Tuning of Type-2 Fuzzy Systems by Simulated Annealing to Predict Time Series

Majid Almaraashi, *Member, IAENG,* and Robert John

***Abstract*—In this paper, a combination of interval type-2 fuzzy system (IT2FS) models and simulated annealing are used to predict the Mackey-Glass time series by searching for the best configuration of the IT2FS. Simulated annealing is used to optimise the parameters of the antecedent and the consequent parts of the rules for a Mamdani model. Simulated annealing is combined with a method to reduce the computations associated with it using an adaptive step size. The results of the proposed methods are compared to results of a type-1 fuzzy system.**

***Index Terms*—Type-2-Fuzzy-Systems, Simulated-Annealing, Time-Series-Forecasting.**

I. Introduction

ONE of the features of fuzzy systems is that they can be hybridised with other methods such as neural networks, genetic algorithms and other search and optimisation approaches. These approaches have been proposed because generally fuzzy systems are difficult to learn from data[1]. Fuzzy systems are good at explaining how they reached a decision but can not automatically acquire the rules or membership functions to make a decision [2, p.2]. On the other hand, learning methods such as neural networks can not explain how a decision was reached but have a good learning capability [2, p.2]. Hybridisation overcomes the limitations of each method in an approach such as neuro-fuzzy systems or genetic fuzzy systems.

Soft Computing is a branch of computer science described as "a collection of methodologies aim to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness and low solution cost" [3]. In this research we are interested in the combination of fuzzy logic with simulated annealing to design a high-level performance and low-cost system. When designing a simple fuzzy system with few inputs, the experts may be able to use their knowledge to provide efficient rules but as the complexity of the system grows, the optimal rule base and membership functions become difficult to acquire. So, researchers often use some automated tuning and learning methods and evaluate their solutions by some criterion [4]. From an optimisation perspective, the task of finding a good knowledge base (KB) for a problem is equivalent to the task of parameterising the fuzzy knowledge base (KB) and equivalent to the task of finding the parameters values that are optimal based on the criteria of the problem design [1].

Simulated annealing has been used in some fuzzy systems to learn or tune the fuzzy system. For example, see [5] [6] [4]. In addition, the combination of simulated annealing and type-1 t Mamdani fuzzy systems exhibited a good performance in forecasting Mackey-Glass time series as shown in [7] and [8]. In this paper, a forecasting method is proposed using an interval type-2 Mamdani model optimised using simulated annealing. The Mackey-Glass time series is a well known bench mark which will be used here as an application of forecasting.

Manuscript received March 01, 2001; revised March 21, 2011.

The authors are with the Centre for Computational Intelligence, Department of Informatics, De Montfort University, Leicester LE1 9BH, U.K. (e-mail: almaraashi@dmu.ac.uk.

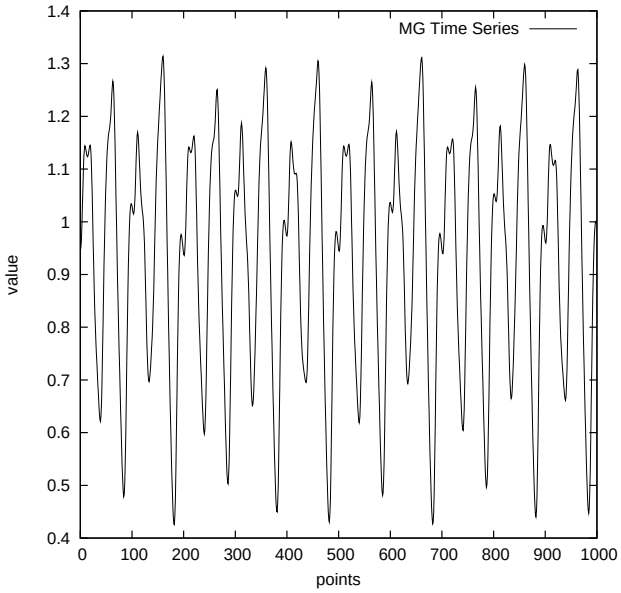


Fig. 1. Mackey-Glass time series when Tau=17

The rest of the paper starts by describing the data sets in section II followed by a review of fuzzy systems (section III) and simulated annealing (section IV). The methodology and the results of this paper are detailed in section V where the conclusion is drawn in section VI.

