T-Norm Adaptation in Fuzzy Logic Systems Using Genetic Algorithms

M. Taner Eskil, M. Onder Efe and Okyay Kaynak
Bogazici University, Electrical and Electronic Engineering Department
Mechatronics Research and Application Center
Bebek, 80815, Istanbul, Turkey

Abstract – This paper investigates the performance of Fuzzy Inference Systems having parameterized T-Norms in control of robotic manipulators. The adaptation of controller parameters is carried out by Genetic Algorithms. The error and the derivative of error are utilized in the decision process. The chromosomes, which include the adjustable parameters, are updated periodically by reproduction, crossover and mutation. Conventional reproduction and crossover methods and simulated-annealing type mutation are applied to find best-fit chromosomes. The efficiency of the proposed method is observed on a two degrees of freedom direct drive SCARA manipulator. It is seen that the proposed approach results in a distinguished performance comparatively to those using gradient based strategies.

1. INTRODUCTION

Fuzzy systems have become a widely used design framework in many engineering sciences. The outcome of ever-increasing interest to the design of fuzzy systems has shown itself in the form of products being used even in the daily life. Therefore it is no more surprising to see that many products or systems employing Fuzzy Inference Systems (FIS) in performing a task which is either from daily life or from industry. Washing machines, sewing machines, cameras, intelligent controllers, identifiers and pattern/image recognition systems are typical examples of such systems, which somehow incorporate a degree of intelligence provided by fuzzy inference mechanisms.

Fuzzy systems are found out to be useful especially in the cases where linguistic manipulation of data has a descriptive quality contrary to what is followed in conventional approaches, which typically require tedious design procedures. This nature of fuzzy inference mechanisms allows the designer to incorporate human expertise into the design thereby increasing the flexibility of the solution. In this respect, fuzzy logic, with the property of representing expert knowledge in the form of IF-THEN statements, constitute an appropriate approach for enabling some kind of human-machine interaction. There are several reasons that make the use of fuzzy systems so attractive. Firstly, the framework enables the designer to impose his/her feelings, intuitions or beliefs into the task space. Secondly, the fuzzy system models possess a great flexibility in choosing the methods of fuzzification, defuzzification and construction of rule base. Thirdly, the adaptability of the architecture makes it useful in fine-tuning of the parameters.

Much of the results, which have been reported so far, have focused on the improvement of fuzzy system performance. In [1], the architecture of a standard adaptive fuzzy system is described. The structure introduced by Wang [1], uses the product inference rule and Gaussian membership functions. The tuning is performed on the parameters of the Gaussians and the weights used in the defuzzification procedure. In [2], an extended version of fuzzy system construction called Adaptive Neuro Fuzzy Inference Systems (ANFIS) is developed. This architecture has greatly improved the realization performance of fuzzy systems and has extensively been used for identification and control purposes [3-4]. The output of ANFIS is a linear function of its augmented input vector. Therefore, by appropriately setting the defuzzifier parameters, one can easily obtain the standard fuzzy system described in [1]. Hence, the dimension of parameter space in ANFIS approach is greater than the standard approach. Another mechanism is proposed by Takagi and Sugeno [5], and is known as Takagi-Sugeno (TS) fuzzy model. The model analyzed in [5] performs an interpolation between dynamic system models on the basis of fuzzy activation. Therefore, TS fuzzy models can make use of the prominent features of conventional design methodologies in analyzing the critical issues such as stability and robustness.

The approaches mentioned so far are the most notable architectural developments in fuzzy system theory. However, the common use of these architectures generally adopts the gradient based training algorithms. The recent studies are addressed to the improvement of aggregation methods. In [6], Batyrshin and Kaynak propose parameterized T-Norms, which offer an additional adaptive learning possibility in aggregating the premises of rules. Introducing design flexibility through the adjustment of aggregation method is of a crucial importance in the cases where nondifferentiable membership functions are used or when membership functions do not have adjustable parameters.

Conventional adaptation methods for fuzzy systems are involved with the parameters of the membership functions or the parameters of the defuzzifier. Utilizing only the sensitivity derivatives in the parameter update procedure constitute a drawback from a safety point of view. These applications either require stabilizing forces on the training dynamics [7-8] or derivative-free optimization techniques such as Genetic Algorithms (GAs).

Until recently, adaptation on aggregation strategy has not received as much attention as it deserved for learning in fuzzy logic systems. Furthermore, even in the case of conventional adaptive fuzzy systems, the use of GAs as a method for adaptation of parameters is not very common. This paper investigates the use of GAs as a tool for training of adaptive fuzzy systems utilizing parameterized T-Norms.

