2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Adaptive Fuzzy Modelling and Control for Discrete-Time Nonlinear Uncertain Systems

Ruiyun Qi and Mietek A. Brdys

Abstract— **This paper presents an adaptive fuzzy modelling and control scheme for discrete-time nonlinear uncertain systems. The proposed adaptive scheme includes two parts: on-line fuzzy modelling using Takagi-Sugeno (T-S) fuzzy systems and model reference adaptive control design. The T-S fuzzy model has a self-organizing structure, i.e. the fuzzy rules can be added, replaced or deleted automatically via on-line clustering and the consequence parameters of the T-S model can be recursively updated by recursive least square estimation (RLSE) method, which allows it to identify complex nonlinear uncertain systems on-line. The adaptive controller is based on the T-S fuzzy model employed as a dynamic model of the plant. The controller can adaptively generate control signals while the structure and parameters of the T-S model are updated on-line. The effectiveness of our approach is verified by its applications in the identification of a second-order nonlinear uncertain system and the tracking control of a single robot arm.**

I. INTRODUCTION

Since the mid 1980s, fuzzy logic techniques have been widely applied to modelling and control of complex nonlinear plants. Theoretical justification of fuzzy model as a universal approximator has been given in the last decade [1],[3],[7]. Among various kinds of fuzzy systems, there is a very important class called Takagi-Sugeno (T-S) fuzzy systems which have recently become a powerful practical engineering tool for modelling and control of complex systems. The T-S fuzzy systems have been proven to be a powerful and efficient solution in the discipline of nonlinear system modelling and control. Their multiple model structure makes them capable of approximating nonlinear dynamic systems. They provide significant parameter and structure variations which can deal with large uncertainties and strong nonlinearities.

Many approaches on using T-S fuzzy models in modelling and control of nonlinear uncertain systems have been developed [2], [4]-[6],[8]. In [2], they present an approach to generating adaptive fuzzy-neural models, which combines structure and parameter identification of T-S fuzzy models. In [8], methods to exact T-S fuzzy models from fuzzy clusters obtained by Gath-Geva clustering are presented. In [11] and [12], genetic algorithms are used for training model structure and parameters. These methods are found to be quite useful in system identification and control but off-line learning is required. They suppose that all the data is available at the start of training and their iterative and time-consuming re-training procedures make real-time implementation very difficult or impossible. In [4], an approach to the on-line learning of T-S models is proposed. It is based on a novel learning algorithm that recursively updates T-S model structure and parameters, which seems very attractive for nonlinear system modelling and adaptive control. However, using its criteria for modifying fuzzy rule base, in some situation, the number of fuzzy rules grows very large because its criteria doesn't consider the condition that some inefficient fuzzy rules need to be deleted from the rule base. In [5] and [6], they present robust adaptive fuzzy controllers suitable for identification and control of nonlinear uncertain systems. However, it is assumed that a part of the system function and full state vector of the controlled plant is available from measurements. Therefore, design of robust adaptive controllers for real-time modelling and control of nonlinear uncertain systems without any prior knowledge about their mathematical model remains one of the most challenging tasks for many control engineers, especially when some of the system states are unmeasurable.

This motivates us to investigate an adaptive modelling and control scheme which can provide a dynamic and accurate model for the unknown plant via on-line learning algorithm and construct an adaptive model reference controller which can generate proper control signals for the plant. In this paper, we propose an approach to modelling and control of nonlinear uncertain systems simultaneously and automatically. The modelling algorithm generates an optimal T-S fuzzy model from one rule, using on-line clustering method to upgrade model structure and recursive least square estimation (RLSE) method to adjust the consequent parameters. The controller generates control signals based on the current updated fuzzy model using feedback linearization. Simulation results show this approach is effective and efficient.

The rest of the paper is organized as follows. Section II describes the overall control system structure. Section III presents the on-line T-S fuzzy model modelling algorithm including on-line structure identification and parameter adjustment. Section IV describes the controller design procedure. This followed by section V that presents the simulation results of two examples: on-line identification of a second-order nonlinear uncertain system and model reference tracking control applied to a single robot arm by

This work has been partly supported by Universities UK on behalf of the Department for Education and Skills (DfES) through the UK funding councils and by the Polish State Committee for Scientific Research under grant No. 4 T11A 008 25.

