Neural Processing Letters (2023) 55:2885–2910
https://doi.org/10.1007/s11063-022-10989-1

Polynomial Recurrent Neural Network-Based Adaptive PID Controller With Stable Learning Algorithm

Youssef F. Hanna[1] · A. Aziz Khater[2] · Ahmad M. El-Nagar[2] · Mohammad El-Bardini[2]

Accepted: 25 July 2022 / Published online: 10 August 2022

Abstract
This paper introduces a novel structure of a polynomial weighted output recurrent neural network (PWORNN) for designing an adaptive proportional—integral—derivative (PID) controller. The proposed adaptive PID controller structure based on a polynomial weighted output recurrent neural network (APID-PWORNN) is introduced. In this structure, the number of tunable parameters for the PWORNN only depends on the number of hidden neurons and it is independent of the number of external inputs. The proposed structure of the PWORNN aims to reduce the number of tunable parameters, which reflects on the reduction of the computation time of the proposed algorithm. To guarantee the stability, the optimization, and speed up the convergence of the tunable parameters, i.e., output weights, the proposed network is trained using Lyapunov stability criterion based on an adaptive learning rate. Moreover, by applying the proposed scheme to a nonlinear mathematical system and the heat exchanger system, the robustness of the proposed APID-PWORNN controller has been investigated in this paper and proven its superiority to deal with the nonlinear dynamical systems considering the system parameters uncertainties, disturbances, set-point change, and sensor measurement uncertainty.

Keywords Lyapunov stability criterion · PID controller · Adaptive learning rate · Polynomial weighted output recurrent neural network

✉ Ahmad M. El-Nagar
ahmed.elnagar@el-eng.menofia.edu.eg

Youssef F. Hanna
Youssef-fawzy@eru.edu.eg

A. Aziz Khater
Abdelazizali2020@yahoo.com

Mohammad El-Bardini
dralbardini@el-eng.menofia.edu.eg

[1] Department of Mechatronics and Robotics, Faculty of Engineering, Egyptian Russian University, Badr City, Egypt

[2] Department of Industrial Electronics and Control Engineering, Faculty of Electronic Engineering, Menoufia University, Menof 32852, Egypt

1 Introduction

Most of the industrial processes are nonlinear dynamical systems. The control of these systems requires a robust controller, which can handle the systems uncertainties, load change, disturbances, and interference noises [1–5]. Conventional PID controller is still widely used in industry on account of its simpler structure. Furthermore, the three terms of the PID controller perform an interpreted and clear action on the system response. Unfortunately, most of the tuning methods of conventional PID parameters require known model parameters and fixed operating points [1, 3, 6]. Therefore, the conventional PID controller fails when it faces a variation in the system parameters, a sudden load change, an external disturbance, a set-point change [2, 6, 7]. For these drawbacks of the conventional PID controller, the researchers worked seriously to find suitable controllers to control such complex nonlinear dynamical systems [8, 9].

In [6], a fuzzy-back propagation (fuzzy- BP) neural network PID was introduced to control a tracking system of a wheeled mobile robot. In [10], the authors presented a single output adaptive PID controller to govern the DC side voltage of a Vienna rectifier. Three training algorithms were used in [8], for an artificial neural network-based PID controller to flight control of quadcopter using at least three input neurons, three hidden neurons and three output neurons, i.e., (3–3-3) neural network structure. The time delay temperature system was controlled using adaptive PID with Lyapunov function in [11]. The level in a tank was governed using (8–4-3) structure PID based on neural network as mentioned in [7]. A radial basis function (RBF) neural network was used to tune the PID controller parameters for DC motor position control in [12]. In [13], a PID controller with (4-5-3) BP neural network structure was applied to an experimental model. An electric-heating reactor was controlled using an RBF neural networks-based PID controller as introduced in [14]. Dynamical systems were controlled using a neural network-based PID controller with (3-20-1) structure and tangent hyperbolic activation function was used as mentioned in [15].

In addition, a PID controller based on general dynamic neural networks (GDNN) with (2-4-3) structure was introduced in [16] to control an inverted pendulum. A liquid in a surge tank was controlled using (3-30-3) neural structure-based PID controller, which is highlighted in [17]. Furthermore, [18] proposed a multiple-input-multiple-output adaptive neural-based PID controller (MIMO-AN-PID) to control a hexacopter, i.e., unmanned aerial vehicles. Recently, in 2018, a contour error identifier based on a neural network is constructed to adapt the three parameters of the PID controller (PID-NNEI) using (15-15-1) neural structure and a hyperbolic tangent activation function that used to control three axes of a computer numerical control (CNC) machine as presented in [19]. In 2020, a PID controller based on an RBF neural network (PID-RBFNN) was introduced for a speed profile control of a subway train using (3-5-3) neural network structure [20]. In 2020, a neural network-based PID controller using Levenberg–Marquardt identifier (NNPID-LM) was introduced in [21]. The NNPID-LM controller, which was optimized by using an LM learning algorithm and an adaptive learning rate, used a neural network structure (2-5-1) and a 'log-sigmoid' activation function and it was used to control nonlinear dynamical systems. Moreover, in 2021, a smart optimized PID controller based on neural networks (SOPIDNN) with (4-18-3) structure using 'tan sigmoid' activation function was introduced to control the two-wheeled differential mobile robot [22]. The neural network of SOPIDNN was trained using the BP algorithm and the weights were adjusted using gradient descent manner. The main challenge faced the previous researchers is the large number of tunable parameters that need a large computation time. On the other

hand, the BP algorithm, which converges along the mean square error (MSE) gradient descent has drawbacks such as falling in local minima and convergence rate is slow.

Early, a polynomial recurrent neural network (PRNN) based identification and control is proposed using a smaller number of tunable parameters (12 parameters) with gradient descent training algorithm and fixed learning rate [23]. PRNN is still suffering from slow convergence speed and a relatively large number of the adjusted input weights, which depends on the number of external inputs and the number of neurons in the hidden layer. Therefore, the motivation of the proposed work is to overcome the mentioned challenges and drawbacks.

In this paper, a novel structure of a polynomial weighted output recurrent neural network (PWORNN) is introduced. To guarantee the stability and speed up the convergence of the weights of the PWORNN, Lyapunov criteria-based adaptive learning rate is developed to update the weights. Furthermore, a Lyapunov criterion is used for optimizing the parameters of the controller and eliminating the problem of gradient descent besides guaranteeing the controller stability. Then, the proposed neural network structure is used to obtain the controller parameters of the PID controller. The proposed adaptive PID controller structure based on a PWORNN (APID-PWORNN) is designed for controlling nonlinear systems to reduce the effect of system uncertainties and external disturbances. The contributions of this paper can be summarized as follows:

- This paper presents an adaptive PID controller based on a novel PWORNN structure with only 6 tunable parameters.
- A stable learning algorithm is proposed in this work by deriving a new weight update rule based on the Lyapunov stability criterion to overcome the drawbacks of the gradient descent learning algorithm and prevent the proposed learning algorithm from falling in local minima.
- Deriving a new adaptation rule for the learning rate based on the Lyapunov stability criterion to guarantee the optimal convergence speed to prevent the proposed learning algorithm from slow convergence speed as in the case of using gradient descent learning algorithm.
- Two cases are studied and comparisons among the six controllers' performance show that the proposed APID-PWORNN controller has a robust performance and it is superior to other existing controllers.

The remainder of this paper is organized as follows: the structure of the polynomial recurrent neural network is described in Sect. 2. Sequentially, the proposed structure is explained in Sect. 3. Section 4 presents Lyapunov stability criteria for deriving a new weight update rule and adaptive learning rate formula following that, simulation results of a two cases studies considering, the system parameters uncertainties, disturbances, sensor measurement uncertainty, noise in the control signal, and set-point change and comparisons are introduced in Sect. 5. Finally, the conclusions are summarized in Sect. 6.

2 Polynomial Recurrent Neural Network Controller (PRNNC) Structure

This section describes the structure of PRNNC [23]. PRNNC consists of three layers; the input layer, the hidden layer and the output layer. The input layer receives the recurrent network inputs; these inputs are weighted then transmitted to the hidden layer. The hidden neurons sum the weighted inputs then send them to the output layer. The output neurons multiply all the outputs coming from the hidden neurons then send them to the output of the network as shown in Fig. 1. In the structure shown below in Fig. 1, the output of the j^{th}

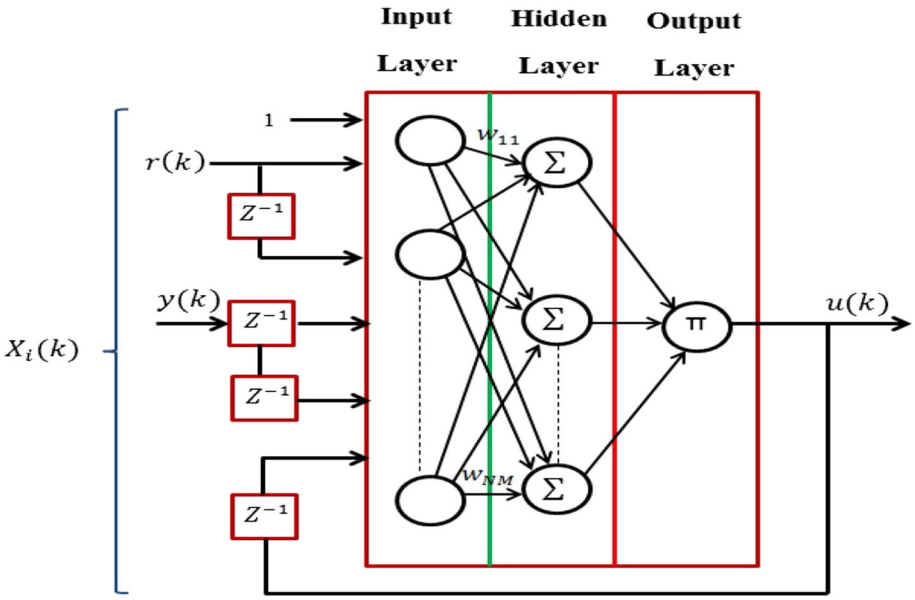

Fig. 1 Structure of PRNNC [23]

hidden neuron; $(S_j(k))$ is given as:

$$S_j(k) = \sum_{i=0}^{N} w_{ij}(k) X_i(k) \tag{1}$$

where k is the sample number, w_{ij} is the connection weight between i^{th} input neuron to j^{th} hidden neuron, N is the number of input neurons and $X_i(k) = [1, r(k), r(k-1), u(k-1), y(k-1), y(k-2)]$; is the input vector, i.e., online training data set. Basically, $S_j(k)$ changes its value based on the weights updating and the input vector. The output of the network is given as:

$$u(k) = \prod_{j=1}^{M} S_j(k) \tag{2}$$

where $u(k)$ is the control signal and M is the number of the hidden neurons. Now, $u(k)$ is expressed as the result of multiplication of aggregated terms (product of sum) operation that means polynomial function.

