis is provided by the Lyapunov method to demonstrate the convergence of the tracking error and internal signals.

This paper is organised as follows. Section 2 introduces a class of systems under our investigation and problem formulation. The proposed scheme is introduced in Section 3 including the network architectures with IF–THEN rules of FRENa and MiFRENc and their formulations. The learning laws and closed-loop analysis are derived in Section 4. Section 5 provides the results of the simulation and experimental system.

## 2. Controlled Plant as a Class of Nonlinear Discrete-Time Systems

In this work, the controlled plant for a class of non-affine discrete time systems is considered as

\[ y(k+1) = f(y(k), \ldots, y(k-n_y), u(k), \ldots, u(k-n_u)) + d(k), \]

where \( y(k+1) \in \mathbb{R} \) is the plant’s output with respect to the control effort \( u(k) \in \mathbb{R} \), \( f(\cdot) \) is an unknown nonlinear function, \( n_u \) and \( n_y \) are unknown system orders and \( d(k) \) denotes a bounded disturbance such that \(|d(k)| \leq d_M\). For further analysis, the following assumptions are expressed according to the unknown nonlinear function \( f(\cdot) \) with respect to the control effort \( u(k) \).

**Assumption 2.1:** The derivative of \( y(k+1) \) with respect to \( u(k) \) is existed and bounded such that

\[ 0 < g_m \leq \frac{\partial y(k+1)}{\partial u(k)} \leq g_M. \]
where \( g_m \) and \( g_M \) are positive constants.

**Remark 2.2:** The condition mentioned in (2) indicates that the controlled plant in (1) is a positive control direction. That will assist the setting of IF–THEN rules according to the change of control effort \( \Delta u(k) \) altogether with the change of output \( \Delta y(k + 1) \).

Referring to condition (2), it is clear that the change of output \( \Delta y(k + 1) \) with respect to the change of control effort \( \Delta u(k) \) can be rewritten as

\[
g_m^d \leq \frac{\Delta y(k + 1)}{\Delta u(k)} \leq g_M^d
\]

where \( \Delta u(k) > 0 \) and \( g_m^d \) and \( g_M^d \) are constants according to \( g_m \) and \( g_M \), respectively. This will lead to the setting of IF–THEN rules such that

- IF \( \Delta u(k) \) is positive-large, THEN \( \Delta y(k + 1) \) should be positive-large
- or
- IF \( \Delta u(k) \) is negative small, THEN \( \Delta y(k + 1) \) should be negative small.

By utilising those IF–THEN rules, the adaptive controller based on FRENs will be established in the next section.

### 3. RL Controller

The proposed controller is illustrated by the block diagram in Figure 1. In this work, the plant is selected as a DC motor current control. Only the armature current is measured as the output \( y(k + 1) \) (mA) when the control effort \( u(k) \) (V) is the voltage fed to the driver unit. Thus, the IF–THEN rules mentioned in Section 2 can be rewritten according to the physical nature such that

- IF we apply *positive-large change of control voltage* \( [\Delta u(k)] \), THEN we should have *positive-large change of armature current* \( [\Delta y(k + 1)] \).

![Figure 1. Closed-loop system architecture.](image-url)
According to this knowledge, the action network (FRENa) is first established to generate the control effort $y(k)$ when the input is the tracking error $e(k)$ defined as

$$e(k) = r(k) - y(k),$$

where $r(k)$ is the desired trajectory. Second, the critic network is designed using MiFRENc to produce the estimated long-term cost function $\hat{L}(k)$ for the controller FRENa. The details of two networks and its IF–THEN rules are given as follows.

### 3.1. Controller or Action Network

To utilise the action network, the IF–THEN rules with the relation between the tracking error $e(k)$ and the control effort $u(k)$ are first established. By considering the basic knowledge such that, positive-large $e(k)$ means lack of $y(k)$ in positive-large. In order to compensate, it clearly requires that the control effort $u(k)$ should be positive-large. For conclusion, we have IF $e(k)$ is positive-large, THEN $u(k)$ should be positive-large. With seven linguistic levels, it leads to the design of IF–THEN rules as

- IF $e(k)$ is NL THEN $u(k)$ is NL,
- IF $e(k)$ is NM THEN $u(k)$ is NM,
- IF $e(k)$ is NS THEN $u(k)$ is NS,
- IF $e(k)$ is Z THEN $u(k)$ is Z,
- IF $e(k)$ is PS THEN $u(k)$ is PS,
- IF $e(k)$ is PM THEN $u(k)$ is PM,
- IF $e(k)$ is PL THEN $u(k)$ is PL,

where notations of linguistic variables N, P, L, M, S and Z denote negative, positive, large, medium, small and zero, respectively.

Employing this set of IF–THEN rules, the network architecture of FRENa is illustrated by Figure 2. According to the network architecture in Figure 2 and the function formulation of FREN in Ref. [24], the control effort $u(k)$ is determined by

$$u(k) = \beta_a^T(k)\phi_a(k),$$

where

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

and

$$\beta_a(k) = [\beta_{NL}^a(k) \quad \beta_{NM}^a(k) \quad \cdots \quad \beta_{PL}^a(k)]^T.$$

Let us consider FRENa as the function estimator of the unknown control effort, thus it exists the ideal control effort $u^*(k)$ with the ideal parameter $\beta_a^*$ such that

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

where $\varepsilon_a(k)$ is a bounded residual error $|\varepsilon_a(k)| \leq \varepsilon_aM$.

