International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:1, 2009

Auto Regressive Tree Modeling for Parametric Optimization in Fuzzy Logic Control System

Arshia Azam, J. Amarnath, and Ch. D. V. Paradesi Rao

***Abstract*—**The advantage of solving the complex nonlinear problems by utilizing fuzzy logic methodologies is that the experience or expert's knowledge described as a fuzzy rule base can be directly embedded into the systems for dealing with the problems. The current limitation of appropriate and automated designing of fuzzy controllers are focused in this paper. The structure discovery and parameter adjustment of the Branched T-S fuzzy model is addressed by a hybrid technique of type constrained sparse tree algorithms. The simulation result for different system model is evaluated and the identification error is observed to be minimum.

***Keywords*—**Fuzzy logic, branch T-S fuzzy model, tree modeling, complex nonlinear system.

I. Introduction

DURING the past few years, fuzzy logic has been finding a rapidly growing number of applications in fields ranging from consumer electronics and application to medical diagnosis systems and security management. In most of these applications the tolerance with respect to variation is the main objective. In this effect, the main operation principle of fuzzy logic is to Precise the operation at lower effort. Minimize the effort needed to perform a task. Fuzzy logic provides a wide variety of concepts and techniques for representing and inferring from knowledge, which is imprecise, uncertain or lacking in reliability. At this point, however, what is used in most practical applications is a relatively restricted and yet important part of fuzzy logic centering on the use of fuzzy if-then rules. This part of fuzzy logic is referred to as the calculus of fuzzy if-then rules (CFR), because it constitutes a fairly self- contained collection of concepts and methods for dealing with varieties of knowledge which can be represented in the form of a system of if-then rules in which the past history and Consequents are evaluated. Fuzzy logic systems [1][8] have been successfully applied to a vast number of scientific and engineering problems in recent years. The advantage of solving the complex nonlinear problems by utilizing fuzzy logic methodologies is that the experience or expert's knowledge described as a fuzzy rule base can be directly embedded into the systems for dealing with the problems. A number of improvements have been made in the aspects of enhancing the systematic design method of fuzzy logic systems [9][13]. Many researches focus on the automatically finding the proper structure and parameters of fuzzy logic systems by using genetic algorithms [10][12][13], evolutionary programming [11], tabu search [14], and so on. But still there remain various problems to be focused and solved, such as, how to automatically partition the input space for each input-output variables, how many fuzzy rules are really needed for properly approximating the unknown nonlinear systems, and how to determine it automatically. In this paper, a method for effectively tuning the parameters and structure of fuzzy logic systems to achieve the required objective is been focused.

Arshia Azam is research scholar at Jawaharlal Nehru Technological University Hyderabad, India (e-mail: aazam_04@yahoo.co.in).

J. Amarnath is with Department of Electrical & Electronics Engineering, Jawaharlal Nehru Technological University, Hyderabad (e-mail: amaranth_jinka@hotmail.com).

Ch. D. V. Paradesi Rao is with Department of Electronics & Communication Engineering, Jawaharlal Nehru Technological University, Hyderabad.

II. Fuzzy Set Definition

Fuzzy sets were introduced by Lotfi Zadeth in 1965. They could be presented as,

Let, X = {x1, x2, x3, x4, x5} crisp set called as universal set and,

Let, $Y \subset x$ = {x1, x2, x3} is its crisp subset.

By using the characteristic function defined as:

$$\mu_Y(x) = \begin{cases} 1 \text{ if } x \in Y \\ 0, \text{ otherwise} \end{cases}$$

The subset Y can be uniquely represented by ordered pairs Y = $\{(x_1, 1), (x_2, 1), (x_3, 0), (x_4, 0), (x_5, 1)\}$.

