Feedforward Controller for Nonlinear Systems Utilizing a Genetically Trained Fuzzy Neural Network

Omar F. Lutfy Al-Karkhy*
Received on: 15/2/2006
Accepted on: 26/11/2006

Abstract

This paper presents an intelligent controller that acts as a FeedForward Controller (FFC). utilizing the benefits of Fuzzy Logic (FL), Neural Networks (NNs) and Genetic Algorithms (GAs), this controller is built to control nonlinear plants, where the GA is used to train this Fuzzy Neural Controller (FNC) by adjusting of its parameters based on minimizing the Mean Square of Error (MSE) criterion.

These parameters of the FNC include the input and output scaling factors, the centers and widths of the membership functions (MFs) for the input variable and the quantisation levels of the output variable, that are subjected to constraints on their values by the expert. The GA used in this work is a real-coding GA with hybrid selection method and elitism strategy. To show the effectiveness of this FNC several invertable (open-loop stable) nonlinear plants have been selected to be controlled by this FNC through simulation.

Key-words: Genetic Algorithms, Fuzzy Logic, Neural Networks, Feedforward Controller.

مسيطر ذو تغذيه أماميه للأنظمه اللاخطية
بإستخدام شبكة ضبابية عصبية مدرّبة جينيا

الخلاصة

يقدم هذا البحث مسيطر ذكي والذي يعمل كمسيطر ذو تغذيه أماميــه. بإســتغلال فوائــد المنطق المضبب والشبكات العصبية والخوارزميات الجينية, تم بناء هذا المسيطر ذو التغذيــه الأماميه للسيطره على أنظمه لا خطية, حيث تستخدم الخوارزمية الجينية لتــدريب المســيطر الضبابي العصبي من خلال ضبط عدة معاملات فيه إعتمادا على تقليل معيار معدل مربع الخطأ. وهذه المعاملات للمسيطر الضبابي العصبي تشمل عوامل التقييس للإدخال والإخراج, المراكــز والامتدادات لدوال العضوية لمتغير الإدخال ومستويات الكميات لمتغير الإخراج والتي تخضــع لقيود على قيمها من قبل الخبير. الخوارزمية الجينية المستخدمة في هذا العمل هي خوارزمية جينية ذات معاملات ترميز بالقيم الحقيقية وطريقة اختيار مهجنة وحكم النخبة.
ولإظهار فعالية هذا المسيطر الضبابي العصبي تم إختيار عدة أنظمة لاخطيــة قابلــة للعكــس ودارتها المفتوحة مستقرة ليتم السيطرة عليها من قبل هذا المسيطر الضبابي العصبي من خلال التمثيل.

*Control and Systems Eng. Department, UOT., Baghdad-Iraq.

https://doi.org/10.30684/etj.25.3.19
2412-0758/University of Technology-Iraq, Baghdad, Iraq

1. Introduction

In recent years it has been recognized that to realize more flexible control systems it is necessary to incorporate other elements, such as logic, reasoning and heuristics into the more algorithmic techniques provided by conventional control theory, and such systems have come to be known as intelligent control systems [1]. Intelligent control is the discipline in which control algorithms are developed by emulating certain characteristics of intelligent biological systems. It is quickly emerging as a technology that may open avenues for significant advances in many areas [2]. Currently, there are a number of techniques that can be used as a basis for the development of intelligent systems, namely, Expert Systems, Fuzzy Logic (FL), Neural Networks (NNs), Genetic Algorithms (GA), and Artificial Life. These artificial intelligent (AI) techniques should be integrated with modern control theory to develop intelligent control systems [3]. It is known that fuzzy control is able to deal with human knowledge. Therefore, a precise mathematical model of the plant is not required for designing the controller. It is difficult, however, to design the fuzzy controller systematically [4]. In other words, FL systems, which can reason with imprecise information, are good at explaining their decisions but they cannot automatically acquire the rules they use to make those decisions. On the other hand, NNs offer a highly structured architecture, with learning and generalization capabilities. However, the meaning of each weight value of the NN is not understandable to the users hence; analysis of the trained network is difficult. For example, while NNs are good at recognizing patterns, they are not good at explaining how they reach their decisions.

These limitations of each approach, and others, have been a central driving force behind the appearance of a rapidly emerging field of Fuzzy Neural Networks (FNNs), which attempt to obtain the advantages of both FL and NNs techniques while avoiding their individual drawbacks [1][5].

As another approach to intelligent control, GA is a directed random search technique, which can find global optimal solution in complex multi-dimensional search spaces [6]. In this search technique, the best regions, as defined by the fitness evaluation function, of the search space gather increasing numbers of vectors. Hence, all regions of the space continue to receive attention. Moreover, the crossover and mutation operators work to ensure that all regions of the search space continue to be explored.

In addition, the need for the availability of supervised data does not appear in GAs while this is a major issue in supervised learning methods for the NNs like the Backprobagation Algorithms (BPAs). Moreover, there is another problem with the supervised learning algorithms, which is the tendency to get stuck at local optima in weight space. While the GA is more likely to finally converge toward finding the global solution of a given problem [1][7][8].

This emerging field of integrating the merits of FL, NNs, and GAs has been termed Soft Computing (SC), representing a flexible and more powerful approach that takes advantage of the three methodologies [9].

Utilizing this powerful approach, this paper presents a feedforward controller for nonlinear systems depending on training a fuzzy neural network by a real-coded GA.

2. Structure of the Fuzzy Neural Controller (FNC):

The structure used by Pham [6] as a feedback controller is used in this work as a feedforward controller but with some important improvements, where the scaling factors for the input and output variables in Pham [6] were fixed and obtained by trial and error while in the present work the GA is used to find the optimal settings for these variables. Moreover, the center of the zero MF for the input variable is allowed to move between –1 and 1, while it was fixed at 0 in Pham. In addition to these improvements, only one type of membership functions is used in this work (the bell-shaped function) instead of the two different types used by Pham [6] in an attempt to make a new change in the basic structure used by the mentioned reference. Fig.(1) depicts the structure of this FNC.

In the figure, (r) is the reference input to be tracked by the plant and it is applied as the input to the FNC and (u) is the FNC output which is applied to the plant as the control signal. Seven fuzzy linguistic terms are used for the input variable (r) and these are; NB: negative big, NM: negative medium, NS: negative small, Z: zero, PS: positive small, PM: positive medium and PB: positive big.

The six layers of this FNC shown in Fig.(1) will be described here, for convenience the following notation will be used to describe the function of each layer:

I_i^L : the net input value to the i**th** node in layer L,

O_i^L : the output value of the i**th** node in layer L.

The function of each layer can be summarized as follows:

Layer One (Input Layer):

There is only one neuron in this layer whose action is to distribute the input signal (r) to the nodes in the next layer after it has been scaled into a predetermined range (Universe Of Discourse (U.O.D)) by being multiplied by a positive factor (c1) representing the input scaling factor of the FNC which is found by the GA.

Layer Two:

The fuzzy set of the input variable (r) in this layer consists of n linguistic terms (in this work n=7 as pointed out earlier). The input of each node in this layer can be expressed as:

$$I_i^2 = c_1 r - b_i \quad for\ i = 1,2,......,\ n \qquad (1)$$

where c_1 is the input scaling factor representing the weights of the links between the input layer and layer two, the b_i's are the biases to the seven-nodes fuzzy set in layer two and these biases determine the positions of the “centers” of the input membership functions (MFs). The activation functions of all nodes in layer two are linear and thus their outputs O_i^2 are:

$$O_i^2 = I_i^2 \qquad for\ i = 1,2,..,n \tag{2}$$

Layer Three:

It has the same number of nodes (n) as layer two. The nodes in the two layers are linked one-to-one and the variable weights of the connections between them (t_i) define the "widths" of the input MFs for the seven-nodes fuzzy set. All nodes in this layer have a bell-shaped activation functions and their outputs O_i^3 are expressed as:

$$O_i^3 = \exp(-1/2.[\frac{O_i^2}{t_i}]^2\ ) \tag{3}$$

where i=1,2,…,n. Layer three together with layer two can be considered as the antecedent part of this FNC.

Layer Four (Rule Layer):

This layer and its links with layer three implement the Max-Product operator for fuzzy inferencing. The connection links between layers three and four have modifiable weights (w_{ji}) and carry out the product operation, while the nodes in layer four (rule nodes) perform the Max. operation. The components O_j^4 of layer four are then expressed by:

$$o_j^4 = I_j^4 = \underset{i=1}{\overset{n}{Max}}\ (w_{ji}\, o_i^3) \tag{4}$$

Where j=1,2,….,S and (S) is the number of nodes in this layer (and it is equal to eleven here). O_j^4 is the membership value of element j of the output fuzzy set.

