We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Adaptive Neural Network Based Fuzzy Sliding Mode Control of Robot Manipulator

Ayca Gokhan AK and Galip Cansever
Marmara University, Yildiz Technical University
Turkey

1. Introduction

Robotic manipulators are highly nonlinear, highly time-varying and highly coupled. Moreover, there always exists uncertainty in the system model such as external disturbances, parameter uncertainty, and sensor errors. All kinds of control schemes, including the classical Sliding Mode Control (SMC) have been proposed in the field of robotic control during the past decades (Guo & Wuo, 2003).

SMC has been developed and applied to nonlinear system for the last three decades (Utkin, 1977, Edwards & Spurgeon, 1998). The main feature of the SMC is that it uses a high speed switching control law to drive the system states from any initial state onto a switching surface in the state space, and to maintain the states on the surface for all subsequent time (Hussain & Ho, 2004). So, there are two phase in the sliding mode control system: Reaching phase and sliding phase. In the reaching phase, corrective control law will be applied to drive the representation point everywhere of state space onto the sliding surface. As soon as the representation point hit the surface the controller turns the sliding phase on, and applies an equivalent control law to keep the state on the sliding surface (Lin & Chen, 1994).

The advantage of SMC is its invariance against parametric uncertainties and external disturbances. One of the SMC disadvantages is the difficulty in the calculation of the equivalent control. Neural network is used to compute of equivalent control. Especially multilayer feed forward neural network has been used. A few examples can be given as (Ertugrul & Kaynak, 1998, Tsai et al., 2004, Morioka et al., 1995). A Radial Basis Function Neural Networks (RBFNN) with a two-layer data processing structure had been adopted to approximate an unknown mapping function. Hence, similar to multilayered feed forward network trained with back propagation algorithm, RBFNN also known to be good universal approximator. The input data go through a non-linear transformation, Gaussian basis function, in the hidden layer, and then the functions responses are linearly combined to form the output (Huang et al., 2001). Also, back propagation NN has the disadvantages of slower learning speed and local minima converge.

By introducing the fuzzy concept to the sliding mode and fuzzifying the sliding surface, the chattering in SMC system can be alleviated, and fuzzy control rules can be determined systematically by the reaching condition of the SMC. There has been much research involving designs for fuzzy logic based on SMC, which is referred to as fuzzy sliding mode control (Lin & Mon, 2003).

In this study, a fuzzy sliding mode controller based on RBFNN is proposed for robot manipulator. Fuzzy logic is used to adjust the gain of the corrective control of the sliding mode controller. The weights of the RBFNN are adjusted according to some adaptive algorithm for the purpose of controlling the system states to hit the sliding surface and then slide along it.

The paper is organized as follows: In section 2 model of robot manipulator is defined. Adaptive neural network based fuzzy sliding mode controller is presented in section 3. Robot parameters and simulation results obtained for the control of three link scara robot are presented in section 4. Section 5 concludes the paper.

2. Model of Robot Manipulator

The dynamic model of an n-link robot manipulator may be expressed in the following Lagrange form:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(q,\dot{q}) = u(t) \tag{1}$$

where $q, \dot{q}, \ddot{q} \in R^n$ are the joint position, velocity, and acceleration vectors, respectively; $M(q) \in R^{nxn}$ denotes the inertia matrix; $C(q,\dot{q}) \in R^{nxn}$ expresses the coriolis and centrifugal torques, $G(q) \in R^n$ is the gravity vector; $F(q,\dot{q}) \in R^{nxn}$ is the unstructured uncertainties of the dynamics including friction and other disturbances; $u(t) \in R^{nx1}$ is the actuator torque vector acting on the joints.

3. Adaptive Neural Network Based Fuzzy Sliding Mode Control

Sliding Mode Control

If the desired system states are available, the desired angle of the manipulator joint are denoted by q_d. The control objective is to drive the joint position q to the desired position q_d. The tracking error equation can be written as follows:

$$e = q - q_d \tag{2}$$

Define the sliding surface of the sliding mode control design should satisfy two requirements, i.e., the closed-loop stability and performance specifications (Chen & Lin, 2002). A conventional sliding surface corresponding to the error state equation can be represented as:

$$s = \dot{e} + \lambda e \tag{3}$$

where λ is a positive constant.

