Australian Journal of Basic and Applied Sciences, 4(12): 6319-6330, 2010
ISSN 1991-8178

Utilizing Global-Best Harmony Search to Train a PID-like ANFIS Controller

[1]O.F. Lutfy, [1]S.B. Mohd Noor, [1]M.H. Marhaban and [2]K.A. Abbas

[1]Electrical and Electronic Engineering Dept., Faculty of Engineering,
[2]Food Technology Dept., Faculty of Food Science & Technology, Universiti Putra Malaysia,
Serdang 43400, Selangor, Malaysia.

Abstract: This paper presents a PID-like adaptive neuro-fuzzy inference system (ANFIS) controller that can be trained by the global-best harmony search (GHS) technique to control nonlinear systems. Instead of the hybrid learning methods that are widely used in the literature to train the ANFIS structure, the GHS technique alone is used to train the ANFIS as a feedback controller, and hence, the necessity for the teaching signal required by other techniques has been eliminated. Moreover, the input and output scaling factors for this controller are also determined by the GHS. To show the effectiveness of this controller and its learning method, two nonlinear plants, including the continuous stirred tank reactor (CSTR), were used to test its performance in terms of generalization ability and reference tracking. In addition, this controller robustness to output disturbances has been also tested and the results clearly indicate the remarkable performance of this controller.

Key words: ANFIS, harmony search, global-best harmony search, CSTR.

INTRODUCTION

One of the most widely used neuro-fuzzy systems is the ANFIS network which was proposed by Jang (1993,1997). The ANFIS network, which is based on the Takagi-Sugeno-Kang (TSK) fuzzy controllers, is a fuzzy inference system (FIS) implemented in the framework of an adaptive fuzzy neural network. It is a very powerful approach for building complex and nonlinear relationship between a set of input and output data (Jang, 1993). The training method plays an important role in determining the final performance of the ANFIS network. In this context, several training methods have been proposed in the literature to train the ANFIS network. For instance, Jang (1993) proposed a hybrid learning method which combines the gradient descent technique for optimizing the antecedent parameters and the least square estimator (LSE) for optimizing the consequent parameters of the ANFIS network. Other researchers (Oliveira, M.V. and R. Schirru, 2009; Shoorehdeli, M.A., 2006; Shoorehdeli, M.A., 2007; Ghomsheh, V.S., 2007) developed a hybrid learning algorithm which composes the particle swarm optimization (PSO) algorithm to optimize the antecedent parameters and the LSE algorithm to optimize the consequent parameters of the ANFIS. Lin (2004) combined the genetic algorithm and the LSE in a hybrid method to optimize the ANFIS parameters. Su and Zhao (2007) proposed a hybrid learning method that combines the expectation maximization (EM) algorithm to estimate the antecedent parameters and the emotional learning, a psychologically motivated algorithm, to learn the consequent parameters of the ANFIS network.

In most of the above mentioned ANFIS training methods, the ANFIS network was trained to act as an identifier with little attention to train it as a feedback controller. In control system design, the necessity for the teaching signal required by the hybrid learning method poses a common problem and serious obstacle encountered when using this method to train the ANFIS network as a controller. In such situations, a more powerful approach to train the ANFIS network as a controller is represented by utilizing the evolutionary algorithms (EAs) which do not require teaching signals in their operation.

One of these EAs is the harmony search (HS) algorithm which is a newly developed approach based on imitating the music improvisation process. In this process musicians improvise their instruments' pitches searching for a perfect state of harmony (Geem, Z.W., 2001). The HS algorithm is simple in concept and easy in implementation.

In this work, the effectiveness of the GHS, which is an improved version of HS proposed by Omran and Mahdavi (Geem, Z.W., 2001) in 2008, is evaluated in optimizing both the antecedent and the consequent

Corresponding Author: O.F. Lutfy, Electrical and Electronic Engineering Dept., Faculty of Engineering,

parameters of the ANFIS network acting as a PID-like controller. In addition, the input and output scaling factors for this controller are also obtained by the GHS method.

The rest of the paper is organized as follows: In Section 2, the PID-like ANFIS structure is explained. An overview of the HS technique, along with the GHS, is given in Section 3. Section 4 explains the implementation of the GHS to train the ANFIS as a feedback controller. The simulation results are presented in Section 5. And finally, Section 6 presents the conclusions.