Layer Five (Consequent Layer):

It has only two nodes, which perform the center of gravity defuzzification method. The total inputs I_1^5 and I_2^5 to both nodes are given by:

$$I_j^5 = \sum_{i=1}^{S} y_{ji} O_i^4 \qquad j = 1,2 \tag{5}$$

For the first node y_{1i} defines the value of the i^th output quantisation level and is obtained through training. For the second node all y_{2i}'s (i=1 to S) are fixed at unity. The outputs of these units are:

$$O_1^5 = I_1^5 \tag{6}$$

$$O_2^5 = 1/I_2^5 \tag{7}$$

Layer Six (Action Layer):

It has only one node, which receives the outputs of layer five (O_1^5 and O_2^5) via links with unit weights and multiplies them together to yield its net input I_6:

$$I_6 = O_1^5 O_2^5 \tag{8}$$

And finally, to obtain the output control action (u) of the FNC, I_6 is multiplied with a factor (c_2) representing the output scaling factor of this FNC:

$$u = O_6 = c_2 I_6 \tag{9}$$

3. Chromosome Representation:

The chromosome representation of each GA individual (controller) for this FNC can be summarized as: one input scaling factor (c_1), 7 bias values for layer two, 7 weight values (t_i) from layer two to layer three, 7×11 weight values (w_{ji}) from layer three to layer four, 11 weight values (y_{1i}) from layer four to layer five and one output scaling factor (c_2). Therefore, 104 real-valued genes are required to represent each chromosome in the GA for this FNC.

4. The Constraints on Each Chromosome Gene:

The parameters of the FNC (which is represented by each GA chromosome gene) are subjected to constraints on their values within the genetic learning. The input and output scaling factors (c_1 and c_2) are allowed to change from 0.1 to 6. The centers of the input variable MFs are distributed over the U.O.D from –6 to 6 in a way that prevents severe overlapping between the adjacent MFs. The width for each MF is allowed to vary between 0.4 and 1 (this range was found to be suitable to give a reasonable width for each MF in the "-6 to 6" U.O.D.). And finally the output quantisation levels are scattered along the output U.O.D form –6 to 6.

5. The Real-Coded GA Operators:

The operators of the real-coded GA used in this work can be summarized as follows:

5.1 Hybrid Selection:

This selection method, which was first introduced by Al-Said [10], is a combination of Roulette Wheel and deterministic selection, forming a robust strategy inspired from the simplex selection method. In this method, it is to accept in the new population only those strings that have better fitness values than the worst individual in the old population. This method is expected to ensure good guidance in the complex and nonlinear search space, due to its ability to improve the strings in a given generation from those in the previous one.

5.2 Elitism:

In this operation, the best n parents (in this work the best two) from the current generation are copied directly into the next generation as they are. This approach prevents the best fitness value in a given generation from becoming worse than that in the previous generation [11].

5.3 Crossover:

In the real-coding crossover operator, which is similar to that of binary coding, a pair of mating chromosomes exchanges information by exchanging a subset of their components, where an integer position k is selected uniformly at random along the chromosome length. Then two new chromosomes are created by swapping all the genes between positions k+1 and L, where L is the chromosome length [11].

For example, the pair of chromosomes a and b as

a=[2.3,5,1.9,7,4,3.2,8.5]

b=[7.6,9,3,2.9,6,5,1]

are crossed over at the third digit position to yield:

a'=[2.3,5,3,2.9,6,5,1]

b'=[7.6,9,1.9,7,4,3.2,8.5]

5.4 Mutation:

This operation causes random changes in the components of the chromosomes in the new population. In binary-coding GA, this operator randomly flips some of the bits in chromosomes. For example, the chromosome 00010 might be mutated in its second position to yield 01010. In real-coded GA this operator is adapted by simply replacing the mutated 'gene' with another random number chosen in the same range assigned for that 'gene'. As an example, the chromosome c=[5,8.1,1.6,4,2,9,3] is mutated at the fifth 'gene' to yield: c'=[5,8.1,1.6,4,3.3,9,3].

6. The Proposed Genetic Learning for the FNC:

The following genetic procedure is used for training the FNC:

Step1: Initialize the genetic operators: the crossover probability Pc, the mutation probability Pm, the population size, and the maximum number of generations.

Step2: Generate randomly the initial population within certain bounds, in which each individual represents the entire modifiable weight connections of a single FNC.

Step3: Evaluate the objective function for each individual in the population using the Mean Square of Error (MSE) criterion having the form:

$$MSE = \sum_{k=1}^{N_p} \frac{[y_p(k) - r(k)]^2}{N_p} \qquad (10)$$

where, $y_p(k)$ is the plant's output at sample k, r(k) is the reference signal at sample k and N_p is the number of the training patterns. Then, for each individual, calculate the fitness function using the Darwinian fitness equation of the form [10]:

$$fitness = \frac{1}{e + objective\ \ function} \qquad (11)$$

where e is a small constant chosen to avoid division by zero.

Step4: Put in descending order all the chromosomes in the current population (i.e. the first one is the fittest). Then apply "Elitism" strategy described in section 5.2.

Step5: Select individuals by using the hybrid selection method [10], and then apply the real-coded genetic operators of crossover and mutation described previously.

Step6: Stop if the maximum number of generations is reached, otherwise increment the generations counter by one and go to step3.

7. Simulation Results:

To examine the performance of the FNC described in section (2), four simulation examples have been adopted (based on using a discrete model of the plant) to be controlled by the FNC acting as a feedforward controller as shown in Fig. (2).

In this scheme the reference signal is applied as an input to the FNC and the output of the FNC is used as an actuating signal to drive the plant.

The difference between the plant output and the reference input is the error signal, which is used to adjust the parameters of the FNC during the genetic learning.

The weights of the FNC in this scheme represent the inverse dynamics of the plant. Therefore, this scheme could be used to control open loop stable plants only, because the FNC stores the inverse dynamics of the controlled plant and hence, if the plant was unstable, the FNC would not be able to find the inverse dynamics of the plant when diverges to an infinite steady state value.

The U.O.D for both the input MFs and the output quantisation levels are selected to be from –6 to 6 (another range could also be used since the input and output scaling factors can be modified genetically).

To test the generalization ability of this FNC the following signal is used as the training signal:

$$r_{train}(k) = 0.1 * \sin(\frac{2p\,k}{100}) \qquad 0 \le k < 200$$

while the best individual (FNC) in the GA is tested using the following two test signals:

$$r_{test}1(k) = 0.1 * \sin(\frac{2pk}{50}) \qquad 0 \le k < 200$$

$$r_{test}2(k) = \begin{cases} 0.1 & 0 \le k \le 100 \\ -0.1 & 100 < k < 200 \end{cases}$$

The real-coded GA is set to the following parameters:
Population size: 30
Maximum number of generations: 1000
Pc (crossover probability): 0.8
Pm (mutation probability): 0.05

As pointed out earlier, all the plants used in this simulation are open-loop stable and invertible plants [12].

Plant 1:

$$y(k) = \frac{(y(k-1) * y(k-2)) * (y(k-1) + 2.5)}{1 + y^2(k-1) + y^2(k-2)} + u(k-1)$$

Plant 2:

$$y(k) = (0.8y(k-1) + (u(k-2) - 0.8))$$
$$* u(k-2) * (u(k-2) + 0.5) + u(k-1)$$

Plant 3:

$$y(k) = \frac{(5 * y(k-1) * y(k-2))}{1 + y^2(k-1) + y^2(k-2) + y^2(k-3)}$$

$$+ u(k-1) + 1.1 * u(k-2)$$

Figures (3), (4), (5), (6), (7) and (8) show the output response of the controlled plant, FNC control action, best MSE against the generations, and the learned MFs of the input variable for the two test signals for each plant respectively.

And finally, to compare the performance of this FFC with the Feedback controller used by Pham, the same plant used by Pham is used here to be controlled by the FFC:

Plant 4 [6]:

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

Fig. (9) shows the output response to a step input (as used by Pham), FNC control action, best MSE against the generations, and the learned MFs of the input variable.

From this figure, we can see the fast response and the zero steady-state error with some overshoot (which did not appear in Pham) due to the difference in the control structure used by Pham (Feedback Control).