In general, sliding mode control law can be represented as:

$$u = u_{eq} + u_c \tag{4}$$

where u_{eq} is the equivalent control law for sliding phase motion and u_c is the corrective control for the reaching phase motion. The control objective is to guarantee that the state trajectory can converge to the sliding surface. So, corrective control u_c is chosen as follows:

$$u_c = K sign(s) \tag{5}$$

where K is a positive constant. The *sign* function is a discontinuous function as follows:

$$sign(s) = \begin{cases} 1 & s > 0 \\ 0 & s = 0 \\ -1 & s < 0 \end{cases} \tag{6}$$

Notice that (5) exhibits high frequency oscillations, which is defined as chattering. Chattering is undesired because it may excite the high frequency response of the system. Common methods to eliminate the chattering are usually adopting the following. i) Using the saturation function. ii) Inserting a boundary layer (Tsai et al., 2004). In this paper, shifted sigmoid function is used instead of *sign* function:

$$h(s_i) = \frac{2}{1 + e^{-s_i}} - 1 \tag{7}$$

3.2 Fuzzy Sliding Mode Controller
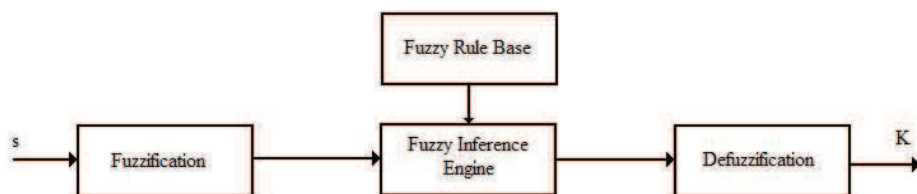Control gain K is fuzzified with the fuzzy system that shown in Fig. 1 (Guo & Wuo, 2003).

Figure 1. Diagram for a fuzzy system

The rules in the rule base are in the following form:

$$\text{IF } s_i \text{ is } A_i^m, \text{THEN } K_i \text{ is } B_i^m$$

where A_i^m and B_i^m are fuzzy sets. In this paper it is chosen that s_i has membership functions: NB, NM, NS, Z, PS, PM, PB and K_i has membership functions: Z, PS, PM, PB; where N stands for negative, P positive, B big, M medium, S small, Z zero. They are all Gaussian membership functions defined as follows:

$$\mu_A(x_i) = \exp\left(-\left(\frac{x_i - \alpha}{\sigma}\right)^2\right) \tag{8}$$

From the knowledge of the fuzzy systems, K_i can be written as

$$K_i = \frac{\sum_{m=1}^{M} \theta_{\lambda_i}^m \mu_{A^m}(s_i)}{\sum_{m=1}^{M} \mu_{A^m}(s_i)} = \theta_{ki}^T \psi_{ki}(s_i) \tag{9}$$

where $\theta_{ki} = \left[\theta_{ki}^1, \ldots, \theta_{ki}^m, \ldots, \theta_{ki}^M \right]^T$ is the vector of the center of the membership functions of K_i, $\psi_{ki}(s_i) = \left[\psi_{ki}^1(s_i), \ldots, \psi_{ki}^m(s_i), \ldots \psi_{ki}^M(s_i) \right]^T$ is the vector of the height of the membership functions of K_i in which $\psi_{ki}^m(s_i) = \mu_{A^m}(s_i) / \sum_{m=1}^{M} \mu_{A^m}(s_i)$, and M is the amount of the rules.

3.3 Radial Basis Function Network

A whole knowledge of the system dynamics and the system parameters is required to be able to compute the equivalent control (Ertugrul & Kaynak, 1998). This is practically very difficult. To be able to solve this problem, neural network can used to compute the equivalent control.

A RBFNN is employed to model the relationship between the sliding surface variable, s, and equivalent control, u_{eq}. In other words, sliding variable, s, will be used as the input signal for establishing a RBFNN model to calculate the equivalent control.

