Hybrid Evolutionary Algorithms based on PSO-GA for Training ANFIS Structure

S.Milad.Nayyer Sabeti [1] and MR.Deevband [2]
[1] Department of Medical Physics and Biomedical engineering, Shahid Beheshti University of Medical Sciences, Tehran, Iran

[2] Department of Medical Physics and Biomedical engineering, Faculty of Medicine, Shahid Beheshti University of Medical Sciences, Tehran, Iran

Abstract
This paper introduces a new approach for training the adaptive network based fuzzy inference system (ANFIS).In this study we apply hybrid of Particle swarm optimization with Genetic Algorithm (PSOGA) in it to the training the antecedent parameters and the conclusion parameters ANFIS structure Finally the method is applied to the identification of nonlinear dynamical system and is compared with basic BP, GA, PSO and showed quite satisfactory results. The proposed method is applied to identification of the nonlinear systems and prediction chaos systems

Keywords: *ANFIS, Genetic Algorithm, Swarm Intelligent, Identification, Training.*

1. Introduction

TSK type [1] [2] is a fuzzy system with crisp functions in consequent, which perceived proper for complex applications [3].It has been proved that with convenient number of rules, a TSK system could approximate every plant [4]. The TSK systems are widely used in the form of a neural-fuzzy system called Adaptive Network-based Fuzzy Inference System ANFIS [5]. ANFIS is a very efficient modeling method by combining the attributes of both of fuzzy inference system and neural network. The combination of fuzzy logic with architectural design of neural network led to creation of neuro- fuzzy systems which benfit from feed forward calculation of output and back propagation learning capability of neural networks, while keeping interpretability of a fuzzy system [3].ANFIS has good ability and performance in system identification, prediction and control and has been applied in many different systems. The ANFIS has the advantage of good applicability as it can be interpreted as local linearization modeling and conventional linear techniques for state estimation and control are directly applicable. The updating and training of ANFIS parameters that consist of the antecedent and conclusion parameters is one of the main problems. The training ANFIS, in the antecedent parameters is more difficult than the conclusion parameters. Various methods have been used to optimize the fuzzy membership functions in recent years. These methods can be divided into two types including derivative based and heuristic algorithms in general [12].

Shoorehdeli et al [6-11] proposed hybrid methods composed particle swarm optimization (PSO) and et al [7] [8] used recursive least square (RLS) and extended Kalman filter (EKL) et al [9] for training. In different studies, they proposed forgetting factor recursive least square for training the conclusion parameters and Lyapunov stability theory to study the stability of used algorithm [6]. In addition to these, they used NSGA-II the training of all parameters of ANFIS structure [11].

Zangeneh et al [13] proposed a new type of training ANFIS is applying complex type (DE/current-to-best/1+1/bin & DE/rand/1/bin) on predicting of Mackey-glass time series.

In this paper, we propose training ANFIS by using hybrid evolutionary Algorithms based on PSO-GA algorithm. The proposed method is applied to prediction of Mackey-glass time series and identification of a nonlinear dynamic system revealing the efficiency of proposed structure. This paper is organized into six sections. In the next section, we review ANFIS. In section 3, we discuss learning algorithms for ANFIS structure. In section 4, the training ANFIS using PSOGA is explained. In section 5, the simulation results are given .Finally, section 6 presents our concluding.

2. The Concept of ANFIS

ANFIS technique was originally presented by Jang [14]. It combines the advantages of fuzzy logic and neural networks. The ANFIS is composed of two approaches neural network and fuzzy. If we compose these two intelligent approaches, it will be achieve good reasoning in quality and quantity. In other words we have fuzzy reasoning and network calculation. ANFIS's structure organizes two parts. The first part consists of antecedent

part and the second part consists of conclusion part. Antecedent part and conclusion part are connected to each other by the fuzzy rules in network form. The updating and training of ANFIS depends on these parts.

An adaptive network is a multilayer feed-forward network in which each node performs a node function on the incoming signal as well as a set of parameters pertaining to this node of which its output depends. These parameters can be fixed or variable, and it is through the change of the last ones that the network is tuned. ANFIS has nodes with variable parameters, called square nodes which will represent the membership functions of the antecedents, and the linear functions for TSK-type consequent. The nodes in the intermediate layers connect the antecedent with the consequent. Their parameters are fixed and they are called circular nodes. Moreover, the network obtained this way would not remain a black box, since this network would have fuzzy inference system capabilities to interpret in terms of linguistic variables [15].The ANFIS structure is demonstrated in five layers. It can be described as a multi-layered neural network as shown in Fig.1.

Layer 1: The first layer is called Fuzzification layer.

The parameters in this layer are referred to as premise parameters. In fact, any differentiable function such as bell and triangular-shaped membership functions (MFs) are valid for the nodes in this layer. Every node i in this layer is a square node with a node function. Usually MFs are used as Gaussian with maximum equal to 1 and minimum equal to 0 such as:

$$\mu_{A_i}(\mathrm{x}) = \frac{1}{1+[(\frac{x-c_i}{a_i})^2]^{b_i}} \quad (1)$$

$$\mu_{B_i}(\mathrm{x}) = \frac{1}{1+[(\frac{x-c_i}{a_i})^2]^{b_i}} \quad (2)$$

Where $\{a_i, b_i, c_i\}$ are the parameters of MFS which are affected in shape of MFs. The parameters in this layer are called the antecedent parameters.