8. Conclusions

In this paper, a feedforward FNC that can be trained by the GA is introduced. A real-coding operators GA has been utilized to adjust the parameters of this FNC based on minimizing the MSE criterion. The difficult problem of finding the optimal values of the input and output scaling factors for the FNC in [6] has been solved by making the GA find these optimal values. Moreover, the center of the Zero MF has been allowed to move between – 1 and 1 here, instead of being fixed at zero by Pham. In addition, a new change has been made to the basic structure of the FNC used by Pham by using only one type of MFs (bell-shaped function) instead of the two different types used by Pham. The simulation results showed the effectiveness of the proposed controller in controlling several plants to track the two desired reference signals. And the performance comparison made with the feedback controller used by Pham has showed the fast and the zero steady-state error of the response despite some overshoots in the transient response.

References:

1. Linkens D.A. and Nyongesa H.O.; "Learning Systems in Intelligent Control: an Appraisal of Fuzzy, Neural and Genetic Algorithm Control Applications"; IEE proceeding on Control Theory Applications, Vol.143, No.4, pp.367, July 1996.
2. Passino K.M.; "Intelligent Control for Autonomous Systems"; IEEE spectrum methods of control, integrated digital communications and solar wind, Vol.32, No.6, pp.55, June 1995.
3. Omatu S., Khalid M. and Yusof R.;"Neuro-Control and its Applications"; Springer-Verlag London Ltd, Second Printing, 1996.
4. Kiguchi K. and Fukuda T.; "Intelligent Position/Force Controller for Industrial Robot Manipulators-Application of Fuzzy Neural Networks"; IEEE Transactions on Industrial Electronics, Vol.44, No.6, pp.753, December 1997.
5. Ruan D.;"Intelligent Hybrid Systems, Fuzzy Logic, Neural Networks, and Genetic Algorithms"; Kluwer Academic Publishers, 1997.
6. Pham D.T and Xing L.;"Neural Networks for Identification, Prediction and Control"; Springer-Verlag, London Ltd.; third printing 1997.
7. Rzempoluck E.J.; "Neural Network Data Analysis Using Simulnet" Springer-Verlag New York, Inc.; 1998.
8. Morles E.C. and Mort N.; "Neural Network-Based Adaptive Control Design"; Journal of Systems Engineering; Department of Automatic Control and Systems Engineering; University of Shefield; P.O. Box 600; Mappin Street; Shefield Sl 4DV; UK; 1993.
9. Alcalā R., Casillas J., Cordōn O., Herrera F. and Zwir S. "Techniques for Learning and Tuning Fuzzy Rule-Based Systems for Linguistic Modelling and their Application"; Knowledge Engineering Systems Techniques and Applications; Vol. 3; Leondes (Ed.); Academic Press 2000.
10. Al-Said I.A.M.; "Genetic Algorithms Based Intelligent Control"; Ph.D. Thesis; University of Technology, January 2000.
11. Mitchell M.; "An Introduction to Genetic Algorithms" First MIT Press Paperback Edition, Cambridge 1998.
12. Abeer F. Shimal; "CMAC as an Adaptive Controller of Nonlinear Systems"; M.Sc. Thesis; University of Technology, May 2001.

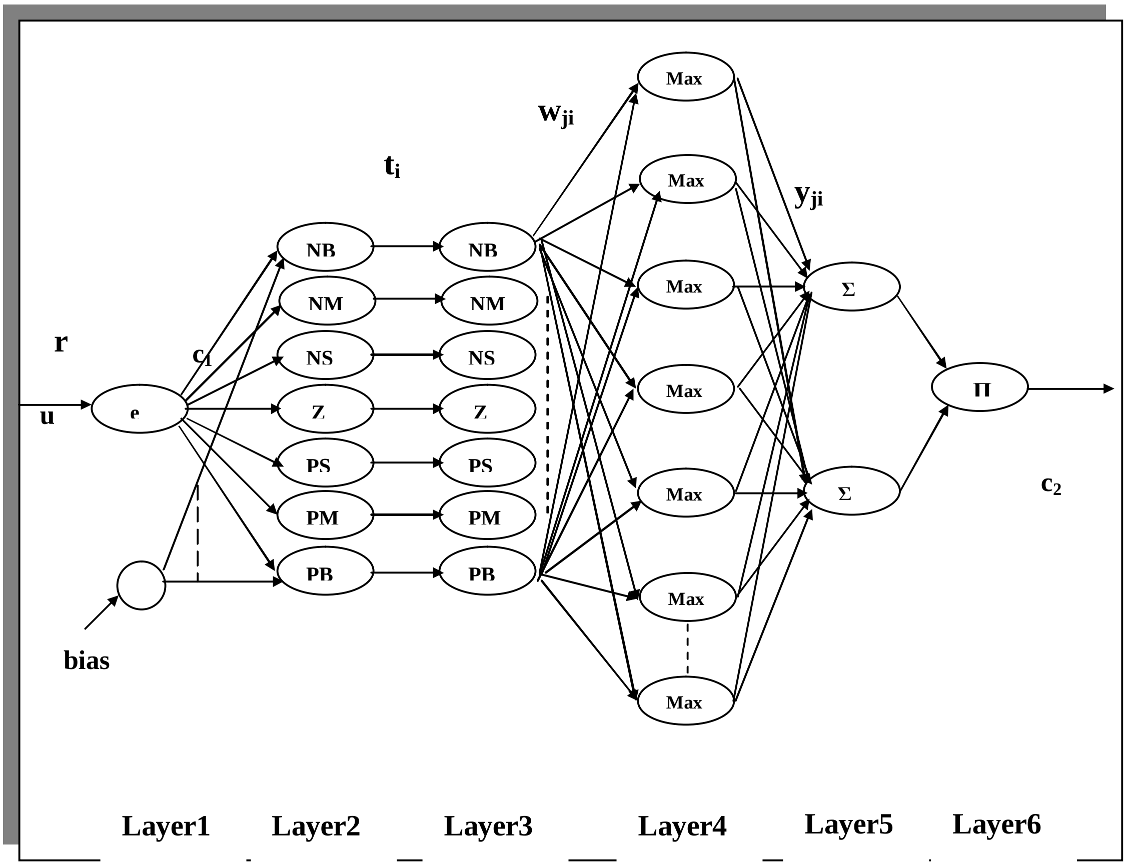

Fig. (1) Structure of the FNC.

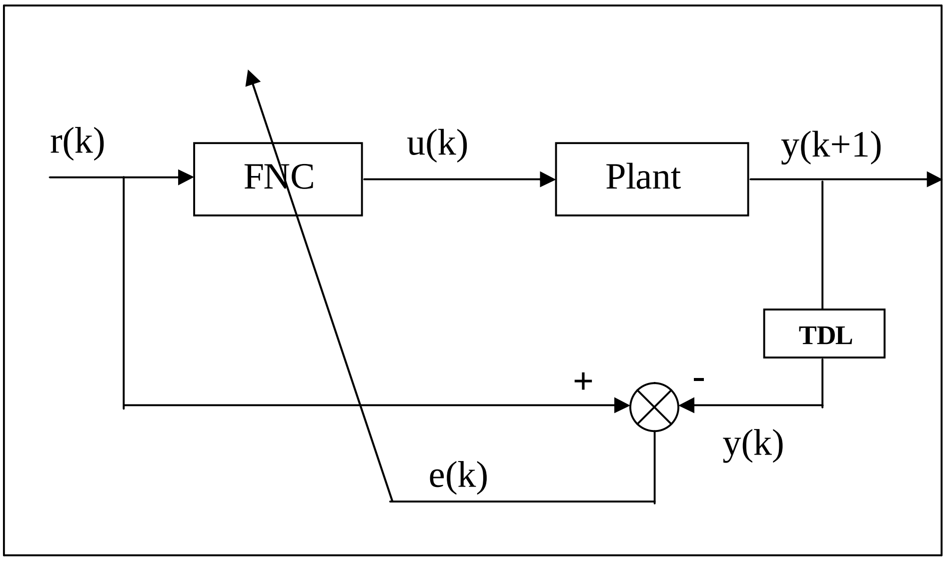

Fig. (2) The FNC as a feedforward controller.

