

# Automatic Learning of General Type-2 Fuzzy Logic Systems using Simulated Annealing

Majid Almaraashi, *Member, IEEE*, Robert John, *Senior Member, IEEE*,  
Adrian Hopgood

**Abstract**—This paper reports on a new approach for automatic learning of general type-2 fuzzy logic systems (GT2FLSs) using simulated annealing (SA). The learning process in this work starts without an initial interval type-2 fuzzy system and has an objective to optimize all membership function parameters involved in the general type-2 fuzzy set in two stages. This is a novel methodology for learning GT2FLSs using the vertical-slices representation. The methodology used here is based on a proposed parameterization method presented in a previous work to ease the design of GT2FLSs. Two models of GT2FLSs have been applied using two different type-reduction techniques. The first technique is the sampling method, which is non-deterministic. The second technique is the vertical-slices centroid type-reduction (VSCTR), which is deterministic. Both models as well as an interval type-2 fuzzy logic system (IT2FLS) model have been applied to predict a Mackey-Glass time series. A comparison of the results of modeling these problems using the three models showed more accurate modeling for the GT2FLSs when using the VSCTR deterministic defuzzification method. It has also been shown that a GT2FLS with VSCTR defuzzification is more able to handle uncertainty than an IT2FLS, although the latter was faster.

## I. INTRODUCTION

Fuzzy logic systems (FLSs) are a well-known methodology that has been applied in a wide number of problems in different domains [1]. Type-2 FLSs are an extension of FLSs of type-1 that aim to extend the FLSs' ability to model uncertain information. Type-2 fuzzy logic is a research area with a history of successful implementation with applications [2]. However, almost all previous attempts to develop T2FLSs were dependent on IT2FLSs with few exceptions. These exceptions mainly depended on applying different representations of general type-2 sets and systems. Examples include geometric representation of T2FLS [3], alpha-planes representation [4], alpha cuts representation [5] and Z-slices representation [6][7]. The reason behind the wide usage of IT2FLSs is the simplicity of their computation compared to the general form. In fact, the complexity of general type-2 sets comes with an important feature in its third dimension. This is the extra degrees of freedom brought by the third dimension over interval type-2 fuzzy sets [8]. One of the criticisms of interval type-2 sets is that the uncertainty is spread equally across the footprint of uncertainty in which variations of the uncertainty can not be fully modeled [7]. In fact, an interval type-2 set is

a simplified form of a general type-2 set restricted using uniform secondary grades. The complication of GT2FLSs comes from three areas which are the excessive computation associated with meet operations and the type-reduction process as well as lack of rational basis for the choice of secondary memberships [9, p.302]. Regarding these issues, some good research attempts have been conducted in the first two areas such as proposed works conducted by [3][10][11][12][13][14]. The third area is still waiting for a solid basis for choosing secondary membership grades to bring more information into modeling.

One way to resolve the third area issue is to apply some learning methodologies to design a GT2FLS. An attempt to use survey data and device characteristics and zSlices representation to design general type-2 sets automatically has been described in [7]. In addition, an alpha-planes representation with a learning method has been proposed to tune the secondary membership functions in the GT2FLS [4]. Neural networks have been used with GT2FLSs in works like type-2 adaptive network-based fuzzy inference system [15] and general type-2 fuzzy neural networks (GT2FNN) [16]. However, no work has been proposed to exploit learning techniques to GT2FLSs using the vertical-slice representation other than the authors' work in [17] where a new practical parameterization mechanism of general type-2 set with a lower computational burden has been presented.

The simulated annealing algorithm has been proposed to optimize IT2FLS [18][19]. Also, it was recently used to design a GT2FLS using an optimized IT2FLS that has been used to initialize GT2FLS [17]. Simulated annealing was used in that work to optimize interval type-2 fuzzy sets parameters in the first step. Then, using the novel parametrization method mentioned above, simulated annealing has been used to optimize the secondary membership functions (SMFs) parameters in general type-2 fuzzy sets.

In this paper, we are interested in designing GT2FLSs using the simulated annealing algorithm by learning all GT2FLS parameters. The paper starts by a describing fuzzy sets and systems of type-2 in section II followed by ideas to design GT2FLSs in sections III and IV. An overview of the simulated annealing algorithm is found in section V followed by the methodology in section VI and the results in section VII. Finally, the conclusions are drawn in section VIII.

Majid Almaraashi with The University College in Aljamoum, Umm Al-Qura University, Makkah, Saudi Arabia (e-mail:msmaraashi@uqu.edu.sa). Robert John is with the Automated Scheduling Optimisation and Planning Group (ASAP) at the University of Nottingham. Adrian Hopgood is with Sheffield Business School at Sheffield Hallam University.

