https://doi.org/10.1007/s40998-018-0069-1

RESEARCH PAPER

Chaotic Self-Tuning PID Controller Based on Fuzzy Wavelet Neural Network Model

Mehrnoush Davanipour[1] · Hamidreza Javanmardi[1] · Nasrin Goodarzi[1]

Received: 22 July 2016 / Accepted: 12 June 2018 / Published online: 27 June 2018

Abstract

PID controller has been quite successful when its parameters are tuned properly. However, it fails in varying situations. This is even more critical where the system is not known. In this paper, after illustrating the capability of fuzzy wavelet neural network (FWNN) in modeling of nonlinear systems, a self-tuning PID controller based on this model has been designed. The auto-tuner effectively handles the limitation of PID controller in unpredictable conditions such as environmental changes. Chaotic optimization method which is a robust algorithm of escaping from local minimum is applied for tuning of the controller parameters. This supports finding optimal values of controller parameters in a short time which enables online implementation. Unlike most of the researches in this area, the tuning rules could be begun without any trial and error. Since it contains no extra parameters, it has the feature of simplicity in the tuning. The proposed method with few parameters has the ability to increase the speed of tracking with very little steady-state error. The capability of FWNN in the modeling of nonlinear systems with a few rules and susceptibility of the proposed controller will be shown by simulation.

Keywords Fuzzy wavelet neural network · Auto-tuning controller · Chaos optimization

1 Introduction

Nowadays, fuzzy logic system is widely used in control and modeling applications. The learning ability of the fuzzy logic system is improved in fuzzy neural networks, due to the capability of fuzzy reasoning in the management of uncertain information and the ability of neural networks in learning from processes (Wai et al. 2015; Tang et al. 2017). Also, combining the wavelets with neural networks yields quick convergence, high precision and reduced network size. Hence, for optimizing the number of fuzzy rules and improving the approximation error and control precision, fuzzy wavelet neural network (FWNN) has been constructed. The first FWNN was proposed by Ho et al. (2001). It was based on multi-resolution analysis of wavelet transforms and was applied for approximation of nonlinear functions. Another FWNN structure has been presented in Abiyev and Kaynak (2008), which is based on the addition of wavelet functions in consequent parts of fuzzy rules. Then, it was used to control nonlinear dynamic plants. There have been many researches in recent years using FNN (FWNN and ANFIS) in various applications (Davanipour et al. 2012; Lin et al. 2014; Hung et al. 2015; Chen et al. 2015; Mai and Wang 2014; Engin et al. 2004; Kayacan and Kaynak 2006; Loussifi et al. 2016; Wang and Cao 2015; Tofighi et al. 2015). To control a six-phase permanent magnet synchronous motor for an electric power steering system, Lin et al. (2014) proposed an intelligent second-order sliding mode control using a FWNN with an asymmetric membership function estimator. For the same system, Hung et al. (2015) used FWNN based on asymmetric membership function with improved differential evolution algorithm. A recurrent FWNN has been adopted to control the rotor position of a thrust magnetic bearing in Chen et al. (2015). Control of manipulator robot (Mai and Wang 2014), liquid-level systems (Engin et al. 2004; Kayacan and Kaynak 2006), system identification (Loussifi et al. 2016), predicting power consumption (Wang and Cao 2015) and control of

✉ Mehrnoush Davanipour
m.davanipour@shirazu.ac.ir

[1] Department of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran

chaotic systems (Tofighi et al. 2015) are some of FWNN applications.

The PID-type controller is most widely used in many applications, primarily because of its simple structure, clear functionality and easy implementation (Kansha et al. 2008). However, it suffers from a main drawback. The precise knowledge of the plant should be available to tune the PID controller. This requirement is not easily removed in most industrial cases. In addition, even if there is a mathematical model of the system, environmental conditions may cause the obtained model to change, so it may be necessary to design another PID controller to fit the new changed model.

To overcome the mentioned limitations of the conventional PID controller, researchers have tried to apply self-tuning PID controller (Al Gizi et al. 2014; Li et al. 2005; Woo et al. 2000; Dhaouadi et al. 2008; Zheng et al. 2009; Nguyen et al. 2015). A self-tuning PID controller has the ability to tune its parameters according to the changes of the process. A radial basis function neural network has been used to enhance the PID parameters obtained from genetic algorithm in order to design Sugeno fuzzy PID controller (Al Gizi et al. 2014). A self-tuning PID controller based on wavelet neural networks was proposed in Li et al. (2005). In that work, two wavelet neural networks were used for identification and online tuning. The parameters of PID controller were the output of the second employed wavelet network. So, it was necessary to obtain suitable amounts of parameters as training data. In other words, it is not useful in practice when we have no previous information about the closed-loop system. In Woo et al. (2000), a PID-type fuzzy controller with self-tuning scaling factors has been proposed. They developed a method for online tuning of the scaling factors of PID controller. However, they have added four extra parameters which should be tuned. Determination of these extra parameters is another limitation, too. In Dhaouadi et al. (2008), a self-tuning adaptive PID controller was proposed using a dynamic wavelet network. The control scheme was tested with a second-order system with input saturation. The controller finally has tracked the desired signal; however, as it can be seen from the simulation results, it takes a long time. To improve the overall performance of servo-hydraulic press driven directly by switched reluctance motor, a fuzzy PID control method has been introduced in Zheng et al. (2009). The fuzzy rules have been established based on the error and change in error to adaptive adjustment of PID parameters. But, a lot of rules are needed for tuning of the PID coefficients. In Dhaouadi et al. (2008), an auto-tuning of the PID controller based on the radial basis function neural network and relay feedback approach has been introduced. As stated in Nguyen et al. (2015), ultimate gain and ultimate frequency of the control system have been recognized by the relay feedback approach for initial conditions of the neural network controller. But, this method increases the complexity of the control system.

In this paper to solve the mentioned problems concerned with PID controller, a self-tuning PID controller based on the FWNN has been proposed. By this way, there is no need to have an exact or approximation mathematical model of the plant. Besides, it enables us to benefit the advantages of FWNN. FWNN has some advantages over other intelligent networks. Since the consequent parts contain wavelet functions, it is able to approximate the details of system with higher precision.