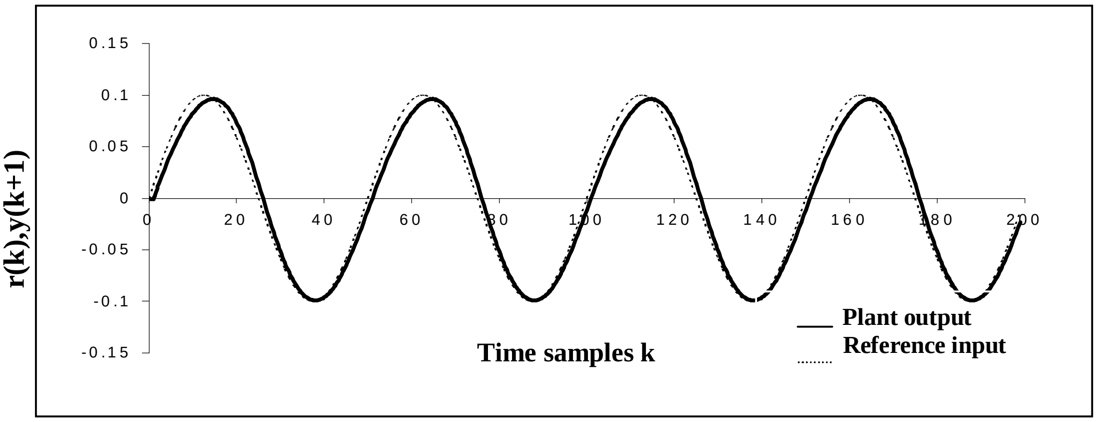

Fig. (3) Plant 1 (a) Output response to sine input (b) Control signal (c) Best MSE (d) Learned MFs of the input variable.

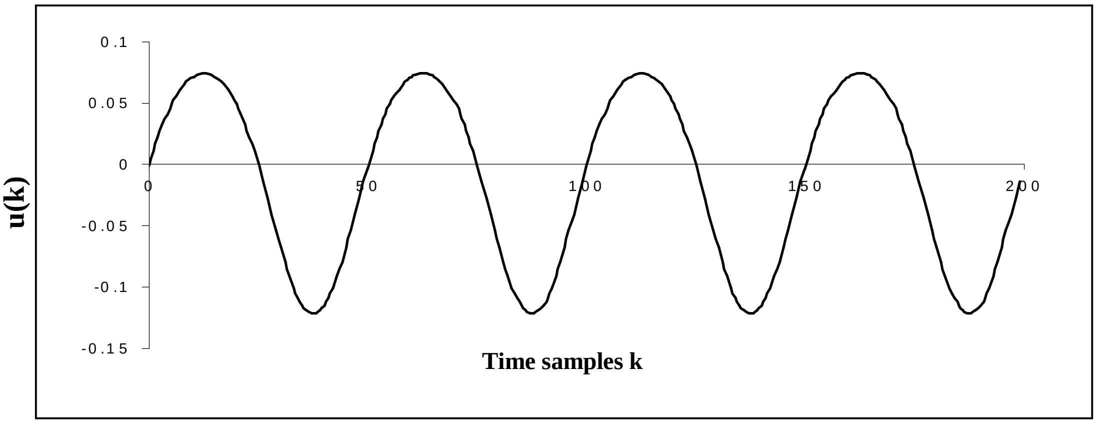

Fig. (3 b) Control signal.

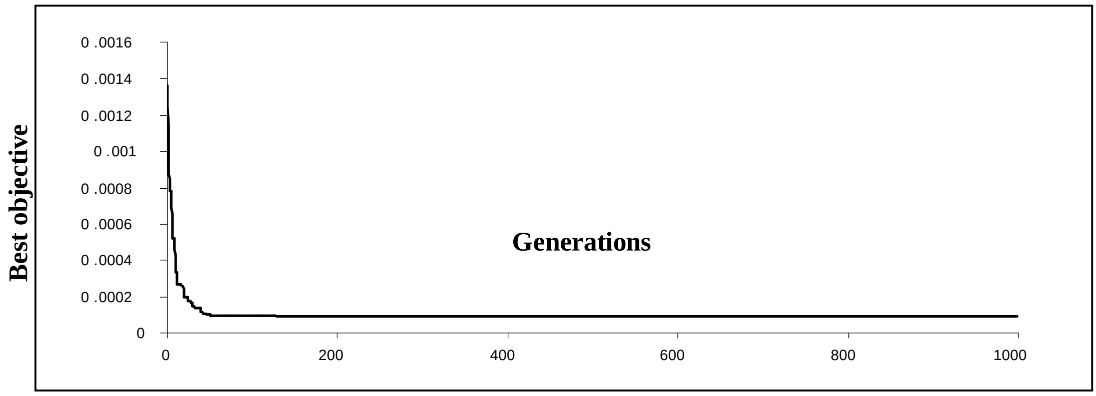

Fig. (3 c) Best MSE.

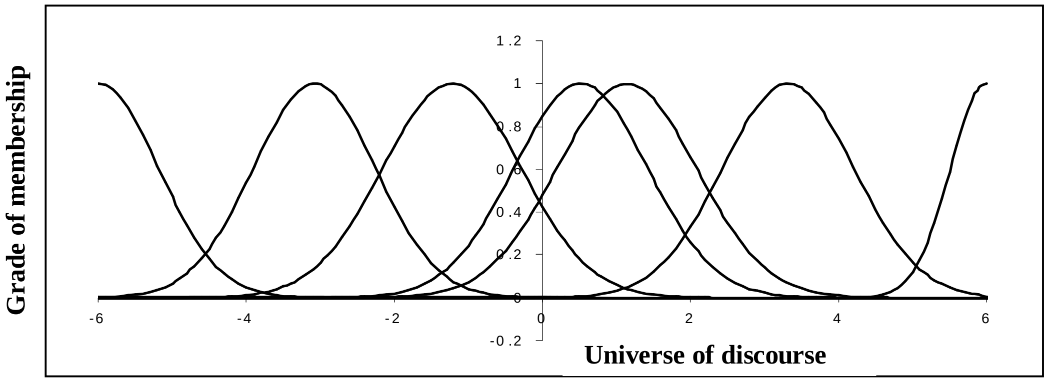

Fig. (3 d) Learned MFs of the input variable.

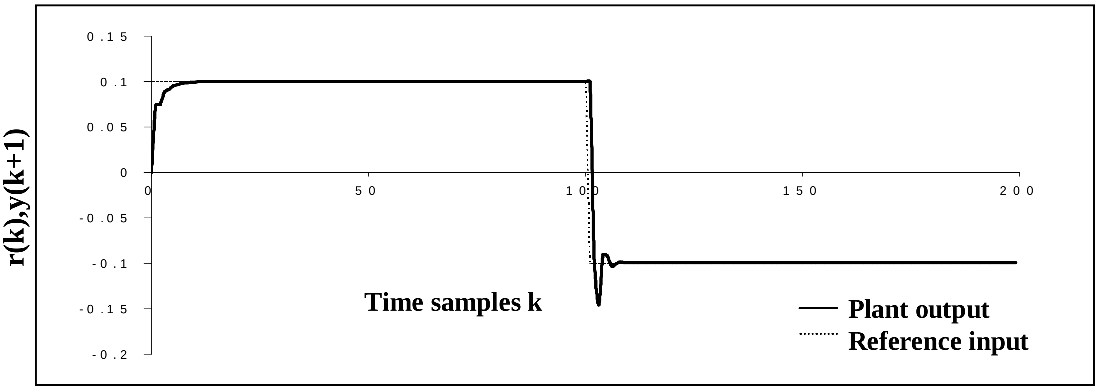

Fig. (4) Plant 1 (a) Output response to step input (b) Control signal (c) Best MSE (d) Learned MFs of the input variable.

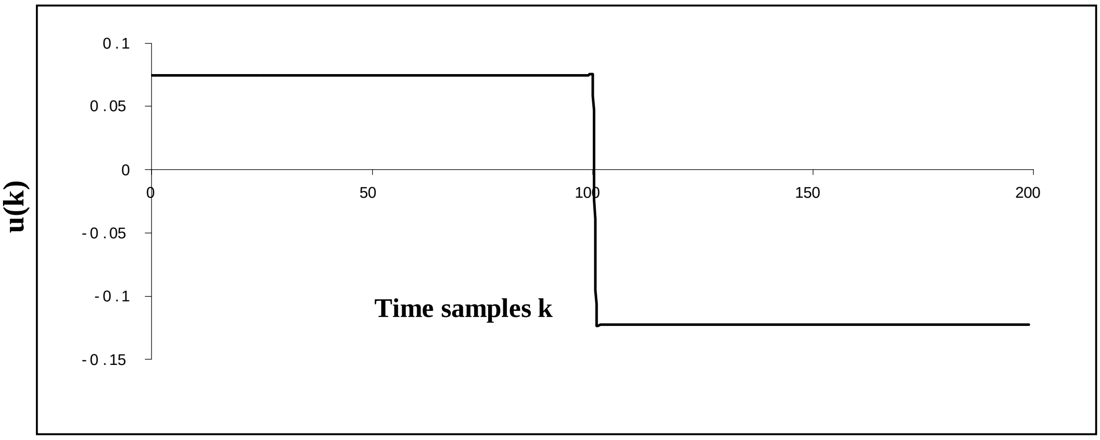

Fig. (4 b) Control signal.

