Techniques in Neural Network Based Fuzzy System Identification and Their Application in Control of Complex Systems

Yaochu Jin[§,¶] Jingping Jiang[§]
[§] Department of Electrical Engineering, Zhejiang University
Hangzhou 310027, P. R. China
[¶] Institute for Neuroinformatics, Ruhr-University Bochum
D-44780 Bochum, Germany

Abstract

Identification of fuzzy systems with artificial neural networks is discussed in this chapter. By use of an updated version of pi-sigma neural network, both premise and consequent parameters of the fuzzy system can be efficiently identified on-line or off-line. Learning algorithms for both Gaussian and triangular forms of membership functions are presented. The consequent part of the fuzzy rules is represented by a sub-network, which enables the algorithm applicable to high order Takagi-Sugeno fuzzy systems. Some measures are taken to preserve the interpretability of the fuzzy system in the course of learning. The proposed method is applied to the nonlinear decoupling control of robot manipulators and satisfactory simulation results are obtained.

I. Introduction

Two basic forms of fuzzy rules, namely, Mamdani fuzzy rules and Takagi-Sugeno fuzzy rules, have been developed to date. The main difference between these two types of fuzzy rules lies in the fact that the consequent part of the Takagi-Sugeno fuzzy rules is normally a concrete linear function of input variables instead of some fuzzy linguistic variables. Generally speaking, Takagi-Sugeno fuzzy rule systems are more flexible and thus have stronger modeling capability to solve some complex problems. Theoretically, fuzzy rules can be built based either on expert knowledge or on a group of observed data. However, it is very difficult, if not impossible, for human beings to establish an acceptable fuzzy rule system when the input dimension of the system becomes high. Therefore, building fuzzy rule systems on the basis of collected data is becoming more and more important.

We limit our attention to identification of Takagi-Sugeno fuzzy rule systems in this chapter. It is noticed that identification methods for such fuzzy systems have been proposed by Takagi and Sugeno[1-2]. The algorithms are quite complicated and are mainly suitable for fuzzy rules with piece-wise linear fuzzy membership functions and linear consequents. Moreover, their algorithms have difficulties in real-time implementation, which has limited its application seriously.

In the recent years, methodologies in artificial neural networks, fuzzy systems and evolutionary computation have been successfully combined and new techniques called soft computing or computational intelligence have been developed. These techniques are attracting more and more attention in several research fields because they are able to tolerate wide range of uncertainty. Use of neural networks to performe the adjustment of membership functions and modification of the consequent of fuzzy rules makes it practical to design adaptive fuzzy models and self-organizing fuzzy control. Different neural networks, such as backpropagation networks[3], RBF neural networks[4], hybrid pi-sigma networks[5], B-spline networks[6] and neural-like structure[7], have been applied to adaptation of fuzzy membership functions and consequent parameters.

With all these successes, it is also necessary to point out that some important features of fuzzy systems have been lost. One common problem for most neurofuzzy algorithms is that the interpretability of fuzzy systems is deteriorated. After adaptation, either the distinguishability among the fuzzy subsets in a fuzzy partitioning is blurred, or the fuzzy partitionings of the input space are incomplete. In fact, most neurofuzzy schemes have failed to pay attention to preserve the rule structure of the fuzzy systems in the course of adaptation. This problem has been alleviated in [5] by introducing the hill-climbing method. Further discussions on the interpretability of fuzzy systems, including the completeness and consistency considerations, can be found in [8][9].

This chapter is an extension of the work in [5]. A systematic approach to identification of the Takagi-Sugeno fuzzy systems is described. The algorithm has been applied to a wide range of modeling and control problems and inspiring results have been achieved.