However, as we know, one of the main difficulties with auto-tuning is having an efficient optimization method which could be suitable for online applications. One of the modern optimization algorithms is the chaos-based optimization (Jiaqiang et al. 2015; Farahani et al. 2012; Lv and Wu 2006; Ganjefar and Alizadeh 2012; Li and Jiang 1998; Yuan et al. 2012; Yuan et al. 2015). In Farahani et al. (2012), in order to solve the load frequency control problem, a PID optimized by Lozi map chaotic algorithm was proposed. In chaos optimization algorithm, the optimizing variables are transferred into chaotic variables according to the index function and the initial value. Then, by considering the condition of constraint, the optimal values are originated through the search of chaotic variables. Finally, to determine the ultimate optimal values of the system, a fine chaotic search is performed in the neighborhood of the optimal values (Jiaqiang et al. 2015). Chaos optimization algorithm finds the optimal values in higher speed than stochastic ergodic searches. Escape from local minima, independence from strict mathematical properties of optimization problem, easy implementation and short execution time are the main advantages of chaos optimization algorithm (Yuan et al. 2012; Yuan et al. 2015).

In this paper, tent map chaotic optimization has been applied to find the optimal values of the PID coefficients. Design of the controller is performed in two steps. In the first step, the system is modeled using FWNN. This network can identify the unknown model with appropriate accuracy using a few number of parameters. Then, in the second step, a self-tuning PID controller is designed based on the obtained model. The controller's parameters are tuned using simple but powerful and robust tent map chaotic optimization algorithm. The tuning process can be begun without any trial and error unlike most of the researches in this area (Lin et al. 2014; Hung et al. 2015; Chen et al. 2015; Woo et al. 2000). Also, it does not include any additional parameters to be tuned or to be selected randomly. As the simulation results will show, the control performance is suitable, especially, in the steady-state error and in the speed of setpoint tracking.

This paper is organized as follows: Sect. 2 gives the structure of FWNN. Section 3 represents the proposed controller design. Simulation results are provided in Sect. 4. Section 5 includes the conclusion remarks.

2 Fuzzy Wavelet Neural Network

A fuzzy wavelet neural network integrates wavelet functions with Takagi–Sugeno–Kang fuzzy model. The structure of FWNN is given in Fig. 1. It is constructed by seven layers. The fourth layer is the consequent layer which includes wavelet neural network. In the figure, this subnet has denoted by WNN block. Each WNN corresponds to a three-layer structure, using wavelets as activation functions. The relation of (1) describes the FWNN structure (Abiyev and Kaynak 2008):

$$\begin{aligned} R_i:\quad & \text{If} x_1 is A_1^i, \quad \text{and} \quad x_2 is A_2^i \ldots \quad \text{and} \quad x_q \quad is \quad A_q^i \\ & \text{Then} \quad y_i = w_i \sum_{j=1}^{q} \psi_{ij}(x_j) \end{aligned} \tag{1}$$

where $x_j (1 \leq j \leq q)$ is the jth input and $y_i (1 \leq i \leq c)$ is the output of the local model for rule R_i. A_q^i is a membership function for ith rule of the qth input defined as a Gaussian function. ψ_{ij} is defined later in (3). The output signal of each WNN will be such as

$$y_i = w_i \sum_{j=1}^{q} \psi_{ij}(x_j) \tag{2}$$

in which w_i shows the weight coefficient between the inputs and ith output. ψ_{ij} indicates wavelet family defined in (3):

$$\psi_{ij}(x_j) = \psi\left(\frac{x_j - b_{ij}}{a_{ij}}\right), \quad a_{ij} \neq 0 \tag{3}$$

In the above relation, ψ_{ij} is made of a mother wavelet function using dilations and translations parameters (a, b).

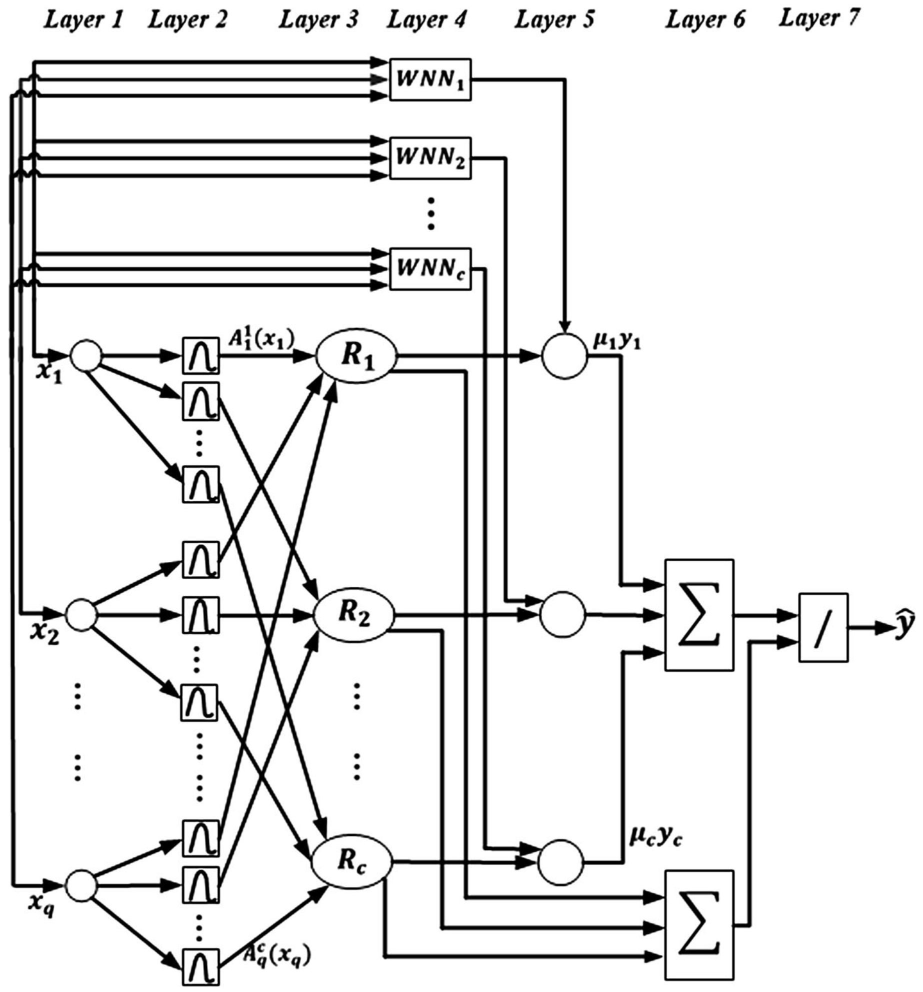

Fig. 1 Structure of FWNN (Davanipour et al. 2012)