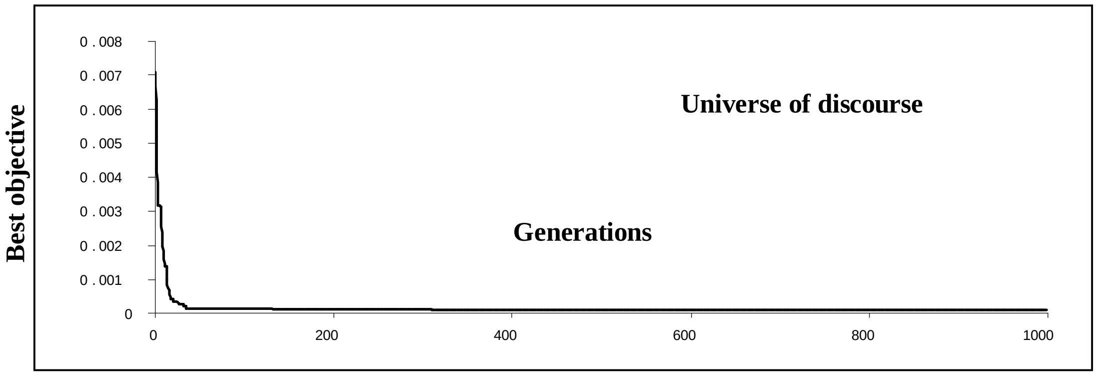

Fig. (4 c) Best MSE.

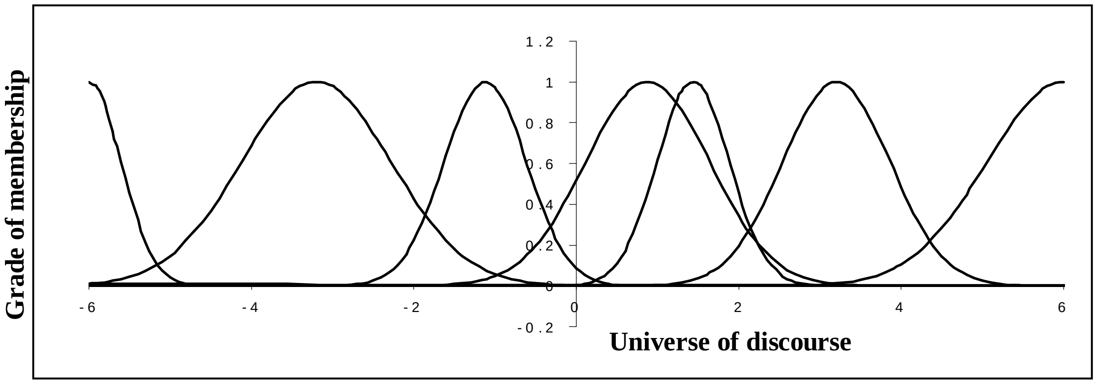

Fig. (4 d) Learned MFs of the input variable.

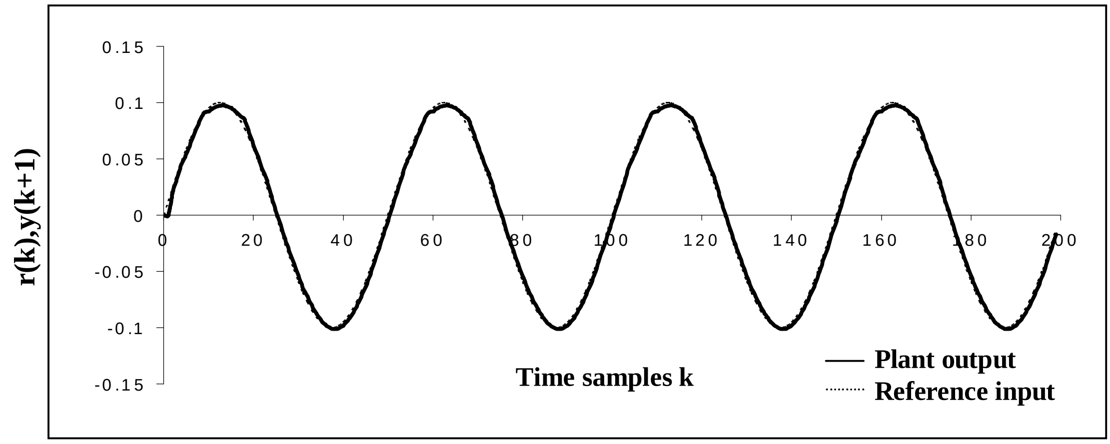

Fig. (5) Plant 2 (a) Output response to sine input (b) Control signal (c) Best MSE (d) Learned MFs of the input variable.

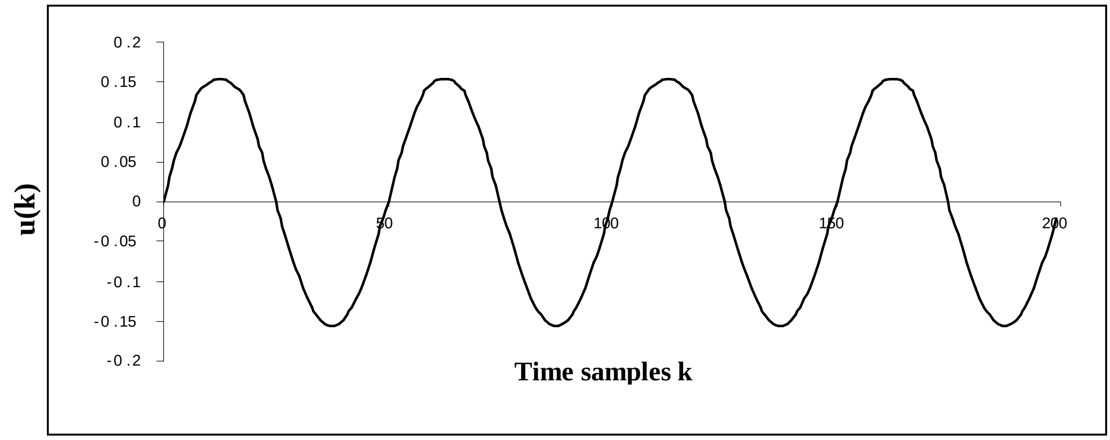

Fig. (5 b) Control signal.

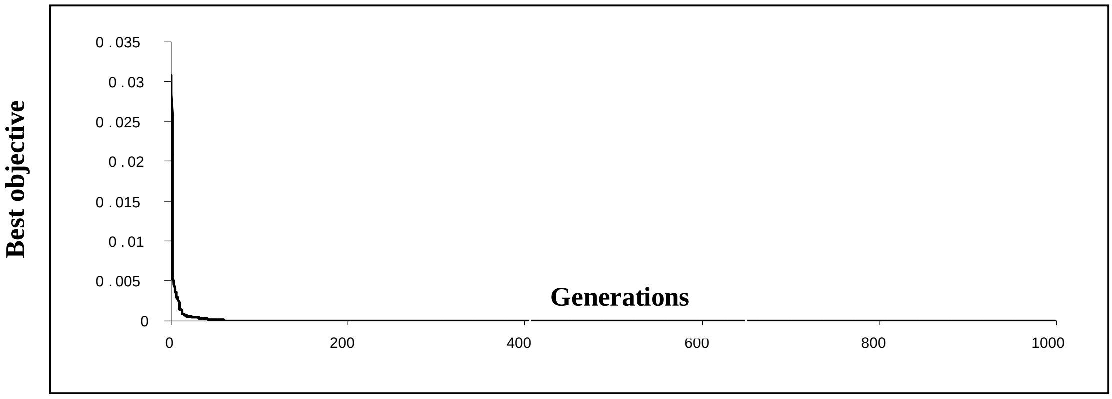

Fig. (5 c) Best MSE.

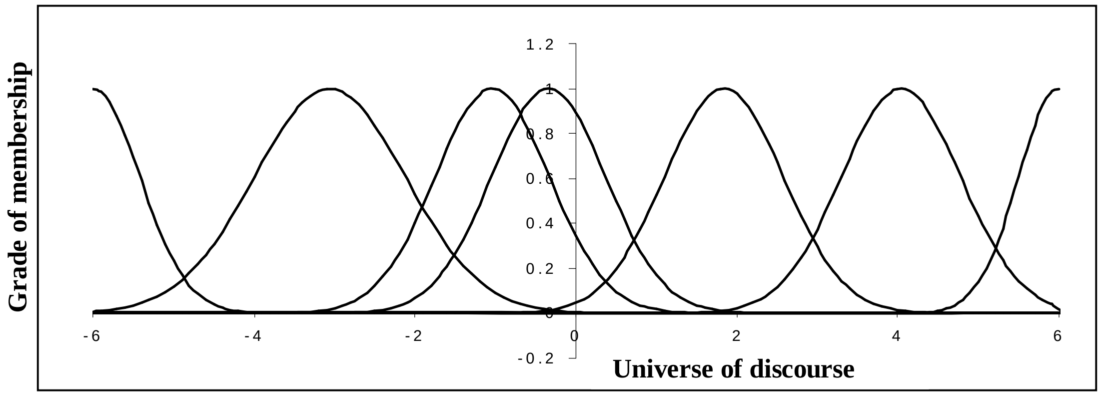

Fig. (5 d) Learned MFs of the input variable.