For the control purpose, the squared error is used as a cost function, and the gradient descent is applied to minimize the accumulative sum of the cost function as follows:

$$E(k) = \frac{1}{2}(r(k) - y(k))^2 \tag{3}$$

where $E(k)$ is the cost function, $r(k)$ is the reference input, $y(k)$ is the plant output. This method produces the following update rule:

$$w_{ij}(k) = h(r(k) - y(k)) \frac{\partial y(k)}{\partial u(k)} \frac{\partial u(k)}{\partial w_{ij}(k)} \tag{4}$$

where h is the fixed learning rate.

3 Proposed APID-PWORNN Controller Structure

The structure of the proposed APID-PWORNN controller is highlighted in Fig. 2. The proposed adaptive PID controller is constructed based on a polynomial network with weighted outputs. The input layer receives the recurrent input vector. Every hidden neuron sums the incoming inputs and then generates an output to the output layer. The output layer consists of three neurons, each output neuron products all the weighted outputs coming from the hidden neurons and then generates its own output. The three outputs coming from the output layer represent the three adaptive PID controller parameters, $K_P(k)$, $K_I(k)$, and $K_D(k)$.

The inputs and outputs of each layer are given as follows:

Input layer: The input vector to this layer is set as; $X_i(k) = [e(k), u(k-1), y(k-1)]$, i.e., online training data set, where, $y(k-1)$, $u(k-1)$, and $e(k)$ are the recurrent plant output, the recurrent control signal, and the error signal between the reference input and the plant output, respectively.

Hidden layer: The inputs of each neuron in this layer are the elements of the input vector; $X_i(k)$. While the output of the jth hidden neuron is given as:

$$F_j(k) = \sum_{i=1}^{N} X_i(k) \tag{5}$$

where N is the number of input neurons and $F_j(k)$ is the hidden neuron output.

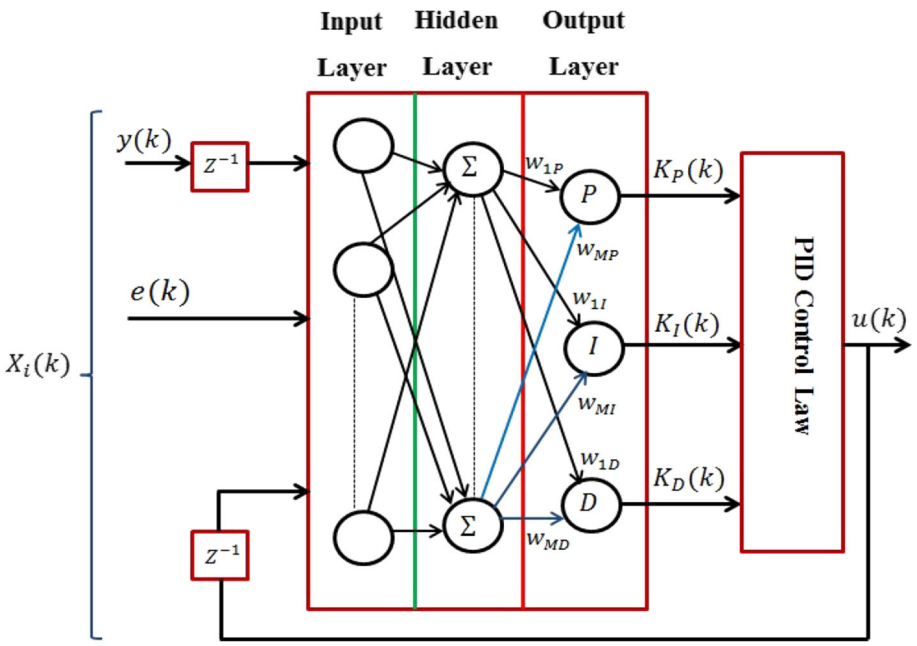

Fig. 2 Structure of the proposed APID-PWORN controller

Output layer: The inputs to the output neuron are $F_j(k)$. While the three outputs of the output layer are defined as follows:

$$K_P(k) = \prod_{j=1}^{M} w_{jP}(k) F_j(k) \tag{6}$$

$$K_I(k) = \prod_{j=1}^{M} w_{jI}(k) F_j(k) \tag{7}$$

$$K_D(k) = \prod_{j=1}^{M} w_{jD}(k) F_j(k) \tag{8}$$

where M is the number of the hidden neurons, and w_{jP}, w_{jI}, w_{jD} are the connection weights between the jth hidden neuron and the proportional output neuron (P node), integral output neuron (I node), and the derivative output neuron (D node), respectively.

Obviously, in this simple structure, the number of the tunable parameters (adjustable weights) doesn't depend on the number of inputs neurons (N) but only it depends on the number of the hidden neurons (M). Accordingly, the number of the tunable parameters always equals $3\ M$. In this work (3-2-3) structure is used, which results in 6 tunable parameters. Therefore, the proposed structure aims to reduce the number of the tunable parameters, which leads to a reduction of the computation time.

Figure 3 describes the block diagram of the closed loop control system based on the proposed APID-PWORNN controller. In this block diagram, an incremental PID controller based on discrete-time form is used as follows:

$$\Delta u(k) = K_P(k) e_1(k) + K_I(k) e_2(k) + K_D(k) e_3(k) \tag{9}$$

$$u(k) = \Delta u(k) + u(k-1) \tag{10}$$

where $K_P(k), K_I(k), K_D(k)$ are the adaptive PID controller parameters, $e_1(k) = e(k) - e(k-1), e_2(k) = e(k), e_3(k) = e(k) - 2e(k-1) + e(k-2), \Delta u(k) = u(k) - u(k-1)$,

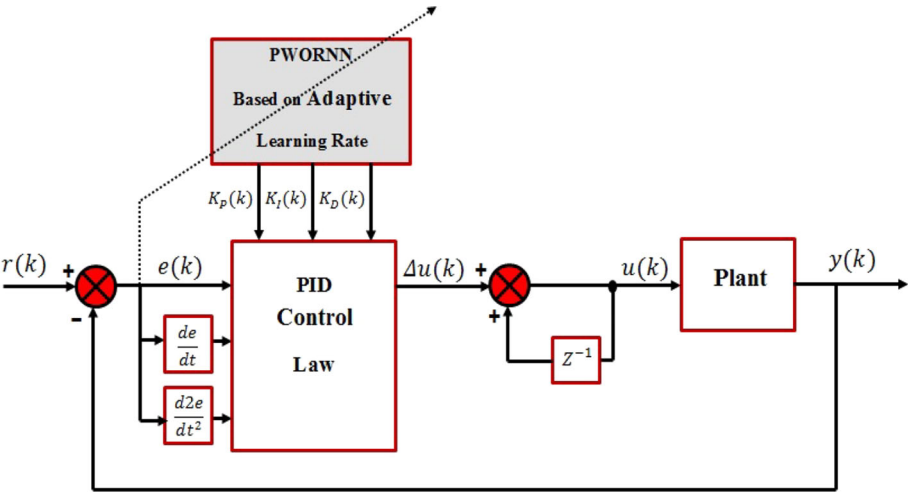

Fig. 3 Block diagram of the proposed controller

$u(k)$ is the control signal, and $e(k) = r(k) - y(k)$ is the error signal between the reference input and the plant output.

4 Lyapunov Stability Analysis

In this section, the Lyapunov stability analysis-based updating parameters and learning rate is presented to overcome the shortcomings of the gradient descent learning algorithm, which are mentioned above in the introduction section. The first part of this section explains the deriving of a new update rule for the adjusted weights of the proposed structure based on the Lyapunov stability criterion. This solution aims to prevent the proposed learning algorithm from falling in local minima. The other part explains the deriving of a new adaptation rule for the learning rate based on the Lyapunov stability criterion. This solution guarantees the optimal convergence speed and the stability for the proposed learning algorithm.

4.1 Update Rule Based on the Lyapunov Stability Criterion

For deriving a new update rule, a more flexible positive definite Lyapunov function is chosen as in [24]:

$$V_L(k) = ax_v^2(k) + 2bx_v(k)y_v(k) + cy_v^2(k) \tag{11}$$

with these constrains;$a > 0$and$ac - b^2 > 0$.

In this work, the parameters of Eq. (11) are replaced by the error signal; $e(k)$, and the connection weights vector; $w(k) = [w_{1P}, w_{2P}, w_{1I}, w_{2I}, w_{1D}, w_{2D}]^T$, where $x_v(k) = e(k)$, $y_v(k) = w(k)$. Therefore, the Lyapunov function can be rewritten as:

$$V_L(k) = ae^2(k) + 2be(k)w(k) + cw^2(k) \tag{12}$$

The Lyapunov stability criterion states that the controlled system is asymptotically stable if the following condition is satisfied as [24]:

$$\Delta V_L(k) = V_L(k+1) - V_L(k) \leq 0 \tag{13}$$

Then:

$$\begin{aligned}\Delta V_L(k) = &\left[ae^2(k+1) + 2be(k+1)w(k+1) + cw^2(k+1)\right] \\ &- \left[ae^2(k) + 2be(k)w(k) + cw^2(k)\right] \leq 0\end{aligned} \tag{14}$$

Substituting $\Delta e(k) = e(k+1) - e(k)$, and$\Delta w(k) = w(k+1) - w(k)$ in Eq. (14) and performing some simple mathematical operations those lead to:

$$\begin{aligned}\Delta V_L(k) = &c\left[\Delta w(k)\right]^2 + \Delta w(k)\left[2b\Delta e(k) + 2be(k) + 2cw(k)\right] \\ &+ a\left[\Delta e(k)\right]^2 + \Delta e(k)\left[2ae(k) + 2bw(k)\right] \leq 0\end{aligned} \tag{15}$$

Equating Eq. (15) by zero and dividing both sides by $\Delta w(k)$ that gives:

$$\begin{aligned}\Delta V_L(k) = &c\Delta w(k) + \left[2b\Delta e(k) + 2be(k) + 2cw(k)\right] \\ &+ a\frac{\left[\Delta e(k)\right]^2}{\Delta w(k)} + \frac{\Delta e(k)}{\Delta w(k)}\left[2ae(k) + 2bw(k)\right] = 0\end{aligned} \tag{16}$$

At small change, $\frac{\Delta e(k)}{\Delta w(k)}$ can be replaced by $\frac{\partial e(k)}{\partial w(k)}$ in Eq. (16); Then:

$$\Delta w(k) = -\frac{1}{c}\left[(2b\Delta e(k) + 2be(k) + 2cw(k)) + \frac{\partial e(k)}{\partial w(k)}(2a\Delta e(k) + 2ae(k) + 2bw(k))\right] \tag{17}$$

For simplifying the incremental term form in Eq. (17), let:

$$L_1 = (2b\Delta e(k) + 2be(k) + 2cw(k)) \text{ and } L_2 = (2a\Delta e(k) + 2ae(k) + 2bw(k))$$

Then Eq. (17) can be rewritten as follows:

$$\Delta w(k) = -\frac{1}{c}\left[L_1 + \frac{\partial e(k)}{\partial w(k)} L_2\right] \tag{18}$$