II. Takagi and Sugeno's Fuzzy Model

Takagi and Sugeno[1-2] proposed a new fuzzy model by replacing the linguistic variables in the THEN-part of the fuzzy rules with a crisp linear function of the input variables. Since a multi-input multi-output(MIMO) fuzzy system can always be separated into a group of multi-input single-output (MISO) systems, we discuss here only the MISO fuzzy systems without the loss of generality. Given an MISO system with n inputs, the Takagi-Sugeno fuzzy rules have the following form:

$$R_i: \text{ If } x_1 \text{ is } A_1^i \text{ and ... and } x_n \text{ is } A_n^i \text{ , then } y^i = p_0^i + p_1^i x_1 + ... + p_n^i x_n \tag{1}$$

where $R_i (i = 1, 2, ..., N)$ denotes the i-th rule, $x_j (j = 1, ..., n)$ are the premise variables, A_j^i are the fuzzy subsets defined by corresponding piecewise linear membership functions such as triangle or trapezoid, y_i is the consequent of the i-th rule. According to Takagi and Sugeno, the final output of the fuzzy system can be written in the following form:

$$y = \frac{\sum_{i=1}^{N} (w^i y^i)}{\sum_{i=1}^{N} w^i} \tag{2}$$

where w^i is calculated by

$$w^i = t_{j=1}^{n} A_j^i (x_j) \tag{3}$$

where, t represents the t-norm operator. Currently, a number of fuzzy t-norms, as well as some extended forms[10] have been proposed. Discussions on some widely used t-norms are given in [11-12] among others. In this work, the most widely used minimum operator is considered.

If some pairs of input and output data are available, the parameters in THEN-part of the rules can be estimated using the least square method. However, estimation of the parameters of the membership functions are a little complicated, even if the membership functions are supposed to be piecewise linear[1-2].

The Takagi-Sugeno fuzzy model is believed to be more flexible than the Mamdani fuzzy model and has found wider application in the recent years. Nevertheless, two general shortcomings still exist. First, identification of the fuzzy system is not trivial and therefore it is hard to apply the fuzzy system to real-time systems. Second, not only the membership functions are limited to piecewise functions, but the consequent part is also assumed to be linear. This problem remains unsolved until neural networks are combined systematically with fuzzy systems and the so-called neurofuzzy system theory appears. Such hybrid systems are preferred because they possess the main features of both fuzzy and neural systems. That is to say, they have clear physical meanings and easy to interpret like conventional fuzzy systems, on the other hand, they have good learning ability and nonlinear mapping capacity. Despite that, special attention should be paid in the course of learning so that both of the merits could be preserved.

III. Neural Network Based Identification of Fuzzy Systems

A. Hybrid Neural Networks

Most researchers use multiplication as t-norm so that the conventional multiple layered perceptions or RBF neural networks can be directly implemented. In fact, the Gaussian based RBF neural networks with some minor conditions are shown to be mathematically equivalent to fuzzy systems. One condition is that the multiplication operator should be used as the fuzzy t-norm. In our work, however, minimum operator is kept. To this end, the neural network used to identify the fuzzy system should contain not only summing and multiplication neurons, but also fuzzy neurons that are able to perform fundamental fuzzy operations such as minimum operation. Therefore, a hybrid neural network structure inspired by the pi-sigma neural network[13] is suggested. For clarity, a neural network with two inputs is illustrated in Fig. 1. In Fig. 1, a shaded circle denotes a summing neuron with nonlinear activation function, while the blank circle represents a neuron without nonlinear activation, Π means the product neurons and Λ represents the minimum neurons. From Fig.1, it can also be noticed that the consequent part of the fuzzy rules is implemented by a feedforward sub-network. Suppose the sub-network has one hidden layer with H neurons, the overall output of the hybrid neural network can be written by:

$$y = \frac{\sum_{i=1}^{N}(w^i y^i)}{\sum_{i=1}^{N} w^i} \tag{4}$$

$$y^i = p_i^{(0)} + \sum_{k=1}^{H}\left(p_{ki}^{(2)} \cdot g\left(\sum_{j=1}^{n} p_{jk}^{(1)} x_j\right)\right) \tag{5}$$

$$w^i = min_{j=1}^{n} A_j^i(x_j) \tag{6}$$

where, $g(\cdot)$ is a sigmoid function, $\mathbf{p}^{(0)}$, $\mathbf{p}^{(1)}$ and $\mathbf{p}^{(2)}$ are the weights in the sub-network. It is noticed that the output layer of the sub-network is linear.