The authors are with the School of Electronic, Electrical and Computer Engineering, the University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK. (email: rxq338@bham.ac.uk, m.brdys@bham.ac.uk).

0-7803-9098-9/05/$25.00 ©2005 AACC

using the proposed scheme. Finally, section VI concludes the paper.

II. CONTROL SYSTEM STRUCTURE

In this paper, we seek to design an indirect model reference controller for a class of SISO nonlinear uncertain systems that can be represented by the following discrete difference equation model:

$$y_p\ (k+1) = f[y_p(k), y_p(k-1), \cdots, y_p(k-n+1); \\ u(k), u(k-1), \cdots, u(k-m+1)] \quad (m \leq n) \qquad (1)$$

where $u(k) \in R$ and $y_p(k) \in R$ represent the I/O pair of the SISO plant at time k and $f(\cdot)$ is an unknown function. The control objective is to make the system output track a specified trajectory $y_{ref} \in R$.

A diagram of the overall control system structure is shown in Fig. 1. It is similar to that presented in [9] and [10]. However, instead of the dynamic neural network, a T-S fuzzy model is used here as a design model of the system (1), based on which the controller is constructed. The scheme consists of two parts: a dynamic T-S fuzzy model and an adaptive controller based on the model. Both the structure and parameters of the T-S fuzzy model can be updated on-line using the modelling algorithm described in Section III so that it can provide a dynamic model of the plant. The controller generates control signals adaptively according to an adaptive law given in Section IV based on the current local model of the plant.

III. ON-LINE MODELLING OF NONLINEAR UNCERTAIN SYSTEMS WITH T-S FUZZY MODELS

A. Characterization of Fuzzy Model

The T-S fuzzy model can be expressed as
R_i:
IF $y(k)$ is M_{i1} AND $\cdots$ AND $y(k-n+1)$ is M_{in} AND $u(k)$ is N_{i1} AND $\cdots$ AND $u(k-m+1)$ is N_{im}
THEN $y_i(k+1) = a_{i0} + a_{i1}y(k) + \cdots + a_{in}y(k-n+1) + b_{i1}u(k) + \cdots + b_{im}u(k-m+1) \quad i = 1, \cdots, N$

(2)

Reference fuzzy sets M_{i1},$\cdots$,M_{in} and N_{i1},$\cdots$,N_{im} represent linguistic labels that are defined over the input and output spaces. Define $\mathbf{x_k}$ as the input vector $\mathbf{x_k} = [y(k), \cdots, y(k-n+1), u(k), \cdots, u(k-m+1)] = [x_1(k), x_2(k), \cdots, x_{m+n}(k)]^T$ and $\mathbf{x_i^*} = [x_{i1}^*, x_{i2}^*, \cdots, x_{im+n}^*]^T$ as the focal point of the ith rule antecedent. The membership function for the input variables is chosen as a Gaussian function.

$$\mu_{ij} = exp(-\alpha\|x_j - x_{ij}^*\|^2) \qquad (3)$$

where $i = 1, \cdots, N, j = 1, \cdots, m+n$ and $\alpha = 4/r^2$. The constant r is effectively the radius defining a neighborhood.

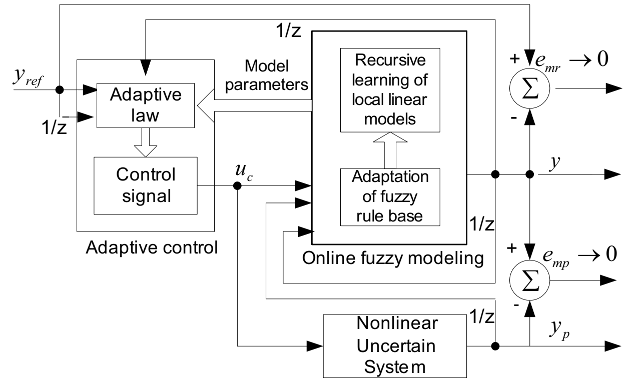

Fig. 1. Adaptive control based on T-S fuzzy model

The output $y(k+1)$ of the overall model is defined from the individual linear subsystems associated with the rules, according to the T-S reasoning method as follows

$$y(k+1) = \frac{\sum_{i=1}^{N} \omega_i y_i(k+1)}{\sum_{i=1}^{N} \omega_i} = \sum_{i=1}^{N} \bar{\omega}_i y_i(k+1) \quad (4)$$

where $\omega_i = \prod_{j=1}^{m+n} \mu_{ij}(x_j)$ is the firing strength of the ith rule and $\bar{\omega}_i = \omega_i / \sum_{i=1}^{N} \omega_i$ is the normalized firing strength.