By using the dynamics (1) with the control laws (5) and (8), the tracking error $e(k + 1)$ is rearranged as

$$e(k + 1) = r(k + 1) - y(k + 1)$$
Recalling Assumption 1 and using mean value theorem, the error dynamic (9) can be rewritten as

\[ e(k+1) = \frac{\partial f(x)}{\partial x} \bigg|_{x=u^m(k)} [u^*(k) - u(k)] - d(k), \]  

where

\[ g(k) = \frac{\partial f(u^m(k))}{\partial u^m(k)}, \]  

and \( u^m(k) \in [\min\{u_k^*, u_k\}, \max\{u_k^*, u_k\}] \). Employing the control laws (8) and (5), it yields

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

Let us define \( \tilde{\beta}_a(k) = \beta_a^* - \beta_a(k), d_a(k) = g(k)\varepsilon_a(k) - d(k) \) and

\[ \Lambda_a(k) = \tilde{\beta}_a^T(k) \phi_a(k), \]  

and we obtain

\[ e(k+1) = g(k)\Lambda_a(k) + d_a(k). \]
It is worth to note that the tracking error obtained by (14) is functional by \( \tilde{\beta}_a(k) \) and the unknown but bounded \( d_a(k) \) such that \(|d_a(k)| \leq d_aM\). This relation will be used for the performance analysis afterward.

### 3.2. Estimated Cost–Function or Critic Network

In this work, the long-term cost function \( L(k) \) is employed by an infinite-horizon of the tracking error \( e(k) \) and the control effort \( u(k) \) with the discount factor \( \gamma_L \) as

\[
L(k) = \sum_{i=k}^{\infty} \gamma_L^{i-k} l(i), \tag{15}
\]

where

\[
l(k) = p e^2(k) + q u^2(k), \tag{16}
\]

where \( p \) and \( q \) are positive constants and \( 0 < \gamma_L \leq 1 \).

\( L(k) \) in (15) is functional by two input arguments with the quadratic functions \( (f_x = x^2) \) of \( e(k) \) and \( u(k) \). Thus, an adaptive network MiFRENc is utilised to estimate \( L(k) \) as the block diagram in Figure 1. In order to design MiFRENc, IF–THEN rules are first established by Table 1. Thereafter, the network architecture of MiFRENc is illustrated by Figure 3. By utilising the network in Figure 3 and results in Ref. [24], the estimated cost function \( \hat{L}(k) \) is determined by

\[
\hat{L}(k) = \hat{\beta}_c^T(k) \phi_c(k), \tag{17}
\]

where

\[
\beta_c(k) = [\beta_{ZZ}(k) \quad \beta_{ZS}(k) \quad \cdots \quad \beta_{LL}(k)]^T \tag{18}
\]

and

\[
\phi_c(k) = [\phi_1(k) \quad \phi_2(k) \quad \cdots \quad \phi_9(k)]^T. \tag{19}
\]

Using the universal approximation property of MiFREN [24], there exists an ideal parameter \( \beta^*_c \) such that

\[
L(k) = \beta^*_c^T \phi_c(k) + \varepsilon_c(k), \tag{20}
\]

where \( \varepsilon_c(k) \) is a bounded residual error such that \(|\varepsilon_c(k)| \leq \varepsilon_cM\). Adding and subtracting \( \beta^*_c^T \phi_c(k) \) on the left-hand side of (17) yields

\[
\hat{L}(k) = \tilde{\beta}_c^T(k) \phi_c(k) + \beta^*_c^T \phi_c(k) = \Lambda_c(k) + \beta^*_c^T \phi_c(k) \tag{21}
\]

where \( \tilde{\beta}_c(k) = \beta^T_c(k) - \beta^*_c \) and \( \Lambda_c(k) = \tilde{\beta}_c^T(k) \phi_c(k) \).

### Table 1. MiFRENc: IF–THEN rules.

| \( \hat{L}(k) \) | \( e^2(k) \) | \( u^2(k) \) |
|------------------|------------------|------------------|
| \( Z \) | \( Z \) \( S \) | \( \phi_1; Z \) \( \phi_4; S \) | \( \phi_7; Z \) \( \phi_8; S \) |
| \( S \) | \( Z \) \( S \) | \( \phi_2; Z \) \( \phi_5; S \) | \( \phi_9; S \) |
| \( L \) | \( Z \) \( S \) | \( \phi_3; Z \) \( \phi_6; S \) | \( \phi_9; S \) |
In order to improve the performance of FRENa and MiFRENc, the learning laws will be developed and explained in the next section.