Now Eq. (18) guarantees the convergence stability. Moreover, to satisfy the optimization, the main formula of weight update, which minimizes the cost function that defined in Eq. (12), is given as:

$$w(k+1) = w(k) + h(k)\Delta w(k) \tag{19}$$

where $h(k)$ is the adaptive learning rate, and $\Delta w(k)$ is the incremental term, which is given in Eq. (18), and the last term $(h(k)\Delta w(k))$ in Eq. (19) can be called the updating term. Using the chain rule, the term $\frac{\partial e(k)}{\partial w(k)}$ in Eq. (18) can be replaced by $\frac{\partial e(k)}{\partial w(k)} = \frac{-\partial y(k)}{\partial w(k)} = \frac{-\partial y(k)}{\partial u(k)} \frac{\partial u(k)}{\partial w(k)}$. Then, the six adjusted weights (connecting weights from the hidden layer to the output layer) of the proposed APID-PWORNN controller can be updated using Eq. (19) as follows:

$$w_{1P}(k+1) = w_{1P}(k) - h(k)\frac{1}{c}\left[L_{11P} - \frac{\partial y(k)}{\partial u(k)}\frac{\partial u(k)}{\partial w_{1P}(k)} L_{21P}\right] \tag{20}$$

$$w_{2P}(k+1) = w_{2P}(k) - h(k)\frac{1}{c}\left[L_{12P} - \frac{\partial y(k)}{\partial u(k)}\frac{\partial u(k)}{\partial w_{2P}(k)} L_{22P}\right] \tag{21}$$

$$w_{1I}(k+1) = w_{1I}(k) - h(k)\frac{1}{c}\left[L_{11I} - \frac{\partial y(k)}{\partial u(k)}\frac{\partial u(k)}{\partial w_{1I}(k)} L_{21I}\right] \tag{22}$$

$$w_{2I}(k+1) = w_{2I}(k) - h(k)\frac{1}{c}\left[L_{12I} - \frac{\partial y(k)}{\partial u(k)}\frac{\partial u(k)}{\partial w_{2I}(k)} L_{22I}\right] \tag{23}$$

$$w_{1D}(k+1) = w_{1D}(k) - h(k)\frac{1}{c}\left[L_{11D} - \frac{\partial y(k)}{\partial u(k)}\frac{\partial u(k)}{\partial w_{1D}(k)} L_{21D}\right] \tag{24}$$

$$w_{2D}(k+1) = w_{2D}(k) - h(k)\frac{1}{c}\left[L_{12D} - \frac{\partial y(k)}{\partial u(k)}\frac{\partial u(k)}{\partial w_{2D}(k)} L_{22D}\right] \tag{25}$$

where $L_{11P}, L_{21P}, L_{12P}, L_{22P}, L_{11I}, L_{21I}, L_{12I}, L_{22I}, L_{11D}, L_{21D}, L_{12D}$ and L_{22D} are given in Appendix.

The value of the partial derivative; $\frac{\partial y(k)}{\partial u(k)}$ has no major effect on the learning algorithm in Eqs. (20–25) because it can be absorbed by the learning rate; $h(k)$[23]. Therefore, it is considered a constant value in this work, while $\frac{\partial u(k)}{\partial w(k)}$ can be calculated as the following remark:

Remark: For deriving the formulas of $\frac{\partial u(k)}{\partial w(k)}$ in Eqs. (20–25), since the two hidden neurons used in this structure generate the same output value from Eq. (5), and then, let, $F_j(k) = F$, accordingly Eqs. (6–8) can be rewritten in the following form:

$$K_P(k) = F^2 w_{1P}(k) w_{2P}(k) \tag{26}$$

$$K_I(k) = F^2 w_{1I}(k) w_{2I}(k) \tag{27}$$

$$K_D(k) = F^2 w_{1D}(k) w_{2D}(k) \tag{28}$$

Substituting Eqs. (26–28) in Eq. (9) and Eq. (10) and performing the partial differentiation on Eq. (10), the required formulas of $\frac{\partial u(k)}{\partial w(k)}$ for the six adjusted weights can easily derived as follows:

$$\frac{\partial u(k)}{\partial w_{1P}(k)} = \frac{\partial u(k)}{\partial K_P(k)} \frac{\partial K_P(k)}{\partial w_{1P}(k)} = e_1(k) F^2 w_{2P}(k) + \frac{\partial u(k-1)}{\partial w_{1P}(k-1)} \tag{29}$$

$$\frac{\partial u(k)}{\partial w_{2P}(k)} = \frac{\partial u(k)}{\partial K_P(k)} \frac{\partial K_P(k)}{\partial w_{2P}(k)} = e_1(k) F^2 w_{1P}(k) + \frac{\partial u(k-1)}{\partial w_{2P}(k-1)} \tag{30}$$

$$\frac{\partial u(k)}{\partial w_{1I}(k)} = \frac{\partial u(k)}{\partial K_I(k)} \frac{\partial K_I(k)}{\partial w_{1I}(k)} = e_2(k) F^2 w_{2I}(k) + \frac{\partial u(k-1)}{\partial w_{1I}(k-1)} \tag{31}$$

$$\frac{\partial u(k)}{\partial w_{2I}(k)} = \frac{\partial u(k)}{\partial K_I(k)} \frac{\partial K_I(k)}{\partial w_{2I}(k)} = e_2(k) F^2 w_{1I}(k) + \frac{\partial u(k-1)}{\partial w_{2I}(k-1)} \tag{32}$$

$$\frac{\partial u(k)}{\partial w_{1D}(k)} = \frac{\partial u(k)}{\partial K_D(k)} \frac{\partial K_D(k)}{\partial w_{1D}(k)} = e_3(k) F^2 w_{2D}(k) + \frac{\partial u(k-1)}{\partial w_{1D}(k-1)} \tag{33}$$

$$\frac{\partial u(k)}{\partial w_{2D}(k)} = \frac{\partial u(k)}{\partial K_D(k)} \frac{\partial K_D(k)}{\partial w_{2D}(k)} = e_3(k) F^2 w_{1D}(k) + \frac{\partial u(k-1)}{\partial w_{2D}(k-1)} \tag{34}$$

Sequentially, the proposed APID-PWORNN controller parameters given by Eqs. (26–28) can be updated directly by inserting the updated weights Eqs. (20–25) into Eqs. (26–28).

4.2 Adaptation of the Learning Rate Based on the Lyapunov Stability Criterion

To guarantee the optimization of the convergence speed and the convergence stability that may be lost when using the gradient descent learning algorithm. An adaption rule for the learning rate based on the Lyapunov function is derived for the proposed learning algorithm in this subsection. Following the same manner used in [25–28] an adaptation rule can be obtained as follows:

Let the Lyapunov function is as follows:

$$L_v(k) = \frac{1}{2} e^2(k) \tag{35}$$

where $L_v(k)$ is a Lyapunov function, $e(k)$ is the error signal. To guarantee the stability, the following condition should be achieved:

$$\Delta L_v(k) = L_v(k+1) - L_v(k) \leq 0 \tag{36}$$

Then,

$$\Delta L_v(k) = \frac{1}{2} \left[e^2(k+1) - e^2(k) \right] \tag{37}$$

Equation (37) can be rewritten as:

$$\Delta L_v(k) = \frac{1}{2} [e(k+1) + e(k)][e(k+1) - e(k)] \tag{38}$$

Since $\Delta e(k) = e(k+1) - e(k)$, then Eq. (38) can be rewritten as:

$$\Delta L_v(k) = \Delta e(k) \left[\frac{1}{2} (\Delta e(k)) + e(k) \right] \leq 0 \tag{39}$$

Now Tayler series expansion is used to express $e(k+1)$ as:

$$e(k+1) = e(k) + \frac{\partial e(k)}{\partial Q(k)} \Delta Q(k) + h_{ot} \tag{40}$$

where $Q(k)$ is any tuned parameter in PWORNN, which can be considered the output weight vector; $w(k)$, and h_{ot} is the higher order terms that can be neglected. So, $\Delta e(k)$ can be written as:

$$\Delta e(k) = \frac{\partial e(k)}{\partial Q(k)} \Delta Q(k) = \frac{\partial e(k)}{\partial w(k)} \Delta w(k) = \frac{\partial (r(k) - y(k))}{\partial w(k)} \Delta w(k) = \frac{-\partial y(k)}{\partial w(k)} \Delta w(k) \tag{41}$$

Now, for guaranteeing the weight updating stability, $\Delta w(k)$ in Eq. (41) can be considered the updating term $(h(k)\Delta w(k))$ from the update rule; Eq. (19). Then, replacing $\Delta e(k)$ in Eq. (39) by $\Delta e(k) = \frac{-\partial y(k)}{\partial w(k)} \Delta w(k) h(k)$, which yields that:

$$\Delta L_v(k) = \frac{1}{2} h(k) \Delta w(k) \frac{\partial y(k)}{\partial w(k)} \left[h(k) \Delta w(k) \frac{\partial y(k)}{\partial w(k)} - 2e(k) \right] \leq 0 \tag{42}$$

Then, replacing $\Delta e(k)$ in Eq. (17) by $\Delta e(k) = \frac{-\partial y(k)}{\partial w(k)} \Delta w(k) h(k)$, which gives $\Delta w(k)$ as follows:

$$\Delta w(k) = -\frac{1}{c} \frac{\left[\frac{\partial y(k)}{\partial w(k)} [-2bw(k) - 2ae(k)] + 2be(k) + 2cw(k) \right]}{\left[1 - \frac{2h(k)}{c} \frac{\partial y(k)}{\partial w(k)} \left[b - a \frac{\partial y(k)}{\partial w(k)} \right] \right]} \tag{43}$$

Finally, substituting Eq. (43) in Eq. (42) that leads to:

$$0 \leq h(k) \leq \frac{ce(k)}{\left[\left[\frac{\partial y(k)}{\partial w(k)} \right]^2 (bw(k) - ae(k)) + \frac{\partial y(k)}{\partial w(k)} (be(k) - cw(k) \right]} \tag{44}$$

Taking the Euclidean norm, the adaptive learning rate, which guarantees the learning stability is given as:

$$0 \leq h(k) \leq \frac{ce(k)}{\left[\left[\frac{\partial y(k)}{\partial w(k)} \right]^2 (bw(k) - ae(k)) + \frac{\partial y(k)}{\partial w(k)} (be(k) - cw(k) \right]} \tag{45}$$

The adaptation of the learning rate; $h(k)$ can be performed by using the Euclidean norm of the previous equation, which mainly depends on the weight updating and the error signal $(e(k))$.