B. On-line Structure Identification

Usually identification of a T-S fuzzy model consists of two parts: structure identification and parameter adjustment. Structure identification includes determining the following: 1)input variables to the rules; 2)shapes and initial parameters of membership functions; 3)number of fuzzy rules; 4)focal points for each rule. In this paper, we assume we have known the input variables and preselect the shapes of the membership functions to be Gaussian type as described in (3). Then the left task for on-line structure identification is to decide the number of fuzzy rules and the focal points for each rule, which can be solved by our on-line fuzzy clustering procedure.

Our on-line clustering method is motivated from the procedure called subtractive clustering. It starts with the first data point rather than supposing that all the data is available at the beginning. When the first data comes, it is chosen as the first cluster center (focal point) and its coordinates are used to form the antecedent part of the first fuzzy rule. Its potential is assumed to be 1. Since the new data affects the potentials of all the previously found cluster centers, the set of cluster centers are updated as the following steps.

At time step k, the new data v_k comes. v_k is a vector that consists of all the input and output variables up to k.

$$v_k = [y(k), \cdots, y(k-n), u(k-1), \cdots, u(k-m)]^T \quad (5)$$

The potential of v_k is computed as

$$P_k(v_k) = \sum_{j=1}^{k} exp(-\alpha\|v_k - v_j\|^2) \qquad (6)$$

When the new data comes, the potentials of current cluster centers $\{v_1^*, v_2^*, \cdots, v_p^*\}$ are updated as

$$P_k(v_i^*) = P_{k-1}(v_i^*) + exp(-\alpha\|v_i^* - v_k\|^2) \quad (7)$$

where $i = 1, 2, \cdots, p$. The potential of the new data is compared with the updated potentials of all the existing cluster centers to decide the new set of cluster centers using the following criteria:

To add or replace a rule:

if $P_k(v_k) > max\{P_k(v_i^*), i = 1, \cdots, p\}$
 let $\delta_{min} = min\{exp(-\alpha\|v_k - v_i^*\|^2), i = 1, \cdots, p\}$.
 let v_l^* is the center which is closest to v_k.
 if $\frac{\delta_{min}}{r} + \frac{P_k(v_l^*)}{P_k(v_k)} < 1$
  v_k replaces v_l^* as a new cluster centers.
 else
  Accept v_k as a new cluster center.
 end
else
  Reject v_k as a new cluster center.
end

To delete a rule:

After updating their potentials and a new data vector is added, rejected or replaces an old cluster center as a new center, we get a set of q cluster centers.

let $d_{min} = min\{exp(-\alpha\|v_i^* - v_j^*\|^2), i = 1, \cdots, q-1, j = 2, \cdots, q\}$ and $P_{max} = max\{P_k(v_i^*), i = 1, \cdots, q\}$

let v_i^* and v_j^* be the two centers which have the closest distance and assume $P_k(v_i^*) < P_k(v_j^*)$

if $\frac{d_{min}}{r} + \frac{P_k(v_i^*)}{P_{max}} < 1$
  Delete v_i^* from current set of cluster centers.
end

In the above criteria, r is the radius of influence, which effects the number and spread of cluster centers.

The above on-line clustering approach provides a dynamic and evolving rule base by upgrading it when incoming new data brings new information. The number of the cluster centers determines the number of the fuzzy rules and the values of them are used to construct the membership functions of the input variables.

C. On-line Parameter Adjustment

We seek to identify a T-S fuzzy model in two phases. In the first phase, we upgrade the model structure by fuzzy clustering; in the second phase, we adjust the parameters of the consequent part of fuzzy rules. In principle, structure and parameter identification of the fuzzy model cannot be completely separated because, to identify the structure, we need a initial set of parameters for model verification. The initial parameters of the rule consequents can be chosen by off-line training or arbitrarily set. With the initial consequent parameters, the on-line fuzzy clustering procedure can work to upgrade the rule base and the parameters of the membership functions of the antecedent variables. Then for a fixed upgraded rule base and antecedent parameters, the rule consequents form a set of linear equations leading to a linear regression problem, which can be solved by recursive least squares estimation(RLSE). We can recast (4) as a matrix equation