The hybrid neural network has very clear physical meanings. It is easy to observe that the neural system described by Eqs. (4)-(6) is functionally equivalent to the Takagi-Sugeno fuzzy system expressed in Eqs. (2)-(3), except for the fact that the linear consequent functions in the Takagi-Sugeno fuzzy model has been extended to general nonlinear functions described by a neural network. In addition, the membership functions can be arbitrary continous functions that satisfy the definitions for fuzzy memberships. In practice, triangle, trapezoid and Gaussian functions are most widely used, although recently, spline[14] and polynominal[15] functions are also suggested as membership functions. However, they sometimes do not satisfy the definitions for fuzzy memberships, and they are in most cases not normal.

B. Identification Algorithms Based on Neural Networks

Since the hybrid neural network system consists of minimal nodes that are not differential, the gradient method based backpropagation algorithm can not be directly applied. There are some alternatives to deal with this problem. In this work, two different approaches are suggested. The first approach carries out an equivalent transformation of the minimum operator, so that it can be treated as differential and thus the gradient method can be applied. The other approach uses an one step backward searching algorithm to avoid the differential operation on the minimum nodes. To prevent the fuzzy subsets from losing their prescribed physical meanings, some additional measures are suggested in Section III.

1. Equivalent Transformation of the Minimum Operator

It is well known that the gradient method can only be applied to differentiable functions. Therefore, functions with minimum operators do not satisfy this condition. To solve this problem, the minimum operator will be transformed equivalently so that the gradient method could be used to derive the learning algorithm for identifying the fuzzy system. For the minimum operation with n elements:

$$w^i = min\{A_1^i(x_1), A_2^i(x_2), \cdots, A_n^i(x_n)\} \tag{7}$$

we have,

$$w^i = \sum_{j=1}^{n} \prod_{m \neq j} \bigcup [A_m^i(x_m) - A_j^i(x_j)] \tag{8}$$

where,

$$\bigcup [A_m^i(x_m) - A_j^i(x_j)] = \begin{cases} 1 & , \quad A_m^i(x_m) > A_j^i(x_j) \\ 0 & , \quad A_m^i(x_m) \leq A_j^i(x_j) \end{cases} \tag{9}$$

In this way, the learning algorithm can be derived based on the gradident method. Let y^d the desired output of the system, and define the following quadratic cost function:

$$E = \frac{1}{2}(y - y^d)^2 \tag{10}$$

thus,

$$\frac{\partial E}{\partial p_i^{(0)}} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial y^i}\frac{\partial y^i}{\partial p_i^{(0)}} = (y - y^d)w^i / \sum_{i=1}^{N} w^i \tag{11}$$

$$\frac{\partial E}{\partial p_{jk}^{(1)}} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial y^i}\frac{\partial y^i}{\partial p_{jk}^{(1)}} = (y - y^d)w^i \sum_{k=1}^{H} \{p_{ki}^{(2)} g\prime(\cdot) x_j\} / \sum_{i=1}^{N} w^i \tag{12}$$

$$\frac{\partial E}{\partial p_{ki}^{(2)}} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial y^i}\frac{\partial y^i}{\partial p_{ki}^{(2)}} = (y - y^d)w^i g(\cdot) / \sum_{i=1}^{N} w^i \tag{13}$$

According to Eqs. (11)-(13), the learning algorithm of the consequent parameters could be expressed as follows:

$$\Delta p_i^{(0)} = -\eta \delta_1^i \tag{14}$$

$$\Delta p_{jk}^{(1)} = -\eta \sum_{k=1}^{H} \{p_{ki}^{(2)} g\prime(\cdot) x_j\} \delta_1^i \tag{15}$$

$$\Delta p_{ki}^{(2)} = -\eta g(\cdot) \delta_1^i \tag{16}$$

where η is a positive learning rate and the general error δ_1^i is defined by:

$$\delta_1^i = (y - y^d) w^i / \sum_{i=1}^{N} w^i \tag{17}$$

Next, we will derive the learning algorithms for the membership parameters. As indicated above, Gaussian and triangular functions are most widely used fuzzy membership functions. Due to this reason, we first suppose the membership functions are Gaussians in the following form:

$$A_j^i(x_j) = \exp(-(x_j - a_j^i)^2 / b_j^i) \tag{18}$$

thus we have:

$$\frac{\partial E}{\partial a_j^i} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial w^i} \frac{\partial w^i}{\partial A_j^i(x_j)} \frac{\partial A_j^i(x_j)}{\partial a_j^i} \tag{19}$$

