Robot trajectory tracking with adaptive RBFNN-based fuzzy sliding mode control

Ak, G. A., Cansever, G. & Delibasi, A.

Original citation:
Ak, GA, Cansever, G & Delibasi, A 2011, 'Robot trajectory tracking with adaptive RBFNN-based fuzzy sliding mode control', Information Technology and Control, vol. 40, no. 2, pp. 151-156.
https://dx.doi.org/10.5755/j01.itc.40.2.430

DOI 10.5755/j01.itc.40.2.430
ISSN 1392-124X
ESSN 2335-884X

Publisher: Kaunas University of Technology

ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2011, Vol.40, No.2

ROBOT TRAJECTORY TRACKING WITH ADAPTIVE RBFNN-BASED FUZZY SLIDING MODE CONTROL

Ayca Gokhan Ak

Vocational School of Technical Sciences
Marmara University, Goztepe, Istanbul, Turkey
e-mail: aycaak@marmara.edu.tr

Galip Cansever, Akin Delibasi

Department of Control and Automation Engineering
Yildiz Technical University, Besiktas, Istanbul, Turkey

crossref http://dx.doi.org/10.5755/j01.itc.40.2.430

Abstract. Due to computational burden and dynamic uncertainty, the classical model-based control approaches are hard to be implemented in the multivariable robotic systems. In this paper, a model-free fuzzy sliding mode control based on neural network is proposed. In classical sliding mode controllers, system dynamics and system parameters are required to compute the equivalent control. In Radial Basis Function Neural Network (RBFNN) based fuzzy sliding mode control, a RBFNN is developed to mimic the equivalent control law in the Sliding Mode Control (SMC). The weights of the RBFNN are changed for the system state to hit the sliding surface and slide along it with an adaptive algorithm. The initial weights of the RBFNN are set to zero and then tuned online, no supervised learning procedures are needed. In the proposed method, by introducing the fuzzy concept to the sliding mode and fuzzifying the sliding surface, the chattering can be alleviated. The proposed method is implemented on industrial robot (Manutec-r15) and compared with a PID controller. Experimental studies carried out have shown that this approach is a good candidate for trajectory tracking applications of industrial robot.

Keywords: neural network; fuzzy logic; sliding mode control; robot control.

1. Introduction

As robotics systems make their way into standard practice, they have opened the door a wide spectrum of complex applications. Such applications usually demand the robots to be highly intelligent. In order to provide a high-quality control and performance in robotics, new intelligent control techniques must be developed. Hence, the pursuit of intelligent robotics systems has been a topic of much fascinating research in recent years.

On the other hand, as emerging technologies, soft computing paradigms consisting of complementary elements of fuzzy logic and neural network are viewed as the most promising methods towards intelligent robotics systems [1].

Sliding Mode Control (SMC), based on the theory of variable structure systems, has attracted a lot of research on control systems for the last two decades. This control scheme suffers from some problems, however. In order to guarantee the stability of the sliding mode system, the boundary of the uncertainty has to be estimated. A large value has to be applied to the control gain when the boundary is unknown. But, this large control gain may cause chattering on the sliding surface and therefore deteriorate the system performance [2]. Control input may be fuzzified to solve this problem.

In recent years, fuzzy logic in control design has been widely used for numerous electrical systems, robotic systems and mechanical systems. The prominent advantage of the fuzzy controller is that it can effectively control complex ill-defined systems having nonlinearities, parameter variations and disturbances [3].

The second disadvantage of the SMC is the difficulty in the calculation of the equivalent computation. A thorough knowledge of the plant dynamics is required for this purpose [4]. The equivalent control has a role similar to the inverse dynamics. An estimation technique can be used to compute the equivalent control [5]. Neural networks, especially multilayered perceptron, are the popular tool to compute the equivalent control [6], [7].

Choi and Kim designed a fuzzy-sliding mode controller by fuzzifiying the sliding surfaces in order to attenuate the chattering [3]. Derbel and Alimi realized a hybrid sliding-and-fuzzy logic controller by fuzzy logic without the computation of the equivalent control [8]. Ertugrul and Kaynak utilized two parallel neural networks to realize a neuro-SMC. In their work, equivalent control and corrective control terms of SMC are the outputs of the two layer feed-forward neural networks [7]. Abdelhameed used chattering index to tune adaptively the switching gain of the sliding mode controller in order to shorten the duration of reaching phase and to minimize chattering of the control action of SMC [9]. Hussain and Ho incorporated both prior knowledge and neural networks in the SMC with boundary layer approach [10]. Javaheri and Vossoughi designed a fuzzy controller to enhance the performance. Controller continuously optimizes the sliding mode parameters including hitting control gain, boundary layer thickness, sliding surface slope and intercept [11]. A discussion on how some "intelligence" can be incorporated in SMCs by the use of computational intelligence methodologies and an overview of the research and applications reported in the literature in this respect is presented in Kaynak et al. paper [12].