Layer 2: This layer is called rule layer.

The rule layer represents the firing strength for each rule being generated in fuzzification layer. They are circular nodes with a node function:

$$R_1 = O_1O_3, R_2 = O_1O_4, R_3 = O_2O_3, R_4 = O_2O_4 \quad (3)$$

Layer 3: This layer is called normalization layer.

Normalize layer calculates the normalized firing strength for each of the inputs. This normalization is the ratio of the firing strength of the i[th] rule to the total of all firing strengths as given (4).

$$R_i^{'} = \frac{\sum_{i=1}^{4} R_i}{R_1 + R_2 + R_3 + R_4} \quad (4)$$

Layer 4: This layer is called defuzzification layer.

Every node i in this layer is a square node with a node function:

$$F_i(\mathrm{x}_1, \mathrm{x}_2) = \alpha_0^i + \alpha_1^i x_1 + \alpha_2^i x_2 \quad (5)$$

$$g_i = \frac{F_i(\mathrm{x}_1, \mathrm{x}_2)\,\mathrm{R}_i}{R_1 + R_2 + R_3 + R_4} \quad (6)$$

Layer 5: This layer is called sum layer.

The single node in this layer computes the overall output as the sum of all incoming signals.

$$O(\mathrm{x}_1, \mathrm{x}_2) = \frac{\sum_{i=1}^{4} F_i(\mathrm{x}_1, \mathrm{x}_2)\,\mathrm{R}_i}{R_1 + R_2 + R_3 + R_4} = g_1 + g_2 + g_3 + g_4 \quad (7)$$

$$O(\mathrm{x}_1, \mathrm{x}_2) = \frac{\sum_{i=1}^{4} F(\mathrm{x}) \prod_{i=1}^{2} \mu_{A_i^1}(\mathrm{x}_i)}{\sum_{i=1}^{4} \prod_{i=1}^{2} \mu_{A_i^1}(\mathrm{x}_i)} \quad (8)$$

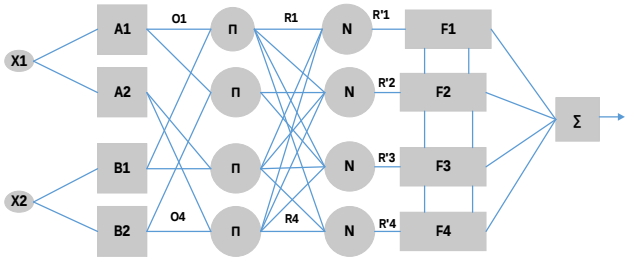

Fig. 1. The ANFIS network with 2 inputs and 2 MF for an input

ANFIS approximation ability will depend on the resolution of the input space partitioning which is determined by the number of MFs in ANFIS and the number of layers.

3. Learning Algorithms

The subsequent to the development of ANFIS approach, a number of methods have been proposed for learning the parameters and for obtaining an optimal number of MFs. Jang [16] is introduced four methods to update the parameters of ANFIS structure, as listed below according to their computation complexities:

- All parameters are updated by the gradient descent.
- After the network parameters are set with their initial values, the consequent part parameters is adjusted through the LSE which it is applied only once at the very beginning. Then, the gradient decent updates all parameters.
- The hybrid learning combining gradient descent and LSE.
- Using extended kalman filter to update all parameters.

In this paper we used a hybrid method which has less complexity and fast convergence

3.1 Genetic algorithms

Genetic algorithms (GAS) are a family of computational models developed by Holland who is inspired by evolution. [17] [18]. These algorithms encode a potential solution to a specific problem on a simple chromosome like data structure and apply recombination operators to these structures so as to preserve critical information. GAS are often viewed as function optimizers, although the range of problems to which GAS have been applied is quite broad. This GA procedure is sketched in Algorithm1.

Algorithm 1 (Pseudo code version of the GA algorithm):

{Initialization}
g←0 /* g: generation counter */
for i = 1 to M **do** /* M: population size*/
Initialize individuals xi to random values
Fi ← f (x_i) /* f: fitness function */
end for
Pop←{x_1, x_2. . . x_M}
F← {F_1, F_2, . . ., F_M}
{Main Loop}
while (not termination condition) **do**
{Genetic Operators}
Pop← Selection (Pop, F)
Pop ←Crossover (Pop, p_c) /*pc: probability of crossover*/
Pop ←Mutation (Pop, p_m)/* pm: probability of mutation */
{Evaluation Loop}
for i = 1 to M **do**
Fi← f (xi)
end for
F← {F_1, F_2, . . ., F_M}
g← g + 1
end while