The structure of WNN is given in Fig. 2. Each WNN corresponds to a three-layer structure, using wavelets as activation functions.

The defuzzification is made to calculate the output of the whole network in the sixth and the seventh layers. Consequently, the output of FWNN could be calculated as

$$\hat{y} = \frac{\sum_{i=1}^{c} \mu_i y_i}{\sum_{i=1}^{c} \mu_i} \quad \text{where} \quad \mu_i(x) = \prod_{j=1}^{q} A_j^i(x_j) \tag{4}$$

Here, c is the number of fuzzy rules, q is used as an indicator for dimension of input vector, and y_i are the output signals of the wavelet neural networks. In this paper, in order to describe the linguistic terms, the Gaussian membership functions are used. Gaussian membership function can approximate triangular and trapezoidal membership functions (Ho et al. 2001):

$$A_j^i(x_j) = \exp\left[-\frac{\left(x_j - c_j^i\right)^2}{\left(\sigma_j^i\right)^2}\right] \tag{5}$$

in which σ_j^i and c_j^i indicate the center and the half-width of membership function, respectively. Also, Mexican Hat wavelet function which is commonly applied in FWNN is used here (Ho et al. 2001; Abiyev and Kaynak 2008; Davanipour et al. 2012):

$$\psi(x) = \frac{1}{\sqrt{|a|}}\left(1 - 2x^2\right)\exp\left(-\frac{x^2}{2}\right) \tag{6}$$

It is a mother wavelet function, the dilated and translated versions of which will be used in consequent parts of each fuzzy rule. The use of wavelets with different dilation and translation values allows us to capture different behaviors and essential features of the nonlinear model under fuzzy rules.

3 Proposed Controller

In the previous section, the FWNN model was described. It is used for modeling of the unknown plant. In this section, the proposed controller will be designed. At first, the structure of the self-tuning controller is designed, and then, the optimization procedure is explained.

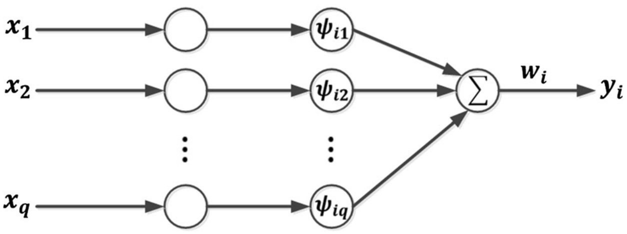

Fig. 2 Structure of WNN (Davanipour et al. 2012)

3.1 Auto-Tuning Controller Design

The structure of closed-loop system constructed with identifier network and auto-tuning PID controller is shown in Fig. 3. In that figure, e_{id} is the identification error which is the difference between the actual output and the output of the model. The unknown plant is identified with FWNN. Then, the PID controller is designed based on this model. FWNN has some advantages over other intelligent networks. Since the consequent parts contain wavelet functions, it is able to approximate the details of system with higher precision.

The adjustment of control parameters (P, I, D) to reach optimum values is the duty of the control loop tuning. The auto-tuner calculates the optimum control parameters by minimizing the following cost function:

$$E = \frac{1}{2}\sum_{k=1}^{T}\left(y_d(k) - \hat{y}(k)\right)^2 \tag{7}$$

where $y_d(k)$ is the desired output and $y(k)$ is the actual output of the system in sample time k. T is integrating horizon. Here, it is equal to 20. The structure of PID controller is as follows:

$$\begin{aligned} u(k) = u(k-1) + P[e(k) - e(k-1)] + Ie(k) \\ + D[e(k) - 2e(k-1) + e(k-2)] \end{aligned} \tag{8}$$

where $e(k)$ is the difference between the desired output and the actual output. According to (7) and (8), the cost function contains P, I and D called control parameters. So, by finding the optimum values of these parameters the cost function in (7) will be minimized. The next section is allocated to finding these optimum parameters.

3.2 Optimization Procedure

After designing the structure of model-based controller, it is necessary to apply an efficient optimization algorithm to tune the controller parameters. By comparison with the next section, one can conclude that the gradient method

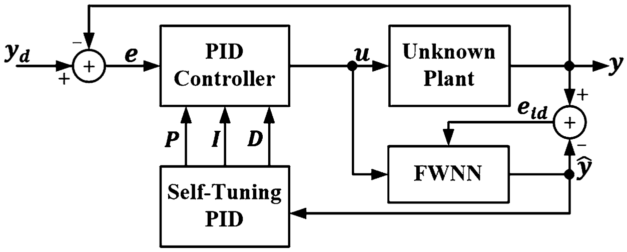

Fig. 3 Structure of the proposed controller

described at Appendix needs more computations and handling with formula and derivation calculations. To reduce the computations, increase the robustness in noisy conditions and get faster and confident tracking, it is appropriate to apply chaos-based optimization.

As it was discussed in the Introduction section, chaos-based optimization can escape from local minima. The full description of this group of algorithms can be found in Li and Jiang (1998). In this context, tent map chaos optimization has been used. The equation of tent map to generate chaotic time series is:

$$x_{i+1} = \begin{cases} 2x_i & x_i \in [0, 0.5] \\ 2(1 - x_i) & x_i \in [0.5, 1] \end{cases} \tag{9}$$

where x_i is the chaotic variable.

