Journal of Intelligent and Robotic Systems **38:** 213–235, 2003.

Recurrent Neuro-Fuzzy Modeling and Fuzzy MDPP Control for Flexible Servomechanisms

CHORNG-SHYAN LIN
Chung-Shan Institute of Science and Technology

TACHUNG YANG
Department of Mechanical Engineering, Yuan-Ze University

YEONG-CHAU JOU and LIH-CHANG LIN*
Department of Mechanical Engineering, National Chung Hsing University, Taiwan, R.O.C.

(Received: 31 July 2002; in final form: 7 April 2003)

Abstract. This paper considers the nonlinear system identification and control for flexible servomechanisms. A multi-step-ahead recurrent neuro-fuzzy model consisting of local linear ARMA (autoregressive moving average) models with bias terms is suggested for approximating the dynamic behavior of a servomechanism including the effects of flexibility and friction. The RLS (recursive least squares) algorithm is adopted for obtaining the optimal consequent parameters of the rules. Within each fuzzy operating region, a local MDPP (minimum degree pole placement) control law with integral action can be constructed based on the estimated local model. Then a fuzzy controller composed of these local MDPP controls can be easily constructed for the servomechanism. The techniques are illustrated using computer simulations.

Key words: recurrent neuro-fuzzy model, TS fuzzy model, RLS algorithm, fuzzy MDPP control, servomechanism, flexibility, friction.

1. Introduction

Servomechanisms play a key role in mechatronics, and for satisfying stringent performance requirements their accurate models need to be built for dynamic analysis and control design. Due to the complex transmission mechanism, the complete model of a servomechanism is usually difficult to construct based on the first principles. Thus, system identification based on some suitable model structure and input–output data is a practical approach to developing a more accurate and tractable model. Many model-based control designs can be found in the literature, such as those surveyed in [5, 6]. Recently, fuzzy model-based control design has obtained great interest in the control community since fuzzy logic provides a simple and straightforward way to decompose the task of modeling and control design into a group of local tasks easier to handle, and elegant linear control design tools can be used [11, 12].

* Corresponding author.

For dynamic systems modeling, the recurrent neuro-fuzzy approach [8, 15] and the TS fuzzy system approach [11] are promising since a complex global nonlinear dynamic model instead of conventional static nonlinear mappings can be constructed in terms of only several fairly simple local models via fuzzy theory. One possible limitation of conventional "black-box" neural-network models is that they could be difficult to interpret and lack robustness for unseen data. Zhang and Morris [15] propose a type of recurrent neuro-fuzzy network for nonlinear process modeling. The process operation is partitioned into several fuzzy operating regions, and within each region, a local linear AutoRegressive Moving Average (ARMA) model is used to model the process. The global model output is obtained through the center average defuzzification [13] which is essentially the nonlinear smooth interpolation of local model outputs. This approach can improve model robustness and open up the "black-box" neural models. Process knowledge can be used to decompose the process operation into several fuzzy operating regions and to set up their membership functions definition. Process input/output data can be used to train the parameters of the recurrent neuro-fuzzy network (actually it is also a type of TS fuzzy model).

The TS fuzzy model proposed by Takagi and Sugeno [11] is a universal approximator and has found great potential for the fields about fuzzy systems and control [10, 12]. The main feature of a TS fuzzy model is to describe the complex dynamics of a nonlinear process by fuzzy IF–THEN rules with local linear input–output relations, such as linear state space model or linear discrete-time ARMA model. The overall fuzzy model of the nonlinear process is then achieved by fuzzy "blending" of the local linear system models. In general, there are two approaches to constructing a TS fuzzy model for a nonlinear process: (1) analytical derivation from its nonlinear physical model, and (2) systematic identification using its input–output data.

In this paper, we will consider the application of recurrent neuro-fuzzy modeling (that is also a type of TS fuzzy modeling) using the efficient RLS algorithm [2], and the design of a fuzzy MDPP (minimum degree pole placement) control based on the identified fuzzy model with local linear models for servomechanisms including the effects of transmission flexibility and nonlinear friction. The techniques are illustrated using representative computer simulations.

2. A Servomechanism with Flexibility and Friction Effects

The effects of transmission flexibility and complex nonlinear friction on a mechatronic system are two main problems to be considered for improving its dynamic performance and control accuracy. In this paper, we consider a torque-motor driven servomechanism shown in Figure 1.

The rotational motion of a vector-controlled three-phase brushless torque motor is transformed to the nut's translational motion of the ball screw through a pair of speed reduction gears. The linear motion of the nut is then used for driving the pay-

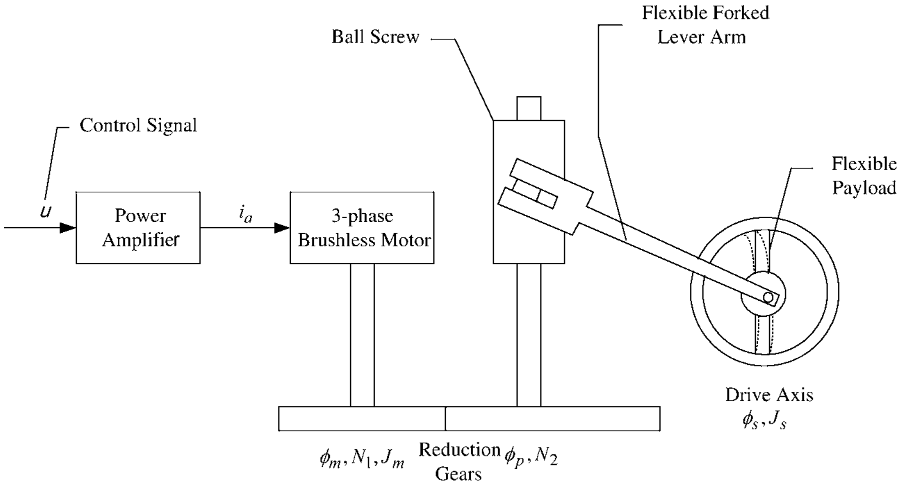


Figure 1. A torque motor driven servomechanism.

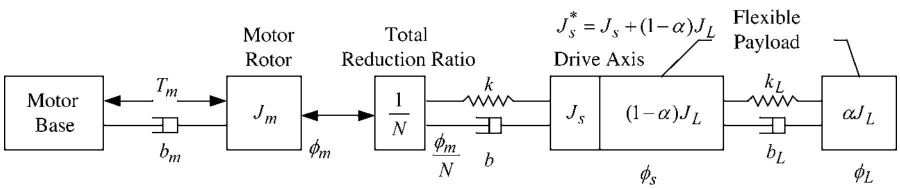


Figure 2. A simplified model for the servomechanism.

load via a pinned lever link (with flexibility) to perform its rotational motion within a finite range, e.g., $\pm 15°$, about the hub drive axis. While dynamic operation the ribbed payload may have severe vibration, and there exists transmission flexibility and serious friction in the whole servomechanism. In order to derive a tractable dynamics model for the servomechanism by the first principles, a simplified model is considered in Figure 2, where the payload flexibility is lumped using a simplifying coefficient α and the equivalent stiffness k_L and viscous damping coefficient b_L. For payloads of the type of large thin plates, $\alpha \approx 0.5$.

By the Newton laws of motion, we can derive the dynamics model of the servomechanism as follows:

$$\begin{aligned} &J_m\ddot{\phi}_m + b_m\dot{\phi}_m + \frac{1}{N}\left[b\left(\frac{\dot{\phi}_m}{N} - \dot{\phi}_s\right) + k\left(\frac{\phi_m}{N} - \phi_s\right)\right] + \tau_f = T_m, \\ &J_s^*\ddot{\phi}_s = b\left(\frac{\dot{\phi}_m}{N} - \dot{\phi}_s\right) + k\left(\frac{\phi_m}{N} - \phi_s\right) - b_L(\dot{\phi}_s - \dot{\phi}_L) - k_L(\phi_s - \phi_L), \\ &\alpha J_L\ddot{\phi}_L = b_L(\dot{\phi}_s - \dot{\phi}_L) + k_L(\phi_s - \phi_L) \end{aligned} \tag{1}$$

where τ_f is the equivalent friction torque reflected to the motor rotor; T_m is the motor driving torque, $T_m = K_T i_a$, $i_a = K_A u$, i_a and K_T are respectively the armature current and torque constant of the motor, and K_A and u are the current amplification gain and the input current command of the motor driver, respectively; J_m is the moment of inertia of the transmission mechanism reflected to the rotor side; $N = (N_2/N_1)\cdot(2\pi r/p)$ is the total speed reduction ratio from the motor rotor angle ϕ_m to the payload hub angle ϕ_s about its drive axis, here N_1 and N_2 are respectively the numbers of teeth at the rotor and screw sides, r is the effective length of the forked lever arm, and p is the pitch of the ball screw; $(\phi_m/N)-\phi_s$ represents the transmission flexibility; $J_s^* = J_s + (1-\alpha)J_L$, here J_s and J_L are the moments of inertia of the payload's drive axis and payload itself, respectively, and α is a simplifying coefficient for considering the flexibility effect of the payload; b_m, b, and b_L are the equivalent viscous damping coefficients (refer to Figure 2); k is the equivalent stiffness of the transmission system; k_L is the equivalent stiffness of the payload; and ϕ_L is the rotation angle of the payload.

