行政院國家科學委員會專題研究計畫　期中進度報告

最佳線上即時智慧型控制系統的設計與作(1/3)

計畫類別：個別型計畫
計畫編號：NSC92-2213-E-009-056-
執行期間：92 年 08 月 01 日至 93 年 07 月 31 日
執行單位：國立交通大學電機與控制工程學系

計畫主持人：王啟旭
共同主持人：李祖添

報告類型：精簡報告
報告附件：出席國際會議研究心得報告及發表論文
處理方式：本計畫可公開查詢

中　華　民　國 93 年 5 月 24 日

ABSTRACT

This project is to explore the theoretical and practical issues of real-time applications of FNN (Fuzzy Neural Network) for linear and nonlinear dynamical systems. The first year goal has been achieved by developing a new on-line TS-type FNN training architecture with dynamical optimal training under real-time environment to represent the original dynamical system. Therefore the TS-type FNN can be optimally trained with maximum error reduction in minimum time period. The inverted pendulum system is illustrated by our new TS-type FNN training architecture and a proportional controller is also designed based upon pole-placement technique to achieve the balancing of the rod. By the end of the first year, we assume the computational time of the controller and optimal training mechanism is small enough and can be neglected. This is not realistic at all. However, this has also paved a healthy way for us to explore the second year goal of finding the maximum computational time, or the maximum delay time, for the controller and optimal training mechanism, so that the closed-loop system can be asymptotical stable.

I. The TS-Type FNN Model for Uncertain Nonlinear Systems

We consider an uncertain n^{th} order nonlinear system with m inputs described by the nonlinear equation as

$$\dot{\underline{x}}(t) = f(\underline{x}(t), \underline{u}(t)) \qquad (1)$$

where

$$\underline{x}(t) = [x_1(t) \quad x_2(t) \quad ... \quad x_n(t)]^T,$$

$$\underline{u}(t) = [u_1(t) \quad u_2(t) \quad ... \quad u_m(t)]^T,$$

$$\dot{\underline{x}}(t) = [\dot{x}_1(t) \quad \dot{x}_2(t) \quad ... \quad \dot{x}_n(t)]^T.$$

The states $\underline{x}(t)$ and the inputs $\underline{u}(t)$ of the nonlinear system are assumed to be measurable, and $\dot{\underline{x}}(t)$ can be obtained from $\underline{x}(t)$ with some past information. The i[th] rule of TS-type fuzzy model to represent the nonlinear system can be described by:

Rule *i***:** If $z_1(t)$ is F_{i1} and **. . .** and $z_g(t)$ is F_{ig} then $\dot{\underline{x}}_i(t) = A_i \underline{x}(t) + B_i \underline{u}(t)$,

for i=1, 2, …, r.

where F_{ij} is the fuzzy set, $A_i \in \Re^{n \times n}$, $B_i \in \Re^{n \times m}$; r is the number of *If-then* rules.; and $z_1(t)$, $z_2(t)$, **. . .** , $z_g(t)$ are the premise variables. The A_i and B_i matrices are normally obtained from the Jacobian matrix to locally linearize the well-specified nonlinear systems [5]-[7]. For uncertain nonlinear systems, the A_i and B_i matrices can only be obtained from on-line training process which will be discussed in the following sections. To start with, we propose the following configuration as the TS-type fuzzy model to represent the uncertain nonlinear system:

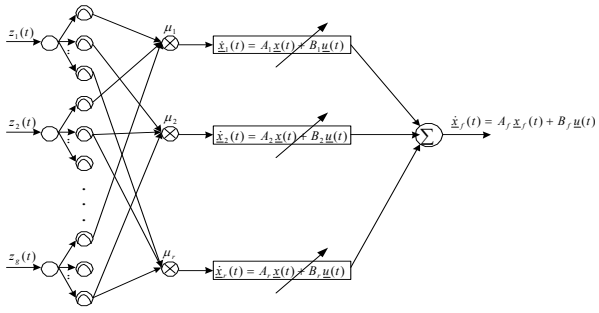

Fig. 1 TS type fuzzy model for uncertain nonlinear systems

The above fuzzy system should be inferred as:

$$\dot{\underline{x}}_f(t) = \sum_{i=1}^{r} \mu_i(\underline{z}(t))[A_i \underline{x}(t) + B_i \underline{u}(t)] = A_f \underline{x}_f(t) + B_f \underline{u}(t) \qquad (2)$$