The Gaussian function is used as the activation function of each neuron in the hidden layer (10). The excitation values of these Gaussian functions are distances between the input values of the sliding surface value, s, and the central positions of the Gaussian functions (Huang et al., 2001).

$$g(x) = \exp\left(-\frac{\left\| s - c_j \right\|^2}{\sigma_j^2} \right) \tag{10}$$

where j is the j. neuron of the hidden layer, c_j is the central position of neuron j. σ_j is the spread factor of the Gaussian function.

Weightings between input and hidden layer neurons are specified as constant 1. Weightings between hidden and output layer neurons (w_j) are adjusted based on adaptive rule.

The output of the network is:

$$u_{eq} = \sum_{j=1}^{n} w_j g_j(s) \tag{11}$$

Based on the Lyapunov theorem, the sliding surface reaching condition is $s\dot{s} < 0$. If control input chooses to satisfy this reaching condition, the control system will converge to the

origin of the phase plane. Since a RBFNN is used to approximate the non-linear mapping between the sliding variable and the control law, the weightings of the RBFNN should be regulated based on the reaching condition, $s\dot{s} < 0$. An adaptive rule is used to adjust the weightings for searching the optimal weighting values and obtaining the stable converge property. The adaptive rule is derived from the steep descent rule to minimize the value of $s\dot{s}$ with respect to w_j. Then the updated equation of the weighting parameters is (Huang et al., 2001):

$$\dot{w}_j = -\tau \frac{\partial s(t)\dot{s}(t)}{\partial w_j(t)} \tag{12}$$

where τ is adaptive rate parameter. Using the chain rule, (12) can be rewritten as follows:

$$\dot{w}_j = \eta s(t)g(s) \tag{13}$$

where η is the learning rate parameter. The structure of RBFNN is shown in Fig. 2.
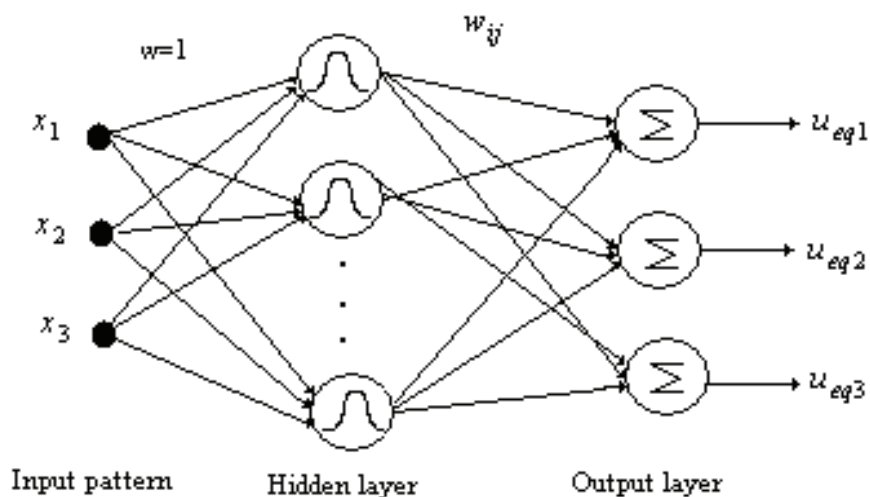
Figure 2. Radial Basis Function Neural Network (RBFNN)