2. Structure of the PID-like ANFIS Controller:

To describe the structure of the PID-like ANFIS controller, shown in Fig. 1, let $\overline{x}_1$, $\overline{x}_2$, and $\overline{x}_3$

represents the three input variables *e*, *Δe*, and *δe*, respectively, and *y* represents the single output variable *u* of this controller. The ANFIS network utilizes the Sugeno fuzzy models to constitute its structure. Compared to the first-order Sugeno fuzzy model, the zero-order Sugeno fuzzy model requires less number of consequent parameters. Therefore, the zero-order Sugeno fuzzy model is adopted in this work to represent the ANFIS structure in order to reduce the number of consequent parameters to be optimized by the GHS. Five fuzzy linguistic terms are used for each input variable. The output of the i^{th} node in layer *k* will be expressed as $O_{k,i}$, and the function of each layer is discussed in the following:

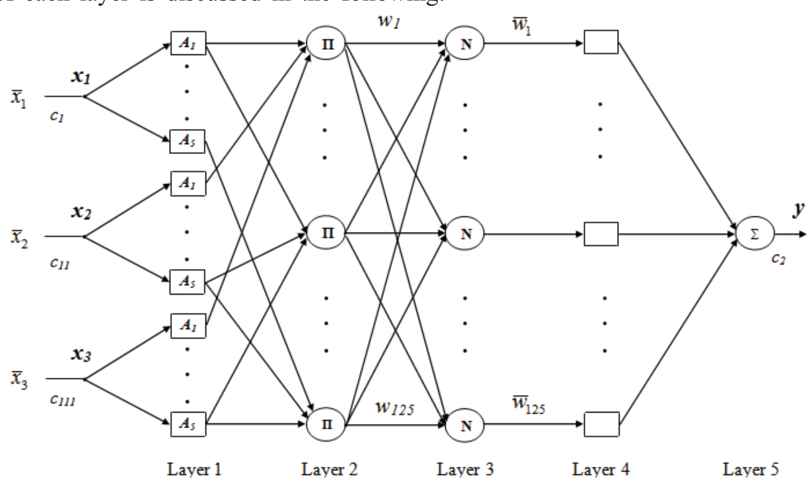


Fig. 1: Structure of the PID-like ANFIS controller.

Layer 1:

The nodes in this layer generate the degree of membership for each of the three input variables to the ANFIS network. The node function is:

$$
\begin{aligned}
O_{1,i} &= \mu_{A_i}(x_1), \qquad & i &= 1,2,\ldots,5 \\
O_{1,i} &= \mu_{A_{i-5}}(x_2), \qquad & i &= 6,7,\ldots,10 \\
O_{1,i} &= \mu_{A_{i-10}}(x_3), \qquad & i &= 11,12,\ldots,15
\end{aligned}
\tag{1}
$$

The membership functions of A_a, a = 1, 2, …, 5, for each input variable are chosen to be bell-shaped activation functions. Only two parameters have been used for each of these functions instead of the widely used generalized bell function, which uses three parameters, in order to reduce the number of parameters to be optimized by the GHS technique. These bell-shaped functions can be represented by:

$$
\mu_{A_a}(x_k) = \exp\left(-1/2\left(\frac{x_k - C_{A_a}^{x_k}}{\sigma_{A_a}^{x_k}}\right)^2 \right) \tag{2}
$$

where x_k, k {1,2,3}, represents the scaled input variables after multiplying the input variables by the input scaling factors (c_1 for the error, c_{11} for its rate of change, and c_{111} for the summation of errors, as indicated in Fig. 1). $C_{A_a}^{x_k}$ and $\sigma_{A_a}^{x_k}$ are the centers and widths of these bell-shaped functions, respectively.

Layer 2:

There are 125 nodes in this layer. The nodes in this layer perform the multiplication operation for fuzzy inferencing. Each node output represents the activation level of a rule. The node output is given by:

$$O_{2,i} = w_i = \mu_{A_{a1}}(x_1) \cdot \mu_{A_{a2}}(x_2) \cdot \mu_{A_{a3}}(x_3) \tag{3}$$

where i=1, 2, …, 125 and $a1,a2,a3$, $\in$ {1,2,3,4,5}. The output of each node in this layer represents the firing strength of the rule.

Layer 3:

This layer has the same number of nodes as layer 2. The i^{th} node output is expressed as:

$$O_{3,i} = \overline{w}_i = \frac{w_i}{\sum_{j=1}^{125} w_j} \qquad i = 1,2,...,125 \tag{4}$$

Layer 4:

There are 125 nodes in this layer. The output of each node is given by:

$$O_{4,i} = \overline{w}_i \cdot k_{0i} \tag{5}$$

where i=1, 2, …, 125, $\overline{w}_i$ is the output of layer 3, and k_{0i} is the i^{th} consequent parameter in the ANFIS structure.

