Indian Journal of Science and Technology, *Vol 8(35), DOI: 10.17485/ijst/2015/v8i35/87127, December 2015*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

New Hybrid Approach to Control the Arm of Flexible Robots by using Neural Networks, Fuzzy Algorithms and Particle Swarm Optimization Algorithm

Khoobjo E*

Department of Electrical Engineering, South Tehran Branch, Islamic Azad University, Iran;
Elhamkhoobjou@yahoo.com

Abstract

Background/Objectives: Mechanizing the instruments is one of the most important and widespread fields which is used in the processes of production and control. **Methods/Statistical Analysis**: Given the complexity and distrust of mechanizing processes, soft computing techniques which are based on physical models have been preferred to common methods in order to predict the performance of processes and optimize them. **Results:** The combination of fuzzy logic and neural networks enables the system to have the capability of learning and adapting to the environment, as well as tolerating the imprecise circumstances which is an advantage of fuzzy logic methods. In this paper, a new hybrid approach is proposed to control the arm of flexible robots by using neural networks, fuzzy algorithms and particle swarm optimization algorithm. The objective is to control robot's claw with two movable arms.

Keywords: Controlling the Flexible Arm, Fuzzy Neural Networks, Particle Swarm Optimization Algorithm, Soft Computing

1. Introduction

In the real world, there are many types of fuzzy knowledge which indicate vague, imprecise, incorrect, uncertain, and obscure behaviors. Computing systems which are based on the classical theory of permutation or two-valued (binary) logic are not able to answer all the questions which human can answer. In many cases, they produce numerous errors even if they have answers. Although the assumption that the machine operates similarly to human is ideal, it is reasonable to expect the proposed system to realize the significant relationships in a problem (with an acceptable margin of error). It is quite clear that the behavior toward an uncertain problem should be flexible; therefore, fuzzy logic is applied to deal with such problems[1].

In this theory, the membership is specified through function μ(x) in the set while *x* indicates a specific member, and μ(x) is a fuzzy function which determines the membership degree of *x* in the respective set. Its value is between zero and one.

$$\tilde{A} = \left\{ (x, \mu_A(x))\, X \right\} \tag{1}$$

The most intriguing application of fuzzy logic is the interpretation that this science provides for the structure of decisions made by smart beings, human intelligence above all. This logic well indicates why binary logic of classic mathematics is not able to explain and describe imprecise concepts, such as heat and cold, which constitute the basis of many smart decisions.

The development of artificial neural networks has started since almost 60 years ago. This science is based on imitating the brain performance in recreating the smart reasoning by computer. This imitation, however, is not the recreation of biological details or nerve cells. It is the exploitation of logical methods of brain performance, and the main principle is transferring these concepts in organization by special software applications. As computing speed increased in computers, the tendency to use neural networks in solving different problems is increasing. Although common computing methods have advantages,

**Author for correspondence*

the nature of their phase-to-phase and non-parallel processing is an obstacle to parallel processes pertaining to neural networks[2].

The artificial neuron is designed in order to imitate the first-order specifications of biological nerve cell. Naturally, a group of inputs are applied, and each one represents the output of another nerve cell. Each input is multiplied by its corresponding weight which represents the connection power. Then all these weighted inputs are summed in order to determine the nerve stimulation level. Figure 1 indicates the neuron performance mathematically.

A group of inputs which are shown as x_k, $k=1, \dots, K$ here are applied to the neuron. These inputs which are totally considered as a vector resemble the signals which are sent to the synapses of the nerve cell. Before being applied to summation unit shown with a , each signal is multiplied by its corresponding weight which represents the power of a biologically synaptic single connection. The summation unit which is slightly similar to the body of biological cell sums all weighted inputs algebraically and produces the output which is here shown by *n* representing NET. This procedure may be briefly stated through a vector symbol which is NET = X.W.