3.3 Proposed Controller
The configuration of proposed controller is shown in Fig. 3. The control law for proposed controller is as (4) form. K gain of the corrective control u_c is adjusted with fuzzy logic and is the equivalent control u_{eq} is computed by RBFNN. The structure of RBFNN using this study has three inputs and three outputs. The number of hidden nodes is equal to 5. The weights of the RBFNN are changed with adaptive algorithm in adaptive law block. Outputs of the corrective control and RBFNN are summed and then applied to robotic manipulator.
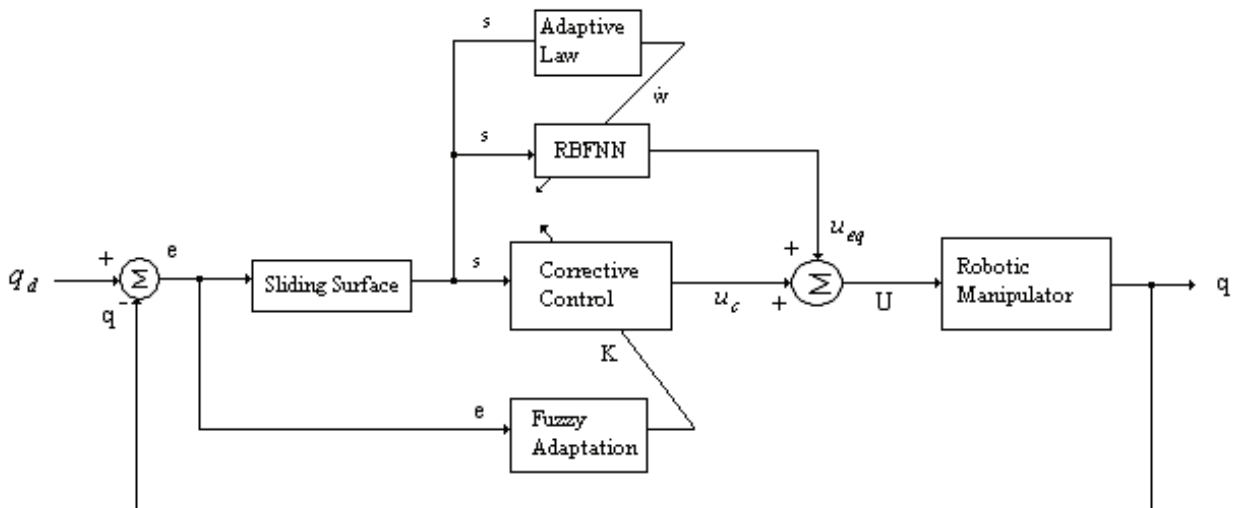
Figure 3. Block diagram of the proposed control system

4. Application

4.1 Robot Parameters
A three link scara robot parameters are utilized in this study to verify the effectiveness of the proposed control scheme. The dynamic equations which derived via the Euler-Lagrange method are presented as follows (Wai & Hsich, 2002):

$$
\begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + l_1 l_2 \sin(q_2) \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -m_3 g \end{bmatrix} + f(\dot{q}) + t_1(t) = u(t) \tag{14}
$$

where,

$$
M_{11} = l_1^2 \left(\frac{m_1}{3} + m_2 + m_3 \right) + l_1 l_2 (m_2 + 2m_3) \cos(q_2) + l_2^2 \left(\frac{m_2}{3} + m_3 \right)
$$

$$
M_{13} = M_{23} = M_{31} = M_{32} = 0
$$

$$
M_{12} = -l_1 l_2 \left(\frac{m_2}{2} + m_3 \right) \cos(q_2) - l_2^2 \left(\frac{m_2}{3} + m_3 \right) = M_{21}
$$

$$
M_{22} = l_2^2 \left(\frac{m_2}{3} + m_3 \right)
$$

$$
M_{33} = m_3
$$

$$
C_{11} = -\dot{q}_2 (m_2 + 2m_3)
$$

$$
C_{12} = -\dot{q}_2 \left(\frac{m_2}{2} + m_3 \right)
$$

$$C_{22} = -\dot{q}_2 \left(\frac{m_2}{2} + m_3 \right)$$

$$C_{13} = C_{22} = C_{23} = C_{31} = C_{32} = C_{33} = 0$$

in which q_1, q_2, q_3, are the angle of joints 1,2 and 3; m_1, m_2, m_3 are the mass of the links 1,2 and 3; l_1, l_2, l_3 are the length of links 1,2 and 3; g is the gravity acceleration. Important parameters that affect the control performance of the robotic system are the external disturbance $t_1(t)$, and friction term $f(\dot{q})$.

