最佳線上即時智慧型控制系統的設計與作**(3/3)**

國科會計畫編號**: NSC-92-2213-E-009-056-**
成果發表會編號**: 94-2213-E-009-014**

計畫主持人**:** 王啓旭 教授
國立交通大學電機與控制工程系

計畫協同主持人**:** 李祖添教授
國立台北科技大學校長

以下內容爲本計劃三年來的執行成果。
預計發表於 **IEEE Trans. On Control Systems Technology**。

.

Real-Time Hardware Implementation of Intelligent Adaptive Fuzzy Neural Network Controller For Uncertain Nonlinear Systems

Abstract- In the thesis, true hardware implementation of an on-line intelligent adaptive TSK FNN controller is performed to control the planetary inverted pendulum. The hardware platform is dSPACE DS 1104 R&D control board under Windows XP running with MatLab. Excellent agreements have been obtained between theoretical simulation and hardware implementation. The effects of computational time delay for controller is also investigated through both software simulation and hardware emulation and by building SimuLink blocks in MatLab. The estimated maximum computational time delay can be quite practical for the industrial applications to choose cheaper hardware platform with less cost

Keywords: Fuzzy Neural Network, TS-type FNN Model, On-line Inteligent Adaptive Control, Delay Time.

1 Introduction

When we want to control a system, it usually has to be identified to have a mathematical model. But, in fact, it is often difficult to get the mathematical model, because the mathematical models of most systems are complex. Fuzzy neural network such as Takagi-Sugeno(TS)-Type FNN Model can simplify the process of identification of a system, and then the identified model can be controlled by the control technique. An on-line intelligent adaptive control was proposed in [1], which uses optimally trained TS-type FNN model [2] to identify an uncertain nonlinear system and then designs the controller with pole placement technique. Theoretically, applying the approach can obtain the TS-type FNN model of an uncertain nonlinear system in real-time environment and get a good controller to achieve the control specifications. The results of simulating some examples in [1] are good. But, it needs a large amount of computation to accomplish the training model, even though the optimal training [3] and the least square initialization [4] have shortened the process of training. In the implementation, computational delay will be an important factor in time delay.

The theme of the thesis is to implement a controller to control a system with on-line intelligent adaptive control for uncertain nonlinear system using TS-type FNN models. The planetary inverted pendulum [7] will be controlled within the hardware platform, DS 1104 R&D control board. As the excellent results of some examples presented in the thesis [1], our simulation for the planetary inverted pendulum controlled by the on-line intelligent adaptive control approach is fine, too. For implementation of the on-line optimal training controller, the design of on-line training process and the estimation for initial matrices for TS-type FNN model are the two major problems. The first one is how to train TS-type FNN model on-line. We use the "***memory***" and "***pulse generator***" blocks to build a training block to achieve the on-line training. We can determine the re-training time instant by setting the "***pulse generator***" block. For the second problem, it will be explained later.

2 TS-Type FNN Model for Uncertain Systems

A system can be represented by the following rules of TS-type FNN model [2], where F_{ij} is the FNN set.

Rule 1:

$$\textbf{If } z_1(t) \textbf{ is } F_{1,1} \textbf{ and} \ldots \textbf{and } z_g(t) \textbf{ is } F_{1,g}$$
$$\textbf{then } \underline{\dot{x}}_1(t) = A_1\underline{x}_1(t) + B_1\underline{u}(t)$$

$$\vdots \quad \vdots \tag{1}$$

Rule r:

$$\textbf{If } z_1(t) \textbf{ is } F_{r,1} \textbf{ and} \ldots \textbf{and } z_g(t) \textbf{ is } F_{r,g}$$
$$\textbf{then } \underline{\dot{x}}_r(t) = A_r\underline{x}_r(t) + B_r\underline{u}(t)$$

The above FNN system should be inferred as [1]:

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

where

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

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

.

and r is the number of rules for the uncertain nonlinear systems. Equation (2) shows that the system can be represented by a linear dynamical equation at any time instant, which can be inferred from the TS-type FNN model. The following Figure 1 shows the TS-type FNN model explained above.