4. Learning Algorithms and Performance Analysis

4.1. Action Network Learning Law

Considering the tracking error within $\Lambda_a(k)$ as (14) and the estimated cost function $\hat{L}(k)$, in this work, the error function of action network is given as

$$e_a(k) = \sqrt{g(k)}\Lambda_a(k) + \frac{1}{\sqrt{g(k)}}\hat{L}(k).$$  \hspace{1cm} (22)

Thereafter, the cost function to be minimised is utilised as

$$E_a(k) = \frac{1}{2}e_a^2(k).$$  \hspace{1cm} (23)

Applying the gradient descent, the tuning law for $\beta_a$ is derived as

$$\beta_a(k+1) = \beta_a(k) - \eta_a \frac{\partial E_a(k)}{\partial \beta_a(k)},$$  \hspace{1cm} (24)

where $\eta_a$ is the learning rate. By using the chain rule and (13), it yields

$$\frac{\partial E_a(k)}{\partial \beta_a(k)} = \frac{\partial E_a(k)}{\partial e_a(k)} \frac{\partial e_a(k)}{\partial \Lambda_a(k)} \frac{\partial \Lambda_a(k)}{\partial \beta_a(k)}$$

$$= -e_a(k)\sqrt{g(k)}\phi_a(k).$$  \hspace{1cm} (25)
Recalling (24) with (25) and using $e_a(k)$ in (22), it leads to

$$\beta_a(k+1) = \beta_a(k) + \eta_a e_a(k) \sqrt{g(k)} \phi_a(k)$$

$$= \beta_a(k) + \eta_a \left[ \sqrt{g(k)} \Lambda_a(k) + \frac{1}{\sqrt{g(k)}} \hat{L}(k) \right] \sqrt{g(k)} \phi_a(k)$$

$$= \beta_a(k) + \eta_a [g(k) \Lambda_a(k) + \hat{L}(k)] \phi_a(k).$$  \hfill (26)

By eliminating $d_a(k)$ in (14), the learning law (26) is rewritten as

$$\beta_a(k+1) = \beta_a(k) + \eta_a [e(k+1) + \hat{L}(k)] \phi_a(k).$$ \hfill (27)

The final learning law of FRENa given by (27) is a practical one because all parameters required on the left-hand side are certainly obtained at the time index $k+1$.

### 4.2. Critic Network Learning Law

In general, the error function of critic networks is employed by the estimated cost function $\hat{L}(k)$. Therefore, in this work, the error function $e_c(k)$ is given as

$$e_c(k) = \delta \hat{L}(k) - \hat{L}(k-1) + l(k),$$\hfill (28)

where $\delta$ is a positive constant. In order to tune $\beta_c$, the cost function $E_c(k)$ is defined as

$$E_c(k) = \frac{1}{2} e_c^2(k).$$ \hfill (29)

Applying the gradient descent at (29) with respect to $\beta_c(k)$, we have

$$\beta_c(k+1) = \beta_c(k) - \eta_c \frac{\partial E_c(k)}{\partial \beta_c(k)},$$ \hfill (30)

where $\eta_c$ is the learning rate. Using the chain rule along $E_c(k)$ in (29), $e_c(k)$ in (28) and $\hat{L}(k)$ in (17), it yields

$$\frac{\partial E_c(k)}{\partial \beta_c(k)} = \frac{\partial E_c(k)}{\partial e_c(k)} \frac{\partial e_c(k)}{\partial \hat{L}(k)} \frac{\partial \hat{L}(k)}{\partial \beta_c(k)}$$

$$= e_c(k) \delta \phi_c(k).$$ \hfill (31)

Rewriting (30) with (31), it leads to

$$\beta_c(k+1) = \beta_c(k) - \eta_c e_c(k) \delta \phi_c(k)$$

$$= \beta_c(k) - \eta_c \delta [l(k) - \hat{L}(k-1)] + \delta \hat{L}(k) \phi_c(k).$$ \hfill (32)

Finally, we have a practical tuning law for MiFRENc.
Figure 4. FRENa membership functions: simulation case.

Figure 5. MiFRENc membership functions: simulation case.
Table 2. Initial setting $\beta_{\square}(1)$: simulation system.

| Parameter | Value | Parameter | Value |
|-----------|-------|-----------|-------|
| $\beta_{NN}(1)$ | $-0.25$ | $\beta_{ZZ}(1)$ | $0$ |
| $\beta_{NM}(1)$ | $-0.15$ | $\beta_{ZL}(1)$ | $0$ |
| $\beta_{NS}(1)$ | $0.05$ | $\beta_{ZS}(1)$ | $0.1$ |
| $\beta_{Z}(1)$ | $0$ | $\beta_{PS}(1)$ | $0.3$ |
| $\beta_{PS}(1)$ | $0.05$ | $\beta_{SZ}(1)$ | $0.4$ |
| $\beta_{PM}(1)$ | $0.15$ | $\beta_{SS}(1)$ | $0.5$ |
| $\beta_{PL}(1)$ | $0.25$ | $\beta_{SS}(1)$ | $0.6$ |
| $\beta_{PL}(1)$ | $0.25$ | $\beta_{LS}(1)$ | $0.7$ |

Figure 6. Tracking performance $y(k)$ and $e(k)$: simulation system.

4.3. Closed-Loop Analysis

In the following theorem, the closed-loop performance of the output feedback controller is demonstrated while the tracking error and internal signals are bounded.

Theorem 4.1: For the non-affine discrete time system mentioned in Section 2, the performance of the closed-loop system configured by the structure of FRENa and MiFREnc in Section 3 is guaranteed in terms of the bonded tracking error and internal signals when the designed parameters are selected as follows:

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

$$0 < \eta_a \leq \frac{g_m}{\nu_a g_M^2}$$
Figure 7. Control effort $u(k)$: simulation system.

Figure 8. Estimated cost function $\hat{L}(k)$: simulation system.
Figure 9. $u(k)$ and $e(k)$: simulation system.

Figure 10. FRENa membership functions: experimental system.
Figure 11. MiFRENc membership functions: experimental system.