Genetic Algorithms constitute a class of stochastic search algorithms. Procedurally, the approach starts with a randomly generated population. As the population evolves in time, candidates of the optimal or near optimal solutions to the problem in hand are generated through the use of operators in the terminology of GAs. During the evolution phase, many solution candidates are generated and their fitness measures are evaluated. This property of GAs make them attractive in the sense of tackling with large set of points without getting stuck into local minima or maxima [9-11].

The paper is organized as follows: The second section briefly considers the evolutionary computation and its operators. The next section describes the plant to be controlled. In the fourth section, fuzzy system model is briefly introduced. In this section, the design of a fuzzy controller using the standard architecture with parameterized T-Norms is discussed. The fifth section describes the control strategy utilizing the proposed approach. Simulation results are presented in the sixth section. The concluding remarks on the system performance are given in the last section.

2. EVOLUTIONARY COMPUTATION

Genetic algorithms begin with a population of string structures created randomly. Thereafter, the fitness measure of each string in the population is evaluated. The population is then reformed by three main operators, namely, reproduction, crossover, and mutation. Depending on the fitness measure for each solution candidate, the operation is continued until the termination criteria are met. One cycle of the application of above mentioned operators and the evaluation procedure is known as a *generation* in the terminology of GAs [12]. In this section, the operators in the terminology are briefly reviewed.

2.1. Population

The solution to the optimization problem requires a set of candidates, which are structurally represented as chromosomes. The content of the chromosomes is composed of zeros and ones. It is therefore evident that the length of the binary coded representation will determine the maximum number of individuals in the population as well as the accuracy of the solution. Namely, if a parameter is encoded with B bits and if Q parameters are adjustable, then each chromosome will include BxQ bits in the representation [13]. Typically, the initial population is generated randomly.

2.2. Objective Function

The fitness/objective function is chosen depending on the problem in hand such that the individuals having high fitness values are the good solution candidates for the optimization. Therefore, reproduction of the next generation will strictly be dependent on the fitness measure.

2.3. Reproduction

Reproduction is a method for increasing the number of solution candidates having high fitness values. A mating pool is formed by reproduction, which, at the same time, eliminates the least-fit strings from the pool.

2.4. Crossover

Crossover operator is applied to the strings in the mating pool. Like reproduction operator, there exist a number of crossover operators in the literature. Conceptually, two strings are picked from the mating pool randomly and depending on the crossover criteria, an exchange of bits of same length is performed. In a single-point crossover operator, this is performed by randomly choosing a crossing site along the string and by exchanging all bits on the right side of the crossing site as shown below.

$$\begin{array}{cc|ccc} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{array} \Rightarrow \begin{array}{cc|ccc} 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{array}$$

It is intuitive from this construction that good substrings from either parent string can be combined to form a better child string if an appropriate site is chosen. Since the knowledge of an appropriate site is usually now known, a random site is chosen. Therefore the fitness measure of the child string is not guaranteed to be better than its parents. In this respect, one should understand the motivation behind the use of such an operator as capability of maintaining the diversity in the population. If the fitness of the child is high, there will be more copies of it in the next generation, if not, it will not survive beyond the next generation due to the selective behavior of reproduction operator [12].

2.5. Mutation

Mutation operator is another degree of freedom in search procedure, which is frequently used in GA based designs. Functionally, the operator negates the value of a bit in the string. This makes it possible to reach to the inaccessible regions of search space. The need for mutation is to keep the diversity in the population too. For example, if beyond a particular position along the all strings in the population have a value 0, and if a 1 is needed beyond that position to obtain the optimum, then neither reproduction nor crossover operator described above will be able to create a 1 in that position. The activation of the mutation operator is generally controlled by the excess of a certain probability threshold. Therefore the design must include such a criterion.

3. PLANT MODEL

In this study, a two degrees of freedom direct drive robotic manipulator, which is illustrated in Fig. 1, has been used as the test bed. Since the dynamics of such mechatronic systems is modeled by coupled and complicated differential equations, being in pursuit of output tracking precision becomes a tedious work due to the strong interdependency between the variables involved. Besides, the ambiguities concerning the friction related dynamics in the plant model make the design much more complicated if one utilizes the formalism described in conventional design techniques, which defy accurate analytical modeling. Therefore the methodology adopted must be intelligent in some sense.