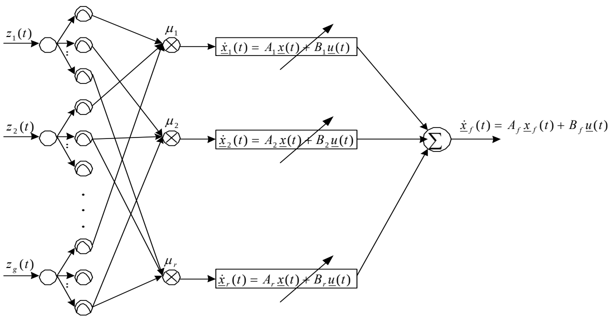

Figure 1. TS-type FNN model for Uncertain Systems

In Fig. 2-1, $z_1(t)$, $z_2(t)$, ..., $z_g(t)$, are the premise variables and r is the number of if-then rules and the A_i and B_i matrices are the locally linearized well-specified systems. The A_i and B_i matrices in Fig. 1 are called Jacobian matrices [1] with

$$\dot{\tilde{\underline{x}}}_i(t) = A_i\tilde{\underline{x}}(t) + B_i\tilde{\underline{u}}(t) \tag{4}$$

In the on-line intelligent adaptive control method in [1], the TS-type FNN model will be trained, so that the matrices A_i and B_i will be different at different time instants. Furthermore if better initial A_i and B_i matrices are chosen, it can take less time to converge. Hence, the least square estimation-technique [4] is chosen to estimate the initial value of the A_i and B_i [1]. Assume the order of the linear subsystems of TS-type FNN model is n and the number of input is m, then we must measure $n+m$ outputs to get the estimated initial matrices [1].

$$\begin{aligned} W_i(0) &= \begin{bmatrix} A_i^T \\ B_i^T \end{bmatrix} \\ Y &= \theta W_i(0) + \boldsymbol{\varepsilon} \\ W_i(0) &= (\theta^T\theta)^{-1}\theta^T Y \end{aligned} \tag{5}$$

Where the $\boldsymbol{\varepsilon}$ is the noise matrix, and θ and the set of outputs Y are presented as follows.

$$\theta = \begin{bmatrix} \underline{x}_1^T & \underline{u}_1^T \\ \underline{x}_2^T & \underline{u}_2^T \\ \vdots & \\ \underline{x}_{n+m}^T & \underline{u}_{m+n}^T \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} \dot{\underline{x}}_1^T \\ \dot{\underline{x}}_2^T \\ \vdots \\ \dot{\underline{x}}_{n+m}^T \end{bmatrix} \tag{6}$$

The TS-type FNN model for uncertain systems can be trained by using the on-line optimal training [1]. Because the dynamic optimal training [3] is very powerful for on-line disturbance rejection, it will be utilized in the on-line optimal training of TS-type FNN model for uncertain nonlinear systems.

Let the input training matrix R be

$$R = [x_1(t) \quad x_2(t) \quad \ldots \quad x_n(t) \quad u_1(t) \quad \ldots \quad u_2(t)]^T \in \Re^{(m+n)\times 1} \tag{7}$$

And the output matrix of the TS-type FNN model is

$$Y = \dot{\underline{x}}_f(t) = [\dot{x}_{f_1}(t) \quad \dot{x}_{f_2}(t) \quad \ldots \quad \dot{x}_{f_n}(t)]^T \in \Re^{n\times 1} \tag{8}$$

And the output matrix of the uncertain nonlinear system is

$$D = \dot{\underline{x}}(t) = [\dot{x}_1(t) \quad \dot{x}_2(t) \quad \ldots \quad \dot{x}_n(t)] \in \Re^{1xn} \tag{9}$$

The overall weighting matrix to include the uncertain A_i and B_i matrices can be shown as [1]:

$$W = \begin{bmatrix} \sum_{i=1}^{r} \mu_i A_i^T \\ \sum_{i=1}^{r} \mu_i B_i^T \end{bmatrix} = \sum_{i=1}^{r} \mu_i W_i \quad \text{and} \quad W_i = \begin{bmatrix} A_i^T \\ B_i^T \end{bmatrix} \tag{10}$$

Therefore the output of TS-type FNN model Y can be shown as:

$$Y = R^T W \tag{11}$$

It is therefore the purpose of on-line training to obtain the weighting matrix W.