$$y_{k+1} = \Lambda_k^T \Theta \quad (8)$$

where $\Lambda_k = [\bar{\omega}_1(\mathbf{x_k})\mathbf{x_{ek}^T}, \cdots, \bar{\omega}_N(\mathbf{x_k})\mathbf{x_{ek}^T}]^T$, $\Theta = [\theta_1^T, \theta_2^T, \cdots, \theta_N^T]^T$, $\mathbf{x_{ek}} = [1, \mathbf{x_k}^T]^T$, and $\theta_i = [a_{i0}, a_{i1}, \cdots, a_{in}, b_{i1}, \cdots, b_{im}]^T$.

In (8), only Θ is an unknown vector whose elements are the consequent parameters of the rules. Our goal is to find a least square estimate (LSE) of Θ, namely, Θ^*, to minimize the cost function $\Psi = \sum_{k=1}^{L}(y_k - \Lambda_{k-1}^T\Theta)^2$, where L is the number of data. Θ can be calculated iteratively using the recursive least square estimate (RLSE) formulas that are widely adopted in the literature

$$\begin{aligned} \Theta_k &= \Theta_{k-1} + S_k\Lambda_{k-1}(y_k - \Lambda_{k-1}^T\Theta_{k-1}) \\ S_k &= S_{k-1} - \frac{S_{k-1}\Lambda_{k-1}\Lambda_{k-1}^T S_{k-1}}{1 + \Lambda_{k-1}^T S_{k-1}\Lambda_{k-1}} \end{aligned} \quad (9)$$

where S_k is often called the covariance matrix and the least square estimate Θ^* is equal to Θ_L. The initial conditions to bootstrap (9) are $\Theta_0 = 0$ and $S_0 = \gamma I$, where γ is a positive large number and I is the identity matrix of dimension $M \times M$, where $M = (m+n+1) \times N$ is the number of consequent parameters.

It is important to note that when RLSE works to adjust the consequent parameters, the rule base and antecedent parameters are still suboptimal and may changes while new data comes. Therefore, we need to check if the rule base has changed before we apply RLSE to tune the consequent parameters.

When a new rule is added to current rule base, Θ_{k-1} is modified as

$$\tilde{\Theta}_{k-1} = [\Theta_{k-1}^T, \theta_{N+1}^T]^T \quad (10)$$

where $\theta_{N+1} = [0, \cdots, 0]^T$ is a $1 \times (m+n+1)$ vector. The covariance matrix is reset as

$$\tilde{S}_{k-1} = diagonal\{\eta S_{k-1}, \gamma, \cdots, \gamma\} \quad (11)$$

where η is a rationale larger than 1 needed for the correction of the covariance matrix with the new rule. And Λ_{k-1} is reset as

$$\tilde{\Lambda}_{k-1} = [\Lambda_{k-1}^T, \bar{\omega}_{N+1}(\mathbf{x_{k-1}})\mathbf{x_{ek-1}^T}]^T \quad (12)$$

When a rule or several rules are deleted from the rule base, the covariance matrix S_{k-1}, Θ_{k-1} and Λ_{k-1} are updated by deleting the corresponding elements in the matrix or vectors.

When an old rule is replaced by a new rule, the above matrices are inherited from the previous time step.

A block diagram that represents the overall identification algorithm is demonstrated in Fig. 2