Since,

$$\frac{\partial A_j^i(x_j)}{\partial a_j^i} = \frac{\partial \{exp(-(x_j - a_j^i)^2 / b_j^i)\}}{\partial a_j^i} = 2A_j^i(x_j)(x_j - a_j^i)/b_j^i \tag{20}$$

Reminding the transformation of the minimum operation in equation (8) and equation (9), it is straightforward that:

$$\begin{aligned} \frac{\partial w^i}{\partial A_j^i(x_j)} &= \frac{\partial \{\sum_{j=1}^{n} A_j^i(x_j) \prod_{m \neq j} \bigcup [A_m^i(x_m) - A_j^i(x_j)]\}}{\partial A_j^i(x_j)} \\ &= \bigcup_{m \neq j} [A_m^i(x_m) - A_j^i(x_j)] = \left\{ \begin{array}{lll} 1 & , & \text{if } A_j^i(x_j) \text{ minimum;} \\ 0 & , & \text{else} \end{array} \right. \end{aligned} \tag{21}$$

Substitute Eq. (20) and Eq. (21) into Eq. (19) and if we define:

$$\delta_2^i = (y - y^d)(y^i - y) / \sum_{i=1}^{N} w^i \tag{22}$$

then the learning algorithm for the membership parameters a_j^i can be written by:

$$\Delta a_j^i = \begin{cases} -2\xi(x_j - a_j^i)w^i\delta_2^i/b_j^i & ; \quad \text{if } A_j^i(x_j) \text{ minimum} \\ 0 & ; \quad \text{else} \end{cases} \tag{23}$$

Similarily, the learning algorithm for b_j^i:

$$\Delta b_j^i = \begin{cases} -\xi w^i(x_j - a_j^i)^2\delta_2^i/(b_j^i)^2 & ; \quad \text{if } A_j^i(x_j) \text{ minimum} \\ 0 & ; \quad \text{else} \end{cases} \tag{24}$$

where ξ is a positive learning rate.

Now we consider the situation where the membership functions are triangular. For the sake of simplicity, we assume the membership functions are all isosceles as shown in Fig.2. In this case, it can be described by:

$$A_j^i(x_j) = 1 - \frac{2|x_j - a_j^i|}{b_j^i} \tag{25}$$

Hence, the learning algorithm of a_j^i and b_j^i can be described by the following equations:

$$\Delta a_j^i = \begin{cases} -2\xi sgn(x_j - a_j^i)\delta_2^i/b_j^i & ; \quad \text{if } A_j^i(x_j) \text{ minimum} \\ 0 & ; \quad \text{else} \end{cases} \tag{26}$$

$$\Delta b_j^i = \begin{cases} -\xi[1 - w^i]\delta_2^i/b_j^i & ; \quad \text{if } A_j^i(x_j) \text{ minimum} \\ 0 & ; \quad \text{else} \end{cases} \tag{27}$$

2. One-step Backward Searching Algorithm

In the above described method, the minimum operation is equivalently transformed so that gradient method can be used to derive the learning algorithms. However, if we have a close look at the transformation, we find that this transformation leads the gradient search toward the input node of the fuzzy neuron that has the minimum value. This works if the initial distribution of the membership functions agree with the real distributions approximately. However, if the initial distributions have significant differences with the real distributions, this algorithm may fail to find the optimal solutions. To expound this, consider a fuzzy neuron that implements minimum operation as in Fig.3. In Fig.3, we have

$$O = min\{A, B, C\} \tag{28}$$

If $A = 0.5$, $B = 0.4$, $C = 0.3$, then $O = min(A, B, C) = 0.3$. Suppose the desired value of the node output is 0, then the error will be -0.3. According to the leraning algorithm developed above, the input weight of connection C will be adjusted because C has the minimum value. However, it is easy to notice that the error of the fuzzy neuron is not necessarily caused by weight C. It may be the case that the truth value of B should be 0 and the error is fully caused by input weight B. To cope with this situation, we introduce the hill-climbing searching method, which is a counterpart of the gradient method and does not require the differentiability of the cost function. The learning algorithm will be carried out in the following two phases. At first, all the input weights A, B and C are updated simultaneously if there exists an error of e at the output O. Then comparisons are made to see which modification reduces the error most significantly. As supposed above, if the error before learning is -0.3 and after adjusting weight A, B and C, the new errors are -0.25, -0.05 and -0.2 respectively. It is found that the smallest error is obtained by updating weight B. Consequently, only weight B is actually modified.