Here, the main problem is finding the parameters $[P, I, D]$ via tent map chaos optimization. Since three parameters are needed to be optimized, three initial chaotic variables $0 \leq x_{1,0}, x_{2,0}, x_{3,0} \leq 1$ are selected randomly. Then, for each variable the lower bound and upper bound are defined and indicated by l_{bd} and u_{bd}. After that, the initial value for $[P, I, D]$ and E (cost function) should be selected and denoted as $[P^*, I^*, D^*]$ and E^*. Here, zero and infinite have been considered, respectively. The major steps of the FWNN-PID-based tent map chaos optimization are described as follows:

Step 1. At first sample time, compute the cost function and determine $E([P, I, D])$.

Step 2. Apply the tent map chaos optimization to find new parameters which minimize the cost function:

- Substitute $x_{1,k}, x_{2,k}, x_{3,k}$ in (9) to generate three new chaotic variables via tent map in sample time k.
- Using the following formula, namely first carrier wave, compute the $(k+1)$th parameters:

$$\begin{bmatrix} P_{k+1} \\ I_{k+1} \\ D_{k+1} \end{bmatrix} = \begin{bmatrix} l_{\mathrm{bd}-P} \\ l_{\mathrm{bd}-I} \\ l_{\mathrm{bd}-D} \end{bmatrix} + \left(\begin{bmatrix} u_{\mathrm{ub}-P} \\ u_{\mathrm{ub}-I} \\ u_{\mathrm{ub}-D} \end{bmatrix} - \begin{bmatrix} l_{\mathrm{ld}-P} \\ l_{\mathrm{ld}-I} \\ l_{\mathrm{ld}-D} \end{bmatrix} \right) \cdot \begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \\ x_{3,k+1} \end{bmatrix}. \tag{10}$$

- Compute the cost function in (7) and assign the new optima as follows: If $E_{k+1} \leq E^*$, then $E^* = E_{k+1}, [P^*, I^*, D^*] = [P_{k+1}, I_{k+1}, D_{k+1}]$; otherwise, do nothing. Repeat the above steps until there is no improvement for cost function during certain steps. Then, go to the next step as follows:
- Again, use the tent map equation in (9) to generate new chaotic variables. Then, start to search utilizing the second carrier wave.

$$\begin{bmatrix} P_{k+1} \\ I_{k+1} \\ D_{k+1} \end{bmatrix} = \begin{bmatrix} P^* \\ I^* \\ D^* \end{bmatrix} + \beta \begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \\ x_{3,k+1} \end{bmatrix} \tag{11}$$

in which β is a chaos variable with small ergodic interval. It is normally chosen as a small positive constant, here, $\beta = 0.001$.

- Again, compute the cost function in (7), and assign the new optima such as follows:

  If $E_{k+1} \leq E^*$, then $E^* = E_{k+1}, [P^*, I^*, D^*] = [P_{k+1}, I_{k+1}, D_{k+1}]$; otherwise, do nothing.

  Repeat the above steps until there is no improvement for cost function during certain steps.

Step 3. Fit the obtained optimum parameters to the controller and send the controller output to the process.

Step 4. For the next sample time, go to the step 1.

4 Simulation Results

In this section, two examples have been considered to investigate the proposed controller. The first example is a nonlinear dynamic system, and the second example is a nonlinear liquid-level system.

Example 1 Consider the nonlinear dynamic plant as below:

$$y(k) = f(y(k-1), y(k-2), y(k-3), u(k), u(k-1)) \tag{12}$$

in which

$$f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_2^2 + x_3^2} \tag{13}$$

The identification process of the above system is done using the FWNN described in Sect. 2. Here, FWNN is constructed with only two rules. Then, it is trained using the algorithm proposed in Davanipour et al. (2012). Details of the separate parts of this algorithm such as clustering, recursive least square and back propagation algorithms can be found in many reports (Shi et al. 2010; Nells 2001; Jang et al. 1997). After training, an input such as what is shown below has been applied to the FWNN as the test input.

$$u(k) = \begin{cases} \sin\left(\frac{\pi k}{25}\right) & k < 250 \\ 1.0 & 250 \leq k < 500 \\ -1.0 & 500 \leq k < 750 \\ 0.3\sin\left(\frac{\pi k}{25}\right) + 0.1\sin\left(\frac{\pi k}{32}\right) + 0.6\sin\left(\frac{\pi k}{10}\right) & 750 \leq k < 1000 \end{cases} \tag{14}$$

The result of system identification is shown in Fig. 4. The capability of FWNN in nonlinear dynamic system identification can be seen in that figure. After the identification stage, the proposed controller has been applied to the plant and its performance has been verified in both normal and noisy conditions. The reference signal is unit step, and the final values of the PID parameters are $P = 1.4, I = 2, D = 1.2$. Figure 5 shows the performance of the controller when there is no noise in the measurement. Tracking error and the control input are illustrated in Figs. 6 and 7, respectively. The performance of the controller has been also tested when the output is contaminated with a 3 db white noise. Figure 8 illustrates that the chaos-based FWNN-PID controller can be robust in this condition. In order to have a comparison, the gradient-based controller is also tested in noisy condition and Fig. 9 shows that the output has become unstable. Therefore, it cannot track the desired output. Table 1 represents a brief summary of this example in normal condition and compares the proposed controller with the other approaches reported in the literature. In Table 1, MAE points to mean absolute of the tracking error. As can be seen, no overshoot, less settling time and less MAE value in chaos-based FWNN-PID controller have been achieved, despite the considerably smaller number of parameters and rules.