where

$$\underline{z}(t) = [z_1(t) \quad z_2(t) \quad .... \quad z_g(t)]^T$$

$$\mu_i(\underline{z}(t)) = \frac{\prod_{j=1}^{g} F_{ij}}{\sum_{i=1}^{r}(\prod_{j=1}^{g} F_{ij})} \rightarrow \sum_{i=1}^{r} \mu_i(\underline{z}(t)) = 1 \qquad (3)$$

$$A_i = \begin{bmatrix} A_i^{(1,1)} & \cdots & A_i^{(1,n)} \\ \vdots & & \vdots \\ A_i^{(n,1)} & \cdots & A_i^{(n,n)} \end{bmatrix}, B_i = \begin{bmatrix} B_i^{(1,1)} & \cdots & B_i^{(1,m)} \\ \vdots & & \vdots \\ B_i^{(n,1)} & \cdots & B_i^{(n,m)} \end{bmatrix} \qquad (4)$$

and r is the number of rules, or the number of nominal operating points for the uncertain nonlinear systems. Equation 2 shows that the uncertain nonlinear system can be represented by a linear dynamical equation at any time instant, which can be inferred from the TS-type fuzzy model. Also, the A_f and B_f matrices are different at different time instants. Further we need to derive the on-line BP training rules to update the A_i and B_i matrices. We define the cost function H as follows:

$$H = \frac{1}{2} \| \dot{\underline{x}}_f(t) - \dot{\underline{x}}(t) \|^2 \qquad (5)$$

To minimize H, we have the following BP equations to update A_i and B_i matrices:

$$A_i(k+1) = A_i(k) - L\frac{\partial H}{\partial A_i}, \quad B_i(k+1) = B_i(k) - L\frac{\partial H}{\partial B_i}$$

Where

$$\frac{\partial H}{\partial A_i} = \begin{bmatrix} \frac{\partial H}{\partial A_i^{(1,1)}} & \cdots & \frac{\partial H}{\partial A_i^{(1,n)}} \\ \vdots & & \vdots \\ \frac{\partial H}{\partial A_i^{(n,1)}} & \cdots & \frac{\partial H}{\partial A_i^{(n,n)}} \end{bmatrix},$$

$$\frac{\partial H}{\partial B_i} = \begin{bmatrix} \frac{\partial H}{\partial B_i^{(1,1)}} & \cdots & \frac{\partial H}{\partial B_i^{(1,m)}} \\ \vdots & & \vdots \\ \frac{\partial H}{\partial B_i^{(n,1)}} & \cdots & \frac{\partial H}{\partial B_i^{(n,m)}} \end{bmatrix}$$

and

$$\frac{\partial H}{\partial A_i^{(p,q)}} = \mu_i(t)[\dot{x}_{f_p}(t) - \dot{x}_p(t)]x_q(t) \text{ ; } (p, q = 1, 2, \ldots, n) \qquad (6)$$

$$\frac{\partial H}{\partial B_i^{(r,s)}} = \mu_i(t)[\dot{x}_{f_r}(t) - \dot{x}_r(t)]u_s(t) \text{ ; } (r = 1, 2, \ldots, n; s = 1, 2, \ldots, m) \qquad (7)$$

II On-Line Optimal Training with Least-Squared Initialization

Dynamical optimal off-line training for a two-layer NN was proposed in [13]. It can be guaranteed in [13] that the back-propagation training process can be optimally converged in the sense of maximum error reduction after each iteration. For on-line training purpose, it may still not be suitable due to its unpredictable initial values of the weighting factors. But the dynamical optimal training in [13] can be very powerful for on-line disturbance rejection due to its guaranteed maximum error reduction. In this section, the optimal training process in [13] will be

utilized in the on-line optimal training of TS-type fuzzy model for uncertain nonlinear systems (Fig. 1). Also the least-squared initialization for Fig. 1 will be proposed to provide proper initial values for the weighting factors so as to speed-up the on-line optimal training process.