Zadeth proposed that the second member of an ordered pair (which is called the membership grade of the appropriate element) can take its value not only from the set {0, 1} but from the closed interval [0, 1] as well. By using this idea fuzzy sets are defined as;

Let X a universal crisp set. The set of ordered pairs $Y = \{(x, \mu_Y(x)) | x \in X, \mu_Y : X \quad [0, 1]\}$ is said to be the fuzzy subset of X. The $\mu_Y: X \quad [0, 1]$ function is called as membership function and its value is said to be the membership grade of x. An modified format of fuzzy model proposed by Takagi and Sugeno [1] is described by fuzzy if-then rules whose resulting parts are represented by linear equations. This fuzzy model is of the following form:

If x_1 is A_{i1} . . . , x_n is A_{in} then $y_i = c_{i0} + c_{i1}x_1 + \ldots + c_{in}x_n$ where i = 1, 2, . . .,N, N is the number of if-then rules, c_{ik}(k = 0, 1, . . . , n) are the consequent parameters, y_i is the output from the i^{th} if-then rule, and A_{ik} is a fuzzy set. Given an input $(x_1, x_2, \ldots, x_n)$, the final output of the fuzzy model is referred as follows:

$$y=\frac{\sum_{i=1}^{N}\omega_i y_i}{\sum_{i=1}^{N}\omega_i}=\frac{\sum_{i=1}^{N}\omega_i(c_i0+c_{i1}x_1+....+c_{in}x_n)}{\sum_{i=1}^{N}\omega_i} \\ =\frac{\sum_{k=0}^{n}\sum_{i=1}^{N}\omega_i c_{ik}x_k}{\sum_{i=1}^{N}\omega_i} \quad (1)$$

Where $x_0 = 1$, ω_i is the weight of the i^{th} IF-THEN rule for the input and is calculated as,

$$\omega_i=\prod_{k=1}^{n}\mathrm{A}_{\mathrm{ik}}\left(x_k\right) \quad (2)$$

$A_{ik}(x_k)$, where $A_{ik}(x_k)$ is the grad of membership of x_k in A_{ik}.

To Takagi-Sugeno approach, the universal approximation property was proved in [2][3]. In addition, a further generalization of this approach was proposed in [4][5], in which in the conclusion of each rule, the desired output y is given not by an explicit formula, but by a (crisp) dynamical systems, i.e., by a system of differential equations that determine the time derivative of the output variable as a function of the inputs and of the previous values of output. This generalization also has universal approximation property.

A simplified Takagi-Sugeno fuzzy model was proposed by Ying Hao [6] with the following rule base.

If x_1 is A_{i1} . . . , x_n is A_{in} then $y_i = k_i(c_0 + c_1x_1 + . . . + c_nx_n)$

where i = 1, 2, . . .,N, N is the number of if-then rules. From this it can be seen that the free parameters in the consequent part of the IF-THEN rules are reduced significantly in this case. The universal approximation property of this simplified T-S fuzzy model has also been proved, and successfully applied to the identification and control of nonlinear systems [7]. The advantage of solving the complex nonlinear problems by utilizing fuzzy logic methodologies is that the experience or expert's knowledge described as a fuzzy rule base can be directly embedded into the systems for dealing with the problems. A number of improvements have been made in the aspects of enhancing the systematic design method of fuzzy logic systems [9][13]. In these researches, the needs for effectively tuning the parameters and structure of fuzzy logic systems are increased. Many researches focus on the automatically finding the proper structure and parameters of fuzzy logic systems by using genetic algorithms [10][12][13], evolutionary programming [11], tabu search [14], and so on. But there still remain the problem of automatically partition the input space for each input output variables, optimal selection of the fuzzy rules needed for properly approximating the unknown nonlinear systems, and to automize the selectivity.

