International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 4, No.4 (Oct-2015)
http://dx.doi.org/10.12785/ijcds/040405

Takagi-Sugeno Fuzzy System Accuracy Improvement with A Two Stage Tuning

Hassan M. Elragal

Department of Electrical and Electronics Engineering, University of Bahrain, Kingdom of Bahrain
Department of Electrical Engineering, Faculty of Engineering, Alexandria University, Alexandria, Egypt

Received 1 May 2015, Revised 1 July 2015, Accepted 20 August 2015, Published 1 October 2015

Abstract: In this paper a method for improving accuracy of a Takagi-Sugeno type fuzzy system used in Matlab Fuzzy Logic Toolbox as genfis3 (Sugeno) is proposed. For this fuzzy system, the input space is partitioned using Fuzzy C-means (FCM) clustering algorithm and the consequent parameters are optimized using least square. This improvement is done in a two stage tuning using particle swarm optimization (PSO).In the first stage, PSO is used to optimize input membership functions (mean and variance) and consequent parameters of Takagi-Sugeno fuzzy system. In the second stage, PSO is used to optimize a weighting factor for the rules and a scaling universe of discourse for inputs and outputs variables. To simplify the tuning process and performing it in one optimization stage, a one stage tuning is also discussed to tune same parameters optimized in the two stage tuning. Experimental results with real data applied in data classification problem shows a consistency of getting higher classification accuracy with the proposed tuning methods over the original system.

Keywords: Fuzzy system, Particle swarm optimization, Optimization of fuzzy system parameters

1. INTRODUCTION

Fuzzy systems have been successfully applied in many fields such as expert systems, prediction, computer vision, control, decision analysis, and data classification [1]-[2]. Many approach such as statistical clustering methods, neural networks and evolutionary programmings have been proposed for designing an optimal fuzzy system, [3]-[5].

In data classification problems, fuzzy rules are derived from human experts. Since it is not easy to derive fuzzy rules from human experts, many approaches have been proposed to generate fuzzy rules automatically from training patterns. Fuzzy partition in the input space is generally considered to generate fuzzy rules from training patterns. Two types of input space partition have been often used to model fuzzy systems. The first one is a grid fuzzy partition [6]-[8] and the second one is a scatter fuzzy partition [9].

Fuzzy C-mean (FCM) [10] and subtractive clustering algorithms [11] are always been used in fuzzy partitioning. A critical problem for the fuzzy clustering algorithms is how to determine the optimal number of clusters. Too many clusters result in an unnecessarily complicated rule base, while too few clusters result in a poor performance. Many papers proposed different methods for building and tuning fuzzy model directly from the input-output data [12]-[15].

Evolutionary algorithms such as genetic algorithms (GAs), simulated annealing (SA), particle-swarm optimization (PSO) and differential evolution (DE) have been widely used in solving global optimization problems [16].Evolutionary algorithms are used to tune fuzzy membership functions, size and structure of fuzzy rules, and optimize the partition number of input variables [14]-[15],[17].

A method for improving accuracy of a Takagi-Sugeno type fuzzy system used in Matlab Fuzzy Logic Toolbox as genfis3 (Sugeno) [18] is discussed. This improvement is done in a two stage tuning using (PSO).In the first stage, PSO is used to optimize the parameters of input membership functions (mean and variance) and consequent parameters of Takagi-Sugeno fuzzy system. In the second stage, PSO is used to optimize a weighting factor for the rules and a scaling factor for inputs and outputs. The performance of the proposed tuning method is tested on five real-world databases.

E-mail:helragal@uob.edu.bh

The rest of this paper is organized as follows: Section2 shows the structure of Takagi- Sugeno fuzzy systems. Section 3 describes a special version of PSO algorithm used in this paper. Section 4 discuss the original Takagi-Sugeno genfis3 fuzzy system, the two stage tuning method used and combining the two stages into only one stage tuning. Section 5shows the experimental results. Concluding remarks are shown in Section 6.