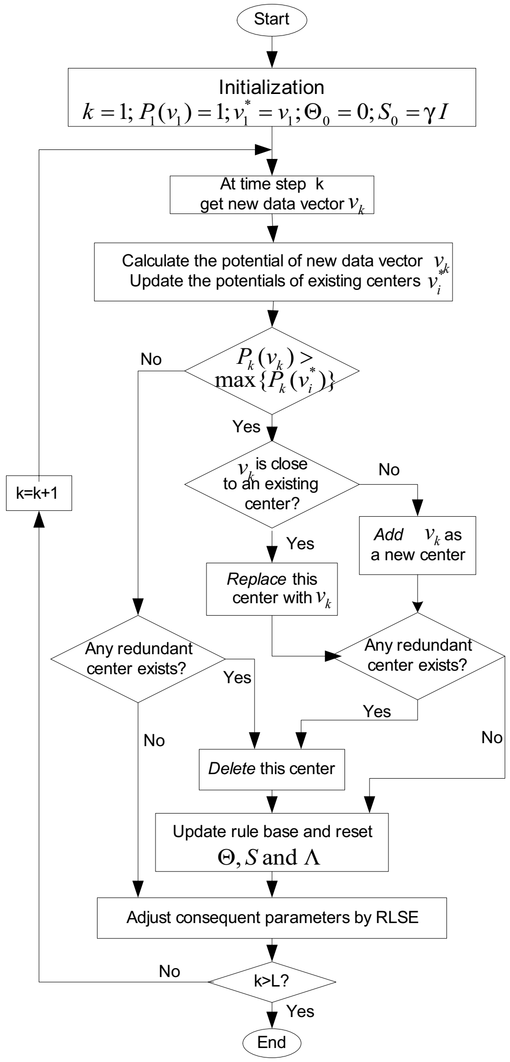

Fig. 2. Block diagram for on-line modelling algorithm

IV. Control Algorithm

We define $\mathbf{z}(k) = [y(k)\, y(k-1)\, \cdots\, y(k-n+1)]^T$ and for simplicity, we assume $b_j = 0, \forall_j > 1$, then (2) can be changed into

$$\mathbf{z}(k+1) = \hat{A}\mathbf{z}(k) + \hat{B}u(k) + \hat{D} \tag{13}$$

where

$$\hat{A} = \sum_{i=1}^{N} \bar{\omega}_i \cdot \begin{bmatrix} a_{i1} & a_{i2} & \cdots & a_{in} \\ 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix}, \hat{B} = \sum_{i=1}^{N} \bar{\omega}_i \cdot \begin{bmatrix} b_{i1} \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$\hat{D} = \sum_{i=1}^{N} \bar{\omega}_i \cdot \left[a_{i0}\, 0 \cdots 0\right]^T. \tag{14}$$

Note that the actual output of T-S model $y(k+1)$ is

$$y(k+1) = C\mathbf{z}(k+1) = C\hat{A}\mathbf{z}(k) + C\hat{B}u(k) + C\hat{D} \tag{15}$$

where $C = [1\ 0 \cdots\ 0]$.

An assumption is made: $C\hat{B}$ is *invertible*. It means in practice that the relative degree of the nonlinear plant, hence its nonlinear T-S model, is equal to one. It is straightforward to extend the consideration to cover systems when the relative degree is larger than one. The feedback linearizing control can be calculated as

$$u_c(k+1) = (C\hat{B})^{-1}(-C\hat{A}\mathbf{z}(k) - C\hat{D} + \xi(k)) \tag{16}$$

One-step-delay for computing $u(k)$ is used here. The delay is necessary to avoid the so-called *logical inconsistency* in operation of the resulting global controller, which occurs if the control signal appears in the premises of the controller rules.

Eq. (16) applied to (15) results in it being decoupled and linear with respect to the new input $\xi(k)$. The auxiliary control input $\xi(k)$ is designed as a simple linear pole placement

$$\xi(k) = y_{ref}(k+1) + \beta(y(k) - y_{ref}(k)) \tag{17}$$

where $0 < \beta < 1$. The control input, as defined by (16) and (17), is applied to both the plant and the T-S model. The error between the output of the model and the reference signal, $e_{mr}(k+1) = y(k+1) - y_{ref}(k+1)$, obeys $e_{mr}(k+1) = \beta e_{mr}(k)$ and converges exponentially to the origin with the rate α.

Thus, the control algorithm operates perfectly well when applied to the model. We can assume that after some initial transient, $e_{mr}(k) \to 0$. Hence, if the model identified in Fig. 1 achieves $e_{mp}(k) \to 0$ when $k \to \infty$ by updating the model in the close-loop, the plant tracking error $e_{rp}(k)$ is equivalent to the modelling error $e_{mp}(k)$

$$e_{rp}(k) = y_{ref}(k) - y_p(k) = y(k) - y_p(k) = e_{mp}(k) \tag{18}$$

so that $e_{rp}(k) \to 0$ when $k \to \infty$.

The overall stability analysis is still under research.