Layer 5:

The single node in this layer computes the ANFIS output as the summation of all incoming signals, according to the following:

$$O_5 = \sum_{i=1}^{125} O_{4,i} \tag{6}$$

And finally, the overall output of the ANFIS network is computed according to the following:

$$y = u = c_2 \cdot O_5 \tag{7}$$

where c_2 represents the output scaling factor of this controller.

3. The Harmony Search Algorithm:

The HS algorithm was developed by Geem *et al.* 2001. The concept of HS is simple and its procedure is easy in implementation. The HS procedure involves the following steps (Omran, M.G.H. and M. Mahdavi, 2008; Geem, Z.W., 2001; Zarei, O., 2009):

Step 1:

Initialize the problem and HS parameters. The optimization problem is defined as minimize (or maximize) *f(x)* such that, $LB_i \leq x_i \leq UB_i$. Where *f(x)* is the objective function, x is a candidate solution consisting of N decision variables, and LB_i and UB_i are the lower and upper bounds for each decision variable, respectively. In addition, the HS parameters are specified in this step, which include; the harmony memory size (*HMS*), or the number of solution vectors in the harmony memory, harmony memory considering rate (*HMCR*), pitch adjusting rate (*PAR*) and the number of improvisations (*NI*).

Step 2:

Initialize the harmony memory, which is a memory location where all the solution vectors are stored. As mentioned before, the HM contains *HMS* solution vectors. Each of these vectors contains *N* decision variables that represent the dimensions of the problem being optimized. In order to initialize the HM, each value of a decision variable in a solution vector (x_i^j) is generated from a uniform distribution in the ranges [LB_i, UB_i], where j = 1, 2, …, *HMS*, and i = 1, 2, …, *N*.

Step 3:

Improvise a new harmony. Generating a new harmony is called “improvisation”. A new harmony vector, $x' = (x'_1, x'_2, ..., x'_N)$, is generated based on three rules, namely: (1) memory consideration, (2) pitch adjustment and (3) random selection. In memory consideration stage, the value of a given decision variable (x'_n), where $1 \le n \le N$, is selected from any of the values for that decision variable in the HM range ($x_n^1 - x_n^{HMS}$). The *HMCR*, which varies between 0 and 1, is the rate of choosing one value from the historical values stored in the HM for a given decision variable, while (*1-HMCR*) is the rate of randomly selecting one value from the possible range of values for that decision variable, as shown in the following:

$$
\begin{aligned}
& f(rand() < HMCR) \\
& x'_i \leftarrow x'_i \in \{x_i^1, x_i^2, ..., x_i^{HMS}\} \\
& else \\
& x'_i \leftarrow x'_i \in X'_i \\
& end
\end{aligned}
\tag{8}
$$

where *rand ( )* is a random number between 0 and 1 and X'_i is a random value from the range assigned for member *i*. Every component obtained by the memory consideration is examined to determine whether it should be pitch-adjusted or not. This operation uses the *PAR* parameter, which is the rate of pitch adjustment as follows:

$$
\begin{aligned}
& f(rand() < PAR) \\
& x'_i = x'_i \pm rand() \times bw \\
& else \\
& x'_i = x'_i \\
& end
\end{aligned}
\tag{9}
$$

where *bw* is an arbitrary distance bandwidth.

Step 4:

Update the harmony memory. If the new harmony vector, $x' = (x'_1, x'_2, ..., x'_N)$, has better fitness function than the worst harmony in the HM, the new harmony is included in the HM and the existing worst harmony is excluded from the HM.

Step 5:

Check the stopping criterion. The HS algorithm is terminated when the stopping criterion (e.g. maximum number of improvisations) has been met. Otherwise, Steps 3 and 4 are repeated.

A major drawback of the standard HS, described above, is that the value of the bandwidth parameter (*bw*) for each decision variable has to be specified by the user in advance. These values are difficult to guess and problem dependent (Geem, Z.W., 2001), especially when the problem being optimized involves many decision variables, as in the case of optimizing the ANFIS parameters considered in this work.

In order to alleviate this difficulty, a new modified version of HS algorithm was proposed by Omran and Mahdavi (Geem, Z.W., 2001) in 2008. Inspired by some concept borrowed from PSO algorithm, this new HS version modifies the pitch-adjustment step such that the new harmony can mimic the best harmony in the HM. Therefore, this HS version was called global-best harmony search (GHS). In particular, the *bw* parameter is totally eliminated in the GHS, as shown in the following:

$$\begin{aligned} & f(rand(\,) < PAR) \\ & x_i^{'} = x_i^{best} \\ & else \\ & x_i^{'} = x_i^{'} \\ & end \end{aligned} \tag{10}$$

where (*best*) is the index of the best harmony in the HM.