2. STRUCTURE OF TAKAGI–SUGENO FUZZY SYSTEM

The basic architecture of a fuzzy system is shown in Fig. 1. The main components are a fuzzification interface, a fuzzy rule base (knowledge base), an inference engine (decision-making logic), and a defuzzification interface.

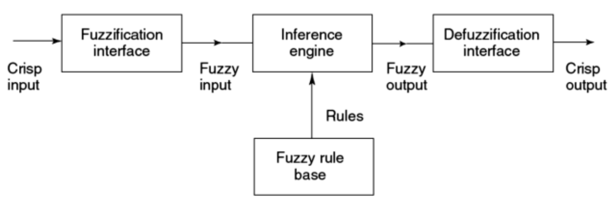

Figure 1. Basic fuzzy system

The two most popular fuzzy inference systems that have been widely deployed in various applications are Mamdani fuzzy inference system [19] and Takagi-Sugeno fuzzy inference system [20]. The differences between these two fuzzy inference systems lie in the consequents of their fuzzy rules, and thus their aggregation and defuzzification procedures differ accordingly.

For Mamdani, fuzzy inference system, the rule antecedents and consequents are defined by fuzzy sets and have the following structure:

Rule i: ***IF*** *x is* A_i ***AND*** *y is* B_i ***THEN*** Z_i *is* C_i

There are several defuzzification techniques. The most widely used defuzzification technique is centroid of area.

Takagi–Sugeno fuzzy inference system is illustrated in Fig. 2. Takagi and Sugeno proposed an inference scheme in which the conclusion of a fuzzy rule is constituted by a weighted linear combination of the crisp inputs rather than a fuzzy set. The fuzzy rule has the following structure:

Rule i: ***IF*** *x is* A_i ***AND*** *y is* B_i ***THEN*** $Z_i = p_{i,1}x + p_{i,2}\ y + p_{i,0}$

Where $p_{i,1}$, $p_{i,2}$, and $p_{i,0}$ are the consequent parameters.

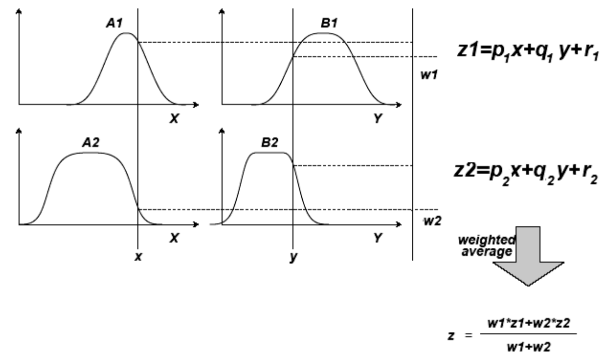

Figure 2. Takagi–Sugeno fuzzy inference system

3. PARTICLE SWARM OPTIMIZATION (PSO)

Particle swarm optimization (PSO) is a swarm intelligence method for global optimization [21]. For classical PSO, each individual of the population adjusts its trajectory toward its own previous best position, and toward the previous best position attained by any member of its topological neighborhood. PSO algorithm can be summarized as follows [16]:

Step 1: The positions, $x^i(k)$ and velocities, $v^i(k)$ of the initial population of particles are randomly generated for the i^{th} particle at time k. Where i is the current particle number in the swam, $i \in \{1, 2, ..., P\}$ and P is the swarm size. For each particle, define a small social network by selecting n neighbours or friends (star topology is used for selecting neighbours).

Step 2: Evaluate the fitness function values for all particles and save:

- Particle with the best position value $p^i(k)$(local best) over the current swarm.
- Neighbourhood optimum value $p_n^i(k)$ (neighbour best) found by the neighbours of each particle.

Step 3: Updates the global best position $p^g(k)$ for the current and all the previous swarm moves.