First, the squared error J and the error function E have to be defined:

$$J = \frac{1}{2mn} \| \dot{\underline{x}}_f - \dot{\underline{x}} \|^2 \tag{12}$$

$$E = \dot{\underline{x}}_f(t) - \dot{\underline{x}}(t) = R^T W - D \tag{13}$$

From (12) and (13), we can have

$$J = \frac{1}{2mn} Tr(EE^T) \tag{14}$$

Then the dynamical learning rates β_{opt} for each iteration k can be determined by the dynamical optimal training in [3]. Define

$$J_{k+1} - J_k = a\beta^2 + b\beta \tag{15}$$

where

$$a = \frac{1}{2}(mn)^{-3} Tr\left[R^T R E_k E_k^T R^T R\right] > 0 \tag{16}$$

$$b = -(mn)^{-2} Tr\left[E_k^T R^T R E_k\right] < 0 \tag{17}$$

The roots of $a\beta^2 + b\beta = 0$ are (β_u, β_l). The optimal learning rate β_{opt} will be

$$\beta_{opt} = (\beta_u + \beta_l)/2 = \beta_t \tag{18}$$

This learning rate will not only guarantee the stability of the training process, but also have the fastest speed of convergence. Then, the on-line training rule for each subsystem is shown as

$$W_i(k+1) = W_i(k) - \beta_{opt,k} \frac{1}{mn} R E_k \tag{19}$$

The A_i and B_i matrices for each subsystem of TS-type FNN model can be updated simultaneously by using (20) at the beginning of each time interval. The final linear dynamic equation, (A_f, B_f), of the uncertain nonlinear system has been inferred from (2) at the beginning of any time interval. Then, the pole placement technique can be used

.

to design the controller [1]. The overall design process can be seen from the following Figure 2, which is from [1].

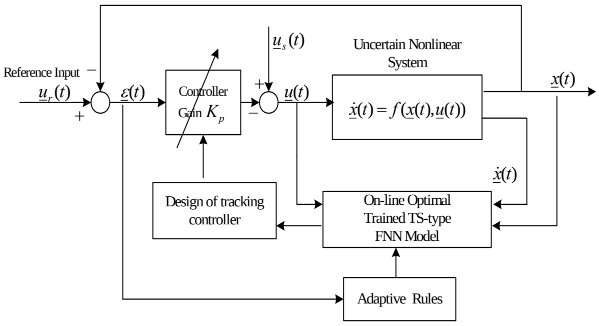

Figure 2. Overall design process

3 Design Algorithm

Step (1) Specify desired stable poles.

Step (2) Define the r nominal operating points and the corresponding membership functions for $\underline{x}(t)$ and $\dot{\underline{x}}(t)$. Use any input $u(t)$ to excite the uncertain nonlinear system and measure sufficient data information of $\underline{x}(t)$ and $\dot{\underline{x}}(t)$. In real implementation, it is easier to use step commands to trigger the system to get sufficient response data. Apply (5) to find the initial weighting matrix $W_i(0)$ of each subsystem for i=1, …, r.

Step (3) Apply pole placement to design the controller.

Step (4) If the norm of tracking error > e_1, a specified threshold, GOTO Step (2), Else GOTO Step (5).

Step (5) Measure on-line $\underline{x}(t)$ and $\dot{\underline{x}}(t)$. For i=1,…, r, applying (18) to find the optimal learning rates to train the weighting matrix of each subsystem. The optimal training must continue until relative errors of W_i are less than another pre-defined threshold e_2.

Step (6) GOTO Step (3)

Adaptive Rules for updating the closed-loop system in Figure 2-2

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

Apply all Steps to update the TS-type FNN model, us(t) and Kp

Else

Stand Still

End.

4 Real-Time Control of Planetary Inverted Pendulum

We perform the simulation for the inverted pendulum system in SimuLink. The control objective of this control system is to stabilize the inverted pendulum. The dynamic equation of the planetary train type inverted pendulum is shown in (20) [7].

$$\ddot{\theta}_0 = 149.3003 \sin\theta_0 + 2214.3u \qquad (20)$$

The control system model and the controller are shown in Figures 3 ~ 5. Figure 6 illustrates the trajectory for θ_0 with the initial condition $\left[\dot{\theta}_0 \quad \theta_0\right] = \left[-2 \quad 0.17\right]$.