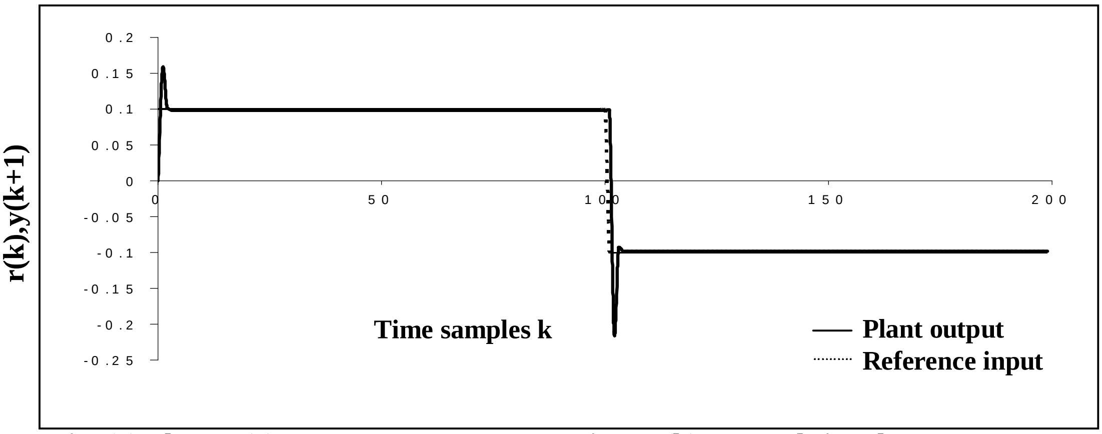

Fig. (6) Plant 2 (a) Output response to step input (b) Control signal (c) Best MSE (d) Learned MFs of the input variable.

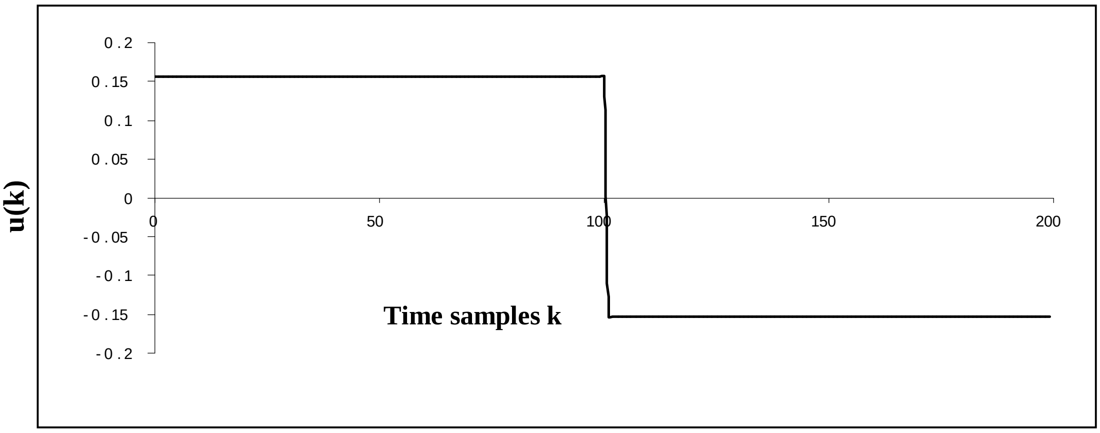

Fig. (6 b) Control signal.

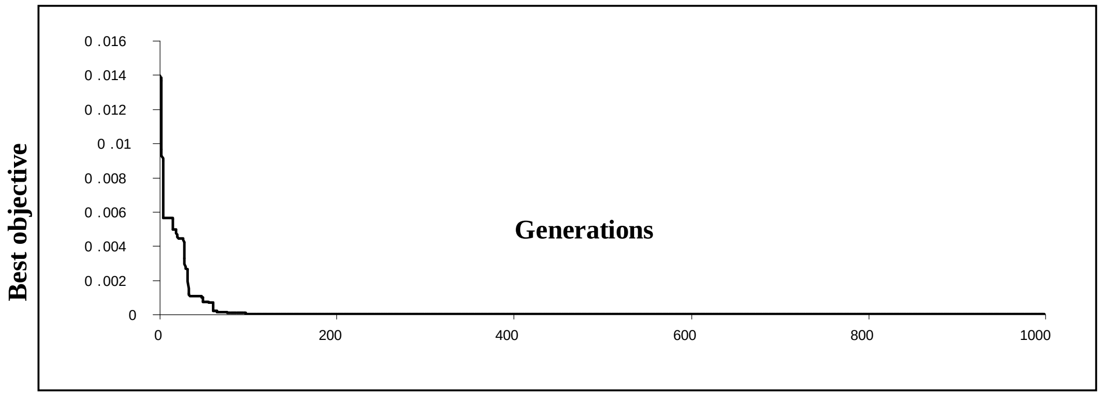

Fig. (6 c) Best MSE.

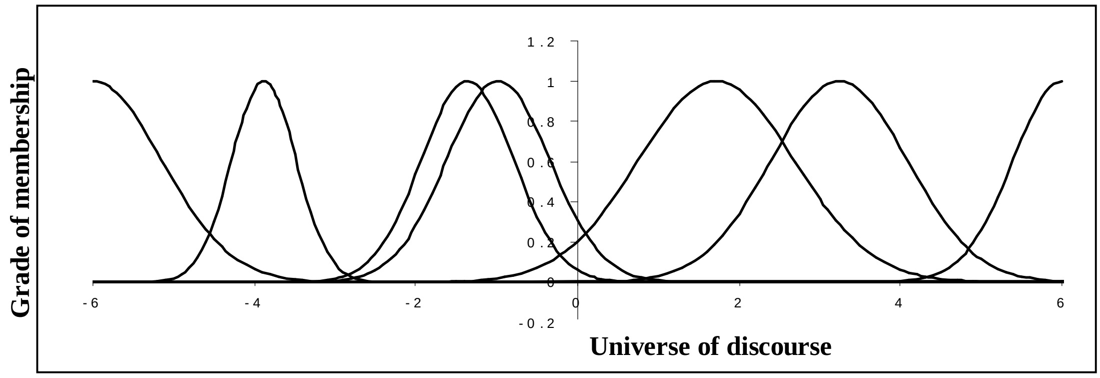

Fig. (6 d) Learned MFs of the input variable.

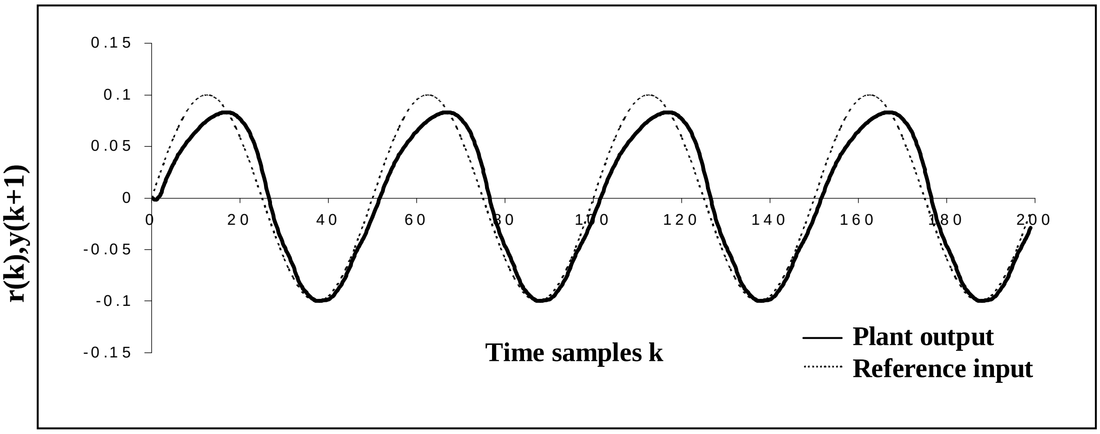

Fig. (7) Plant 3 (a) Output response to sine input (b) Control signal (c) Best MSE (d) Learned MFs of the input variable.