Let us define the input training matrix R, the output matrix of the TS-type fuzzy model (for uncertain nonlinear system) Y and the actual output matrix of the uncertain nonlinear system D as:

$$R = [x_1(t) \quad x_2(t) \quad ... \quad x_n(t) \quad u_1(t) \quad ... \quad u_m(t)]^T \in \Re^{(m+n)\times 1} \quad (8)$$

$$Y = \underline{\dot{x}}_f(t) = [\dot{x}_{f_1}(t) \quad \dot{x}_{f_2}(t) \quad ... \quad \dot{x}_{f_n}(t)]^T \in \Re^{n\times 1} \quad (9)$$

$$D = \underline{\dot{x}}(t) = [\dot{x}_1(t) \quad \dot{x}_2(t) \quad ... \quad \dot{x}_n(t)] \in \Re^{1xn} \quad (10)$$

Also we can define an augmented weighting matrix W to include the uncertain A_i and B_i matrices defined in (4):

$$W = \begin{bmatrix} \sum_{i=1}^{r} \mu_i A_i^T \\ \sum_{i=1}^{r} \mu_i B_i^T \end{bmatrix} = \sum_{i=1}^{r} \mu_i W_i, \text{ and } W_i = \begin{bmatrix} A_i^T \\ B_i^T \end{bmatrix} \quad (11)$$

Therefore the output of fuzzy model Y in (9) can be shown as:

$$Y = R^T W \quad (12)$$

Theorem 1

The BP training rule to update the weighting matrix W_i in (11) is

$$W_i(k+1) = W_i(k) - \beta_k \frac{1}{mn} RE_k \quad (17)$$

where β_k is the learning rate for k^{th} iteration, R is defined in (8) and E_k is the error matrix after k^{th} iteration, defined in (14).

Theorem 1 is to simultaneously update the A_i and B_i matrices for each subsystem in the TS-type fuzzy model in Fig. 1. It is our goal to guarantee the convergence of the on-line training process in Theorem 1. Therefore the dynamical optimal training in [13] can be applied in this case. The dynamical optimal learning rate can guarantee the fastest convergent speed for the on-line training process. Even if the initial values of the weighting matrix W_i in (11) is far from satisfactory, the dynamical optimal training can still guarantee the optimal convergence, just takes longer time to converge. But this is not good enough for on-line training. In other words, choosing random initial value for weighting matrix W_i is unacceptable for on-line training purpose. Further the least-squared estimation techniques for linear systems can be found in [1]. Since the TS-type fuzzy model for uncertain nonlinear system contains many linear subsystems, therefore it is reasonable to use the least-squared estimation techniques [1] to estimate the initial values of W_i. If the order of the linear subsystems of TS-type fuzzy model is n and the number of input ($u(t)$)is m, then we must measure $n+m$ outputs to get the estimated initial value of W_i. Assume the initial value of W_i is W_i^0:

$$W_i^0 = M = \begin{bmatrix} A_i^T \\ B_i^T \end{bmatrix}^0 \quad (18)$$

Further, input $\underline{u}(t)$ and measured state vectors $\underline{x}(t)$ can be combined as θ:

$$\theta = \begin{bmatrix} \underline{x}_1^T & \underline{u}_1^T \\ \underline{x}_2^T & \underline{u}_2^T \\ & : \\ \underline{x}_{n+m}^T & \underline{u}_{m+n}^T \end{bmatrix} \quad (19)$$

The set of measured outputs (i.e., $\underline{\dot{x}}$) is also defined as:

$$Y = \begin{bmatrix} \underline{\dot{x}}_1^T \\ \underline{\dot{x}}_2^T \\ \vdots \\ \underline{\dot{x}}_{n+m}^T \end{bmatrix} \qquad (20)$$

Therefore we have the following matrix equation:

$$Y = \theta M + \varepsilon \qquad (21)$$

Equation (21) includes the noise matrix term $\boldsymbol{\varepsilon}$ from the measurement. Then the estimated $\hat{M}$ (via least squared method) can be shown as

$$\hat{M} = (\theta^T \theta)^{-1} \theta^T Y = W_i^0 \qquad (22)$$

From (22), we can get the least-squared initial parameters of each linear subsystem in the TS-type fuzzy model in Fig. 1. The following ***Algorithm I*** summarizes the procedures to perform the on-line optimal training (with least-squared initialization) of TS-type fuzzy model for uncertain nonlinear systems.

Algorithm I: Dynamical Optimal Training of TS-type Fuzzy Model with Least-Squared Initialization

[Step 1]: Use any input $u(t)$ to excite the uncertain nonlinear system and measure sufficient data information of $\underline{x}(t)$ and $\underline{\dot{x}}(t)$. Then define the r nominal operating points and the corresponding membership functions for $\underline{x}(t)$ and $\underline{\dot{x}}(t)$. Apply (22) to find the initial weighting matrix W_i^0 of each subsystem for i=1, …, r.