The general form of robot dynamics is described by (1) where $M(\theta)$, $V(\theta,\theta')$, $\tau(t)$ and f stand for the state varying inertia matrix, vector of coriolis terms, applied torque inputs and friction terms respectively. The plant parameters are given in Table 1 in standard units.

$$M(\theta)\theta'' + V(\theta,\theta') = \tau(t) - f \tag{1}$$

Table 1. Manipulator Parameters

Motor 1 Rot. Inertia	0.2670	Payload Mass	0.000
Arm 1 Inertia	0.3340	Arm 1 length	0.359
Motor 2 Rot. Inertia	0.0075	Arm 2 length	0.240
Motor 2 Stat. Inertia	0.0400	Arm 1 CG	0.136
Arm 2 inertia	0.0630	Arm 2 CG	0.102
Payload Inertia	0.0000	Axis 1 Friction	5.300
Motor 1 Mass	73.000	Axis 2 Friction	1.100
Arm 1 Mass	9.7800	Torque Limit 1	245.0
Motor 2 Mass	14.000	Torque Limit 2	39.20
Arm 2 Mass	4.4500		

If the angular positions and angular velocities are described as the state variables of the system, four differential equations being coupled and first order can define the model. In (2) and (3), the terms seen in (1) are given explicitly [14].

$$M(\theta) = \begin{bmatrix} p_1 + 2p_3 cos(\theta_2) & p_2 + p_3 cos(\theta_2) \\ p_2 + p_3 cos(\theta_2) & p_2 \end{bmatrix} \tag{2}$$

$$V(\theta,\theta') = \begin{bmatrix} -\theta_2'(2\theta_1' + \theta_2')p_3 sin(\theta_2) \\ \theta_1'^2 p_3 sin(\theta_2) \end{bmatrix} \tag{3}$$

where, $p_1 = 2.0857$, $p_2 = 0.1168$ and $p_3 = 0.1630$.

4. FUZZY SYSTEM MODEL

The fuzzy system model analyzed in this section is used as the controller. As introduced in the third section, the manipulator has two control inputs and each torque input is evaluated by the use of architecture discussed next. For each link, a separate Fuzzy Logic Controller (FLC) is used with the relevant error and the rate of error as the input variables. In this respect, it should be emphasized that the coupling effects are not taken into consideration by the controller. Clearly, this is a difficulty, which is to be alleviated by the controller.

The fuzzy system employed in this study has the following type of rules in the rule base. In this representation, lowercase variables denote the inputs, uppercase variables stand for the fuzzy sets corresponding to the domain of each linguistic label.

IF u_1 *is* U_1 *AND* u_2 *is* U_2 *AND ... AND* u_m *is* U_m
THEN $f = y_i$

The structure of fuzzy system is illustrated in Fig. 2. In this study, $R=25$, $m=2$ which describe the number of rules contained in the rule base and number of inputs of each fuzzy controller respectively. As the membership functions, Gaussian functions are used as described by (4).

$$\mu_{ij}(u_j) = \exp\left\{ -\left(\frac{u_j - c_{ij},}{\sigma_{ij}} \right)^2 \right\} \tag{4}$$

The overall realization performed by FLC can now be formulated as follows:

$$F = \frac{\sum_{i=1}^{R} y_i w_i}{\sum_{i=1}^{R} w_i} \tag{5}$$

where, the firing strengths denoted by w are evaluated through the use of following aggregation operator proposed by Batyrshin and Kaynak [6].

$$\begin{aligned} w_i &= T(\mu_{i1}, \mu_{i2}, p_i) \\ &= \min(\mu_{i1}, \mu_{i2})(\mu_{i1} + \mu_{i2} - \mu_{i1}\mu_{i2})^{p_i} \end{aligned} \tag{6}$$

The described T-Norm is clearly commutative but not associative.

5. GENETIC FUZZY CONTROL OF DIRECT DRIVE SCARA ROBOT

The control mechanism utilized in this study uses the described fuzzy system directly as the controller whose parameters are tuned by GAs. Two FLCs are employed in the design where the controllers use the related error and the rate of error as the input variables for each link. The structure of the control system is illustrated in Fig. 3 where the algorithmic structure of the adaptation mechanism is depicted in Fig. 4. One can directly infer from the controller design that the coupling effects are not taken into consideration by the controller, i. e., the base axis controller does not use the data observed from elbow link and vice versa. Therefore, the state tracking precision becomes a challenge that is to be achieved by the controller.