Step 4: Update velocities of all particles at time k +1 as follows:

$$v^i(k+1) = w.v^i(k) + c_l.(p^i(k) - x^i(k)) + c_g.(p^g(k) - x^i(k)) + c_n.(p_n^i(k) - x^i(k))$$

Where, w is inertia factor, c_l is the cognitive learning factor, c_g is the cooperative learning factor and c_n is the social learning factor.

Step 5: The position of each particle is updated using its velocity vector at time k +1 as:

$$x^i(k+1) = x^i(k) + v^i(k+1)$$

Step 6: Repeat from step 2 until convergence criterion is met.

4. The Proposed Tuning Method for Takagi-Sugeno Fuzzy Systems

In this work, particle swarm optimization based approach is used to optimize the parameters of fuzzy systems from the training data. In PSO fuzzy based approach, each particle in the swarm is considered to represent a fuzzy classification system. Then, a fitness function is evaluated to guide the search and select an appropriate fuzzy classification system such that the number of incorrectly classified patterns is minimized.

A. The original Takagi-Sugeno system

This fuzzy system is implemented in Matlab Fuzzy Logic Toolbox as genfis3(sugeno).Given separate sets of input and output data, genfis3 generates a fuzzy inference system (FIS) by extracting a set of rules that models the data behavior. FCM clustering algorithm is used to partition the input space into number of clusters. Number of rules and number of Gaussian membership functions for each input is the same as the number of clusters. FCM is used to optimize parameters of the input membership functions while least square is used to optimize the consequence parameters of the Takagi-Sugeno system [18], [22]. The steps for the original Takagi-Sugeno fuzzy system are as follows:

Step 1: Divide the database into training data set and testing data set.

Step 2: Apply genfis3 function to generate a fuzzy system using training data set.

Step 3: Apply testing data on the fuzzy system and record classification accuracy.

B. Stage 1 for tuning Takagi-Sugenofuzzy system

This tuning stage is implemented because there is no assurance that FCM will produce the best or the optimum values for the input membership (mean and variance) and consequently the consequent parameters. So, in this stage, starting with the generated fuzzy system from genfis3, PSO is used to optimize, input membership functions (mean and variance) and consequent parameters. The steps for tuning of Takagi-Sugeno fuzzy system are as follows:

Step 1: Divide the database into training data set and testing data set.

Step 2: Create a population of N particles (systems). Each particle consists of:

- Parameters of input membership functions (mean and variance), where each input will be represented by number of membership functions equal to number of rules
- Consequent parameters of Takagi-Sugeno fuzzy model according to number of rules.

Step 3: For each particle, construct Takagi-Sugeno fuzzy system, apply training data set and calculate the corresponding fitness value (classification accuracy).

Step 4: Apply PSO algorithm and produce a new population.

Step 5: Repeat steps 2 to 4 until a solution results in perfect classification of all training samples or until a maximum number of fitness function evaluation is reached.

Step 6: Save the new tuned fuzzy system (to be used in the stage 2).

Step 7: Apply testing data on the selected fuzzy system and record classification accuracy.

To identify the size of each particle, assume there are Ni inputs and one output. Assume there are Nr rules. Each input will be represented by number of membership functions equal to assigned number of rules. Each membership function is represented by two variables (mean and variance). Then the numbers of premise parameters are2 $Ni\ Nr$. For the consequent parameters we have $(Ni+1)Nr$ variables. The total number of variables that need to be optimized (particle size) will be $(3Ni+1)Nr$.

C. Stage 2 for tuning Takagi-Sugeno fuzzy system

For extra improving the tuning of Takagi-Sugeno fuzzy system a second stage tuning is implemented. In this stage, starting with the tuned fuzzy system from stage 1, PSO is used to optimize external weighting factor for each rule and scaling universes of discourse for inputs and outputs variables. Scaling universes of discourse as shown in Fig. 3 have different effect for both inputs and outputs variables. For input scaling, $g1$, $g2$, …,gn, the choice of a scaling gain g results in scaling the horizontal axis of the membership functions by $1/g$. If $g< 1$, the membership functions are uniformly spread, while if $g> 1$, the membership functions are uniformly contracted. For output scaling, h, the choice of a scaling gain h results in scaling the horizontal axis of the membership functions by h. If $h< 1$, there is the effect of contracting the output membership functions, while if $h> 1$, there is the effect of spreading out the output membership functions. Since for Takagi-Sugeno fuzzy system, there is no membership function in the output where the output are a weighted average of a linear equations produced from different rules, so the effect in this case is just multiplying the output of the system by a factor h.