In this paper, a synergistic combination of Radial Basis Function Neural Network (RBFNN) with fuzzy sliding mode control methodology is proposed. The slope of the sliding surface is used to adjust with Fuzzy logic. The weights of the RBFNN are adjusted according to an adaptive algorithm for the purpose of controlling the system states to hit the sliding surface and then slide along it. The proposed method and PID control are implemented on an industrial robot (Manutec-r15) and the results obtained from the applications are presented.

The paper is organized as follows: In Section 2, adaptive RBFNN-based fuzzy sliding mode control design is presented. Experimental setup is explained in Section 3. Results are presented in Section 4. Section 5 concludes the paper.

2. Design of the control system

2.1. Sliding Mode Control

Variable Structure System (VSC) and SMC that is a special kind of VSC are widely used in robotics. The basic idea is to alter the system dynamics some surfaces in the state space so that the states of the system are attracted to these surfaces. During the sliding motion of the state on the surface, the system remains insensitive to parameter deviation and external disturbances.

The dynamic model of the robot is the following:

$$u = M(q)\ddot{q} + c(q,\dot{q}) + g(q)\,, \tag{1}$$

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; $u \in R^{nx1}$ is the actuator torque vector acting on the joints.

After transformations $x_1 = q$ and $x_2 = \dot{q}$ are done, robot dynamic will take the form $M\dot{x} + Cx + g = u$. The state space mathematical model of the robot is obtained as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ -M(x_1)^{-1}\left[C(x_1,x_2)x_2 + g(x_1)\right] \end{bmatrix} + \begin{bmatrix} 0 \\ M(x_1)^{-1} \end{bmatrix} u\,. \tag{2}$$

Then, the closed form of the robot dynamic is written as follows:

$$\dot{x}(t) = f(x,t) + Bu\,, \tag{3}$$

where B is the input gain matrix.

Sliding surface for system (3) can be chosen as follows:

$$s = \{x : s(x,t) = \phi(t) - s_a(x) = 0\}\,, \tag{4}$$

where $\phi(t) = Gx_d(t)$ is the time independent part of the sliding function, $s_a(x) = Gx(t)$ is the state independent part of the sliding function and x_d is the target state (reference). Sliding surface variable is in the following form

$$s_i = \left(\frac{d}{dt} + \lambda_i\right)^{k_i - 1} e_i\,, \tag{5}$$

where e is the error ($e = x_d - x$) and λ_i is a positive constant defined by designer.

In a robotic system, the purpose of the control is to provide tracking of $q(t) = q_d(t)$ trajectory by robot. In this situation, simple and general selection of sliding mode surface is $s = 0$ for $\lambda > 0$ [13].

The control should be chosen in such a way that the candidate Lyapunov function would satisfy Lyapunov criteria. The Lyapunov function (positive definite) is selected as follows:

$$V(s) = \frac{s^T s}{2}\,. \tag{6}$$

It is aimed that the derivative of the Lyapunov function would be negative definite. This can be guaranteed if one can assure that

$$\frac{dV(s)}{dt} = -s^T D sign(s)\,, \tag{7}$$

where D is a positive definite diagonal matrix.

Taking the derivative of (6), equating (7) and using system equation we obtain

$$s^T \dot{s} = -s^T D sign(s)\,. \tag{8}$$

Time derivative expression of sliding function is the following,

$$\dot{s} = \dot{\phi} - \frac{\partial s_a}{\partial x}\dot{x} = \dot{\phi} - G\left(f(x,t) + Bu(t)\right). \quad (9)$$

If expression (8) is replaced with (9), sliding mode control law can be represented as:

$$u = u_{eq} + u_c, \quad (10)$$

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:

$$u_c = K sign(s), \quad (11)$$

where K is a positive constant. The controller in (10) results with high frequency oscillations, defined as chattering. The $sign$ function is a discontinuous function:

$$sign(s) = \begin{cases} 1 & s > 0 \\ 0 & s = 0 \\ -1 & s < 0 \end{cases}. \quad (12)$$

2.2. Adaptive RBFN-based fuzzy sliding mode control

The control law for the proposed controller is as (9) form. The slope of the sliding surface (λ) is adjusted with fuzzy logic and the equivalent control u_{eq} is computed by RBFNN.