## II. TYPE-2 FUZZY SETS

A type-2 fuzzy set  $\tilde{A}$  is an extension of a type-1 fuzzy set defined using type-2 membership function  $\mu_{\tilde{A}}(x, u)$  where  $x \in X$  and  $u \in J_x \subseteq [0, 1]$  [8]. For instance:

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

Where  $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$ . It is known as an interval type-2 fuzzy set if all secondary membership grades  $\mu_{\tilde{A}}(x, u)$  equal 1. Otherwise, it is known as a general type-2 fuzzy set. Interval type-2 fuzzy sets are simpler and easier to compute than the general form which explains the reason behind the wide usage of the interval form. Interval type-2 fuzzy set can be represented in a 2D representation known as footprint of uncertainty (FOU) which is the union of all primary memberships. The FOU is defined fully by a lower and upper membership function. The most used operations on general type-2 fuzzy sets are union (join) and intersection (meet) that were defined using Zadeh extension principle [20][21]. A calculation methodology for meet and join operations when using normal and convex secondary memberships has been proposed by Karnik and Mendel [21]. In the case of non-normal secondary membership functions, an extension to this formula proposed by [3] can be used.

### A. Representations of Type-2 Fuzzy Set

Research on general type-2 fuzzy sets can be described using several representations including vertical-slice representation, wavy-slice representation, alpha-planes representation [4], Z-slices representation [6], geometric [3] and alpha cuts representations [5]. The first representation proposed was the vertical-slice representation [8] where fuzzy sets are represented with secondary sets in a vertical-slice manner where :

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid \forall x \in X\} \quad (2)$$

The vertical-slice representation is said to be more useful for computing purposes than wavy-slice representation which uses a huge number of embedded type-2 sets [22].

### B. Type-2 Fuzzy Logic Systems

A type-2 fuzzy logic system GT2FLS is a rule based system that uses type-2 fuzzy sets. It is similar to a type-1 fuzzy logic system but with an extra type-reduction stage before defuzzification. The Mamdani type-2 fuzzy logic system is composed of five components; fuzzifier, rules, inference engine, type-reducer and defuzzifier. The type-reduction stage aims to reduce output type-2 sets to type-1 sets followed by the defuzzification stage which reduces them to crisp outputs. The most expensive part in terms of computational cost in GT2FLSs is the type-reduction stage which represents a bottleneck stage.

## III. A PRACTICAL FOEM FOR DESIGNING GENERAL TYPE-2 FUZZY SYSTEMS

Previous work by the authors [17] has presented a practical parameterization mechanism for general type-2 sets that have

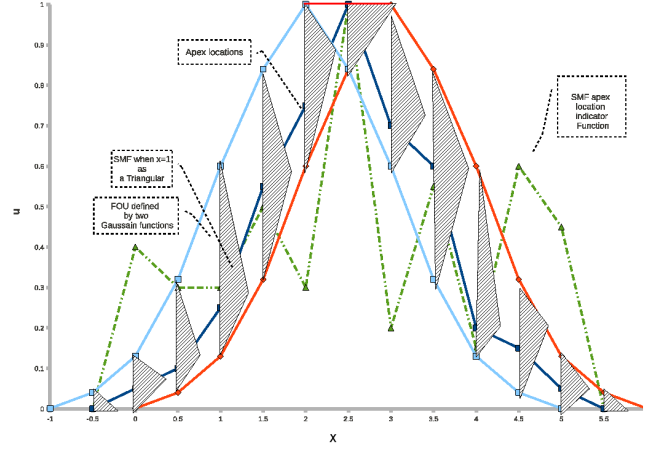


Fig. 1. General type-2 set determined by its FOU (by using two piecewise linear functions) and a triangular SMF (by using linear interpolation of apexes factors). [17]

a lower computational burden. The idea is to use asymmetric, convex and parameterized secondary membership functions to allow some practical operations to work. Therefore, they proposed a formula to normalize the FOU apex locations within  $FOU(x)$  in the primary domain for each  $x$  by defining a function for  $SMF_{apex}(x)$  as following [17]:

$$SMF_{apex}(x) = 1 / (FOU_{low}(x) + g(x) \times (FOU_{up}(x) - FOU_{low}(x))). \quad 0 \leq g \leq 1. \quad (3)$$

The parameter  $g$  is called the ‘‘apex factor’’ which is used for changing apexes locations within the boundaries. This is useful when applying learning or optimization processes to GT2FLSs. To show how this idea can be applied consider using a piecewise linear function to define  $g$  for all  $x$  values where  $k_1, k_2, \dots, k_n$  are ordered points in the primary domain  $x$  and  $g(k_1), g(k_2), \dots, g(k_n)$  are their apexes factors, then:

$$g(x) = \begin{cases} 0.5, & x < k_1 \\ g(k_i) + \frac{x-k_i}{k_{i+1}-x} \times (g(k_{i+1}) - g(k_i)), & k_i \leq x \leq k_{i+1} \\ 0.5, & x > k_n \end{cases} \quad (4)$$