Table 3. Initial setting $\beta_{\square}(1)$: experimental system.

| FRENa Parameter | Value | MiFRENc Parameter | Value |
|-----------------|-------|-------------------|-------|
| $\beta_{NL}(1)$ | $-3.25$ | $\beta_{Z}(1)$ | $0$ |
| $\beta_{NM}(1)$ | $-2.5$ | $\beta_{ZS}(1)$ | $0.1$ |
| $\beta_{NS}(1)$ | $-1.05$ | $\beta_{ZL}(1)$ | $0.2$ |
| $\beta_{Z}(1)$ | $0$ | $\beta_{ZS}(1)$ | $0.3$ |
| $\beta_{PS}(1)$ | $1.05$ | $\beta_{ZL}(1)$ | $0.4$ |
| $\beta_{PM}(1)$ | $2.5$ | $\beta_{PL}(1)$ | $0.5$ |
| $\beta_{PL}(1)$ | $3.25$ | $\beta_{L}(1)$ | $0.6$ |
| $\beta_{LM}(1)$ | $2.5$ | $\beta_{L}(1)$ | $0.7$ |
| $\beta_{LM}(1)$ | $3.25$ | $\beta_{L}(1)$ | $0.8$ |

and

$$0 < \eta_c \leq \frac{1}{\nu_{c} \delta^2},$$

(35)

where $\nu_{a}$ and $\nu_{c}$ are upper limits of $||\phi_a(k)||^2$ and $||\phi_c(k)||^2$, respectively.

Proof: The proof is given in Appendix.

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. Simulation and Experimental Systems

5.1. Simulation System and Results

The controller developed in this work is first implemented on the nonlinear discrete time given as

\[ y(k + 1) = \sin(y_k) + [5 + \cos(y_ku_k)]u(k). \] (36)

It is worth to mention that the mathematic model in (36) is used only to establish the simulation. In this test, the desired trajectory is given as

\[ r(k + 1) = A_r \sin\left(\frac{\omega_r \pi k}{k_M}\right), \] (37)

where \( k_M = 500 \) as the maximum time index, \( A_r = 1.0 \) and \( \omega_r = 8. \) To follow (33), \( \delta \) is selected as \( \delta = 0.75 \) and \( \nu_a = \nu_c = 1.5. \) By using this setting and (35), the learning rate of MiFRENc is determined as

\[ 0 < \eta_c \leq \frac{1}{\delta^2 \nu_c^2} = \frac{1}{0.75^2 1.5^2} = 0.7901. \] (38)

In this case, the learning rate for MiFRENc is selected as \( \eta_c = 0.5. \) To select the learning rate of FRENa, let us chose \( g_m \) and \( g_M \) as 1 and 6, respectively. By using (34), the learning rate of FRENa is determined as

\[ 0 < \eta_a \leq \frac{g_m}{\nu_a^2 g_M^2} = \frac{1}{1.5^2 6^2} = 0.0123. \] (39)

Thus, the learning rate for FRENa is selected as \( \eta_a = 0.01. \)

Figures 4 and 5 illustrate the setting of membership functions for FRENa and MiFRENc, respectively. The initial setting of adjustable parameters \( \beta_{\square}(1) \) for FRENa and MiFRENc is given as Table 2.

Figure 6 displays the tracking performance with both plots of \( y(k) \) and \( e(k) \) and Figure 7 represents the control effort \( u(k). \) The estimated cost function \( \hat{L}(k) \) is illustrated in Figure 8. The phase plane trajectory of \( u(k) \) and \( e(k) \) is depicted in Figure 9 to demonstrate the closed-loop system’s behaviour.

5.2. Experimental System and Results

The experimental system is constructed by a DC motor current control. The output \( y(k + 1) \) is the armature current (mA) and the input \( u(k) \) is the control voltage applied to the driver circuit depicted in Figure 1. Same as the simulation systems, let us select \( \delta = 0.75, \nu_a = \nu_c = 1.5, g_m = 5 \) and \( g_M = 10. \) Thus, the learning rate of FRENa is designed as

\[ 0 < \eta_a \leq \frac{g_m}{\nu_a^2 g_M^2} = \frac{5}{1.5^2 10^2} = 0.0222. \] (40)

In this case, we select \( \eta_a = 0.01. \) For MiFRENc, we use the same learning rate as the simulation system such that \( \eta_c = 0.5 \) because of the same network architecture. The desired
Figure 12. Tracking performance $y(k)$ and $e(k)$: experimental system.

Figure 13. Control effort $u(k)$: experimental system.
Figure 14. Estimated cost function $\hat{L}(k)$: experimental system.

Figure 15. $u(k)$ and $e(k)$: experimental system.
Figure 16. Pulse response: experimental system.

The trajectory is given as

\[ r(k + 1) = I_r \sin \left( \omega_r \pi \frac{k}{k_M} \right), \]  

(41)

where

\[ I_r = \begin{cases} 15 \text{ (mA)} & \text{if } 0 \leq k < \frac{k_M}{2}, \\ 30 \text{ (mA)} & \text{otherwise,} \end{cases} \]  

(42)

\[ \omega_r = \begin{cases} 8 & \text{if } 0 \leq k < \frac{k_M}{2}, \\ 4 & \text{otherwise}, \end{cases} \]  

(43)

and \( k_M = 2000 \).