The dynamic friction model suggested by Canudas de Wit et al. [3] is shown below and will be used in the dynamics simulation for generating the training data set:

$$\tau_f = \sigma_0 z + \sigma_1 \frac{\mathrm{d}z}{\mathrm{d}t} + \sigma_2 \dot{\phi}_m \tag{2}$$

where

$$\frac{\mathrm{d}z}{\mathrm{d}t} = \dot{\phi}_m - \frac{|\dot{\phi}_m|}{g(\dot{\phi}_m)} z, \tag{3}$$

$$\sigma_0 g(\dot{\phi}_m) = T_c + (T_s - T_c)\mathrm{e}^{-(\dot{\phi}_m/\omega_s)^2}. \tag{4}$$

Here, z is the average deflection of the bristles (which can deflect like springs) between the two contact surfaces and modeled by Equation (3); σ_0 and σ_1 are the stiffness and the damping coefficient of the bristles, respectively; σ_2 is the viscous coefficient; parameterization of $\sigma_0 g(\dot{\phi}_m)$ using (4) is used for describing the Stribeck effect; T_c is the Coulomb friction torque; T_s is the stiction torque; and ω_s is the Stribeck velocity. Thus, the friction torque can be expressed as

$$\tau_f = \left(\sigma_0 - \sigma_1 \frac{|\dot{\phi}_m|}{g(\dot{\phi}_m)}\right) z + (\sigma_1 + \sigma_2)\dot{\phi}_m. \tag{5}$$

By defining the state variables as $x_1 = \phi_m$, $x_2 = \dot{\phi}_m$, $x_3 = \phi_s$, $x_4 = \dot{\phi}_s$, $x_5 = \phi_L$, $x_6 = \dot{\phi}_L$, and $x_7 = z$, the state equation of the flexible servomechanism can be expressed as follows:

$$\begin{aligned}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= \ddot{\phi}_m = \frac{1}{J_m}\left\{ -b_m x_2 - \frac{1}{N}\left[b\left(\frac{1}{N}x_2 - x_4\right) + k\left(\frac{1}{N}x_1 - x_3\right)\right] \right. \\
&\qquad \left. - \left(\sigma_0 - \sigma_1 \frac{|x_2|}{g(x_2)}\right) x_7 - (\sigma_1 + \sigma_2) x_2 \right\} + \frac{1}{J_m} K_T K_A u, \\
\dot{x}_3 &= x_4,
\end{aligned} \tag{6}$$

$$\dot{x}_4 = \ddot{\phi}_s = \frac{1}{J_s^*}\left\{b\left(\frac{1}{N}x_2 - x_4\right) + k\left(\frac{1}{N}x_1 - x_3\right) - b_L(x_4 - x_6) \right.$$
$$\left. - k_L(x_3 - x_5)\right\},$$
$$\dot{x}_5 = x_6,$$
$$\dot{x}_6 = \ddot{\phi}_L = \frac{1}{\alpha J_L}\left\{b_L(x_4 - x_6) + k_L(x_3 - x_5)\right\},$$
$$\dot{x}_7 = x_2 - \frac{|x_2|}{g(x_2)}x_7$$

where

$$g(x_2) = \frac{T_c}{\sigma_0} + \frac{1}{\sigma_0}(T_s - T_c)\mathrm{e}^{-(x_2/\omega_s)^2}. \tag{7}$$

And the output variable y can be selected as

$$y = x_5. \tag{8}$$

3. Recurrent Neuro-Fuzzy Modeling for Flexible Servomechanisms

In this section, we will consider a type of recurrent neuro-fuzzy modeling (being also a type of TS fuzzy modeling), based on input/output data with sufficient persistent excitation (PE) for flexible servomechanisms with nonlinear friction.

3.1. STRUCTURE OF RECURRENT NEURO-FUZZY MODEL

A TS fuzzy model can be used to describe the complex dynamics of a nonlinear process by fuzzy IF–THEN rules with local linear state space model or discrete-time ARMA (autoregressive moving average) model. We consider the following TS fuzzy estimate model:

R_i: IF operating condition i

$$\text{THEN}\quad \hat{y}_i(t) = \sum_{j=1}^{n_o} -a_{ij}\hat{y}(t-j) + \sum_{j=1}^{n_i} b_{i,j-1}u(t-d-j+1) + c_i,$$
$$i = 1, 2, \ldots, M, \tag{9}$$

where M is the number of fuzzy rules; $u(t)$ is the input at time instant tT, T is the sampling period; $\hat{y}(t)$ is the total output of the TS fuzzy model at time tT; $\hat{y}_i(t)$ is the output value suggested by the ith local ARMA model at time tT; n_o and n_i are the orders of the autoregressive (feedback) and moving average (feedforward) parts of the local ARMA structure, respectively; $d = n_o - (n_i - 1)$ is the time delay; a_{ij}, $j = 1, 2, \ldots, n_o$, are the (AR) feedback coefficients, and $b_{i,j-1}$, $j = 1, 2, \ldots, n_i$, are the (MA) feedforward coefficients for the ith local linear model; c_i is a bias term for the ith local model. The local models adopted are multi-step-ahead prediction models. Thus, the suggested TS fuzzy estimate model can also be used for long-term global prediction about the system output variable. The total output of the TS fuzzy model can be obtained through center average defuzzification [12, 13] as

$$\hat{y}(t) = \frac{\sum_{i=1}^{M} \mu_i \hat{y}_i(t)}{\sum_{i=1}^{M} \mu_i} = \sum_{i=1}^{M} \xi_i \hat{y}_i(t) \tag{10}$$

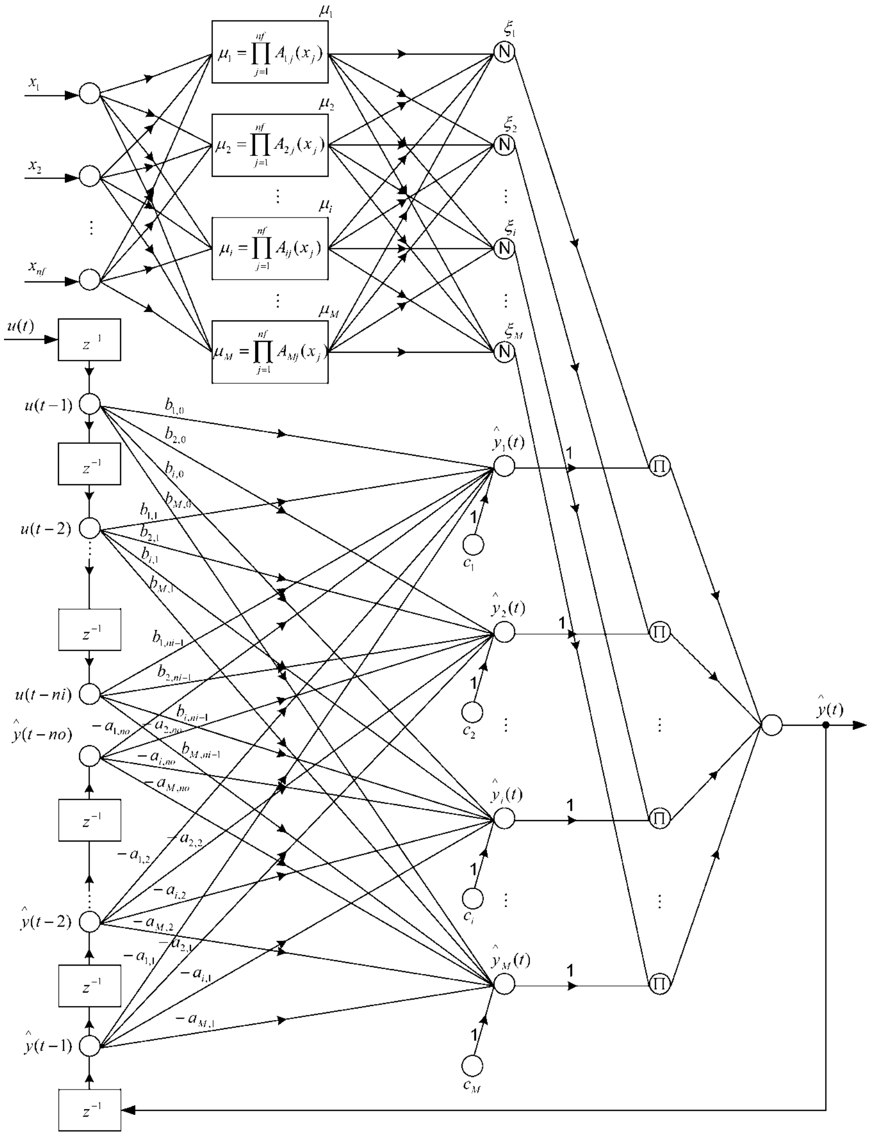


Figure 3. Network representation of the recurrent neuro-fuzzy model.