In addition, the height of the apexes  $h(x)$  when non-normal  $SMF$  can be determined by another similar function such as:

$$h(x) = \begin{cases} 1, & x < k_1 \\ h(k_i) + \frac{x-k_i}{k_{i+1}-x} \times (h(k_{i+1}) - h(k_i)), & k_i \leq x \leq k_{i+1} \\ 1, & x > k_n \end{cases} \quad (5)$$

This idea is not identical to the principal function presented in [9, p.86] or the fuzzy truth numbers presented in [23] because the  $SMF$  is more flexible and can be non-normal for each  $x$  value. The  $(FOU_{lower}$  and  $FOU_{upper})$  points for each  $x$  can be defined using different functions similar to the way in interval type-2 sets. An example of this methodology is depicted in figure III while an example of learning the  $SMF$  is depicted in figure III.

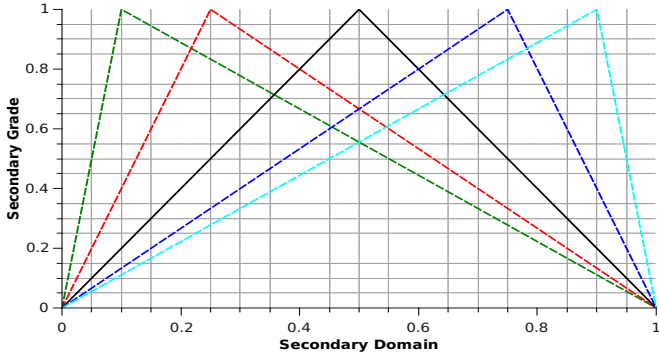


Fig. 2. An example of adapting the apex locations for learning of SMF. [17]

#### IV. DESIGNING GENERAL TYPE-2 FUZZY LOGIC SYSTEM

Using the proposed parametrization methodology presented above and first described in [17], we can design GT2FLS using the following two stage procedure:

- The first step is to design the FOU of the general type-2 set while fixing the secondary membership function. This is done by defining a FOU using any function used to define interval type-2 fuzzy sets. The lower and upper membership functions that bound the FOU in interval type-2 fuzzy sets can bound the FOU in general type-2 fuzzy sets. To get a good FOU, automated learning as well as expert opinions can be used exactly as in the case when designing IT2FLS.
- Learning the secondary membership functions of general type-2 sets is the second step. The SMF can be optimized by fixing the found FOU and tuning the apex location factors by a suitable value.

This two-stage method is based on the idea that the definition of the uncertainty boundaries (mainly modeled by primary memberships) should be defined before defining the secondary membership grades (uncertainty distribution) that will be assigned to each primary membership value. The other choice is to start by learning the primary and the secondary grades together but this seems excessive, computationally expensive and might be impossible to have a SMF without defining its FOU.

#### V. SIMULATED ANNEALING ALGORITHM

Simulated annealing is a well-known simple and general optimization algorithm that was proposed by Kirkpatrick *et al* in [24]. The simulated annealing algorithm is used to search for a global minima by imitating the atoms rearrangement in a cooling metal [25, p.6][26, p.170]. Simulated annealing searches for a global minima by picking one state from neighboring states and calculating the difference between the current and the new state costs. A new state is chosen as the current state only if the new state has a better or equal cost or if the probability of accepting the new state is greater

than or equal to a random value  $e^{-d/T} > Rnd$ .  $T$  refers to a control parameter called *Temperature* that is gradually decreased during the search process to gradually decrease the probability of accepting bad states. Accepting bad states with probability aims to escape from local minima. A suitable cooling schedule can be determined using adequate parameters such as: cooling rate, initial temperature value, termination temperature value and the length of homogeneous Markov Chains that are used to model iterations [27]. Simulated annealing does not require the existence of specific mathematical properties such as differentiability, which allows the use of different varieties of fuzzy structure components including non-differentiable t-norms and non-differentiable membership functions.

#### VI. METHODOLOGY

The whole experiment can be carried out in four stages : preparing data, initializing interval and general T2FLSs, learning *FOUs* parameters and finally learning *SMFs* parameters.

##### A. Data

1) *Mackey-Glass Time Series* : A well known chaotic time series presented in [28]. It is based on the following non-linear equation :

$$\frac{dx(t)}{dt} = \frac{a * x(t - \tau)}{1 + x^n(t - \tau)} - b * x(t) \quad (6)$$

Where  $a, b$  and  $n$  are constant numbers,  $t$  refers to the current time and  $\tau$  shows the delay between the current and the previous time  $t - \tau$ . The generation of the time series is carried out with the following values;  $a = 0.2$ ,  $b = 0.1$ ,  $\tau = 17$  and  $n = 10$ . Then, the previous equation is discretized using the Fourth-Order Runge-Kutta method with a time step of 0.1 and an initial value of  $x(0) = 1.2$  where  $x(t) = 0$  for  $t < 0$ . The chosen configuration is the same as [29]. The step size is 6 causing the input-output training and testing data samples to be extracted in the form  $x(t - 18), x(t - 12), x(t - 6)$  and  $x(t)$  where  $t = 118$  to  $t = 417$ . A sample of the generated time series is shown in figure 3. The target is to predict the output from the fuzzy system  $x(t + 6)$  while  $x(t - 18), x(t - 12), x(t - 6)$  and  $x(t)$  values represent fuzzy logic system inputs. Initial input values for  $x(114)$  and  $x(115)$  and  $x(116)$  and  $x(117)$  are used to initialize the training. 400 samples were generated where 200 samples were used in the training phase and the remaining 200 samples were used in the testing phase.

