Complex & Intelligent Systems (2024) 10:2271–2297
https://doi.org/10.1007/s40747-023-01254-6

ORIGINAL ARTICLE

Event-triggered model-free adaptive control for nonlinear systems using intuitionistic fuzzy neural network: simulation and experimental validation

Sameh Abd-Elhaleem[1] · Mohamed A. Hussien[1] · Mohamed Hamdy[1,2] · Tarek A. Mahmoud[1]

Received: 6 March 2023 / Accepted: 18 September 2023 / Published online: 14 November 2023

Abstract
This article presents model-free adaptive control based on an intuitionistic fuzzy neural network for nonlinear systems with event-triggered output. Essentially, model-free adaptive control (MFAC) is constructed by establishing an online approximate model of the controlled system using the pseudo-partial derivative (PPD) form. By the proposed scheme, first, an intuitionistic fuzzy neural network (IFNN) is developed as an estimator for time-varying PPD in both compact-form dynamic linearization (CFDL) and partial-form dynamic linearization (PFDL) for the MFAC technique. Second, two periodic event-triggered output methods are integrated with the proposed IFNN-based MFAC in both forms to save communication resources and reduce the computation burden and energy consumption. Based on the Lyapunov theory and BIBO stability approach, necessary conditions are established to guarantee the convergence of the adaptive law of the IFNN controller and the boundary of the tracking error of the closed loop system. Third, regarding the feasibility and the effectiveness of the developed control method, two simulation examples including the continuous stirred-tank reactor (CSTR) system and the heat exchanger system are given. Finally, the practical validation of the proposed data-driven control method is conducted via the speed control of a DC motor.

Keywords Data-driven control · Model-free adaptive control (MFAC) · Intuitionistic fuzzy neural network (IFNN) · Discrete event-triggered

Introduction

Nowadays, the network control system (NCS) has attracted much interest because it is easy to install and has high flexibility [1–3]. On the other hand, the bandwidth, and the computing capabilities of NCS are limited. Hence, in order to reduce the computational burden and enhance the limited communication resources, event-triggered control (ETC) schemes have been widely concerned [4, 5]. Fundamentally, the event-triggered control scheme samples the system's output and performs the control action whenever a specific event exceeds a defined threshold. There are two types of event-triggered control schemes: continuous event-triggered control (CETC) scheme and periodic event-triggered control (PETC) scheme. Based on CETC, dedicated analog hardware is needed to continuously check the event condition, which is difficult in digital implementations [6–11]. Hence, to make be more suited for practical implementations, PETC only checks the event condition periodically at explicit sampling instances [12–14].

Recently, the ETC has been developed for nonlinear model-based control systems to improve its ability to deal with the limited resources of NCS. In this context, H∞ ETC is introduced for vehicle active suspension systems under linear fraction uncertainties [15]. In [12], a predictor-based PETC is developed for nonlinear uncertain systems with

✉ Tarek A. Mahmoud
tarek.mohamed@el-eng.menofia.edu.eg

Sameh Abd-Elhaleem
sameh.abdelhaleem@el-eng.menofia.edu.eg

Mohamed A. Hussien
mohamed.hussien@el-eng.menofia.edu.eg

Mohamed Hamdy
mohamed.hamdy@el-eng.menofia.edu.eg

1 Industrial Electronics and Control Engineering Department, Faculty of Electronic Engineering, Menoufia University, Menouf, Egypt

2 Department of Electrical Engineering, Faculty of Engineering, Pharos University, Alexandria, Egypt

input delay, adaptive PETC output-feedback control is used to control switched nonlinear systems [13], and a nonlinear NCS with a PETC is developed in [14]. A modified repetitive PETC with equivalent input disturbance for linear systems subject to unknown disturbance was proposed in [16]. In [17], T-S fuzzy model is used to approximate the nonlinear system with a repetitive control based on a PETC. The sliding-mode control (SMC) approaches for nonlinear systems have been developed using ETC schemes [18, 19]. It is noteworthy that the common features in most of the above-mentioned results are presented on the basis of the dynamic equations of the system known in advance. However, the system dynamics in real applications are complex and therefore it is difficult to obtain the precise dynamic equations of the systems. Therefore, several results of ETC approaches based on data-driven control methods have been emergence.

Basically, the data-driven control (DDC) schemes are constructed directly using input–output (I/O) measured data (online or offline) [20, 21]. Based on this concept, many strategies have been presented, such as dynamic programming [22], iterative feedback tuning [23, 24], iterative learning control [6, 9], virtual reference feedback tuning [25], and model free adaptive control (MFAC) [26–30]. Among them, the MFAC scheme has attracted a lot of attention in controlling nonlinear systems with unknown dynamics [31]. This scheme replaces the global nonlinear model with a sequence of dynamic equivalent linearized time-varying data models which are constructed at each current operating point using a time-varying term known as the pseudo-partial derivative (PPD). The online dynamic linearization model (DLM) can be accomplished in three different methods: full-form dynamic linearization (FFDL), partial-form dynamic linearization (PFDL), and compact-form dynamic linearization (CFDL) [26, 31, 32]. Consequently, based on the MFAC framework, the ETC approaches have been established for nonlinear systems [33]. Regarding the effectiveness of MFAC when working with unidentified systems, an ETC technique uses the estimated weight of the neural network to estimate the pseudo-gradient vector has been presented [34]. In [9], a low computation cost MFAC triggering approach is proposed where the MFAC with adaptive iterative learning control based on ETC is constructed for nonlinear systems.

Over the past decades, due to the capability of fuzzy reasoning and the learning abilities of neural networks, the fuzzy neural network (FNN) has become an effective approach in the modeling and control of nonlinear processes [35–39]. One of the developments of the fuzzy theory, the intuitionistic fuzzy (IF) theory includes the membership function and non-membership function in describing confusion [40]. Using this concept would get more flexibility and then more realistic results in several applications. Consequently, IF has been used in medical applications [41], control, and synchronization of chaotic systems [42, 43]. In [44] an adaptive IF neural network (IFNN) has been proposed for the control and synchronization of chaotic systems. Type-2 IFNN for regression problems has been proposed in [45].

Based on the above discussion, in this paper, a discrete event triggered MFAC using an IFNN for nonlinear systems is developed. In this proposed scheme, the IFNN is developed as an approximation tool for the time-varying PPD in the MFAC approach in both the partial and compact forms. Based on the discrete event-triggering framework to save system resources and reduce the computation time, the proposed MFAC using the IFNN is developed. Here, two PETC mechanisms are investigated; the first one tests the event-triggered condition at the fixed sampling period of the controlled systems whereas the second one examines the condition at certain instants depending on a predetermined period to save system resources even more. Using the backpropagation algorithm, the updating of the parameters of the IFNN is provided when the event-triggered condition is satisfied. On the basis of the Lyapunov theory and the BIBO method, sufficient conditions for the convergence and the stability of the proposed scheme are derived. In order to investigate the effectiveness of the proposed MFAC-PETC, two simulation examples including the CSTR system and heat exchanger system are given. Moreover, the proposed controllers are validated practically in a real-time application via the speed control for a shunt direct current (DC) motor. In the end, we can summarize the contribution of this article as follows:

(i) Developing the IFNN as an approximator tool in the MFAC approach for nonlinear systems using the PFDL and the CFDL.
(ii) Combining two PETC methods with the proposed MFAC based on the IF approximator to save system resources and reduce the computation time.
(iii) Extracting the conditions that guarantee the convergence of the training process of the IFNN approximator and the stability of the proposed data-driven control system.
(iv) The practical validation of the proposed scheme by means of the speed control of a DC motor.

The rest of this paper is organized as follows. In "MFAC scheme", the MFAC based on both PFDL and CFDL is explored. The proposed MFAC based on an IFNN including the network structure and the adaptive update laws is presented in "MFAC based on IFNN". In "Event-triggered MFAC scheme", the MFAC with the periodic event-triggered output scheme is described. The stability analysis is studied in "Convergence and stability analysis". In "Simulation results", simulation and practical results are introduced to demonstrate the performance of the proposed control scheme. Finally, the conclusions are summarized in "Conclusion".

MFAC scheme

Consider a discrete-time nonlinear single-input single-output (SISO) system given in the following general form:

$$y(k+1) = f\Big(y(k), y(k-1), \ldots, y\big(k-n_y\big), u(k), u(k-1)\ldots, u(k-n_u)\Big) \quad (1)$$

where, $u(k)$and $y(k) \in R$ stand to the control input and the system output at the time instant k, respectively. The two unknown orders of the above nonlinear system are represented by two unknown positive integers n_y, n_u with an unknown nonlinear function $f(.)$. For system (1), the following assumptions are defined.

Assumption 1. The partial derivatives of nonlinear function $f(.) \in R$ related to all variables are continuous.

Assumption 2. System (1) satisfies the generalized Lipschitz condition which states that $|\Delta y(k+1)| \leq s\|\Delta U(k)\|$ and $\|\Delta U(k)\| \neq 0$ for any k, where s is a positive constant, $\Delta y(k+1) = y(k+1) - y(k)$, $\Delta U(k) = [\Delta u(k), \Delta u(k-1), \ldots \Delta u(k-L+1)]^T$, $\Delta u(k-i) = u(k-i) - u(k-i-1)$ and $i = 0, 1, \ldots, L-1$, where $L > 0$ is called control input length constant of linearization.

Remark 1 It is worth noting that assumptions 1 and 2 are acceptable and reasonable from the perspective of practical applications. Assumption 1 is a standard constraint of controller design for typical nonlinear systems and Assumption 2 is a linear-like constraint that guarantees that the rate of change of the system output has an upper bound limitation with bounded change in the control input. For example, the output speed changes of a dc motor cannot go to infinity if the changes in the input voltage to its drive circuit are bounded. Also, many practical systems such as pressure, refrigeration, and temperature control can satisfy the above two assumptions [32].

Based on the assumptions mentioned above, system (1) can be linearized into the following form [32].

$$\hat{y}(k+1) = y(k) + \Phi^T(k)\Delta U(k), \quad (2)$$

where $\Phi(k)$ is the bounded pseudo-partial derivative (PPD) function. PPD is a slowly changed time-varying parameter that can be estimated at every sample to approximate the system attitude in a linear dynamic form. This function can be established using only the historical I/O measurement data of the closed-loop system. Generally speaking, this linearization form does not require any type of system parameter identification or structural information of the controlled system [34]. Based on the PPD approximation, different forms for the dynamic linearized model given in (2) including the CFDL and PFDL are defined. For the CFDL method, the PPD $\Phi(k)$ is given as a scalar value $\phi_1(k)$ and then model (2) becomes:

$$\hat{y}(k+1) = y(k) + \phi_1(k)\Delta u(k). \quad (3)$$

On the other side, the PPD value for the PFDL for model (2) is given as a vector:

$$\Phi(k) = [\phi_1(k), \phi_2(k), \ldots \phi_L(k)]^T, \ \|\Phi(k)\| < s, \quad (4)$$

where L could be chosen as an integer in the range $L = 2$ to $L = n_y + n_u$ [32]. It is noted that, the CFDL is simpler and faster when compared with the PFDL but with some reduction in the accuracy of the dynamic linearization. Moreover, the CFDL is a PFDL in case of $L = 1$.

Based on the linearized model (2), the control laws of the MFAC based on both the CFLD and PFDL are defined as below [32]:

$$u(k) = u(k-1) + \frac{p_1\phi_1(k)(y^*(k+1) - y(k))}{\lambda + |\phi_1(k)|^2} \quad (5)$$

$$u(k) = u(k-1) + \frac{p_1\phi_1(k)\big(y^*(k+1) - y(k)\big) - \phi_1(k)\sum_{i=2}^{L} p_i\phi_i(k)\Delta u(k-i+1)}{\lambda + |\phi_1(k)|^2}, \quad (6)$$

where $p = [p_1, \ldots, p_L]^T$is a step-size vector, $p_i \in (0, 1]$, $y^*(k)$is the desired output, $\lambda > 0$ is the weighting factor, $|\phi_1(k)| > \varepsilon$ at each sampling instant k, and ε is a positive constant.

Actually, the control laws (5) and (6) cannot be implemented because the elements of $\Phi(k)$ are unknown and time-varying parameters. Thus, the IFNN has developed to estimate the PPD function within the MFAC approach in the next section.

MFAC based on IFNN

To solve the MFAC problem (6), each element of $\Phi(k)$ can be approximated by the following form:

$$\widehat{\phi}_m(k) = \mathcal{F}_m(z, W) \quad (7)$$

where $z \in \mathcal{R}^n$ is the input vector and W are the approximation weights. In this work, the nonlinear function $\mathcal{F}_m(.)$ has been estimated using an intuitionistic fuzzy logic system in the neural network structure as described in the following sub-section followed by the adaptive law of the network weights.

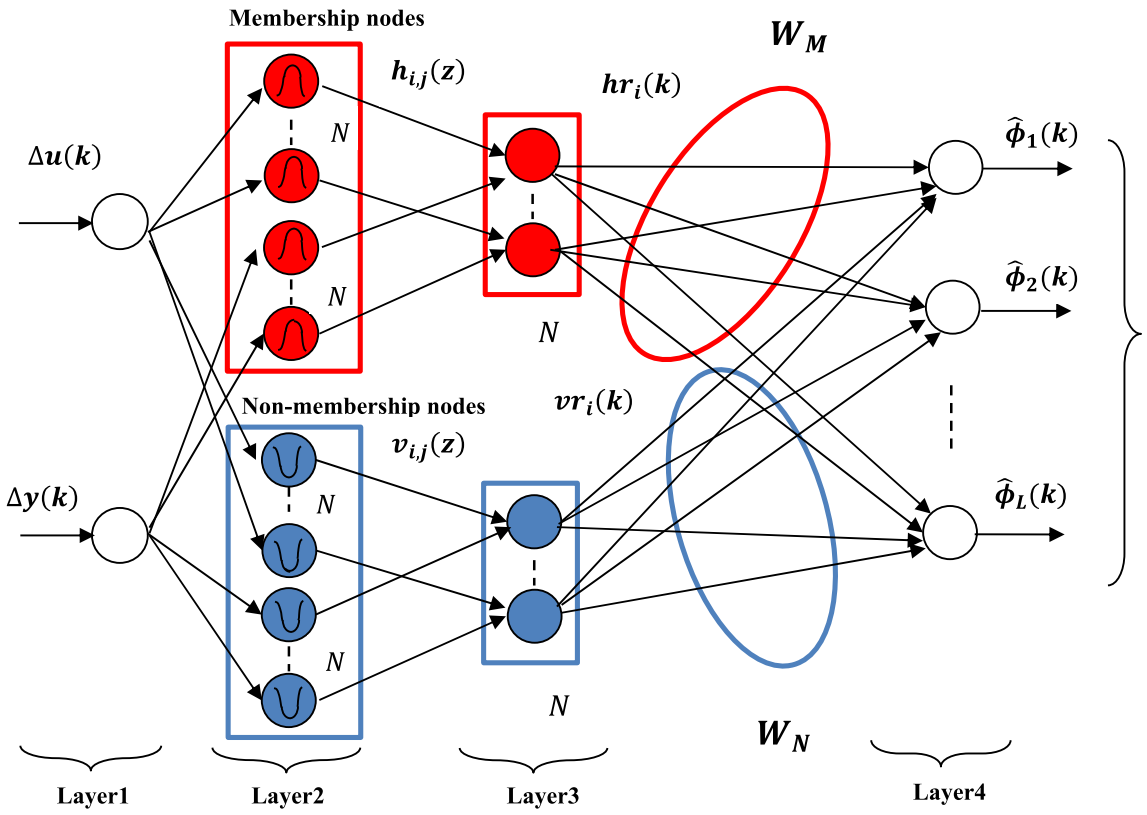


Fig. 1 Layers structure of IFNN

Structure of IFNN

To approximate the nonlinear function (7), an IFNN with two inputs and multi outputs has been developed as shown in Fig. 1. This network consists of four layers and the function of each layer is illustrated as follows.

Layer 1 (Input layer): this layer only accepts the input vector of the network. Here, it consists of two nodes such that $z = [\Delta y(k)\ \Delta u(k)]$.

Layer 2 (Fuzzification layer): this layer includes two groups of nodes. The first one represents the membership function part and the output of each node in this group can be calculated as:

$$h_{i,j}(z) = e^{\frac{-0.5(z_j - cm_{i,j})^2}{dm_{i,j}^2}} \quad i = 1, \ldots, N,\ j = 1, 2 \tag{8}$$

where N stands to the number of membership functions nodes for each input, $cm_{i,j}$ and $dm_{i,j}$ are the center and the width of the membership function, respectively. Besides, the second group represents the non-membership function part, and the output for each node is defined as:

$$v_{i,j}(z) = e^{\frac{-0.5(z_j - cn_{i,j})^2}{dn_{i,j}^2}} \quad i = 1, \ldots, N,\ j = 1, 2 \tag{9}$$

For this layer, the necessary condition for the intuitionistic fuzzy logic has been considered [42, 43]. This condition is defined as:

$$0 \le h_{i,j}(z) + v_{i,j}(z) \le 1,\ \ h_{i,j}(z),$$
$$v_{i,j}(z) \in [0,\ 1],\ i = 1,\ ..,\ N \text{ and } j = 1,\ 2. \tag{10}$$

Layer 3 (Rule layer): this layer is composed of two groups and each group with N nodes. The output for each node for the first group is the rule firing strength for the membership function part which can be calculated using the product operation:

$$hr_i(k) = \prod_{j=1}^{2} h_{i,j}(z) \tag{11}$$

Similarly, the nodes in the second group represent the rule firing strength for the non-membership function part and can be expressed as:

$$vr_i(k) = \prod_{j=1}^{2} v_{i,j}(z) \tag{12}$$

Layer 4 (Output layer): in this layer, the m^{th} output of the IFNN based on the center of area (COA) method is defined as:

$$\hat{\phi}_m(k) = \frac{\sum_{i=1}^{n}\left(W_{M,(i,m)} hr_i(k) - W_{N,(i,m)} vr_i(k)\right)}{\sum_{i=1}^{n}(hr_i(k) - vr_i(k))}, \quad \text{where } m = \{1, 2, \ldots, L\}, \tag{13}$$

W_M, W_N are the membership output weights and non-membership output weights, respectively. Thus, for $L = 1$, the IFNN is an approximator for the PPD in the CFDL. Generally, the estimation of the output of DLM can be defined as follows:

$$\hat{y}(k+1) = y(k) + \hat{\Phi}^T(k)\Delta U(k), \quad \text{where}$$
$$\hat{\Phi}(k) = \left[\hat{\phi}_1(k), \hat{\phi}_2(k), \ldots \hat{\phi}_L(k)\right]^T. \tag{14}$$

Updating laws

To derive the updating laws of the parameters of IFNN approximator for the PPD vector, the following cost function is defined:

$$E(k) = \frac{1}{2}e_p(k)^2 = \frac{1}{2}(y(k) - \widehat{y}(k))^2, \tag{15}$$

where $e_p(k)$ is the modeling error between the actual output $y(k)$ and the DLM output $\widehat{y}(k)$ at the sampling instant k. Hence, the free parameters of IFNN including the output weights (i.e., W_M, W_N), the membership function parameters (i.e., cm, dm) and the non-membership function (i.e., cn, dn) can be updated according to the gradient of descent method as below.