4. Implementation of the GHS to Train the ANFIS as a Feedback Controller:

Utilizing the MATLAB software, a specific program has been written in an m-file to implement the GHS technique for training the ANFIS as a feedback controller without using any function in the fuzzy logic Toolbox equipped with MATLAB software. Each solution vector in the GHS used in this work contains 159 cells, or decision variables, representing all the ANFIS parameters. These cells are summarized as: 3 cells for the input scaling factors, 1 cell for the output scaling factor, 30 cells for the antecedent parameters, and 125 cells for the consequent parameters. The universe of discourse (UOD) for each input variable was chosen to be from -6 to 6, keeping in mind that other range can also be used since there are input and output scaling factors. The GHS learning procedure used to optimize the ANFIS parameters in this work is summarized in the following:

Step 1:

Initialize the GHS parameters, which include the following parameters: *HMS*, *HMCR*, *PAR*, and *NI*.

Step 2:

Initialize the HM by randomly generating *HMS* solution vectors within certain bounds. Each of these solution vectors represents the entire antecedent and consequent parameters along with the input and output scaling factors for a single ANFIS controller.

Step 3:

Evaluate the objective function for each solution vector in the HM using the half integral square of errors (0.5ISE) criterion, which has the form:

$$0.5ISE = 0.5\sum_{k=0}^{T_o} e^2(k) \tag{11}$$

where $e(k)$ is the error between the desired output and the plant output at sample k, and T_o is the observation time.

Step 4:

Search for the index of the best and worst harmonies in the HM, and call them best_index and worst_index, respectively.

Step 5:

Improvise a new harmony vector from the HM utilizing the three rules, namely, the memory consideration, the pitch adjustment, and the random selection, that have been described in the previous sections.

Step 6:

Update the HM. If the new harmony vector has better fitness function compared to the worst harmony in the HM, it replaces that worst harmony in the HM.

Step 7:

Stop if the maximum number of improvisations, *NI,* is reached, and the vector which has achieved the latest best fitness value is the optimal, or best, controller generated by the GHS, Otherwise, go to Step 4.

5. Simulation Results:

Depending on the generalization ability of the PID-like ANFIS controller, an input training signal has been applied for plant 1 to optimize the controller parameters. This signal has the following definition:

$$r_{train}(k) = \begin{cases} 0.4 & 0 \le k < 51 \\ 0.6 & 51 \le k < 101 \\ 0.4 & 101 \le k < 151 \\ -0.4 & 151 \le k \le 200 \end{cases} \tag{12}$$

hile the best solution vector (controller) in the GHS is tested by applying an input test signal, which has the following definition:

$$r_{test}(k) = \begin{cases} 0.5 \times \sin(\dfrac{2\pi k}{66}) & 0 \le k < 67 \\ 0.4 & 67 \le k < 134 \\ 0.6 & 134 \le k \le 200 \end{cases} \tag{13}$$

The GHS parameters are set to the following values: *HMS*: 50; *NI*: 2000; *HMCR*: 0.8; *PAR*: 0.5. These settings for the GHS parameters were arrived at after a fair amount of simulation tests to control the following plants:

Plant 1:

This is a nonlinear plant which has the following difference equation (Pham, D.T. and L. Xing, 1997):

$$y(k) = \frac{y(k-1)}{1.5 + y^2(k-1)} - 0.3y(k-2) + 0.5u(k-1) \tag{14}$$

Fig. 2 shows the output response, control action, and best objective function against iterations for this plant. The superb ability of the PID-like ANFIS controller in controlling this nonlinear plant can be clearly seen in Fig. 2 (a), where the plant output is almost identical to the test signal. This performance for the ANFIS controller is due to its control action given in Fig. 2 (b). From Fig. 2 (c), it can be seen that the objective function has reached to its near optimal value within the first few iterations, which indicates the fast convergence achieved by the GHS technique in training the ANFIS controller.