##### B. The initial and optimized interval type-2 fuzzy logic systems

The IT2FLS is composed of four rules that encapsulate a number of independent input fuzzy sets for each input and one independent output fuzzy set for each rule output. The number of rules was chosen with an intention to reduce the system's computations. Interval type-2 fuzzy sets are determined using Gaussian primary membership functions with uncertain means which represents each set using two means values  $m_1, m_2$  and one standard deviation  $\sigma$  as described in [9, p.91] as follows:

$$\tilde{f}(x) = \exp^{-\left(\frac{x-m}{2\sigma}\right)^2} \quad m \in \{m_1, m_2\} \quad (7)$$

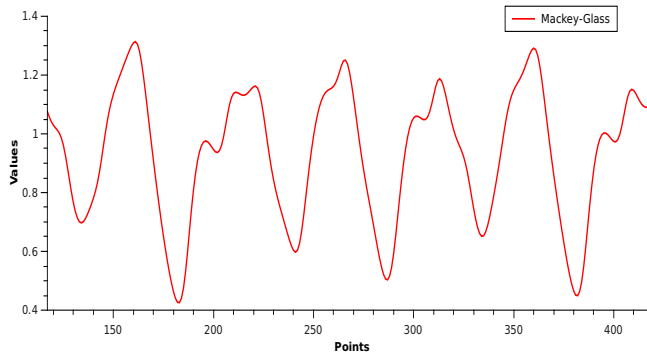


Fig. 3. Generated Mackey-Glass time series sample values

The upper  $\bar{\mu}_{\bar{A}}(x)$  and lower  $\underline{\mu}_{\bar{A}}(x)$  membership functions are determined by the following functions [9, p.91]:

$$\bar{\mu}_{\bar{A}}(x) = \begin{cases} \exp^{-\left(\frac{x-m_1}{2\sigma}\right)^2} & \text{if } x < m_1 \\ 1 & \text{if } m_1 \leq x \leq m_2 \\ \exp^{-\left(\frac{x-m_2}{2\sigma}\right)^2} & \text{if } x > m_2 \end{cases} \quad (8)$$

$$\underline{\mu}_{\bar{A}}(x) = \begin{cases} \exp^{-\left(\frac{x-m_2}{2\sigma}\right)^2} & \text{if } x \leq \frac{m_1+m_2}{2} \\ \exp^{-\left(\frac{x-m_1}{2\sigma}\right)^2} & \text{if } x > \frac{m_1+m_2}{2} \end{cases} \quad (9)$$

The upper membership functions  $\bar{\mu}_{\bar{A}}(x)$  and the lower membership functions  $\underline{\mu}_{\bar{A}}(x)$  in these equations will be used later to define  $FOU_{lower}$  and  $FOU_{upper}$  for general type-2 sets. The parameters of each interval type-2 fuzzy sets are the means and standard deviations. These parameters were initialized for antecedent fuzzy sets randomly within input variables space. Differently, output fuzzy sets are initialized randomly with values around the average values of outputs. The minimum t-norm is used as the operator in the fuzzification stage. The type-reduction was based on the collapsing method proposed by [30] to calculate the center of area values for the output interval type-2 sets. The form of collapsing used is the composite outward right-left as described in [31].

The optimization procedure in the training stage aims to search for the best antecedent and the consequent parts parameters for each rule. Then, these parameters are used in the testing stage. The number of all parameters in the IT2FLS is  $4 * 4 * 3 + 4 * 3 = 60$ . Hence, only the FOU's parameters are optimized in interval type-2 fuzzy sets.

### C. The initial general type-2 fuzzy logic system

To construct the initial GT2FLSs, two methods are used together. The first is the parameterization formula proposed in [17] and described in section III to parameterize general type-2 fuzzy sets. The second is the FLSs construction stages presented in section IV. The fuzzy model consists of the same number of rules and fuzzy sets used with IT2FLSs. The number of rules was chosen as four with an intention to reduce the system's complexity and computations. The work in this paper requires more computations than previous work in [17] as all FOU and SMF parameters will be optimized. Therefore,

we reduced the rules number to four instead of eight in [17]. The system is constructed from scratch rather than starting from optimized type-1 or interval type-2 fuzzy sets to initialize general type-2 fuzzy sets.