The updating law for the output weights can be given as:

$$\Delta W_{M(i,m)}(k) = -\eta\frac{\partial E(k)}{\partial W_{M(i,m)}(k)} = -\eta\frac{\partial E(k)}{\partial \widehat{y}(k)}\frac{\partial \widehat{y}(k)}{\partial \widehat{\phi}_m(k)}\frac{\partial \widehat{\phi}_m(k)}{\partial W_{M(i,m)}(k)} \tag{16}$$

$$\Delta W_{N(i,m)}(k) = -\eta\frac{\partial E(k)}{\partial W_{N(i,m)}(k)} = -\eta\frac{\partial E(k)}{\partial \widehat{y}(k)}\frac{\partial \widehat{y}(k)}{\partial \widehat{\phi}_m(k)}\frac{\partial \widehat{\phi}_m(k)}{\partial W_{N(i,m)}(k)}. \tag{17}$$

From (13) to (15), we have:

$$\Delta W_{M,(i,m)}(k) = -\eta e_p(k)\Delta u(k-m+1) \times \frac{hr_i(k)}{\sum_{i=1}^{N}(hr_i(k) - vr_i(k))} \tag{18}$$

$$\Delta W_{N,(i,m)}(k) = -\eta e_p(k)\Delta u(k-m+1) \times \frac{vr_i(k)}{\sum_{i=1}^{N}(hr_i(k) - vr_i(k))} \tag{19}$$

where $\eta > 0$, is the step size of the steepest descent algorithm. For the membership function parameters, the updating laws are:

$$\Delta cm_{i,j}(k) = -\eta\frac{\partial E(k)}{\partial cm_{i,j}(k)} = -\eta\frac{\partial E(k)}{\partial \hat{y}(k)}\frac{\partial \hat{y}(k)}{\partial \hat{\phi}_m(k)}\frac{\partial \hat{\phi}_m(k)}{\partial hr_i(k)}\frac{\partial hr_i(k)}{\partial h_{i,j}(z)}\frac{\partial h_{i,j}(z)}{\partial cm_{i,j}(k)} \tag{20}$$

$$\Delta dm_{i,j}(k) = -\eta\frac{\partial E(k)}{\partial dm_{i,j}(k)} = -\eta\frac{\partial E(k)}{\partial \hat{y}(k)}\frac{\partial \hat{y}(k)}{\partial \hat{\phi}_m(k)}\frac{\partial \hat{\phi}_m(k)}{\partial hr_i(k)}\frac{\partial hr_i(k)}{\partial h_{i,j}(z)}\frac{\partial h_{i,j}(z)}{\partial dm_{i,j}(k)} \tag{21}$$

From (8) to (15), the above laws can be calculated by:

$$\Delta cm_{i,j}(k) = -\eta e_p(k)\Delta u(k-m+1) \times \frac{W_{M,(i,m)}(k)}{\sum_{i=1}^{N}(hr_i(k) - vr_i(k))}\frac{hr_i(k)}{h_{i,j}(z)}\frac{\partial h_{i,j}(z)}{\partial cm_{i,j}(k)} \tag{22}$$

$$\Delta dm_{i,j}(k) = -\eta e_p(k)\Delta u(k-m+1) \times \frac{W_{M,(i,m)}(k)}{\sum_{i=1}^{N}(hr_i(k) - vr_i(k))}\frac{hr_i(k)}{h_{i,j}(z)}\frac{\partial h_{i,j}(z)}{\partial dm_{i,j}(k)} \tag{23}$$

Similarly, the updating laws for the non-membership function parameters can be given as:

$$\Delta cn_{i,j}(k) = -\eta e_p(k)\Delta u(k-m+1) \frac{W_{N,(i,m)}(k)}{\sum_{i=1}^{N}(hr_i(k) - vr_i(k))}\frac{vr_i(k)}{v_{i,j}(z)}\frac{\partial v_{i,j}(z)}{\partial cn_{i,j}(k)} \tag{24}$$

$$\Delta dn_{i,j}(k) = -\eta e_p(k)\Delta u(k-m+1) \times \frac{W_{N,(i,m)}(k)}{\sum_{i=1}^{N}(hr_i(k) - vr_i(k))}\frac{vr_i(k)}{v_{i,j}(z)}\frac{\partial v_{i,j}(z)}{\partial dn_{i,j}(k)} \tag{25}$$

Event-triggered MFAC scheme

In this section, a periodic event-triggering control mechanism is established with the proposed MFAC based on the IFNN. Figure 2 presents the proposed MFAC-IFNN-PETC, where the IFNN is used to estimate the PPD of the MFAC

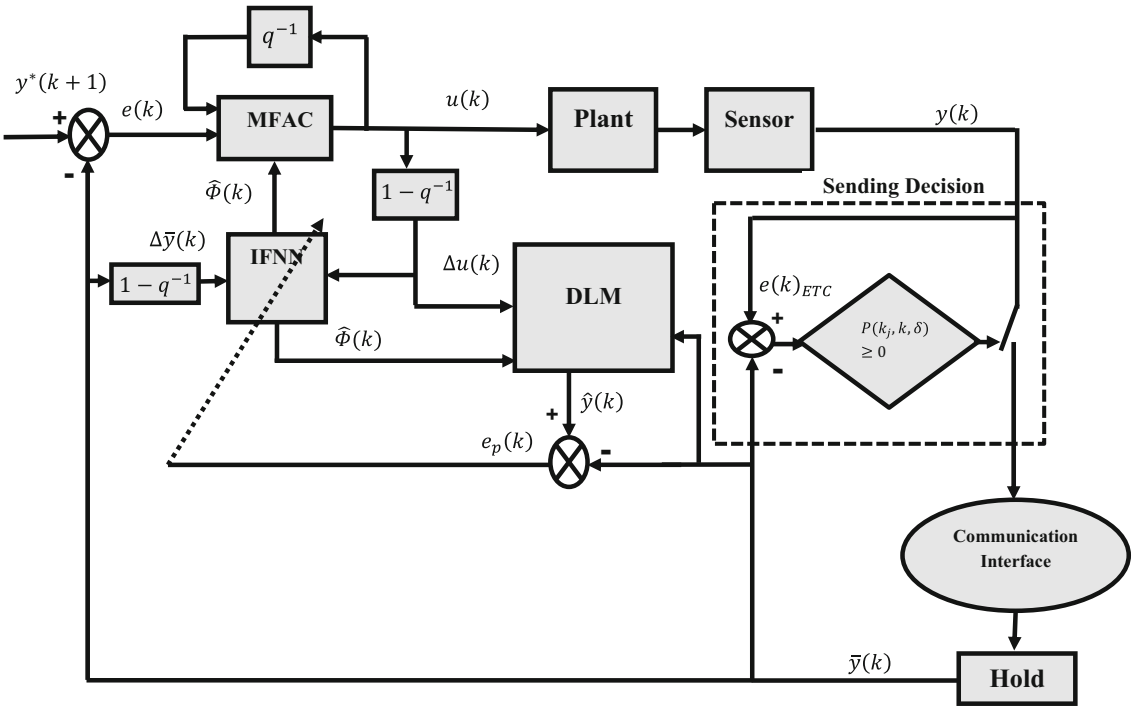


Fig. 2 The block diagram of the proposed MFAC-IFNN-PETC. (where q^{-1} is the back shift operator)

which is combined with the PETC in the sensor-to-controller transmission channel. In view of the PETC, the signal transmission from the sensor to the controller is executed only when the following condition is satisfied.

$$k_{j+1} = k_j + \min\left\{k > k_j \,|\left|y\left(k_j\right) - y\left(k\right)\right| \geq \delta\right\}, \quad j = 1, 2, 3, \ldots, \tag{26}$$

where $k_j$is the transmission instant; and $\delta > 0$ is the triggering parameter. The discrete event-triggering condition (26) shows that the next triggering instant $k_{j+1}$is influenced not only by the variation of the system's output, but also by the given triggering parameter δ, so it reduces the number of transmission instants.

According to (26), a discrete event function can be defined as:

$$P\left(k_j, k, \delta\right) = \left|y\left(k_j\right) - y(k)\right| - \delta \tag{27}$$

For every given set of triggering parameters, the measurement output will be sent to the controller only if $P(k_j, k, \delta) \geq 0$ is true. Otherwise, the controller will continue to use the most recent incoming data due to the influence of the hold block in Fig. 2.

Now, we can define:

$$\bar{y}(k) = y\left(k_j\right),\ k_j \leq k \leq k_{j+1} \tag{28}$$

where $\bar{y}(k)$ is the most recent event-triggered data until the next arrives.

Hence, the event-triggering error is given by:

$$e(k)_{\mathrm{ETC}} = \bar{y}(k) - y(k) \tag{29}$$

Accordingly, the event triggering condition can be formulated as:

$$|e(k)_{\mathrm{ETC}}| \geq \delta \tag{30}$$

Thus, based on the event triggering mechanism, the model free control laws given in (5) and (6) can be modified as:

$$u(k) = u(k-1) + \frac{p_1\widehat{\phi}_1(k)(y^*(k+1) - \bar{y}(k))}{\lambda + |\widehat{\phi}_1(k)|^2} \tag{31}$$

$$u(k) = u(k-1) + \frac{p_1\widehat{\phi}_1(k)(y^*(k+1) - \bar{y}(k)) - \widehat{\phi}_1(k)\sum_{i=2}^{L} p_i\widehat{\phi}_i(k)\Delta u(k-i+1)}{\lambda + |\widehat{\phi}_1(k)|^2} \tag{32}$$

Moreover, the adaptation of the IFNN approximator given in (16–25) will be only executed at the triggering instants.

In summary, the steps of the proposed MFAC based on IFNN in the event triggering form are given in Algorithm (1).

- Initialize the IFNN parameters $(\boldsymbol{i.e.}, \boldsymbol{W_M(0)}, \boldsymbol{W_N(0)}, \boldsymbol{cm(0)}, \boldsymbol{dm(0)}, \boldsymbol{cn(0)}, \boldsymbol{and\ dn(0)})$.
- Set the MEAC parameter $(\boldsymbol{i.e.}, \boldsymbol{\lambda})$ **.**
- Set event condition threshold **(i.e.,** $\boldsymbol{\delta}$)
- Set the initial value of the transmitted output $\overline{\boldsymbol{y}}(\boldsymbol{0}) = \boldsymbol{0}$
- Set the refrence output**:** $\boldsymbol{y}^*$
- Find the IFNN output $\widehat{\boldsymbol{\Phi}}(\boldsymbol{k})$.

On-line control process

1) Read sampled system output $\boldsymbol{y(k)}$.

2) Calculate the event triggering error (29), $\boldsymbol{e(k)_{ETC}}$

- **If** $|\boldsymbol{e(k)_{ETC}}| \geq \boldsymbol{\delta}$.
  - a) $\overline{\boldsymbol{y}}(\boldsymbol{k}) = \boldsymbol{y(k)}$.
  - b) Find the output of DLM : $\widehat{\boldsymbol{y}}(\boldsymbol{k+1}) = \overline{\boldsymbol{y}}(\boldsymbol{k}) + \widehat{\boldsymbol{\Phi}}^{\boldsymbol{T}}(\boldsymbol{k})\boldsymbol{\Delta U(k)}$.
  - **c)** Find the error $\boldsymbol{e_p(k)} = \widehat{\boldsymbol{y}}(\boldsymbol{k+1}) - \overline{\boldsymbol{y}}(\boldsymbol{k})$**.**
  - d) Update the IFNN parameters using (14)-(22).
  - **e) goto step 3**
- **Else goto step 3**

3) Execute the control action using (31) for CFLD or (32) for PFDL.

4) Use $\overline{\boldsymbol{y}}(\boldsymbol{k})$ as an input for IFNN.

5) Find the IFNN output $\widehat{\boldsymbol{\Phi}}(\boldsymbol{k+1})$.

6) **Go to step 1**

Algorithm (1)

It was remarking that the triggering condition in Algorithm (1) is measured at each sampling instant. Thus, to make the proposed controller algorithm with a low computation burden, a modified triggering method is developed. In this method, the event-triggered condition is tested after a number Q of sampling moments have passed to save system resources even more. The value of Q is selected as an integer number such that the following inequality is held:

$$1 < Q < \frac{n_k}{2}, \tag{33}$$

where n_k is the number of samples which is chosen within the controlled system time constant. Here, the discrete event-triggering condition (26) will be modified to

$$k_{j+1} = k_j + \min\{k > k_j | \left| y\left(k_j\right) - y(kQ) \right| \geq \delta\} \tag{34}$$

As a result, the steps of the proposed MFAC scheme based on the modified triggering condition are described in Algorithm (2).

- Initialize the IFNN parameters $(\boldsymbol{i.e.}, \boldsymbol{W_M(0)}, \boldsymbol{W_N(0)}, \boldsymbol{cm(0)}, \boldsymbol{dm(0)}, \boldsymbol{cn(0)}, \boldsymbol{and\ dn(0)})$.
- Set the MEAC parameter $(\boldsymbol{i.e.}, \boldsymbol{\lambda})$ **.**
- Set event condition threshold **(i.e.,** $\boldsymbol{\delta}$)
- Define the window of samples ($\boldsymbol{Q}$) and $\boldsymbol{k_{ET} = 0.}$
- Set the initial value of the transmitted output $\overline{\boldsymbol{y}}(\boldsymbol{0}) = \boldsymbol{0}$
- Set the refrence output**:** $\boldsymbol{y}^*$
- Find the IFNN output $\widehat{\boldsymbol{\Phi}}(\boldsymbol{k})$.

On-line control process

1) Read sample system output $\boldsymbol{y(k)}$.
2) **If** $\boldsymbol{k - k_{ET} \neq Q}$ **(** $\boldsymbol{Q}$ samples not passed) **goto step 5**
3) **Else** $\boldsymbol{k_{ET} = k}$.
4) Calculate the event triggering error (29), $\boldsymbol{e(k)_{ETC}}$
   - **If** $\boldsymbol{|e(k)_{ETC}| \geq \delta}$
     a) $\overline{\boldsymbol{y}}(\boldsymbol{k}) = \boldsymbol{y(k)}$**.**
     b) Find the output of DLM**:** $\widehat{\boldsymbol{y}}(\boldsymbol{k+1}) = \overline{\boldsymbol{y}}(\boldsymbol{k}) + \widehat{\boldsymbol{\Phi}}^{\boldsymbol{T}}(\boldsymbol{k})\boldsymbol{\Delta U(k)}$**.**
     c) Find the error $\boldsymbol{e_p(k)} = \widehat{\boldsymbol{y}}(\boldsymbol{k+1}) - \overline{\boldsymbol{y}}(\boldsymbol{k})$**.**
     d) Update the IFNN parameters using (14)-(22).
     e) **Go to step 5**
   - **Else go to step 5**
5) Execute the control action using (31) for CFDL or (32) for PFDL.
6) Use $\overline{\boldsymbol{y}}(\boldsymbol{k})$ as an input for IFNN.
7) Find the IFNN output $\widehat{\boldsymbol{\Phi}}(\boldsymbol{k+1})$**.**
8) **Go to step 1**

Algorithm (2)

Remark 2 For Algorithm 1, the initial weights of the IFNN are chosen as small random values, and λ is chosen according to theorem 2 $\lambda > \lambda_{\min}$. The event condition threshold δ is chosen as a small value greater than zero or as a percentage of the error between the current value of the output and the most recent incoming data, [46]. Moreover, the window of samples Q is chosen according to (33).

Convergence and stability analysis

In this section, the convergence of the updating laws of the IFNN parameters and the closed loop stability of the proposed algorithm are investigated.