Plant 2:

The PID-like ANFIS controller is used to control the CSTR process which exhibits a highly nonlinear dynamical behavior. The process model consists of the following two nonlinear ordinary differential equations (Nahas, E.P., 1992):

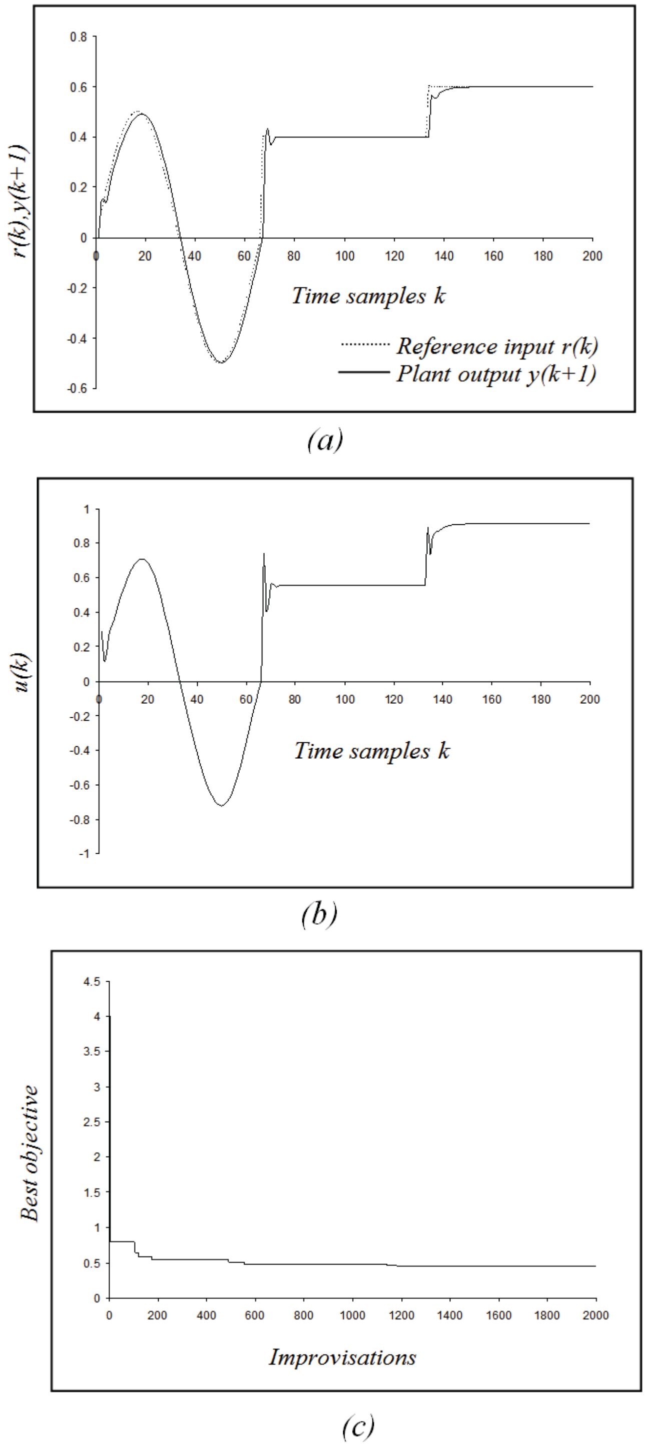


Fig. 2: Plant 1 (a) Output response (b) Control signal (c) Best 0.5ISE.

$$\dot{C}_A = \frac{q}{V}(C_{AF} - C_A) - k_0 C_A \exp\left(\frac{-E}{RT}\right),$$
$$\dot{T} = \frac{q}{V}(T_f - T) + \frac{(-\Delta H) k_0 C_A}{\rho C_\rho} \exp\left(\frac{-E}{RT}\right) + \frac{\rho_c C_{\rho c}}{\rho C_\rho V} q_c \left[1 - \exp\left(\frac{-hA}{q_c \rho_c C_{\rho c}}\right)\right] \times (T_{cf} - T) \quad (15)$$

where C_A is the product concentration of component A, T is the reactor temperature, q is the feed flowrate, and q_c is the coolant flowrate. The objective is to control C_A by manipulating q_c. The remaining model parameters, defined in nominal operating conditions, are given in Table 1. This continuous time model has been numerically solved by using the fourth-order Runge–Kutta method with a simulation step size of 0.1 s. The *NI* parameter in the GHS has been set to 5000 due to the difficulty of this plant. Fig. 3 shows the simulation results of controlling this process. In spite of the nonlinearity of the CSTR process, the PID-like ANFIS controller has performed well in tracking the test signal with zero steady-state error in all the signal parts with some overshoots at the beginning of each step change in the signal, as can be seen from Fig. 3 (a). Fig. 3 (b) shows the variations in the control signal to deal with the step changes in the test signal. Fig. 3 (c) indicates the ability of the GHS in training the ANFIS parameters by minimizing the 0.5ISE through the iterations.