Figures 10 and 11 represent the setting of membership functions of FRENa and MiFRENc, respectively. All adjustable parameters \( \beta_{\square}(1) \) for FRENa and MiFRENc are initialised as the setting in Table 3.

Figure 12 displays the motor current \( y(k) \) and the tracking error \( e(k) \) to demonstrate the performance of the closed-loop system. The maximum absolute value of tracking error is \( |e(k)|_{\text{max}} = 48.2936 \) (mA) and the average absolute value of tracking error at steady state is 0.4924 (mA) when \( k = 1500–2000 \). Figure 13 shows the control effort \( u(k) \). The estimated cost function \( \hat{L}(k) \) is illustrated in Figure 14. The phase plane trajectory of \( u(k) \) and \( e(k) \) is plotted in Figure 15. Thus, the large variation is detected because of the back-EMF. In order to evaluate the proposed scheme working under the situation of back-EMF, the pulse-train trajectory is implemented with the response displayed in Figure 16. It is clear that the effect of back-EMF is eliminated within the second pulse (B).

6. Conclusions

A model-free adaptive control for a class of non-affine discrete time systems has been developed by RL. The closed-loop system has been established by the output feedback with two
adaptive networks FRENa and MiFRENc. The initial settings of FRENa and MiFRENc have been conducted according to the human knowledge of the controlled plant within the format of IF–THEN rules. The performance has been enchanted by the learning laws for both FRENa and MiFRENc while the tracking error and internal signals have been guaranteed the convergence over the reasonable compact sets. The numerical system and experimental results have represented to verify theoretical conjecture.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work has been supported by Fundamental Research Funds for CINVESTAV-IPN and Mexican Research Organization CONACyT [grant number 257253].

Notes on contributor

C. Treesatayapun received the Ph.D. in electrical engineering from Chiang-Mai University, Thailand, in 2004. He was a production engineer at SAGA Electronics (JRC-NJR) from 1998-2000 and was a head of electrical engineering program at North Chiang-Mai University, Thailand from 2001-2007. He is currently a senior researcher at Department of robotic and advanced manufacturing, Mexican Research Center and Advanced Technology, CINVESTAV-IPN, Saltillo campus, Mexico. His current research interests include automation and robotic system control and optimization, adaptive and learning algorithms and electric machine drives.

ORCID

C. Treesatayapun http://orcid.org/0000-0002-8574-672X

References

[1] Hou ZS, Wang Z. From model-based control to data-driven control: survey, classification and perspective. Inf Sci. 2013;235:3–35.
[2] Zhu Y, Hou ZS. Data-driven MFAC for a class of discrete-time nonlinear systems with RBFNN. IEEE Trans Neural Netw Learn Syst. 2014;25(5):1013–1020.
[3] Wang X, Li X, Wang J, et al. Data-driven model-free adaptive sliding mode control for the multi degree-of freedom robotic exoskeleton. Inf Sci. 2016;327:246–257.
[4] Lin N, Chi R, Huang B. Data-driven recursive least squares methods for non-affined nonlinear discrete-time systems. Appl Math Modell. 2020;81:787–798.
[5] Kaldmae A, Kotta U. Input–output linearization of discrete-time systems by dynamic output feedback. Eur J Control. 2014;20:73–78.
[6] Treesatayapun C. 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. 2015;28:661–668.
[7] Liu YJ, Tong S. Adaptive NN tracking control of uncertain nonlinear discrete-time systems with nonaffine dead-zone input. IEEE Trans Cybern. 2015;45(3):497–505.
[8] Zhang CL, Li JM. Adaptive iterative learning control of non-uniform trajectory tracking for strict feedback nonlinear time-varying systems with unknown control direction. Appl Math Model. 2015;39:2942–2950.
[9] Precup RE, Radac MB, Roman RC, et al. Model-free sliding mode control of nonlinear systems: algorithms and experiments. Inf Sci. 2017;381:176–192.
[10] Raj R, Mohan BM. Stability analysis of general Takagi–Sugeno fuzzy two-term controllers. Fuzzy Inf Eng. 2018;10(2):196–212.
[11] Zhang X, Zhang HG, Sun QY, et al. Adaptive dynamic programming-based optimal control of unknown nonaffine nonlinear discrete-time systems with proof of convergence. Neurocomputing, 2012;35:48–55.

[12] Eftekhari M, Zeinalkhani M. Extracting interpretable fuzzy models for nonlinear systems using gradient-based continuous ant colony optimization. Fuzzy Inf Eng, 2013;5(3):255–277.

[13] Liu D, Wang D, Yang X. An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs. Inf Sci, 2013;220(20):331–342.

[14] Jiang H, Zhang H. Iterative ADP learning algorithms for discrete-time multi-player games. Artif Intell Rev, 2018;50(1):75–91.

[15] Liu D, Wang D, Zhao D, et al. Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming. IEEE Trans Autom Sci Eng, 2012;9(3):628–634.

[16] Kiumarsi B, Lewis FL, Modares H, et al. Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. Automatica, 2014;50(4):1167–1175.

[17] Yang Q, Jagannathan S. Reinforcement learning controller design for affine nonlinear discrete-time systems using online approximators. IEEE Trans Syst Man Cybern B Cybern, 2012;42(2):377–390.

[18] Ha M, Wang D, Liu D. Event-triggered constrained control with DHP implementation for non-affine discrete-time systems. Inf Sci, 2020;519:110–123.