IV. Interpretability Considerations

Given the above introduced learning algorithms, the rule parameters can be identified on-line without any difficulty. However, if no other constraints are imposed, some problems may appear, especially when adjusting the membership parameters. The problems that appear most frequently are:

- Two neighbouring fuzzy subsets in a fuzzy partitioning have no overlapping and consequently the partitioning is incomplete.
- The membership functions of two fuzzy subsets are so similar that the distinguishability of the fuzzy partitioning is lost. This can not only make the fuzzy system unnecessarily complicated, but also give rise to difficulties in assigning suitable physical meanings to them.
- The membership functions lose their prescribed physical meanings. For example, the center

value of “small” may preceed that of “big”.

All these phenomena harm the interpretability of fuzzy systems. To avoid these problems, we suggest here some constraints on the membership parameters which should be checked on-line during learning. First of all, to keep the prescribed physical meanings, the center values of each fuzzy subset in one fuzzy partitioning should satisfy the following condition:

$$a_i^1 < a_i^2 < \cdots < a_i^n \tag{29}$$

where we suppose the fuzzy partitioning of x_i is composed of n fuzzy subsets $\{A_i^1(x_i),\ A_i^2(x_i),\ \cdots,\ A_i^n(x_i)\}$. Before we introduce the conditions to guarantee the completeness and distinguishability of the fuzzy partitionings, we first introduce the concept of *fuzzy similarity measure* between two fuzzy subsets:

$$FSM(A, B) = \frac{M(A \cap B)}{M(A) + M(B) - M(A \cup B)} \tag{30}$$

where $M(A)$ is called the size of fuzzy set A and is calculated as:

$$M(A) = \int_{x \in U} A(x)dx \tag{31}$$

where U is the universe of discourse of x. It is noticed that if $FSM(A, B) = 0$, the two fuzzy sets have no overlapping, on the contrary, they are completely equal if $FSM(A, B) = 1$. Therefore, the fuzzy partitioning is complete if any of its two neighbouring fuzzy subsets satisfy:

$$FSM(A_i^j, A_i^{j+1}) > 0 \tag{32}$$

To ensure a good distinguishability of a fuzzy partitioning, the following condition should hold for two arbitrary fuzzy subsets in the fuzzy partitioning:

$$FSM(A_i^j, A_i^k) < \delta \tag{33}$$

where $0 < \delta < 1$ is a constant to be determined.

V. An Application Example: Decoupled Control of Robot Manipulators

Dynamic control of robot manipulators is a challenging task in the field of system control. Various modern control strategies have been widely investigated to deal with the high nonlinearity and strong coupling of the robot dynamics. In this chapter, we try to control the robot manipulators using Takagi-Sugeno fuzzy models.

Although Takagi-Sugeno fuzzy rules are believed to have stronger mapping ability than the Mamdani fuzzy rules, they are much more complex, especially when the input dimension is high. For example, for a rigid robot with N-degree-of-freedom, there are $3N$ input parameters (link position, velocity and acceleration) and if each input is divided into 6 fuzzy subspaces, there will be 6^{3N} fuzzy rules in total. Suppose first-order Takagi-Sugeno rules are adopted, then each rule consists of $N \times (3N+1)$ consequent parameters and consequently there are $N(3N+1)6^{3N}$ parameters to be estimated. If $N = 6$, the number is about 1.16×10^{16}, which is very huge and makes it impossible for real-time implementation. It is therefore sensible to decouple the robot dynamics before we apply the Takagi-Sugeno fuzzy model to robot control. In the decoupled robot dynamics, there are only two input variables, namely, position and velocity, for each link. In this case, the number of parameters to be estimated will be greatly reduced. In the above example, only $N \times 6^2$ fuzzy rules are needed, each consisting of 3 consequent parameters. When N is 6, the total number of parameters are only 648.