5 Simulation Results

This section presents the MATLAB simulation results and the comparisons among the performance of the proposed APID-PWORNN controller, and four previous published neural network PID controllers that are pre-described in the introduction section such as the PID-NNEI controller [19], the PID-RBFNN controller [20], the NNPID-LM controller [21], which is optimized by the LM learning algorithm and an adaptive learning rate, and the SOPIDNN controller [22]. In addition, an improved particle swarm-based PID (IPSO-PID) controller [29], is added to the comparisons. All these algorithms are programmed using MATLAB

R2017b scripts. The neural structure of the PID-NNEI controller [19] is (5-5-1) with biases in the hidden and output neurons for the error identifier, in addition three adjusted parameters of the PID controller that totally yields 39 tunable parameters. The used NN structure of the PID-RBFNN controller [20] is (3-5-3) with two adjusted parameters for each RBF hidden neuron (i.e., center and radius), which yields 25 tunable parameters. Moreover, an (2-5-1) NN structure with neurons biases for plant identifier is used for the NNPID-LM controller [21]; in addition to the three adjusted parameters of the PID controller, which totally yields 24 tunable parameters. In addition, a (2-18-3) neural structure that used to control single–input–single–output systems (SISO systems), which yields 90 tunable parameters is used for SOPIDNN controller [22].

Furthermore, to judge the superiority of the performance of the proposed controller, two indices are considered; the integral absolute error (IAE) and the mean absolute error (MAE). Several simulation tasks are preceded on the six controllers to investigate the robustness of the proposed controller performance such as unit step response, set-point change, system parameters uncertainty, system disturbance, and actuator noise. All these simulation tasks are applied through two cases studies that are explained in details in the following subsections. For fairness, the learning rate is unified for all five NN controllers to be $h = 0.001$. Furthermore, all initial weights values are unified and chosen as random numbers in $[-0.5, 0.5]$. For the proposed APID-PWORNN controller; the coefficients of the Lyapunov function in Eq. (12) are chosen as; $a = 1$, $b = 0.2$, $c = 2$; and the initial value of the learning rate is $h(k) = 0.001$. In addition, the total number, which is used for the particles for the IPSO-PID controller, is 25 particles. The optimized IPSO-PID controller parameters are $K_P = 39.45 \times 10^{-3}$, $K_I = 58.4 \times 10^{-3}$, $K_D = 154.3 \times 10^{-3}$ for the case study 1, and $K_P = -9.875 \times 10^{-3}$, $K_I = -1.575 \times 10^{-3}$, $K_D = -2.225 \times 10^{-3}$ for the case study 2. Moreover, the two indices, which are measured for all controllers, are given as:

$$IAE = T \sum_{k=1}^{n} |e(k)| \tag{46}$$

$$MAE = \frac{1}{n} \sum_{k=1}^{n} |e(k)| \tag{47}$$

where T is the sampling time and n is the total number of the iteration.

5.1 Case Study 1

Consider the mathematical nonlinear dynamical system described as [23]:

$$y_p(k+1) = \frac{P_1 y_p(k)}{1 + y_p^2(k) + y_p^2(k-1)} + \frac{P_2}{1 + e^{-P_3[y_p(k)+y_p(k-1)]}} + P_4 u(k) + P_5 u(k-1) + d_p \tag{48}$$

where $y_p(k)$ is the system output, $u(k)$ is the control input (control signal), and the system parameters set as $P_1 = 1$, $P_2 = 0.1$, $P_3 = 1$, $P_4 = 1$, $P_5 = 0.4$, $d_p = 0$.

5.1.1 Task 1: Unit Step Response

The control scheme is built as in Fig. 3. A unit step input is applied to the closed-loop system to depict the system response for the six controllers. Figure 4 shows the system response for the proposed APID-PWORNN controller and other controllers. It's clear that the proposed

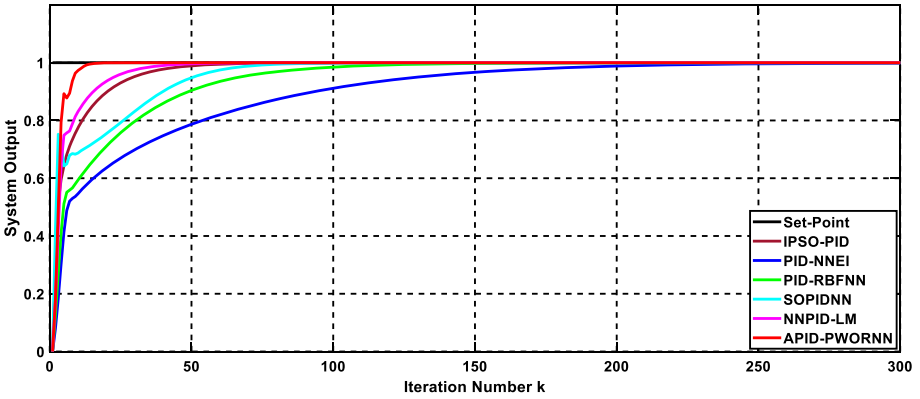

Fig. 4 System output (Task 1)

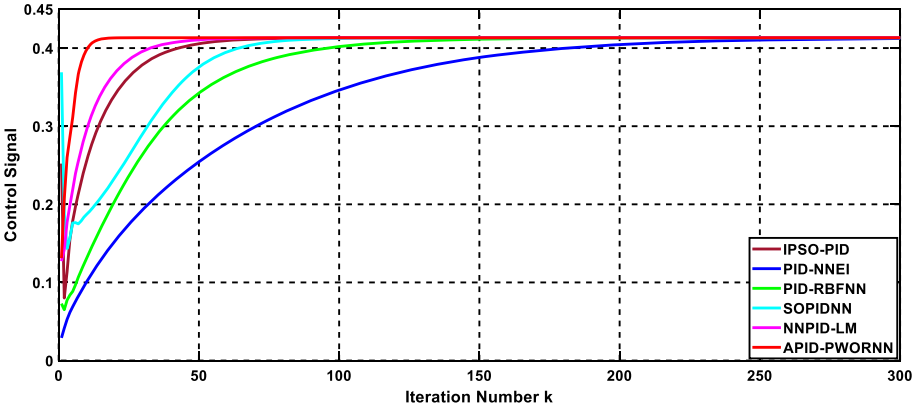

Fig. 5 Control Signal (Task 1)

controller, which is indicated with the red curve, reaches to the set-point faster than the other controllers. Furthermore, Fig. 5 shows the control signal of all controllers.

5.1.2 Task 2: Set-Point Change

Figure 6 presents the behavior of the six controllers when the set-point is changed. Obviously, the proposed APID-PWORNN controller has more convergence speed and accuracy than the other controllers. NNPID-LM controller causes some overshoot in the beginning and it is relative slow convergence. SOPIDNN and PID-RBFNN controllers are slowed down at the last stage of the learning (from $k = 2000$ to $k = 2500$). PID-NNEI controller is slowed down at stage (from $k = 1500$ to $k = 2000$). Figure 7 shows the control signal for all controllers. Moreover, Fig. 8 shows the adaptation of the APID-PWORNN controller parameters.

5.1.3 Task 3: System Parameters Uncertainty

In this task, all system parameters are decreased to 80% as parameters uncertainty. Figure 9

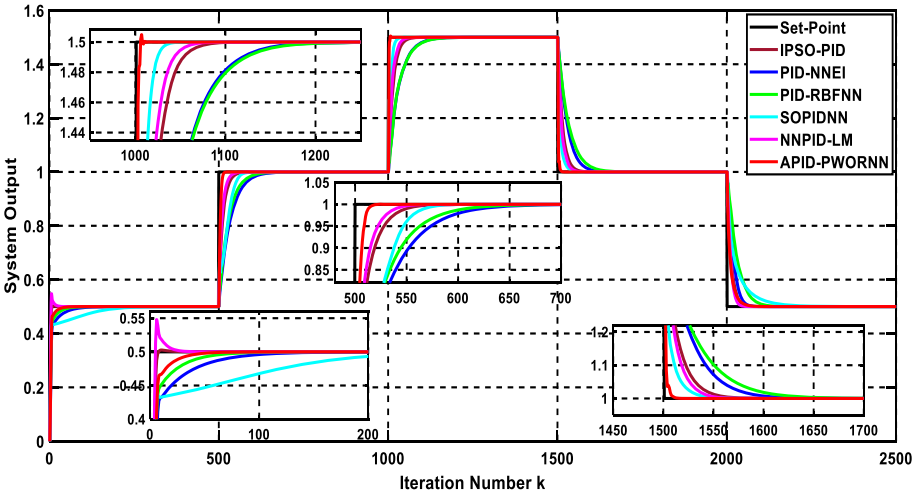

Fig. 6 System output (Task 2)

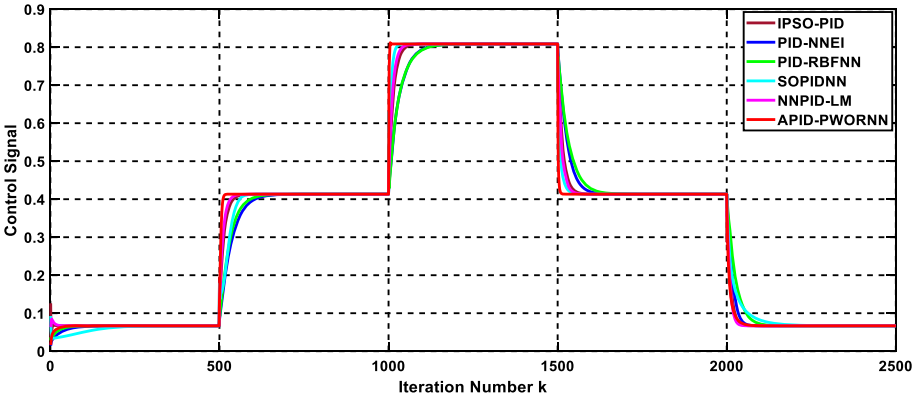

Fig. 7 Control Signal (Task 2)

depicts the unit step response with the effect of this uncertainty. The proposed APID-PWORNN controller is still the superior controller. It is the least affected and faster retraced the reference input. Furthermore, the control signal is presented in Fig. 10 for all controllers.

5.1.4 Task 4: Disturbance Model

A disturbance signal is added to Eq. (48) as $d_p(k) = 0.7$ (70% of the reference input) at iteration number $k = 500$. Figure 11 shows how each controller handles the system with this disturbance model. Here, the six controllers proved the robust performance but still the proposed APID-PWORNN controller is the fastest controller for handling this disturbance and the least affected one. Figure 12 shows the control signal for all controllers. Figure 13 indicates the changing of the APID-PWORNN controller parameters along the time in particular during applying the disturbance.