Theorem 1. *Assuming an IFNN with the structure given in* Fig. 1 *to approximate the* $\Phi(k)$ *in the dynamic linearized model given in* (3)*. For any given input vector, the training process of the IFNN is convergent if the learning rate* η *in the updating laws* (16–25) *of the network satisfies the following condition*:

$$0 < \eta < \frac{2}{[\zeta(k)]^T \zeta(k)} \tag{35}$$

where:

$$\zeta(k) = \left[\frac{\partial e_P(k)}{\partial W_M(k)} \quad \frac{\partial e_P(k)}{\partial W_N(k)} \quad \frac{\partial e_P(k)}{\partial cm(k)} \quad \frac{\partial e_P(k)}{\partial cn(k)} \quad \frac{\partial e_P(k)}{\partial dm(k)} \quad \frac{\partial e_P(k)}{\partial dn(k)} \right]. \tag{36}$$

Proof: Define the Lyapunov function with respect to DLM output error as the cost function defined in (15). Hence, we have:

$$V(k) = E(k) = 0.5e_p^2(k) \tag{37}$$

Then, the change of Lyapunov function can be computed as:

$$\Delta V(k+1) = V(k+1) - V(k) = 0.5e_p^2(k+1) - 0.5e_p^2(k). \tag{38}$$

By:

$$e_P(k+1) = e_P(k) + \Delta e_P(k). \tag{39}$$

So (38) can be given as:

$$\Delta V(k+1) = \Delta e_P(k)(e_P(k) + 0.5\Delta e_P(k)) \tag{40}$$

According to the first-order Taylor expansion, the change of the training error can be expressed in term of the incrementing rate of IFNN weights as:

$$\Delta e_P(k) \cong \left[\frac{\partial e_P(k)}{\partial \psi(k)}\right]^T \Delta\psi(k) \tag{41}$$

where $\psi(k)=\begin{bmatrix} W_M(k) & W_N(k) & cm(k) & cn(k) & dm(k) & dn(k) \end{bmatrix}^T$ and $\Delta\psi(k)$ is defined as:

$$\Delta\psi(k) = -\eta\frac{\partial E(k)}{\partial \psi(k)} = -\eta\frac{\partial E(k)}{\partial e_P(k)}\frac{\partial e_P(k)}{\partial \psi(k)} \tag{42}$$

Hence, using (37), (42) can be rewritten as:

$$\Delta\psi(k) = -\eta e_P(k)\frac{\partial e_P(k)}{\partial \psi(k)} \tag{43}$$

So, (41) can be rewritten as,

$$\Delta e_P(k) \cong -\eta e_P(k)\left[\frac{\partial e_P(k)}{\partial \psi(k)}\right]^T \frac{\partial e_P(k)}{\partial \psi(k).} \tag{44}$$

Then, the change of the Lyapunov function in (40) can be reformulated as:

$$\Delta V(k+1) = -\eta e_p^2(k)\left[\frac{\partial e_P(k)}{\partial \psi(k)}\right]^T \frac{\partial e_P(k)}{\partial \psi(k)} \left(1 - 0.5\eta\left[\frac{\partial e_P(k)}{\partial \psi(k)}\right]^T \frac{\partial e_P(k)}{\partial \psi(k)}\right) \tag{45}$$

By defining $\zeta(k) = \frac{\partial e_P(k)}{\partial \psi(k)}$, we have:

$$\Delta V(k+1) = -\eta e_p^2(k)[\zeta(k)]^T\zeta(k)\left(1 - 0.5\eta[\zeta(k)]^T\zeta(k)\right) \tag{46}$$

Then, the condition of $\Delta V(k+1)$ is negative when

$$\eta > 0 \text{ and } \left(1 - 0.5\eta[\zeta(k)]^T\zeta(k)\right) > 0 \tag{47}$$

Finally, the convergence of the IFNN approximator in the MFAC scheme is guaranteed if the range of the learning rate η is

$$0 < \eta < \frac{2}{[\zeta(k)]^T\zeta(k)} \tag{48}$$

Remark 3. In condition (35), $[\zeta(k)]^T\zeta(k)$ can be considered as $\|\zeta(k)\|^2$, hence this condition can be rewritten as follows:

$$0 < \eta < \frac{2}{\zeta(k)^2}$$

where $\|\zeta(k)\|$ is the Euclidean norm of $\zeta(k)^2$. Hence, to guarantee to select of the learning rate η satisfying the given condition, the learning rate at any instant t can be set as follows:

$$\eta(t) = \eta(0)\frac{2}{G_{\max}(t).}$$

where $\eta(0)$ is a small positive number < 1, and $G_{\max}(t) = \max\left(\|\zeta(k)\|^2, G_{\max}(t-1)\right)$.

To study the tracking error convergence for the closed loop system based on the proposed MFAC, the following theorem is given.

Theorem 2. *For a constant regulator $y^*(k+1)$, the MFAC methods (31) and (32) regulate the nonlinear system (1) that meets Assumptions 1, 2, and there is $\lambda > \lambda_{\min}$, the tracking error $e(k)$ is uniformly ultimately bounded, and the ultimate bound is defined by the event-triggering error $e(k)_{\text{ETC}}$.*

Proof: Let's start with the MFAC in the PFDL form given in (32). According to the definition of the tracking error given by:

$$e(k) = y^*(k+1) - y(k) \tag{49}$$

By subtracting the event triggering error defined in (29) from (49), we have:

$$y^*(k+1) - \overline{y}(k) = e(k) - e(k)_{\text{ETC.}} \tag{50}$$

By using (50), the control law defined (32) can be reformulated as:

$$u(k) = u(k-1) + \frac{p_1\widehat{\phi}_1(k)\left(e(k) - e(k)_{ETC}\right) - \widehat{\phi}_1(k)\sum_{i=2}^{L} p_i\widehat{\phi}_i(k)\Delta u(k-i+1)}{\lambda + |\widehat{\phi}_1(k)|^2} \tag{51}$$

From (51), we have:

$$\Delta U(k) = u(k) - u(k-1) = \frac{p_1\widehat{\phi}_1(k)(e(k) - e(k)_{ETC}) - \widehat{\phi}_1(k)\sum_{i=2}^{L} p_i\widehat{\phi}_i(k)\Delta u(k-i+1)}{\lambda + |\widehat{\phi}_1(k)|^2} \tag{52}$$

By using the dynamic linearization form defined in (2) and (52), $\Delta y(k+1)$ can be given as:

$$\Delta y(k+1) = \frac{\|\Phi(k)\| p_1\widehat{\phi}_1(k)(e(k) - e(k)_{ETC}) - \|\Phi(k)\|\widehat{\phi}_1(k)\sum_{i=2}^{L} p_i\widehat{\phi}_i(k)\Delta u(k-i+1)}{\lambda + |\widehat{\phi}_1(k)|^2} \tag{53}$$

For a constant $y^*(k+1)$ and by using the tracking error defined in (49), we can write:

$$\begin{aligned} e(k+1) - e(k) &= y^*(k+2) - y(k+1) - y^*(k+1) + y(k) \\ &= -\Delta y(k+1) \end{aligned} \tag{54}$$

Hence, we have

$$e(k+1) = e(k) - \Delta y(k+1) \tag{55}$$

By substituting from (52) in (55), we obtain:

$$\begin{aligned} e(k+1) = &\left(1 - \frac{\|\Phi(k)\| p_1\hat{\phi}_1(k)}{\lambda + \left|\hat{\phi}_1(k)\right|^2}\right) e(k) \\ &+ \frac{\Phi(k) p_1\hat{\phi}_1(k) e(k)_{ETC}}{\lambda + \left|\hat{\phi}_1(k)\right|^2} \\ &+ \frac{\|\Phi(k)\| \hat{\phi}_1(k)\sum_{i=2}^{L} p_i\hat{\phi}_i(k)\Delta u(k-i+1)}{\lambda + \left|\hat{\phi}_1(k)\right|^2}. \end{aligned} \tag{56}$$

From (56), we can write as:

$$\begin{aligned} &|e(k+1)| \\ &= \left|1 - \frac{\|\Phi(k)\| p_1\hat{\phi}_1(k)}{\lambda + \left|\hat{\phi}_1(k)\right|^2}\right| |e(k)| + \left|\frac{\|\Phi(k)\| p_1\hat{\phi}_1(k)}{\lambda + \left|\hat{\phi}_1(k)\right|^2}\right| |\delta| \\ &\quad + \left|\frac{\|\Phi(k)\| \hat{\phi}_1(k)}{\lambda + \left|\hat{\phi}_1(k)\right|^2}\right| \left|\sum_{i=2}^{L} p_i\hat{\phi}_i(k)\Delta u(k-i+1)\right|. \end{aligned} \tag{57}$$

By assuming $\lambda_{min} = \frac{s^2}{4}$, where s is defined in Assumption 2, using the inequality defined in (58) for the denominator of (57), the inequality defined in (59) is given for a bounded value $M_1 > 0$.

$$a^2 + b^2 \geq 2ab \tag{58}$$

$$0 < M_1 \leq \left|\frac{\|\Phi(k)\|\widehat{\phi}_1(k)}{\lambda + \left|\widehat{\phi}_1(k)\right|^2}\right| \leq \left|\frac{s\widehat{\phi}_1(k)}{2\sqrt{\lambda}\widehat{\phi}_1(k)}\right| < \frac{s}{2\sqrt{\lambda_{\min}}} = 1 \tag{59}$$

Then, there exists a constant $\beta < 1$ such that $\left|1 - \frac{p_1\|\Phi(k)\|\widehat{\phi}_1(k)}{\lambda + |\widehat{\phi}_1(k)|^2}\right| \leq 1 - p_1M_1 = \beta$, so (57) can be rewritten as:

$$\begin{aligned} |e(k+1)| \leq \beta\,|e(k)| + p_1M_1\,|\delta| \\ + M_1\left|\sum_{i=2}^{L} p_i\widehat{\phi}_i(k)\Delta u(k-i+1)\right| \end{aligned} \tag{60}$$

According to that PPD has an upper limit $\overline{\Phi}$, and $p_i \epsilon (0,\ 1]$, the last term of (60) can be:

$$\begin{aligned} &\left|\sum_{i=2}^{L} p_i\hat{\phi}_i(k)\Delta u(k-i+1)\right| \\ &\leq (\max p_i)_{i\epsilon\{2,n\}}\left|\sum_{i=2}^{L}\hat{\phi}_i(k)\Delta u(k-i+1)\right| \\ &\leq \left(\max\hat{\phi}_i\right)_{i\epsilon\{2,n\}}\left|\sum_{i=2}^{L}\Delta u(k-i+1)\right| \\ &\leq \overline{\Phi}\left|\sum_{i=2}^{L}\Delta u(k-i+1)\right| \end{aligned} \tag{61}$$

When the control signal may flip from $U_{\max}$ to $-U_{\max}$, it gives the maximum$\Delta u(k-i+1)$. Hence, there is a bounded value M_2 where:

$$0 < M_2 \leq \left|\sum_{i=2}^{L} p_i\widehat{\phi}_i(k)\Delta u(k-i+1)\right| \leq 2\overline{\Phi}(L-1)U_{\max}. \tag{62}$$

So (60) can be redefined as:

$$|e(k+1)| \leq \beta|e(k)| + M_1(|\delta| + M_2) \tag{63}$$

Based on (63), we have:

$$\begin{aligned} &|e(k)| \leq \beta|e(k-1)| + M_1(|\delta| + M_2) \\ &|e(k-1)| \leq \beta|e(k-2)| + M_1(|\delta| + M_2) \\ &\quad\vdots\vdots \\ &|e(2)| \leq \beta|e(1)| + M_1(|\delta| + M_2). \end{aligned} \tag{64}$$

As a result, we can write:

$$\begin{aligned} &|e(k+1)| \leq \beta^2|e(k-1)| + \beta M_1(|\delta| + M_2) + M_1(|\delta| + M_2) \\ &\leq \beta^3|e(k-2)| + \beta^2 M_1(|\delta| + M_2) \\ &\quad + \beta M_1(|\delta| + M_2) + M_1(|\delta| + M_2) \\ &\quad\vdots\vdots \\ &\leq \beta^k|e(1)| + \beta^{k-1}M_1(|\delta| + M_2) + \beta^{k-2}M_1(|\delta| + M_2)\ldots \\ &\quad + \beta M_1(|\delta| + M_2) + M_1(|\delta| + M_2). \end{aligned} \tag{65}$$

Hence,

$$|e(k+1)| \leq \beta^k|e(1)| + \frac{M_1(|\delta| + M_2)}{1-\beta} \tag{66}$$

Based on $\beta<1$, as k increases, the part $\beta^k \to 0$. Thus, it yields:

$$|e(k+1)|_{k\to\infty} \leq \frac{M_1(|\delta| + M_2)}{1-\beta} \tag{67}$$

Accordingly, we can see that the tracking error is bound. Finally, since both $y^*(k)$ and $e(k)$ are bounded, the system output $y(k)$ is bounded too.

In a similar manner, for the MFAC in the CFDL form given in (31), the above sequence can be used. Hence, (57) can be rewritten as:

$$|e(k+1)| = \left|1 - \frac{p_1\phi_1(k)\widehat{\phi}_1(k)}{\lambda + \left|\widehat{\phi}_1(k)\right|^2}\right||e(k)| + \left|\frac{p_1\phi_1(k)\widehat{\phi}_1(k)}{\lambda + \left|\widehat{\phi}_1(k)\right|^2}\right||\delta| \tag{68}$$

By using (59), (68) can be rewritten as:

$$|e(k+1)| \leq \beta|e(k)| + M_1|\delta| \tag{69}$$

Then:

$$|e(k+1)| \leq \beta^k|e(1)| + \beta^{k-1}M_1|\delta| + \ldots\beta M_1|\delta| + M_1|\delta| \tag{70}$$

$$|e(k+1)| \leq \beta^k|e(1)| + \frac{M_1|\delta|}{1-\beta} \tag{71}$$

$$|e(k+1)|_{k\to\infty} = \frac{M_1|\delta|}{1-\beta} \tag{72}$$

Finally, we can conclude that the BIBO stability of the proposed algorithm is guaranteed, and the proof is completed.

It's worth mentioning that when the MFAC algorithm has no event-triggered transmission scheme $\delta = 0$, the tracking error (72) of the system is reduced to zero.

Simulation results

In this section, the proposed IFNN-MFAC technique is tested using two simulation examples: the control problem of CSTR and the regulation problem of a steam-water heat exchanger system. The proposed MFAC is implemented in the compact and the partial dynamic linearization forms (i.e., CFDL and PDLF). Moreover, each form is established in the event triggering mechanism according to Algorithm (1) and Algorithm (2). Thus, four controllers are considered here. These controllers are abbreviated as below:

- The CFDL-IFNN-MFAC-Event-1 denotes to the proposed MFAC in the CFDL form based on the event-triggering mechanism given in Algorithm (1).
- The CFDL-IFNN-MFAC-Event-2 denotes to the proposed MFAC in the CFDL form based on the event-triggering mechanism given in Algorithm (2).
- The PFDL-IFNN-MFAC-Event-1 denotes to the proposed MFAC in the PFDL based on the event-triggering mechanism given in Algorithm (1).
- The PFDL-IFNN-MFAC-Event-2 denotes to the proposed MFAC in the PFDL based on the event-triggering mechanism given in Algorithm (2).

Besides, the comparison between the developed controllers is performed using the performance indices; the root mean squared error (RMSE), integral of squared error (ISE), and integral of absolute error (IAE) criteria for evaluating the performance of the proposed techniques. These indices are defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N_s}\sum_{k=1}^{N_s}(y^*(k) - y(k))^2} \tag{73}$$

$$\text{ISE} = \int_0^\infty [e(t)]^2 \mathrm{d}t \tag{74}$$

$$\text{IAE} = \int_0^\infty |e(t)| \mathrm{d}t \tag{75}$$

where N_s is the number of the samples.

Example 1: CSTR system.

The dynamics of the considered CSTR system is described by the following equations [34]:

$$\begin{aligned} x_1(k+1) &= x_1(k) + \left[-x_1(k) + D_1(1-x_1(k))e^{A(k)}\right] \times 0.05 \\ x_2(k+1) &= x_2(k) + \Big[-x_2(k) + B_2 D_1(1-x_1(k))e^{A(k)} \\ &\quad -D_2(x_2(k)-u(k))\Big] \times 0.05 + d(k) \\ y(k) &= x_1(k), \end{aligned} \tag{76}$$

$x_1(k)$, $x_2(k)$ are the system states; $y(k)$ is the output of the system; $D_1 = 0.036$, $D_2 = 25.2$, $B_1 = 28.5$, $B_2 = 21.5$ are scalar parameters of the system; $A(k) = \frac{B_1 x_2(k)}{B_1 + x_2(k)}$ and $d(k) = 0.01\cos(0.05k)\cos(x_1(k))$ are time varying parameters of the system. It should be noted that the proposed control schemes do not use any system information, such as linear or nonlinear features, system order, and so on. The system model (73) is solely used to generate the necessary I/O data and is not used in the controller design process. The proposed controllers are performed by using IFNN has the following hyperparameter $N = 5$, $cm = [0.20\ 0.40\ 0.06\ 0.81]^T$, $cn = [1.4\ 1.6 - 0.6 - 0.4 - 0.2]^T$, $d_m = 20$, *and* $d_n = 1$. The CFDL-IFNN-MFAC law (31) is implemented with $p_1 = 1$, and $\lambda = 0.06$, on the other side; the PFDL-IFNN-MFAC (32) is implemented with $L = 3$, $p_1 = p_2 = p_3 = 1$, and $\lambda = 0.04$. For the event-triggering Algorithm (2), the parameter Q is set to 8. Two simulation cases are performed to demonstrate the performance of the proposed control schemes with the CSTR system.

Case 1: Tracking response for a square wave reference. In this task, the developed controllers are applied to the chemical reactors such that, the desired system output is given by:

$$y^*(k+1) = \begin{cases} 0.6, & k < 500 \\ 0.2, & 500 \le k < 1000 \\ 0.6, & 1000 \le k < 1500 \\ 0.2, & k \ge 300 \end{cases} \tag{77}$$

CFDL results: Figure 3 shows the tracking performances of the output using CFDL-IFNN-MFAC-Event-1 and CFDL-IFNN-MFAC-Event-2 and Fig. 4 depicted the obtained control signals for the two controllers. The inter-event intervals can be seen in Figs. 5, 6. It is clear that the system response under the CFDL-IFNN-MFAC-Event-1 is faster and smoother than CFDL-IFNN-MFAC-Event-2.

PFDL results: Figure 7 shows the tracking performances of the output using both PFDL-IFNN-MFAC-Event-1 and PFDL-IFNN-MFAC-Event-2 and Fig. 8 shows the control signals for the two controllers. The inter-event intervals can be seen in Figs. 9, 10. The system response under the PFDL-IFNN-MFAC-Event-1 has smaller overshoots and is faster than PFDL-IFNN-MFAC-Event-2.