where μ_i is the matching degree (or degree of firing) of the ith model, and $\xi_i = \mu_i / \sum_{i=1}^{M} \mu_i$ is its normalized matching degree. Substituting the consequent local output $\hat{y}_i(t)$ in Equation (10), we can obtain

$$\hat{y}(t) = \sum_{i=1}^{M}\sum_{j=1}^{n_o} -\Big[\xi_i \hat{y}(t-j)\Big]a_{ij} + \sum_{i=1}^{M}\sum_{j=1}^{n_i}\Big[\xi_i u(t-d-j+1)\Big]b_{i,j-1} + \sum_{i=1}^{M} \xi_i c_i. \tag{11}$$

The antecedent "operating condition i" of the ith rule means the ith fuzzy operating region of the process that can be defined using fuzzy sets on the operating variables. If there are n_f operating variables, x_i, $i = 1, 2, \ldots, n_f$, then the ith rule can be represented as

R_i: IF x_1 is A_{i1} and x_2 is A_{i2} and … and x_{n_f} is A_{i,n_f}

$$\text{THEN} \quad \hat{y}_i(t) = \sum_{j=1}^{n_o} -a_{ij}\hat{y}(t-j) + \sum_{j=1}^{n_i} b_{i,j-1}u(t-d-j+1) + c_i, \quad i = 1, 2, \ldots, M. \tag{12}$$

The matching degree μ_i for the ith rule can be evaluated as

$$\mu_i = \prod_{j=1}^{n_f} A_{ij}(x_j) \tag{13}$$

using the product inference engine [13, 14].

The above TS fuzzy model can be represented using the recurrent network with model output feedback as shown in Figure 3. While suitable training/learning algorithms used in the neural network field are adopted for the optimal parameters learning/estimation of this TS fuzzy model, it can then be called a recurrent neuro-fuzzy model [14, 15].

3.2. PARAMETER OPTIMAL ESTIMATION OF TS FUZZY MODEL

By defining the parameters vector and the regression vector for the ith local model as

$$\theta_i(t-1) = [a_{i1}, a_{i2}, \ldots, a_{i,n_o}, b_{i0}, b_{i1}, \ldots, b_{i,n_i-1}, c_i]^{\mathrm{T}}, \quad i = 1, 2, \ldots, M, \tag{14}$$

$$\varphi^{\mathrm{T}}(t-1) = [-\hat{y}(t-1) \ \ -\hat{y}(t-2) \ \ -\cdots \ \ -\hat{y}(t-n_o) \ \ u(t-d) \ u(t-d-1) \ \cdots \ u(t-d-n_i+1) \ 1], \tag{15}$$

the parameters vector and the regression vector for the whole fuzzy model can be defined as

$$\widehat{\Theta}(t-1) = \left[\theta_1^{\mathrm{T}}(t-1), \theta_2^{\mathrm{T}}(t-1), \ldots, \theta_M^{\mathrm{T}}(t-1)\right]^{\mathrm{T}}, \tag{16}$$

$$\Psi(t) = \left[\xi_1(t)\varphi^{\mathrm{T}}(t-1), \xi_2(t)\varphi^{\mathrm{T}}(t-1), \ldots, \xi_M(t)\varphi^{\mathrm{T}}(t-1)\right]^{\mathrm{T}}. \tag{17}$$

Then the TS fuzzy model output (11) can be expressed in a linear parameterization form as

$$\hat{y}(t) = \Psi^{\mathrm{T}}(t)\widehat{\Theta}(t-1). \tag{18}$$

Equation (18) is called a regression model for the TS fuzzy estimate model if the fuzzy sets definition in the antecedents of all the fuzzy rules is kept fixed based on process knowledge, and only ARMA parameters of the local models are to be estimated using some training input–output data set. The input $u(t)$ and output $y(t)$ data pairs with sufficient persistent excitation (PE) are usually obtained from an experiment. For illustration, we simply use simulated data pairs from a closed-loop simulation shown in Figure 4 for testing the suggested modeling technique, where a more complex controller than usual simple PID controller, such as an approximate feedback linearization controller is chosen for tracking a sufficient PE trajectory to obtain sufficiently excited data.

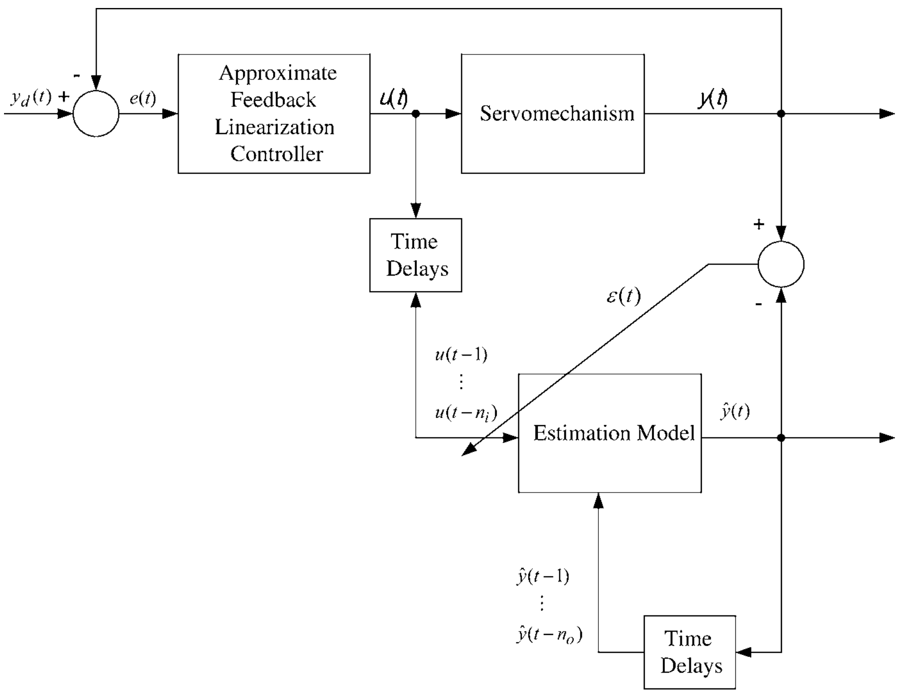


Figure 4. Estimation in a closed-loop configuration.

The parameter estimation problem is to determine the parameters of all the local ARMA parameters in such a way that the outputs computed from the TS fuzzy model agree as closely as possible with the real outputs of the servomechanism in the sense of least squares. Minimizing the following least-squares loss function,

$$V(\widehat{\Theta}, t) = \frac{1}{2}\sum_{i=1}^{t}\big(y(i) - \Psi^{\mathrm{T}}(i)\widehat{\Theta}(i-1)\big)^2 \tag{19}$$

the well-known recursive least-squares (RLS) estimation algorithm can be derived as follows [2, pp. 42–53]:

$$\widehat{\Theta}(t) = \widehat{\Theta}(t-1) + \mathbf{K}(t)\big(y(t) - \Psi^{\mathrm{T}}(t)\widehat{\Theta}(t-1)\big), \tag{20}$$

$$\mathbf{K}(t) = \mathbf{P}(t)\Psi(t) = \mathbf{P}(t-1)\Psi(t)\big(1 + \Psi^{\mathrm{T}}(t)\mathbf{P}(t-1)\Psi(t)\big)^{-1}, \tag{21}$$

$$\mathbf{P}(t) = \big(\mathbf{I} - \mathbf{K}(t)\Psi^{\mathrm{T}}(t)\big)\mathbf{P}(t-1). \tag{22}$$

The above RLS algorithm can be started with initial conditions

$$\widehat{\Theta}(0) = 0,$$
$$\mathbf{P}(0) = diag(\mathbf{P}_1(0), \mathbf{P}_2(0), \ldots, \mathbf{P}_M(0))$$

where the initial covariance matrices $\mathbf{P}_i(0)$, $i = 1, 2, \ldots, M$, are chosen positive definite and sufficiently large [2] for having fast convergence rate.