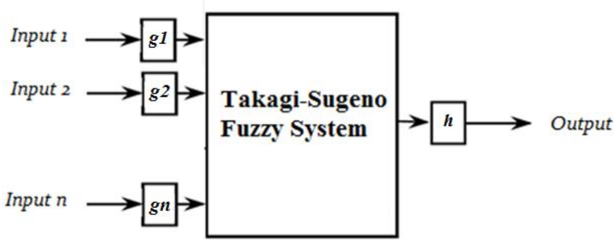

Figure 3. Scaling universes of discourse for inputs and outputs

Starting with the tuned fuzzy system from stage 1, the steps for tuning of Takagi-Sugeno fuzzy system are follows:

Step 1: Divide the database into training data set and testing data set.

Step 2: Create a population of N particles (systems). Each particle consists of:

- External weighting factor each rule
- Scaling universes of discourse for inputs and outputs

Step 3: For each particle, construct Takagi-Sugeno fuzzy system, apply training data set and calculate the corresponding fitness value (classification accuracy).

Step 4: Apply PSO algorithm and produce a new population.

Step 5: Repeat steps 2 to 4 until a solution results in perfect classification of all training samples or until a maximum number of fitness function evaluation is reached.

Step 6: Apply testing data on the selected fuzzy system and record classification accuracy.

To identify the size of each particle, assume there are Ni inputs and one output. Assume there are Nr rules. The total number of variables that need to be optimized (particle size) will be (Ni+Nr +1).

D. Combining Stage 1 and Stage 2 for only one stage tuning of Takagi-Sugeno fuzzy system

In this case, instead of performing the tuning process in two stages, only one stage is used. Starting with the original generated fuzzy system from genfis3, PSO is used to optimize:

- Input membership functions (mean and variance)
- Consequent parameters
- External weighting factor each rule
- Scaling universes of discourse for inputs and outputs variables

Same procedure is used as in stage 1 with increase in the particle size so it includes all the above parameters. The total number of variables that need to be optimized (particle size) will be (3Ni+ 1)Nr+ (Ni+Nr +1).

5. Experimental Results

The performance of the proposed tuning method for Takagi-Sugeno fuzzy system is studied using five widely used real-world data sets from UCI Machine Learning Repository (Iris, Phoneme, Diabetes, Heart and Wine) [23]. The details for these five data sets are shown in Table 1, it includes number of inputs, number of classes and total number of pattern for each data set. Each data set is divided into training data set and testing data set.

Table 1. Detail of each data set

Data Set	Number of inputs	Number of classes	Number of patterns
Iris	4	3	150
Phoneme	5	2	5404
Diabetes	8	2	768
Heart	13	2	303
Wine	13	3	178

Based on testing the original Takagi-Sugeno fuzzy system (genfis3) and the tuned Takagi-Sugeno fuzzy systems on testing data sets, Table 2, shows percentage of correct classification for each class and overall classification accuracy. Table 3shows summary of overall percentage of correct classification for the Takagi-Sugeno fuzzy system before and after tuning.

From Table 3, it can be seen that, there is about 4.04 % improvement in the average of overall percent of correct classification after the first stage of tuning, followed by 1.8 % in the second stage of tuning with a total of 5.84 % improvement in the two stage tuning system. For a one stage tuning that combine stage 1 and stage 2, the percentage improvement is about 5.3 %.