II. Mackey-Glass Time Series

The Mackey-Glass Time Series is a chaotic time series proposed by Mackey and Glass [9]. It is obtained from this non-linear equation :

$$\frac{dx(t)}{dt} = \frac{a * x(t-\tau)}{1 + x^n(t-\tau)} - b * x(t)$$

Where a, b and n are constant real numbers and t is the current time where τ is the difference between the current time and the previous time $t - \tau$. To obtain the simulated data, the equation can be discretised using the Fourth-Order Runge-Kutta method. In the case where $\tau > 17$, it is known to exhibit chaos and has become one of the benchmark problems in soft computing [10, p.116].

III. Type-2 Fuzzy Systems

Type-1 fuzzy logic has been successful in many applications, However, the type-1 approach has problems when

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

faced with dynamical environments that have some kinds of uncertainties. These uncertainties exist in the majority of real world applications and can be a result of uncertainty in inputs, uncertainty in outputs, uncertainty that is related to the linguistic differences, uncertainty caused by the conditions change in the operation and uncertainty associated with the noisy data when training the FLC [11]. All these uncertainties translate into uncertainties about fuzzy sets membership functions [11]. Type-1 fuzzy Logic can not fully handle these uncertainties because type-1 fuzzy logic membership functions are totally precise which means that all kinds of uncertainties will disappear as soon as type-1 fuzzy set membership function has been used [12]. The existence of uncertainties in the majority of real world applications makes the use of type-1 fuzzy logic inappropriate in many cases especially with problems related to inefficiency of performance in fuzzy logic control [12]. Also, interval type-2 fuzzy sets can be used to reduce computational expenses. Type-2 fuzzy systems have, potentially, many advantages over type-1 fuzzy systems including the ability to handle numerical and linguistic uncertainties, allowing for a smooth control surface and response and giving more freedom than type-1 fuzzy sets [12]. Since last decade, type-2 fuzzy logic is a growing research topic with much evidence of successful applications [13].

A type-2 fuzzy set [11], denoted $\tilde{A}$, is characterized by a type-2 membership function $\mu_{\tilde{A}}(x,u)$ where $x \in X$ and $u \in J_x \subseteq [0,1]$. For example :

$$\tilde{A} = ((x,u), \mu_{\tilde{A}}(x,u)) \mid \forall x \in X, \forall u \in J_x \subseteq [0,1]$$

where $0 \leq \mu_{\tilde{A}}(x,u) \leq 1$.
Set $\tilde{A}$ also can be expressed as:

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x,u)/(x,u), J_x \in [0,1]$$

where $\int$ denotes union. When universe of discourse is discrete, Set $\tilde{A}$ is described as :

$$\tilde{A} = \sum_{x \in X} \sum_{u \in J_x} \mu_{\tilde{A}}(x,u)/(x,u), J_x \in [0,1]$$

When all the secondary grades $\mu_{\tilde{A}}(x,u)$ equal 1 then $\tilde{A}$ is an interval type-2 fuzzy set. Interval type-2 fuzzy sets are easier to compute with than general type-2 fuzzy sets. See Figure 2 for an example of an interval type-2 fuzzy set called "About 10". The ease of computation and representation of interval type-2 fuzzy sets is the main reason for their wide usage in real world applications.

Type-2 fuzzy logic systems are rule based systems that are similar to type-1 fuzzy logic systems in terms of the structure and components but type-2 FLS has an extra output process component which is called the type-reducer before defuzzification. The type-reducer reduces output type-2 fuzzy sets to type-1 fuzzy sets then the defuzzifier reduces it to a crisp output. The components of a type-2 fuzzy system are:

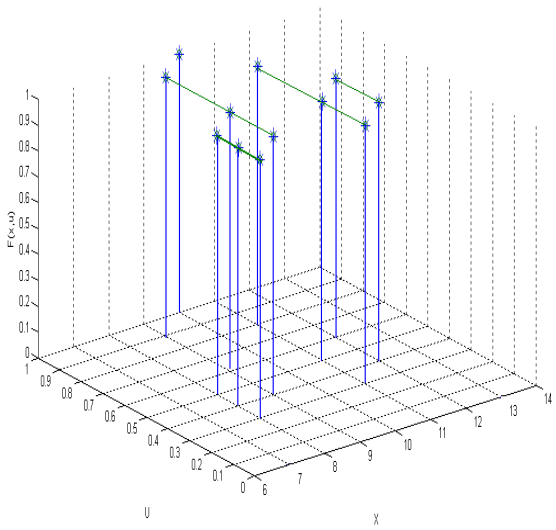
- Fuzzifier :
  Fuzzifier maps crisp inputs into type-2 fuzzy sets by evaluating the crisp inputs $x = (x1, x2, \ldots, xn)$ based on the antecedents part of the rules and assigns each crisp input to its type-2 fuzzy set $\tilde{A}(x)$ with its membership grade in each type-2 fuzzy set.
- Rules:
  A fuzzy rule is a conditional statements in the form of IF-THEN where it contains two parts, the IF part called the antecedent part and the Then part called the consequent part.
- Inference Engine:
  Inference Engine maps input type-2 fuzzy sets into output type-2 fuzzy sets by applying the consequent part where this process of mapping from the antecedent part into the consequent part is interpreted as a type-2 fuzzy implication which needs computations of union and intersection of type-2 fuzzy sets and a composition of type-2 relations by using the extended sup-star composition for type-2 set relations. The inference engine in a Mamdani system maps the input fuzzy sets into the output fuzzy sets then the defuzzifier converts them to a crisp output. The rules in Mamdani model have fuzzy sets in both the antecedent part and the consequent part. For example, the *i*th rule in a Mamdani rule base can be described as follows:

  $$R^i : \text{ IF } x_1 \text{ is } \tilde{A}_1^i \text{ and } x_2 \text{ is } \tilde{A}_2^i \ldots \text{ and } x_p \text{ is } \tilde{A}_p^i$$

  $$\text{THEN } y \text{ is } \tilde{B}^i$$

- Output Processor:
  There are two stages in the output process:
  - Type-Reducer:
    Type-reducer reduces type-2 fuzzy sets that have been produced by the inference engine to type-1 fuzzy sets by performing a centroid calculation [10].T
  - Defuzzifier:
    Defuzzifier maps the reduced output type-1 fuzzy sets that have been reduced by type-reducer into crisp values exactly as the case of defuzzification in type-1 fuzzy logic systems.

Fig. 2. Interval type-2 fuzzy set "About 10".



ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

IV. Simulated Annealing Algorithm

The concept of annealing in the optimisation field was introduced by Kirkpatrick *et al* in 1982 [14]. Simulated annealing uses the Metropolis algorithm to imitate metal annealing in metallurgy where heating and controlled cooling of materials is used to reshape metals by increasing the temperature to the maximum values until the solids almost melted then decreasing the temperature carefully until the particles are arranged and the system energy becomes minimal. Simulated annealing is a powerful randomised local search algorithm that has shown great success in finding optimal or nearly optimal solutions of combinatorial problems [15]. SA is particularly useful in high dimensionality problems as it scales so well with the increase of variable numbers which allows SA to be a good candidate for fuzzy systems optimisation [16]. Many comparative studies for solving problems such as job shop scheduling and travelling sales man suggest that SA can outperform most other local search algorithms in terms of effectiveness [15]. In general, SA can find good solutions for a wide range of problems but normally with the cost of high running times [15].

We now define the simulated annealing algorithm. Let s be the current state and $N(s)$ be a neighbourhood of s that includes alternative states. By selecting one state $s' \in N(s)$ and computing the difference between the current state cost and the selected state energy as $D = f(s') - f(s)$, s' is chosen as the current state based on *Metropolis criterion* in two cases:

- If $D < 0$ means the new state has a smaller cost, then s' is chosen as the current state as downhills always accepted.
- If $D > 0$ and the probability of accepting s' is larger than a random value Rnd such that $e^{-D/T} > Rnd$ then s' is chosen as the current state where T is a control parameter known as *Temperature* which is gradually decreased during the search process making the algorithm more greedy as the probability of accepting uphill moves decreasing over time. Rnd is a randomly generated number where $0 < Rnd < 1$. Accepting uphill moves is important for the algorithm to avoid being stuck in a local minima.