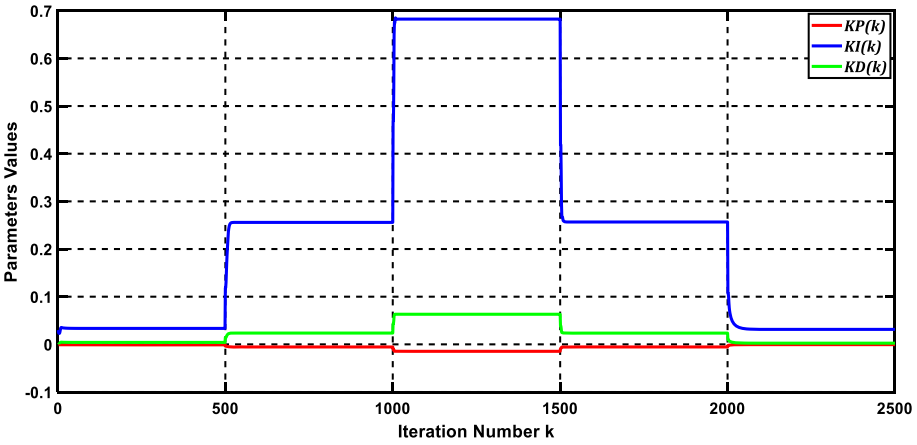

Fig. 8 APID-PWORNN controller parameters (Task 2)

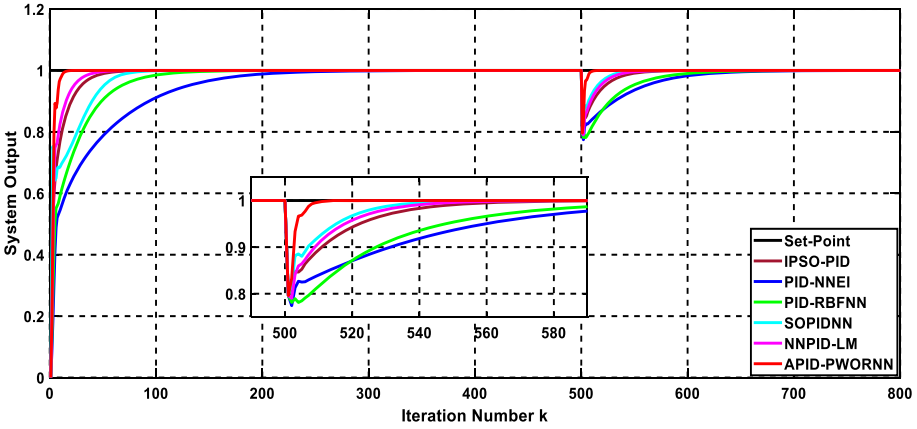

Fig. 9 System output (Task 3)

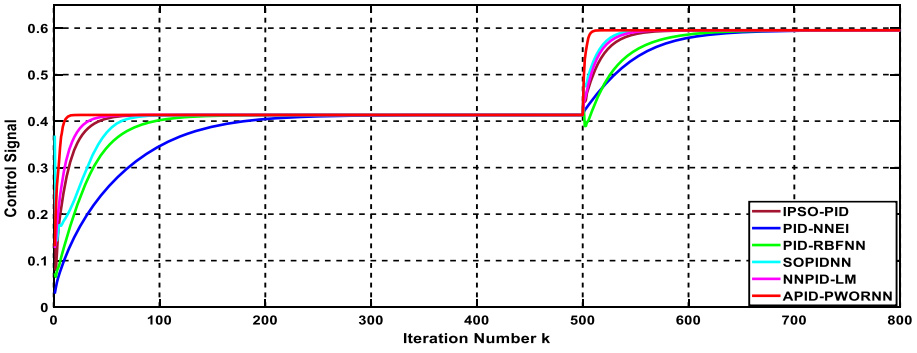

Fig. 10 Control Signal (Task 3)

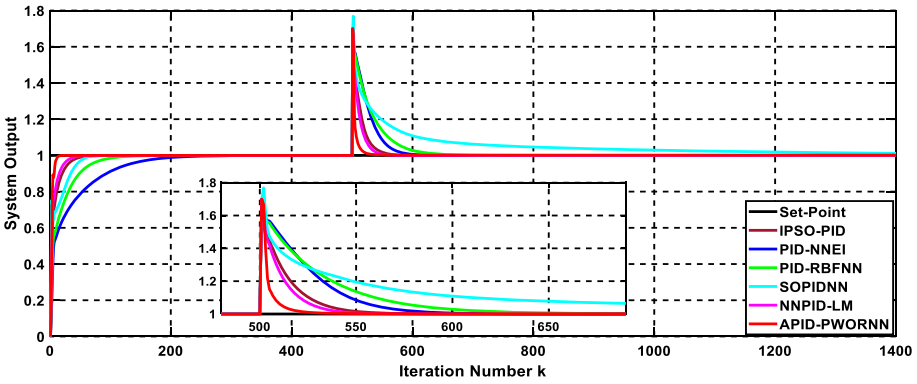

Fig. 11 System output (Task 4)

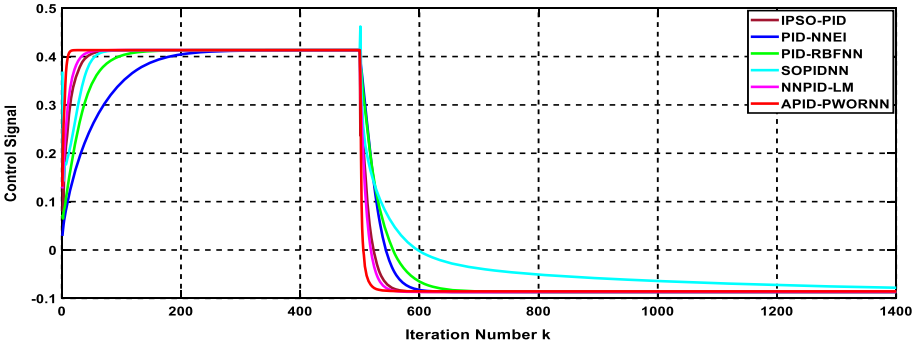

Fig. 12 Control Signal (Task 4)

The IAE and MAE are listed in Tables 1 and 2, respectively for the five neural controllers and the IPSO-PID controller that are used in this work. Moreover, to investigate the powerful computation of the proposed neural structure (3-2-3), which yields only 6 tunable parameters, the neural structure of the SOPIDNN and PID-RBFNN controllers are unified to be (3-2-3) i.e., SOPIDNN (3-2-3), which results in 12 tunable parameters and PID-RBFNN (3-2-3), which results in 10 tunable parameters. In addition, the conventional PID controller is performed for purpose of comparison.

Tables 1 and 2 show that the proposed APID-PWORNN controller has the least values of IAE and MAE in all tasks. Furthermore, the proposed controller with an adaptive learning rate (APID-PWORNN-AL) recorded fewer values of the proposed controller with a fixed learning rate (APID-PWORNN-FL). It's clear that the proposed controller has a simple structure with fewer parameters and a good robustness compared with other controllers. Moreover, the robustness performance is proved through the above tests.

5.2 Case Study 2

Consider the heat exchanger system described in [30]. This system aims to raise the temperature of the process water; $y(k)$ by steam flow rate. The system has two inputs and one output

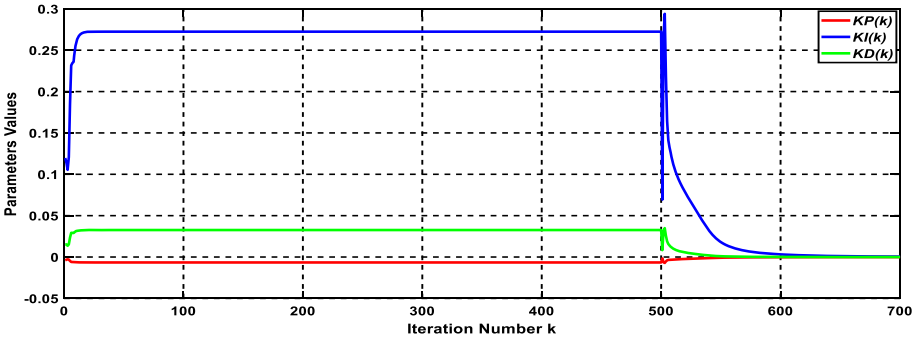

Fig.13 APID-PWORNN controller parameters (Task 4)

Table 1 Values of IAE

Algorithms	NN structure	Task 1	Task 2	Task 3	Task 4
APID-PWORNN-AL	3-2-3	0.0310	0.1229	0.0369	0.0753
APID-PWORNN-FL	3-2-3	0.0316	0.1726	0.0497	0.2033
SOPIDNN [22]	2-18-3	0.1155	0.3943	0.1358	0.7936
SOPIDNN [22]	3-2-3	0.1991	3.0355	0.2361	2.8321
NNPID-LM [21]	2-5-1	0.0573	0.2244	0.0826	0.1278
PID-RBFNN [20]	3-5-3	0.1697	0.6177	0.2437	0.3743
PID-RBFNN [20]	3-2-3	0.5589	2.0507	0.7999	3.0927
PID-NNEI [19]	5-5-1	0.7725	1.3491	0.8572	0.8456
IPSO-PID [29]	–	0.0708	0.2661	0.1019	0.1564
Conventional-PID	–	1.6672	6.3614	2.6052	3.5946

Table 2 Values of MAE

Algorithms	NN structure	Task 1	Task 2	Task 3	Task 4
APID-PWORNN-AL	3-2-3	0.0062	0.0049	0.0041	0.005
APID-PWORNN-FL	3-2-3	0.0063	0.0069	0.0052	0.0102
SOPIDNN [22]	2-18-3	0.0231	0.0158	0.0151	0.0529
SOPIDNN [22]	3-2-3	0.0398	0.1214	0.0262	0.1888
NNPID-LM [21]	2-5-1	0.0115	0.0090	0.0092	0.0085
PID-RBFNN [20]	3-5-3	0.0339	0.0247	0.0271	0.0250
PID-RBFNN [20]	3-2-3	0.1118	0.0820	0.0889	0.2062
PID-NNEI [19]	5-5-1	0.1545	0.0540	0.0952	0.0564
IPSO-PID [29]	–	0.0142	0.0106	0.0113	0.0104
Conventional-PID	–	0.3334	0.2545	0.2895	0.2396

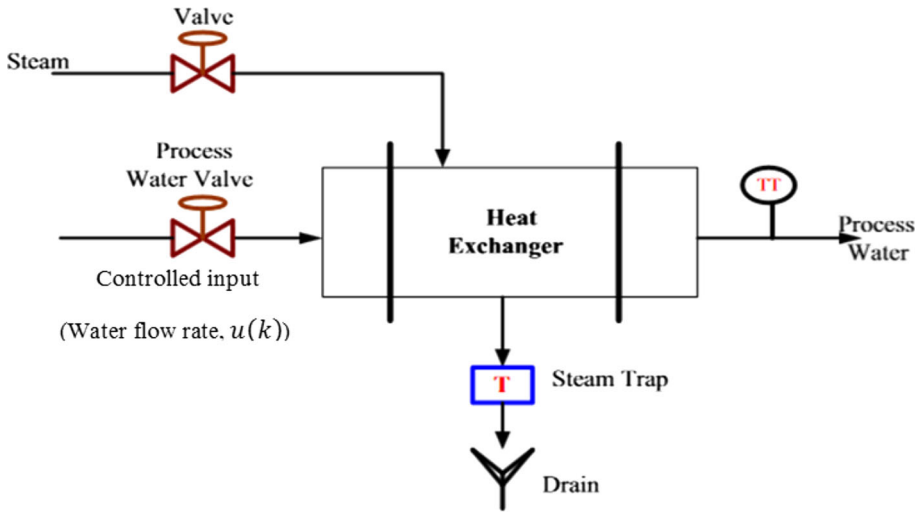

Fig. 14 Heat Exchanger System [30]

and classified from the control view point as a temperature control system. The first input is the steam flow rate, which is a fixed rate and another is the water flow rate; $u(k)$, which is controlled by the control signal of the APID-PWORNN controller and other controllers in the simulations. The two inputs can be controlled using pneumatic control valves as shown in Fig. 14. Temperature is the output of this system, which has a nonlinear behavior. The dynamics of the steam-heat exchanger are given as:

$$y(k) = q_1 y(k-1) + q_2 y(k-2) + q_3 z(k-1) + q_4 z(k-2) + d_q(k) \tag{49}$$

$$z(k) = u(k) + q_5 u^2(k) + q_6 u^3(k) + q_7 u^4(k) \tag{50}$$

where $y(k)$ is the plant output (process temperature),$u(k)$ is the control input (input water flow rate), and the system parameters are set as; $q_1 = 1.608$, $q_2 = -0.6385$, $q_3 = -6.5306$, $q_4 = 5.5652$, $q_5 = -1.3228$, $q_6 = 0.767$, $q_7 = -2.1755$, $d_q(k) = 0$. These parameters values are derived from real data for a practical system as explained in [31–33]. So, the model given by Eqs. (49) and (50) represents a real system.