III. BRANCHED T-S FUZZY MODEL

The problems mentioned above can be partially solved by the recent developments of Branched fuzzy systems [15][19]. As a way to overcome the curse-of dimensionality, it was suggested in [19] to arrange several low-dimensional rule bases in a Branched structure, i.e. a tree, causing the number of possible rules to grow in a linear way with a number of inputs. But no method was given on how the rules for these Branched fuzzy systems could be determined automatically. In [18] the author describes a new algorithm, which derives the rules for Branched fuzzy associative memories that are structured as a binary tree. In Ref. [17][16], a specific Branched fuzzy system is proposed and its universal approximation property was proved. But the main problems in fact lies that this is a specific Branched fuzzy systems which lack of the flexibility in the structure adaptation, and it is difficult to arrange the input variables as there is no general method to determine which inputs to the Branched fuzzy system are more influential to the output. In [20], a genetic algorithm-based evolutionary learning approach to the search for the best Branched structure of the five input-variable fuzzy controller and the parameters of the combined controller is proposed for the low-speed control. In intuition, the Branched fuzzy logic systems not only provide a more complex and flexible structure for modeling the nonlinear systems, but can also reduce the size of rule base in some extend. But there is no systematic method for designing of the Branched T-S fuzzy systems now. The problems in designing of Branched fuzzy logic system are:

Selecting a proper Branched structure;

Selecting the inputs for each partial fuzzy model;

Determining the rule base of each fuzzy logic T-S model;

Optimizing the parameters used in the fuzzy membership functions and the then part of T-S fuzzy model.

In this sense, finding a proper Branched T-S fuzzy model can be posed as a problem in the structure and parameter space. Fig. 1 shows some possible Branched T-S fuzzy models for the number of input variables 4 and the number of Branched layers. It can be seen that; it is important to select a proper Branched T-S fuzzy model structure for approximating an unknown nonlinear system;

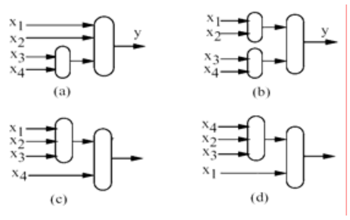

Fig 1: Possible Branched fuzzy logic models for a multi input single output system

As there is no general conclusion about which inputs are more influential to the system output, thus it is important to select the appropriate input in Branched TS fuzzy model. In fact, different inputs selection can form the different Branched models. If each variable is divided into 2 fuzzy sets, then the number of fuzzy rules in each of Branched fuzzy systems is 12, which is generally smaller than the number of rules in non-Branched fuzzy systems with complete rule set.

In this paper, a method for automatically designing of Branched T-S fuzzy systems is proposed. The structure discovery and parameter adjustment of the Branched T-S fuzzy model is addressed by a hybrid technique of type constrained sparse tree algorithms. The model structure selection and parameters optimization of the Branched T-S fuzzy model can be coded as a type constrained sparse tree. Therefore, the optimal nonparametric model of nonlinear systems can be obtained by the evolutionary induction of the type constrained sparse tree.

IV. Representation of Branched T-S Fuzzy Model

A tree structural representation of the Branched T-S fuzzy model is been presented, in which each of the Branched T-S fuzzy models can be coded as a type constrained sparse tree. There is

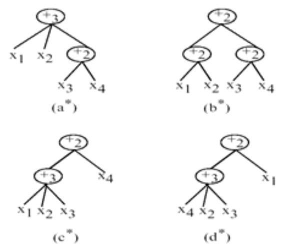

Fig 2 Tree structural representation of Branched T-S fuzzy models for Fig. 1

no need to encode and decode between the tree and the Branched T-S fuzzy model in the calculation of the Branched T-S fuzzy model as a flexible data structure into the nodes of the tree which can be directly calculated in a recursive way. Therefore, the optimization of the Branched T-S fuzzy model can be directly replaced by the evolutionary induction of the type constrained sparse tree.