4. TS Fuzzy Model-Based Fuzzy MDPP Control Design

After the TS fuzzy model has been identified, within each fuzzy operating region, an individual MDPP (minimum-degree pole placement) control law can be synthesized based on the local ARMA model of that region. The local MDPP control law is selected as the following two degree-of-freedom general linear control:

$$R(q)u(t) = T(q)u_c(t) - S(q)y(t) \tag{23}$$

where R, S, and T are polynomials of q, R is assumed monic, and q is the forward shift operator; $u_c(t)$ is the command signal, and $u(t)$ and $y(t)$ are the input and output variables of the process, respectively; $-S/R$ and T/R are a negative transfer feedback operator and a feedforward transfer operator, respectively. The degrees of R, S, and T are selected as $\deg R = \deg S = \deg T = n_o - 1$, where n_o is the degree of the local model's AR part, that is, the order of the local ARMA model. Equation (23) can be written in terms of backward operator q^{-1} as

$$R^*(q^{-1})u(t) = T^*(q^{-1})u_c(t) - S^*(q^{-1})y(t) \tag{24}$$

where

$$\begin{aligned} R^*(q^{-1}) &= 1 + r_1 q^{-1} + \cdots + r_{n_o-1}q^{-(n_o-1)}, \\ S^*(q^{-1}) &= s_0 + s_1 q^{-1} + \cdots + s_{n_o-1}q^{-(n_o-1)}, \\ T^*(q^{-1}) &= t_0 + t_1 q^{-1} + \cdots + t_{n_o-1}q^{-(n_o-1)}. \end{aligned} \tag{25}$$

Thus the control law can be implemented as

$$u(t) = \left[1 - R^*(q^{-1})\right]u(t) + T^*(q^{-1})u_c(t) - S^*(q^{-1})y(t) \tag{26}$$

where $1 - R^*(q^{-1}) = -r_1 q^{-1} - \cdots - r_{n_o-1}q^{-(n_o-1)}$.

Consider the fuzzy controller with local MDPP control rules as follows:

R_i: IF operating condition i

$$\begin{aligned} \text{THEN} \quad & u_i(t) = \left[1 - R_i^*(q^{-1})\right]u(t) + T_i^*(q^{-1})u_c(t) - S_i^*(q^{-1})y(t), \\ & i = 1, 2, \ldots, M, \end{aligned} \tag{27}$$

where $u_i(t)$ is the control signal suggested by the ith MDPP control rule, and R_i^*, S_i^*, and T_i^* are the polynomials of the ith MDPP control law. The total control signal generated by the fuzzy MDPP controller can be obtained through center average defuzzification as

$$u(t) = \frac{\sum_{i=1}^{M} \mu_i u_i(t)}{\sum_{i=1}^{M} \mu_i} = \sum_{i=1}^{M} \xi_i u_i(t) \tag{28}$$

where μ_i is the matching degree (or degree of firing) of the ith control rule, and $\xi_i = \mu_i / \sum_{i=1}^{M} \mu_i$ is its normalized matching degree.

The ith local model of the fuzzy model for the servomechanism

$$\begin{aligned} \hat{y}_i(t) &= \sum_{j=1}^{n_o} -a_{ij}\hat{y}(t-j) + \sum_{j=1}^{n_i} b_{i,j-1}u(t-d-j+1) + c_i, \\ d &= n_o - (n_i - 1), \end{aligned}$$

can be written as

$$A_i^*(q^{-1})\hat{y}(t) = B_i^*(q^{-1})(u(t-d) + \nu_i(t-d)) \tag{29}$$

where

$$\begin{aligned} A_i^*(q^{-1}) &= 1 + a_{i1}q^{-1} + \cdots + a_{in_o}q^{-n_o}, \\ B_i^*(q^{-1}) &= b_{i0} + b_{i1}q^{-1} + \cdots + b_{i,n_i-1}q^{-(n_i-1)}, \\ \nu_i &= \frac{c_i}{b_{i0} + b_{i1} + \cdots + b_{i,n_i-1}}. \end{aligned} \tag{30}$$

For control design, estimated local model of Equation (29) can be represented as

$$A_i(q)y(t) = B_i(q)(u(t) + \nu_i) \tag{31}$$

where

$$\begin{aligned} A_i(q) &= q^{n_o} + a_{i1}q^{(n_o-1)} + \cdots + a_{in_o}, \\ B_i(q) &= b_{i0}q^{(n_i-1)} + b_{i1}q^{(n_i-2)} + \cdots + b_{i,n_i-1}. \end{aligned}$$

Notice that ν_i is a constant and can be considered as a step disturbance. Thus, the standard pole placement procedure can be modified for taking disturbance into account. A step disturbance can be modeled as

$$A_d^*(q^{-1})\nu_i = \nu_i\delta(t) \tag{32}$$

where $A_d^*(q^{-1}) = 1 - q^{-1}$, and $\delta(t)$ is the unit impulse at $t = 0$. Equation (32) can be expressed in terms of forward shift operator q as

$$A_d(q)\nu_i = \nu_i\delta(t+1) \tag{32a}$$

where $A_d(q) = q - 1$. With local controller (consequent part of Equation (27))

$$R_i(q)u_i(t) = T_i(q)u_c(t) - S_i(q)y(t) \tag{33}$$

and by substituting (33) in (31), we can obtain the local closed-loop system as

$$y(t) = \frac{B_i T_i}{A_i R_i + B_i S_i} u_c(t) + \frac{B_i R_i}{A_d(A_i R_i + B_i S_i)} \nu_i \delta(t+1). \tag{34}$$

In order to maintain a finite output $y(t)$, R_i must contain a factor A_d, i.e., $R_i = A_d R'$. First, find the solution R_i^0 and S_i^0 that satisfies the Diophantine equation for the ith local model

$$A_i R_i^0 + B_i S_i^0 = A_{c,i}^0 \tag{35}$$

where $A_{c,i}^0 = A_{o,i} A_{m,i} B_i^+$; $A_{o,i}$ is the observer polynomial (monic, and stable), and $\deg A_{o,i} = \deg A_i - \deg B_i^+ - 1$; B_i is factored as $B_i = B_i^+ B_i^-$, here $B_i^+(q)$ is a monic, stable, and well-damped factor with zeros to be shifted by the feed-forward compensation, and $B_i^-(q)$ is the other factor with unstable and weakly-damped zeros that cannot be shifted; and the ith local reference model is selected as $A_{m,i} y_m(t) = B_{m,i} u_c(t)$ with $A_{m,i}$ monic and stable, $\deg A_{m,i} = \deg A_i = n_o$, $\deg B_{m,i} = \deg B_i$, and $B_{m,i} = B_i^- B'_{m,i}$. Since

$$R_i = X_i R_i^0 + Y_i B_i, \tag{36}$$
$$S_i = X_i S_i^0 - Y_i A_i \tag{37}$$

satisfy the following equation

$$A_i R_i + B_i S_i = X_i A_{c,i}^0 \tag{38}$$

where X_i is a stable polynomial that represents the augmented closed-loop poles for disturbance rejection and Y_i can be arbitrary, we can select

$$R_i = X_i R_i^0 + Y_i B_i = A_d R'. \tag{39}$$

By the identity

$$X_i R_i^0 + Y_i B_i = A_d R', \tag{40}$$

we can determine R' and Y_i. Thus, from (39) we can obtain R_i, and using (37) we can obtain S_i. The feedforward polynomial T_i of the ith local control law can be obtained by the MDPP algorithm:

$$T_i = A_{o,i} B'_{m,i}. \tag{41}$$

Notice that the above obtained fuzzy MDPP controller (27) can compensate for the constant bias terms very well because that it possesses integral control action (since $R_i = A_d R'$, $A_d(q) = q - 1$).

5. Simulation Examples

In this section, the proposed recurrent neuro-fuzzy modeling and fuzzy MDPP control design for servomechanisms will be tested. Extensive simulations are conducted and only the representative cases are to be illustrated.

The parameters of the torque-motor driven servomechanism are selected as follows:

$$\begin{aligned}
&J_m = 8.1 \times 10^{-5}\ \mathrm{Kg\,m^2},\ J_s = 5.0 \times 10^{-3}\ \mathrm{Kg\,m^2},\ J_L = 6.8 \times 10^{-2}\ \mathrm{Kg\,m^2},\\
&J_s^* = J_s + (1-\alpha) J_L,\ \alpha = 0.5,\ K_L = 2.81 \times 10^4\ \mathrm{N\,m/rad},\\
&K = 2.8 \times 10^4\ \mathrm{N\,m/rad},\ b_L = 0,\ b_m = 2.7 \times 10^{-4}\ \mathrm{N\,m\,s/rad},\\
&b = 7.3 \times 10^{-4}\ \mathrm{N\,m\,s/rad},\ N = 166,\ K_T = 0.25\ \mathrm{N\,m/A},\\
&K_A = 20 \times 1.6\ \mathrm{A/V},\ \sigma_0 = 10^5\ \mathrm{N},\ \sigma_1 = \sqrt{10^5}\ \mathrm{N\,s},\ \sigma_2 = 0.4\ \mathrm{N\,m\,s},\\
&T_c = 1\ \mathrm{N\,m},\ T_s = 1.5\ \mathrm{N\,m},\ \omega_s = 0.001\ \mathrm{rad/s},\quad \text{and}\\
&T\,(\text{integration time step}) = 0.0004\ \mathrm{s}.
\end{aligned}$$