[Step 2]: Release the system to run freely to track a reference input signal.

[Step 3]: If the norm of tracking error is less than a pre-defined threshold e_1, GOTO Step 2. Otherwise GOTO Step 4.

[Step 4]: Measure on-line $\underline{x}(t)$ and $\underline{\dot{x}}(t)$. For i=1,…, r, apply Theorem 1 in [13] to find the optimal learning rates to train the weighting matrix of each subsystem via (17). The optimal training must continue until relative errors of W_i is less than another pre-defined threshold e_2.

[Step 5]: GOTO Step 2.

III. Stable Tracking Controller for TS-Type Fuzzy Model of Uncertain Nonlinear Systems

In this section, a stable tracking controller for TS-type fuzzy model of uncertain nonlinear system will be proposed. The full state feedback with pole placement techniques will be adopted to construct the tracking controller. The tracking controller will be updated via adaptive rules during on-line operations. The adaptive rules are designed to supervise the ability of the controller based on tracking error. If tracking error is over some threshold value, the TS-type fuzzy model must be optimally re-trained. This will imply the update of the tracking controller. Fig. 3 shows the overall

closed-loop configuration.

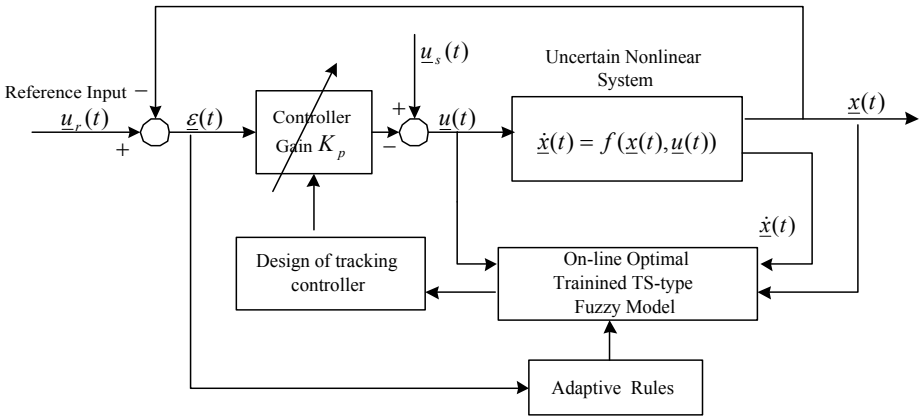

Fig. 3. Closed-loop configuration of tracking controller

To start with, we assume that during the p^{th} time interval, we can use (23) to represent the uncertain nonlinear system,

$$\dot{\underline{x}}(t) = A_f(p)\underline{x}(t) + B_f(p)\underline{u}(t) \qquad (23)$$

where $A_f(p)$ and $B_f(p)$ are updated from the on-line optimal training process at the beginning of p^{th} time interval. The time interval sequence is shown in Fig. 4.

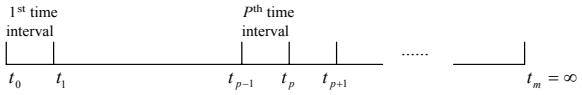

Fig. 4. Time sequence of control intervals

For simplicity, we adopt A and B to replace A(p) and B(p) in p^{th} time interval in the following derivations. In order to design a tracking controller for the uncertain nonlinear system, we can simplify Fig. 3 into Figure 5 as follows:

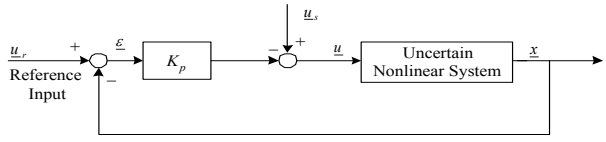

Fig. 5. Simplification of Fig. 3

In Fig. 5, it is desired to let the output $\underline{x}$ to track the reference input $\underline{u}_r(t)$ by the proportional controller K_{p}. An extra $\underline{u}_s(t)$ is also desired to satisfy the following linear dynamics of $\underline{u}_r(t)$:

$$\dot{\underline{u}}_r(t) = A_f\underline{u}_r(t) + B_f\underline{u}_s(t) \qquad (24)$$

In order to find $\underline{u}_{\text{s}}(t)$, the following lease-squared estimation can be applied:

$$\underline{u}_s(t) = \left(B_f^T B_f\right)^{-1} B_f^T[\dot{\underline{u}}_r(t) - A_f\underline{u}_r(t)]$$

However, as mentioned in section II, the above lease-squared estimation is not suitable for on-line purpose. Therefore we propose a new iterative approach to estimate $\underline{u}_{\text{s}}(t)$ during on-line operations. This can be shown in the following Theorem 2.