5.2.1 Task 1: STEP Response

Figure 15 shows the heat exchanger response of a unit step input signal. The proposed APID-PWORNN controller based on adaptive learning rate has more convergence speed than other controllers. NNPID-LM and PID-NNEI controllers caused some overshoots in the transient period. The PID-RBFNN controller showed the least convergence speed in this task. Figure 16 presents the control signals of all controllers.

5.2.2 Task 2: Set-Point Change

Figure 17 depicts the response of the heat exchanger system when the reference input is changed. From the figure, the superiority (more accurate, and more convergence speed) of

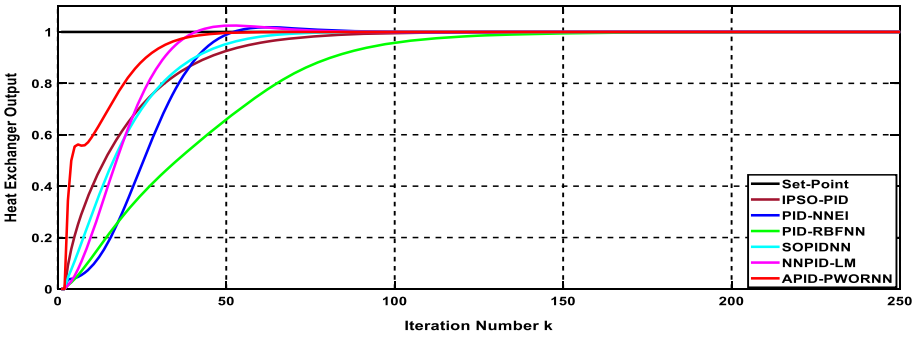

Fig. 15 Heat exchanger output (Task 1)

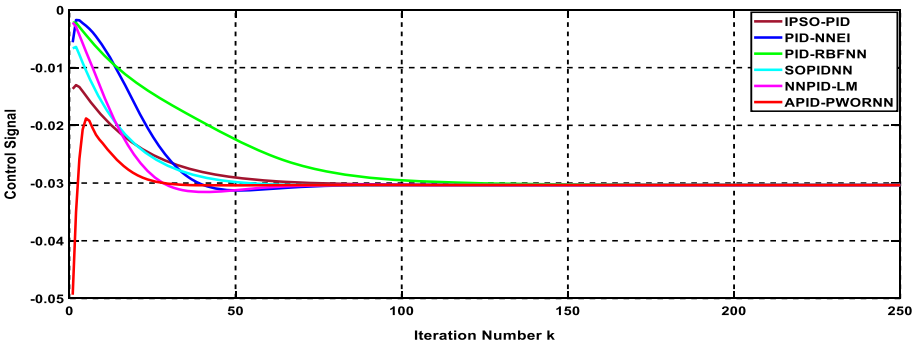

Fig. 16 Control signal for heat exchanger (Task 1)

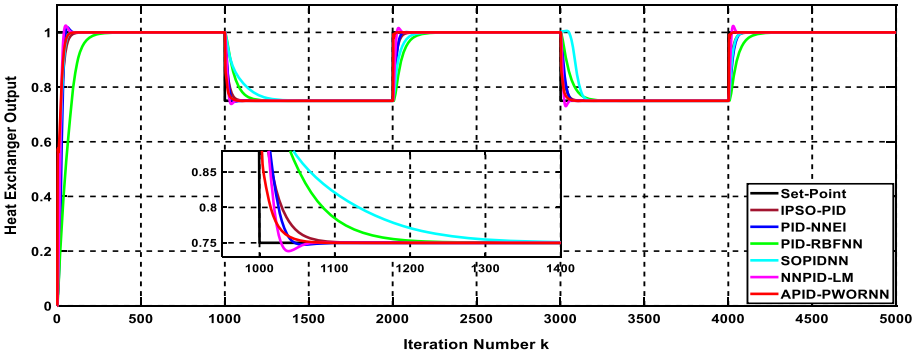

Fig. 17 Heat exchanger output (Task 2)

the proposed APID-PWORNN controller compared with other controllers is shown. The control signals for all controllers are shown in Fig. 18. The adapting of the APID-PWORN controller parameters is depicted in Fig. 19.

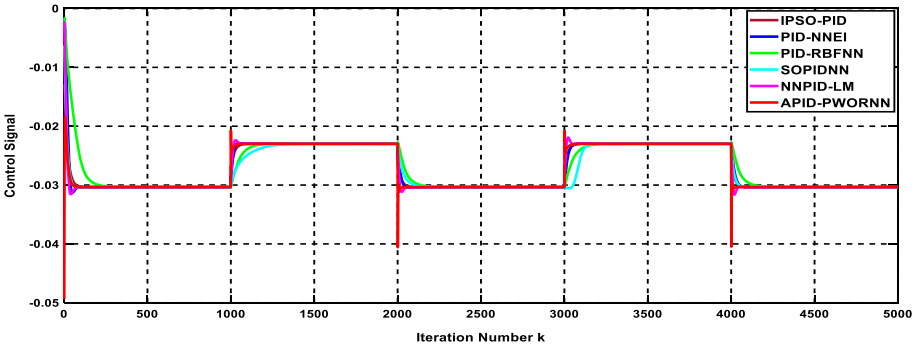

Fig. 18 Control signal for heat exchanger (Task 2)

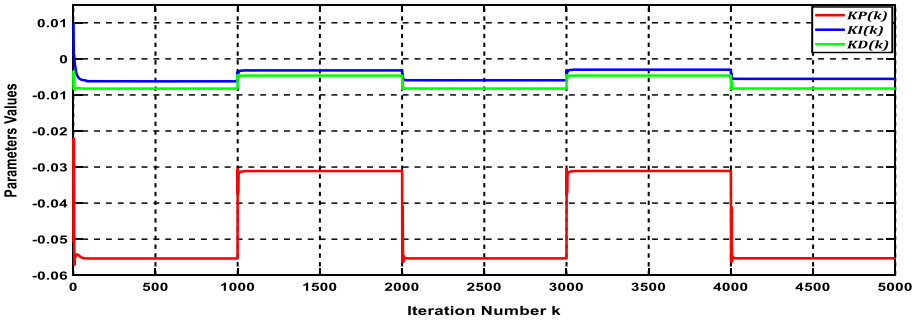

Fig. 19 Changing the APID-PWORNN controller parameters (Task 2)

5.2.3 Task 3: Heat Exchanger System Uncertainty

All the parameters of the heat exchanger have acquired an uncertainty by adding them with 80% at the 500th instant. The adaptation of the learning rate of the APID-PWORNN controller increased the convergence speed and accordingly minimized the effect of the uncertainty in the response that is shown in Fig. 20. Moreover, the control signal is depicted in Fig. 21 for all controllers.

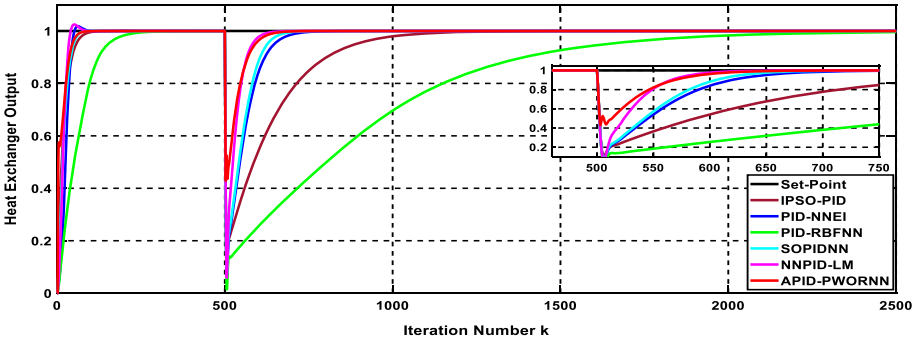

Fig. 20 Heat exchanger output (Task 3)

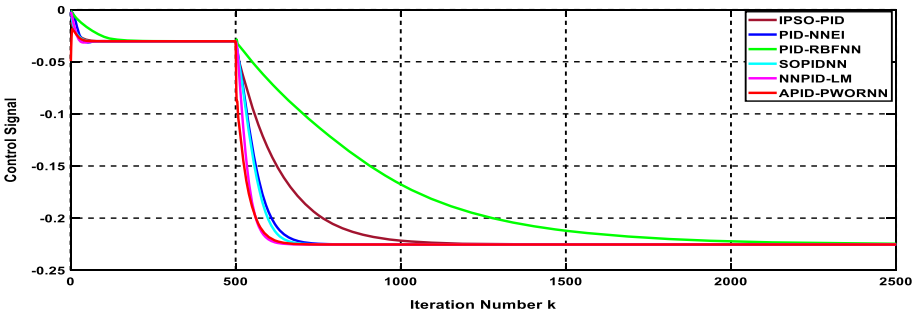

Fig. 21 Control signal for heat exchanger (Task 3)

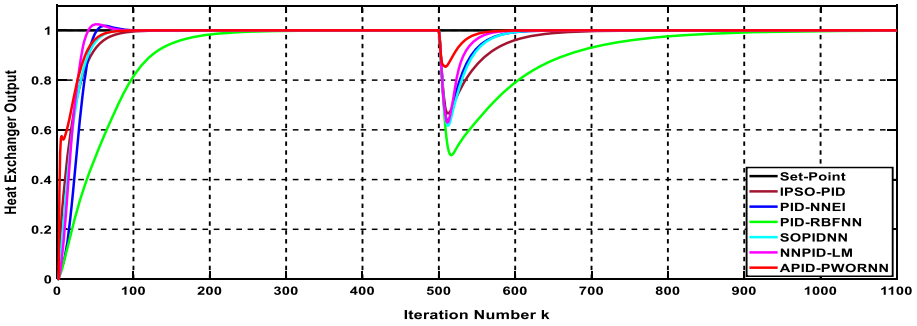

Fig. 22 Heat exchanger output (Task 4)

5.2.4 Task 4: Heat Exchanger With an External Disturbance Model

This disturbance in the output measurements can be caused by the sensor uncertainty. This task is performed by substituting $d_q(k) = -0.284\sin(0.1\mathrm{y(k)})$ in Eq. (49). Figure 22 shows the simulated heat exchanger response for all controllers due to this disturbance model. The proposed controller is the least affected controller. Figure 23 shows the control signal for all controllers. Furthermore, the self-changing of the APID-PWORNN controller parameters to eliminate the effect of the external disturbance is shown in Fig. 24.