V. Simulation Results

Example 1: In this example, we test the on-line fuzzy modelling algorithm in identifying a nonlinear uncertain plant that is given by the second-order nonlinear difference equation

$$\begin{aligned} y(k+1) &= f[y(k), y(k-1), u(k)] \\ &= \frac{y(k)y(k-1)[y(k)+2.5]}{1+y^2(k)+y^2(k-1)} + u(k) \end{aligned} \tag{19}$$

where $f(\cdot)$ is assumed to be unknown. Our objective is to find a T-S fuzzy model of $f(\cdot)$ by using the modelling algorithm proposed in Section III. The input signal is chosen as $u(k) = 0.5sin(2\pi k/5) + 0.5sin(2\pi k)$.

Using the proposed on-line fuzzy modelling algorithm, the T-S model structure and parameters are generated simultaneously and automatically. By choosing difference r, a different number of fuzzy rules is generated, which is illustrated in Table I. Too small a value of r leads to too many rules which brings too many parameters that will slow

TABLE I

SIMULATION RESULTS WITH DIFFERENT R

r	num. of rules	MSE
0.1	8	0.0072
0.4	4	0.0090
0.8	3	0.1788

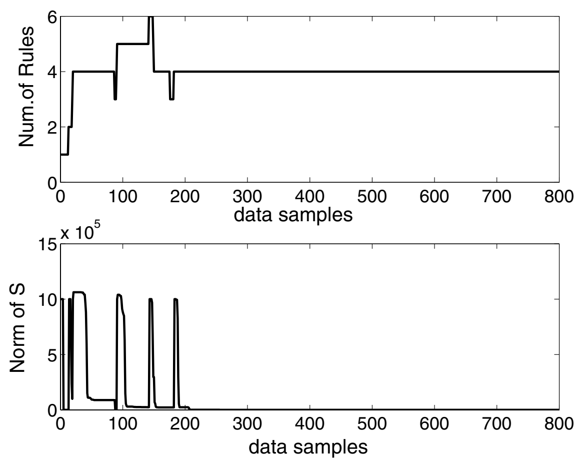

Fig. 3. Evolution of fuzzy rules and norm of covariance matrix

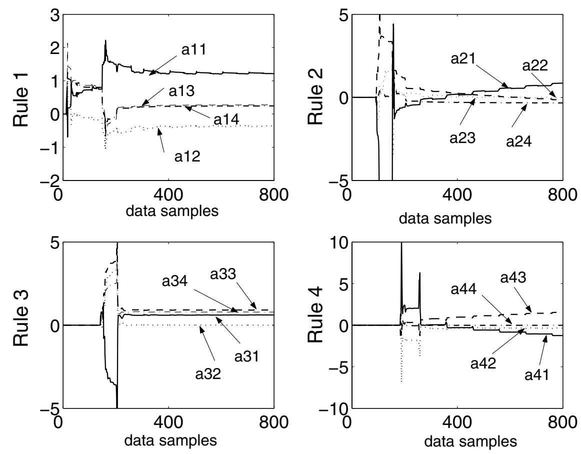

Fig. 4. Evolution of consequence parameters

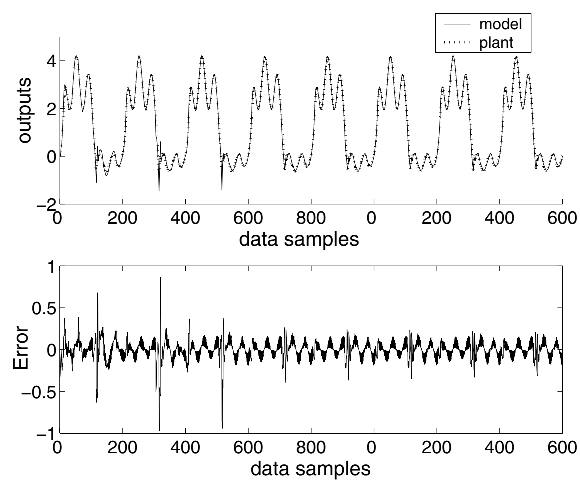

Fig. 5. Error between the outputs of model and plant

down the adaptation speed; too large a value of r leads to insufficient rules which will increase modelling error. In this example, choosing $r = 0.4$, we obtain four rules that have comparatively few parameters to tune and small MSE. The evolutions of fuzzy rules and consequent parameters are shown in Fig. 3 and Fig. 4. Fig. 5 illustrates the comparison of model and plant outputs and the error between them.