Example 2 The plant considered in this example is a nonlinear model of a liquid-level system because it is difficult to be controlled optimally using only a conventional PID controller as the parameters of the plant are changing continuously. The plant has a relation as shown below:

$$h(t) = h(0) + \frac{1}{A}\int_0^t (q_{\text{in}}(\tau) - q_{\text{out}}(\tau))\mathrm{d}\tau \tag{15}$$

in which

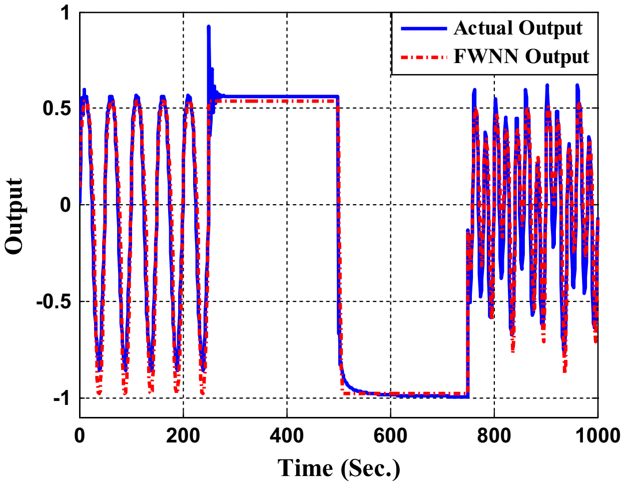

Fig. 4 The comparison between the actual output and the output of FWNN in the identification of system (Example 1)

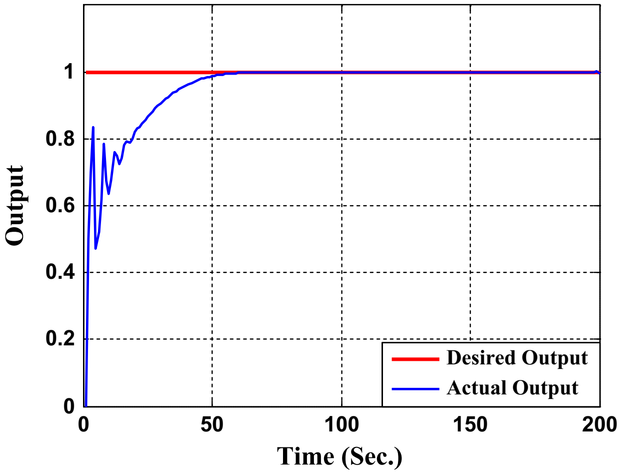

Fig. 5 Performance of the proposed controller (Example 1)

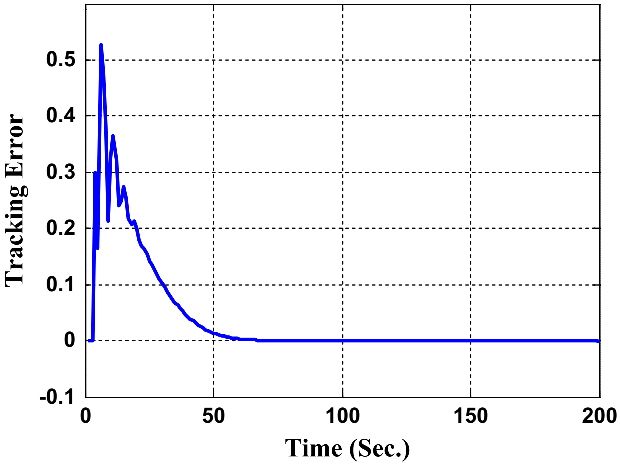

Fig. 6 Tracking error (Example 1)

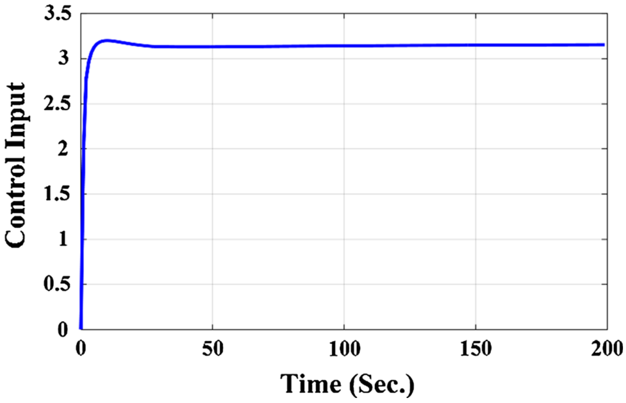

Fig. 7 Control Input

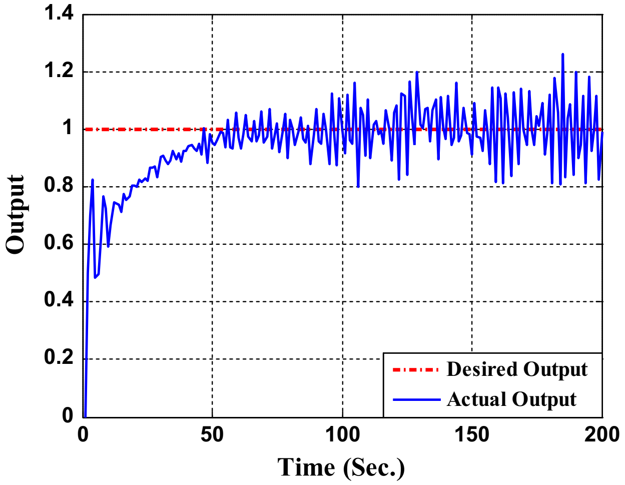

Fig. 8 Performance of the proposed controller in noisy condition (Example 1)

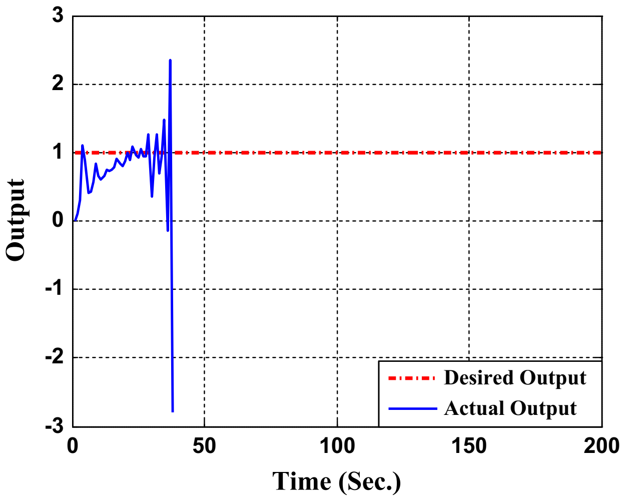

Fig. 9 Performance of the gradient-based controller in noisy condition (Example 1)

$$
\begin{aligned}
q_{\text{in}} &= Q_{\text{in}}\sin(\theta(t)), \quad \theta(t) \in \left[0, \frac{\pi}{2}\right] \\
q_{\text{out}} &= a_{\text{out}}\sqrt{2gh(t)}
\end{aligned} \tag{16}
$$