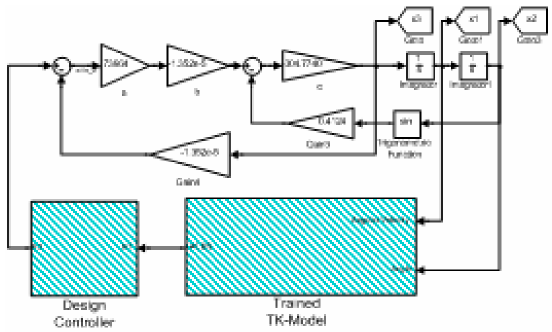

Figure 3. Simulation model of the planetary inverted pendulum system

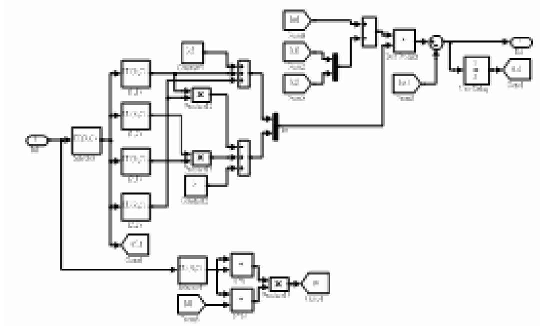

Figure 4. Block diagram of "***Design Controller***" in Figure 3

.

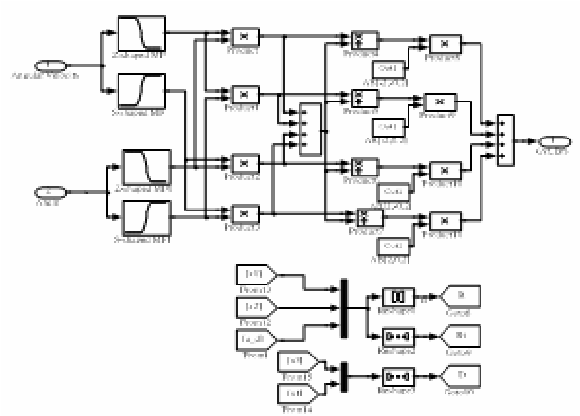

Figure 5. Block diagram of “***Trained TK-Model***” in Figure 3

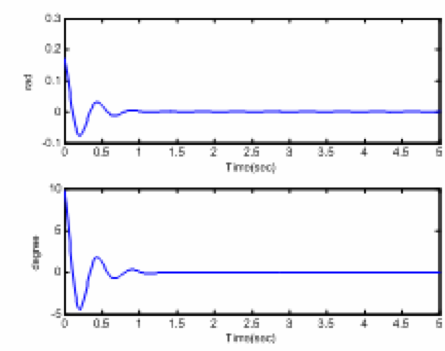

Figure 6. Trajectories for θ_0 in Figure3

we consider the effect of time delay on the angular control system of planetary inverted pendulum. We can find that $\dot{\theta}_0$ and θ_0 are always oscillatory and the longer is the delay time, the more oscillatory the responses will be. From Figure 7, the time delay of 0.022s is almost the maximum time delay for the inverted pendulum system with the on-line optimal trained controller.

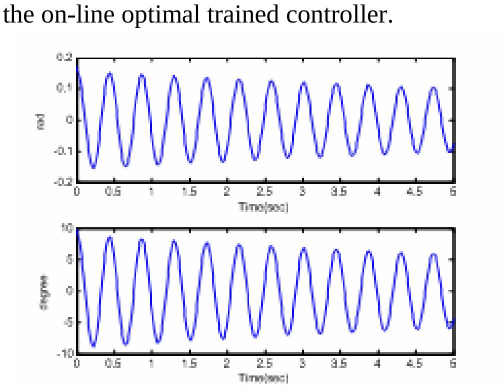

Figure 7. Trajectories of θ_0 with delay time = 0.022s

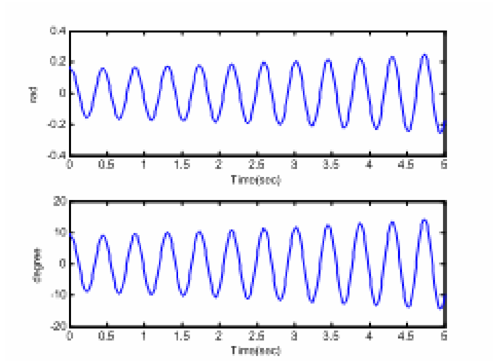

Figure 8. Trajectories of θ_0 with delay time = 0.023s

5 Hardware Implementation

We design the controller in SimuLink and generate the control program and then sent it into the hardware platform, DS 1104 R&D control board, to control the inverted pendulum system. The overall control process is shown in Figure 9.