Absolute error is the input and λ is the output of the fuzzy system. The rules in the rule base are as follows:

IF e_i is vs, THEN λ_i is xxl

IF e_i is s, THEN λ_i is xl

IF e_i is m, THEN λ_i is l

IF e_i is l, THEN λ_i is m

IF e_i is xl, THEN λ_i is s

IF e_i is xxl, THEN λ_i is vs

It is chosen that both e_i and λ_i have the same kind of membership functions: vs, s, m, l, xl, xxl. The input and output membership functions are of the same structure (Figure 1).

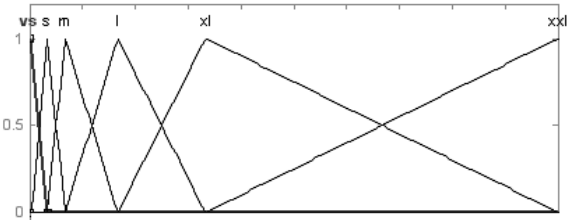


Figure 1. Input and output membership functions

From the knowledge of the fuzzy systems, λ_i can be written as;

$$\lambda_i = \theta_{ki}^T \psi_{ki}(e_i), \quad (13)$$

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

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 [2].

The architecture of RBFNN is shown in Figure 2. The network consists of three layers: an input layer, a single layer of nonlinear processing neurons, and an output layer. The output of the RBFNN is calculated according to

$$u_{eq} = \sum_{j=1}^{N} w_{ij}\varphi_j(x, c_j) = \sum_{j=1}^{N} w_{ij}\varphi_j\left(\left\|x - c_j\right\|\right) \quad i = 1, 2, ..., m, \quad (14)$$

where $x \in \Re^{nx1}$ is an input vector, $\varphi_j(.)$ is a function from $\Re^+$ (set of all positive real numbers) to $\Re^-$, $\|.\|$ is the Euclidean norm, w_{ij} are the weights in the output layer, N is the number of neurons in the hidden layer, and $c_j \in \Re^{nx1}$ are the RBF centers in the input vector space. For each neuron in the hidden layer, the Euclidean distance between its associated center and the input to the network is computed. The output of the neuron in a hidden layer is a nonlinear function of the distance. Finally, the output of the network is computed as a weighted sum of the hidden layer outputs [14]. Most preferred is the Gaussian function:

$$\varphi_j(s) = \exp\left(-\frac{\left(\left\|s - c_j\right\|^2\right)}{\sigma_j^2}\right). \quad (15)$$

Based on the Lyapunov theorem, the condition of reaching the sliding surface is $s\dot{s} < 0$. If the 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 weights of the RBFNN should be adjusted based on the reaching condition.

Weights between hidden and output layer neurons (w_j) are adjusted based on adaptive rule.

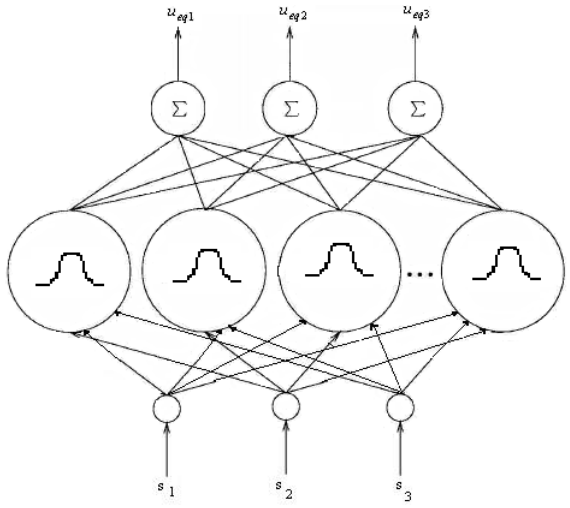


Figure 2. Radial Basis Function Neural Network (RBFNN)

The adaptive rule is derived from the steepest descent rule to minimize the value of $s\dot{s}$ with respect to w_j. Then the updated equation of the weight parameters is [15]:

$$\dot{w}_j = -\tau \frac{\partial s(t)\dot{s}(t)}{\partial w_j(t)}, \qquad (16)$$

where τ is adaptive rate parameter. Using the chain rule, (16) can be rewritten as follows:

$$\begin{aligned} \dot{w}_j &= -\tau \frac{\partial s(t)\dot{s}(t)}{\partial u_{eq}(t)} \frac{\partial u_{eq}(t)}{\partial w_j(t)} = \tau B s(t) \frac{\partial u_{eq}(t)}{\partial t} \\ &= \gamma s(t) \exp\left(-\frac{\left\| s - c_j \right\|}{\sigma_j^2} \right) = \gamma s(t) \varphi_j(s) \end{aligned}, \qquad (17)$$

where τ and system parameter are combined as a learning parameter, γ.