Table 1: Nominal CSTR operating conditions.

Parameter	Description	Nominal value
q	Process flowrate	100 $l\ min^{-1}$
C_{AF}	Inlet feed concentration	1 $mol\ l^{-1}$
T_f	Feed temperature	350 K
T_{cf}	Inlet coolant temperature	350 K
V	Reactor volume	100 l
hA	Heat transfer coefficient	$7\times10^5\ cal\ min^{-1}\ .\ K^{-1}$
k_0	Reaction rate constant	$7.2\times10^{10}\ min^{-1}$
E/R	Activation energy	$9.95\times10^3\ K$
$-\Delta H$	Heat of reaction	$2\times10^5\ cal\ mol^{-1}$
$\rho,\ \rho_c$	Liquid densities	1000 $g\ l^{-1}$
$C_\rho,\ C_{\rho c}$	Specific heats	1 $cal\ g^{-1}\ .\ K^{-1}$
q_c	Coolant flowrate	103.41 $l\ min^{-1}$
T	Reactor temperature	440.2 K
C_A	Product concentration	$8.36\times10^{-2}\ mol\ l^{-1}$

5.1 Generalization Test:

The generalization ability of a controller refers to its ability to deal with reference signals not encountered during the training phase. Although this test was already conducted for the previous two plants, a more difficult generalization test has been also performed on plant 1 to further evaluate the generalization ability of the ANFIS controller. As before, the signal in equation (12) has been used as the training signal in the training phase, while the signal shown in Fig. 4 (a) was the test signal for the trained controller. As can be seen from Fig. 4 (a), the PID-like ANFIS controller has done well in tracking the test signal which differs significantly from the training signal of equation (12). This result indicates the remarkable generalization ability of this controller.

5.2 Robustness Test:

This test is done to establish how robust the PID-like ANFIS controller is to environmental changes that might encounter the control system. This test was achieved on plant 1 by applying bounded external disturbances of 30% of the plant output, for 30 samples in two periods, at the test phase (equation (13)). By examining Fig. 5 (b), it is clear that the control actions of the PID-like ANFIS controller have been successfully adapted to eliminate the effect of the external disturbances, where the convergence to the desired response is achieved with zero steady-state error during the effect of each disturbance and after their disappearance, see Fig. 5 (a). This result strongly gives an indication that this intelligent controller, trained by the GHS technique, has the ability to handle external disturbances.