5.2.5 Task 5: Actuator Noise

Actuator noise can be expressed by adding a noise signal to the plant output at the 500th instant. So, in this work, a noise signal with *-0.05 rand (1)* is added to the heat exchanger output in Eq. (49). Figure 25 shows the superiority of the proposed APID-PWORNN controller compared with other controllers. Furthermore, the control signal is presented in Fig. 26 for all controllers.

For any nonlinear system, the stability and the limit cycle can be defined as follows:

Stability: the dynamical system is stable if all state variables of it are converged to the equilibrium point after the internal or external perturbation is applied to the dynamical system. And the system is called unstable if at least one of its state variables has diverged in an oscillatory or exponential manner [34].

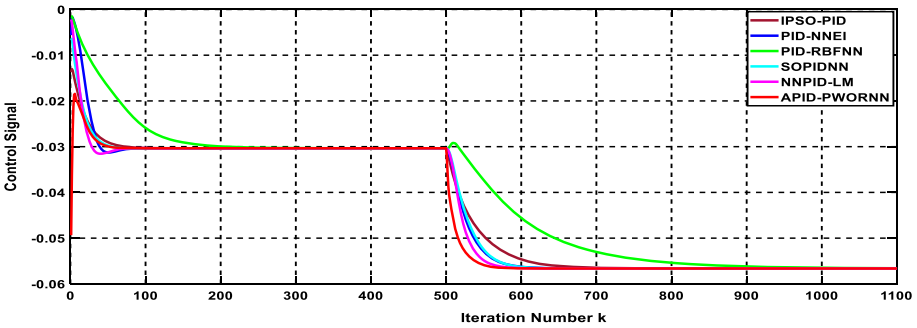

Fig. 23 Control signal for heat exchanger (Task 4)

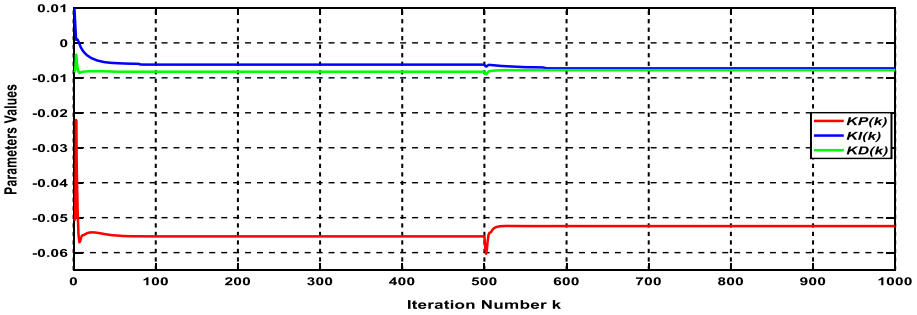

Fig. 24 Adapting the APID-PWORNN controller parameters (Task 4)

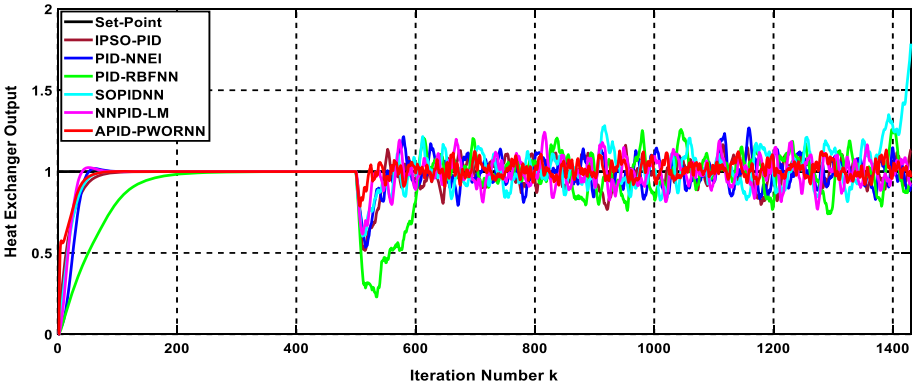

Fig. 25 Heat exchanger output (Task 5)

A limit cycle: is a closed trajectory in phase space having a property that at least one other trajectory spirals into it [34].

Based on the above definitions, the stability and limit cycles (i.e., phase portrait) for the open-loop system and closed-loop system of the heat exchanger system are highlighted in Fig. 27 (a, b), respectively.

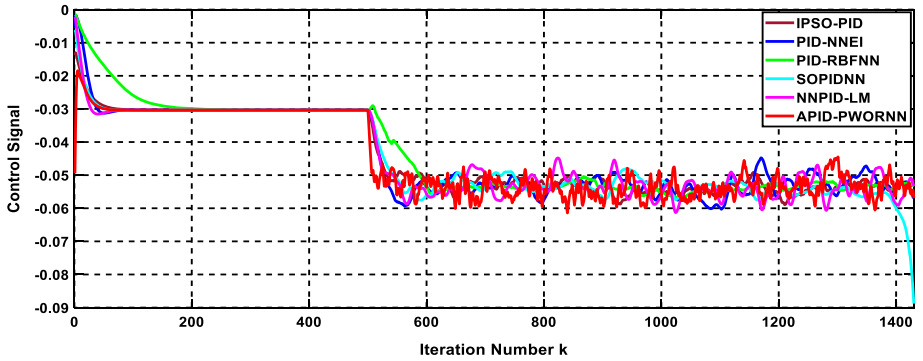

Fig. 26 Control signal for heat exchanger (Task 5)

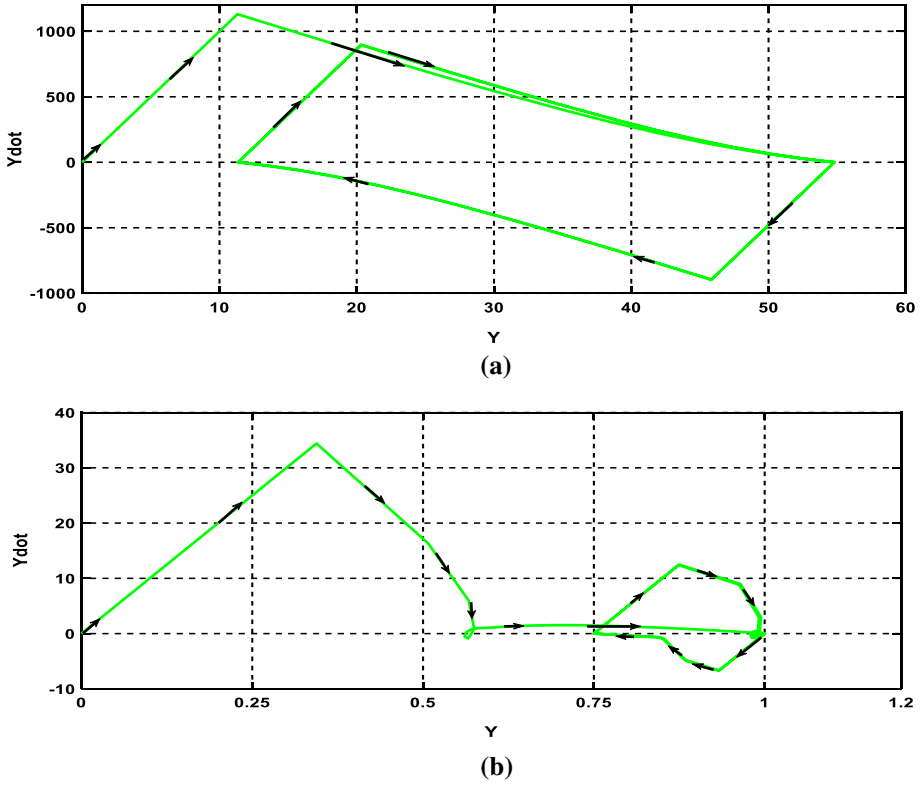

Fig. 27 Phase portrait of heat exchanger system **a** Open loop. **b** Closed loop

Figure 27a shows the phase portrait between the heat exchanger output; $(Y = y(k))$ on the horizontal axis, and the derivative of the output Ydot $= \frac{y(k)-y(k-1)}{T}$ on the vertical axis at input changes such as $u(k) = 1, 0.75, 1, 0.75,$ and 1 without controller (i.e., open-loop), while Fig. 27b shows the phase portrait at a set-point change such as $r(k) = 1, 0.75, 1, 0.75,$ and1 with the proposed controller (i.e., closed-loop). Clearly, the proposed controller rapidly attracted the state variables of the heat exchanger to the equilibrium point and stabilizes the system.

Finally, in this subsection, the comparisons of the performance for all six algorithms are introduced considering again the IAE, MAE. Tables 3 and 4 list the IAE and MAE, respectively for all controllers. The proposed algorithm has fewer values of IAE and MAE than other algorithms. Table 5 lists the computation time, NN structure and number of parameters for NN algorithms and only the computation time of the IPSO-PID controller. It' is clear that the computation time for the proposed controller is lower than other NN algorithms. On the other hand, the number of parameters for the proposed NN structure is smaller than those parameters for other NN algorithms. All simulations are performed by MATLAB scripts on

Table 3 Values of IAE

Algorithms	NN structure	Task 1	Task 2	Task 3	Task 4	Task 5
APID-PWORNN-AL	3-2-3	0.1023	0.2011	0.3718	0.1745	1.2660
APID-PWORNN-FL	3-2-3	0.1398	0.2404	0.3786	0.2564	1.5218
SOPIDNN [22]	2-18-3	0.1961	0.7761	0.6726	0.3203	2.3400
SOPIDNN [22]	3-2-3	2.1464	2.5925	2.7139	2.3263	4.0470
NNPID-LM [21]	2-5-1	0.1846	0.3163	0.4718	0.2746	1.8463
PID-RBFNN [20]	3-5-3	0.4087	1.1633	4.5168	1.1426	4.2659
PID-RBFNN [20]	3-2-3	1.0170	1.9461	7.1941	2.6590	4.7354
PID-NNEI [19]	5-51	0.2560	0.4671	0.7890	0.3717	1.9126
IPSO-PID [29]	–	0.1927	0.3798	1.4306	0.3596	1.7104
Conventional-PID	–	2.3021	5.1825	15.873	5.1544	6.2223

Table 4 Values of MAE

Algorithms	NN structure	Task 1	Task 2	Task 3	Task 4	Task 5
APID-PWORNN-AL	3-2-3	0.0205	0.0040	0.0124	0.0058	0.0422
APID-PWORNN-FL	3-2-3	0.0279	0.0048	0.0126	0.0085	0.0507
SOPIDNN [22]	2-18-3	0.0392	0.0155	0.0224	0.0107	0.078
SOPIDNN [22]	3-2-3	0.4293	0.0519	0.0905	0.0775	0.1349
NNPID-LM [21]	2-5-1	0.0369	0.0063	0.0157	0.0092	0.0615
PID-RBFNN [20]	3-5-3	0.0817	0.0233	0.1506	0.0381	0.1422
PID-RBFNN [20]	3-2-3	0.2034	0.0389	0.2398	0.0886	0.1578
PID-NNEI [19]	5-5-1	0.0512	0.0093	0.0263	0.0124	0.0638
IPSO-PID [29]	–	0.0385	0.0076	0.0477	0.0120	0.0570
Conventional-PID	–	0.4604	0.1036	0.5291	0.1718	0.2074

Table 5 Computation time, NN structure and number of parameters for all algorithms

Algorithms	NN structure	No of parameters	Computation time (ms)
APID-PWORNN -FL	3-2-3	6	0.3451
APID-PWORNN-AL	3-2-3	6	0.373
SOPIDNN [22]	2-18-3	90	2.900
SOPIDNN [22]	3-2-3	12	0.4266
NNPID-LM [21]	2-5-1	24	0.5448
PID-RBFNN [20]	3-5-3	25	0.6480
PID-RBFNN [20]	3-2-3	10	0.4148
PID-NNEI [19]	5-5-1	39	1.800
IPSO-PID [29]	–	–	2.420
Conventional-PID	–	–	0.1019

a PC with a processor Intel (R) Core (TM) i3 CPU M350 @ 2.27 GHz, RAM 6.0 GB, 64-bit operating system, and Windows 7.