In this case, it is shown clearly in Figs. 3 and 7 that the response of CSTR under the CFDL/PFDL-IFNN-MFAC-Event-1 is faster with smaller overshoots than CFDL/PFDL-IFNN-MFAC-Event-2. The inter-event intervals in Figs. 5, 6 and Figs. 9, 10 show that the second algorithm when combined with the MFAC in the CFDL-IFNN-MFAC-Event-2 and PFDL-IFNN-MFAC-Event-2 lead to a noticeable reduction in the number of events and the minimum inter-event interval is determined by the value of the parameter Q.

Case 2: Tracking response under time-varying system parameters for a step reference. In this case, the proposed controllers are tested with time-varying system parameters to investigate the adaptation property of the proposed controller. During this simulation case,$d(k)$ at time instant k is given by:

$$d(k) = \begin{cases} 0.01\cos(0.05k)\cos(x_1(k))\ 0 \le k < 1300 \\ 0.01\cos(0.05k)\cos(x_1(k)) + 1\ k \ge 1300. \end{cases} \tag{78}$$

CFDL results The results of the second simulation case are illustrated in Figs. 11 and the obtained control signals for the two controllers are depicted in Fig. 12. The inter-event intervals of the two control schemes are shown in Figs. 13, 14.

The control performance with time-varying parameters is quite satisfactory and the system response under the CFDL-IFNN-MFAC-Event-1 is faster and smoother than CFDL-IFNN-MFAC-Event-2.

PFDL results: Figure 15 shows the tracking performances of the output using both PFDL-IFNN-MFAC-Event-1 and PFDL-IFNN-MFAC-Event-2 and Fig. 16 shows the control signals for the two controllers. The inter-event intervals can be seen in Figs. 17, 18.

In case 2, it is obvious that the CFDL/PFDL-IFNN-MFAC-Event-1 controller can deal with the time-varying system parameters and overcomes their effect in a shorter time than the CFDL/PFDL-IFNN-MFAC-Event-2. The number of events is reduced when using the second algorithm and the minimum inter-event interval is determined by the value of the parameter Q.

The performance indices for the proposed controllers are shown in Table 1 which shows that the tracking performance of CFDL/PFDL-IFNN-MFAC-Event-1 and Event-2 methods are not as good as the traditional fixed periodic sampling MFAC schemes where weights are updated periodically. The CFDL/PFDL-IFNN-MFAC-Event-1 gives a faster response with a smaller overshoot when compared with

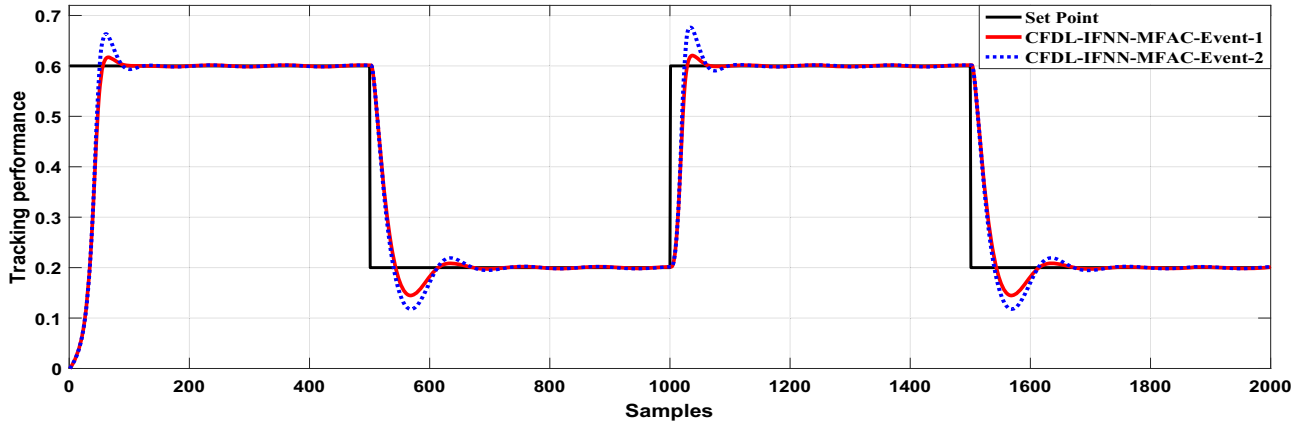


Fig. 3 Tracking response of the CSTR system using CFDL-IFNN-MFAC-Event-1 and CFDL-IFNN-MFAC-Event-2 (case: 1)

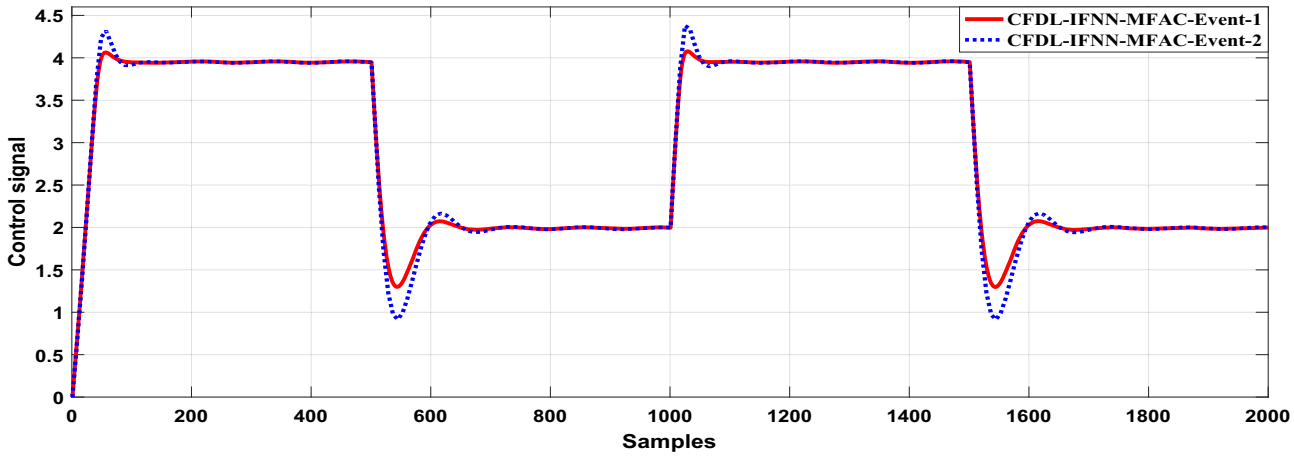


Fig. 4 Control signals for CFDL-IFNN-MFAC-Event-1 and CFDL-IFNN-MFAC-Event-2 with the CSTR system (case: 1)

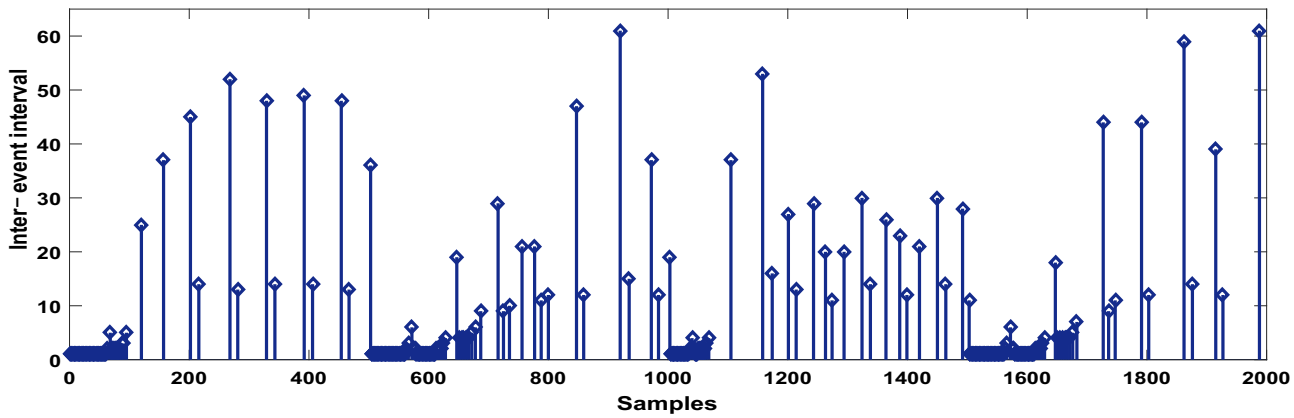


Fig. 5 Inter-event intervals for CFDL-IFNN-MFAC-Event-1 with the CSTR system (case: 1)

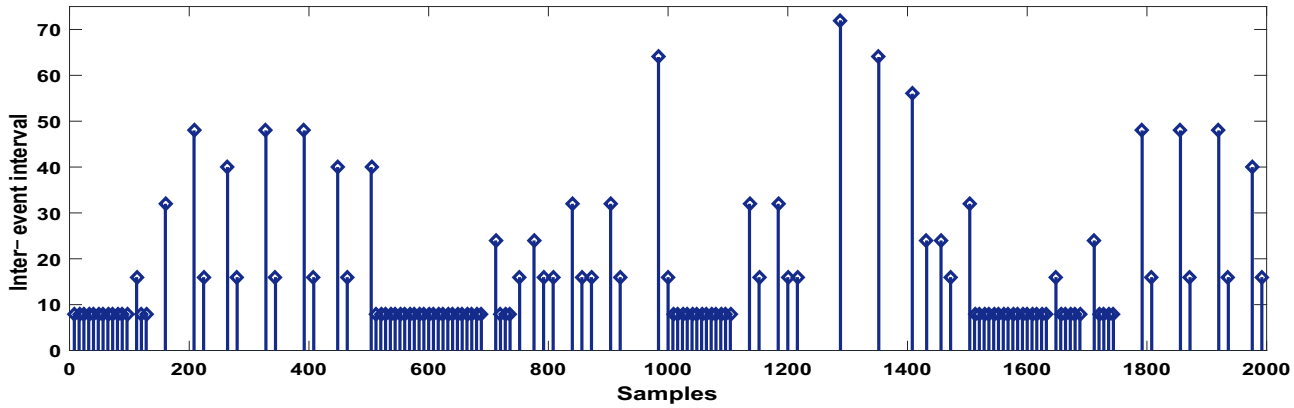


Fig. 6 Inter-event intervals for CFDL-IFNN-MFAC-Event-2 with the CSTR system (case: 1)

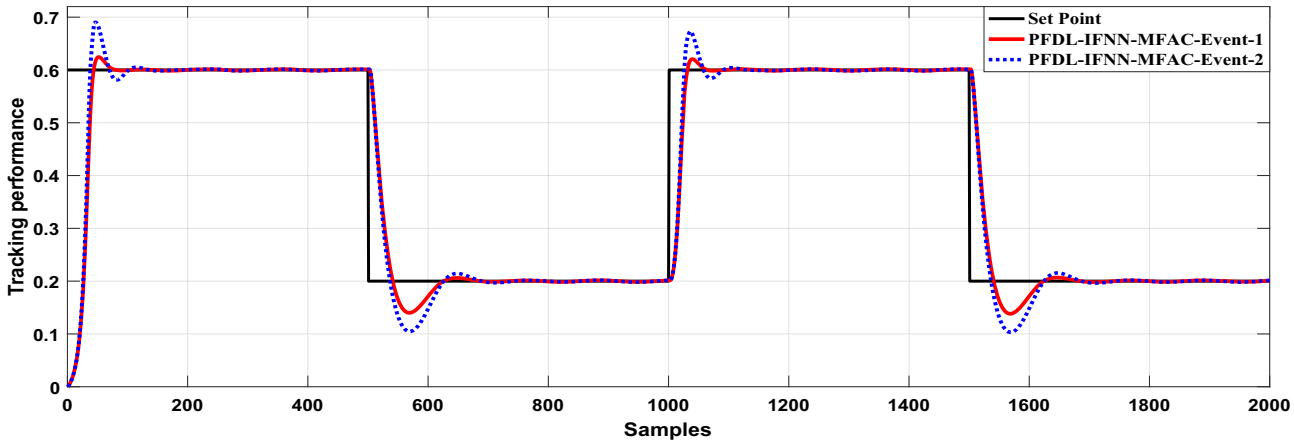


Fig. 7 Tracking response of the CSTR system using PFDL-IFNN-MFAC-Event-1 and PFDL-IFNN-MFAC-Event-2 (case: 1)

the CFDL/PFDL-IFNN-MFAC-Event-2 but the second algorithm has a significant reduction in the computational burden, energy consumption, and communication resources.

Figure 19 shows that out of 2000 sample instants, the first case had a total of 405/394 events in the case of CFDL/PFDL-IFNN-MFAC-Event-1 and 125/122 when the CFDL/PFDL-IFNN-MFAC-Event-2 are applied. In comparison to a standard periodic sampled discrete-time control, it exhibits the efficiency of these schemes in reducing the number of transmissions I/O data over the network. When compared to the controller in [34] in case (1), which uses neural network ETC-MFAC and takes a total of 884 events, it is clear that the proposed controllers provide better performance in terms of system resource saving than the controller presented in [34]. We must notice that case (2) is not valid in [34].

Example 2: A steam-water heat exchanger.

In this example, the PFDL-IFNN-MFAC and the CFDL-IFNN-MFAC are used to control a steam-water heat exchanger to validate the proposed MFAC-PETC methods. A Hammerstein model may be used to express the heat exchanger dynamics, which can be written as [34]:

$$G_H(z) = \frac{y(z)}{N_u(z)} = \frac{1.2z - 0.1}{z^2 - 0.6z + 0.1} \tag{79}$$

$$N_u = 1.5u - 1.5u^2 + 0.5u^3 \tag{80}$$

The proposed controllers are performed by using IFNN has the following parameters $N = 5$, $cm = [0.20\ 0.40\ 0.06\ 0.81]^T$, $cn = [1.4\ 1.6 - 0.6 - 0.4 - 0.2]^T$, $d_m = 10$, $and\, d_n = 2.5$. The CFDL-IFNN-MFAC law (31) is implemented with $p_1 = 1$, and $\lambda = 0.6$, on the other side; the PFDL-IFNN-MFAC (32) is implemented with $L = 3$, $p_1 = p_2 = p_3 = 1$, and $\lambda = 0.4$. For the event-triggering Algorithm (2), the parameter Q is set to 2. Two simulation cases are introduced to demonstrate the performance of the proposed control schemes. Due to there being no observed difference between the response of the CFDL-IFNN-MFAC-Event-1 and the PFDL-IFNN-MFAC-Event-1 with the second example, only the results of PFDL-IFNN-MFAC-Event-1 and CFDL/PFDL-IFNN-MFAC-Event-2 are presented here.

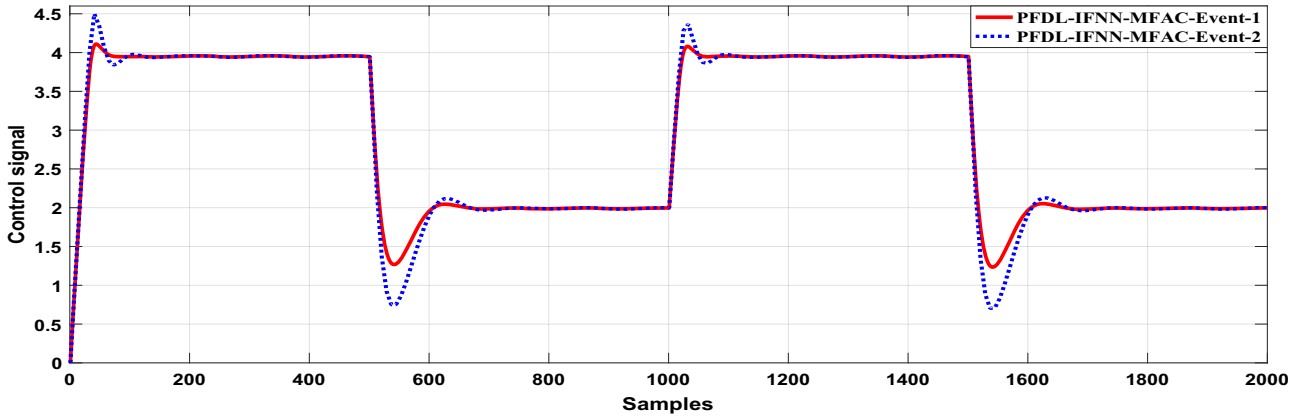


Fig. 8 Control signals for PFDL-IFNN-MFAC-Event-1 and PFDL-IFNN-MFAC-Event-2 with the CSTR system (case: 1)

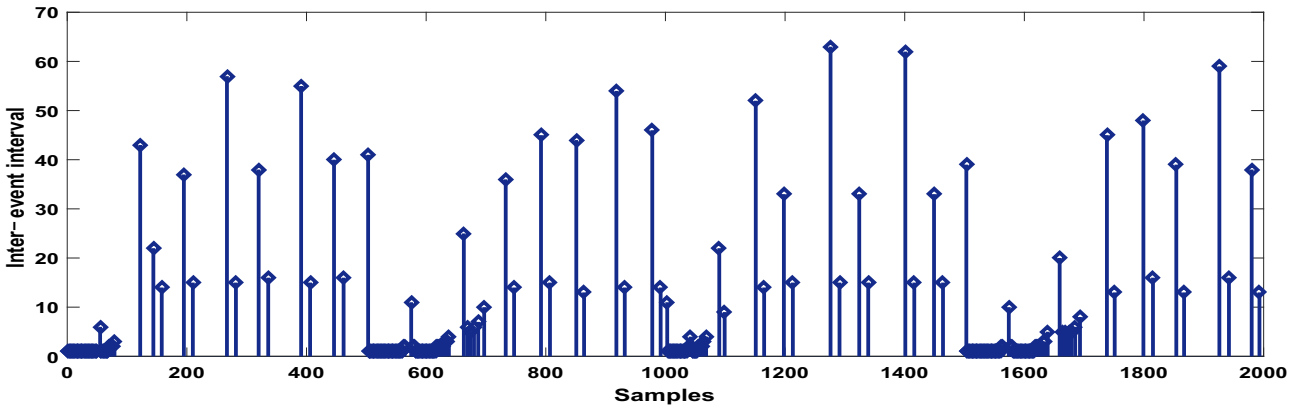


Fig. 9 Inter-event intervals for PFDL-IFNN-MFAC-Event-1 with the CSTR system (case: 1)