Fig. 4 shows a bar plot to compare overall classification accuracy for Takagi-Sugeno fuzzy system before and after stage 1 and stage 2 tuning. Fig. 5 shows a bar plot to compare overall classification accuracy for Takagi-Sugeno fuzzy system before and after combining stage 1 and 2 into one stage tuning. Fig. 6 shows a bar plot to compare overall classification accuracy for Takagi-Sugeno fuzzy system after tuning with two stages or combining stage 1 and 2 into one stage.

Table 2. Percentage of correct classification for each class and overall percentage for Takagi-Sugeno fuzzy system before and after tuning

		Original (Before Tuning)	After first stage Tuning	After second stage Tuning	Combining Stage 1 and 2 into one stage
Iris	Class 1	100.00	96.55	96.55	96.55
	Class 2	73.08	84.62	88.46	92.31
	Class 3	85.00	85.00	85.00	85.00
	Overall	**86.03**	**88.72**	**90.00**	**91.29**
Phoneme	Class 1	88.21	79.41	79.31	78.00
	Class 2	68.85	84.62	85.62	86.63
	Overall	**78.53**	**82.01**	**82.47**	**82.32**
Diabetes	Class 1	87.55	75.10	80.50	78.84
	Class 2	44.06	65.04	62.94	63.64
	Overall	**65.80**	**70.07**	**71.72**	**71.24**
Heart	Class 1	61.63	70.59	80.00	85.88
	Class 2	65.67	75.76	69.70	63.64
	Overall	**63.65**	**73.17**	**74.85**	**74.76**
Wine	Class 1	89.66	89.66	96.55	96.55
	Class 2	90.24	85.36	90.24	80.49
	Class 3	94.44	100.00	100.00	100.00
	Overall	**91.45**	**91.67**	**95.60**	**92.35**

Table 3. Summary of overall percentage of correct classification for Takagi-Sugeno fuzzy system before and after tuning

	Original (Before Tuning)	After first stage Tuning	After second stage Tuning	Combining Stage 1 and 2 into one stage
Iris	86.03	88.72	90.00	91.29
Phoneme	78.53	82.01	82.47	82.32
Diabetes	65.80	70.07	71.72	71.24
Heart	63.65	73.17	74.85	74.76
Wine	91.45	91.67	95.60	92.35
OVERALL AVERAGE	77.09	81.13	82.93	82.39

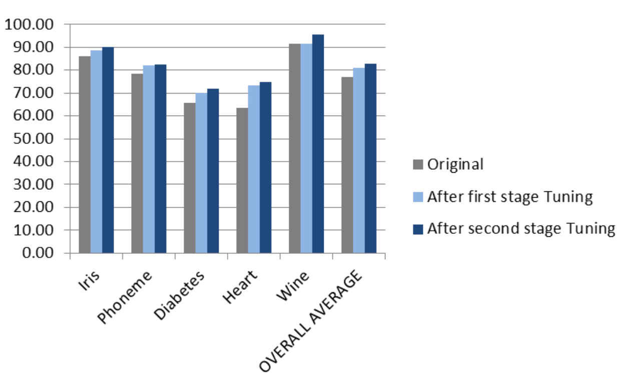

Figure 4. Comparison of overall classification accuracy for Takagi-Sugeno fuzzy system before and after stage 1 and stage 2 tuning.

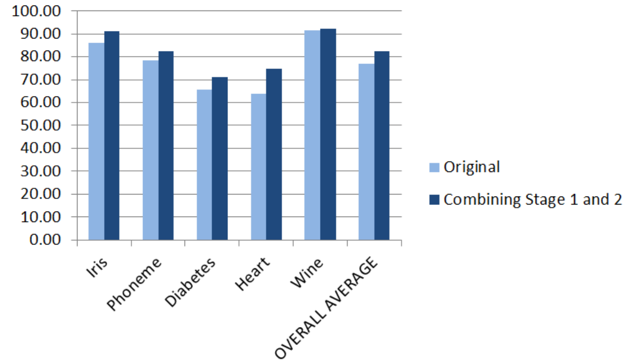

Figure 5. Comparison of overall classification accuracy for Takagi-Sugeno fuzzy system before and after combining stage 1 and stage 2 in one stage tuning.