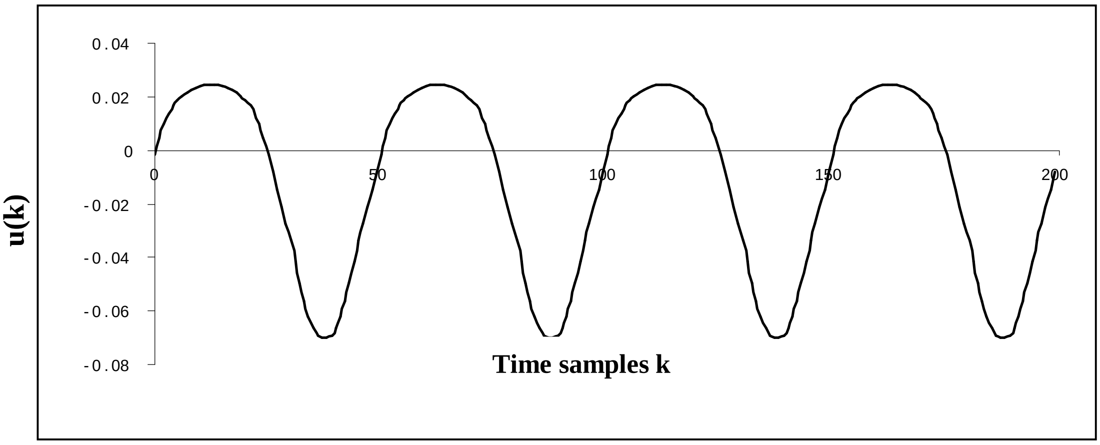

Fig. (7 b) Control signal.

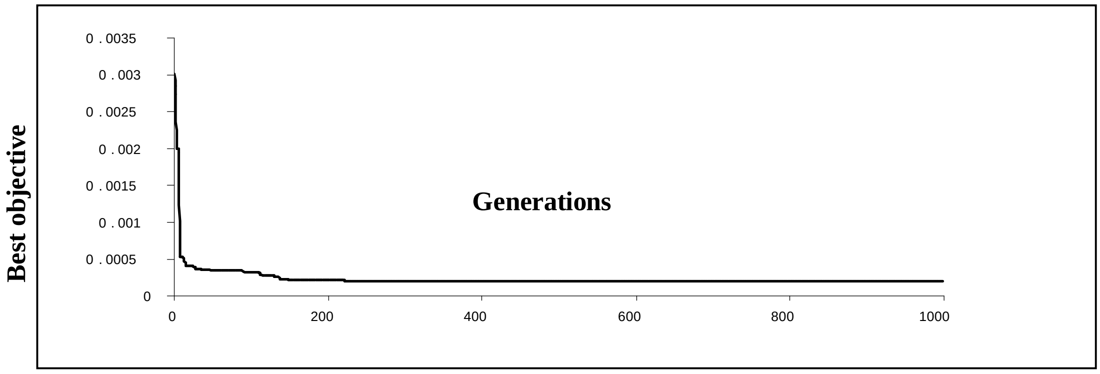

Fig. (7 c) Best MSE.

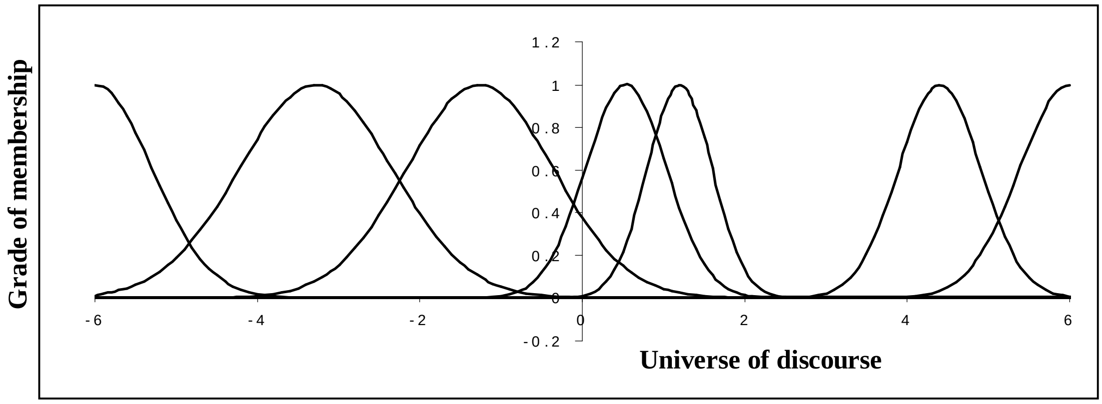

Fig. (7 d) Learned MFs of the input variable.

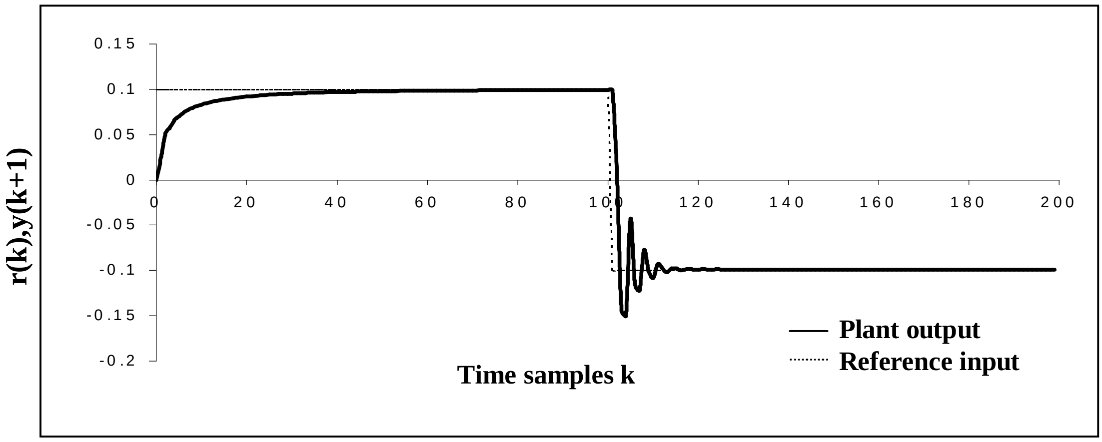

Fig. (8) Plant 3 (a) Output response to step input (b) Control signal (c) Best MSE (d) Learned MFs of the input variable.

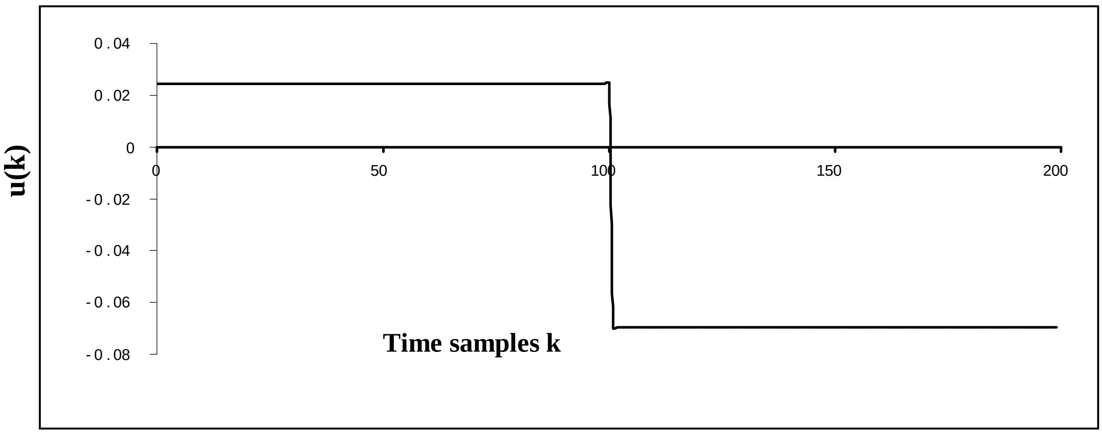

Fig. (8 b) Control signal.