V. Auto Regressive Tree Modeling

Suppose that the Branched T-S fuzzy models have three Branched layers. In this case, three instruction sets I_0, I_1 and I_2 can be used for generating the tree. In general, the used instruction sets are $I_0 = \{k_2, k_3, \ldots, k_{n1}\}$, $I_1 = \{x_1, x_2, \ldots, x_n, k_2, k_3, \ldots, k_{n2}\}$ and $I_2 = \{x_1, x_2, \ldots, x_n\}$, where the instruction k_{ni} ($i = 2, 3, \ldots, n1$) in the instruction set I_0 means that there are ni input variables in the output layer of the T-S fuzzy model, which is also the number of branches of the root node. The instructions in the instruction set I_1 are used to generate the hidden layer of the T-S fuzzy models. It can be seen that if all the instructions used in the hidden layer are input variables the T-S Branched model becomes an usually used non-Branched fuzzy model. In contrast, if there is one instruction which is not any one of the input variables $x_1, x_2, \ldots, x_n$, a Branched T-S fuzzy model is created. This means that the non-input variable instruction becomes a sub-T-S fuzzy model, which has its input variables come from the instruction set I_2 and it's output is the input of the next layer of the T-S fuzzy model. The instructions used in the layer 0, 1 and 2 of the tree are randomly selected from instruction set I_0, I_1 and I_2, respectively. The initial probability of selecting instructions is given by

$$P(I_{d,\omega}) = \frac{1}{l_i}, \forall I_{d,\omega} \in I_i, i = 0,1,2 \tag{3}$$

Where I_d, ω denotes the instruction of the node with depth d and width ω, l_i ($i = 0, 1, 2$) is the number of instructions in the instruction set I_i. As mentioned above, the tree can be generated in a recursive way as follows:

(1) Select a root node from the instruction set I_0 according to the probability of selecting instructions. If the number of arguments of the selected instruction is larger than one, then create a number of parameters (including the fuzzy membership function parameters for each input variable, the THEN part parameters in each fuzzy rule) attached to the node.
(2) Create the left sub-node of root from the instruction set I_1. If the number of arguments of the selected instruction is larger than one, then create a number of parameters attached to the node, and then create its left sub-node as same way from the instruction set I_2. Otherwise, if the instruction of the node is an input variable, return to upper layer of the tree in order to generate the right part of the tree.
(3) Repeat this process until a full tree is generated.
There are two key points in the generation of the tree. The one is that the instruction is selected according to the initial probability of selecting instructions, and the probability will be changed with generation by a probabilistic incremental algorithm. The other is that if the selected node is a non-terminal node, then generate the corresponding data structure attached to the node.

VI. Results

The proposed approach is tested for various system models to verify the effectiveness of the proposed method for the auto identification of the fuzzy system. The architecture of the Branched T-S fuzzy model is evolved and is optimized by a global search algorithm called modified evolutionary programming. The used parameters for the simulation evaluation is as given below,
Learning probability (P_b) = 0.01
Learning rate (l_r) = 0.01
Fitness constant (f) =10e-6
Mutation probability (P_m) = 0.4
Mutation rate (m_r) = 0.4
The maximum number of EP search is given by,
Step = δ * (1 + generation), where δ is the basic steps of EP search.
for a given system defined by,
$y(k + 1) = 2.6y(k) - 2.3y(k - 1) + 0.69y(k - 2) + 0.01u(k) - 0.03u(k - 1) + 0.014u(k - 2)$
400 data points were generated with the randomly selected input signal u(k) between -1.0 and 1.0. The first 200 points were used as an estimation data set and the remaining data were used as a validation data set. The input vector is set as x = [y(k), y(k-1), y(k-2), u(k), u(k - 1), u(k - 2)]. The used instruction sets are I_0 = {k2, k3, . . . , k6}, I_1= {x0, x1, x2, x3, x4, x5,k2,k3} and I_2 = {x0, x1, x2, x3, x4, x5}. The evolved Branched T-S fuzzy model is obtained as;

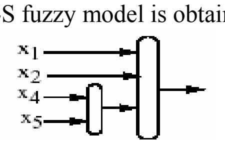

Fig. 3 Branched T-S Fuzzy model

The output of the evolved model and the actual output for validation data set is obtained as;

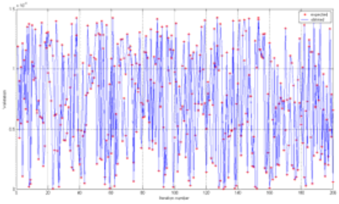

Fig. 4 Output of the evolved model

and the identification error is obtained as

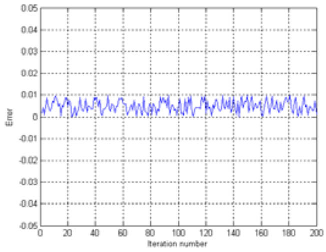

Fig. 5 Identification error of the model

for a system defined as $y(k + 1) = y(k)\ 1.5 + y2(k) - 0.3y(k - 1) + 0.5u(k)$

The input and output of system are $x(k) = [u(k), u(k-1), y(k), y(k-1)]$ and $y(k+1)$, respectively.