In the last case where $D > 0$ and the probability is lower than the random value $e^{-d/T} <= Rnd$, no moves are accepted and the current state s continues to be the current solution. In the original proposed version of simulated annealing by Kirkpatrick, Gelatt and Vecchi, the probability of accepting s' equals 1 when $f(s') <= f(s)$. When starting with a large cooling parameter, large deteriorations are accepted. Then, as the temperature decreases, only small deteriorations are accepted until the temperature approaches zero when no deteriorations are accepted. Therefore, adequate temperature scheduling is important to optimise the search. Simulated annealing can be implemented to find the optimal annealing by allowing infinite number of transitions or can be implemented to find a closest possible optimal value within a finite time where the cooling schedule is specified by four components [15]:

1) Initial value of temperature.
2) A function to decrease temperature value gradually.
3) A final temperature value.
4) The length of each homogeneous Markov chains. Markov chains is a sequence of trials where the probability of the trial outcome depends on the previous trial outcome only and called homogeneous when the transition probabilities do not not depend on the trial number [17, p.98].

The choice of good SA parameters is important for the success of SA. For example, small initial temperatures could cause the algorithm to get stuck in local minimas as the first stages of the search supposed to aim for exploration of regions while large ones could cause unneeded excessive running times. In addition, an appropriate cooling schedule is important for the same reason as fast cooling causing getting stuck in local minima and slow cooling causing the algorithm to be very slow. In the fuzzy system optimisation literature, few researchers used adaptive step sizes such as [18] while most of the approaches reported were using small fixed step sizes [19]. One of the methods used to determine the initial step size was proposed by [20] which starts by using large step sizes and decrease them gradually while one of the methods to determine the initial temperature value was proposed by [21] is to choose the initial temperature value within the standard deviation of the mean cost. When using finite Markov chains to model the simulated annealing mathematically, the temperature is reduced once for each Markov chain while the length of each chain should be related to the size of the neighbourhood in the problem [15].

V. Methodology and Results

The experiment can be divided into three steps : generating time series, constructing the initial fuzzy system and optimising the fuzzy system parameters. Firstly, the time series is generated with the following parameters : $alpha = 0.2$, $Beta = 0.1$, $\tau = 17$. The input-output samples are extracted in the form $x(t-18), x(t-12), x(t-6)$ and $x(t)$ where $t = 118$ to $t = 1117$ using a step size of 6. Samples of the generated data are depicted in figures 1. Then the generated data are divided into 500 data points for training and the remaining 500 data points for testing. Using a step size of 6, the input values to the fuzzy system are the previous data points $x(t-18), x(t-12), x(t-12)$ and $x(t)$ while the output from the fuzzy system is the predicted value $x(t+6)$. Four initial input values $x(114)$ and $x(115)$ and $x(116)$ and $x(117)$ are used to predict the first four training outputs.

Two fuzzy systems have been chosen: type-1 and type-2 FLS. The fuzzy system has four-inputs and one-output. The fuzzy model consists of four input fuzzy sets $A_1, A_2 A_3$ and A_4 and one independent output fuzzy set B_i for each rule. Gaussian membership functions were chosen to define the fuzzy sets. Any other types of membership functions can be chosen but we are interested in reducing the computation time as gaussain type has only two parameters instead of three in triangular type or four as the case in trapezoidal type. The parameters of the Gaussian membership functions are the mean m and the standard deviation σ in type-1 FLS. For type-2 FLS, each fuzzy set is represented by two means and one standard deviation. All the means and standard deviations are initialised for all the input fuzzy sets by dividing the each input space into two fuzzy sets and enabling enough overlapping between them. The

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

fuzzification process is based on the product t-norm while the centre-of-sets has been chosen for defuzzification. Hence, this is the same as height defuzzification method because all sets are convex, symmetric and normal [10]. The training procedure aims to optimise the parameters of the antecedent parts and the consequent parts of the fuzzy system rules. Then, the found parameters are used to predict the next 500 testing data points. By using four inputs and two fuzzy sets for each input, we end up with 16 rules and 8 input fuzzy sets representing all possible combinations of input values with input fuzzy sets. While each rule in Mamdani is linked with 1 independent output set. The total number of optimised parameters for type-1 FLS is $8 + 8 + (16 * 2) = 48$ while it is 72 parameters in type-2 FLS where $(8 * 2) + 8 = 24$ parameters in the antecedent part and $(16 * 2) + 16 = 48$ in the consequent part.