In GA populations are formulated as abstract representations (called chromosomes) of candidate solutions (called individuals or phenotypes) to an optimization problem. Typically, the algorithm maintains a population of M individuals Pop (g) ={x_1(g),..., x_M(g)} for each iteration g (also called generation), where each individual represents a potential solution of the problem. The algorithm is an iterative process in which new populations are obtained using a selection process based on individual adaptation and some "genetic" operators (crossover and mutation). In each generation, the fitness (a measure of the quality of the represented solution) of every individual in the population is evaluated. The individuals with the best adaptation measure have more chance of reproducing and generating new individuals by crossing and muting. The selection process is repeated several times and the selected individuals form a tentative new population for further genetic operator actions. After selection some of the members of the new tentative population undergo transformations. A crossover operator creates two new individuals (off springs) by combining parts from two randomly selected individuals of the population. In GA the crossover operator is randomly applied with a specific probability, p_c. A good GA performance requires the choice of a high crossover probability. Mutation is a unitary transformation which creates, with certain probability, pm, a new individual by a small change in a single individual. In this case, a good algorithm performance requires the choice of a low mutation probability (inversely proportional to the population size).The mutation operator guarantees that all the search space has a nonzero probability of being explored. Using these genetic operators, the general structure of the algorithm is sketched in Algorithm 1. This procedure is repeated several times (thus yielding successive generations) until a termination condition has been reached. Common terminating criteria are that a solution is found that satisfies a lower threshold value, or that a fixed number of generations has been reached, or that successive iterations no longer produce better results. Ideally, the algorithm is expected to evolve over time toward better solutions, although convergence to global optima cannot be generally assured.

3.2 Particle swarm optimization

Particle swarm optimization (PSO) is also an evolutionary computational model which is based on swarm intelligence. PSO is developed by Kennedy and Eberhart who have been inspired by the research of the artificial life [18]. Similar to GA, PSO is also an optimizer based on population. The PSO does not possess the crossover and mutation processes adopted in GA. It finds the optimum solution by swarms following the best particle. Compared to GA, the PSO has much more profound intelligent background and could be performed more easily. In PSO algorithm, the solution of each problem is

a bird in the search space called particle. Every particle has a fitness value which is obtaining by fitness function. The bird which is closer to food has more fitness value. PSO starts with a group of random answers (particles), and then it searches for finding optimal answer in problem space by updating particles position. Every multi-dimensional particle (depending on the problem nature) is specified by P_{id} and V_{id} which denote the location position and speed of dimension d^{th} of particle i. In every phase of swarm movement, position of each particle is updated by two best values. The first value is the best answer according to fitness which is obtained for each particle separately until now and is called P_g^b. Another value is the best value that is obtained by all of particles through total swarms until now and is called P_i^b. If a neighborhood is defined for every particle, this value is computed in that neighborhood. In this case, this value is called P_i^g. In every iterations of the algorithm, the new speed and position of each particle are updated by (9) and (10), after finding and [19]:

$$V_i(s+1) = wV_{id}(s) + \gamma_1 R_1[P_g^b(s) - P_i(s)] + \gamma_2 R_2[P_b(s) - P_i(s)] \quad (9)$$

$$P_i(s+1) = P_{id}(s) + V_{id}(s) \quad (10)$$

Where $P_{id}(s)$ and $V_{id}(s)$ are respectively the position and the velocity of particle i at time s, w is called inertia weight and decide how much the old velocity will affect the new one and coefficients _1and _2 are constant values called learning factors, which decide the degree of affection of P_g^b and P_i^b. In particular, _1 is a weight that accounts for the "social" component, while _2 represents the "cognitive" component, accounting for the memory of an individual particle along the time. Two random numbers, R_1 and R_2, with uniform distribution on [0, 1], are included to enrich the searching space. Finally, a fitness function must be given to evaluate the quality of a position. This PSO procedure is sketched in Algorithm 2.

Algorithm 2 (pseudo code version of the PSO algorithm):

{Initialization}
s←0 /* s: time variable*/
for i = 1 to N **do** /* N: size of the swarm */
Initialize vectors V_i and P_i to random values
$P_i^b \leftarrow P_i$
end for
$P_g^b \leftarrow$best { P_i^b ; i=1. . . N} /* initial global best */
{Main Loop}
while (not termination condition) **do**
{Evaluation Loop}
for i = 1 to N **do**
if f(P_i) is better than f (P_i^b) **then** /* f: fitness function*/
$P_g^b \leftarrow P_i$ /* particle's best position */
end if
if f (P_i^b) is better than f (P_g^b) **then**
$P_g^b \leftarrow P_i^b$ /* swarm's best position */
end if
end for
{Update Loop}
for i = 1 to N **do**
$V_i \leftarrow w.V_i + \gamma_1.R(0,1).(P_g^b - P_i) + \gamma_2.R_2(0,1).(P_i^b - P_i)$
$P_i \leftarrow P_i + V_i$
end for
s ←s + 1
end while

This procedure is repeated several times (thus yielding successive generations) until a termination condition is reached. Common terminating criteria are that a solution is found that satisfies a lower threshold value, or that a fixed number of generations has been reached, or that successive iterations no longer produce better results.