5.1. RECURRENT NEURO-FUZZY MODELING USING SIMULATED DATA

Usually the nonlinear friction torque is mostly governed by the relative speed between the two contact surfaces, and it can be approximated using piecewise linear friction characteristics, that is, $\tau_{f,i}(t) = d_i\dot{\theta}_s(t) + h_i$ within the ith operation region, where d_i and h_i are local viscous damping coefficient and Coulomb friction (intercept), respectively. For simplicity, thus the angular velocity of the payload $\dot{\theta}_L$ can be selected as the only operating variable of the servomechanism, and the TS fuzzy model is

R_i: IF $\dot{\theta}_L(t)$ is A_i

$$\begin{aligned}
\text{THEN}\quad &\hat{y}_i(t) = \sum_{j=1}^{n_o} -a_{ij}\hat{y}(t-j) + \sum_{j=1}^{n_i} b_{i,j-1}u(t-d-j+1) + c_i,\\
&i = 1, 2, \ldots, M,
\end{aligned} \tag{42}$$

where A_i is the ith fuzzy set defined in the operating variable range and represents the ith fuzzy operation region; $n_o = n_i = 5$ is selected since the servomechanism

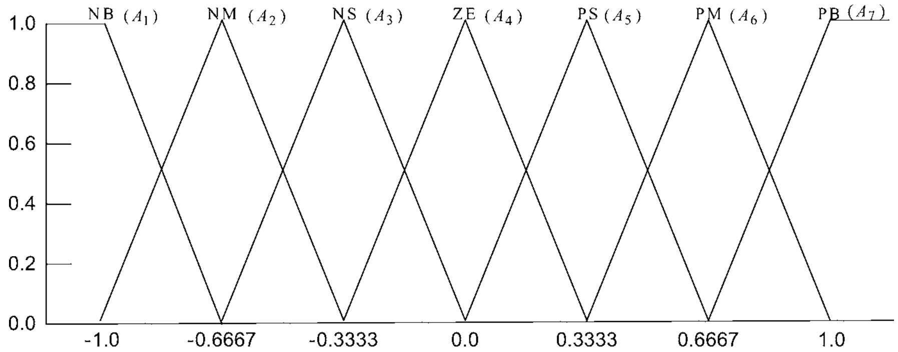


Figure 5. Definition of the fuzzy sets.

Table I. The estimated optimal a_{ij}

a_{ij}	$j = 1$	2	3	4	5
$i = 1$	−0.045404	−0.049882	−0.05385	−0.053408	−0.046356
2	−0.11569	−0.10541	−0.093058	−0.075361	−0.050062
3	−0.059179	−0.044554	−0.027555	−0.0089454	0.010030
4	−0.010015	−0.0052427	7.4873×10^{-5}	0.0047150	0.0082064
5	−0.058261	−0.042296	−0.024248	−0.0050522	0.014325
6	−0.12599	−0.11606	−0.10218	−0.081411	−0.052313
7	−0.037934	−0.040865	−0.042027	−0.039362	−0.03268

is a fifth-order system from input $u(t)$ to output $\dot{\theta}_L(t)$ and the number of finite zeros of a discretized dynamic system model is equal to the number of finite poles minus one [2, 1]. In the simulations, seven ($M = 7$) triangular fuzzy sets are defined for simplicity in the normalized universe of discourse $[-1, +1]$ as shown in Figure 5. The scaling factor for $\dot{\theta}_L$ is 0.0295. Using triangular membership functions for the definition of fuzzy sets has the advantage that at most two neighboring rules are fired for evaluation the model output at any time. The matching degree μ_i for the ith rule is

$$\mu_i = A_i(\dot{\theta}_L). \tag{43}$$

The training input–output data set is obtained using a closed-loop dynamic simulation with approximate feedback linearization control (refer to Figure 4) using command signal (desired payload velocity)

$$y_d(t) = G_{sf} \sum_{i=1}^{50} a_i \sin(2\pi f_i t),$$

where $0.01 \leqslant a_i \leqslant 3.01, 0.1 \leqslant f_i(\text{Hz}) \leqslant 199.9$, and G_{sf} is a scaling factor for obtaining the command signals with different ranges. If all the amplitudes

Table II. The estimated optimal b_{ij}

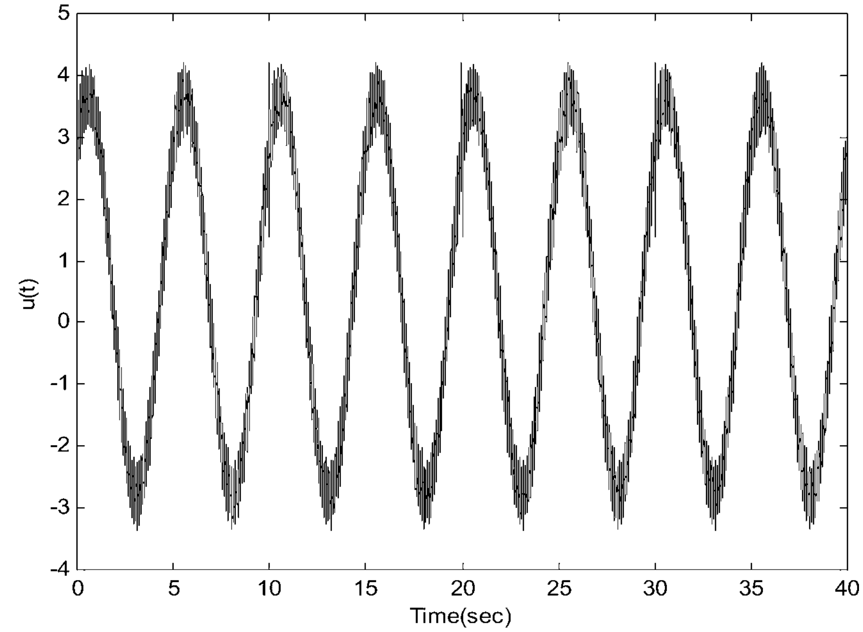
b_{ij}	$j=0$	1	2	3	4
$i=1$	2.2387×10^{-3}	7.0127×10^{-4}	6.1476×10^{-5}	4.0058×10^{-4}	1.6478×10^{-3}
2	1.3388×10^{-3}	-2.9473×10^{-4}	-9.1071×10^{-4}	-4.6606×10^{-4}	9.3295×10^{-4}
3	-2.929×10^{-5}	-1.2006×10^{-5}	3.4758×10^{-6}	-4.0119×10^{-6}	-3.8778×10^{-5}
4	3.5259×10^{-5}	-9.2916×10^{-6}	-2.3211×10^{-5}	-6.4954×10^{-6}	4.1056×10^{-5}
5	1.0784×10^{-4}	-3.5567×10^{-5}	-7.7507×10^{-5}	-3.6621×10^{-5}	7.6706×10^{-5}
6	1.0289×10^{-3}	-2.9643×10^{-4}	-7.8812×10^{-4}	-3.9792×10^{-4}	8.15×10^{-4}
7	1.5283×10^{-3}	6.9933×10^{-4}	4.1215×10^{-4}	7.1934×10^{-4}	1.5788×10^{-3}

Table III. The estimated optimal c_i

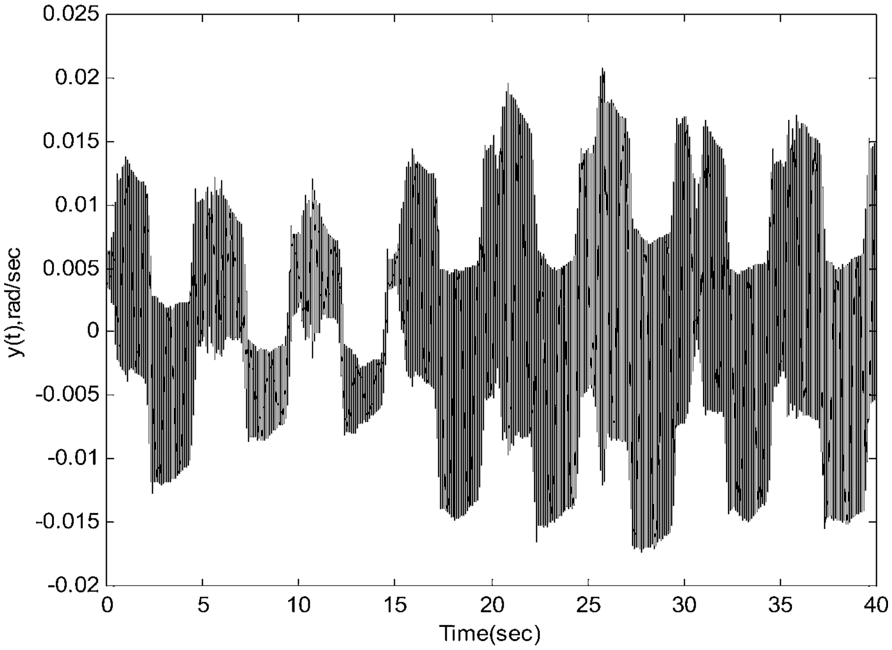
	$i=1$	2	3	4	5	6	7
c_i	-0.0044241	-0.011304	-0.0084998	-0.000058231	0.0085762	0.011491	0.0028688