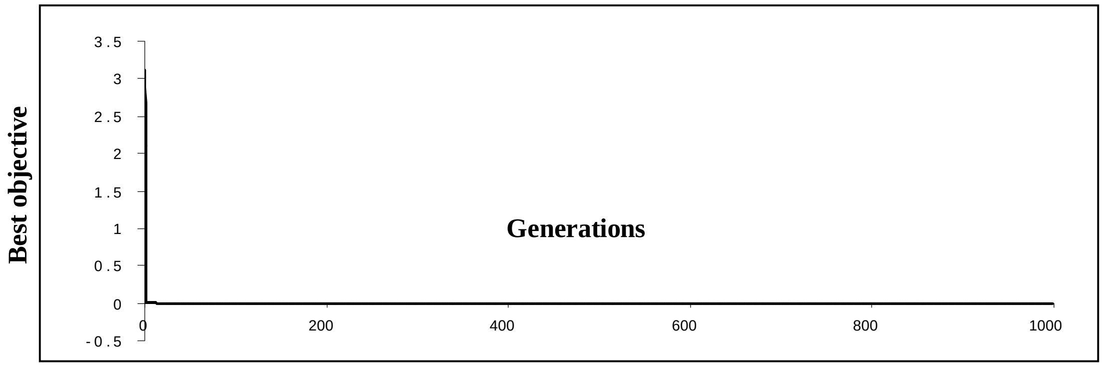

Fig. (8 c) Best MSE.

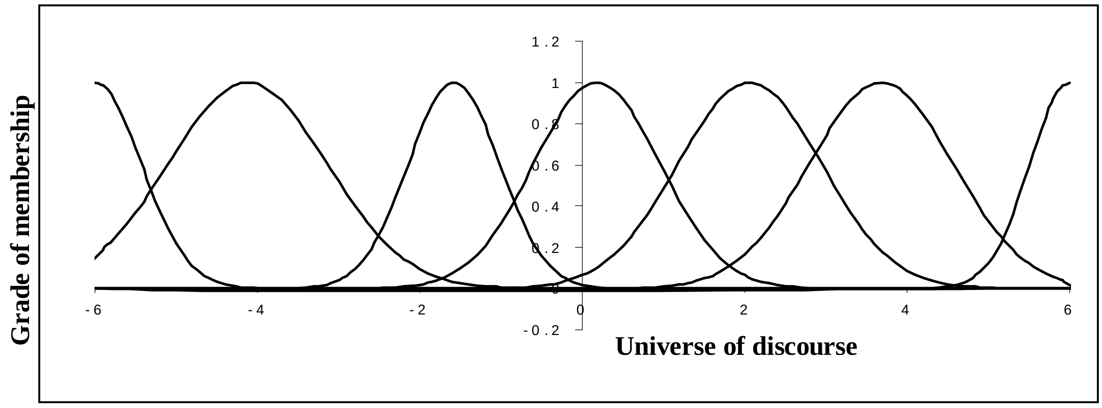

Fig. (8 d) Learned MFs of the input variable.

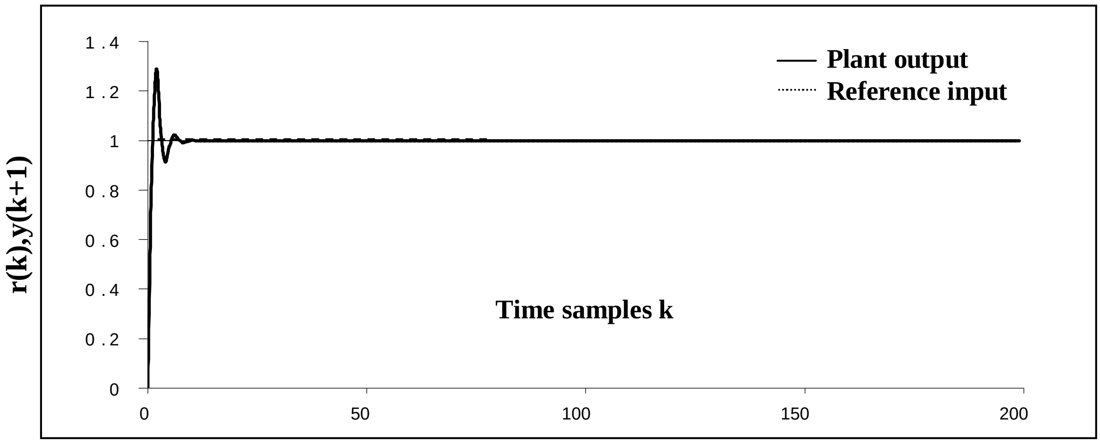

Fig. (9) Plant 4 (a) Output response to step input (b) Control signal (c) Best MSE (d) Learned MFs of the input variable.

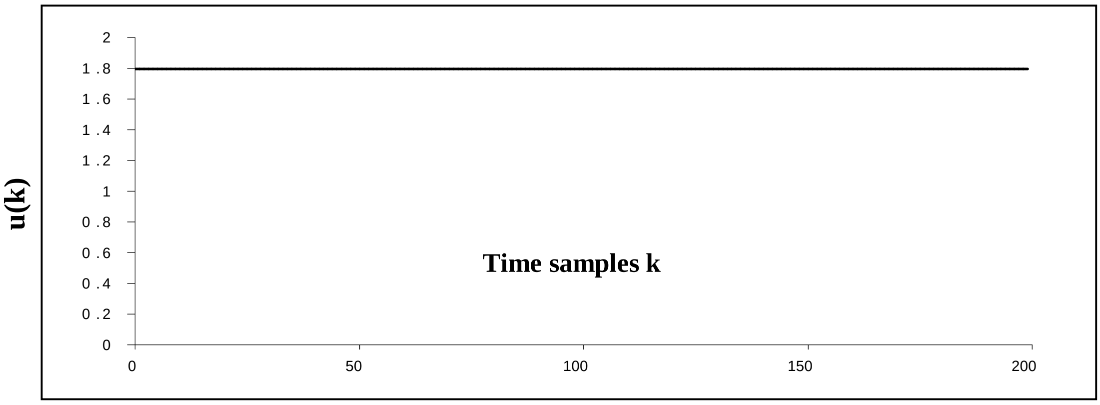

Fig. (9 b) Control signal.

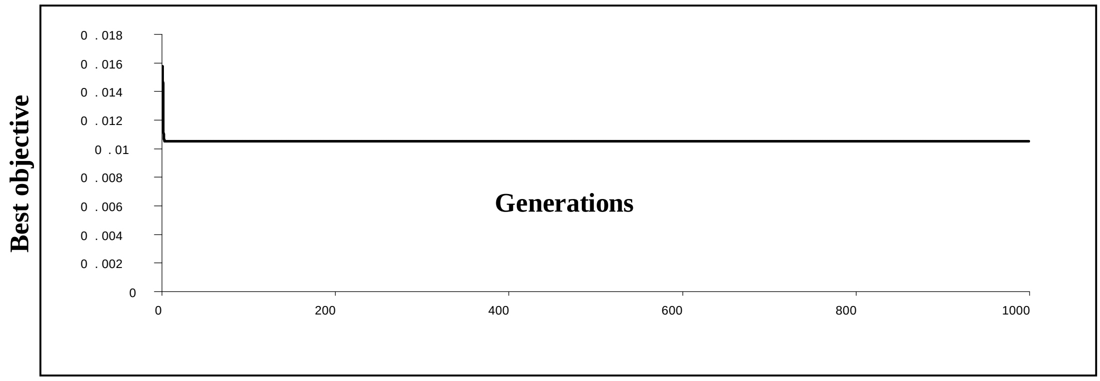

Fig. (9 c) Best MSE.

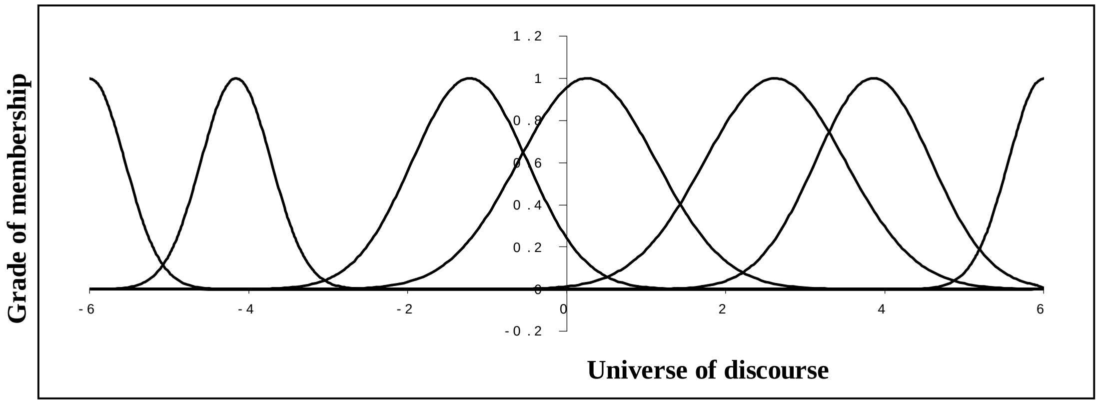

Fig. (9 d) Learned MFs of the input variable.