3.3 Hybridization of GA and PSO

The drawback of PSO is that stochastic approaches have problem-dependent performance. This dependency usually results from the parameter settings in each algorithm. In general, no single parameter setting can be applied to all problems. Increasing the inertia weight (w) will increase the speed of the particles resulting in more exploration (global search) and less exploitation (local search). Thus finding the best value for the parameter is not an easy task and it may differ from one problem to another. A further drawback is that the swarm may prematurely converge. Therefore, from the above, it can be concluded that the PSO performance is problem-dependent. The problem-dependent performance can be addressed through hybrid mechanism. It combines different approaches to be benefited from the advantages of each approach. To overcome the limitations of PSO, hybrid algorithms with GA are proposed. The basis behind this is that such a hybrid approach is expected to have merits of PSO with those of GA. One advantage of PSO over GA is its algorithmic simplicity [20]. This simplicity will result in the increase of speed calculations and the reaching to the desired answer with low volume of memory .Another clear difference between PSO and GA is the ability to control convergence. Crossover and mutation rates can subtly affect the convergence of GA, but these cannot be analogous to the level of control achieved through

manipulating of the inertia weight. In fact, the decrease of inertia weight dramatically increases the swarm's convergence. The main problem with PSO is that it prematurely converges to stable point, which is not necessarily maximum. To prevent the occurrence, position update of the global best particles is changed. The position update is done through some hybrid mechanism of GA. By applying crossover operation, information can be swapped between two particles to have the ability to fly to the new search area. The purpose of applying mutation to PSO is to increase the diversity of the population and the ability to have the PSO to avoid the local maxima. There are three different hybrid approaches are proposed in [20].in this paper used PSO-GA-series hybrid evolutionary algorithm (PSOGA). The PSO-GA performs N PSO's populations simultaneously at first. After M_1 iterations the best particles in each population are selected to encode into chromosomes to constitute an N-individual-population for GA operations. Then the population should be run using GA-operators. After M_2 iterations the best solution of GA should be transmitted back to PSO populations. We define gap-PSO, gap-GA and gap as the iteration numbers of PSO subsystem, GA sub-system, and whole system, respectively; M_1, M_2 and GAP-MAX the corresponding permissible maximum iteration numbers, respectively[21].The flow chart of the PSO-GA is shown in Fig 2.

4. Learning by PSOGA

In this section, the training and updating of ANFIS parameters using PSO-GA is explained. The ANFIS has two types of parameters which need training, the antecedent part parameters and the conclusion part parameters. The membership functions are assumed Gaussian as in equation (1, 2), and their parameters are $\{a_i, b_i, c_i\}$, where a_i is the variance of membership functions and c_i is the center of MFs. Also b_i is usually equal to 1. The parameters of conclusion part are trained and here are represented with $\{p_i, q_i, r_i\}$. There are 3 sets of trainable parameters in antecedent part , each of these parameters has N genes. Where, N represents the number of MFs. The conclusion parts parameters also are trained during optimization algorithm. Each chromosome in conclusion part has (I +1) ×R genes that R is equal to Number of rules and I denotes dimension of data inputs. The fitness is defined as RMSE [9]. Parameters are initialized randomly in first step and then are being updated using PSO-GA algorithms. In each iteration, one of the parameters set are being updated. i.e. in first iteration for example a_i s are updated then in second iteration bi are updated and then after updating all parameters again the first parameter update is considered and so on.