1) *The General type-2 Sets*: The general type-2 sets are defined using their  $FOU$ 's and  $SMF$ 's functions as follows:

- 1) *FOU*: The same membership functions that are used to define interval type-2 fuzzy sets upper and lower membership functions in subsection (VI-B) are used to define FOU parameters. The upper membership functions  $\bar{\mu}_{\bar{A}}(x)$  and the lower membership functions  $\underline{\mu}_{\bar{A}}(x)$  in this equation are used to define  $FOU_{upper}$   $FOU_{lower}$ . The means and standard deviations values that define each FOU are initialized by the same procedure described in subsection (VI-B).
- 2) *SMF*: SMFs are defined using a triangular SMF with a normal apex value in the middle between ( $FOU_{lower}$  and  $FOU_{upper}$ ) values for the ordered points  $k_1, k_2, \dots, k_n$  where ( $n = 9$ ). This is done by assigning a value of 0.5 to all apexes factors  $g(k_1) = g(k_2) = \dots = g(k_n) = 0.5$ . Then the apexes factors values for other  $x$  points are extracted using the linear interpolation function described in section III. This method to parameterize the general type-2 set is shown in figure III.

2) *The initial GT2FLS components*: The configurations of IT2FLS and GT2FLS used in this experiment are detailed in Table I. The initial GT2FLS configurations of fuzzification, implication and defuzzification are chosen as described in [17]. However, the type-reduction stage is different where two methods for type-reduction were used. These methods are the embedded sets based sampling method [12] and the vertical slice centroid type-reducer (VSCTR). The (VSCTR) which was initially proposed by [32] then detailed by [33] does not calculate the union for all the embedded sets involved in the general type-2 fuzzy sets. Although, this method does not depend on the concept of embedded sets, it is a good approach for practical usage. [33]. Therefore, this is an approximation of the centroid as long as the operation of union of all embedded type-2 fuzzy sets involved in the general type-2 fuzzy set has not been carried out. In sampling method, we used 100 embedded sets samples. The rationale for using two type-reduction methods is to test the true effects of learning SMF in general type-2 fuzzy sets without being distracted by the stochastic evaluation using sampling.

### D. Learning of FOU parameters

The training process attempts to find the optimal parameters of FOU's for all fuzzy sets to use them later in the testing phase. Therefore, the number of  $FOU$ s parameters in the system will be  $4 * 4 * 3 + 4 * 3 = 60$ . The learning process is conducted by modifying one parameter at a time and evaluating the cost of the new cost function using Root Mean Square Error (RMSE)

TABLE I  
THE INTERVAL AND GENERAL TYPE-2 FUZZY LOGIC SYSTEMS  
CONFIGURATIONS

Stage	IT2FLS	GT2FLS
Membership Function	Gaussian	Gaussian + triangular SMF
Number of parameters	60	60+180=240
Fuzzification	singleton fuzzification	singleton fuzzification
Antecedent combination	minimum t-norm	Coupland's meet
Implication	minimum t-norm	Coupland's meet
Join	maximum t-conorm	Coupland's join
SMF discretized points	none	9
Type-reduction centroid	collapsing method	sampling and VSCTR
Defuzzification	centroid	centroid
Y Discretization points	101	101

defined as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{s=1}^n [f(s) - f(s^*)]^2} \quad (10)$$

There is one constraint to the parameters in the optimization problem which requires proposed values of standard deviations to be  $\geq 0$ . Simulated annealing is started with these parameter settings:

- The new states are obtained randomly by adding a small step size to one chosen parameter at a time. Step size values are linked to the input space boundaries of each variable and  $= \max\text{-min}/25$  with a random direction.
- An initial temperature value is chosen as described by [34]. Therefore, this value is equal to the standard deviation of the average values of 1000 trial runs RMSE's of training samples.
- The cooling schedule of the temperature is calculated using a static cooling rate of 0.9 updated in the beginning of each Markov chain.
- The Markov chain length is dependent on the number of variables in the problem search space. It is chosen to be 5 times the number of total parameters.
- The search is terminated after 40 Markov chains.

The new state cost is then calculated using the RMSE's of the training samples. After terminating the search process, the test sample results are calculated using RMSE.

#### E. The learning of the secondary membership functions

This stage is concerned with finding the optimal locations of apexes (apexes factors) for all triangular secondary membership functions. This is carried out by fixing the two boundaries for each triangular SMF. Therefore, optimized parameters in this stage are the apexes factors  $g(k_1), g(k_2), \dots, g(k_n)$ . A simulated annealing algorithm is used for optimization using the same configurations described above with these differences:

- 1) The first is related to the constraints of apexes factors  $g(k_i)$  values which are constrained by their  $(FOU_{lower}(k_i)$  and  $FOU_{upper}(k_i))$  points.
- 2) The new step size values must be between  $[0, 1]$  and should be large enough to optimize the results. The step size is chosen to be 0.225.