PSO algorithm is a method which was first recommended[3]. It is based on the simulation of a simplified social model. The specifications of particles which seek supply (target) in swarms resulted in using it as a model to achieve optimized answers. In the usual implementation of this algorithm, the first position of the particle is determined haphazardly. Using a series of assumptions and definitions, we can write this natural model in the form of mathematic definitions. To establish a logical correspondence, we assume that a number of data inputs which are called particles are placed in a search-space in which each particle can consider a candidate for the solution. At first, each particle is placed in a random position, and then it flies in a multi-dimensional space. The position of each particle toward itself and toward particles existing in its proximity is adjusted in each repetition (search). In these relocations, each particle stores the best position which is called pbest. It lets the particle select the best solution (fitness) for the local answers to the problem. Also, the best general position of particles is searched and tracked. It is called gbest. PSO algorithm adjusts the position each particle compared with the best value of pbest which has been experienced. Also, the value of gbest is updated in accordance with the best answer to the problem.

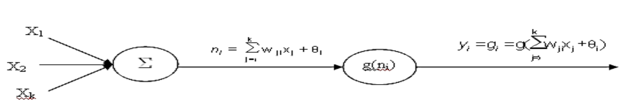


Figure 1. The neuron performance in mathematic language.

2. The Recommended Control System

Integral and derivation operands are frequently used in controlling the smart robot. These operands, which are known as common methods in control systems known as PID, are simulation and maintained in non-classic methods in terms of general structure.

Unlike fuzzy logic controllers which are described in[4,5], a great deal of input and output variables are used. It is because each section is fed by more than one input in systems with high connectivity. Therefore, a multivariable fuzzy controller is designed to control such situation in this system (MIMO). The usual procedure of designing these systems is in a way that the general system is divided into some sub-systems which are Multiple Input Single Output (MISO). Each of these sub-systems goes through the path from input to output in a parallel and coordinated manner[6]. In the proposed method, the minimal values of membership functions are selected in accordance with three reference values, and then the maximizing value is applied. MIMO structures of fuzzy logic controller are shown in Figure 2 for the flexible control of multi-connectional hard connections.

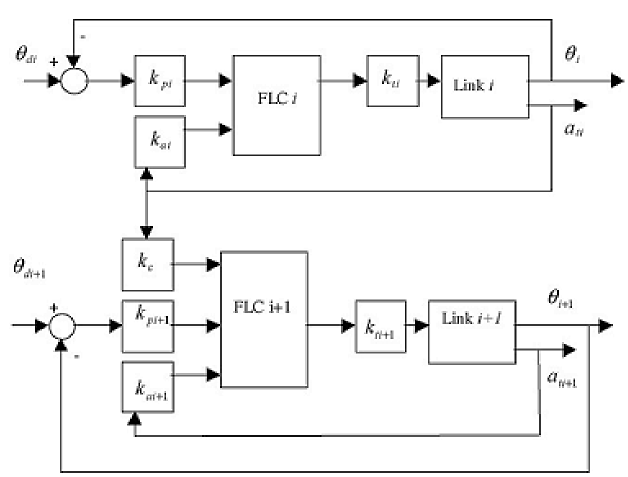


Figure 2. MIMO structure of coupled fuzzy controller.

Two consecutive links (i & $i+1$) are assumed in Figure 3. FLC of input i include angle error connection ($e_i = \theta_{di} - \theta_i$) and acceleration signal at the tip (toe) of a_{ti}. The scale factor is related to the tracking error, acceleration at tip, and input torque for FLC of input i. It is respectively equal to k_{pi}, k_{ai} and k_{ti}. FLC of input $i+1$ also follows the procedure of input i and controls its relevant output. It is recommended that scale factors be normalized in period [-1, 1], so the computing process becomes easier.

3. Radial based Function Neural Network (RBFNN) Training

In RBFNN training, the center and width of the Gaussian Function is kept constant in order to generate a fixed mapping between the input and the hidden layer. RBFNN output weights can be determined by common training algorithms such as Least Mean Square (LME), normalized least mean square, evolutionary algorithms, or PSO. The main advantage of RBF neural networks is its structure simplicity, rapid structure, and high estimation[7]. The structure of this combination consists of three layers named: 1. The input layer which includes input nodes, 2. The hidden layer in which each neuron uses radial based function as its activity function and 3. The output layer which transfers a linear combination of summed values of output layer activities to the output. The proposed neural network structure is shown in Figure 3.