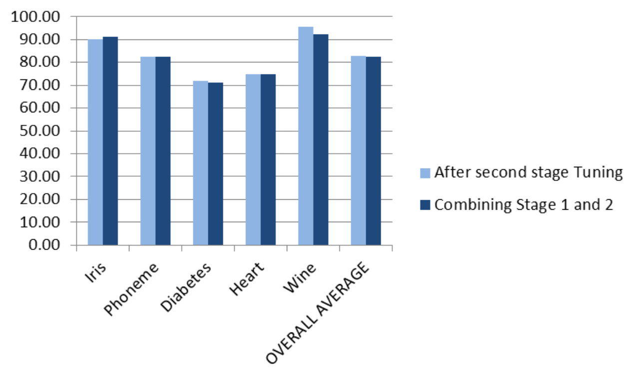

Figure 6. Comparison of overall classification accuracy for Takagi-Sugeno fuzzy system after tuning with two stages or combining stage 1 and stage 2 in one stage tuning.

From Table 3, Figs. 4, 5 and 6, we can conclude that:

- For two tuning stage system:
  - The first tuning stage has a significant improvement in the average of overall percent of correct classification (4.04 %)
  - The second tuning stage add more improvement in the average of overall percent of correct classification (1.8 %)
  - A total of (5.84 %) improvement in the average of overall percent of correct classification after the two stage tuning system.
- For combining stage 1 and stage 2 into one tuning stage system:
  - A total of (5.3 %) improvement in the average of overall percent of correct classification after the two stage tuning system.
- As a comparison between the two stage tuning system and the one stage tuning system, it can be seen in three of the data sets the performance is very close (Phoneme, Diabetes and Heart). In Iris data set, one stage tuning perform better than two stage tuning, while in Wine data set, two stage tuning perform better than one stage tuning. So, we cannot draw a solid conclusion about which one is better in

terms of classification accuracy. But from point of view of the complexity of the system, solving one optimization problem with a little increase in the particle size is easier than performing two optimization stages.

- From these results, we can see the consistency of getting higher overall classification accuracy for tuning Takagi-Sugeno fuzzy systems with both two stage tuning and one stage tuning.

6. CONCLUSION

A method for tuning a commonly used Takagi-Sugeno fuzzy system found in Matlab control Toolbox as genfis3 is proposed. In this genfis3 fuzzy system, the input space is partitioned using FCM clustering algorithm and the consequent parameters are optimized using least square. A two stage tuning is proposed to improve the accuracy of genfis3(sugeno) fuzzy system. In the first stage, PSO is used to optimize parameters of input membership functions (mean and variance) and consequent parameters. Extra tuning is done in the second stage where PSO is used to optimize a weighting factor for each rules and a scaling factor for inputs and outputs. A one stage tuning is also investigated by optimizing the same parameters is the two stage tuning but in only one stage with a bigger particle size. The performances of the proposed tuning methods are tested on real-world data sets. Experimental results show a consistency of getting higher overall classification accuracy using the proposed tuning methods.

REFERENCES

[1] Yager, R. R. and Filev, D. P. “Essentials of Fuzzy Modeling and Control,”John Wiley, New York, U.S.A. (1994).

[2] Jang, J. S., Sun, C. T. and Mizutani, E.,“Neuro-Fuzzy and Soft Computing,” Prentice Hall, New Jersey, U.S.A. (1997).

[3] C.T. Lin, C.S.G. Lee, “Neural network-based fuzzy logic control and decision system,”IEEE Trans. Computation. 40 (12) (1991) 1320–1336.

[4] M. Setnes, H. Roubos, “GA-Fuzzy modeling and classification: complexity and performance,”IEEE Trans. Fuzzy Systems. 8 (4) (2000) 509–522.