In the above relation, h is the output variable in m which is the level of the liquid, $a_{out} = 0.01\,\mathrm{m}^2$ is the surface area of the outlet, A is the surface of the tank ($1\ \mathrm{m}^2$), θ is the control valve's flap angle in *rad*, and $g = 9.81\,\mathrm{m/s}^2$. The details of this example can be found in Engin et al. (2004).

As the previous example, here, the plant is also first identified by a FWNN constructed with two rules; then, the proposed controller has been applied. Here, the reference signal is unit step and the final values of the PID parameters are $P = 0.95$, $I = 0.15$, $D = 0.03$. The results are shown in Figs. 10, 11, 12 and 13.

In order to consider some practical points of PID controllers, in this example disturbance attenuation, control signal amplitude limitations and windup problem have also been included. It is assumed that there is a constraint on the control input such that the input cannot be larger than 1.4, in amplitude. So, the control input is saturated, even though the output has not tracked the desired output correctly. The result of this situation is shown in Fig. 14. As the figure shows, the response is still satisfactory, although there is a larger overshoot and settling time in comparison with the previous status. In Fig. 15, the output tracking has been depicted when there is a 30 percent step signal as an input disturbance. As the figure shows, the disturbance is attenuated after a short time.

In Table 2, a brief comparison of the proposed method with gradient-based FWNN-PID, ANFIS-based PID and ANFIS-based fuzzy (Engin et al. 2004) is presented. Improvements in settling time, overshoot and MAE are clear, in spite of the smaller number of parameters and rules, based on the results of Table 2.

5 Conclusion

Using the capability of FWNN in modeling of nonlinear dynamic systems, an auto-tuning controller was designed. Tent map chaotic optimization algorithm made the controller powerful in finding the optimal amounts of parameters. The proposed controller obviated the famous drawback of PID controller which is still an open problem, especially when the system has an unknown structure. The simulation part demonstrated the good performance of the controller in both transient and steady-state response.

Table 1 The results of Example 1

Controller	Number of rules	Number of parameters	Settling time (s)	Overshoot (percentage)	MAE
Chaos-based FWNN-PID	2	18	50	–	0.02
Gradient-based FWNN-PID	2	18	60	20	0.06
ANFIS-based PID	25	175	90	40	0.1

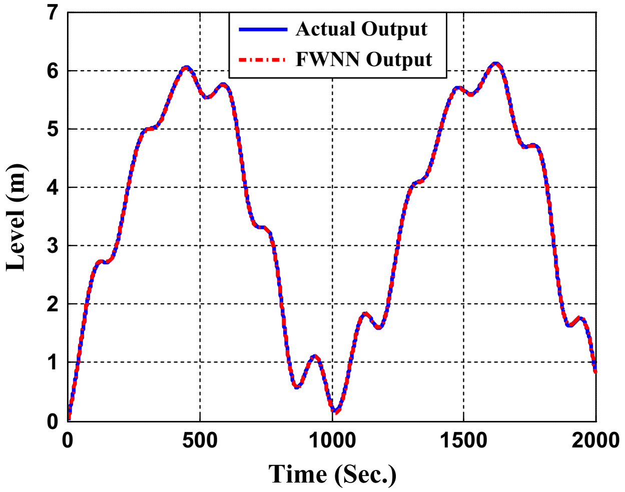

Fig. 10 The comparison between the actual output and the output of FWNN in the identification of system (Example 2)

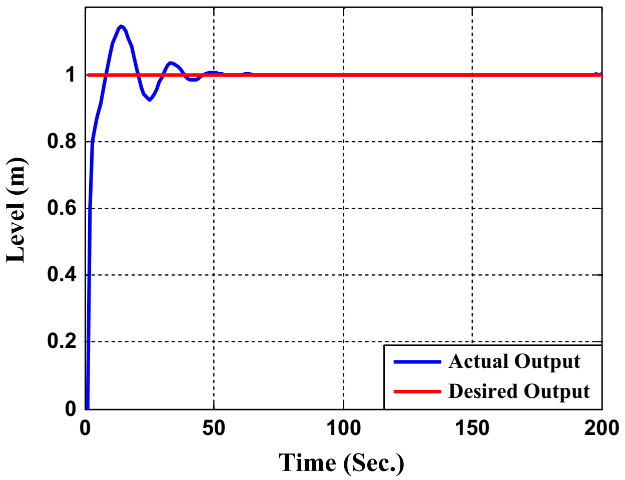

Fig. 11 Performance of the proposed controller (Example 2)

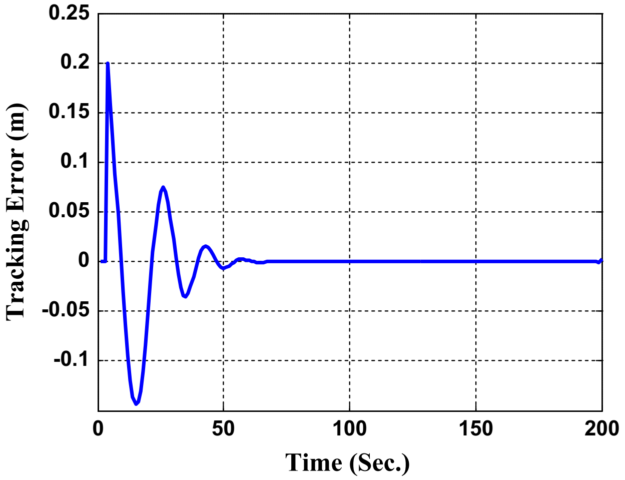

Fig. 12 Tracking error (Example 2)