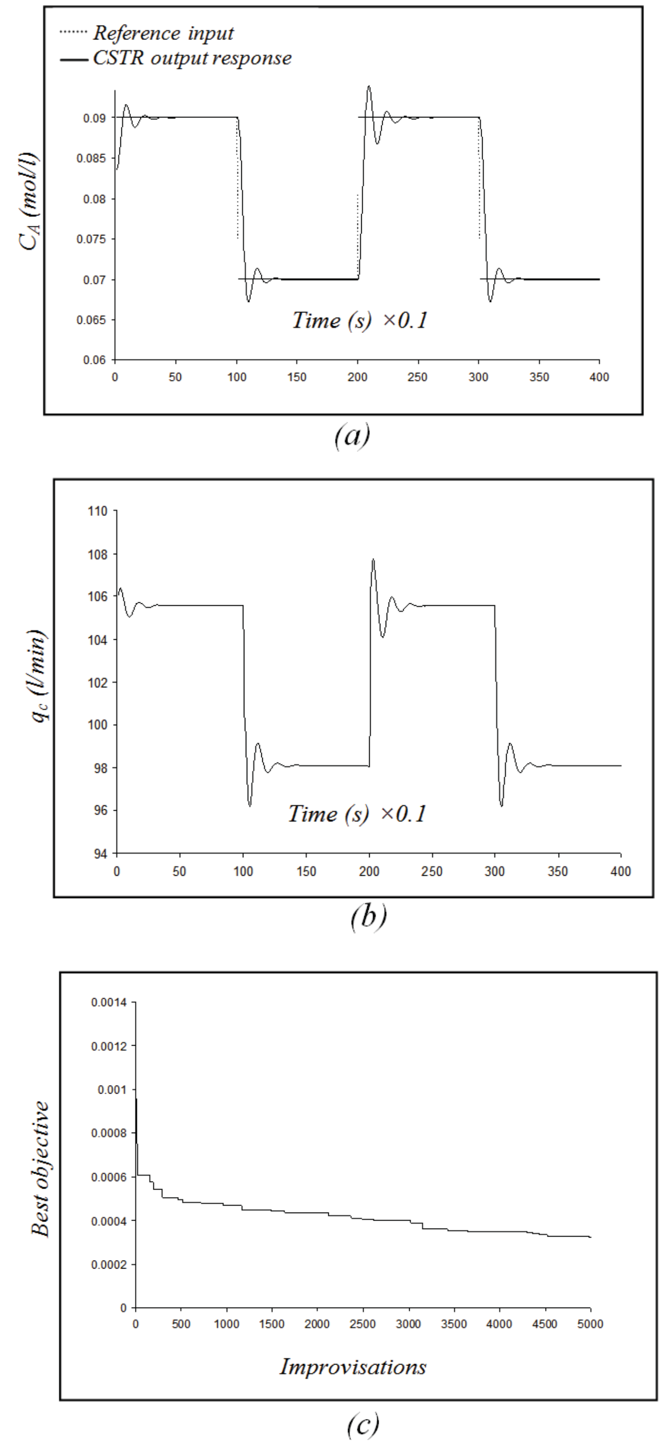
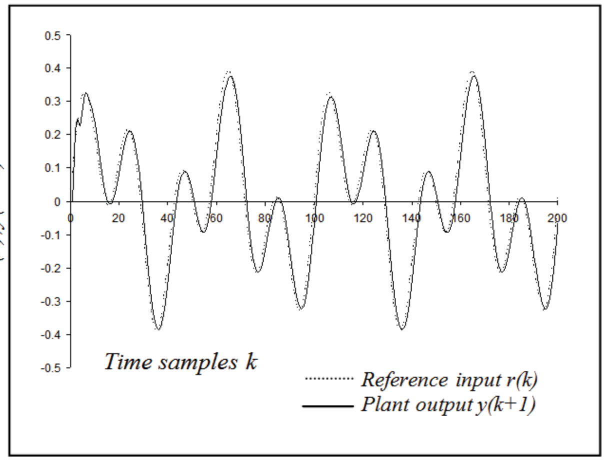
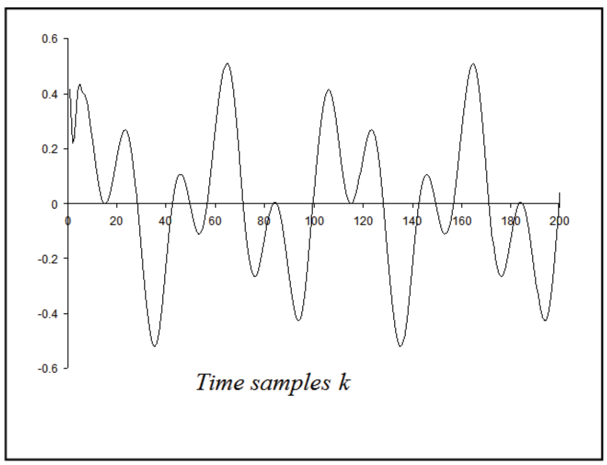


Fig. 3: The CSTR process (a) Output response (b) Control signal (c) Best 0.5ISE.



(a)

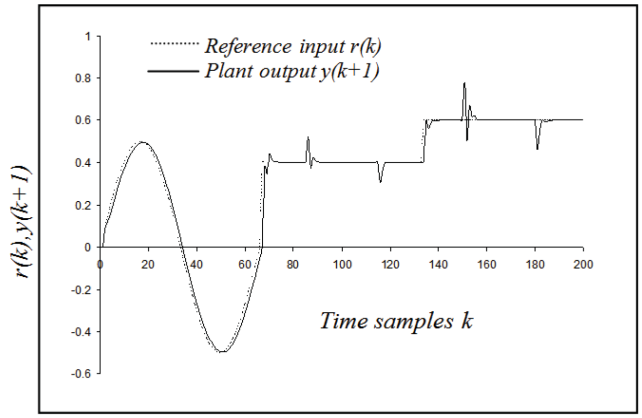


(b)

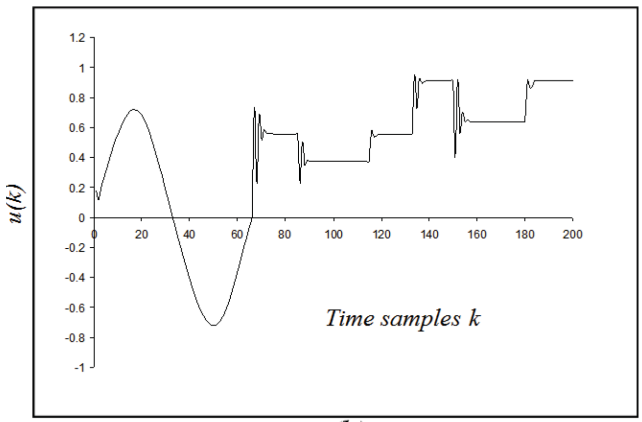
Fig. 4: Plant 1 subjected to a different test signal (a) Output response (b) Control signal.