- 3) Each Markov chain length is chosen to be 5 times the number of parameters in the problem search space.
- 4) The search process terminated after 10 Markov chains. These choices are designed to reduce the experiment's time.

The number of parameters optimized in this stage is  $n = 9$  for each fuzzy set. Therefore, a total of  $4 * 5 * 9 = 180$  parameters are optimized during this stage. The experiment has been carried out 20 times.

## VII. RESULTS AND DISCUSSION

The experiments were repeated 20 times. The results of learning Mackey-Glass time series are detailed in table II where the average RMSEs curves and the acceptance ratios during search are depicted in figures 4 and 5 respectively. The main observations are :

- 1) The best average RMSE in the testing samples was obtained by GT2FLS with VSCTR defuzzification (GT2FLS-VSCTR) followed by IT2FLS (IT2FLS).
- 2) The best average RMSE in the training samples was obtained by GT2FLS with VSCTR defuzzification followed by IT2FLS.
- 3) The average RMSEs curves when learning FOU's (training samples) have exhibited similar performances by the three models. However, IT2FLS obtained the best average RMSEs in testing phase followed by GT2FLS-VSCTR which was the best in training phase followed by IT2FLS.
- 4) The learning of SMFs using GT2FLS-VSCTR adds about 11.7% to the average testing RMSEs and about 17.7% to the average training RMSEs over the FOU's learning best results. The learning of SMFs using GT2FLS-Sampling adds about 0.86% to the training RMSEs but worsened the testing RMSEs by about  $-0.059$ .
- 5) The learning curves of SMFs showed a clear difference in performance between GT2FLS-VSCTR and GT2FLS-Sampling models. GT2FLS-VSCTR shows continuous improvements compared to very small improvements obtained by GT2FLS-Sampling.
- 6) The acceptance ratio curves when learning FOU's show similar behaviors between GT2FLS-VSCTR and IT2FLS, which are better than the narrower acceptance behavior obtained by GT2FLS-Sampling. The last one shows undesirable behavior where it converges to values close to 0% quickly in less than 30 Markov chains which means no improvements were observed in the rest of iterations.
- 7) The acceptance ratio curves when learning SMFs show a clear difference in behaviors between GT2FLS-VSCTR and GT2FLS-Sampling models. The GT2FLS-Sampling shows a very narrow acceptance behavior compared to GT2FLS-VSCTR. Interestingly, the acceptance ratios curves of the GT2FLS-Sampling model show narrower behavior when learning SMFs from its behavior with

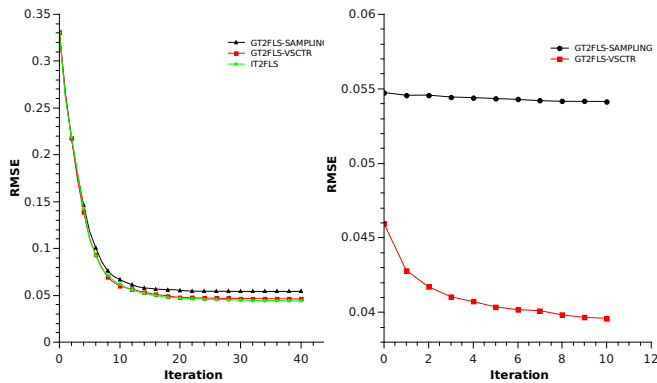


Fig. 4. The average convergence of the three models for noise-free Mackey-Glass time series problem when learning FOU (left) and SMF (right).

FOUs. However, as mentioned above, the initial temperatures were set separately in each stage to be proportional to the objective function differences brought by these moves in the two parameters groups (FOU and SMF). This is important to avoid starting with very large or very small initial temperatures and to have acceptable curves of best results and acceptance ratios. In other words, the observed acceptance behaviors for GT2FLS-Sampling model are not related to the settings of simulated annealing. This behavior can be easily explained by the effects of the defuzzification method which is the only difference between the two models of GT2FLS. The effects of the stochastic objective function when using the sampling method can be ignored when moves from state to state can bring relatively large differences compared to the random noise but this noise can deteriorate the search when moves bring improvements comparable to that noise. In other words, when learning FOU, the differences brought by moves are large enough to accept very small errors of approximated objective functions due to the larger contributions of FOU parameters to the objective functions compared to the SMF contributions. Hence, we do not expect a large contribution from learning SMF parameters compared to learning the FOU parameters due to the fact that SMF is dependent on FOU and bounded by its endpoints. This behavior of acceptance ratios when using GT2FLS-Sampling has been observed with other problems examined by the author and will be presented in the near future.

- 8) The time taken by IT2FLS was the shortest. It was 5.8 times faster than GT2FLS-VSCTR and 21.8 times faster than GT2FLS-Sampling. Therefore, IT2FLS is preferred in terms of speed.