The system parameters of the scara robot are selected as following:

$$l_1 = 1.0m \quad l_2 = 0.8m \quad l_3 = 0.6m \quad m_1 = 1.0kg \quad m_2 = 0.8kg \quad m_3 = 0.5kg \quad g = 9.8$$

External disturbances are selected as:

$$t_1(t) = \begin{bmatrix} 5\sin(2t) \\ 5\sin(2t) \\ 5\sin(2t) \end{bmatrix} \tag{15}$$

In addition, friction forces are also considered in this simulation and are given as following:

$$f(\dot{q}) = \begin{bmatrix} 12\dot{q}_1 + 0.2sign(\dot{q}_1) + 3\sin(3t) \\ 12\dot{q}_2 + 0.2sign(\dot{q}_2) + 3\sin(3t) \\ 12\dot{q}_3 + 0.2sign(\dot{q}_3) + 3\sin(3t) \end{bmatrix} \tag{16}$$

4.2 Simulation Results

Central positions of the Gaussian function c_j are selected from -2 to 2. Spread factors σ_j are specified from 0.1 to 0.7. Initial weights of network are adjusted to zero.

The proposed fuzzy SMC based on RBFNN in Fig. 3 is applied to control the Scara robot manipulator.

The desired trajectories for three joint to be tracked are,

$$q_d(t) = 1 + 0.1\sin(pi * t) \tag{17}$$

Tracking position, tracking error, control torque and sliding surface of joint 1 is shown in Fig. 4 a, b, c and d respectively. Tracking position, tracking error, control torque and sliding surface of joint 2 is shown in Fig. 5 a, b, c and d respectively. Fig 6 a, b, c and d shows the tracking position, tracking error control torque and sliding surface of joint 3. Position of joint 1, 2 and 3 is reached the desired position at 2s, 1.5s and 0.5s, respectively.

In Fig 4c, 5c and 6c it can be seen that there is no chattering in the control torques of joints. Furthermore, sliding surfaces in Fig 4d, 5d, 6d converge to zero. It is obvious that the chattering in the sliding surfaces is eliminated.

5. Conclusion

The dynamic characteristics of a robot manipulator are highly nonlinear and coupling, it is difficult to obtain the complete model precisely. A novel fuzzy sliding mode controller based on RBFNN is proposed in this study. To verify the effectiveness of the proposed control method, it is simulate on three link scara robot manipulator.

RBFNN is used to compute the equivalent control. An adaptive rule is employed to on-line adjust the weights of RBFNN. On-line weighting adjustment reduces data base requirement. Adaptive training algorithm were derived in the sense of Lyapunov stability analysis, so that system-tracking stability of the closed-loop system can be guaranteed whether the uncertainties or not. Using the RBFNN instead of multilayer feed forward network trained with back propagation provides shorter reaching time.

In the classical SMC, the corrective control gain may choose larger number, which causes the chattering on the sliding surface. Or, corrective control gain may choose smaller number, which cause increasing of reaching time and tracking error. Using fuzzy controller to adjust the corrective control gain in sliding mode control, system performance is improved. Chattering problem in the classical SMC is minimized.

It can be seen from the simulation results, the joint position tracking response is closely follow the desired trajectory occurrence of disturbances and the friction forces.