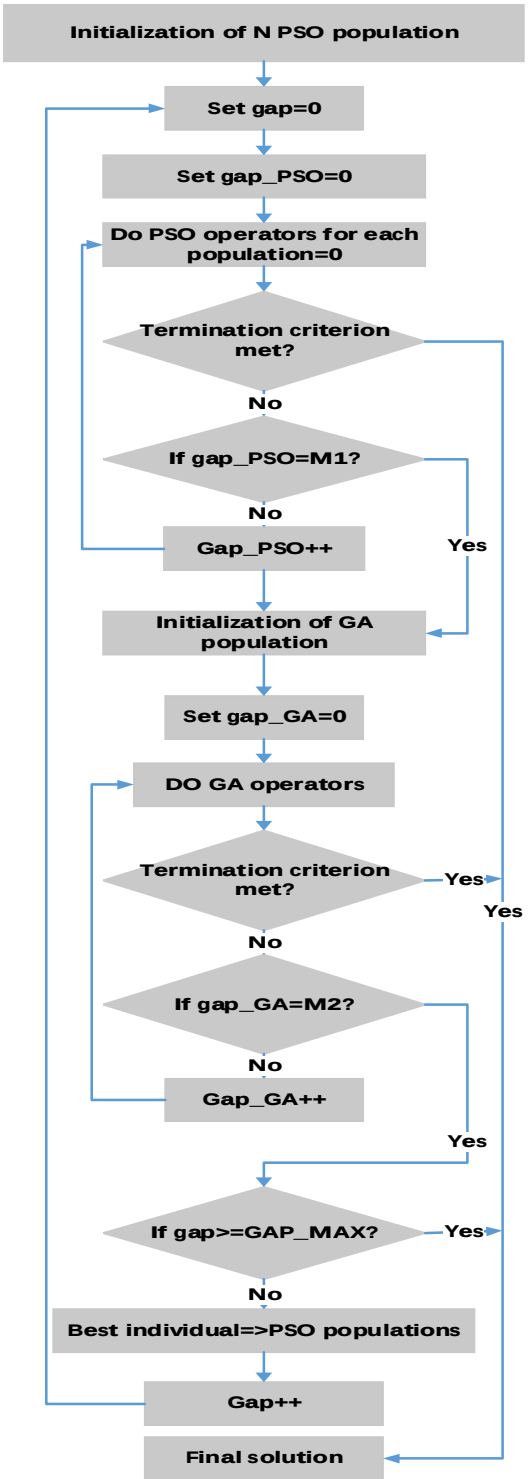

Fig. 2. Flow chart of PSO-GA

5. Simulation Results

This section presents the simulation results of ANFIS training using back propagation (BP), genetic algorithm (GA), particle swarm optimization (PSO) and PSOGA algorithm. In this section, we use two different types of training data set: a single-input and single-output training data set, four-input and single-output training data set.

A. Example 1(Nonlinear Function Modeling)

In this application, a single-input and single-output training data set is used defined by (11). Defining range of x is [-1, 1], the system produces 51 groups of input and output data.

$$y = 0.6\sin(\pi x) + 0.3\sin(\pi x) + 0.1\sin(5\pi x) \qquad (11)$$

The parameters for training the ANFIS by PSO-GA listed in Table 1.

Table1: parameters PSOGA for training ANFIS

Swarm size (population size)	20
Num.of epochs	100
Crossover percentage(p_c)	0.8
Mutation percentage(p_m)	0.3
Mutation rate(μ)	0.02
Gamma(γ)	0.2
Selection pressure(β)	8
Constriction coefficients(φ)	2.05

The results are given in Table 2.This Table shows train, test RMSE and STD (standard deviations) for different types of training ANFIS structure. The RMSE of PSOGA is 0.02 for 10 rules, as given in Table 2. The comparisons with ANFIS validate the performance of the PSOGA.

Table 2: Test, Train RMSE and STD for Example 1

Num. rule	***Train by***	***Train. RMSE***	***Test. RMSE***	***Train. STD***	***Test. STD***
2	***BP***	0.158	0.158	0.16	0.16
4		0.146	0.147	0.14	0.14
6		0.108	0.108	0.109	0.109
8		0.102	0.102	0.103	0.103
10		0.076	0.076	0.075	0.075
2	***PSO***	0.129	0.129	0.13	0.13
4		0.137	0.137	0.138	0.138
6		0.1	0.1	0.1	0.1
8		0.09	0.08	0.09	0.09
10		0.07	0.07	0.07	0.07
2	***GA***	0.158	0.158	0.16	0.16
4		0.145	0.145	0.146	0.146
6		0.127	0.127	0.129	0.129
8		0.171	0.171	0.171	0.171
10		0.16	0.16	0.16	0.16
2	***PSO-GA***	0.153	0.153	0.155	0.152
4		0.118	0.118	0.12	0.12
6		0.037	0.037	0.037	0.037
8		0.042	0.042	0.042	0.042
10		0.02	0.02	0.02	0.02

B. Example 2(Predicting Chaos Dynamics)

The data is generated from the Mackey-Glass time-delay differential equation which is defined by:

$$\dot{x} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \qquad (12)$$

When x (0) = 1.2 and τ = 17, we have a non-periodic and non-convergent time series that is very sensitive to initial conditions. (We assume x (t) = 0 when t < 0.) ANFIS structure has 4 inputs and one output. We use 807/346 data as training/test. Fig.3-11 is depicted predicting of Mackey-Glass time series using PSOGA, GA, BP, and PSO to train parameters in ANFIS structure. This graphic shows train, test data for targets and outputs and RMSE for errors using by different types of training ANFIS structure. The numbers of iterations of this algorithm and population size is 1000 and 100 respectively, we used 10 rules and other parameters is according Table 1.