Example 2: We verify the validity of the proposed T-S fuzzy modelling and control scheme for tracking control of the single robot arm, which is described by

$$\begin{aligned} \dot{\zeta}_1 &= \zeta_2 \\ \dot{\zeta}_2 &= -a_1 sin(\zeta_1) - a_2\zeta_1 + bu \end{aligned} \tag{20}$$

where $y = \zeta_1$ is the arm position which is the measured output,ζ_2 is the (unmeasured) angular velocity and u represents the input force. The system parameters are given as follows: $a_1 = a_2 = b = 1$ and the identification and control algorithm operates in discrete time with the sampling time $T = 0.05s$. The desired position trajectory is assumed to be $y_{ref}(t) = 4sin(4t) + 0.5sin(2t)$. The on-line modelling starts from one fuzzy rule and evolves to two rules after 900 data samples. The evolutions of fuzzy rules and consequent parameters are shown in Fig. 6 and Fig. 7. The controller calculates control signal adaptively based on the current model structure and parameters and the generated control signal is used as the control input for both the model and the plant. The output of the plant is feedback to the model for on-line learning. Fig. 8 and Fig. 9 illustrate the tracking response and control input.

It should be pointed out that the model structure identification method described in Section III is general and can be applied to both the off-line modelling and on-line modelling. The latter is done entirely for the control purposes. Hence, the global model is not needed. Only a local model will enable us to produce suitable control action at the time it required. Hence, good tracking accuracy is achieved with two rules in the robot arm example for the specified selection of reference trajectory. This will change when the reference trajectory changes and the model structure identification algorithm will find required model structure in the close-loop.

VI. CONCLUSIONS

In this paper, an on-line adaptive modelling and control scheme based on T-S fuzzy model has been proposed. Both the structure and parameters of the T-S model can be updated on-line, which makes it capable of approximating complex nonlinear dynamic systems. This scheme