[19] Xu B, Yang C, Shi Z. Reinforcement learning output feedback NN control using deterministic learning technique. IEEE Trans Neural Netw Learn Syst, 2014;25(3):635–641.

[20] Liu YJ, Li S, Tong S, et al. Adaptive reinforcement learning control based on neural approximation for nonlinear discrete-time systems with unknown nonaffine dead-zone input. IEEE Trans Neural Netw Learn Syst, 2019;30(1):295–305.

[21] Allam E, Elbab HF, Hady MA, et al. Vibration control of active vehicle suspension system using fuzzy logic algorithm. Fuzzy Inf Eng, 2010;24(4):361–387.

[22] Niftiyev AA, Zeynalov CI, Poormanuchehri M. Fuzzy optimal control problem with non-linear functional. Fuzzy Inf Eng, 2011;3(3):311–320.

[23] Fei J, Wang T. Adaptive fuzzy-neural-network based on RBFNN control for active power filter. Int J Mach Learn Cybern, 2019;10:1139–1150.

[24] Treesatayapun C, Uatrongjit S. Adaptive controller with fuzzy rules emulated structure and its applications. Eng Appl Intell Intell, 2005;18:603–615.

[25] Treesatayapun C. Adaptive control based on IF–THEN rules for grasping force regulation with unknown contact mechanism. Robot Comput Integr Manuf, 2014;30:11–18.

[26] Abouheaf M, Gueaieb W. Neurofuzzy reinforcement learning control schemes for optimized dynamical performance. 2019 IEEE International Symposium on Robotics and Sensors Environments (ROSE), Ontario, Canada; 2019 June. p. 17–18.

[27] Treesatayapun C. Fuzzy-rule emulated networks based on reinforcement learning for nonlinear discrete-time controllers. ISA Trans, 2008;47:362–373.

[28] Wei Q, Lewis FL, Sun Q, et al. Discrete-time deterministic q-learning: a novel convergence analysis. IEEE Trans Cybern, 2017;47(5):1224–1237.

**Appendix 1. Proof of Theorem 4.1**

Let us refer to the standard Lyapunov function as

\[ V(k) = V_1(k) + V_2(k) + V_3(k) + V_4(k) \]
\[ = \rho_1 e^2(k) + \frac{\rho_2}{\eta} \tilde{p}_a(k) \tilde{p}_a(k) + \frac{\rho_3}{\eta} \tilde{p}_c(k) \tilde{p}_c(k) + \rho_4 \lambda_c^2(k - 1), \]  \hspace{1cm} (A1)

where \( \rho_1, \rho_2, \rho_3 \) and \( \rho_4 \) are positive constants satisfying the following conditions:

\[ \rho_1 > \frac{3}{4} p \rho_3, \]  \hspace{1cm} (A2)
\[ \rho_2 > \frac{\rho_1 g_M^2 + (\rho_3/8)g}{g_m} \rho_3, \]  
(A3)

\[ \rho_3 > \frac{\rho_4}{\delta^2} \]  
(A4)

and

\[ \rho_4 > \frac{\rho_3}{4} \]  
(A5)

Utilising (14), \( \Delta V_1 (k) \) is obtained as

\[ \Delta V_1 (k) = \rho_1 [e^2 (k + 1) - e^2 (k)] = \rho_1 [g(k) \Lambda_a (k) + d_a (k)]^2 - e^2 (k) \leq \rho_1 [2g^2 (k) \Lambda_a^2 (k) + 2d_a^2 (k) - e^2 (k)] \leq - \rho_1 e^2 (k) + 2 \rho_1 g_M^2 \Lambda_a^2 (k) + 2 \rho_1 d_a^2 . \]  
(A6)

Recalling the tuning law in (26), \( \Delta V_2 (k) \) is expressed as

\[ \Delta V_2 (k) = \frac{\rho_2}{\eta_a} \left[ \beta_a^T (k + 1) \beta_a (k + 1) - \beta_a^T (k) \beta_a (k) \right] = -2 \rho_2 [g(k) \Lambda_a (k) + \hat{L}(k)] \beta_a^T (k) \phi (k) + \rho_2 \eta_a [g(k) \Lambda_a (k) + \hat{L}(k)] \phi (k) + \rho_2 \eta_a ||\phi_a (k)||^2 [g(k) \Lambda_a (k) + \hat{L}(k)]^2 . \]  
(A7)