Theorem 2

The estimation of $\underline{u}_{\text{s}}(t)$ in (24) can be obtained from the on-line optimal training with least-squared initialization in *Algorithm I* for a Multi-Input-Single-Output two layer neural network shown in Fig. 6.

***Proof*:**

In order to find $\underline{u}_{\text{s}}(t)$, (24) can be re-organized as

$$B_f\underline{u}_s(t) = \dot{\underline{u}}_r(t) - A_f\underline{u}_r(t) \qquad (25)$$

Since the right-hand-side of (25) is known and can be denoted as

$$\underline{d} = \dot{\underline{u}}_r(t) - A_f \underline{u}_r(t) \qquad (26)$$

Therefore (25) becomes

$$B_f \underline{u}_s(t) = \underline{d} \qquad (27)$$

where B_f is an n×m matrix and $\underline{u}_s(t)$ is an m×1 vector and $\underline{d}$ is an n×1 vector. In comparison with (12), $\underline{u}_s(t)$ in (27) can be viewed as the unknown weighting matrix W to be decided, B_f^T can be viewed as the input training vector R, $\underline{d}$ is the desired output vector. Let

$$B_f = [\underline{b}_1 \quad \underline{b}_2 \quad \ldots \quad \underline{b}_n]^T = \begin{bmatrix} b_1(1) & b_1(2) & \ldots & b_1(m) \\ b_2(1) & b_2(2) & \ldots & b_2(m) \\ \vdots & \vdots & \vdots & \vdots \\ b_n(1) & b_n(2) & \ldots & b_n(m) \end{bmatrix}$$

$$\underline{u}_s(t) = [u_{s,1} \quad u_{s,2} \quad \ldots \quad u_{s,m}]^T;$$

$$\underline{d} = [d_1 \quad d_2 \quad \ldots \quad d_n]^T$$

$$\underline{y} = [y_1 \quad y_2 \quad \ldots \quad y_n]^T$$

where $\underline{y}$ is the actual output vector which should be very close to the desired output vector if $\underline{u}_s(t)$ can be properly found. Equation (27) can be further expanded as:

$$\underline{b}_i^T \times \underline{u}_s(t) = y_i \approx d_i \quad \text{for i=1, …, n.} \qquad (28)$$

Therefore we can draw a two-layer neural network from (27) as follows:

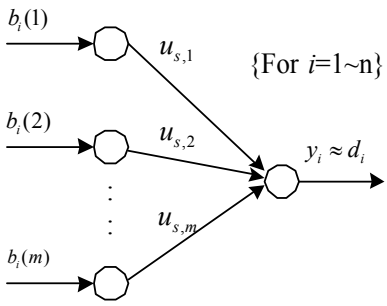

Fig. 6. Two layer NN to find $u_s(t)$

The above Fig. 6 is the MISO (Multi-Input-Single-Output) version of the two-layer NN in [13]. Therefore the on-line optimal training with least-squared initialization in *Algorithm I* in Section III can be applied to find the $\underline{u}_s(t)$ at any time instant t.

Q.E.D.

Subtracting (24) from (23) we have

$$\dot{\underline{\varepsilon}} = A_f \underline{\varepsilon} + B_f(\underline{u} - \underline{u}_s) \qquad (29)$$

where $\underline{\varepsilon} = \underline{x} - \underline{u}_r$. In order to guarantee the asymptotic stability of the error vector $\underline{\varepsilon}(t)$, we let

$$\underline{u} - \underline{u}_s = -K_p \underline{\varepsilon} \qquad (30)$$

Then (29) becomes

$$\dot{\underline{\varepsilon}} = (A_f - B_f K_p)\underline{\varepsilon} \qquad (31)$$

where K_p is the state feedback gain and can be obtained by assigning the closed-loop poles of $A_f - B_f K_p$ to lie within the left-half plane. This is a common pole-placement technique. Hence the error dynamics in (31) can be asymptotical stable. This will imply $\underline{x} \to \underline{u}_r$.