[5] Chia-Chong Chen “Design of PSO-based Fuzzy Classification Systems”, Tamkang Journal of Science and Engineering,” Vol. 9, No 1, (2006) pp. 63_70

[6] Jang, J. S., “ANFIS: Adaptive-Network-Based Fuzzy Inference systems,” IEEE Trans. on Systems, Man and Cybernetics, Vol. 23, (1993) pp. 665□685.

[7] Nozaki, K., Ishibuchi, H. and Tanaka, H., “Adaptive Fuzzy Rule-Based Classification Systems,” IEEE Trans. on Fuzzy Systems, Vol. 4, No. 3, Aug., (1996) pp. 238□250.

[8] Wong, C. C. and Chen, C. C., “A GA-Based Method for Constructing Fuzzy Systems Directly from Numerical Data,” IEEE Trans. on Systems, Man and Cybernetics-Part B: Cybernetics, Vol. 30, (2000) pp. 904□911.

[9] Wong, C. C. and Chen, C. C., “A Hybrid Clustering and Gradient Descent Approach for Fuzzy Modeling,” IEEE Trans. on Systems, Man and Cybernetics-Part B: Cybernetics, Vol. 29, (1999) pp. 686□693.

[10] Xie, X.L., and G.Beni, “A validity measure for fuzzy clustering,”IEEE trans. Pattern Anal. Mach. Intell. 3(8): 1991 pp. 841-846.

[11] S. L. Chiu, “Fuzzy Model Identification Based on Cluster Estimation,” Journal of Intelligent and Fuzzy Systems, 1994 pp. 267-278,.

[12] Emami, M.R. Turksen, Burhan. Goldenberg A.A.,. “Development of A Systematic Methodology of Fuzzy Logic Modeling,”IEEE transaction on Fuzzy Systems, 6(3) 1998.

[13] Salehfar, H. et al., “A systematic approach to linguistic fuzzy modeling based on input-output data,”IEEE Proceedings of the 2000 Winter Simulation Conference, 2000.

[14] Enwang Zhou, AlirezaKhotanzad, “Fuzzy classifier design using genetic algorithms”, Pattern Recognition 40 (2007) 3401 – 3414

[15] Elragal, H., “Mamdani and Takagi-Sugeno fuzzy classifier accuracy improvement using enhanced particle swarm optimization” Journal of Intelligent and Fuzzy Systems, (26) (2014) 2445–2457.

[16] Rody P.S. Oldenhuis, "Trajectory Optimization of a Mission to the Solar Bow Shock and Minor Planets," Thesis Report, Delft University of Technology, 2010.

[17] S.Y. Ho, H.M. Chen, S.J. Ho, T.K. Chen, “Design of accurate classifiers with a compact fuzzy-rule base using an evolutionary scatter partition of feature space,”IEEE Trans. Syst. Man Cybern. Part B 34 (2) (2004) 1031–1044.

[18] MATLAB Fuzzy Logic Toolbox™, User’s Guide, The MathWorks, Inc.

[19] Mamdani, E.H. and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," InternationalJournal of Man-Machine Studies, Vol. 7, No. 1, 1975 pp. 1-13.

[20] Sugeno, M., Industrial applications of fuzzy control, Elsevier Science Pub. Co., 1985.

[21] Kennedy, J. and Eberhart, R., “Particle Swarm Optimization,” Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia (1995), 1942-1945.

[22] Kevin M. Passino and Stephen Yurkovich, Fuzzy Control, Addison Wesley Longman, Menlo Park, CA, 1998

[23] A. Asuncion, D. J. Newman, UCI machine learning repository. http://archive.ics.uci.edu/ml/datasets.html (1987).

Hassan M. Elragal received the B.Sc., M.Sc. in Electrical Engineering from Alexandria University, Alexandria, Egypt, in 1991 and 1995 respectively. He received his Ph.D. degree in Electrical Engineering at Southern Methodist University, Dallas, Texas in July 2001. His research interest is in the area of signal processing, fuzzy logic, neural networks, genetic algorithm, particle swarm optimization and their applications.