Applying the lower bound and upper bound of \( g(k) \), it leads to

\[ \Delta V_2 (k) \leq -2 \rho_2 g_m \Lambda_a^2 (k) - \rho_2 \eta_a \Lambda_a (k) \hat{L}(k) + \rho_2 \eta_a ||\phi_a (k)||^2 g_M^2 \Lambda_a^2 (k) + \rho_2 \eta_a ||\phi_a (k)||^2 [\hat{L}(k) - 2g(k) \Lambda_a (k) \hat{L}(k)] \leq -2 \rho_2 g_m \Lambda_a^2 (k) - (g_m - \eta_a ||\phi_a (k)||^2 g_M^2) \Lambda_a^2 (k)  
- \Lambda_a (k) [\hat{L}(k)]^2 + \eta_a [\Lambda_a (k)]^2 \eta_a [\Lambda_a (k)]^2  
\leq \rho_2 \left[- g_m \Lambda_a^2 (k) - (g_m - \eta_a ||\phi_a (k)||^2 g_M^2) \Lambda_a^2 (k) + \eta_a [\Lambda_a (k)]^2 \eta_a [\Lambda_a (k)]^2 \right]  
\leq -2 \rho_2 g_m \Lambda_a^2 (k) - 2 \rho_2 g_m \Lambda_a (k) \hat{L}(k) + \rho_2 \eta_a [\Lambda_a (k)]^2 \eta_a [\Lambda_a (k)]^2 \left[ \Lambda_a^2 (k) + 2 \Lambda_a (k) [\hat{L}(k) + \eta_a ||\phi_a (k)||^2 g(M)] \right]  
\leq -2 \rho_2 g_m \Lambda_a^2 (k) - 2 \rho_2 g_m \Lambda_a (k) \hat{L}(k) + \rho_2 \eta_a ||\phi_a (k)||^2 g_M^2 \left[ \Lambda_a^2 (k) + 2 \Lambda_a (k) \hat{L}(k) + \eta_a ||\phi_a (k)||^2 g_M^2 \right]  
\]
\[
\begin{align*}
\Delta V_3(k) &= -\rho_2 g_m \Lambda^2(k) + \frac{\rho_2}{g_m} \tilde{L}^2(k) - \rho_2 (g_m - \eta_0 \|\phi_0(k)\|^2 g_m^2) \bigg/ \Lambda_0(k) \\
&\quad + \frac{[1 - \eta_0 \|\phi_0(k)\|^2 g_m^2] \tilde{L}(k)}{g_m - \eta_0 \|\phi_0(k)\|^2 g_m^2}^2.
\end{align*}
\]

By using the learning law of MiFREnc in (32), \(\Delta V_3(k)\) is derived as

\[
\begin{align*}
\Delta V_3(k) &= \frac{\rho_2}{\eta_c} \left[ \beta^T_c (k + 1) \hat{H}_c (k + 1) - \beta^T_c (k) \hat{H}_c (k) \right] \\
&= \frac{\rho_2}{\eta_c} \left[ -2 \eta_c \delta e_c (k) \beta^T_c (k) \phi_c (k) + \eta_c^2 \delta^2 e^2_c (k) \|\phi_c (k)\|^2 \right] \\
&= -2 \rho_2 \delta \Lambda_c (k) e_c (k) + \rho_3 \eta_c \delta^2 \|\phi_c (k)\|^2 e^2_c (k).
\end{align*}
\]

Recalling \(e_c (k)\) in (28) with \(\pm \delta L (k)\) and \(\pm L (k - 1)\) and using (A10), it yields

\[
\begin{align*}
e_c (k) &= \delta [\tilde{L}(k) - L(k)] + \delta L(k) - [\tilde{L}(k - 1) - L(k - 1)] - L(k - 1) + I(k) \\
&= \delta [\beta^T_c (k) \phi_c (k) - \beta^T_c (k - 1) \phi_c (k - 1) - \delta e_c (k)] + \delta L(k) - L(k - 1) + I(k) \\
&\quad - \beta^T_c (k - 1) F_c (k - 1) - \beta^T_c (k - 1) \phi_c (k - 1) + \delta e_c (k - 1) \\
&= \delta \Lambda_c (k) - \Lambda_c (k - 1) + \delta L(k) - L(k - 1) + I(k) - \delta e_c (k) \\
&\quad + \delta e_c (k - 1) (A10)
\end{align*}
\]

or

\[
\begin{align*}
\delta \Lambda_c (k) &= e_c (k) - \delta L(k) + \Lambda_c (k - 1) + \delta \epsilon_c (k) - \epsilon_c (k - 1). (A11)
\end{align*}
\]

By using (A11) and (16), (A9) can be derived as

\[
\begin{align*}
\Delta V_3(k) &= -2 \rho_2 e_c (k) \left[ e_c (k) - \delta L(k) + \Lambda_c (k - 1) + L(k - 1) - I(k) \\
&\quad + \delta e_c (k) - \epsilon_c (k - 1) \right] + \rho_3 \eta_c \delta^2 \|\phi_c (k)\|^2 e^2_c (k) \\
&= -\rho_3 \left[ 1 - \eta_c \delta^2 \|\phi_c (k)\|^2 \right] e^2_c (k) - \rho_3 \delta e^2_c (k) + 2 \rho_2 e_c (k) \left[ \delta L(k) \\
&\quad - \Lambda_c (k - 1) - L(k - 1) + I(k) - \delta e_c (k) + \epsilon_c (k - 1) \right] \\
&= -\rho_3 \left[ 1 - \eta_c \delta^2 \|\phi_c (k)\|^2 \right] e^2_c (k) - \rho_3 \delta^2 \Lambda^2_c (k) + \rho_3 \left[ \delta L(k) \\
&\quad - \Lambda_c (k - 1) - L(k - 1) + I(k) - \delta e_c (k) + \epsilon_c (k - 1) \right]^2 \\
&\leq -\rho_3 \left[ 1 - \eta_c \delta^2 \|\phi_c (k)\|^2 \right] e^2_c (k) - \rho_3 \delta^2 \Lambda^2_c (k) + \frac{\rho_3}{4} \Lambda^2_c (k - 1) \\
&\quad + \rho_3 \delta^2 (k) + \frac{\rho_3}{4} \left[ \delta L(k) - L(k - 1) \right]^2 \\
&\quad + \frac{\rho_3}{4} \left[ \delta e_c (k) - \epsilon_c (k - 1) \right]^2 \\
&\leq -\rho_3 \left[ 1 - \eta_c \delta^2 \|\phi_c (k)\|^2 \right] e^2_c (k) - \rho_3 \delta^2 \Lambda^2_c (k) + \frac{\rho_3}{4} \Lambda^2_c (k - 1)
\end{align*}
\]
Finally, \( \Delta V_4(k) \) is formulated as

\[
\Delta V_4(k) = \rho_4 \left[ \Lambda_4^2(k) - \Lambda_4^2(k - 1) \right].
\] (A13)