Algorithm II: On-Line Stable Tracking Controller

[Step 1]: From Figures 5 and 6, apply *Algorithm I* to find $u_s(t)$.

[Step 2]: Specify stable poles for (31). K_p can then be found through pole-placement technique.

For the overall closed-loop configuration in Fig. 5, we have all the necessary blocks and

signals except the Adaptive Rules. The Adaptive Rules can be shown in the following ***Algorithm III***.

Algorithm III**: **Adaptive Rules for updating the closed-loop system in Fig. 5

If $\| \underline{\varepsilon}(t) \| > Threshold$

Apply *Algorithm I* to update the TS-type Fuzzy Model

Apply *Algorithm I* to update $u_s(t)$

Apply *Algorithm II* to update K_p

Else

Stand Still

End.

The adaptive rules in ***Algorithm III*** can stabilize the closed-loop system (Fig. 5) due to the fact that our proposed modified on-line dynamical optimal training [13] can guarantee the fastest convergent speed of the training process.

IV. Example

In this section, we will apply our optimal on-line training to design a tracking controller to control the inverted pendulum to track a sinusoidal signal. For robustness test, the inverted pendulum system will be added with extra 50% of mass to simulate the case of sudden model change or noise at any time instant.

Example 1: Consider the inverted pendulum system [31] as shown in Fig.7. Let $x_1 = \theta$ be the angle of the pendulum with respect to the vertical line.

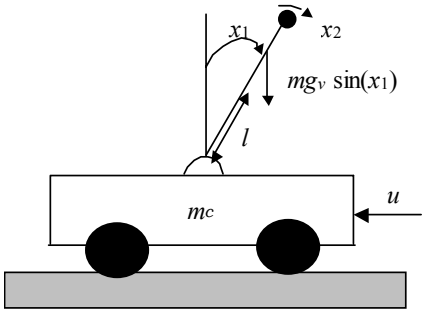

Fig. 7 The inverted pendulum system

The dynamic equations of the inverted pendulum system [31] are

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (f + gu), \qquad (39)$$

where

$$f = \frac{g_v \sin x_1 - \frac{mlx_2^2 \cos x_1 \sin x_1}{m_c + m}}{l(\frac{4}{3} - \frac{m \cos^2 x_1}{m_c + m})}; g = \frac{\frac{\cos x_1}{m_c + m}}{l(\frac{4}{3} - \frac{m \cos^2 x_1}{m_c + m})}$$

and $g_v = 9.8 meter/sec^2$ is the acceleration due to gravity, m_c is the mass of the cart, l is the half-length of the pole, m is the mass of the pole and u is the control input. In this example, we assume that $m_c = 2kg$, $m = 0.21kg$ and $l = 0.75$ *meter*.

We define three membership functions for each state variable as shown in Fig. 10. The Median membership function is defined as:

$$M_m(x_i) = e^{-(\frac{x_i - center_j}{width_j})^2} \qquad (40)$$

The positive and negative membership functions (M_p and M_n) are defined as:

$$M_p(x_i)=\begin{cases} e^{-\left(\frac{x_i-center_i(3)}{width_i(3)}\right)^2} & for \quad x_i < center_i(3) \\ 1 & else \end{cases} \qquad (41)$$

$$M_n(x_i)=\begin{cases} e^{-\left(\frac{x_i-center_i(1)}{width_i(1)}\right)^2} & for \quad x_i > center_i(1) \\ 1 & else \end{cases}$$

For state x_1, $center_1=[-0.4 \quad 0 \quad 0.4]$ and $width_1=[0.2 \quad 0.2 \quad 0.2]$. For state x_2, $center_2=[-1 \quad 0 \quad 1]$ and $width_2=[0.5 \quad 0.5 \quad 0.5]$.