and frequencies of the 50 sinusoidal components are selected different and not in integer multiples, the persistent exciting (PE) order of $y_d(t)$ is 100 (2×50) that is high enough to obtain the unique optimal parameters. The control signal $u(t)$ and the payload angular velocity $\dot{\theta}_L(t)$ are the input and output variables, and their simulated data pairs within [0, 40] seconds are used for training the recurrent neuro-fuzzy model using the RLS algorithm with initial values: $\widehat{\mathbf{\Theta}}_{77\times1}(0)=\mathbf{0}$, and $\mathbf{P}(0)=diag[\mathbf{P}_1(0),\mathbf{P}_2(0),\ldots,\mathbf{P}_7(0)]$, where $\mathbf{P}_i(0)=\mathrm{diag}[6.0\times10^{-5}\mathbf{I}_{5\times5}, 5.0\times10^{7}\mathbf{I}_{5\times5}, 5.0\times10^{7}]$, $i=1,2,\ldots,7$. The optimal consequent parameters are obtained as shown in Tables I–III. Notice that since the training data pairs ($u(t)$ and $y(t)$) are obtained by closed-loop simulations in the system structure of Figure 4 for tracking a complex desired command $y_d(t)$ composed of 50 sinusoidal components, the training data pairs contain the complex flexibility and friction effects of the servomechanism. Thus the estimated fuzzy model can approximate the whole dynamic behavior of the nonlinear plant as shown in the following two validation simulations.

For validating the above estimated TS fuzzy model (also a recurrent neuro-fuzzy model), first consider the test input signal $u(t)$ shown in Figure 6(a) and the corresponding plant output signal $y(t)$ shown in Figure 6(b). Notice that they are also obtained by a closed-loop simulation, and thus include the complex flexibility and friction effects. The model output $\hat{y}(t)$ computed by the above obtained fuzzy model is shown in Figure 6(c), where $y(t)$ is also shown for comparison. From Figures 6(c) and 6(d), where the estimation error $y(t)-\hat{y}(t)$ is shown, we know that the estimate performance is excellent since the high-PE real output signal $y(t)$ can be followed very well by the output estimate $\hat{y}(t)$.

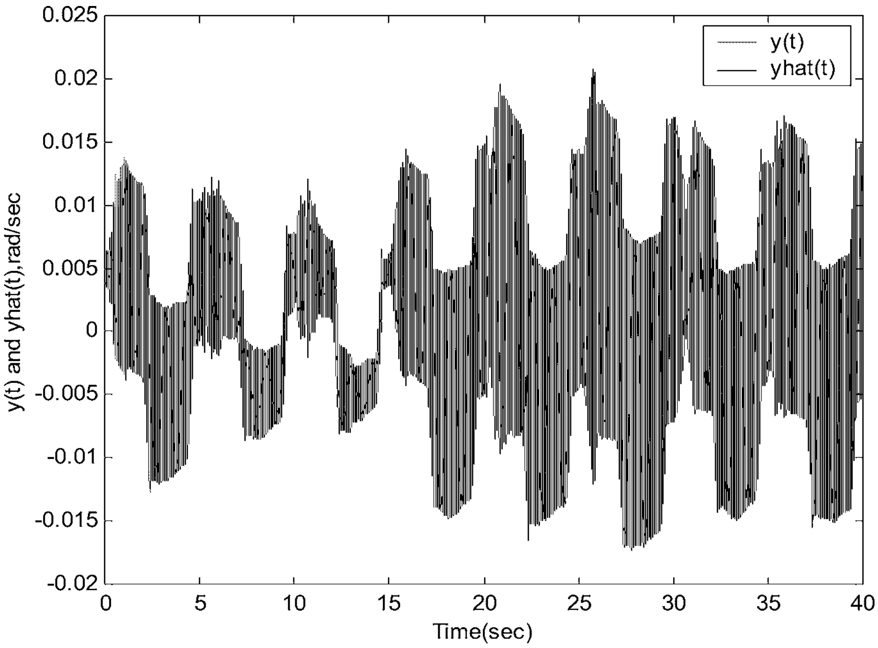


(a)

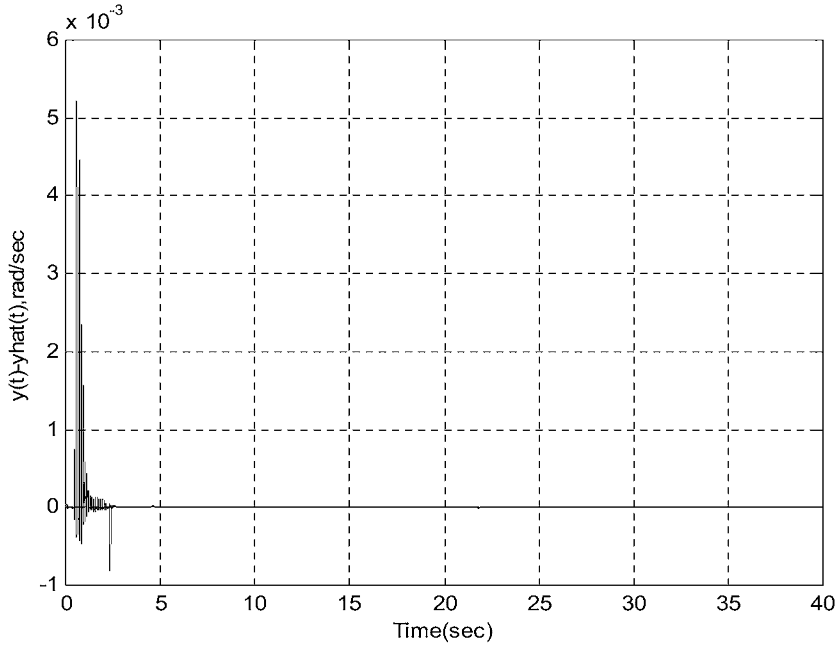


(b)

Figure 6. Model validation using the input/output signals shown in (a) and (b).

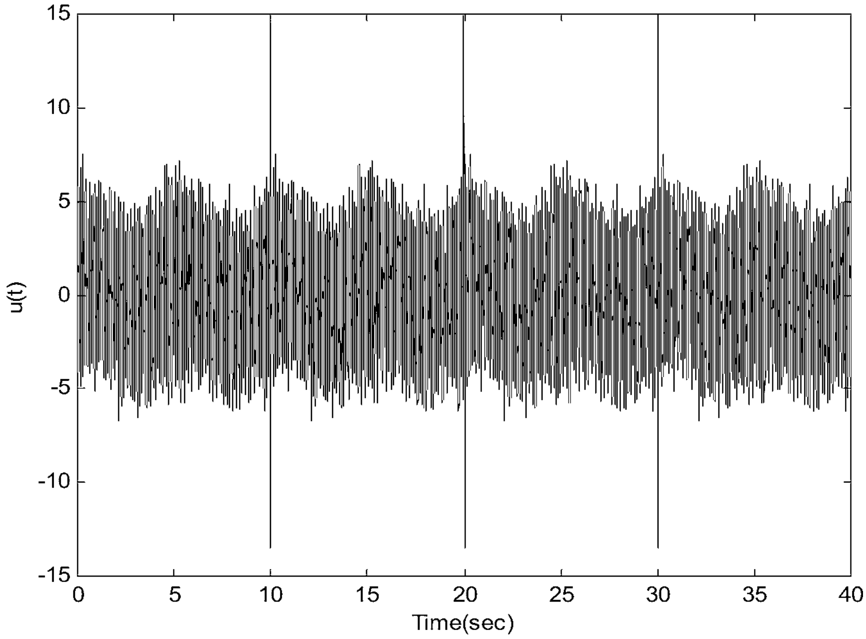


(c)

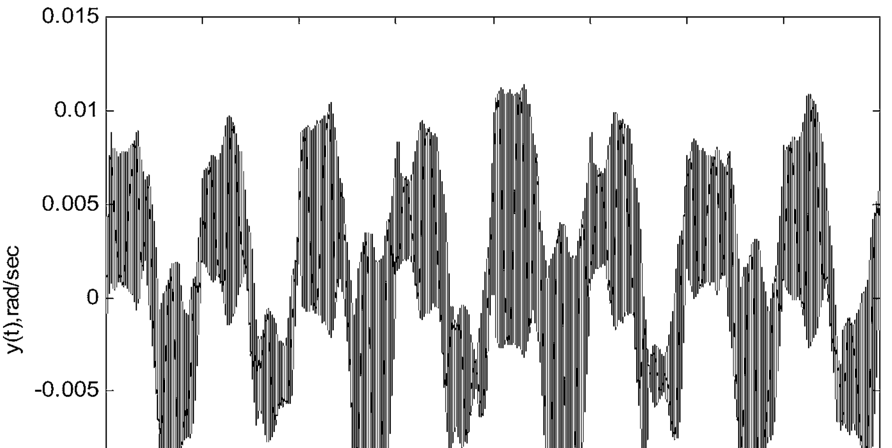


(d)

Figure 6. (Continued) (c) $y(t)$ and $\hat{y}(t)$, (d) estimate error $y(t) - \hat{y}(t)$.