Simulation results demonstrate that the adaptive neural network based fuzzy sliding mode controller proposed in this paper is a stable control scheme for robotic manipulators.
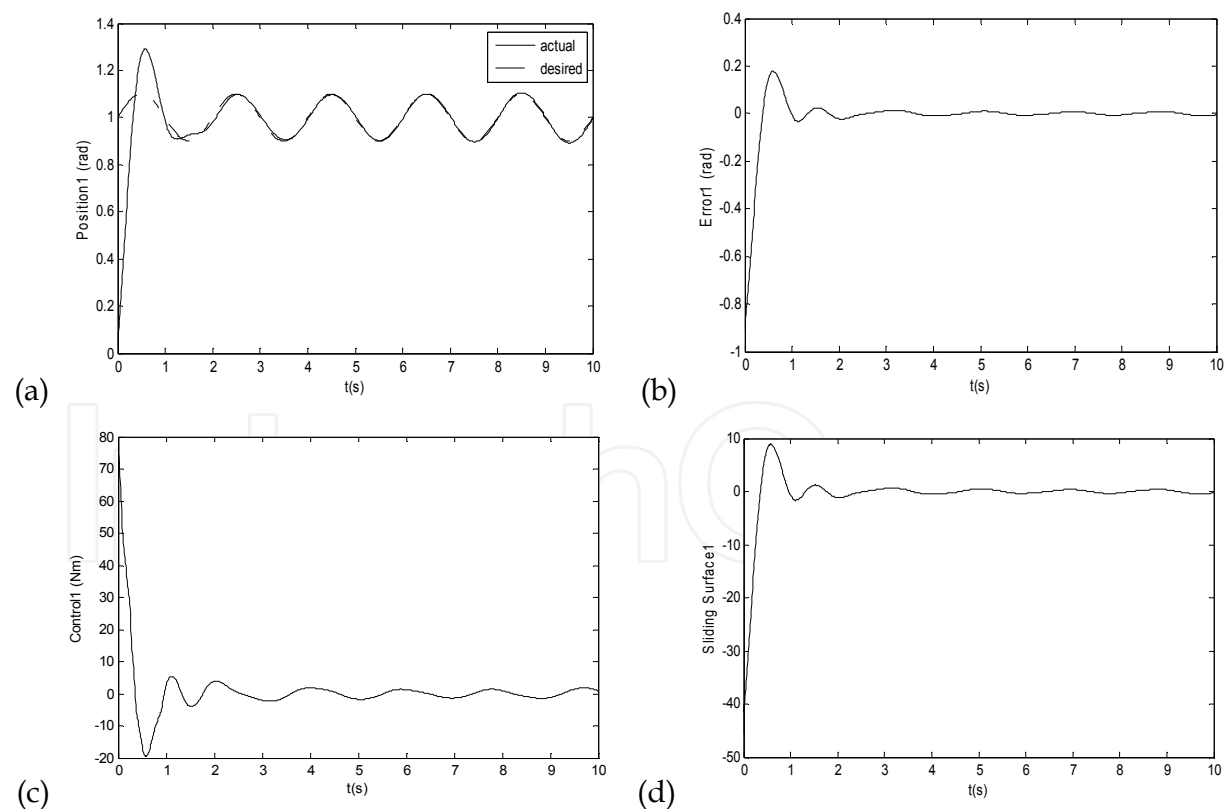