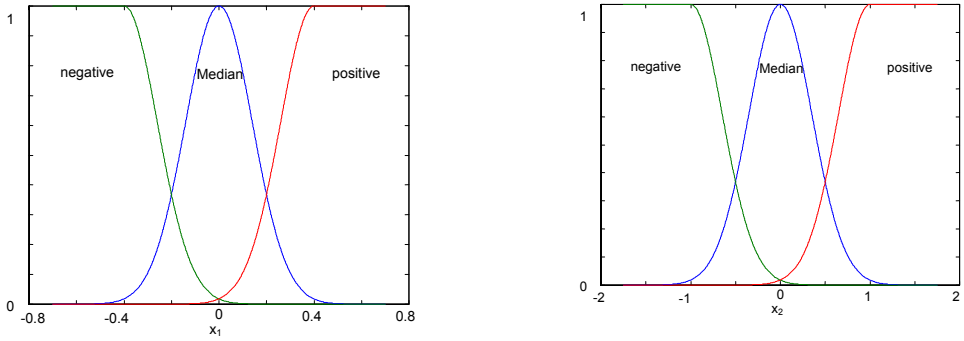

Fig. 8. The membership functions for inverted pendulum system

Following the design procedure, the intelligent closed-loop indirect adaptive controller design for inverted pendulum can be shown in the following steps:

[Step 1]: Apply a light input $u = 0.1sin(t)$ to excite the nonlinear uncertain system, then measure sufficient data information of $x(t)$, $\dot{x}(t)$ and $u(t)$ via (18) - (21).

[Step 2]: Apply ***Algorithm I*** to perform the dynamical optimal training of TS-type fuzzy model with least-squared initialization. The initial parameters for the nine linear subsystems are:

$$A_1=\begin{bmatrix} 0 & 1 \\ 20.6251 & -0.6866 \end{bmatrix},$$

$$A_2=\begin{bmatrix} 0 & 1 \\ 18.9598 & 0.2051 \end{bmatrix},$$

$$A_3=\begin{bmatrix} 0 & 1 \\ 22.6312 & 1.5709 \end{bmatrix},$$

$$A_4=\begin{bmatrix} 0 & 1 \\ 20.9676 & -0.0116 \end{bmatrix},$$

$$A_5=\begin{bmatrix} 0 & 1 \\ 19.9498 & -0 \end{bmatrix},$$

$$A_6=\begin{bmatrix} 0 & 1 \\ 21.1859 & -0.0139 \end{bmatrix},$$

$$A_7=\begin{bmatrix} 0 & 1 \\ 20.6313 & 0.7084 \end{bmatrix},$$

$$A_8=\begin{bmatrix} 0 & 1 \\ 19.0468 & -0.2568 \end{bmatrix},$$

$$A_9=\begin{bmatrix} 0 & 1 \\ 23.1741 & -1.6748 \end{bmatrix}$$

$$B_1=\begin{bmatrix} 0 \\ 0.8174 \end{bmatrix}, \quad B_2=\begin{bmatrix} 0 \\ 0.8478 \end{bmatrix},$$

$$B_3=\begin{bmatrix} 0 \\ 0.8992 \end{bmatrix}, \quad B_4=\begin{bmatrix} 0 \\ 0.9327 \end{bmatrix},$$

$$B_5=\begin{bmatrix} 0 \\ 0.9211 \end{bmatrix}, \quad B_6=\begin{bmatrix} 0 \\ 0.9072 \end{bmatrix},$$

$$B_7=\begin{bmatrix} 0 \\ 0.8236 \end{bmatrix}, \quad B_8=\begin{bmatrix} 0 \\ 0.8473 \end{bmatrix}$$

$$B_9=\begin{bmatrix} 0 \\ 0.8743 \end{bmatrix}$$

By applying Eq. (41) and fuzzy defuzzification, the initial state space of the linear system can be obtained and written as:

$$A_f = \begin{bmatrix} 0 & 1 \\ 20.6317 & 0.7069 \end{bmatrix},$$

$$B_f = \begin{bmatrix} 0 \\ 0.8238 \end{bmatrix},\ C = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, D = 0$$

[Step 3]: Specify closed-loop poles as -10 and –15 and apply ***Algorithm II*** to design an on-line stable tracking controller for the closed-loop system.

[Step 4]: Apply ***Algorithm III***, the Adaptive Rules, to update the TS-type fuzzy model and tracking controller to stabilize the closed-loop system in Fig. 5.

We assume the initial states of x_1 is [-0.5 1.2]^T. The reference trajectory for state x_1 is $y_r = 0.2sin(2\pi t)$ and the reference trajectory for state x_2 is $\dot{y}_r = 0.4\pi cos(2\pi t)$. Figure 16 shows the convergence of state x_1. Figure 9 also shows the disturbance rejection with our overall control algorithm.