The training samples and the test data set are generated using the same sequence of random input signals as mentioned previously. The used instruction sets are $I_0 = \{k2,k3, . . . , k5\}$, $I_1 = \{x0, x1, x2, x3,k2, x3\}$ and $I_2 = \{x0, x1, x2, x3\}$. The evolved Branched T-S fuzzy model is;

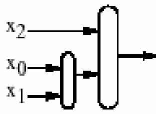

Fig. 6 T-S fuzzy model

The output of the evolved model and the actual output for validation data set are shown,

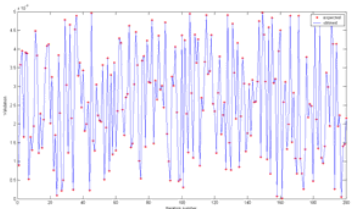

Fig. 7 Out put of the Fig. 6 model

and the identification error obtained is,

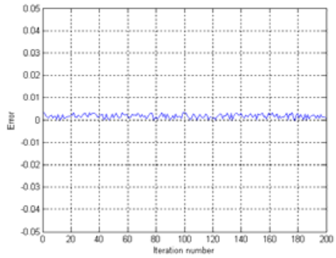

Fig. 8 Identification error of Fig. 6

for a given system model
$y(k + 1) = 1.752821y(k) - 0.818731y(k - 1) + 0.011698u(k) + 0.010942\ u(k - 1)\ 1+ y2(k - 1)$

The input and output of system are $x(k) = [u(k), y(k), y(k - 1)]$ and $y(k + 1)$, respectively. The training samples and the test data set are generated using the same sequence of random input signals as mentioned previously. The used instruction sets are $I0 = \{k2,k3,k4\}$, $I1 = \{x0, x1, x2,k2,k3\}$ and $I2 = \{x0, x1, x2\}$.

The evolved Branched T-S fuzzy model is,

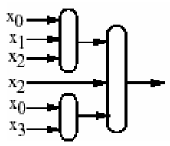

Fig. 9 T-S fuzzy model

The output of the evolved model and the actual output for validation data set and identification error for model in Fig. 9 shown in Figs. 10 & 11 respectively.

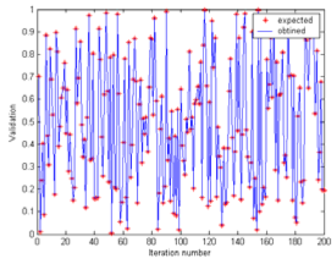

Fig. 10 Output of Fig. 9 model

and the identification error is,

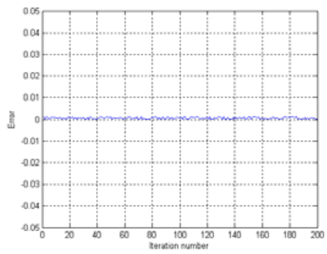

Fig. 11 Identification error of Fig. 9

VII. Conclusion

Based on a tree representation and calculation of the fuzzy models, a framework for evolving the automated designing of Branched T-S fuzzy systems is proposed. In this flexible framework, various computing models and various parameter-tuning strategies can be combined in order to find a proper computing model efficiently. All the fuzzy computing models can be represented and calculated by the type constrained sparse tree with pre-specified data structure, which is attached to the node of the tree. In this sense, the computing models are created automatically, therefore, the difficulties in determining of the architecture of computing models can be avoided to some extend. It can also seen that based on this idea the architecture and parameters used in the computing models can be evolved and optimized by using proper learning algorithm. Simulation results show that the proposed method works very well for the nonlinear system modeling problems for randomly selected systems. The presented approach could be extended with the controller operation with linear mean operator for enhancement of the fuzzy controller operation.