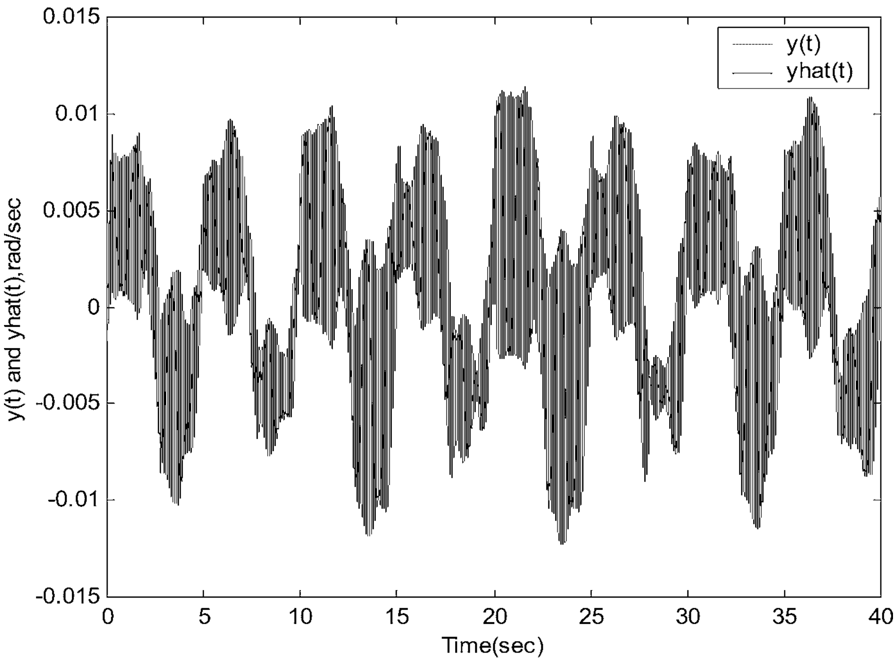


(a)

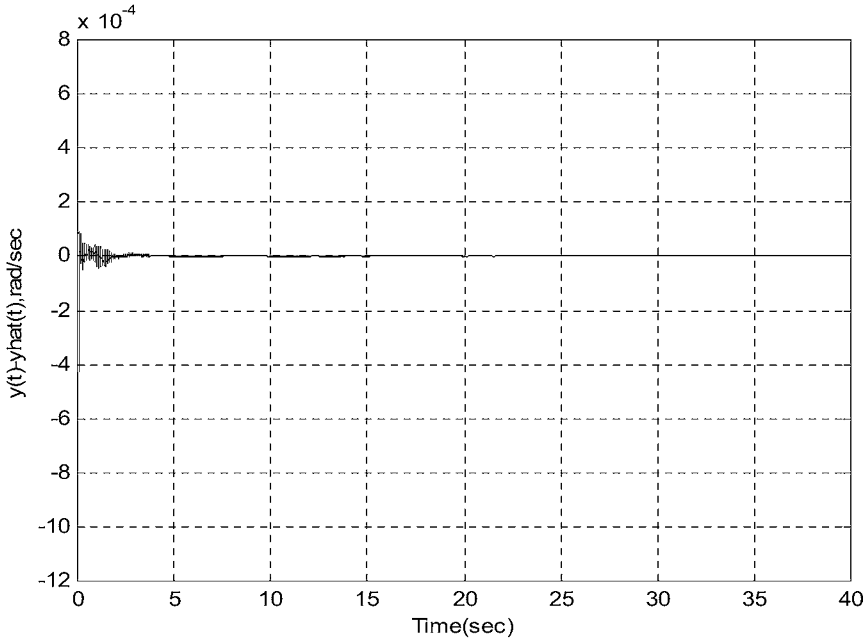


(b)

Figure 7. Model validation using the input/output signals shown in (a) and (b).



(c)



(d)

Figure 7. (Continued) (c) $y(t)$ and $\hat{y}(t)$, (d) estimate error $y(t) - \hat{y}(t)$.

Another test input signal $u(t)$ shown in Figure 7(a) and its corresponding plant output shown in Figure 7(b) are used for further validating the estimate performance. From Figures 7(c) and 7(d), the excellent estimate performance can also be seen. Notice that the above two input signals are not the training input signal used for the system identification. From these results we know that the thus obtained TS fuzzy model can have very high accuracy and can then be used for control design.

5.2. FUZZY MDPP CONTROL DESIGN BASED ON RECURRENT NEURO-FUZZY MODEL

Within each operation region, the local MDPP control law with integral action can be constructed based on the local linear ARMA model:

$$\frac{y(t)}{u(t)} = \frac{B_i(q)}{A_i(q)} = \frac{b_{i0}q^4 + b_{i1}q^3 + b_{i2}q^2 + b_{i3}q + b_{i4}}{q^5 + a_{i1}q^4 + a_{i2}q^3 + a_{i3}q^2 + a_{i4}q + a_{i5}}, \quad i = 1, 2, \dots, 7. \tag{44}$$

Since each local model (44) of the above obtained TS fuzzy model has unstable and weakly damped zeros, no zeros are canceled in the design procedure in this simulation study. Thus, the factorization of $B_i = B_i^+ B_i^-$ is simple and with $B_i^+ = 1$, and $B_i^- = B_i$. The reference models for the local models are selected as

$$\frac{B_{m,i}(q)}{A_{m,i}(q)} = \frac{B_{m,i}(q)}{A_m(q)}, \quad i = 1, 2, \dots, 7, \tag{45}$$

where

$$\begin{aligned} &A_m(q) = (q - p_1)^2(q - p_2)^2(q - p_3), \\ &B_{m,i}(q) = \beta_i B_i(q), \\ &p_1 = 0.02465(\varsigma = 1, \omega_n = 9257.4), \quad p_2 = 0.06829(\varsigma = 1, \omega_n = 6710.0), \\ &p_3 = 0.08729(\varsigma = 1, \omega_n = 6096.3), \end{aligned} \tag{46}$$

with

$$\beta_i = \frac{A_m(1)}{B_i(1)} = \frac{(1 - p_1)^2(1 - p_2)^2(1 - p_3)}{(b_{i0} + b_{i1} + b_{i2} + b_{i3} + b_{i4})}.$$

Since $\deg A_{o,i} = \deg A_i - \deg B_i^+ - 1 = 5 - 0 - 1 = 4$, $i = 1, 2, \dots, 7$, the observer polynomial can be selected as

$$A_{o,i}(q) = A_o(q) = (q - p_{o1})^2(q - p_{o2})^2 \tag{47}$$

with $p_{o1} = 0.01$, $p_{o2} = 0.05$. Since $\deg R_i^0 = \deg S_i^0 = \deg T_i = \deg A_i - 1 = 5 - 1 = 4$, let

$$\begin{aligned} &R_i^0(q) = q^4 + r_{i1}q^3 + r_{i2}q^2 + r_{i3}q + r_{i4}, \\ &S_i^0(q) = s_{i0}q^4 + s_{i1}q^3 + s_{i2}q^2 + s_{i3}q + s_{i4} \end{aligned} \tag{48}$$

and by Equation (45),

$$T_i(q) = \beta_i A_o(q). \tag{49}$$

The coefficients of $R_i^0(q)$ and $S_i^0(q)$ can be obtained by solving the Diophantine equation (35) through equating coefficients of equal powers of q.

To compensate for the step disturbances, we select $X_i = X = q + x_0$ with $x_0 = 0.0$, and $Y_i = y_0^i$, $i = 1, 2, \ldots, 7$. By Equation (39) and letting $q = 1$, we can obtain

$$y_{0,i} = -\frac{(1 + x_0)R_i^0(1)}{B_i(1)} = -\frac{(1 + \sum_{i=1}^{4} r_{ij})}{\sum_{j=1}^{5} b_{i,j-1}}. \tag{50}$$

Then from Equations (37) and (39), we can find $R_i(q)$ and $S_i(q)$ for the ith local MDPP control law with integral action, $i = 1, 2, \ldots, 7$.