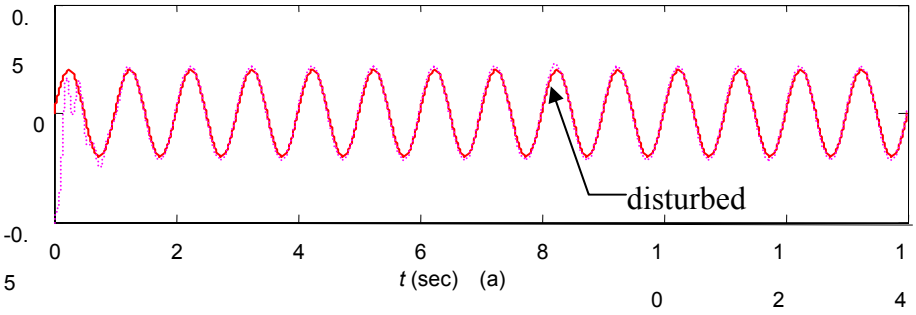

Fig. 9 Trajectories of reference y_r (solid line) with state x_1 (dotted line).

REFERENCES

[1] T. C. Hsia, "System Identification", Lexington Books, 1977.

[2] G. A. Rovithakis and M. A. Christodoulou, "Direct Adaptive Regulation of Unknown Nonlinear Dynamical Systems via Dynamic Neural Networks," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 1578-1594, Dec. 1995.

[3] H. O. Wang, K. Tanaka, M. F. Griffin, "An approach to fuzzy control of nonlinear systems: stability and design issues", *IEEE Transactions on Fuzzy Systems*, Vol. 4, p.p. 14-23, Feb. 1996.

[4] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and control," *IEEE Trans. Syst., Man, Cybern.*, Vol. 15, pp. 116-132, Jan./Feb., 1985.

[5] Shing-Jen Wu and Chih-Teng Lin, "Optimal Fuzzy Controller Design: Local Concept Approach," *IEEE Trans. On Fuzzy Systems*, Vol. 8, No. 2, pp. 171-183, April 2000.

[6] Shing-Jen Wu and Chih-Teng Lin, "Optimal Fuzzy Controller Design in Continuous Fuzzy System: Global Concept Approach," *IEEE Trans. On Fuzzy Systems*, Vol. 8, No. 6, pp. 713-729, Dec., 2000.

[7] Shing-Jen Wu and Chih-Teng Lin, "Discrete-Time Optimal Fuzzy Controller Design: Global Concept Approach," *IEEE Trans. On Fuzzy Systems*, Vol. 10, No. 1, pp. 21-38, Feb., 2002.

[8]. A. Isidori, Nonlinear Control Systems, 2nd ed., Berlin, Germany: Springer-Verlag, 1989.

[9] Y. G. Leu, T. T. Lee and W. Y. Wang, "Observer-based adaptive fuzzy-neural control for unknown nonlinear dynamical systems," *IEEE Trans. Syst., Man, Cybern. Part B: Cybernetics*, vol. 29, pp. 583-591, Oct., 1999.

[10] C. H. Wang, H. L. Liu and T. C. Lin, "Direct Adaptive Fuzzy-Neural Control With State Observer and Supervisory Controller for Unknown Nonlinear Dynamical Systems," *IEEE Trans. On Fuzzy Systems*, Vol. 10, No. 1, Feb., 2002.

[11] B. S. Chen, C. H. Lee, and Y. C. Chang, " H^{∞} tracking design of uncertain nonlinear SISO systems: Adaptive fuzzy approach," *IEEE Trans. Fuzzy Syst.* vol. 4, pp. 32-43, Feb. 1996.

[12]. A. V. Topalov and O. Kaynak, "Online Learning in Adaptive Neurocontrol Schemes with a Sliding Mode Algorithm," *IEEE Trans. Syst., Man, Cybern. Part B: Cybernetics*, Vol. 31, No. 3, pp. 445-450, June 2001.

[13] Chi-Hsu Wang, Han-Leih Liu, Chin-Teng Lin, "Dynamic optimal learning rates of a certain class of fuzzy neural networks and its applications with genetic algorithm", *IEEE Transactions on Systems, Man and Cybernetics, Part B,* Vol. 31, p.p. 467 -475 June 2001.