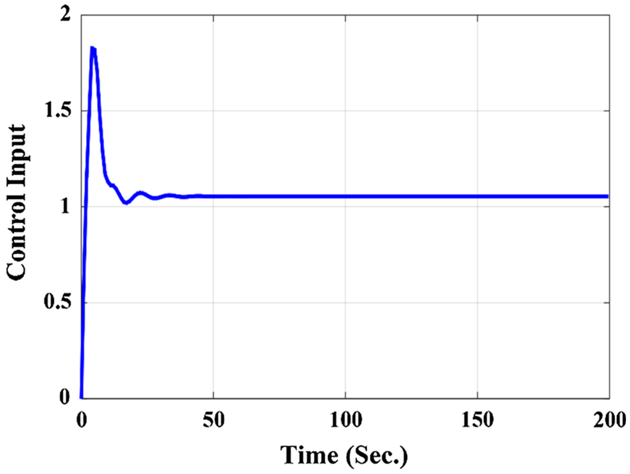

Fig. 13 Control input (Example 2)

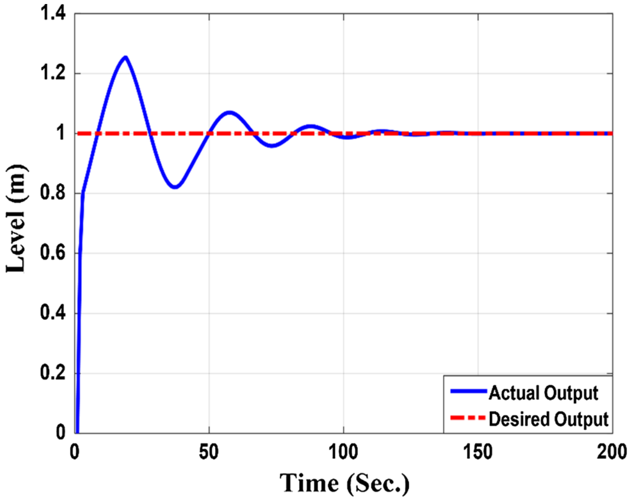

Fig. 14 Performance of the proposed controller with input saturation (Example 2)

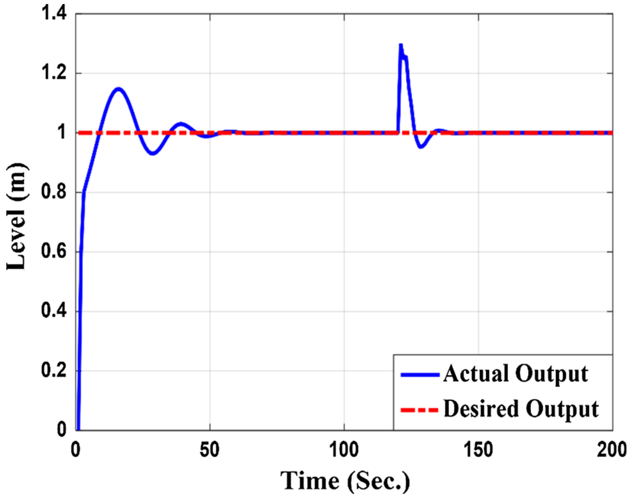

Fig. 15 Performance of the proposed controller in front of disturbance (Example 2)

Table 2 The results of Example 2

Controller	Number of rules	Number of parameters	Settling time (s)	Overshoot (percentage)	MAE
Chaos-based FWNN-PID	2	18	42	17	0.01
Gradient-based FWNN-PID	2	18	60	20	0.03
ANFIS-based PID	25	175	100	38	0.09
ANFIS-based fuzzy (Engin et al. 2004)	25	175	80	10	–

Appendix: Gradient-Based Optimization

According to the gradient descent algorithm and the cost function in (7), the optimization formulas are such as follows:

$$\begin{aligned}\Delta P(k) &= -\eta_P \frac{\partial E}{\partial P}\\ \Delta I(k) &= -\eta_I \frac{\partial E}{\partial I}\\ \Delta D(k) &= -\eta_D \frac{\partial E}{\partial D}\end{aligned} \tag{17}$$

in which, for instance, η_P indicates the learning rate for P parameter and $\frac{\partial E}{\partial P}$ is:

$$\frac{\partial E}{\partial P} = -\sum_{k=1}^{T} e(k)\frac{\partial \hat{y}}{\partial P} \tag{18}$$

In above relation, e is control error and $\hat{y}$ is the output of the model.

Using chain rule in derivation:

$$\frac{\partial E}{\partial P} = -\sum_{k=1}^{T} e(k)\left(\frac{\partial \hat{y}}{\partial u}\frac{\partial u}{\partial P}\right) \tag{19}$$

In above equation, $\frac{\partial \hat{y}}{\partial u}$ will be obtained according to (1–4) and $\frac{\partial u}{\partial P}$ according to (8).

The combination of (1–4) results in:

$$\hat{y} = \frac{\sum_{i=1}^{c}\left(\prod_{j=1}^{q} A_j^i(x_j)\right) w_i \sum_{j=1}^{q} \psi_{ij}(x_j)}{\sum_{i=1}^{c}\left(\prod_{j=1}^{q} A_j^i(x_j)\right)} \tag{20}$$

in which x is the network's input (u). Therefore, the derivation is such as

$$\frac{\partial \hat{y}}{\partial u} = \frac{\frac{\partial m}{\partial u} n - \frac{\partial n}{\partial u} m}{n^2} \tag{21}$$

in which

$$\begin{aligned} m &= \sum_{i=1}^{c}\left(\prod_{j=1}^{q} A_j^i(x_j)\right) w_i \sum_{j=1}^{q} \psi_{ij}(x_j)\\ n &= \sum_{i=1}^{c}\left(\prod_{j=1}^{q} A_j^i(x_j)\right)\end{aligned} \tag{22}$$

$\frac{\partial u}{\partial P}$ is such as:

$$\frac{\partial u}{\partial P}(k) = \frac{\partial u}{\partial P}(k-1) + e(k) - e(k-1) \tag{23}$$

The other optimization principles can be obtained in the same way.