The center and width of the membership functions, defuzzifier parameters and the parameters of T-Norms are updated by utilizing GAs discussed in the second section. The population is composed of 30 members. At each run of the controller, 15 best chromosomes are selected and reproduced to give 30-member population. 10 crossover and 5 mutation processes are applied to this population. The mutation is based on simulated-annealing, in which the probability of occurrence of mutation is decreased with the increasing number of runs. Furthermore, the domains of the rules characterizing the local behavior of the fuzzy controller are selected to be adaptive. The algorithm keeps an error and a derivative of error log for both of the links and decreases the size of the search space as the errors get smaller. This increases the resolution of the space on which the rules are activated and thus and improvement on the performance is observed.

6. SIMULATION RESULTS

The fuzzy model used in the simulations has the membership functions described by (4). The adjustable parameters of the controllers are adapted by GAs whose objective is to maximize the performance index given by (7). In this study, a smooth pulse described by (8) is selected as the reference position trajectory. In Fig. 5, state tracking errors are illustrated. The results observed in the sense of tracking precision clearly imply that the learning can be achieved through the use of derivative-free optimization techniques.

$$J = \frac{1}{e^2 + \dot{e}^2} \tag{7}$$

$$r(t) = \tanh(5(t-1)) - \tanh(5(t-3)) + \tanh(5(t-5)) - \tanh(5(t-7)) \tag{8}$$

In the literature, various fuzzy models are proposed and studied in motion control problems. Most of them consider the adaptation of fuzzifier and defuzzifier parameters. The contribution of this paper is to demonstrate whether a satisfactory performance could be observed or not if one imposes learning by tuning the parameters of aggregation operator by GAs. Obviously, the results confirm that the tracking precision, which can also be achieved by the conventional tuning strategies, can be observed by the approach discussed.

7. CONCLUSIONS

In this study, the fuzzy control of a two degrees of freedom direct drive robotic manipulator is presented. The fuzzy system chosen in the work has the standard architecture, which utilizes a parameterized T-Norm in aggregating the rule premises. The learning algorithm has been derived for the use of discussed T-Norm and the tuning is based on genetic algorithms.

Briefly, the approach presented in this paper achieves the tracking precision, and compensates the deficiencies caused by poor modeling. On the other hand, it has some undesirable characteristics in the sense of losing the physical meanings of fuzzy inference mechanism due to the adjustment performed on aggregation of rule premises. The debate on the use of such aggregation methods has not ended yet but the research on the correspondence between this action and the physical domain is still continuing.

8. REFERENCES

[1] Wang, L., *Adaptive Fuzzy Systems and Control*, Design and Stability Analysis, PTR Prentice Hall, 1994, pp. 29-31.

[2] Jang, J.-S. R., C.-T. Sun and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, PTR Prentice Hall, 1997.

[3] Efe, M. O. and O. Kaynak, "A Comparative Study of Soft Computing Methodologies in Identification of Robotic Manipulators," Proc. 3rd Int. Conf. on Advanced Mechatronics, ICAM'98, 3-6 August, vol. 1, pp. 21-26, Okayama, Japan, 1998.

[4] Efe, M. O. and O. Kaynak, "A Comparative Study of Neural Network Structures in Identification of Nonlinear Systems," *Mechatronics Journal*, Elsevier Science Ltd., v.9, (3), 1999, pp.287-300, United Kingdom.

[5] Takagi T., and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Trans. On Systems, Man and Cybernetics*, pp. 116-132, January 1985.

[6] Batyrshin I., O. Kaynak and I. J. Rudas, "Generalized Conjunction and Disjunction Operators for Fuzzy Control," Proc. EUFIT, Sep. 7-10, pp. 52-57, Aachen, Germany, 1998.

[7] Efe, M. O., and O. Kaynak, "A Novel Optimization Procedure for Training of Fuzzy Inference Systems By Combining Variable Structure Systems Technique and Levenberg-Marquardt Algorithm," *Fuzzy Sets and Systems* (submitted for publication).

[8] Efe, M. O., and O. Kaynak, "Stable Training of Computationally Intelligent Systems By Using Variable Structure Systems Technique," *IEEE Transactions on Industrial Electronics* (submitted for publication).

[9] Grant, K., An Introduction to Genetic Algorithms, *C/C++ Users Journal*, p. 45, March 1995.

[10] Tang K. S., K. F. Man, S. Kwong and Q. He, Genetic Algorithms and their Applications, IEEE Signal Processing Magazine, p. 22,

November 1996.

[11] Kaynak, O., L. A. Zadeh, B. Turksen and I. J. Rudas, Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications, 1998.

[12] Deb, K. "Genetic Algorithms for Function Optimization", eds. O. Kaynak, L. A. Zadeh, B. Turksen and I. J. Rudas, Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications, 1998.