The optimisation process is done using simulated annealing that searches for the best configuration of the parameters by trying to modify one parameter each time and evaluate the cost of the new state which is measured by Root Mean Square Error (RMSE). The simulated algorithm is initialised with a temperature that equals to the standard deviation of mean of RMSE's for 500 runs for the 500 training points. The cooling schedule is based on a cooling rate of 0.9 updated for each Markov chain. Each Markov chains has a length related to the number of variables in the search space. The search ends after a finite number of Markov chains namely 400 Markov chains. The new states for a current state are chosen from neighbouring states randomly as following:

- Adding a number to one of the antecedent parameters or the consequent parameters. This value is related to the maximum and minimum value for each input space and $=$ max-min$/50$ for the first iteration.
- Adapting the step size for each input at each Markov chain by this scaling function proposed by [20]:

$$s_n = \frac{2s_0}{1 + \exp^{\frac{\beta n}{n_{max}}}}$$

Where : s_0 Initial step size. n Current iteration. s_n Step size at current iteration. n_{max} Maximum number of iteration (Markov chains). β Adaptation constant. a value of 7 has been chosen.

The adaptation of the step size is proposed to reduce the computation as the fixed small step sizes needs long time. After that, the new state is evaluated by examining the 500 data points outputs. The experiment has been carried out 15 times and the average and the minimum RMSE of the testing data results has been calculated. It is shown from the results in Table I that type-2 systems outperforms type-1 system with an average RMSE of 0.01383 and a minimum of 0.00898 compared to an average of 0.02501 and a minimum of 0.01335 in T1FLS. This results agree with some previous findings about the ability of the type-2 fuzzy sets to handle uncertain data better than type-1 fuzzy sets [11]. Note that the fuzzy system model and structure has a great impact on the performance of the fuzzy system and normally it is chosen heuristically. For example, TSK might give better results than some Mamdani models for this problem [7]. In this paper, we chose one model which is Mamdani model with dependent fuzzy sets for each input as our aim is to unveil the potential for SA to be a good candidate for tuning fuzzy systems rather than finding the best fuzzy system components. Comparing our SA-T2FLS results with others as Table II shows, We see that our result of (RMSE)$= 0.0089$ is one of the closest result to the best result which was obtained by ANFIS despite the general structure that our method has using a simple combination of a general search algorithm and a fuzzy system compared to the more complicated structures for ANFIS and GEFREX. We believe that by investigating more about the best formalisation of SA structure and that suit type-2 fuzzy systems, we could have improved results.

TABLE I
THE FORECASTING RESULTS FOR MACKEY-GLASS TIME SERIES

Experiment	System	$RMSE_{avg}$	$RMSE_{min}$
Training Results	T1	0.02536	0.01334
Training Results	T2	0.0139	0.009
Testing Results	T1	0.02501	0.01335
Testing Results	T2	0.01383	0.00898

TABLE II
RESULTS COMPARISON FOR PREDICTING MACKEY-GLASS TIME SERIES

Method	RMSE
Wang and Mendel [22]	0.08
Lin and Lin/FALCON-ART [22]	0.04
Kim and Kim/ GA Ensemble [23]	0.026
Juang and Lin/SONFIN [24]	0.018
Lo and Yang/TSK model [25]	0.0161
Russo / GEFREX (GA + NN) [26]	0.0061
Kukolj / Fuzzy cluster + LS + WRLS [27]	0.0061
Almaraashi/ SA-TSK	0.0037
Jang / ANFIS [18]	0.0015
This Model	0.0089

VI. CONCLUSION

Simulated annealing is used to optimise a Mamdani fuzzy system by searching for the best parameters of the antecedent and the consequent parts of the fuzzy system to predict a well known time series. Both type-1 and type-2 FLS have been compared in their ability to handle uncertainty. The result shows the ability for simulated annealing to be a good candidate for type-2 systems to handle uncertain data. The paper also describes a method to reduce the computations associated with simulated annealing using an adaptive step size.