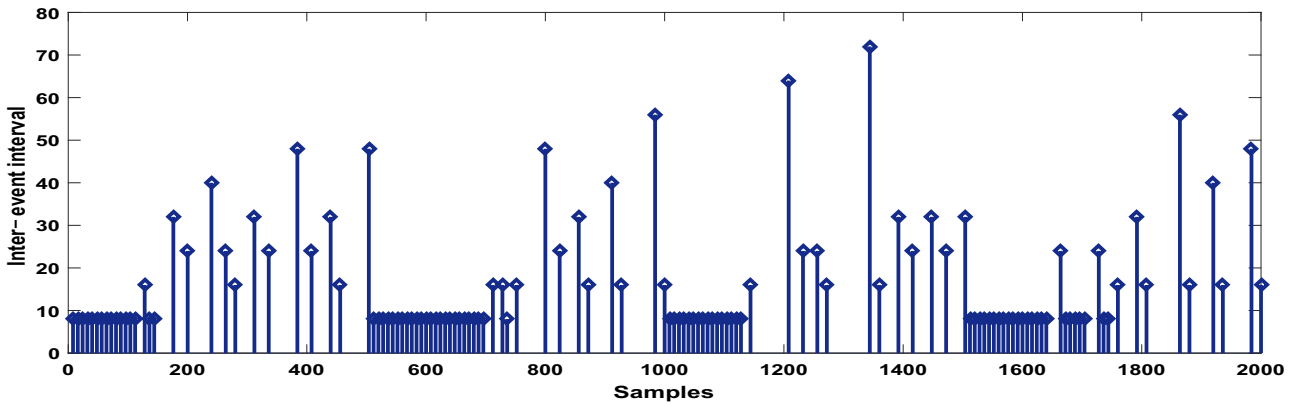


Fig. 10 Inter-event intervals for PFDL-IFNN-MFAC-Event-2 with the CSTR system (case: 1)

Case 1: Tracking response for a set-point change. In this case, the desired trajectory to be tracked is defined as:

$$y^*(k+1) = \begin{cases} 1, & k < 100 \\ 2, & 100 \leq k < 200 \\ 3.5, & 200 \leq k < 300 \\ 1.5, & k \geq 300. \end{cases} \quad (81)$$

Figure 20 shows the tracking performances of the output using PFDL-IFNN-MFAC-Event-1 and CFDL/ PFDL-IFNN-MFAC-Event-2 and Fig. 21 shows the obtained control signals for the controllers. The inter-event intervals can be seen in Figs. 22, 23. In this case, it is shown clearly that the response under the PFDL -IFNN-MFAC-Event-1 and Event-2 are faster but with greater overshoots than CFDL-IFNN-MFAC-Event-2.

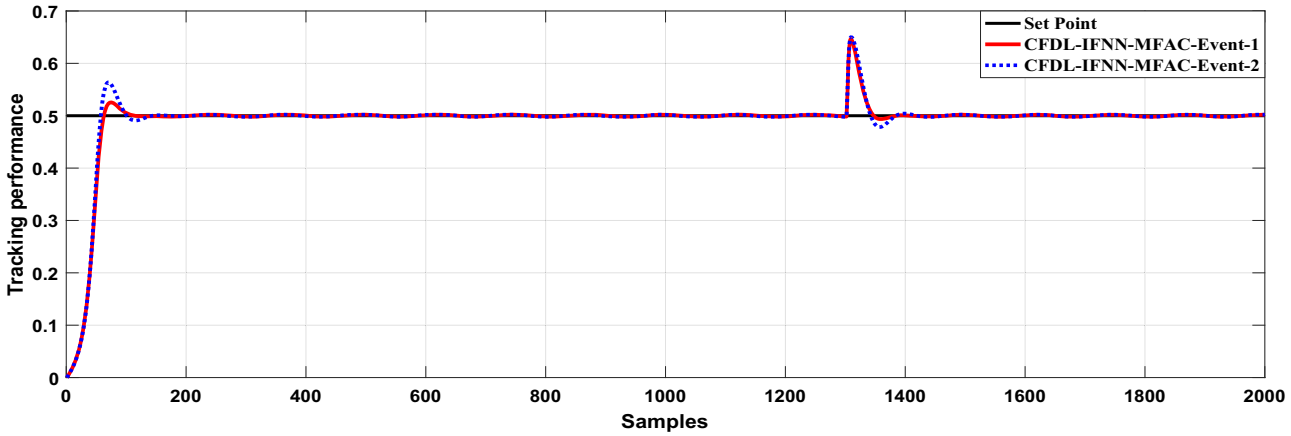


Fig. 11 Tracking response of the CSTR system using CFDL-IFNN-MFAC-Event-1 and CFDL-IFNN-MFAC-Event-2 (case: 2)

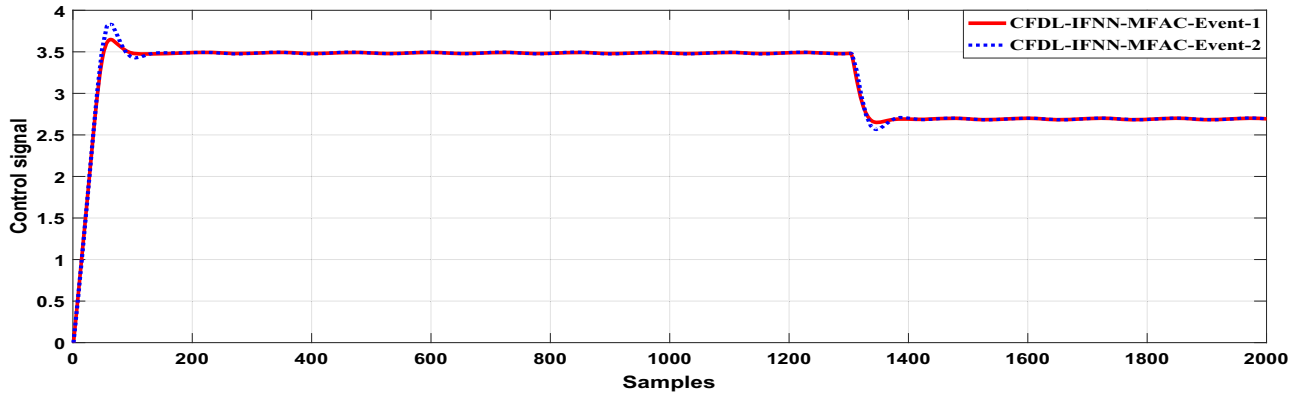


Fig. 12 Control signals for CFDL-IFNN-MFAC-Event-1 and CFDL-IFNN-MFAC-Event-2 with the CSTR system (case: 2)

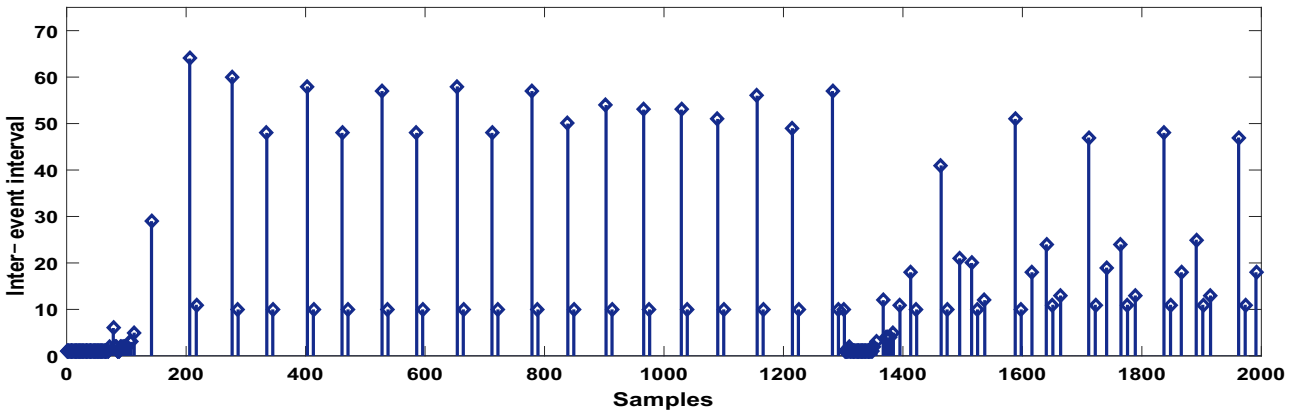


Fig. 13 Inter-event intervals for CFDL-IFNN-MFAC-Event-1 with the CSTR system (case: 2)

Case 2: Tracking response under external output disturbance In this case, the proposed controllers are tested under 15% external disturbance added to the system output at $k = 150$ as shown in Fig. 24, 25. The inter-event intervals are shown in Figs. 26, 27.

In case 2, it is obvious that the PFDL-IFNN-MFAC-Event-1 and Event-2 controllers can deal with the time-varying system parameters and overcome their effect in a shorter time than the CFDL-IFNN-MFAC-Event-2.

The simulation results show that PFDL-IFNN-MFAC-Event-1 and Event-2 gives faster response with smaller errors and lower number of events when compared with the CFDL-IFNN-MFAC-Event-2 as shown in Table 2, so the PFDL-IFNN-MFAC has a better performance when deal

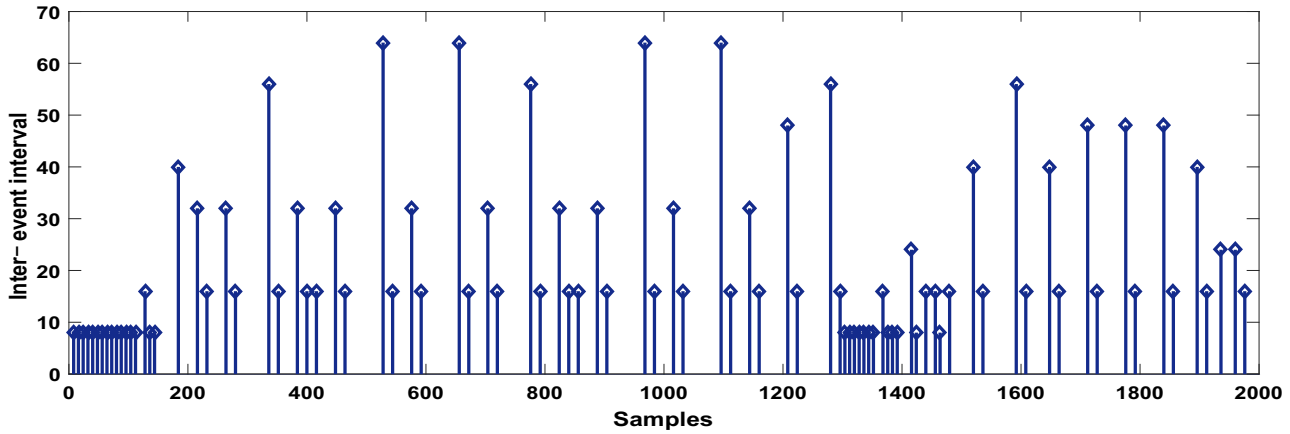


Fig. 14 Inter-event intervals for CFDL-IFNN-MFAC-Event-2 with the CSTR system (case: 2)

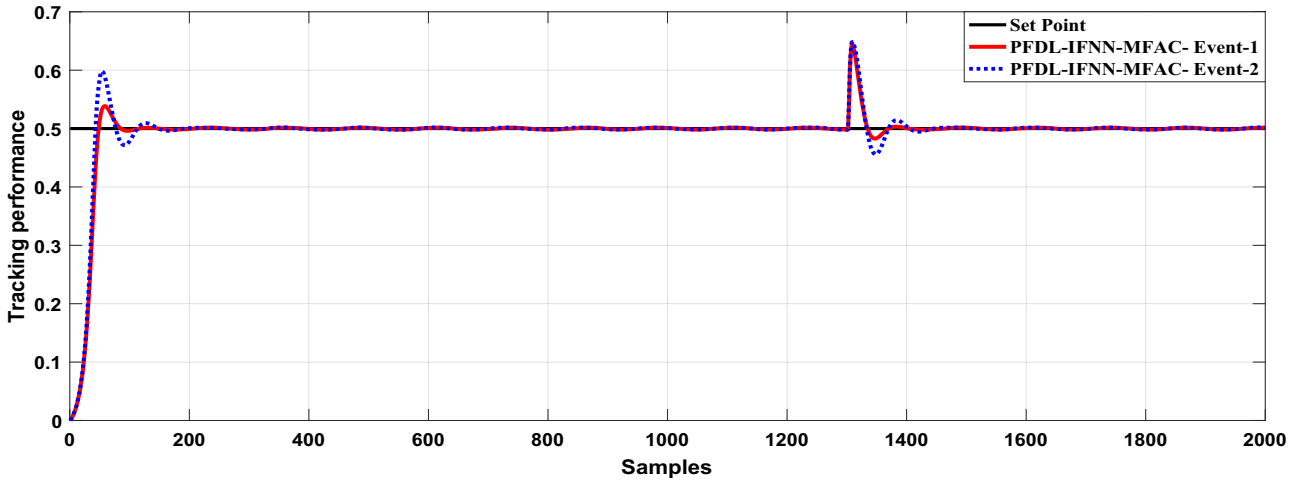


Fig. 15 Tracking response of the CSTR system using PFDL-IFNN-MFAC-Event-1 and PFDL-IFNN-MFAC-Event-2 (case: 2)

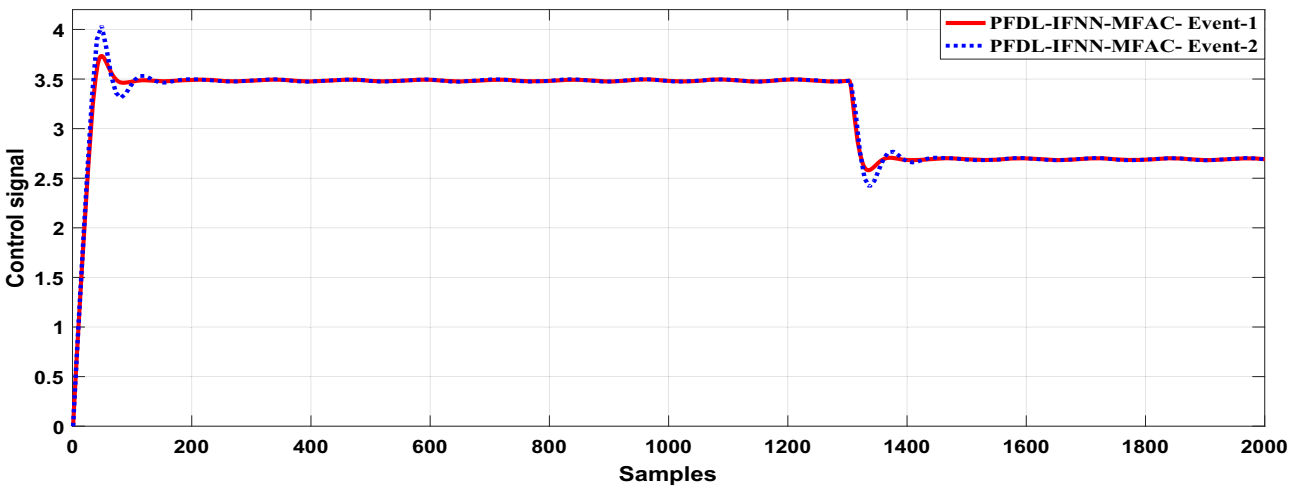


Fig. 16 Control signals for PFDL-IFNN-MFAC-Event-1 and PFDL-IFNN-MFAC-Event-2 with the CSTR system (case: 2)

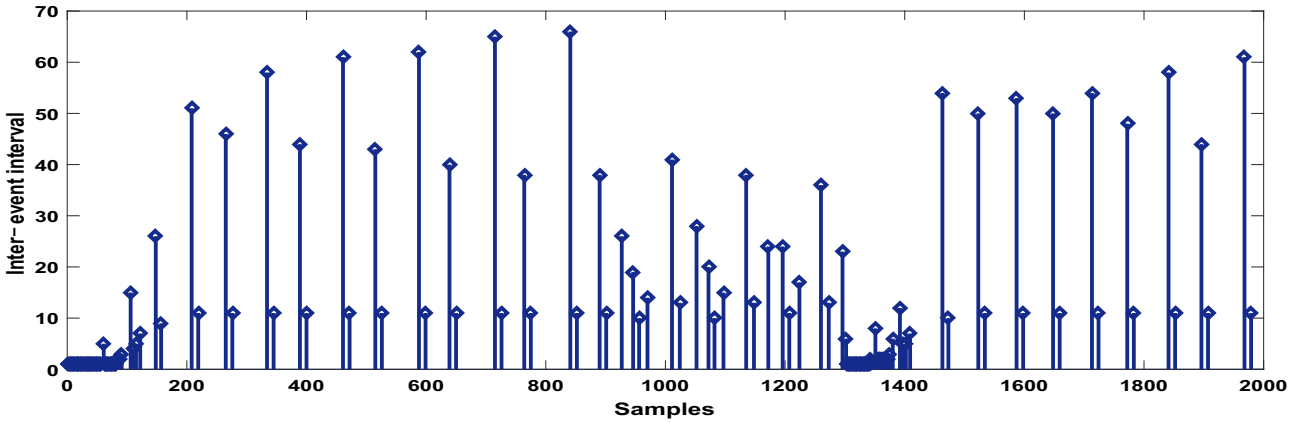


Fig. 17 Inter-event intervals for PFDL-IFNN-MFAC-Event-1 with the CSTR system (case: 2)