A. Decoupling of the Robot Dynamics

Two main approaches are used by most researchers to derive the dynamics of robot manipulators, namely, Lagrange-Euler and Newton-Euler formulations. From the control point of view, the Lagrange-Euler formulation is very desirable. For an N-degree-of-freedom rigid robot, the Lagrange equation of motion is as follows:

$$\tau = H(q)\ddot{q} + M(q, \dot{q}) + G(q) \tag{34}$$

where τ is N-dimension torque vector, $H(q)$ is $N \times N$ inertia matrix, $M(q, \dot{q})$ is N-dimension coriolis and centrifugal force vector, $G(q)$ is N dimensional gravity vector and q, $\dot{q}$ and $\ddot{q}$ are N

dimensional angular, velocity and acceleration respectively. Let $x_i = q_i$, $x_{N+i} = \dot{q}_i (i = 1, 2, ..., N)$, then the robot dynamics can be written as:

$$\dot{X} = A(X) + B(X)U \tag{35}$$

$$Y = C(X) \tag{36}$$

where $X = [X_1 X_2]^T = [x_1, ..., x_N, x_{N+1}, ..., x_{2N}]^T$, $U = [\tau_1, ..., \tau_N]^T$, $C(X) = X_1$, and

$$A(X) = \begin{bmatrix} X_2 \\ -H^{-1}(M+G) \end{bmatrix}, B(X) = \begin{bmatrix} 0 \\ -H^{-1} \end{bmatrix} \tag{37}$$

Define the following operator [16]:

$$N_A^j C_i(X) = [\frac{\partial}{\partial X} N_A^{j-1} C_i(X)] A(X), j = 1, 2, ..., N-1 \tag{38}$$

$$N_A^0 C_i(X) = C_i(X) \tag{39}$$

where $C_i(X)$ is the i-th row of $C(X)$. Define the relative degree of the system:

$$d_i = min_j \{ [\frac{\partial}{\partial X} N_A^{j-1} C_i(X)] B(X) \neq 0 \}, j = 1, 2, ..., N \tag{40}$$

then we have the following control that decouples the robot dynamics:

$$U = F(X) + G(X)V \tag{41}$$

where V is the new control vector of the decoupled linear system, and

$$F(X) = -(D^\star)^{-1}(X)(F_1^\star(X) + F_2^\star(X)) \tag{42}$$

$$G(X) = -(D^\star)^{-1}(X)\Lambda \tag{43}$$

$$D_i^\star(X) = \frac{\partial}{\partial X}[N_A^{d_i} C_i(X)] B(X) \tag{44}$$

$$F_{1i}^\star(X) = N_A^{d_i} C_i(X) \tag{45}$$

$$F_{2i}^\star(X) = \sum_{k=1}^{d_i - 1} a_{k,i} N_A^k C_i(X) \tag{46}$$

$$\Lambda = diag[\lambda_1, ..., \lambda_N] \tag{47}$$

where $D_i^\star(X)$, $F_{1i}^\star(X)$ and $F_{2i}^\star(X)$ are the i-th row of matrix $D^\star(X)$, $F_1^\star(X)$ and $F_2^\star(X)$ respectively, $a_{k,i}$ are some constants to be determined. For the robot system given in Eq. (35), since

$$\frac{\partial}{\partial X}[N_A^0 C_i(X)]B(X) = 0 \tag{48}$$

and

$$\frac{\partial}{\partial X}[N_A^1 C_i(X)]B(X) \neq 0 \tag{49}$$

therefore the relative degree of the system is

$$d_i = 2, i = 1, 2, ..., N \tag{50}$$

Thus we have the following decoupled linear model for the robot system:

$$\begin{bmatrix} \dot{x}_i \\ \vdots \\ \dot{x}_N \\ \dot{x}_{N+1} \\ \vdots \\ \dot{x}_{2N} \end{bmatrix} = \begin{bmatrix} x_{N+1} \\ \vdots \\ x_{2N} \\ -a_{0,1}x_1 - a_{1,1}x_N \\ \vdots \\ -a_{0,N}x_N - a_{1,N}x_{2N} \end{bmatrix} + \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ & & \lambda_N \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix} \tag{51}$$

It is straightforward that the sub-system for each link is a time-constant two-input single-output linear system. The parameters of the linear system should be chosen in such a way that the linear sub-systems are stable.