4. Determining Neural Networks Factors by using PSO Algorithm

PSO algorithm is a method which was first proposed[8]. It is based on the simulation of a simplified social model. The specifications of particles which seek supply (target) in swarms resulted in using it as a model to achieve optimized answers.

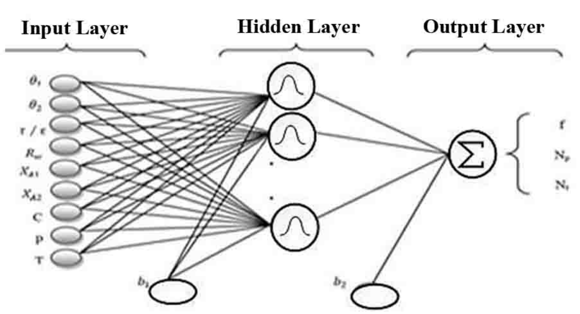


Figure 3. The general structure of proposed radial based neural network.

At first, each particle is placed in a random position, and then it flies in a multi-dimensional space. The position of each particle toward itself and toward particles existing in its proximity is adjusted in each repetition (search). In these relocations, each particle stores the best position which is called pbest. It lets the particle select the best solution (fitness) for the local answers to the problem. Also, the best general position of particles is searched and tracked. It is called gbest. PSO algorithm adjusts the position each particle compared with the best value of pbest which has been experienced. Also, the value of gbest is updated in accordance with the best answer to the problem. This procedure is described by the following relationships.

$$V_i(t+1) = wV_i + c_1 r_1[P_i(t) - X_i(t)] + c_2 r_2[P_g(t) - X_i(t)] \quad (2)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (3)$$

Here, $X_i = \{x_{i1},\dots, x_{in}\}^T \in S$ and $V_i = \{v_{i1},\dots, v_{in}\}^T \in S$, while $S \subset \Re$ is the search-space. $X_i = \{t+1\}$ is the next position of particle which is updated in accordance with the value of $V_i = \{t+1\}$. In fact, this procedure is applied to determine $W_i = \{t+1\}$ from $W_i = \{t\}$; therefore, weight factors are updated.

5. Fuzzy Logic Configuration of the Proposed Method

As it has been pointed out, designing the multi-variable controller for a flexible arm with some links which are in the form of soft and hard connections must benefit from fuzzy logic in order to have a desirable functionality[9]. Control systems are considered to be Multiple Input Multiple Output (MIMO). However, the implementation of such systems is difficult, and its computations are time-consuming. Therefore, the union of some Multiple Input Single Output (MISO) sets is of the proposed method.

Also, a regulation is required for the two-connection arm so that the robot's claw would be placed in the right position. The applied regulation is as follows:

If (A_1 is NM) and (δP_x is PS) then $\delta\, \theta_1$ is NS

Accordingly, the following results will be obtained.

6. Topology of Hybridizing Neural and Fuzzy Sections

Hybridizing fuzzy logic and neural networks falls into two categories for robot control. The first one includes fuzzy-neural controllers which give the training capability of functios to the fuzzy controller. In other words, they conduct some pre-processings before applying fuzzy regulations in order to optimize data. The second one which is named neural-fuzzy controllers embed the linguistic rules and fuzzy logic in the body of neural controller. The topology is used in this paper (Figure 4).

A hybridized method of neural fuzzy network has been used to design a controller in this system. In this method, PSO algorithm was used to decrease error and time in order to generate neural network factors. This network was designed in five layers. It was simulated to control the movement of a robot with two arms in a completely flexible manner in which the network can be momentarily trained by moving the rotation ring. Using Sugeno fuzzy algorithm, circumstances have also been provided for the robot claw to cover even the untrained areas (Figure 5). Moreover, the arm claw can be shown in order to achieve balanced state. This parameter which is very important in movement control systems indicates the speed and precision of the claw in achieving balanced state. It is clearly shown in Figure 6 that this time period is generated with minimal difference (Table 1).