TABLE II  
MACKEY-GLASS TIME SERIES FORECASTING RESULTS (RMSE)

Stage	$Mean_{RMSE}$	$Std_{RMSE}$	$Minimum_{RMSE}$
IT2FLS			
Training	0.04980955	0.0200348	0.026242
Testing	0.0433439	0.010239	0.027117
Time	332.55	21.027488	295
GT2FLS with Sampling Defuzzification			
After FOU's Learning			
Training	0.0553228125	0.01243	0.03761027
Testing	0.0518645455	0.0107249	0.03617023
After SMF's Learning			
Training	0.0548446	0.0119293	0.0372725
Improvement by SMF	0.86%	-	-
Testing	0.051895285	0.010721	0.0362123
Improvement by SMF	-0.059269 %	-	-
Time	7,259.9	992.126	5,724
GT2FLS with VSCTR Defuzzification			
After FOU's Learning			
Training	0.0483079765	0.01089	0.03428513
Testing	0.0446682685	0.0121448	0.02823214
After SMF's Learning			
Training	0.03975027	0.0115896	0.0240021
Improvement by SMF	17.7%	-	-
Testing	0.03943346	0.0116557	0.024325
Improvement by SMF	11.7%	-	-
Time	1,945.45	368.392	1,217

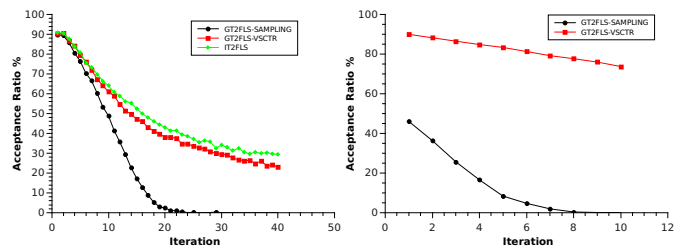


Fig. 5. The average acceptance ratios of the three models when learning FOU (left) and SMF (right) for noise-free Mackey-Glass time series problem.

## VIII. CONCLUSION

The learning of GT2FLSs using simulated annealing has been described and discussed. The learning to configure all GT2FLSs parameters in two stages has been applied in both primary and secondary membership functions (FOU and SMF). The learning process starts from scratch rather than using optimized IT2FLSs to initialize GT2FLSs. The novel parameterization approach has been used to design two models of GT2FLSs. These two models use two type reduction techniques: the sampling method is non-deterministic while the VSCTR method is deterministic. The rationale for using the two type-reduction techniques has been described. Both models, as well as IT2FLS, have been applied to model the four problems presented in previous chapters. The question of whether GT2FLSs can provide greater ability to handle uncertainty has been tackled. The stochastic defuzzification method of sampling embedded sets affected the learning performance in both FOU and SMF learning stages. The best results achieved in all problems have been accredited to GT2FLSs with VSCTR defuzzification. In terms of speed, IT2FLS was the best followed by GT2FLSs with VSCTR. The

results showed that when using the deterministic defuzzification method (VSCTR), the learning of GT2FLSs can provide extra capabilities to handle more information and uncertainties than IT2FLSs that use uniform SMFs. Although, the use of VSCTR is not based on the concept of using embedded sets to calculate the exact centroids of type-2 sets, the method allows the learning process to be carried out in a practical manner. This achievement opens the door to using other learning methods to get more modeling capabilities from the GT2FLSs in real-world applications.