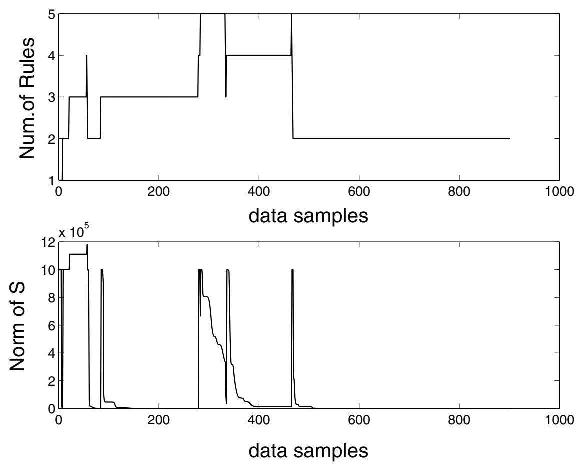

Fig. 6. Evolution of fuzzy rules and norm of covariance matrix

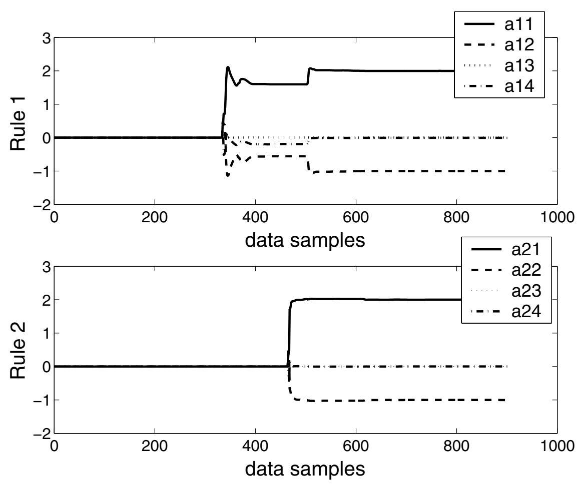

Fig. 7. Evolution of consequent parameters

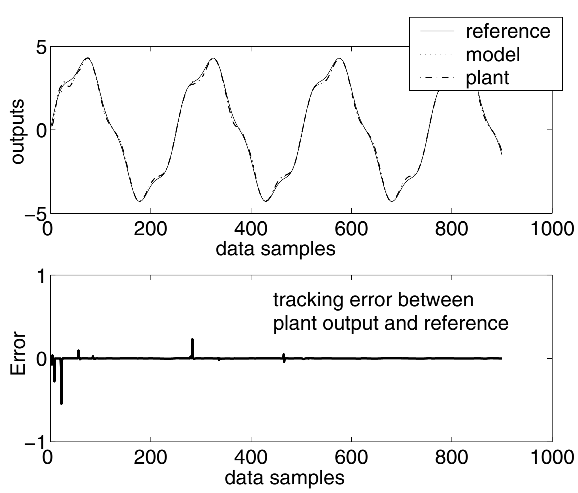

Fig. 8. Tracking response

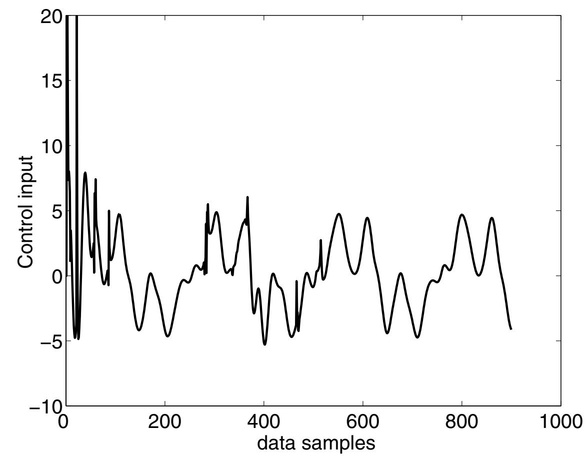

Fig. 9. Control input

is computationally efficient and suitable for real-time implementation as the rule base evolution is recursive based on unsupervised learning and the parameters are adjusted by RLSE. The controller is an indirect model reference controller that generates control signals based the upgraded model of the plant using feedback linearization. Simulation studies of a second-order nonlinear plant and a single robot arm verify the identification, adaptation and tracking performance of the proposed scheme.

REFERENCES

[1] B. Kosko, "Fuzzy systems as universal approximator", *IEEE Trans. computers.*, vol. 43, pp 1329-1333, Nov. 1994.

[2] S. Barada and H. Singh, "Generating optimal adaptive fuzzy-neural models of dynamical systems with applications to control", *IEEE Trans. Syst., Man, Cybern.*, vol. 28, no. 3, pp 371-391, Aug. 1998.

[3] R. Rovatti, "Fuzzy piecewise multilinear and piecewise linear systems as universal approximators in Sobolev norms", *IEEE Trans. Fuzzy Syst.*, vol. 6, pp 235-249, May 1998.

[4] P. P. Angelov and D. P. Filev, "An approach to online identification of Takagi-Sugeno fuzzy models", *IEEE Trans. Syst., Man, Cybern.*, vol. 34, no. 1, pp 484-498, Feb. 2004.

[5] Y. Gao and M. J. Er, "Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems", *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 4, pp 462-477, Aug. 2003.

[6] Y. Yang and J. Ren, "Adaptive fuzzy robust tracking controller design via small gain approach and its application", *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 6, pp 783-795, Dec. 2003.

[7] L. X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Englewood Cliffs, NJ:Prentice-Hall, 1994.

[8] J. Abonyi, R. Babuska, F. Szeifert, "Modified Gath-Geva fuzzy clustering for identification of T-S fuzzy models", *IEEE Trans. Syst., Man, Cybern.*, vol. 32, no. 5, pp 612-621, Oct. 2002.

[9] M. A. Brdys and G. J. Kulawski, "Dynamic neural controllers for induction motor", *IEEE Trans. Neural Networks*, vol. 10, no. 2, pp 340 - 355, Mar. 1999.

[10] G. J. Kulawski and M. A. Brdys, "Stable adaptive control with recurrent networks", *Automatica*, vol. 36, no. 1, pp 5 - 22, Jan. 2000.

[11] C. K. Chiang, H.-Y. Chung, and J. J. Lin, "A self-learning fuzzy logic controller using genetic algorithms with reinforcements", *IEEE Trans. Fuzzy Syst.*, vol. 5, pp 460-467, Aug. 1997.

[12] P. P. Angelov, V. I. Hanby, R. A. Buswell, and J. A. Wright, "Automatic generation of fuzzy rule-based models from data by genetic algorithm", in *Advances in Soft Computing*, R. John and R. Birkenhead, Eds. Heidelberg, Germany: Springer-Verlag, 2001, pp 31-40.