Stable situation occurs all at the same time is a huge advantage in the control system. The Network for Learning PSO algorithm with fuzzy systems Sugeno leading to an arms control is a high-speed stability (Figure 7 and Figure 8).

By using Sugenofuzzy algorithm with 147 Fuzzy Rule,conditions which largely improves mobility paw that the speed is struggling to achieve a better balance

The algorithm is based on a simulation of a social model. The algorithm of the important position of the particle, (Pbest), (gbest) is determined (Figure 9).

The definition of each movement is indicated in Table 2.

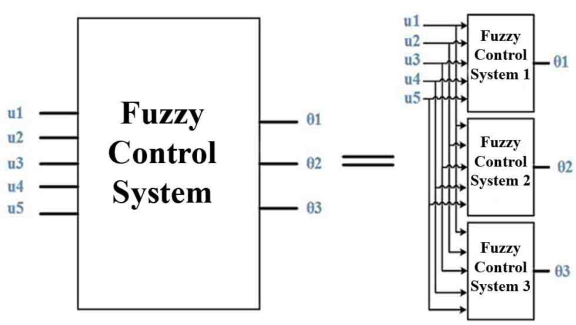


Figure 4. The general structure of proposed radial based neural network.

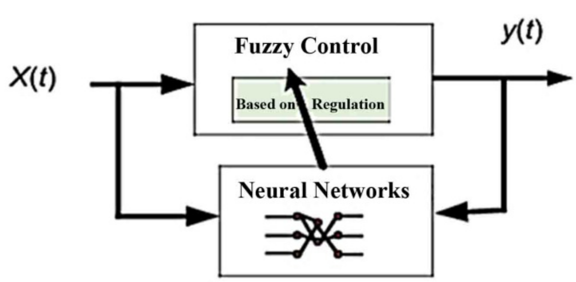
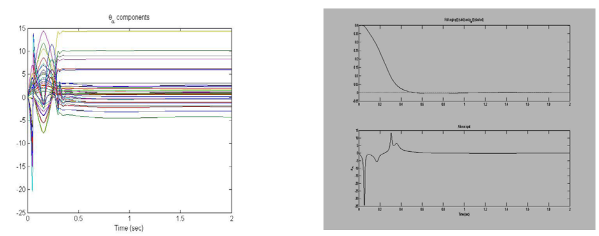


Figure 5. The proposed structure to hybridize neural and fuzzy sections.



(a) (b)

Figure 6. **(a)** Arm's stability in different positions. **(b)** One of the assumed States.

Table 1. Types of movements resulting from fuzzy regulations and claw position

	NB	PB	PM	PS	Z	NS	NM	NB
	NM	PB	PM	PS	Z	NS	NM	NB
	NS	PB	PM	PS	Z	NS	NM	NB
A_i	**Z**	Z	Z	Z	Z	Z	Z	Z
	PS	NB	NM	NS	Z	PS	PM	PB
	PM	NB	NM	NS	Z	PS	PM	PB
	PB	NB	NM	NS	Z	PS	PM	PB

Table 2. Definitions of Robot's claw movement

(PB)	Positive Big
(PM)	Positive Medium
(PS)	Positive Small
(Z)	Zero
(NS)	Negative Small
(NM)	Negative Medium
(NB)	Negative Big

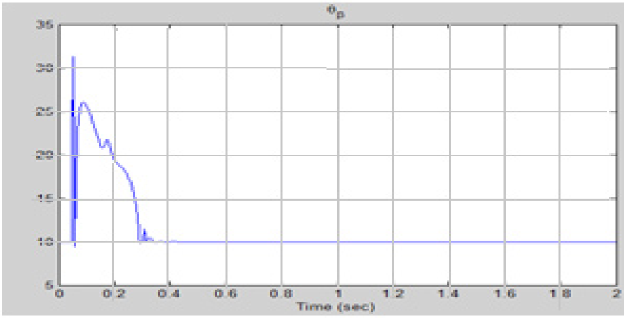


Figure 7. The stability of handles robots.