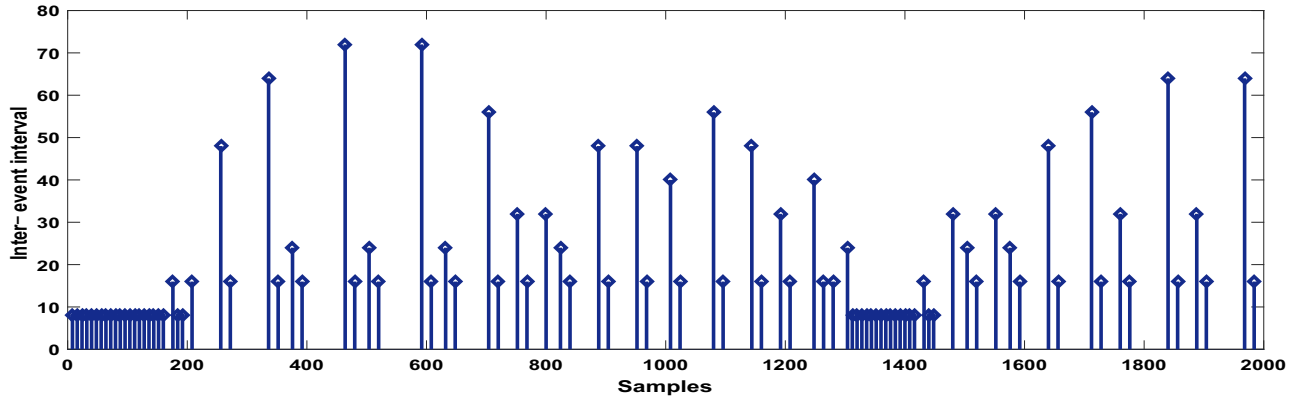


Fig. 18 Inter-event intervals for PFDL-IFNN-MFAC-Event-2 with the CSTR system (case: 2)

Table 1 The RMSE, ISE and IAE indices for CSTR using the proposed controllers

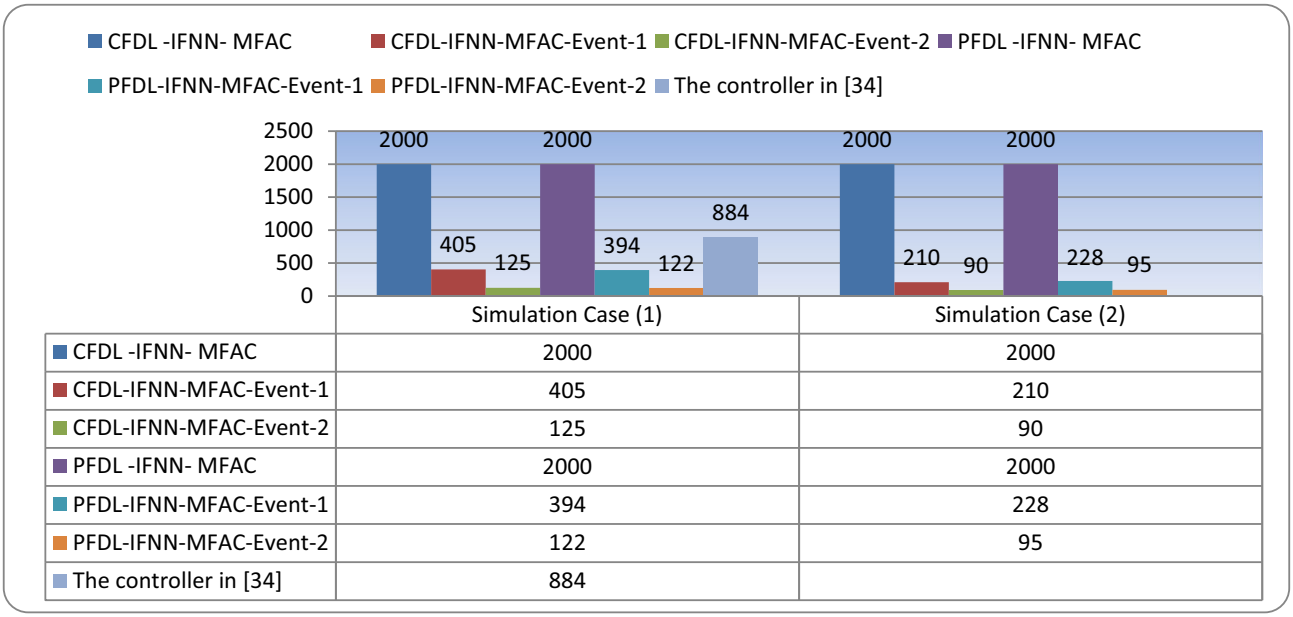
Controller	Simulation case (1)			Simulation case (2)		
	RMSE	ISE	IAE	RMSE	ISE	IAE
CFDL-IFNN-MFAC	0.0949	18.0056	52.4440	0.0650	8.4456	25.4420
CFDL-IFNN-MFAC-Event-1	0.0949	18.0097	52.9180	0.0650	8.4458	26.1326
CFDL-IFNN-MFAC-Event-2	0.0956	18.2911	58.3091	0.0653	8.5308	27.8109
PFDL-IFNN-MFAC	0.0910	16.5670	50.2626	0.0598	7.1485	21.9782
PFDL-IFNN-MFAC-Event-1	0.0910	16.5735	50.6773	0.0598	7.1492	22.6228
PFDL-IFNN-MFAC-Event-2	0.0912	16.6168	56.0452	0.0604	7.2898	26.6838

with nonlinear heat exchanger. As mentioned before the first algorithm gives faster response with smaller overshot when compared with the second algorithm but the second algorithm have a significant reduction in the number of events as shown in Fig. 28.

In Fig. 28, comparison between fixed periodic sampling MFAC, and the proposed MFAC-PETC is shown, significance saving in the number of computational load and transmission data over the network. Furthermore, the comparison with the controller in [34] which takes a total of 124 events in case 1, it is clear that the proposed controllers provide better performance in terms of system resources saving than the controller presented in [34]. This will significantly reduce the communication cost. We must notice that case (2) is not valid in [34].

Practical results

In this subsection, the proposed controllers are applied on a real system. The real system is a shunt-wound DC machine



	Simulation Case (1)	Simulation Case (2)
CFDL -IFNN- MFAC	2000	2000
CFDL-IFNN-MFAC-Event-1	405	210
CFDL-IFNN-MFAC-Event-2	125	90
PFDL -IFNN- MFAC	2000	2000
PFDL-IFNN-MFAC-Event-1	394	228
PFDL-IFNN-MFAC-Event-2	122	95
The controller in [34]	884	

Fig. 19 Comparison of computational load between MFAC, MFAC-Event-1, and MFAC-Event-2 for CSTR

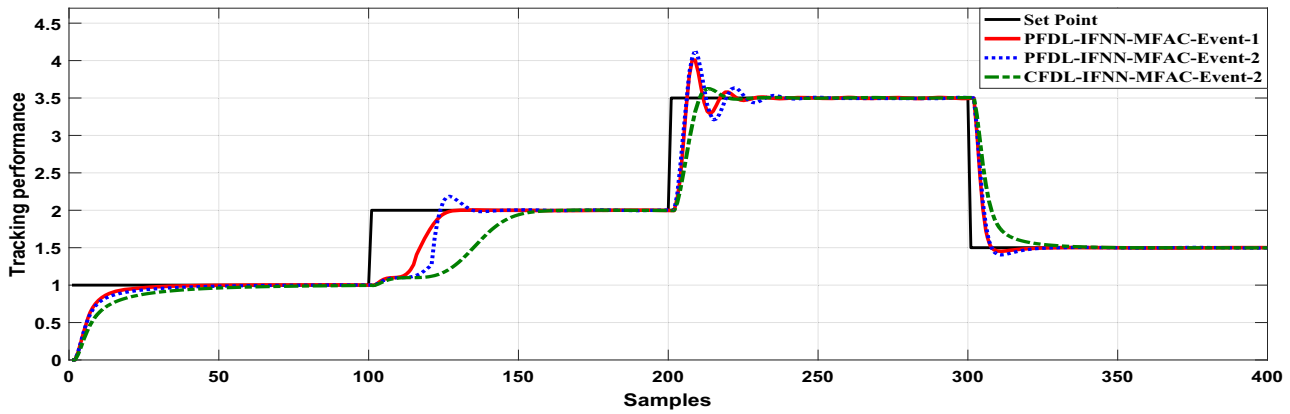


Fig. 20 Tracking response of the heat exchanger using the proposed controllers (case: 1)

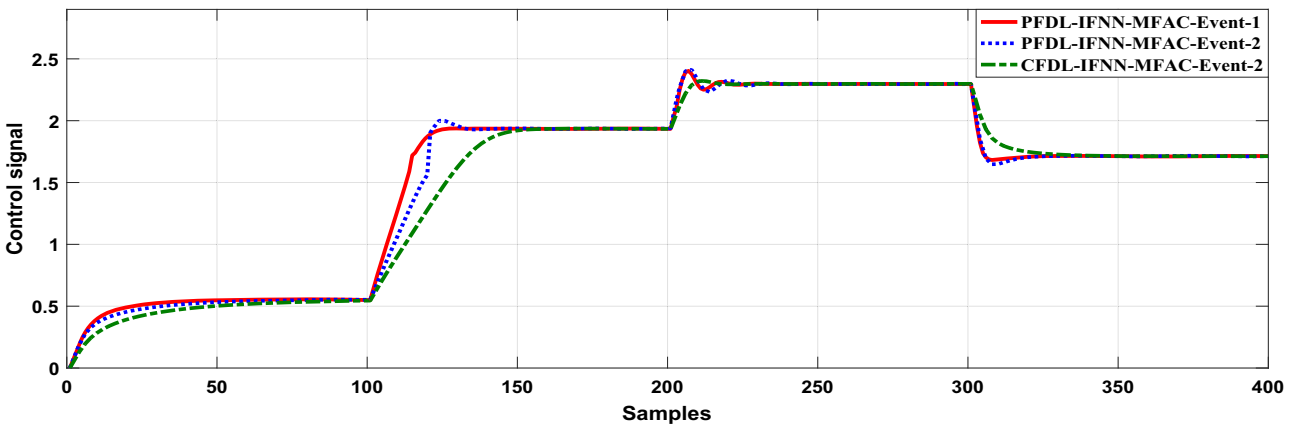


Fig. 21 Control signals of the heat exchanger using the proposed controllers (case: 1)

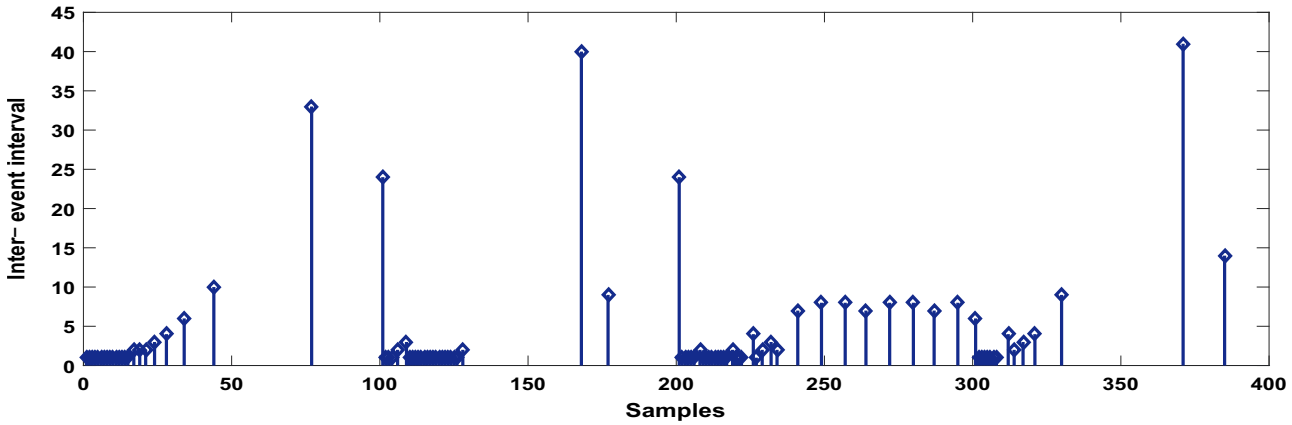


Fig. 22 Inter-event intervals for PFDL-IFNN-MFAC-Event-1 (case: 1)

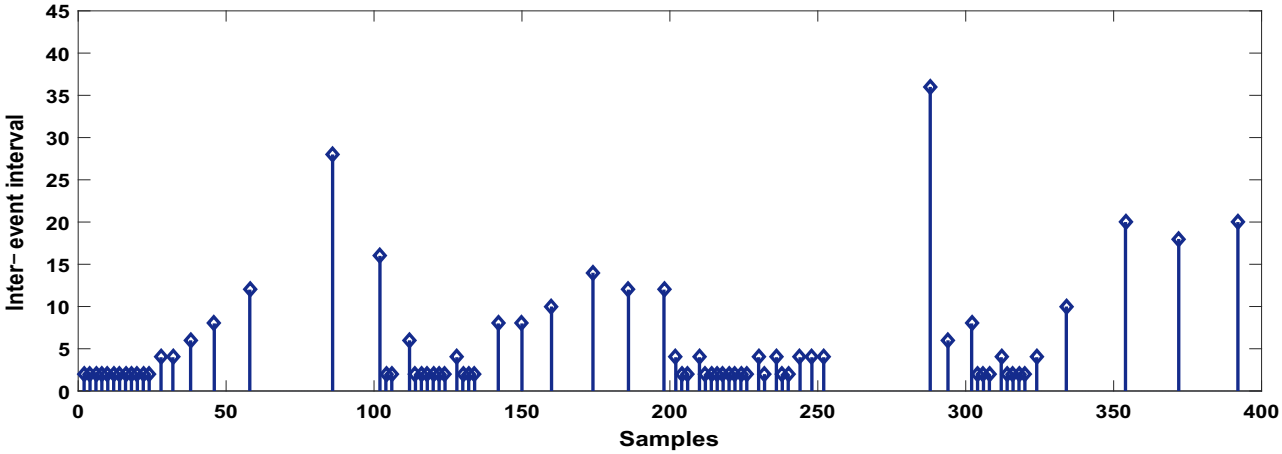


Fig. 23 Inter-event intervals for PFDL-IFNN-MFAC-Event-2 (case: 1)

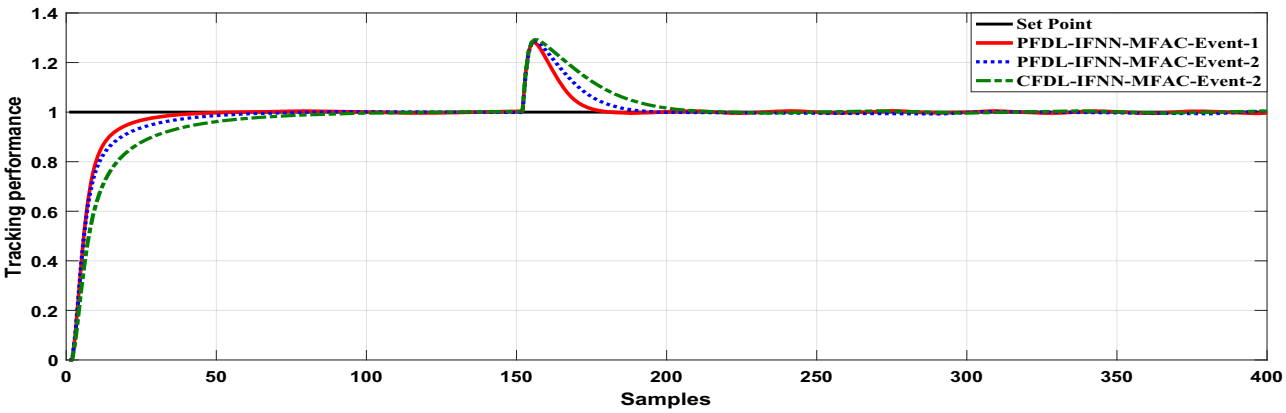


Fig. 24 Tracking response of the heat exchanger using the proposed controllers (case: 2)

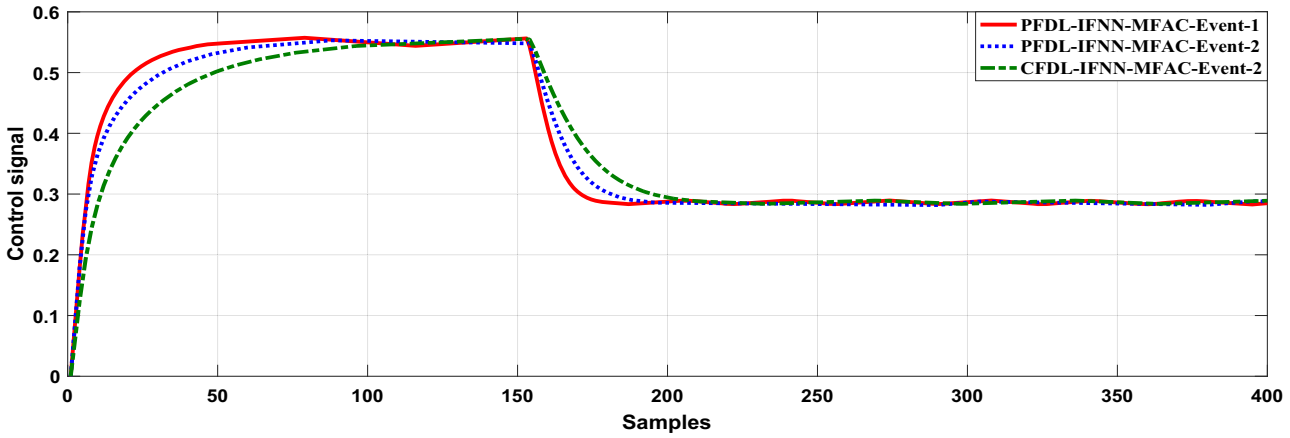


Fig. 25 Control signals of the heat exchanger using the proposed controllers (case: 2)