As we have mentioned above, the decoupled linear systems are time-constant if the dynamics of the robot is exactly known. In this case, a conventional PD controller will perform successfully. However, there are always parameter errors in real robotic systems. Moreover, it is difficult to model such dynamics as nonlinear friction, backlash and other uncertainties in robot systems. Therefore, adaptive fuzzy controlers are necessary to deal with the uncertainties.

B. Simulation Study

For the sake of simplicity, a two-degree-of-freedom rigid manipulator is studied in this simulation. The dynamics of the system is expressed as follows:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \tag{52}$$

where

$$H_{11} = (m_1 + m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 cos(q_2) \tag{53}$$

$$H_{12} = H_{21} = m_2 l_2^2 + m_2 l_1 l_2 cos(q_2) \tag{54}$$

$$H_{22} = m_2 l_2^2 \tag{55}$$

$$M_1 = -2m_2 l_1 l_2 sin(q_2)\dot{q}_1 \dot{q}_2 - m_2 l_1 l_2 sin(q_2)\dot{q}_2^2 \tag{56}$$

$$M_2 = m_2 l_1 l_2 sin(q_2)\dot{q}_1^2 \tag{57}$$

$$G_1 = (m_1 + m_2) g l_1 cos(q_1) + m_2 g l_2 cos(q_1 + q_2) \tag{58}$$

$$G_2 = m_2 g l_2 cos(q_1 + q_2) \tag{59}$$

where m_1 and m_2 are the mass of the two links, l_1 and l_2 are the link lengths, and g is the gravity. Without the loss of generality, the parameters of the linearized model is chosen as follows:

$$0.1\ddot{q}_i + \dot{q}_i = V_i \ (i = 1, 2) \tag{60}$$

The two inputs are $x_1^i = \dot{q}_i$, $x_2^i = q_i^d - q_i$ and the output is $y_i = v_i (i = 1, 2)$. The diagram of the control system is shown in Fig. 4. The desired trajectories for the two links are

$$q_1^d(t) = \ \exp(0.5t)(\text{rad}) \tag{61}$$

$$q_2^d(t) = 0.5 + \ \exp(0.4t)(\text{rad}) \tag{62}$$

In order to observe how the controler behaves in the presence of various uncertainties, two cases of uncertainties, namely, parameter variation and unmodeled friction, are considered.

1. Parameter variations

We first suppose both the mass and the length of the two links have an error of 10%. After 10 iterations of learning, the tracking errors of the two links are acceptable(see Fig. 5), noticing that there exist initial position errors of 0.2(rad) and 0.1(rad). The membership functions of link1 and link 2 are provided in Fig. 6 and Fig. 7, where the membership functions in (a) are for input

x_1^i, whilst those in (b) are for x_2^i. It is seen that all the fuzzy partitionings are complete and fairly distinguishable thanks to the measures that are taken to preserve the interpretability.

2. Unmodeled friction

The Lagrange model of robot dynamics usually does not consider the nonlinear friction. In order to investigate the performance of the controller in the presence of unmodeled nonlinear friction, the following nonlinear friction is added in simulation:

$$f_i = f_{q_i}(\dot{q}_i, \tau_i) + f_{v_i}(\dot{q}_i) \ (i = 1, 2) \tag{63}$$

where f_{q_i} and f_{v_i} are the Columbus and viscous friction respectively, which can be expressed by:

$$f_{q_i}(\dot{q}_i, \tau_i) = \begin{cases} k_i sgn(\dot{q}), & |\dot{q}_i| > 0 \\ k_i sgn(\tau_i), & |\dot{q}_i| = 0, \quad |\tau_i| > k_i \\ \tau_i, & |\dot{q}_i| = 0, \quad |\tau_i| < k_i \end{cases} \tag{64}$$

$$f_{v_i}(\dot{q}_i) = C_i \dot{q}_i \tag{65}$$

where, k_i and C_i are constants. In simulation, we set $[k_1,k_2] = [0.2,0.05]$, $[C_1,C_2] = [0.05,0.01]$. The tracking errors of the two links are presented in Fig. 8 and the membership functions are demonstrated in Fig. 9 and Fig. 10 respectively.