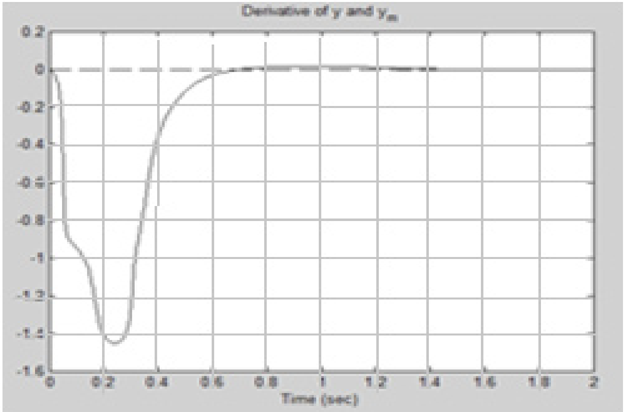


Figure 8. Claw robot reaching equilibrium.

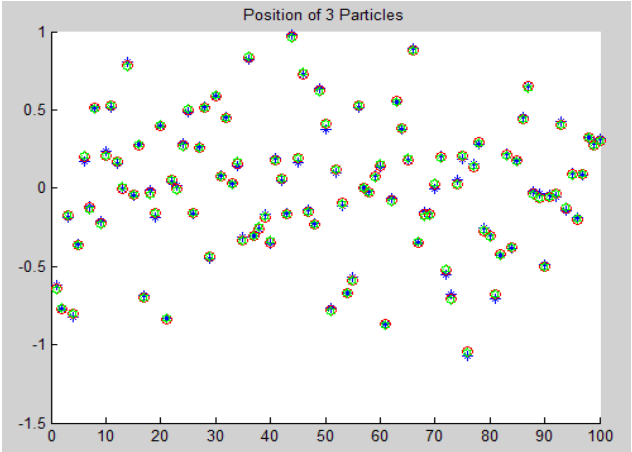


Figure 9. Positions determined by particle swarm algorithm.

7. Conclusion

Soft computing methods are usually used to have an appropriate controller. However, these methods are highly various, and different combinations of them can be used to design a controller. Using the hybridized fuzzy neural method has mostly met the needs of a control system. The results can be improved by applying some changes to the internal structures of each neural or fuzzy framework. For instance, networks such as MLP, Kohonen and other ones can be used instead of RBF neural networks. Also, genetic algorithm, imperialistic competitive algorithm and other ones can be used instead of PSO algorithm in order to determine the factors of neural networks. On the other hand, Mamdani algorithm can be replaced with another algorithm due to high variety of fuzzy algorithm. Each of these changes can cause a change in the result of controller.

8. References

1. Martínez R, Castillo O, Aguilar LT. Optimization of terval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms. Information Sciences. 2009 Jun 13; 179(13):2158–74.
2. Bill R, Jackson VT, Alborzi M. An introduction to neural networks. 3rd ed. Sharif University of Technology Press; 2007. p. 138.
3. Kennedy J, Eberhart RC. Particle swarm optimization. Proc IEEE Int Conf on Neural Networks; Perth, Australia. 1995. p. 1942–8.
4. Lin J, Lewis FL. Fuzzy controller for flexible-link robot arm by reduced-order techniques. IEE Proc Control Theory Appl. 2002; 149(3):177–87.
5. Renno JM. Inverse dynamics based tuning of a fuzzy logic controller for a single link flexible manipulator. J Vibr Control. 2007; 13(2):1741–59.
6. Lee CC. Fuzzy logic in control systems—part I. IEEE Trans Syst Man Cybernetics. 1990 Mar/Apr; 20(2):404–15.
7. Moody J, Darken C. Fast learning in networks of locally-tuned processing units. Neural Comput. 1989; 1(2):281–4.
8. Kasaeipoor A, Ghasemi B, Aminossadati SM. Convection of Cu-water nanofluid in a vented T-shaped cavity in the presence of magnetic field. International Journal of Thermal Sciences. 2015 Aug; 94:50–60.