Figure 4. Simulation results for joint 1: (a) tracking response; (b) tracking error; (c) control torque; (d) sliding surface
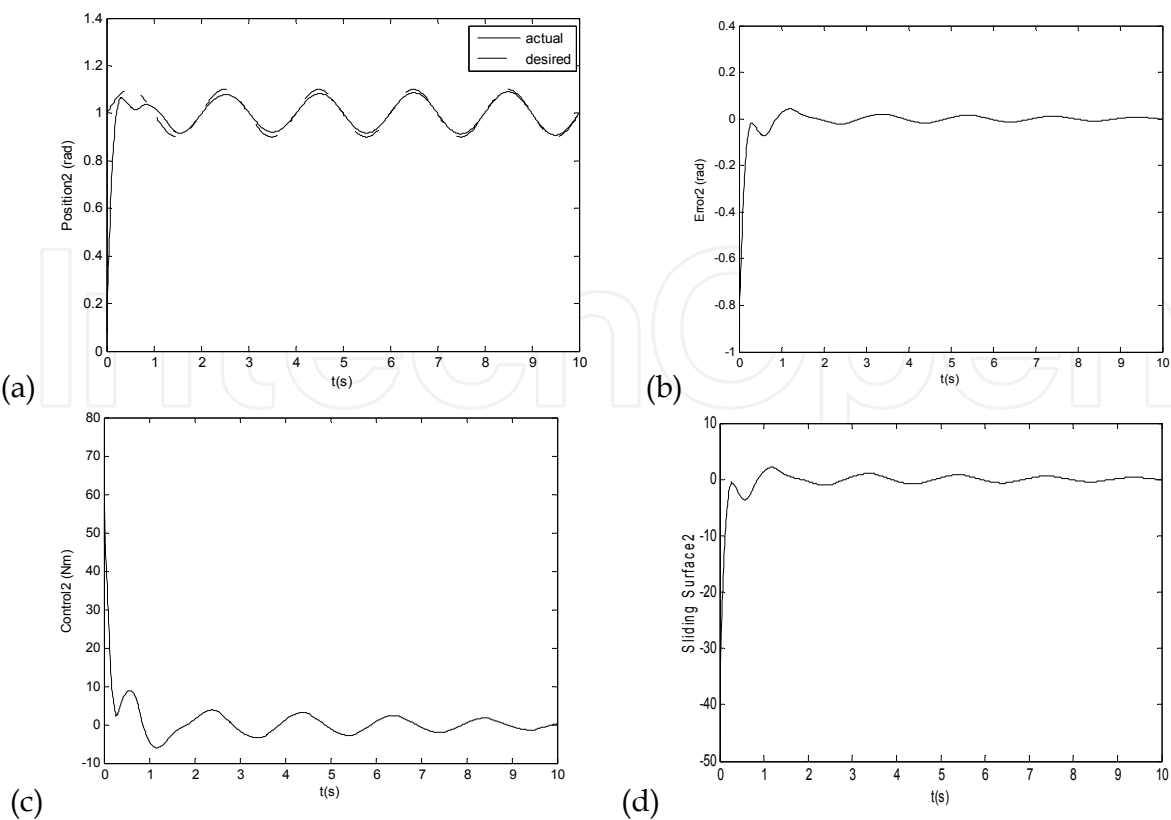
Figure 5. Simulation results for joint 2: (a) tracking response; (b) tracking error; (c) control torque; (d) sliding surface
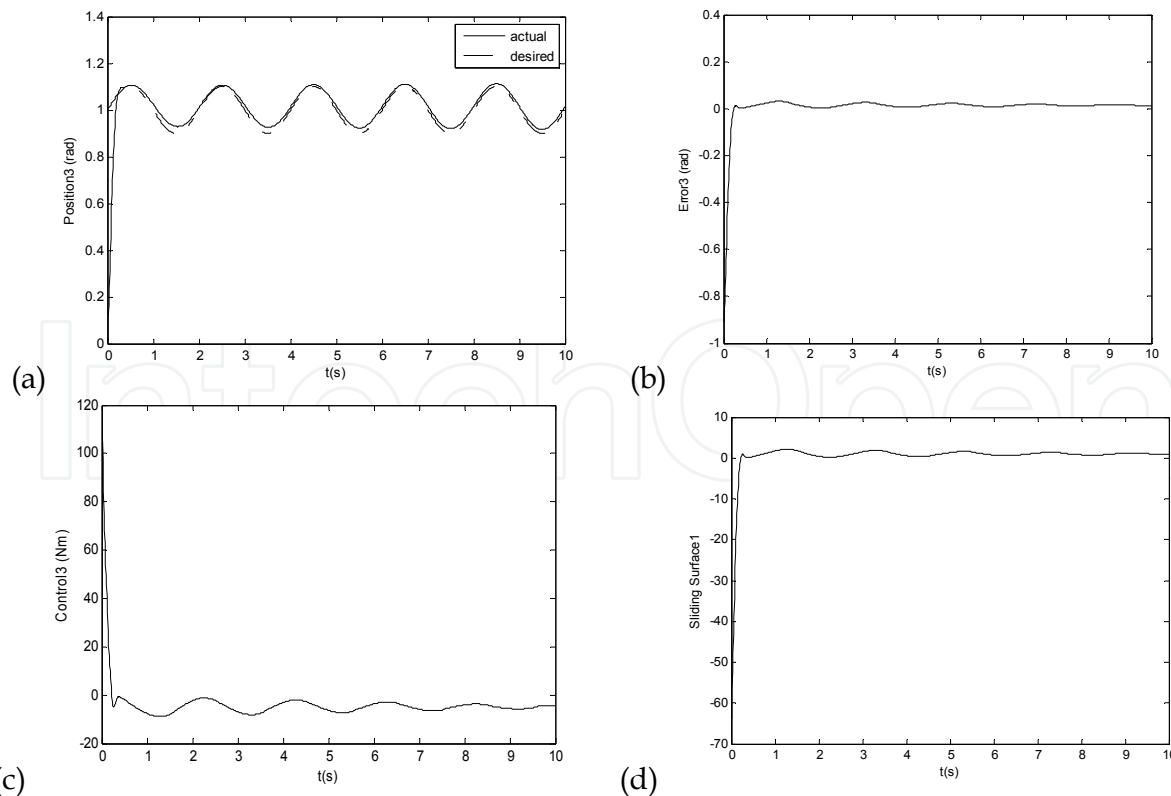