6. Conclusions:

In this paper, a PID-like ANFIS controller trained by the GHS technique has been proposed to control nonlinear systems. Instead of the hybrid learning methods that are widely used in the literature to train the ANFIS structure, the GHS alone has been used to adjust the ANFIS parameters, along with the input and output scaling factors of this controller based on minimizing the 0.5ISE criterion. In the GHS, the necessity for the teaching signal required by other optimization techniques is eliminated. In order to reduce the number of parameters to be optimized by the GHS, only two parameters have been used for the bell-shaped membership functions in the premise part of each rule of the ANFIS structure. Moreover, the zero-order Sugeno fuzzy model is used in order to reduce the number of the consequent parameters to be optimized by the GHS. The simulation results showed the effectiveness of the GHS technique in training the PID-like ANFIS controller in terms of fast response and tracking accuracy to the desired output with zero steady-state error. The generalization tests have indicated the notable generalization ability of this controller. In addition, the robustness test to output disturbances has clearly shown the remarkable performance of this controller in terms of eliminating the effect of external disturbances during their effects and after their disappearance, where the convergence to the desired response was achieved with zero steady-state error after the adaptation of the control signal.



(a)



(b)

Fig. 5: Plant 1 subjected to 30% external disturbances (a) Output response (b) Control signal.

REFERENCES

Geem, Z.W., J.H. Kim and G.V. Loganathan, 2001. A New Heuristic Optimization Algorithm: Harmony Search. Simulation, 76(2): 60-68.

Ghomsheh, V.S., M.A. Shoorehdeli and M. Teshnehlab, 2007. Training ANFIS Structure with Modified PSO Algorithm. Mediterranean Conference on Control and Automation, Athens-Greece, pp: 1-6.

Jang, J.S.R., 1993. ANFIS: Adaptive-Network-Based Fuzzy Inference System. IEEE Transactions on Systems, Man, and Cybernetics, 23(3): 665-685.

Jang, J.S.R., C.T. Sun and E. Mizutani, 1997. Neuro-Fuzzy and Soft Computing; A Computational Approach to Learning and Machine Intelligence. Prentice-Hall, Inc.

Lin, C.J., 2004. AGA-based Neural Fuzzy System for Temperature Control. Fuzzy Sets and Systems, 143(2): 311-333.

Nahas, E.P., M.A. Henson and D.E. Seborg, 1992. Nonlinear Internal Model Control Strategy for Neural Network Models. Computers Chem. Eng., 16(12): 1039-1057.

Oliveira, M.V. and R. Schirru, 2009. Applying Particle Swarm Optimization Algorithm for Tuning a Neuro-Fuzzy Inference System for Sensor Monitoring. Progress in Nuclear Energy, 51(1): 177-183.

Omran, M.G.H. and M. Mahdavi, 2008. Global-best Harmony Search. Applied Mathematics and Computation, 198(2): 643-656.

Pham, D.T. and L. Xing, 1997. Neural Networks for Identification, Prediction and Control. Springer-Verlag, London Ltd., 3rd Printing.

Shoorehdeli, M.A., M. Teshnehlab and A.K. Sedigh, 2006. A Novel Training Algorithm in ANFIS Structure. American Control Conference, Minnesota, USA, pp: 5059-5064.

Shoorehdeli, M.A., M. Teshnehlab and A.K. Sedigh, 2007. Novel Hybrid Learning Algorithms for Tuning ANFIS Parameters Using Adaptive Weighted PSO. IEEE International Fuzzy Systems Conference, pp: 1-6.

Su, H. and F. Zhao, 2007. A Novel Learning Method for ANFIS Using EM Algorithm and Emotional Learning. International Conference on Computational Intelligence and Security, pp: 23-27.

Zarei, O., M. Fesanghary, B. Farshi, R.J. Saffar and M.R. Razfar, 2009. Optimization of Multi-pass Face-milling via Harmony Search Algorithm. Journal of Materials Processing Technology, 209(5): 2386-2392.