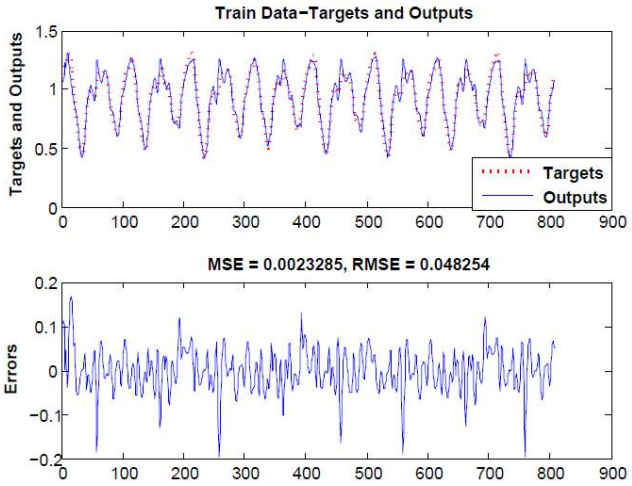

Fig3.The ANFIS trained by GA for Train data and RMSE

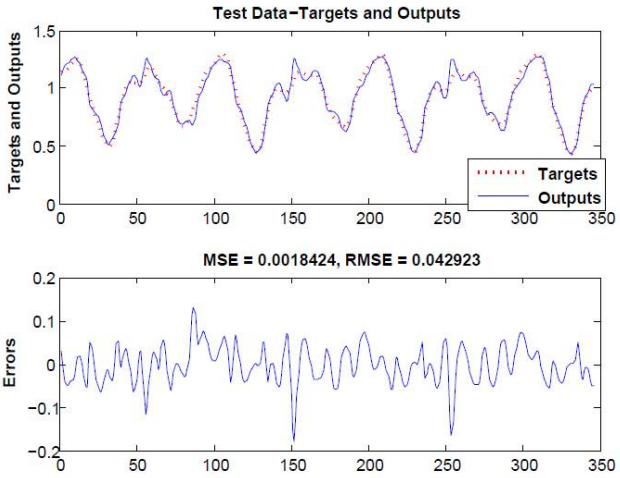

Fig4. The ANFIS trained by GA for Test data and RMSE

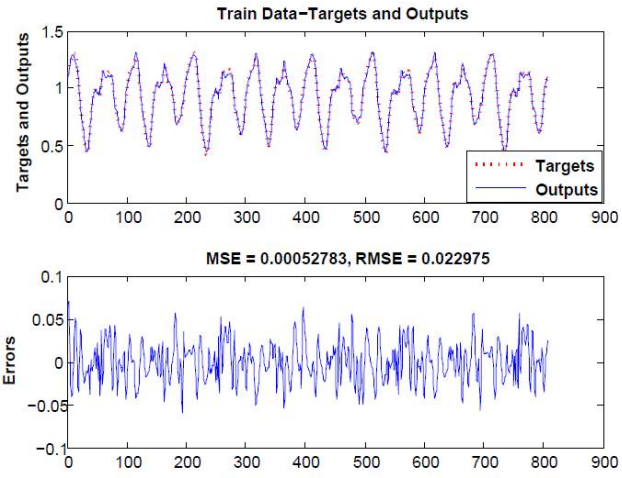

Fig5. The ANFIS trained by PSO for Train data and RMSE

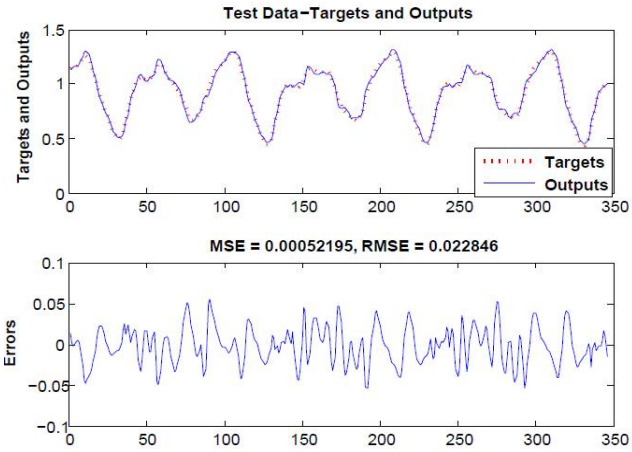

Fig6. The ANFIS trained by PSO for Test data and RMSE

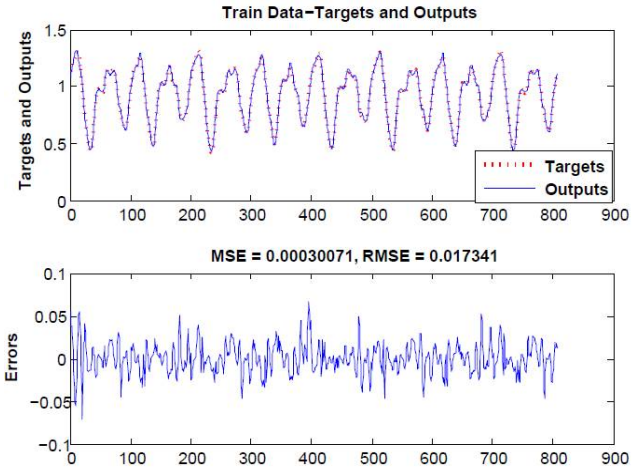

Fig7.The ANFIS trained by BP for Train Data and RMSE

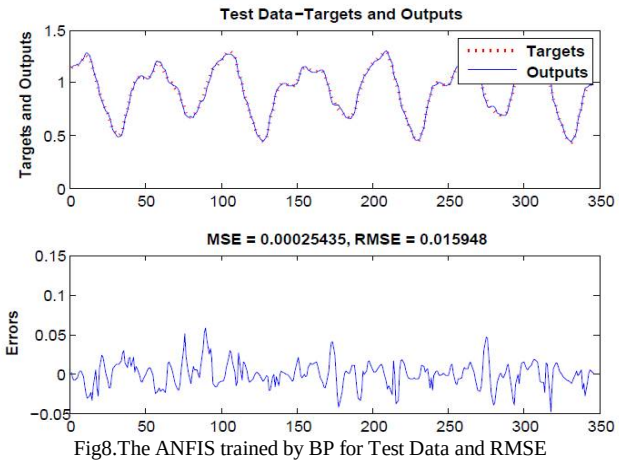

Fig8.The ANFIS trained by BP for Test Data and RMSE

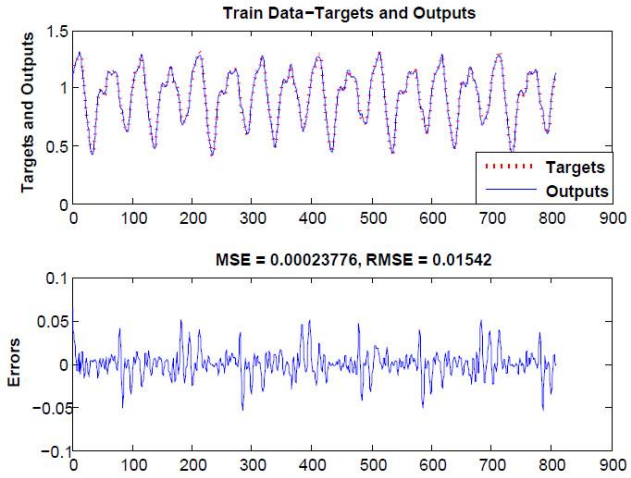

Fig9.The ANFIS trained by PSOGA for Train Data and RMSE

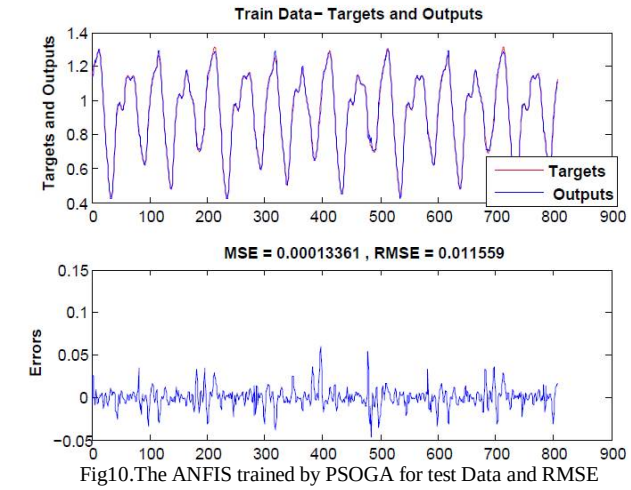

Fig10.The ANFIS trained by PSOGA for test Data and RMSE

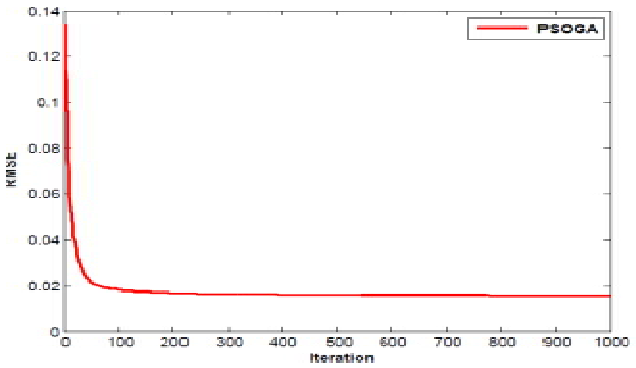

Fig11.The RMSE-Iteration ANFIS using by PSOGA

The simulations results showed PSOGA optimizes ANFIS parameters better than GA, BP and PSO for test and train data.

6. Conclusions

In this paper, we proposed a novel method for training the parameters of ANFIS structure. In our novel method we used PSOGA for updating the parameters. The simulation results show that PSOGA is successful for training ANFIS for complex nonlinear systems and prediction chaos systems. Since this algorithm is free of derivation which is very difficult to calculate for training of antecedent and conclusion part parameters complexity of this new approach is less than other training algorithms like BP,GA and PSO. Also, the local minimum problem in BP algorithm for training in this novel approach is solved. The effectiveness of the proposed PSOGA method was shown by applying it to identification of nonlinear model.

Acknowledgments

This work is part of my master's thesis. Authors are indebted to the Department of Medical Physics and Biomedical engineering of Shahid Beheshti University for the support provided during this study.

References

[1] M.Sugeno and G. T. Kang, "Structure identification of fuzzy model," Fuzzy sets and systems, pp.15-33, 1998.

[2] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," IEEE Trans. Sys., Man& Cybernetics, pp.116-132, 1985.

[3] R. Alcala, J. Casillas, O. Cordon and F. Herrera, Learning TSK rule based system from approximate ones by mean of MAGUL methodology. Granada university of Spain, Oct.2000.

[4] M. Mannle, FSTM: Fast Takagi-Sugeno fuzzy modeling. University of Karsruhe, 1999.