The main advantages of the proposed APID-PWORNN controller over other controllers are summarized as:

- It possesses a stable learning algorithm because the learning algorithm is developed based on the Lyapunov stability criteria.
- It has less computation time and less number of tunable parameters as shown in Table 5, and simple structure as shown in Fig. 2.
- Moreover, the proposed controller recorded the minimum values of the performance indices such as IAE and MAE as indicated by Tables 1, 2, 3, and 4 that explore its computation accuracy compared to the existing controllers published previously.

6 Conclusions

In this paper, a novel structure of an adaptive PID controller based on a polynomial weighted output recurrent neural network and an adaptive learning rate algorithm is introduced. The simulation results proved that the proposed controller has the superiority for controlling the complex nonlinear dynamical systems and the robustness performance is examined through five tasks with two case studies (mathematical nonlinear system and heat exchanger system). Moreover, the optimization, the stability and the convergence speed are achieved by deriving parameters update rule and adaptation formula of the learning rate using a Lyapunov function. The proposed APID-PWORNN structure with a fewer number of tunable parameters, i.e., 6 weights, considered as a simple NN structure that reduced the computation time and it is applicable for microcontrollers with low-speed processors.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB). Authors are required to disclose financial or non-financial interests that are directly or indirectly related to the work submitted for publication.

Declarations

Conflict of interest There is no conflict of interest between the authors to publish this manuscript.

Appendix

$$L_{11P} = (2b\Delta e(k) + 2be(k) + 2cw_{1P}(k)),\ L_{21P} = (2a\Delta e(k) + 2ae(k) + 2bw_{1P}(k))$$

$$L_{12P} = (2b\Delta e(k) + 2be(k) + 2cw_{2P}(k)),\ L_{22P} = (2a\Delta e(k) + 2ae(k) + 2bw_{2P}(k))$$

$$L_{11I} = (2b\Delta e(k) + 2be(k) + 2cw_{1I}(k)),\ L_{21I} = (2a\Delta e(k) + 2ae(k) + 2bw_{1I}(k))$$

$$L_{12I} = (2b\Delta e(k) + 2be(k) + 2cw_{2I}(k)),\ L_{22I} = (2a\Delta e(k) + 2ae(k) + 2bw_{2I}(k))$$

$$L_{11D} = (2b\Delta e(k) + 2be(k) + 2cw_{1D}(k)),\ L_{21D} = (2a\Delta e(k) + 2ae(k) + 2bw_{1D}(k))$$

$$L_{12D} = (2b\Delta e(k) + 2be(k) + 2cw_{2D}(k)),\ L_{22D} = (2a\Delta e(k) + 2ae(k) + 2bw_{2D}(k))$$

References

1. Kumar R, Srivastava S, Gupta JRP (2017) Diagonal recurrent neural network based adaptive control of nonlinear dynamical systems using lyapunov stability criterion. ISA Trans 67:407–427
2. Kumar R, Srivastava S, Gupta JRP, Mohindru A (2018) Self-recurrent wavelet neural network–based identification and adaptive predictive control of nonlinear dynamical systems. Int J Adapt Control Signal Process 32(9):1326–1358
3. Kumar R, Srivastava S (2020) Externally Recurrent Neural Network based identification of dynamic systems using Lyapunov stability analysis. ISA Trans 98:292–308
4. Vázquez LA, Jurado F, Castañeda CE, Alanis AY (2019) Real-time implementation of a neural integrator backstepping control via recurrent wavelet first order neural network. Neural Process Lett 49(3):1629–1648
5. Liu T, Liang S, Xiong Q, Wang K (2020) Data-based online optimal temperature tracking control in continuous microwave heating system by adaptive dynamic programming. Neural Process Lett 51(1):167–191
6. Ma L, Yao Y, Wang M (2016). The optimizing design of wheeled robot tracking system by PID control algorithm based on BP neural network. In: 2016 International Conference on Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII) (pp. 34–39). IEEE
7. Li J, Gómez-Espinosa A (2018) Improving PID Control based on neural network. In: 2018 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE) (pp. 186–191). IEEE
8. Bari S, Hamdani SSZ, Khan HU, ur Rehman M, Khan H (2019). Artificial neural network based self-tuned PID controller for flight control of quadcopter. In: 2019 International Conference on Engineering and Emerging Technologies (ICEET) (pp. 1–5). IEEE.
9. Luoren L, Jinling L (2011) Research of PID control algorithm based on neural network. Energy Procedia 13:6988–6993

10. Yangxu X, Danhong Z, Huaiun Z, Lianshun W, Yue Q, Zhiwen L (2018) Neural network-fuzzy adaptive PID controller based on VIENNA rectifier. In: Chinese Automation Congress (CAC) (pp. 583–588). IEEE
11. Aftab MS, Shafiq M (2015) Adaptive PID controller based on Lyapunov function neural network for time delay temperature control. In: IEEE 8th GCC Conference & Exhibition (pp. 1–6). IEEE.
12. Jacob R, Murugan S (2016). Implementation of neural network based PID controller. In: 2016 International Conference on electrical, electronics, and optimization techniques (ICEEOT) (pp. 2769–2771). IEEE.
13. Mahmud K (2013) Neural network based PID control analysis. In: IEEE Global High Tech Congress on Electronics (pp. 141–145). Kazemy A, Hosseini SA Farrokhi M (2007). Second order diagonal recurrent neural network. In 2007 IEEE International Symposium on Industrial Electronics (pp. 251–256). IEEE
14. Meng Y, Zhiyun Z, Fujian R, Yusong P, Xijie G (2014) Application of adaptive PID based on RBF neural networks in temperature control. In: Proceeding of the 11th World Congress on Intelligent Control and Automation (pp. 4302–4306). IEEE
15. Kumar R, Srivastava S, Gupta JRP (2016). Artificial neural network based PID controller for online control of dynamical systems. In: IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES) (pp. 1–6). IEEE.
16. Günther J, Reichensdörfer E, Pilarski PM, Diepold K (2020) Interpretable PID parameter tuning for control engineering using general dynamic neural networks: an extensive comparison. PLoS ONE 15(12):e0243320
17. Agrawal A, Goyal V, Mishra P (2019) Adaptive control of a nonlinear surge tank-level system using neural network-based PID controller. In: Malik H, Srivastava S, Sood YR, Ahmad A (eds) Applications of artificial intelligence techniques in engineering: SIGMA 2018, Volume 1. Springer, Singapore, pp 491–500. https://doi.org/10.1007/978-981-13-1819-1_46
18. Rosales C, Soria CM, Rossomando FG (2019) Identification and adaptive PID Control of a hexacopter UAV based on neural networks. Int J Adapt Control Signal Process 33(1):74–91
19. Cho CN, Song YH, Lee CH, Kim HJ (2018) Neural network-based real time PID gain update algorithm for contour error reduction. Int J Precis Eng Manuf 19(11):1619–1625
20. Pu Q, Zhu X, Zhang R, Liu J, Cai D, Fu G (2020) Speed profile tracking by an adaptive controller for subway train based on neural network and PID algorithm. IEEE Trans Veh Technol 69(10):10656–10667
21. Hao J, Zhang G, Liu W, Zheng Y, Ren L (2020) Data-driven tracking control Based on LM and PID neural network with relay feedback for discrete nonlinear systems. IEEE Trans Ind Electron 68(11):11587–11597
22. Ben Jabeur C, Seddik H (2021) Design of a PID optimized neural networks and PD fuzzy logic controllers for a two-wheeled mobile robot. Asian J Control 23(1):23–41
23. Patrikar A, Provence J (1996) Nonlinear system identification and adaptive control using polynomial networks. Math Comput Model 23(1–2):159–173
24. González T, Sala A, Bernal M (2019) A generalized integral polynomial Lyapunov function for nonlinear systems. Fuzzy Sets Syst 356:77–91
25. Kazemy A, Hosseini SA, Farrokhi M (2007) Second order diagonal recurrent neural network. In: IEEE International Symposium on Industrial Electronics 2007 (pp. 251-256). IEEE
26. Lisang L, Xiafu P (2012) Discussion of stability on recurrent neural networks for nonlinear dynamic systems. In: 7th International Conference on Computer Science & Education (ICCSE) (pp. 142-145). IEEE.
27. Peng J, Dubay R (2011) Identification and adaptive neural network control of a DC motor system with dead-zone characteristics. ISA Trans 50(4):588–598
28. Kumar R, Srivastava S, Gupta JRP, Mohindru A (2018) Diagonal recurrent neural network based identification of nonlinear dynamical systems with Lyapunov stability based adaptive learning rates. Neurocomputing 287:102–117
29. Feng H, Ma W, Yin C, Cao D (2021) Trajectory control of electro-hydraulic position servo system using improved PSO-PID controller. Autom Constr 127:103722
30. Khater AA, El-Nagar AM, El-Bardini M, El-Rabaie NM (2020) Online learning based on adaptive learning rate for a class of recurrent fuzzy neural network. Neural Comput Appl 32(12):8691–8710
31. Xu D, Jiang B, Shi P (2014) Adaptive observer based data-driven control for nonlinear discrete-time processes. IEEE Trans Autom Sci Eng 11(4):1037–1045
32. Eskinat E, Johnson SH, Luyben WL (1991) Use of Hammerstein models in identification of nonlinear systems. AIChE J 37(2):255–268
33. Berger MA, da Fonseca Neto JV (2013) Neurodynamic programming approach for the PID controller adaptation. IFAC Proc 46(11):534–539
34. Paredes GE (2020) Fractional-order models for nuclear reactor analysis. Woodhead Publishing