References

Abiyev RH, Kaynak O (2008) Fuzzy wavelet neural networks for identification and control of dynamic plants—a novel structure and a comparative study. IEEE Trans Ind Electron 55(8):3133–3140

Al Gizi AJH, Mustafa MW, Jebur HH (2014) A novel design of high-sensitive fuzzy PID controller. Appl Soft Comput 24:794–805

Chen SY, Hung YC, Hung YH, Wu CH (2015) Application of a recurrent wavelet fuzzy-neural network in the positioning control of a magnetic-bearing mechanism, Computers & Electrical Engineering. Comput Electr Eng. https://doi.org/10.1016/j.compeleceng.2015.11.022

Davanipour M, Zekri M, Sheikholeslam F (2012) Fuzzy wavelet neural network with an accelerated hybrid learning algorithm. IEEE Trans Fuzzy Syst 20(3):463–470

Dhaouadi R, Al-Assaf Y, Hassouneh W (2008) A self tuning PID controller using wavelet networks. In: IEEE conference power electronics specialists, pp 773–777

Engin SN, Kuvulmaz J, Omurlu VE (2004) Fuzzy control of an ANFIS model representing a nonlinear liquid level system. Neural Comput Appl 13(3):202–210

Farahani M, Ganjefar S, Alizzadeh M (2012) PID controller adjustment using chaotic optimization algorithm for multi area frequency control. IET Control Theory Appl 6(13):1984–1992

Ganjefar S, Alizadeh M (2012) Online self-learning PID based PSS using self-recurrent wavelet neural network identifier and chaotic optimization. Int J Comput Math Electr Electron Eng 31(6):1872–1891

Ho DWC, Zhang PA, Xu J (2001) Fuzzy wavelet networks for function learning. IEEE Trans Fuzzy Syst 9(1):200–211

Hung YC, Lin FJ, Hwang JC, Chang JK, Ruan KC (2015) Wavelet fuzzy neural network with asymmetric membership function controller for electric power steering system via improved differential evolution. IEEE Trans Power Electron 30(4):2350–2362

Jang JSR, Sun CT, Mizutani E (1997) Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence, 1st edn. Prentice Hall, New Jersey

Jiaqiang E, Qian C, Liu H, Liu G (2015) Design of the robust control for the piezoelectric actuator based on chaos optimization algorithm. Aerosp Sci Technol 47:238–246

Kansha Y, Jia L, Chiu MS (2008) Self-tuning PID controllers based on the Lyapunov approach. Chem Eng Sci 63(10):2732–2740

Kayacan E, Kaynak O (2006) Grey prediction based control of a nonlinear liquid level system using PID type fuzzy controller. In: IEEE international conference on mechatronics, 292–296

Li B, Jiang W (1998) Optimizing complex functions by chaos search. Cybern Syst 24(4):409–419

Li H, Jin H, Guo C (2005) PID control based on wavelet neural network identification and tuning and its application to fin stabilizer. In: IEEE international conference mechatronics and automation, vol 4, pp 1907–1911

Lin FJ, Hung YC, Ruan KC (2014) An intelligent second-order sliding-mode control for an electric power steering system using a wavelet fuzzy neural network. IEEE Trans Fuzzy Syst 22(6):1598–1611

Loussifi H, Nouri K, Benhadj Braiek N (2016) A new efficient hybrid intelligent method for nonlinear dynamical systems identification: the Wavelet Kernel Fuzzy Neural Network. Commun Nonlinear Sci Numer Simul 32:10–30

Lv Q, Wu Y (2006) Research based on chaotic PID parameters optimization control method to the servo system. J Syst Simul 18:750–752

Mai T, Wang Y (2014) Adaptive force/motion control system based on recurrent fuzzy wavelet CMAC neural networks for condenser cleaning crawler-type mobile manipulator robot. IEEE Trans Control Syst Technol 22(5):1973–1982

Nells O (2001) Nonlinear system identification: form classical approaches to neural networks and fuzzy models, 1st edn. Springer, Berlin

Nguyen GH, Shin JH, Kim WH (2015) Auto tuning controller for motion control system based on intelligent neural network and relay feedback approach. IEEE/ASME Trans Mechatron 20(3):1138–1148

Shi N, Liu X, Guan Y (2010) Research on k-means clustering algorithm: an improved k-means clustering algorithm. In: International symposium on intelligent information technology and security informatics, pp 63–67

Tang J, Liu F, Zou Y, Wang Y (2017) An improved fuzzy neural network for traffic speed prediction considering periodic characteristic. IEEE Trans Intell Transp Syst 99:1–11

Tofighi M, Alizadeh M, Ganjefar S (2015) Direct adaptive power system stabilizer design using fuzzy wavelet neural network with self-recurrent consequent part. Appl Soft Comput 28:514–526

Wai RJ, Yao JX, Lee JD (2015) Backstepping fuzzy neural network control design for hybrid maglev transportation system. IEEE Trans Neural Netw Learn Syst 26(2):302–317

Wang H, Cao Y (2015) Predicting power consumption of GPUs with fuzzy wavelet neural networks. Parallel Comput 44:18–36

Woo ZW, Chung HY, Lin JJ (2000) A PID type fuzzy controller with self-tuning scaling factors. Fuzzy Sets Syst 115(2):321–326

Yuan X, Yang Y, Wang H (2012) Improved parallel chaos optimization algorithm. Appl Math Comput 219(8):3590–3599

Yuan X, Zhang T, Xiang Y, Dai X (2015) Parallel chaos optimization algorithm with migration and merging operation. Appl Soft Comput 35:591–604

Zheng JM, Zhao SD, Wei SG (2009) Application of self-tuning fuzzy PID controller for a SRM direct drive volume control hydraulic press. Control Eng Pract 17(12):1398–1404