References

[1] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its application to modeling and control”, IEEE Trans. Syst., Man, Cybern., Vol.15, 116-132, 1985.

[2] Badgwell T. A. and Qin S. J., “Review of nonlinear model predictive control applications, chapter 1”, In: Kouvaritakis B. and Cannon M., Nonlinear predictive control: theory and practice, IEE Control Engineering Series 61, The Institution of Electrical Engineers, London, 2001.

[3] J.J. Buckley et al., “Fuzzy input-output controller are universal approximators”, Fuzzy Sets and Systems, Vol.58, pp.273-278, 1993.

[4] W. Duch, R. Setiono, and J. Zurada, “Computational intelligence methods for rule-based data understanding,” Proc. IEEE, vol. 92, no. 5, May 2004.

[5] W. Duch, R. Adamczak, and K. Grabczewski, “A new methodology of extraction, optimization and application of crisp and fuzzy logical rules,” IEEE Trans. Neural Netw., vol. 12, no. 2, pp. 277–306, Mar. 2001.

[6] H. Ying, “General Takagi-Sugeno “fuzzy systems with simplified linear rule consequent are universal controllers models and filters”, Information Science, Vol.108, pp.91-107, 1998.

[7] H. Ying, “Theory and application of a novel fuzzy PID controller using a simplified Takagi-Sugeno rule scheme”, Information Science, Vol.123, pp.281-293, 2000.

[8] C. Xu and Y. liu, “Fuzzy model identification and self learning for dynamic systems”, IEEE Trans. Syst., Man, Cybern., Vol.17, pp.683-689, 1987.

[9] Q. Gan and C.J. Harris, ”Fuzzy local linearization and logic basis function expansion in nonlinear system modeling”, IEEE Trans. on Systems, Man, and Cybernetics-Part B, Vol.29, N0.4, 1999.

[10] S. Yuhui et al., “Implementation of evolutionary fuzzy systems”, IEEE Trans. Fuzzy Systems, Vol.7, No.2, pp.109-119, 1999.

[11] K. Sin-Jin et al., “Evolutionary design of fuzzy rule base for nonlinear system modeling and control”, IEEE Trans. Fuzzy Systems, Vol.8, No.1, pp.37-45, 2000.

[12] H. Yo-Ping et al., “Designing a fuzzy model by adaptive macroevolution genetic algorithms”, Fuzzy Sets and Systems, Vol.113, pp.367-379, 2000.

[13] W. Baolin et al., “Fuzzy modeling and identification with genetic algorithm based learning”, Fuzzy Sets and Systems, Vol.113, pp.352-365, 2000.

[14] D. Maurizio et al., “Learning fuzzy rules with tabu search-an application to control”, IEEE Trans. on Fuzzy Systems, Vol.7, No.2, pp.295-318, 1999.

[15] G.V.S. Raju et al., “Hierarchical fuzzy control”, Int. J. Contr., Vol.54, No.5, pp.1201-1216, 1991.

[16] L.X. Wang, “Universal approximation by hierarchical fuzzy systems”, Fuzzy Sets and Systems, Vol.93, pp.223-230, 1998.

[17] O. Huwendiek et al., “Function approximation with decomposed fuzzy systems”, Fuzzy Sets and Systems, Vol.101, pp.273-286, 1999.

[18] Al Seyab R. K. and Cao Y., “Nonlinear system identification for predictive control using continuous time recurrent neural networks and automatic differentiation”, Submitted for publication in: IEEE TNN Special Issue on Feedback Control, September, 2005.

[19] C. Wei and Li-Xin Wang, “A note on universal approximation by hierarchical fuzzy systems”, Information Science, Vol.123, pp.241-248, 2000.

[20] Yongquan Y., Ying H., Minghui W., Bi Z., Guokun Z., “Fuzzy neural PID controller and tuning its weight factors using genetic algorithm based on different location crossover”, IEEE International Conference on Systems, Man and Cybernetics pp. 3709-3713, 2004.