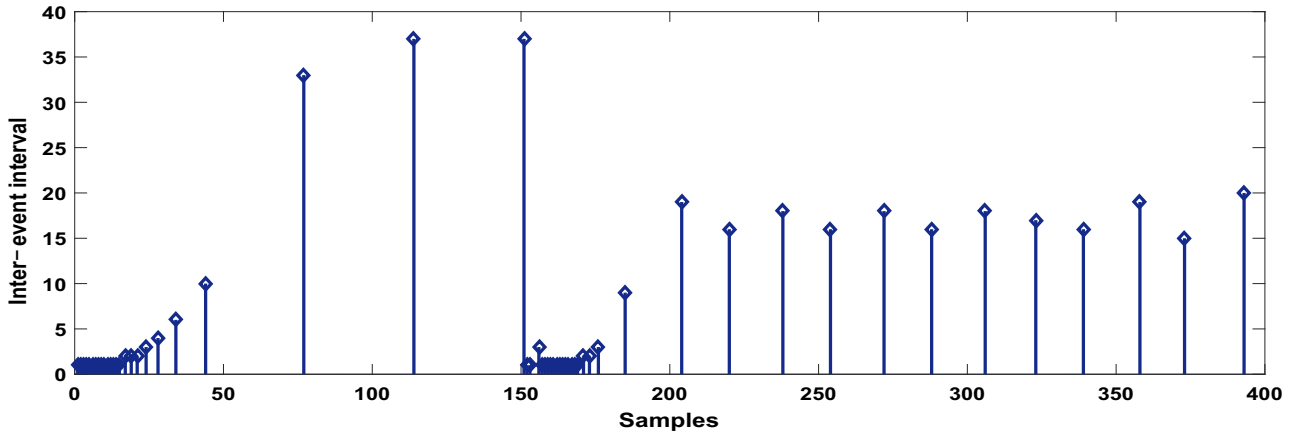


Fig. 26 Inter-event intervals for PFDL-IFNN-MFAC-Event-1 (case: 2)

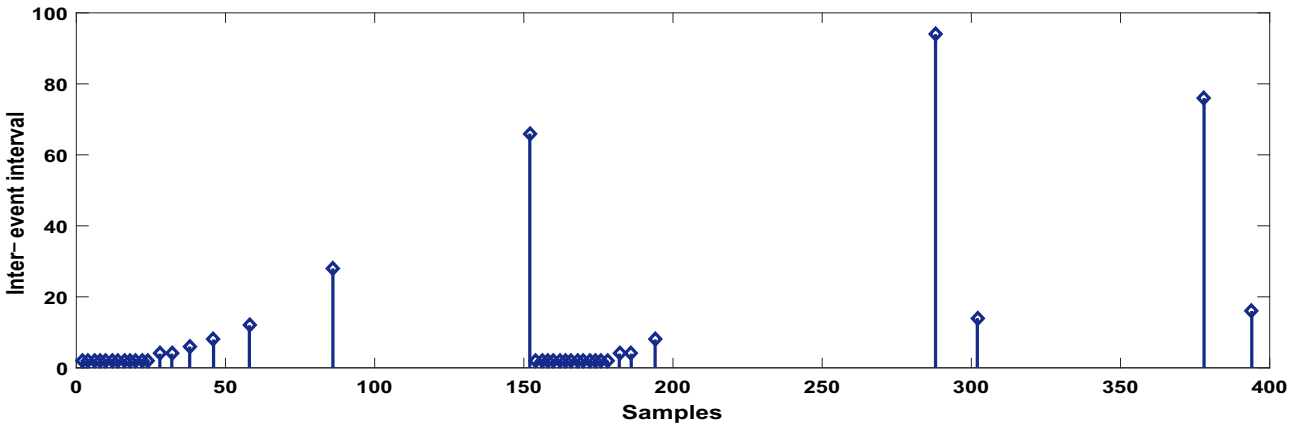


Fig. 27 Inter-event intervals for PFDL-IFNN-MFAC-Event-2 (case: 2)

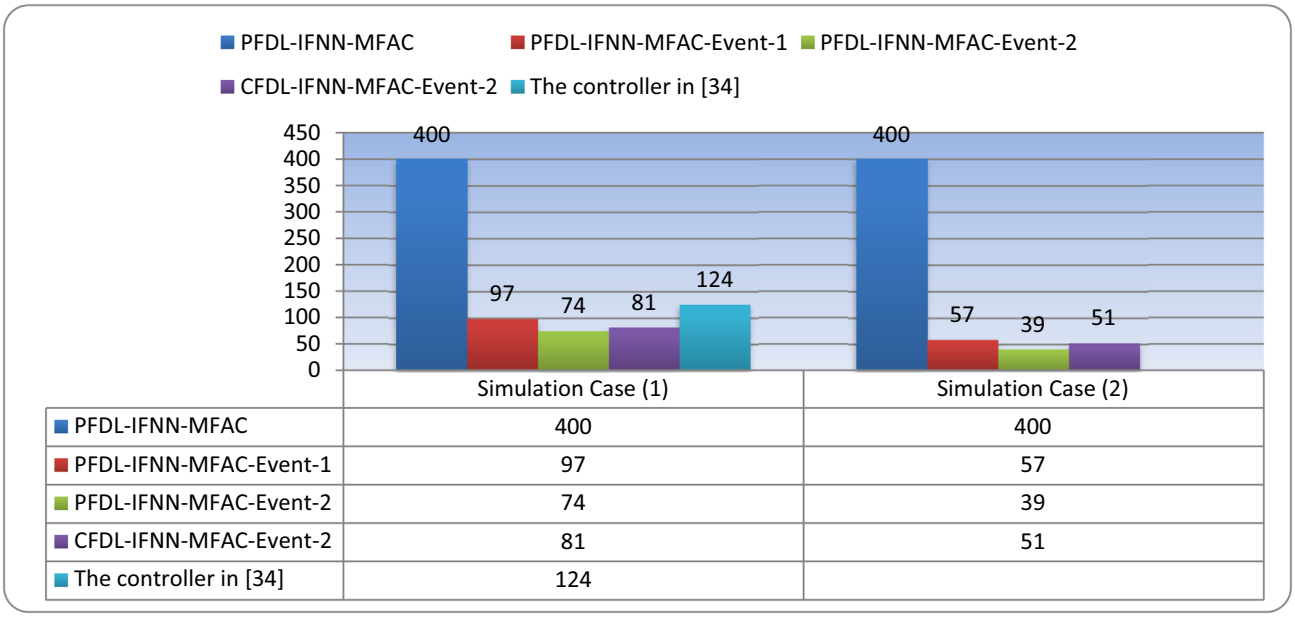


Fig. 28 Comparison of computational load between MFAC, MFAC-Event-1, and MFAC-Event-2 for heat exchanger

Table 2 The RMSE, ISE, and IAE for heat exchanger using the proposed controllers

Controller	Simulation Case (1)			Simulation Case (2)		
	RMSE	ISE	IAE	RMSE	ISE	IAE
PFDL-IFNN-MFAC	0.3073	37.7656	40.1799	0.1121	5.0303	10.5299
PFDL-IFNN-MFAC-Event-1	0.3088	38.1411	41.2978	0.1124	5.0567	11.2325
PFDL-IFNN-MFAC-Event-2	0.3272	42.8124	48.9480	0.1171	5.4878	13.8712
CFDL-IFNN-MFAC-Event-2	0.3839	58.9475	68.1530	0.1350	7.2909	19.3313

$(0.1KW)$. This machine contains two DC motors connected to each other, one for controlling the rotation speed and the other works as a DC generator. The speed of the first motor is controlled by varying the armature voltage. A tacho-generator with an output voltage of $1V/1000\ r.p.m$ is used for the speed measurement. A drive circuit with input varies from 0 to 10 V is used as an actuator. The controller is interfaced to the system through a Mega Arduino. The experimental set-up for the real system application is depicted in Fig. 29. In the following experiments, the proposed CFDL/PFDL-IFNN-MFAC is tested via the speed control of the DC machine.

To investigate the controller performance with a real-time application, two experiments are considered. In the experiments, the PFDL-IFNN-MFAC (32) is used with $L = 3$, $p_1 = p_2 = p_3 = 1$, $\lambda = 0.15$, and the PPD vector uses IFNN with $cm = [0.20\ 0.40\ 0.06\ 0.81]^T$, $cn = [1.4\ 1.6 - 0.6 - 0.4 - 0.2]^T$, $N = 5$, $d_m = 1$, $d_n = 1.5$, $W(0) = 0.01rand \times I_{3\times 10}$. The triggering parameter is chosen $\delta = 0.06$ and for the second algorithm $Q = 10$.

Experiment 1: Tracking response for set-point change. In this experiment, the desired reference is step with different levels $1V$ (i.e., 1000 r.p.m) during the first 500 samples, then $1.5V$(i.e.,1500 r.p.m) during the second 500 samples, then $2V$ (i.e., 2000 r.p.m) during the third 500 samples, and then $1.5V$ (i.e., 1500 r.p.m) within the last 500 samples. The tracking performance and control signal of the DC machine based on the PFDL-IFNN-MFAC-Event-1, and PFDL-IFNN-MFAC-Event-2 are shown in Figs. 30, 31. It is clear that the system response under the PFDL-IFNN-MFAC-Event-1 is faster than PFDL-IFNN-MFAC-Event-2.

Experiment 2: Tracking response under external output disturbance. This experiment is performed to examine the tracking performance of the proposed controller in the presence of load changes. This experiment is conducted using a desired speed = 1000 r.p.m (i.e., $1V$). The load change is applied at sampling instant $k = 700$. Figures 32 and 33 show the speed response, and the control signal for this experiment, respectively. It is clear that the two control algorithms can control the speed of the real system with satisfactory performance. PFDL-IFNN-MFAC-Event-2 gives slower response

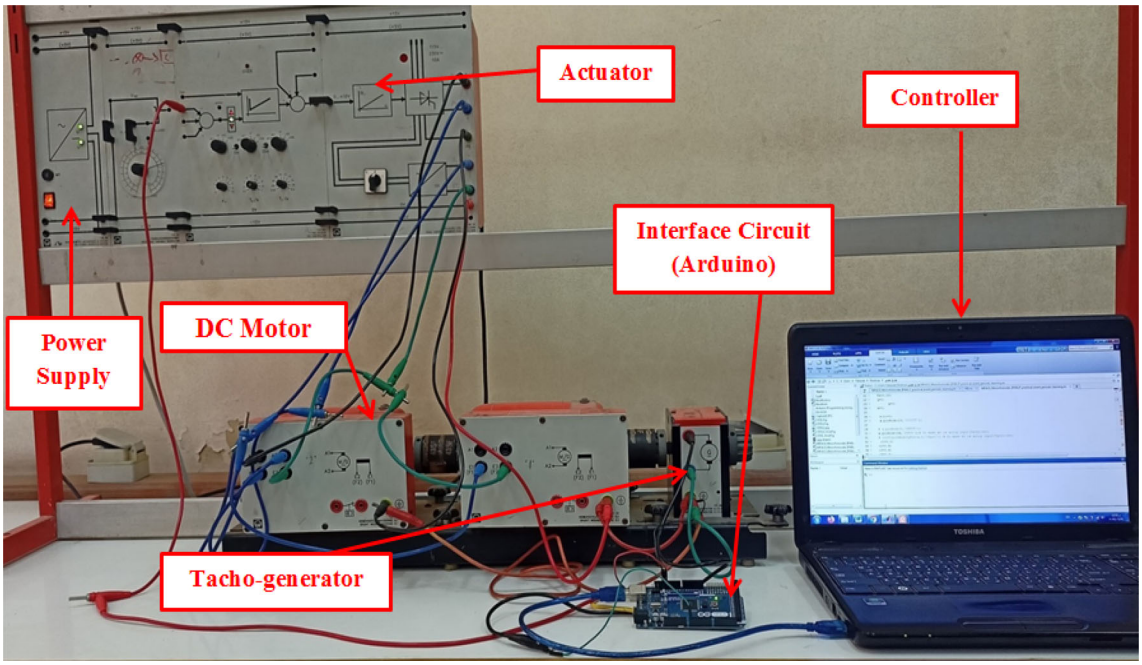


Fig. 29 Actual view of the practical system

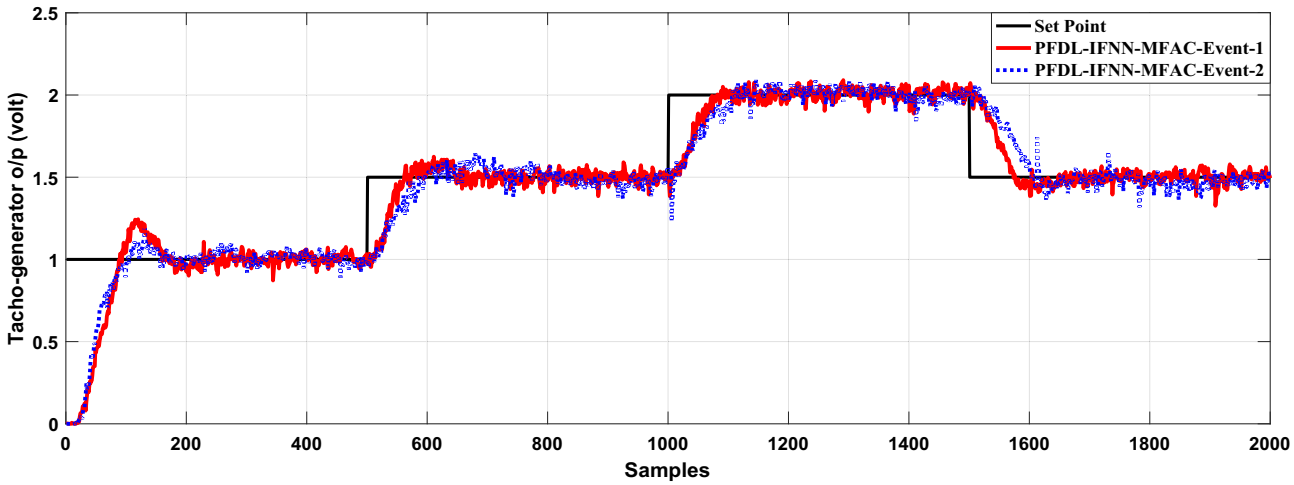


Fig. 30 The speed response using PFDL-IFNN-MFAC-Event-1 and PFDL-IFNN-MFAC-Event-2 (Experiment 1)

compared with the PFDL-IFNN-MFAC-Event-1 but with significant reduction in the computation and communication resources.

For comparison the CFDL-IFNN-MFAC (31) is used with $L = 1$, $p_1 = 1$, $\lambda = 0.2$, and the PPD uses IFNN with $cm = [0.2\ 0.4\ 0.6\ 0.8\ 1]^T$, $cn = [1.4\ 1.6 - 0.6 - 0.4 - 0.2]^T$, $W(0) = 0.01rand \times I_{1\times 10}$, $N = 5$, $d_m = 1$, $d_n = 1.5$. The triggering parameter is chosen $\delta = 0.06$ and for the second algorithm $Q = 10$.

Table 3 and Fig. 34 show the results of both the CFDL-IFNN-MFAC-Event-1 and Event-2 and PFDL-IFNN-MFAC-Event-1 and Event-2 for the two experiments. The practical results show that the PFDL-IFNN-MFAC gives better results when compared with the CFDL-IFNN-MFAC and also shows the significance reduction in the computational load and data transmission over the network when using the second algorithm which supports the previous simulation results.

Conclusion

This paper has developed an MFAC-PETC based on IFNN for controlling nonlinear systems. The proposed technique uses the MFAC based on historical I/O data only to establish an

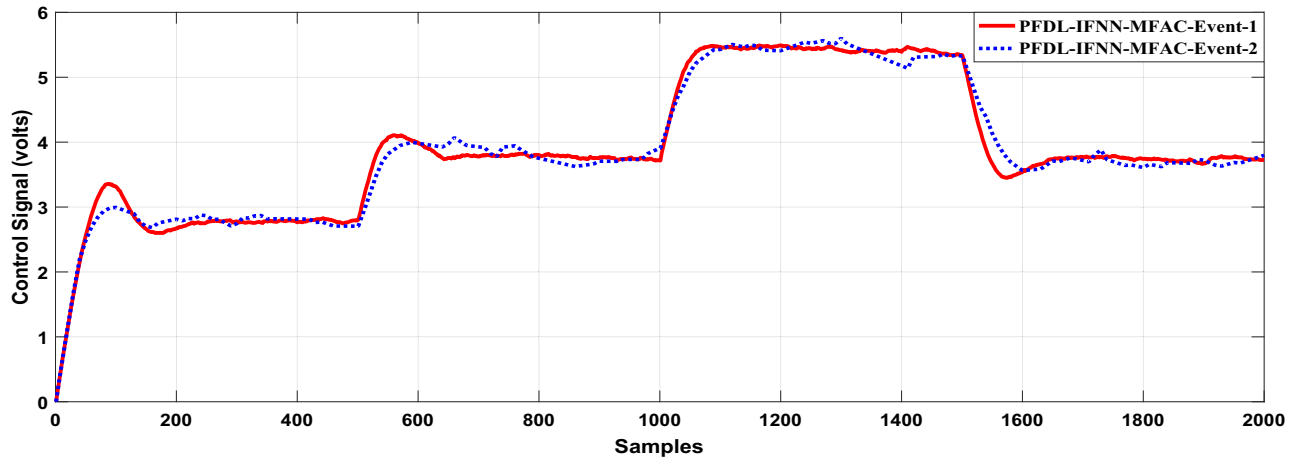


Fig. 31 The control signal applied to the motor drive using the proposed controllers (Experiment 1)