REFERENCES

[1] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, "Ten years of genetic fuzzy systems: current framework and new trends," *Fuzzy Sets and Systems*, vol. 141, no. 1, pp. 5–32, 2004.
[2] S. Goonatilake and S. Khebbal, *Intelligent Hybrid Systems*. John Wiley & Sons, 1995.
[3] L. Zadeh, "Soft computing and fuzzy logic," *IEEE Software*, vol. 11, no. 6, pp. 48–56, 1994.
[4] G. Liu and W. Yang, "Learning and tuning of fuzzy membership functions by simulated annealing algorithm," in *Circuits and Systems, 2000. IEEE APCCAS 2000. The 2000 IEEE Asia-Pacific Conference on*, 2000, pp. 367–370.
[5] E. Huyghe and Y. Hamam, "Simulated annealing for fuzzy controller optimization: Principles and applications," in *IEEE International Conference On Systems and Cybernetics*, vol. 5. Institute of Electrical Engineers INC (IEEE), 1995, pp. 4509–4514.
[6] J. Garibaldi and E. Ifeachor, "Application of simulated annealing fuzzy model tuning to umbilical cord acid-base interpretation," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 1, pp. 72–84, 1999.
[7] M. Almaraashi, R. John, S. Coupland, and A. Hopgood, "Time series forecasting using a tsk fuzzy system tuned with simulated annealing," in *Proceedings of FUZZ-IEEE2010 World Congress on Computational Intelligence*, Barcelona, Spain, July 2010.

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[8] M. Almaraashi and R. John, "Tuning fuzzy systems by simulated annealing to predict time series with added noise," in *Proceedings of UKCI*, Essex, UK, September 2010.
[9] M. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.
[10] J. Mendel, *Uncertain rule-based fuzzy logic systems: introduction and new directions*. Prentice Hall, 2001.
[11] J. Mendel and R. John, "Type-2 fuzzy sets made simple," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 117–127, 2002.
[12] H. Hagras, "Type-2 flcs: A new generation of fuzzy controllers," vol. 2, no. 1, pp. 30–43, Feb. 2007.
[13] R. John and S. Coupland, "Type-2 fuzzy logic: A historical view," *Computational Intelligence Magazine, IEEE*, vol. 2, no. 1, pp. 57 –62, 2007.
[14] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing, 1983," *Science*, vol. 220, pp. 671–680, 1983.
[15] E. H. L. Aarts and H. M. M. T. Eikelder, *Handbook of applied optimization*. Oxford University Press, 2002, ch. Simulated Annealing, pp. 209–220.
[16] L. Drack and H. Zadeh, "Soft computing in engineering design optimisation," *Journal of Intelligent and Fuzzy Systems*, vol. 17, no. 4, pp. 353–365, 2006.
[17] E. Aarts and J. Lenstra, *Local search in combinatorial optimization*. Princeton Univ Pr, 2003.
[18] J. Jang, "ANFIS: Adaptive-network-based fuzzy inference systems," 1993.
[19] P. Dadone, "Design optimization of fuzzy logic systems," Ph.D. dissertation, Citeseer, 2001.
[20] L. Nolle, A. Goodyear, A. Hopgood, P. Picton, and N. Braithwaite, "On step width adaptation in simulated annealing for continuous parameter optimisation," *Computational Intelligence. Theory and Applications*, pp. 589–598, 2001.
[21] S. White, "Concepts of scale in simulated annealing," in *American Institute of Physics Conference Series*, vol. 122, 1984, pp. 261–270.
[22] C.-J. Lin and C.-T. Lin, "An art-based fuzzy adaptive learning control network," *Fuzzy Systems, IEEE Transactions on*, vol. 5, no. 4, pp. 477 –496, Nov. 1997.
[23] D. Kim and C. Kim, "Forecasting time series with genetic fuzzy predictor ensemble," *Fuzzy Systems, IEEE Transactions on*, vol. 5, no. 4, pp. 523–535, 2002.
[24] C.-F. Juang and C.-T. Lin, "An online self-constructing neural fuzzy inference network and its applications," *Fuzzy Systems, IEEE Transactions on*, vol. 6, no. 1, pp. 12 –32, Feb. 1998.
[25] J.-C. Lo and C.-H. Yang, "A heuristic error-feedback learning algorithm for fuzzy modeling," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 29, no. 6, pp. 686 –691, Nov. 1999.
[26] M. Russo, "Genetic fuzzy learning," *Evolutionary Computation, IEEE Transactions on*, vol. 4, no. 3, pp. 259 –273, Sep. 2000.
[27] D. Kukolj, "Design of adaptive takagi-sugeno-kang fuzzy models," *Applied Soft Computing*, vol. 2, no. 2, pp. 89 – 103, 2002.

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)