[5] Jyh-Shing Roger Jang, "ANFIS: Adaptive Network Based Fuzzy Inference System," IEEE Trans. Sys., Man & Cybernetics, vol.23, no.3, May-June 1993.

[6] M. A. Shoorehdeli, M. Teshnehlab and A. Sedigh, "Training ANFIS as an identifier with intelligent hybrid stable learning algorithm based on particle swarm optimization and extended Kalman filter," Fuzzy Sets and Systems , vol. 160, pp. 922-948, 2009.

[7] M. A. Shoorehdeli, M. Teshnehlab and A. Sedigh, "A novel training algorithm in ANFIS structure," in American Control Conference, IEEE, June 2006.

[8] M. A. Shoorehdeli, M. Teshnehlab and A. Sedigh, "Identification using ANFIS with intelligent hybrid stable learning algorithm approaches," Neural computing & applications, vol. 18(2), pp. 157-174, 2009.

[9] M. A. Shoorehdeli, M. Teshnehlab and A. Sedigh, "Novel hybrid learning algorithms for tuning ANFIS parameters

using adaptive weighted PSO," In Fuzzy Systems Conference, fuzzy-IEEE 2007, IEEE International , pp. 1-6, 2007.
[10] M. A. Shoorehdeli, M. Teshnehlab, A. K. Sedigh and M. A. Khanesar, "Identification using ANFIS with intelligent hybrid stable learning algorithm approaches and stability analysis of training methods," Applied Soft Computing, vol. 9(2), pp. 833-850, 2009.
[11] V. S. Ghomsheh, M. A. Shoorehdeli, M. Teshnehlab, "Training ANFIS structure with modified PSO algorithm," In Control & Automation, MED'07. Mediterranean Conference on IEEE, pp. 1-6, June 2007.
[12] A. Chatterjee and K. Watanabe, "An optimized Takagi-Sugeno type neuro-fuzzy system for modeling robot manipulators," Neural Computing & Applications, vol. 15(1), pp. 55-61, 2006.
[13] A. Z. Zangeneh, M. Mansouri, M. Teshnehlab and A. K. Sedigh," Training ANFIS system with DE algorithm," In Advanced Computational Intelligence (IWACI), IEEE 2011 Fourth International Workshop on, pp. 308-314, October 2011.
[14] J. S. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," Systems, Man and Cybernetics, IEEE Transactions on 23.3, pp. 665-685, 1993.
[15] M. Kumar and P. G. Devendra, "Intelligent Learning of Fuzzy Logic Controllers via Neural Network and Genetic Algorithm," Proceedings of 2004 JUSFA 2004 Japan-USA Symposium on Flexible Automation
Denver, Colorado, 2004.
[16] Jyh-Shing Roger Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System," IEEE Trans. Sys.Man and Cybernetics. Vol. 23, No.3, May/June 1993.
[17] J.H. Holland, Adaptation in Natural and Artificial System, the University of Michigan Press, Ann Arbor, 1975.
[18] D.E.Goldberg, Genetic Algorithms in Search, Optimization & Machine Learning. Reading, MA: Addison Wesley, 1989.
[19] J. Kennedy, R.C. Eberhart, "Particle swarm optimization", Proceedings of IEEE International Conference on Neural Networks, vol. 4, 1995.
[20] K. Premalatha and A.M. Natarajan, "Hybrid PSO and GA for Global Maximization", Int. J. Open Problems Compt. Math., Vol. 2, No. 4, pp. 597-608, December 2009.
[21] Shi, X.H.; Lu, Y.H.; Zhou, C.G.; Lee, H.P.; Lin, W.Z.; Liang, Y.C., "Hybrid evolutionary algorithms based on PSO and GA," in Evolutionary Computation, 2003. CEC '03. The 2003 Congress on, vol.4, no., pp.2393-2399 Vol.4, 8-12 Dec. 2003.

S. Milad. Nayyer Sabeti was born in Rasht, Iran, in 1986. He received the B.Sc. degree from the University of Rasht, Iran, in electrical engineering since 2008.and the M.Sc. degree from the Shahid Beheshti University of Medical Sciences and Health Services, Tehran, Iran in biomedical engineering. He has been working toward the M.Sc. degree in biomedical engineering at the Shahid Beheshti University. His main research interests include bio robotic, signal processing& medical image processing, computational intelligence, intelligent optimization, and multi objective optimization. He is currently doing research in the field of control bio robotic systems, as M.Sc. thesis.

MR.Deevband received the B.Sc. degree from the shahid bahonar Kerman, Tehran, Iran, the M.Sc. degree from the Isfahan University of Medical Science, Tehran, Iran, and the Ph.D. degree from Tarbiat Modaress University, Tehran, Iran, in 1998, all in medical phiysic.From 2011 , he was a Faculty Member at the Shahid Beheshti University of Medical Sciences and Health Services, Tehran, Iran. He is currently an Associate Professor with the Faculty Medical Physics and Biomedical engineering Department of Medical Physics and Biomedical engineering. His interest interests include medical image processing, computational intelligence, intelligent optimization and radiation protection.