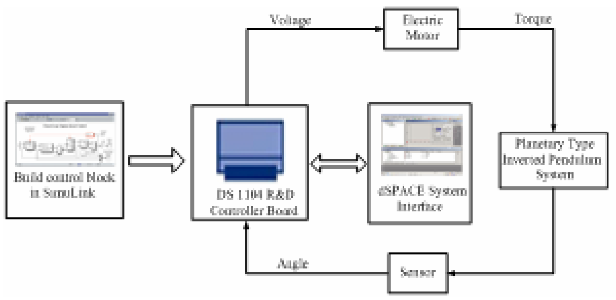

Figure 9. Overall hardware configuration using DS 1104

For the implementation of the controller for the planetary inverted pendulum system, we build the control block in SimuLink and generate the C code, and then sent the C code into the dSPACE hardware. We will operate the hardware to control the planetary inverted pendulum through the dSPACE system interface.

In the real planetary inverted pendulum system, the input is voltage. The mathematical relation between torque and voltage is unknown, so that the mathematical relation between the voltage and the angle of inverted pendulum is uncertain and nonlinear.

.

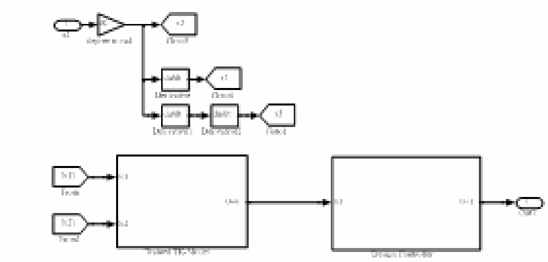

Figure 10. Block diagram of "***FNN controller***"

In Figure 10, the "***Trained TK-Model***" block produces the inferred (A_f, B_f) system matrices to represent the inverted pendulum system and the "***Design Controller***" block applies pole placement technique to design the controller. Figure 11 shows the detail of the "***Trained TK-Model***" block and the detail of the "***Design Controller***" block can be seen in Figure 12. Figure 13 shows the trajectory for θ_0.

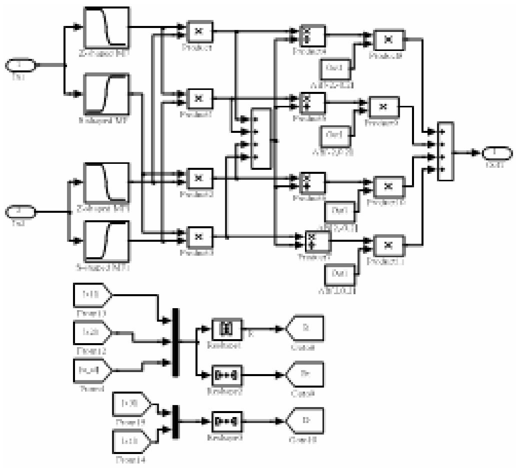

Figure 11. Block diagram of "***Trained TK-Model***"

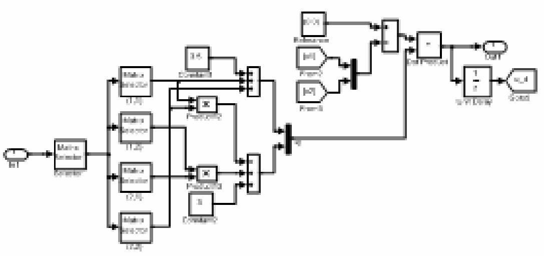

Figure 12. Block diagram of "***Design Controller***"

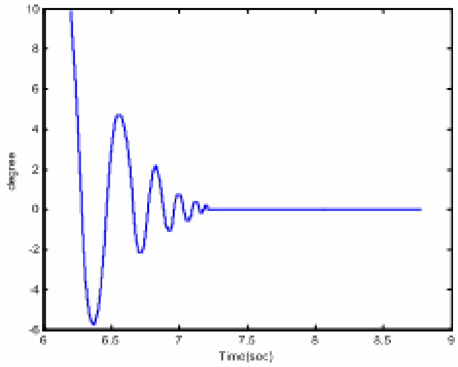

Figure 13. Trajectory of θ_0 during the stabilization of the pendulum

Figure 14 shows the trajectory of θ_0 for the stabilization of the pendulum with time delay = 0.019s.