## REFERENCES

- [1] F. Herrera, "Genetic fuzzy systems: status, critical considerations and future directions," *International Journal of Computational Intelligence Research*, vol. 1, no. 1-2, pp. 59–67, 2005.
- [2] R. John and S. Coupland, "Type-2 fuzzy logic: A historical view," *Computational Intelligence Magazine, IEEE*, vol. 2, pp. 57–62, 2007.
- [3] S. Coupland and R. John, "Geometric type-1 and type-2 fuzzy logic systems," *Fuzzy Systems, IEEE Transactions on*, vol. 15, no. 1, pp. 3–15, Feb. 2007.
- [4] J. Mendel, F. Liu, and D. Zhai, "Alpha plane representation for type-2 fuzzy sets: Theory and applications," *Fuzzy Systems, IEEE Transactions on*, vol. 17, no. 5, pp. 1189–1207, oct. 2009.
- [5] H. Hamrawi, S. Coupland, and R. John, "A novel alpha-cut representation for type-2 fuzzy sets," in *FUZZ IEEE 2010 (WCCI 2010)*, IEEE, Barcelona, Spain: IEEE, July 2010, pp. 1–8.
- [6] C. Wagner and H. H. H. Wagner, "Toward general type-2 fuzzy logic systems based on z-slices," *Fuzzy Systems, IEEE Transactions on*, vol. 18, no. 4, pp. 637–660, aug. 2010.
- [7] H. H. Christian Wagner, "Novel methods for the design of general type-2 fuzzy sets based on device characteristics and linguistic labels surveys," in *2009 IFSA World Congress, EUSFLAT World Conference*, Lisbon, Portugal, July 2009, pp. 537–543.
- [8] J. Mendel and R. John, "Type-2 fuzzy sets made simple," *Fuzzy Systems, IEEE Transactions on*, vol. 10, no. 2, pp. 117–127, apr 2002.
- [9] J. Mendel, *Uncertain rule-based fuzzy logic systems: introduction and new directions*. Prentice Hall, 2001.
- [10] S. Greenfield and R. John, "Optimised generalised type-2 join and meet operations," in *Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International*, july 2007, pp. 1–6.
- [11] J. Starczewski, "Extended triangular norms," *Information Sciences*, vol. 179, no. 6, pp. 742–757, 2009.
- [12] S. Greenfield, R. John, and S. Coupland, "A novel sampling method for type-2 defuzzification," in *Proceedings of UKCI 2005*, London, September 2005, pp. 120–127.
- [13] O. Linda and M. Manic, "Importance sampling based defuzzification for general type-2 fuzzy sets," in *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, july 2010, pp. 1–7.
- [14] F. Liu, "An efficient centroid type-reduction strategy for general type-2 fuzzy logic system," *Information Sciences*, vol. 178, no. 9, pp. 2224–2236, 2008.
- [15] R. John and C. Zarnetcki, "A type 2 adaptive fuzzy inferencing system," in *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, vol. 2. IEEE, 1998, pp. 2068–2073.
- [16] W. Jeng, C. Yeh, and S. Lee, "General type-2 fuzzy neural network with hybrid learning for function approximation," in *Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on*. IEEE, 2009, pp. 1534–1539.
- [17] M. Almaraashi, R. John, and S. Coupland, "Designing generalised type-2 fuzzy logic systems using interval type-2 fuzzy logic systems and simulated annealing," in *Fuzzy Systems (FUZZ), 2012 IEEE International Conference on*. IEEE, June 2012.
- [18] M. Almaraashi and R. John, "Tuning of type-2 fuzzy systems by simulated annealing to predict time series," in *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2011, WCE 2011*, vol. 2. London, U.K: Newswood Limited, 6-8 July 2011, pp. 976–980.
- [19] —, "Tuning type-2 fuzzy systems by simulated annealing to estimate maintenance cost," in *proceedings the UKCI 2011*, Manchester, Sep 2011.
- [20] M. Mizumoto and K. Tanaka, "Some properties of fuzzy sets of type 2," *Information and Control*, vol. 31, no. 4, pp. 312–340, 1976.
- [21] N. Karnik and J. Mendel, "Operations on type-2 fuzzy sets," *Fuzzy Sets and Systems*, vol. 122, no. 2, pp. 327–348, 2001.
- [22] J. Mendel, "Type-2 fuzzy sets and systems: An overview [corrected reprint]," *IEEE Computational Intelligence Magazine*, vol. 2, no. 2, pp. 20–29, 2007.
- [23] J. T. Starczewski, "Efficient triangular type-2 fuzzy logic systems," *International Journal of Approximate Reasoning*, vol. 50, no. 5, pp. 799–811, May 2009.
- [24] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing, 1983," *Science*, vol. 220, pp. 671–680, 1983.
- [25] P. Salamon, P. Sibani, and R. Frost, *Facts, conjectures, and improvements for simulated annealing*. Society for Industrial Mathematics, 2002.
- [26] A. A. Hopgood, *Intelligent systems for engineers and scientists*. CRC press, 2012.
- [27] E. H. L. Aarts and H. M. M. T. Eikelder, "Simulated annealing," in *Handbook of applied optimization*, P. Pardalos and M. Resende, Eds. Oxford University Press, 2002, pp. 209–220.
- [28] M. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.
- [29] J.-S. Jang, "Anfis: adaptive-network-based fuzzy inference system," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 23, no. 3, pp. 665–685, may/jun 1993.
- [30] S. Greenfield, F. Chiclana, S. Coupland, and R. John, "The collapsing method of defuzzification for discretised interval type-2 fuzzy sets," *Information Sciences*, vol. 179, no. 13, pp. 2055–2069, June 2009, ISSN: 0020-0255.
- [31] S. Greenfield, F. Chiclana, and R. John, "The collapsing method: Does the direction of collapse affect accuracy?" in *IFSA-EUSFLAT 2009 Conference*, 2009.
- [32] R. John, "Perception modelling using type-2 fuzzy sets / r. i. john." Ph.D. dissertation, De Montfort University, 2000.
- [33] L. Lucas, T. Centeno, and M. Delgado, "General type-2 fuzzy inference systems: Analysis, design and computational aspects," in *Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International*, july 2007, pp. 1–6.
- [34] S. White, "Concepts of scale in simulated annealing," in *American Institute of Physics Conference Series*, vol. 122, 1984, pp. 261–270.