VI. Conclusions

Identification methods for Takagi-Sugeno fuzzy systems based on neural networks are proposed in this chapter. These methods are suitable for on-line learning and applicable to both zero-order and high-order Takagi-Sugeno fuzzy systems. To maintain the interpretability of fuzzy systems, some measures are taken in the process of learning. The effectiveness of the proposed methods are shown by simulation studies on control of robot manipulators in the presence of various uncertainties.

In this work, the structure of the fuzzy system is fixed beforehand. However, such a standard rule structure is usually not optimal. The optimization of the rule structure can be realized either by introducing neural network pruning techniques[17] or by using evolutionary computation[18].

References

[1] Sugeno, M. and Tanaka, K.(1991). Successive identification of a fuzzy model and its application to prediction of a complex system. *Fuzzy Sets and Systems* **42**, 315-324.

[2] Takagi, T. and Sugeno, M.(1985). Fuzzy identification of systems and its application to modeling and control. *IEEE Trans. Syst., Man, Cyber.* **14**, 349-355.

[3] Lin, C. T. and Lee, C. S. G.(1991). Neural network based fuzzy logic control and decision systems. *IEEE Trans. Comput.* **40**, 1320-1336.

[4] Jang, J.-S. R.(1993). ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Trans. Syst., Man, Cyber.* **23**, 665-685.

[5] Jin, Y. C., Jiang, J. P. and Zhu J.(1995). Neural network based fuzzy identification and its application to modeling and control of complex systems. *IEEE Trans. Syst., Man, Cyber.* **25**, 990-997.

[6] Hunt, K. J., Haas, R. and Brown, M.(1995). On the functional equivalence of fuzzy inference systems and spline-based networks. *Int. J. Neural Syst.* **6**, 171-184.

[7] Berenji, H. R. and Khedkar, P.(1992). Learning and tuning fuzzy controllers through reinforcement. *IEEE Trans. Neural Networks* **3**, 724-739.

[8] Jin, Y. C., v. Seelen, W. and Sendhoff, B.(1997). Generating FC^3 fuzzy rule system from data using evolutionary strategies. *Submitted to IEEE Trans. Syst., Man, and Cyber.*

[9] Jin, Y. C., v. Seelen, W. and Sendhoff, B.(1997). An approach to rule-based knowledge extraction. *Submitted to WCCI'98*

[10] Yager, R. R. and Feliv D. P.(1994). Analysis of flexible structured fuzzy logic controllers. *IEEE Trans. Syst., Man, Cyber.* **24**, 1035-1043.

[11] Gupta, M. M. and Qi, J.(1991). Design of fuzzy logic controllers based on generalized t-operators. *Fuzzy Sets Syst.* **40**, 473-489.

[12] Jin, Y. C. and v. Seelen, W.(1998). Evaluating flexible fuzzy controllers via evolution strate-

gies. *Fuzzy Sets Syst.* (accepted)

[13] Durbin, R. and Rumelhart, D. E.(1989). Product units: A computationally powerful and biologically plausible extension to backpropagation networks. *Neural Computation* **1**, 133-142.

[14] Brown, M. and Harris, C. J.(1994). "Neurofuzzy Adaptive Modeling and Control", Prentice Hall, Hemel Hempstead.

[15] Runkler, T. A. and Bezdek, J. C.(1996) Polynomial membership functions for smooth first order Takagi-Sugeno systems. In "Proc. in Artificial Intelligence"(C. Freksa, Ed.), pp.382-387., Infix, Augustin.

[16] Freud, E. (1975). The structure of decoupled nonlinear system. *Int. J. Contr.* **21**, 443-450.

[17] Russell, R.(1993). Pruning algorithms – A survey. *IEEE Trans. Neural Networks* **4**, 740-747.

[18] Jin, Y. C. (1998). Decentralized adaptive fuzzy control of robot manipulators. *IEEE Trans. Syst., Man, Cyber.* **28**, 1-11.

Figure Lengends