Figure 6. Simulation results for joint 3: (a) tracking response; (b) tracking error; (c) control torque; (d) sliding surface

6. References

Guo, Y., Woo, P. (2003). An Adaptive Fuzzy Sliding Mode Controller for Robotic Manipulator. *IEEE Trans. on System, Man and Cybernetics-Part A: Systems and Humans*, Vol.33, NO.2.

Utkin, V.I., (1997). Variable Structure Systems with Sliding Modes. *IEEE Trans. on Automatic Control AC*-22 212-222.

Edwards, C. and Spurgeon, K. (1998). *Sliding Mode Control*. Taylor&Fransis Ltd.

Hussain, M.A., Ho, P.Y. (2004). Adaptive Sliding Mode Control with Neural Network Based Hybrid Models. *Journal of Process Control* 14 157-176.

Lin, S. and Chen, Y. (1994). RBF-Network-Based Sliding Mode Control. *System, Man and Cybernetics, 'Humans, Information and Technology' IEEE Conf.* Vol.2.

Ertugrul, M., Kaynak, O. (1998). Neural Computation of the Equivalent Control in Sliding Mode for Robot Trajectory Control Applications. *IEEE International Conference on Robotics&Automation*.

Tsai, C., Chung, H.and Yu, F. (2004). Neuro-Sliding Mode Control with Its Applications to Seesaw Systems. *IEEE Trans. on Neural Networks*, Vol.15, NO1.

Morioka, H., Wada, K., Sabanociv, A., Jezernik, K., (1995). Neural Network Based Chattering Free Sliding Mode Control. SICE'95. *Proceeding of the 34th Annual Conf.*

Huang, S., Huang, K., Chiou, K. (2001). Development and Application of a Novel Radial Basis Function Sliding Mode Controller. *Mechatronics*. pp 313- 329.

Lin, C., Mon, Y. (2003). Hybrid adaptive fuzzy controllers with application to robotic systems. *Fuzzy Set and Systems.* 139. pp 151-165.

Chen, C. and Lin, C. (2002). A Sliding Mode Approach to Robotic Tracking Problem. *IFAC*.

Wai, R.J. and Hsich, K.Y. (2002). Tracking Control Design for Robot Manipulator via Fuzzy Neural Network. *Proceedings of the IEEE International Conference on Fuzzy Systems*. Volume 2, pp 1422–1427.

Robot Manipulators

Edited by Marco Ceccarelli

In this book we have grouped contributions in 28 chapters from several authors all around the world on the several aspects and challenges of research and applications of robots with the aim to show the recent advances and problems that still need to be considered for future improvements of robot success in worldwide frames. Each chapter addresses a specific area of modeling, design, and application of robots but with an eye to give an integrated view of what make a robot a unique modern system for many different uses and future potential applications. Main attention has been focused on design issues as thought challenging for improving capabilities and further possibilities of robots for new and old applications, as seen from today technologies and research programs. Thus, great attention has been addressed to control aspects that are strongly evolving also as function of the improvements in robot modeling, sensors, servo-power systems, and informatics. But even other aspects are considered as of fundamental challenge both in design and use of robots with improved performance and capabilities, like for example kinematic design, dynamics, vision integration.

INTECH
open science | open minds