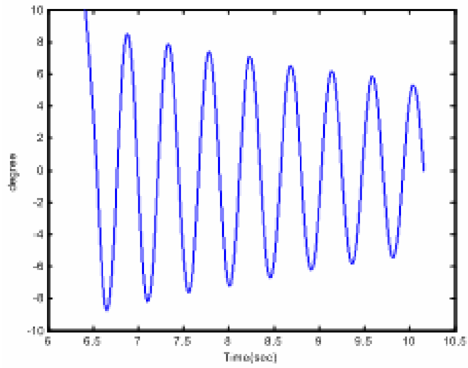

Figure 14. Trajectory of θ_0 to stabilize the pendulum with time delay = 0.019s

Figure 15 shows the trajectory of θ_0 for the stabilization of the pendulum with time delay =0.020s.

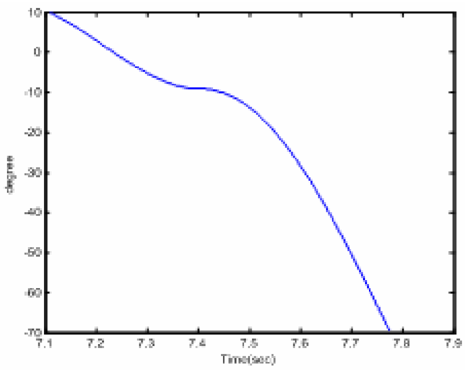

Figure 15. Trajectory of θ_0 to stabilize the pendulum with time delay = 0.020s

From Figure 14, 0.019s is almost the maximum allowed time delay for the real inverted pendulum control system with on-line optimal trained controller in the dSPACE control platform. The maximum allowed delay time for the inverted pendulum control system in simulation is about

.

0.022s. If we assume the maximum time delay from simulation is correct, then the computational time for the on-line controller is less than 0.003s (=0.022s-0.019s). This will change when different hardware platform is applied. Thus it is obvious that we can still use slower hardware platform to control the planetary inverted pendulum system to reduce the cost.

6 Conclusion

The on-line adaptive intelligent control for uncertain nonlinear systems by using TS-type FNN models proposed in [1] has been fully implemented using real hardware platform, i.e., DS 1104 R&D control board, under MatLab SimuLink. The planetary inverted pendulum was adopted as the real example to be controlled. The initial perturbation was done by various step commands to get the initial TS-type FNN model matrices. Then the on-line optimal training algorithm was implemented in SimuLink to drive the DS 1104 R&D control board to control the planetary inverted pendulum. Excellent results have been obtained to show the feasibility of hardware implementation of the control algorithm in [1]. The computational time delay to obtain the control signal has also been studied using real hardware emulations. The computational time delay of the planetary inverted pendulum using DS 1104 R&D control board has been estimated to be less than 0.003 seconds. This is within the maximum allowable computational time of 0.022 seconds, which is assumed to be correct by a pure computer simulation. This result can be very meaningful for industrial applications by choosing cheaper hardware platform with less cost to achieve the same control purpose.

References

[1] Shi-Hao Ker, "On-Line Intelligent Adaptive Control for Uncertain Nonlinear Systems using Optimally Trained TS-Type Fuzzy Models", MS Thesis, Department of Electrical and Control Engineering, National Chiao-Tung University, Hsin-Chu, Taiwan, 2002.

[2] 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.

[3] 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.

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

[5] Shing-Jen Wu and Chin-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 Chin-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] Shuang-Yuan Chen and Chen-Ren Lin, "Design and Control System Simulation of a Planetary Train Type Inverted Pendulum Mechanism," *第六屆機構與機器設計學術研討會暨海峽兩岸機構學術研討會*, Nov. 2003.

[8] "DS1104 R&D Controller Board, Installation and Configuration Guide", dSPACE, July 2001.

[9] "DS1104 R&D Controller Board, ControlDesk Experiment Guide", dSPACE, July 2001.

[10] M. Malek-Zavarei and M. Jamshidi, "Time-Delay Systems, Analysis, Optimization and Applications," Elsevier Science Publishers B.V., 1987

[11] R. N. Bateson, "Introduction to Control System Technology," Prentice Hall International, Inc, 2002

[12] 林群超, "自動控制系統設計與 MATLAB 語言," 全華科技圖書股份有限公司, 1999

[13] William J. Palm III, "Introduction to MATLAB 6 for Engineers," McGraw-Hill Education, Jan. 2003.

[14] Simon Haykin, "Neural Networks, A Comprehensive Foundation, Second Eedition," Prentice Hall International, Inc, 1999.

.