Recalling (A6), (A8), (A12) and (A13), \( \Delta V(k) \) is rewritten as

\[
\Delta V(k) \leq -\frac{\rho_1}{3} e^2(k) + \rho_1 g_m^2 \Lambda_2^2(k) + \rho_1 d_M^2 \\
- \rho_2 g_m \Lambda_3^2(k) - \rho_2 (g_m - \eta_a \| \phi_a(k) \|^2 g_M^2) \bigg( \Lambda_a(k) + \\
\frac{[1 - \eta_a \| \phi_a(k) \|^2 g_M^2] \| \phi_a(0) \| L(k)}{g_m - \eta_a \| \phi_a(k) \|^2 g_M^2} \bigg)^2 + \frac{\rho_2}{g_m} L^2(k) \\
- \rho_3 \left( 1 - \eta_c \delta^2 \| \phi_c(k) \|^2 \right) e_c^2(k) - \rho_3 \delta^2 \Lambda_4^2(k) + \frac{\rho_3}{4 \Lambda_4^2(k - 1)} \\
+ \rho_3 \frac{4}{3} \rho_2 e^2(k) + \rho_3 \frac{4}{3} q \Lambda_2^2(k) + \rho_3 \| \phi_a(k) \|^2 \bigg( \Lambda_a(k) + \\
\frac{[1 - \eta_a \| \phi_a(k) \|^2 g_M^2] \| \phi_a(0) \| L(k)}{g_m - \eta_a \| \phi_a(k) \|^2 g_M^2} \bigg)^2 \\
+ \mathcal{W}_M \\
\leq -\mathcal{W}_e e^2(k) - \mathcal{W}_a \Lambda_4^2(k) - \mathcal{W}_{c0} \Lambda_2^2(k) - \mathcal{W}_{c1} \Lambda_2^2(k - 1) \\
- \mathcal{W}_c e_c^2(k) + \mathcal{W}_M,
\] (A14)

where

\[
\mathcal{W}_M \geq \rho_1 d_M^2 + \rho_3 e^2 \Lambda_2^2 + \frac{\rho_3}{3} \rho_2 \Lambda_4^2 + \frac{\rho_3}{8} (\gamma - 1)^2 + \frac{\rho_2}{g_M^2} L^2 \\
- \rho_2 [g_m - \eta_a \| \phi_a(k) \|^2 g_M^2] \bigg( \Lambda_a(k) + \\
\frac{[1 - \eta_a \| \phi_a(k) \|^2 g_M^2] \| \phi_a(0) \| L(k)}{g_m - \eta_a \| \phi_a(k) \|^2 g_M^2} \bigg)^2,
\] (A15)

\[
\mathcal{W}_e = \frac{\rho_1}{3} - \frac{\rho_3}{4} \rho_r,
\] (A16)

\[
\mathcal{W}_a = \rho_2 g_m - \rho_1 g_M^2 - \frac{\rho_3}{8} q,
\] (A17)

\[
\mathcal{W}_{c0} = \rho_3 \delta^2 - \rho_4,
\] (A18)

\[
\mathcal{W}_{c1} = \rho_4 - \frac{\rho_3}{4}
\] (A19)
\[ V_c = \rho_3 \left[ 1 - \eta_c \delta^2 ||\phi_c(k)||^2 \right]. \] (A20)

According to the conditions in (A2) – (A5), \( V_e, V_a, V_{c0} \) and \( V_{c1} \) are always positive. Furthermore, by setting the membership functions of FRENa and MiFRENc, the upper limits exist such that
\[ 0 < ||\phi_a(k)||^2 \leq \nu_a \] (A21)
and
\[ 0 < ||\phi_c(k)||^2 \leq \nu_c. \] (A22)

Combining with (34) and (35), it leads to
\[ g_m - \eta_a ||\phi_a(k)||^2 g_m^2 > 0 \] (A23)
and
\[ \Omega_M \geq \rho_1 d_m^2 + \rho_3 \varepsilon_{cM}^2 + \frac{\rho_3}{8} \beta_{oM}^2 + \left[ \frac{\rho_3}{8} (\gamma - 1)^2 + \frac{\rho_2}{g_o} \right] L_M. \] (A24)

By this mean, we have
\[ |e(k)| \geq \Omega_e \triangleq \sqrt{\frac{\Omega_{M}}{\Omega_{e}'}}. \] (A25)
\[ |\Lambda_a(k)| \geq \Omega_a \triangleq \sqrt{\frac{\Omega_M}{\Omega_a}} \] (A26)
and
\[ |\Lambda_c(k)| \geq \Omega_c \triangleq \sqrt{\frac{\Omega_M}{\Omega_{c0}}}. \] (A27)

The proof is completed here. □