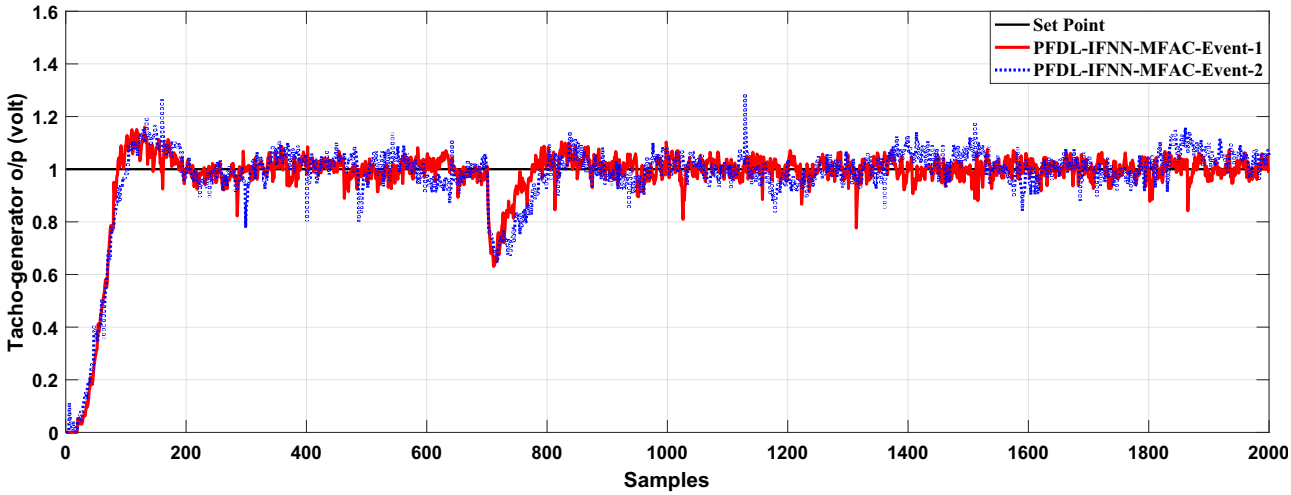


Fig. 32 The speed response using PFDL-IFNN-MFAC-Event-1 and PFDL-IFNN-MFAC-Event-2 (Experiment 2)

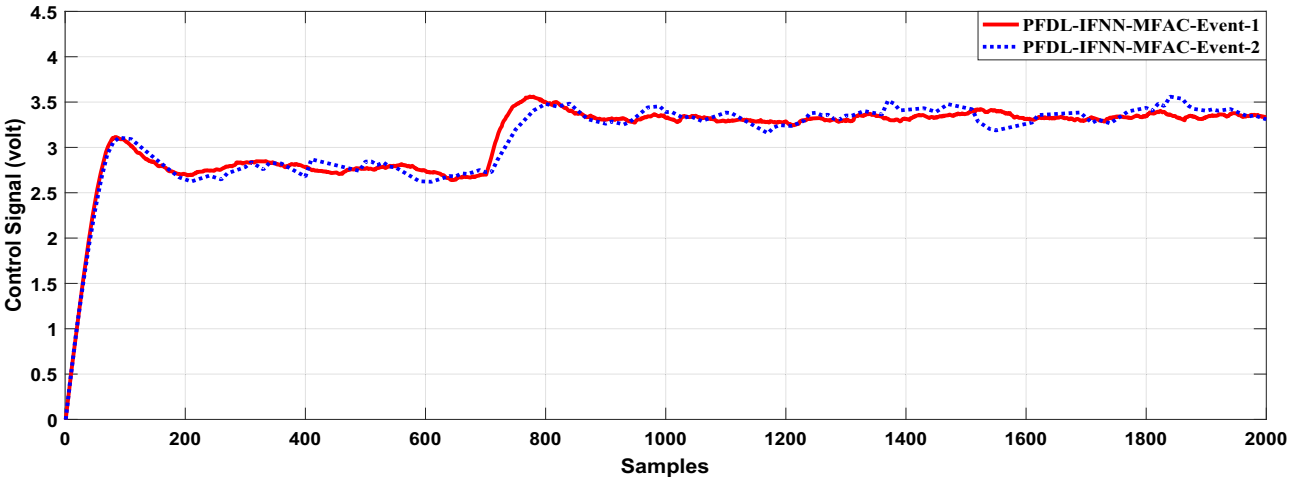
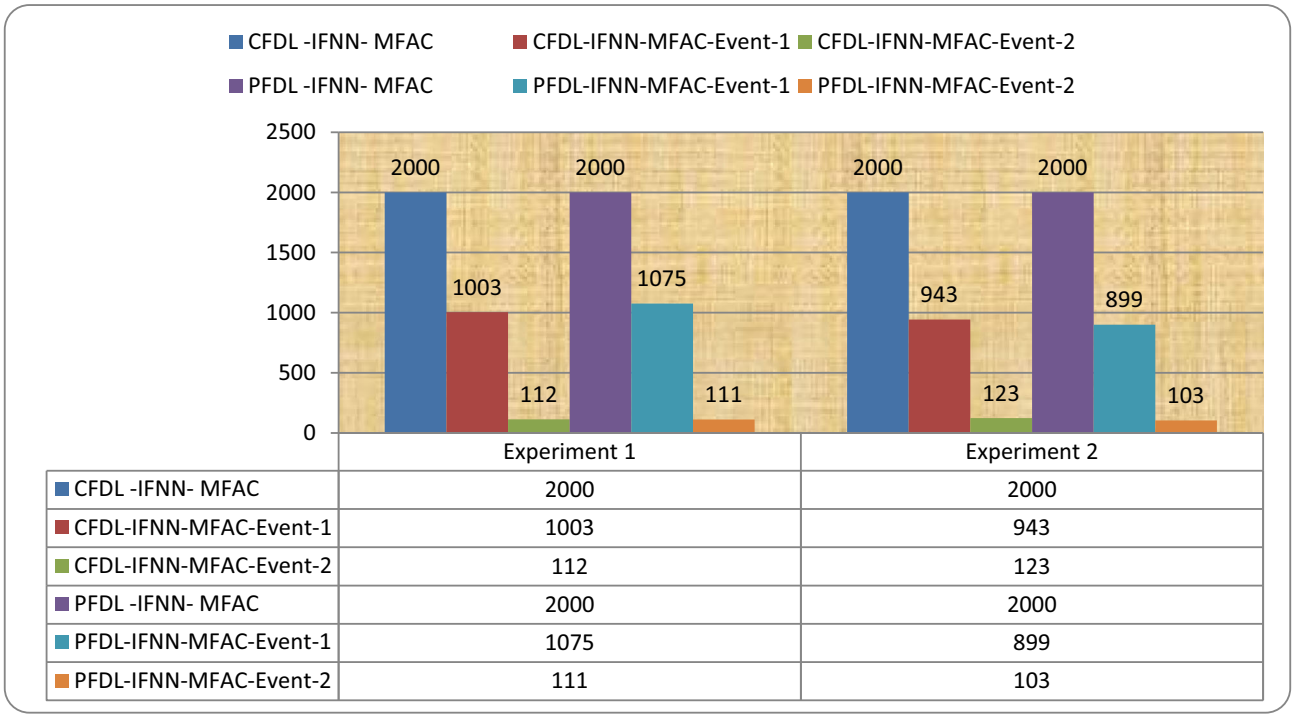


Fig. 33 The control signal applied to the motor drive with the proposed controllers (Experiment 2)



	Experiment 1	Experiment 2
CFDL -IFNN- MFAC	2000	2000
CFDL-IFNN-MFAC-Event-1	1003	943
CFDL-IFNN-MFAC-Event-2	112	123
PFDL -IFNN- MFAC	2000	2000
PFDL-IFNN-MFAC-Event-1	1075	899
PFDL-IFNN-MFAC-Event-2	111	103

Fig. 34 Comparison of computational load between MFAC, MFAC-Event-1, and MFAC-Event-2 for the DC motor

Table 3 The RMSE, ISE, and IAE for the DC motor using the proposed controllers

Controller	Experiment 1			Experiment 2		
	RMSE	ISE	IAE	RMSE	ISE	IAE
CFDL-IFNN-MFAC	0.2062	85.0589	217.6686	0.1772	62.7796	163.5974
CFDL-IFNN-MFAC-Event-1	0.2091	87.4041	223.5337	0.1788	63.9571	168.5240
CFDL-IFNN-MFAC-Event-2	0.2130	90.7313	235.3822	0.1826	66.7132	173.7231
PFDL-IFNN-MFAC	0.1890	71.4577	201.1554	0.1656	54.8490	159.1659
PFDL-IFNN-MFAC-Event-1	0.1951	76.1570	208.9482	0.1727	59.6740	164.6372
PFDL-IFNN-MFAC-Event-2	0.1985	78.7939	219.7478	0.1762	62.1110	169.0670

approximate model of the nonlinear system, where the MFAC creates a series of equivalent local dynamic linearization data models at each operating point of the closed-loop system. The equivalent dynamic-linearization model at each sample is estimated using the IFNN based on the I/O measurement data of the controlled plant. Two DLT methods; CFDL and PFDL are used in this paper. Two periodic event-triggering control mechanisms have been used to save communication resources and reduce computational burden and energy consumption. The BIBO and Lyapunov stability is studied to prove that the stability of the whole closed-loop system is guaranteed via the proposed controllers in the presence of external disturbance and time-varying system parameters. The simulation and the practical results reveal the effectiveness and flexibility of the proposed technique.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

References

1. Hamdy M, Abd-Elhaleem S, Fkirin MA (2018) Adaptive fuzzy predictive controller for a class of networked nonlinear systems with time-varying delay. IEEE Trans Fuzzy Syst 26(4):2135–2144. https://doi.org/10.1109/TFUZZ.2017.2764851
2. Ding D, Han Q-L, Ge X, Wang J (2021) Secure state estimation and control of cyber-physical systems: A survey. IEEE Trans Syst Man Cybern Syst 51(1):176–190
3. Zhang DW, Liu GP, Cao L (2022) Proportional integral predictive control of high-order fully actuated networked multiagent systems with communication delays. IEEE Trans Syst Man Cybern Syst 15:10. https://doi.org/10.1109/TSMC.2022.3188504
4. Deng C, Che W-W, Wu Z-G (2021) A dynamic periodic event-triggered approach to consensus of heterogeneous linear multiagent systems with time-varying communication delays. IEEE Trans Cybern 51(4):1812–1821
5. Li T, Yang D, Xie X, Zhang H (2022) Event-triggered control of nonlinear discrete-time system with unknown dynamics based on HDP(λ). IEEE Trans Cybern 52(7):6046–6058. https://doi.org/10.1109/TCYB.2020.3044595
6. Lin N, Chi R, Huang B, Hou Z (2020) Event-triggered nonlinear iterative learning control. IEEE Trans Neural Netw Learn Syst 32(11):5118–5128
7. Ma YS, Che WW, Deng C (2022) Event-triggered model-free adaptive control for nonlinear cyber-physical systems with false data injection attacks. Int J Robust Nonlinear Control 32(4):2442–2452
8. Lin N, Chi R, Huang B (2021) Event-triggered model-free adaptive control. IEEE Trans Syst Man Cybern 51(6):3358–3369
9. Bu X, Yu W, Yu Q, Hou Z, Yang J (2021) Event-triggered model-free adaptive iterative learning control for a class of nonlinear systems over fading channels. IEEE Trans Cybern 11:1–12. https://doi.org/10.1109/TCYB.2021.3058997
10. Wang Y, Qiu X, Zhang H, Xie X (2021) Data-driven-based event-triggered control for nonlinear CPSs against jamming attacks. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2020.3047931
11. Qi Y, Zhao X, Huang J (2022) Data-driven event-triggered control for switched systems based on neural network disturbance compensation. Neurocomputing 490:370–379
12. Sun J, Yang J, Zeng Z (2022) Predictor-based periodic event-triggered control for nonlinear uncertain systems with input delay. Automatica 136:110055
13. Li S, Ahn CK, Guo J, Xiang Z (2021) Neural-network approximation-based adaptive periodic event-triggered output-feedback control of switched nonlinear systems. IEEE Trans Cybern 51(8):4011–4020
14. Wang W, Postoyan R, Nešić D, Heemels W (2020) Periodic event-triggered control for nonlinear networked control systems. IEEE Trans Autom Control 65(2):620–635
15. Guan Y, Han QL, Yao H, Ge X (2018) Robust event-triggered H-infinity controller design for vehicle active suspension systems. Nonlinear Dyn 94:627–638
16. Abd-Elhaleem S, Soliman M, Hamdy M (2022) Modified repetitive periodic event-triggered control with equivalent-input-disturbance for linear systems subject to unknown disturbance. Int J Control 95(7):1825–1837. https://doi.org/10.1080/00207179.2021.1876924
17. Abd-Elhaleem S, Soliman M, Hamdy M (2022) Periodic event-triggered modified repetitive control with equivalent-input-disturbance estimator based on T-S fuzzy model for nonlinear systems. Soft Comput 26:6443–6459. https://doi.org/10.1007/s00500-022-06973-5
18. Su X, Wen Y, Shi P, Wang S, Assawinchaichote W (2021) Event-triggered fuzzy control for nonlinear systems via sliding mode approach. IEEE Trans Fuzzy Syst 29(2):336–344. https://doi.org/10.1109/TFUZZ.2019.2952798
19. Feng Z, Yang Y, Wu L (2022) Event-triggered sliding-mode control for polynomial fuzzy singular systems. IEEE Trans Syst Man Cybern Syst. https://doi.org/10.1109/TSMC.2022.3222175
20. Hou ZS, Wang Z (2013) From model-based control to data-driven control: survey, classification and perspective. Inf Sci 235(20):3–35
21. Yin S, Li X, Gao H, Kaynak O (2015) Data-based techniques focused on modern industry: an. overview. IEEE Trans Ind Electron 62(1):657–667
22. Wei Q, Liu D, Lin H (2016) Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems. IEEE Trans Cybern 46(3):840–853
23. Chi R, Hou Z, Jin S, Wang D, Hao J (2013) A data-driven iterative feedback tuning approach of ALINEA for freeway traffic ramp metering with PARAMICS simulations. IEEE Trans Ind Inform 9(4):2310–2317
24. Li M, Zhu Y, Yang K, Hu C (2015) A data-driven variable-gain control strategy for an ultra-precision wafer stage with accelerated iterative parameter tuning. IEEE Trans Ind Inform 11(5):1179–1189
25. Radac MB, Precup RE, Roman RC (2018) Data-driven model reference control of MIMO vertical tank systems with model-free VRFT and Q-learning. ISA Trans 73:227–238
26. Hou ZS, Xiong SS (2019) On model-free adaptive control and its stability analysis. IEEE Trans Autom Control 64:4555–4569
27. Wang Y, Wang Z (2020) 'Model free adaptive fault-tolerant tracking control for a class of discrete-time systems.' Neurocomputing 412:143–151
28. Yang Y, Chen C, Lu J (2020) 'Parameter self-tuning of SISO compact-form model-free adaptive controller based on long short-term memory neural network. IEEE Access 8:151926–151937
29. Zhang W, Xu D, Jiang B, Pan T (2021) 'Prescribed performance based model-free adaptive sliding mode constrained control for a class of nonlinear systems.' Inf Sci 544:97–116
30. Chen C, Lu J (2019) 'Design of self-tuning SISO partial-form model-free adaptive controller for vapor-compression refrigeration system.' IEEE Access 7:125771–125782. https://doi.org/10.1109/ACCESS.2019.2939261
31. Hou ZS, Jin ST (2013) Model free adaptive control: theory and applications. CRC Press, Boca Raton
32. Hou ZS, Jin ST (2011) A novel data-driven control approach for a class of discrete-time nonlinear systems. IEEE Trans Control Syst Technol 19(6):1549–1558
33. Lin N, Chi R, Huang B (2021) Event-triggered model-free adaptive control. IEEE Trans Syst Man Cybern Syst 51(6):3358–3369. https://doi.org/10.1109/TSMC.2019.2924356
34. Liu D, Yang G (2018) Neural network-based event-triggered MFAC for nonlinear discrete-time processes. Neurocomputing 272:356–364
35. Deng Y, Ren Z, Kong Y et al (2017) A hierarchical fused Fuzzy deep neural network for data classification. IEEE Trans Fuzzy Syst 25(4):1006–1012
36. Mahmoud TA, Sheta AA, Fikry RM, Ali EH, El-Araby SM, Mahmoud MI (2022) Design of data-driven model for the pressurizer system in nuclear power plants using a TSK fuzzy neural network. Nucl Eng Des 339:112015
37. Han HG, Lin ZL, Qiao JF (2017) Modeling of nonlinear systems using the self-organizing fuzzy neural network with adaptive gradient algorithm. Neurocomputing 266(1):566–578
38. Mahmoud TA, Elshenawy LM (2022) TSK fuzzy echo state neural network: a hybrid structure for black-box nonlinear systems identification. Neural Comput Appl 34:7033–7051. https://doi.org/10.1007/s00521-021-06838-2
39. Mahmoud TA, Abdo MI, Elsheikh EA, Elshenawy LM (2021) Direct adaptive control for nonlinear systems using a TSK fuzzy

echo state network based on fractional-order learning algorithm J. Franklin Inst 358(17):9034–9060
40. Atanassov K (1986) Intuitionistic fuzzy sets. Fuzzy Sets Syst 20(1):87–96
41. Karar M, El-Garawany A, El-Brawany M (2020) Biomedical signal processing and control optimal adaptive intuitionistic fuzzy logic control of anti-cancer drug delivery systems. Biomed Signal Process Control 58:101861
42. Hamdy M, Helmy S, Magdy M (2020) Design of adaptive intuitionistic fuzzy controller for synchronization of uncertain chaotic systems. CAAI Trans Intell Technol 5(4):237–246
43. Hamdy M, Magdy M, Helmy S (2021) Control and synchronization for two Chua systems based on intuitionistic fuzzy control scheme: a comparative study. Trans Inst Meas Control 43(7):1650–1667. https://doi.org/10.1177/0142331220981425
44. Helmy S, Magdy M, Hamdy M (2022) Control in the loop for synchronization of nonlinear chaotic systems via adaptive intuitionistic neuro-fuzzy: a comparative study. Complex Intell Syst 8:3437–3450. https://doi.org/10.1007/s40747-022-00677-x
45. Yuan W, Chao L (2019) Online evolving interval type-2 intuitionistic fuzzy LSTM-neural networks for regression problems. IEEE Access 7:35544–35555
46. Zhang X-M, Han Q-L, Yu X (2016) Survey on recent advances in networked control systems. IEEE Trans Ind Inf 12(5):1740–1752. https://doi.org/10.1109/TII.2015.2506545