The above fuzzy MDPP controller is tested using the following two command signals:

$$\text{Command } A: \quad u_c(t) = \begin{cases} 0.05, & 0 \leqslant t < 5, \\ 0.0, & 5 \leqslant t < 10, \\ -0.05, & 10 \leqslant t < 15, \\ 0.0, & 15 \leqslant t \leqslant 20, \end{cases}$$

$$\text{Command } B: \quad u_c(t) = \begin{cases} 0.04t, & 0 \leqslant t \leqslant 1.25, \\ 0.05, & 1.25 \leqslant t \leqslant 5, \\ 0.05 - 0.04(t-5), & 5 \leqslant t \leqslant 7.5, \\ -0.05, & 7.5 \leqslant t \leqslant 11.25, \\ -0.05 + 0.04(t - 11.25), & 11.25 \leqslant t \leqslant 12.5, \\ 0.0, & 12.5 \leqslant t \leqslant 16. \end{cases}$$

The trajectory tracking performances using commands A and B are shown in Figures 8 and 9, respectively, where the results with no integral compensation (i.e., using only $R_i^0(q)$ and $S_i^0(q)$ and without compensation for the rejection of the local constant-disturbances) are also shown. From these results we can know that the fuzzy MDPP controller with integral action based on the estimated TS fuzzy model has very good tracking capability and is really better than that without integral compensation. Since the included integral control design can completely reject the local constant disturbances (the bias terms in the local models), the suggested fuzzy MDPP control can have satisfactory compensation for the Coulomb friction. Compared with other fuzzy control approaches, e.g., [5, 6] for flexible drive systems, the suggested fuzzy MDPP control design based on the estimated fuzzy model is much simpler and can still meet the control objective very well. Thus, the proposed design approach is much easier for practical applications.

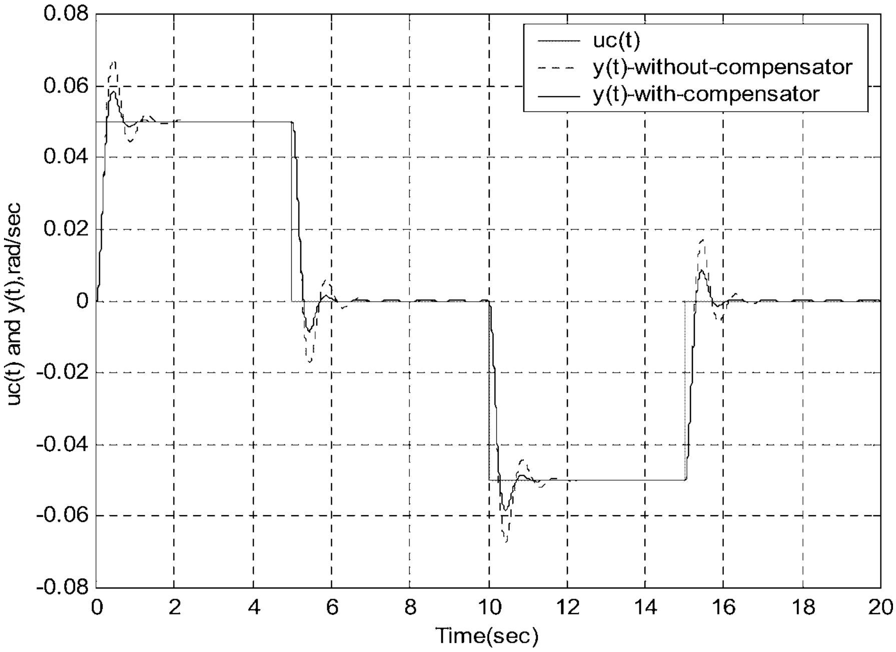


Figure 8. Fuzzy MDPP control result using Command *A*.

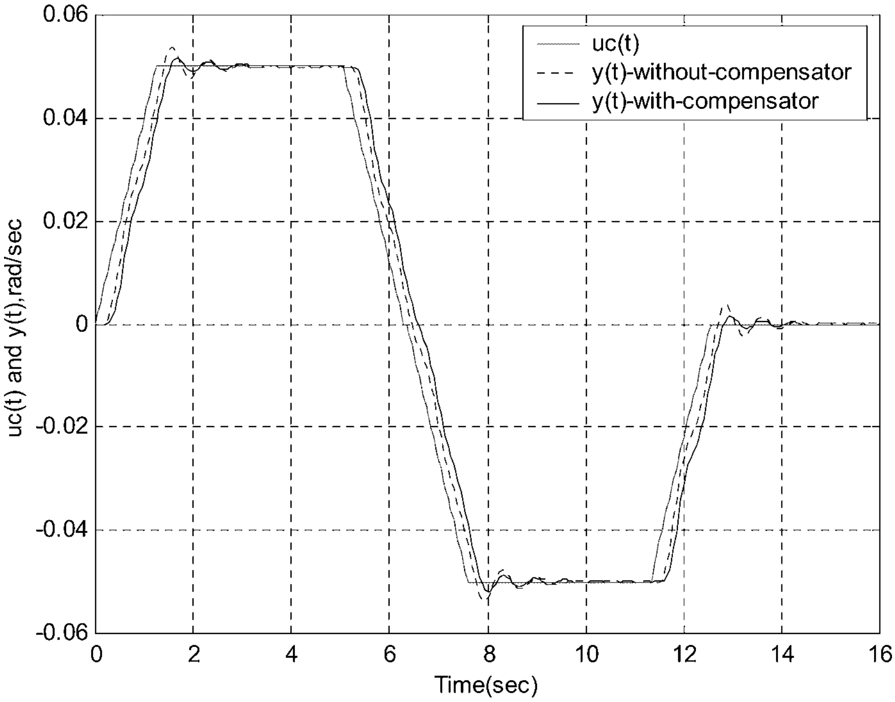


Figure 9. Fuzzy MDPP control result using Command *B*.

6. Conclusion

A simple and practical nonlinear system identification and control design methodology for flexible servomechanisms is proposed in this paper. A recurrent neuro-fuzzy model composed of local linear ARMA models with bias terms is used for approximating the dynamic behavior of a servomechanism with effects of flexibility and friction. After expressing the nonlinear fuzzy model in linear regression form, the most efficient RLS algorithm is adopted for estimating the consequent parameters of the TS fuzzy model. Corresponding to each fuzzy operation region, a local MDPP controller with integral action is constructed based on the obtained local model. Then a fuzzy controller with these local MDPP controllers can be easily constructed for the real servomechanism. The techniques are tested via representative computer simulation results. Since the suggested fuzzy model estimation is based on a multi-step-ahead prediction model structure, it can also be used for constructing a multi-step-ahead adaptive control for flexible servomechanisms, and it is an interesting future study topic.

Acknowledgements

This research was supported by National Science Council of R.O.C. under Grants NSC 89-2623-7-005-004 and NSC 90-2623-7-005-001. The authors would like to thank the referees for their constructive comments.

References

1. Åström, K. J., Hagander, P., and Sternby, J.: Zeros of sampled systems, *Automatica* **20** (1984), 31–38.
2. Åström, K. J. and Wittenmark, B.: *Adaptive Control*, 2nd edn, Addison-Wesley, Reading, MA, 1995.
3. Canudas de Wit, C., Olsson, H., Åström, K. J., and Lischinsky, P.: A new model for control of systems with friction, *IEEE Trans. Automat. Control* **40**(3) (1995), 419–425.
4. Cartinhour, J.: *Digital Signal Processing: An Overview of Basic Principles*, Prentice-Hall, Englewood Cliffs, NJ, 2000.
5. Lin, L. C. and Lin, Y. J.: Fuzzy-enhanced adaptive control for flexible drive systems with friction using genetic algorithms, *J. Intelligent Robotic Systems* **23**(2–4) (1998), 379–405.
6. Lin, L. C. and Lee, G. Y.: Hierarchical fuzzy control for C-axis of CNC turning centers using genetic algorithms, *J. Intelligent Robotic Systems* **25**(3) (1999), 255–275.
7. Mandic, D. P. and Chambers, J. A.: *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*, Wiley, Chichester, England, 2001.
8. Nerrand, O., Roussel-Ragot, P., Urbani, D., Personnaz, L., and Dreyfus, G.: Training recurrent neural networks: why and how? An illustration in dynamical process modeling, *IEEE Trans. Neural Networks* **5**(2) (1994), 178–184.
9. Oppenheim, A. V. and Schafer, R. W.: *Discrete-Time Signal Processing*, 2nd edn, Prentice-Hall, Englewood Cliffs, NJ, 1999.
10. Passino, K. M. and Yurkovich, S.: *Fuzzy Control*, Addison-Wesley, Reading, MA, 1998.
11. Takagi, T. and Sugeno, M.: Fuzzy identification of systems and its application to modeling and control, *IEEE Trans. Systems Man Cybernet.* **15**(1) (1985), 116–132.

12. Tanaka, K. and Wang H. O.: *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*, Wiley-Interscience, New York, 2001.
13. Wang, L. X.: *A Course in Fuzzy Systems and Control*, Prentice-Hall, Englewood Cliffs, NJ, 1997.
14. Yen, J. and Langari, R.: *Fuzzy Logic: Intelligence, Control, and Information*, Prentice-Hall, Englewood Cliffs, NJ, 1999.
15. Zhang, J. and Morris, A. J.: Recurrent neuro-fuzzy networks for nonlinear process modeling, *IEEE Trans. Neural Networks* **10**(2) (1999), 313–326.