[13] Pal, S. K. and P. P. Wang, Genetic Algorithms for Pattern Recognition, CRC Press, USA, 1996.

[14] Direct Drive Manipulator R&D Package User Guide, Integrated Motions Incorporated, 704 Gillman Street, Berkeley, California 94710, U.S.A.

ACKNOWLEDGMENTS

This work is supported in part by a grant of Foundation for Promotion of Advanced Automation Technology and Bogazici University Research Fund, Project No: 97A0202 and 99A202.

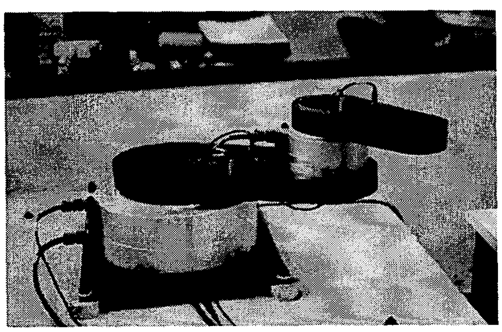

Figure 1. Physical View of the Direct Drive Robotic Manipulator

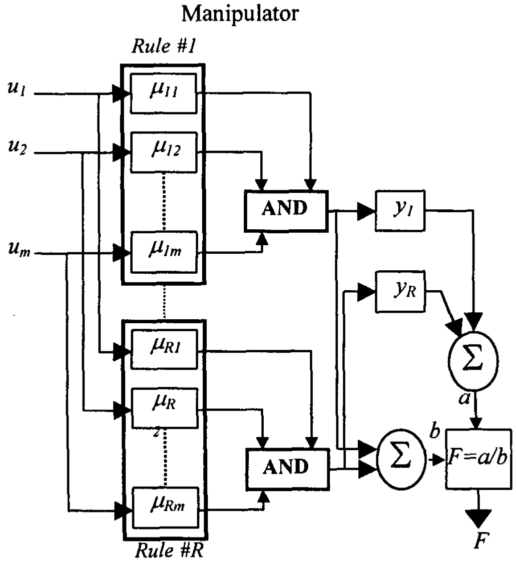

Figure 2. Structure of the Fuzzy System

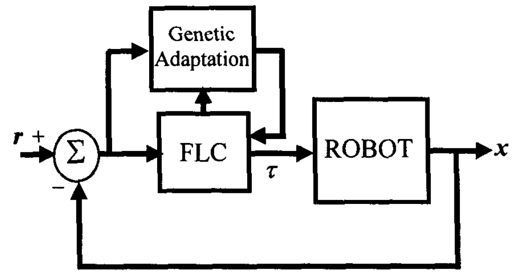

Figure 3. Structure of the Control System

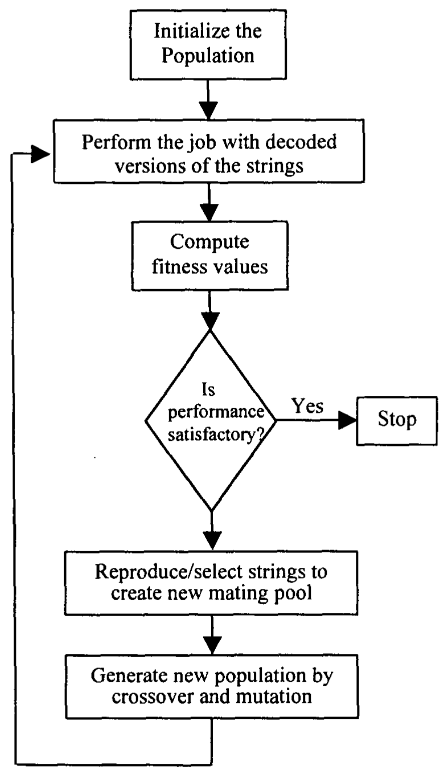

Figure 4. Basic steps of a genetic algorithm [11]

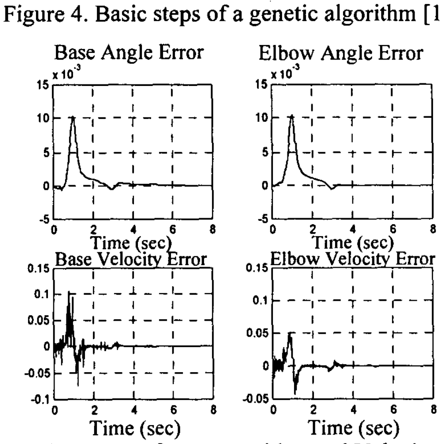

Figure 5. Reference Position and Velocity Trajectories