3. Experimental Setup

The robot manipulator used in experiments is Manutec-r15 with 6 degree of freedom (Figure 3). Robot has 3 phase, brushless synchronous motors. Robot has incremental encoders (axis 1:600, axis 2: 1200, axis 3: 600, axis 4: 360, axis 5: 320, axis 6: 500). The malfunctioned original control and driver unit of the Manutec-r15 were deactivated and a new transformer, a new driver system, a new control system was built up to try the control methods on the robot.

The block diagram of the system is shown in Figure 4. A circuit is designed to isolate the digital input-output. Simple filters are emplaced to eliminate wrong measures because of cable length that connect the systems. Experiments were executed at three axis of the robot. DAQ Card is 16 Bits, 299kS/s and has 8 Digital I/O, 2 Analog output 2, 24 bit counter/timers.

Features of the computer used on the system are Pentium 4, CPU 3.2 GHz and 512 MB RAM.

Robot was controlled with data obtained from DAQ card through Wincon software on Matlab Simulink.

Figure 3. Manutec-r15

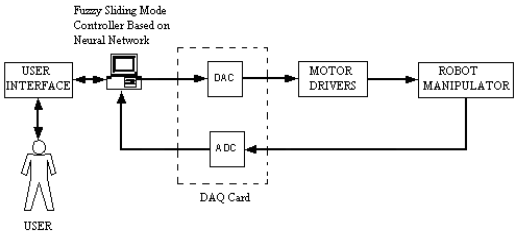


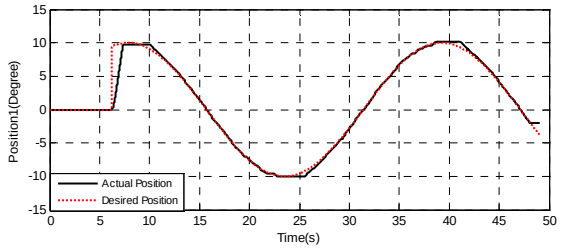
Figure 4. Block diagram of the system

4. Experimental Results

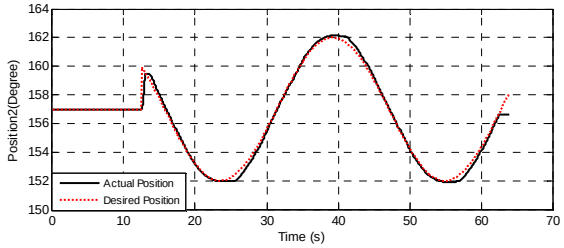
Experiments are realized on base three axes of the robot but only results of two axes are presented here. After the robot was activated, target trajectory (from the software) was given is in 6.2th second for axis 1, 12.5th second for axis 2. System was closed about 48th for axis 1 and 65th second for axis 2. Desired and actual states positions of two axes are shown in Fig 5. Desired and actual positions are closely near to each other.

The System was also tested with classical PID controller. After the brake is off, the target is switched on about 15th second for all axes. The system was closed about 65th second. Desired and actual state positions of two axes for classical PID method are shown in Figure 6.

Tracking errors of two axes for both methods are shown in Figure 7. Errors attain the maximum at zero cross positions of the sinus. Errors are smaller on adaptive RBFNN based fuzzy sliding mode control than PID controller. There is no overshoot with adaptive RBFNN based Fuzzy Sliding Mode Control. Control input torques for both methods are shown in Fig. 8. Torques are almost the same for axis 1 and 2. There is no chattering on control signals.

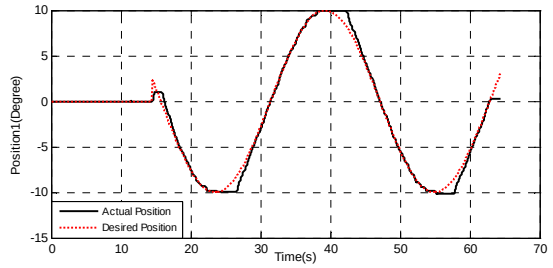


(a)

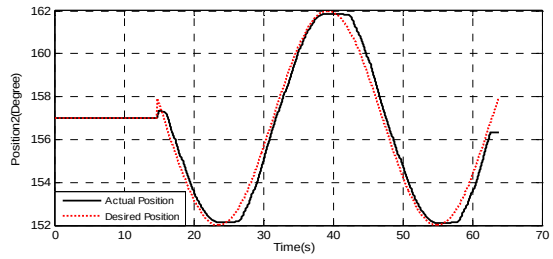


(b)

Figure 5. Desired and actual states positions of two axes for adaptive RBFNN-based fuzzy sliding mode control (a) for axis 1 and (b) for axis 2

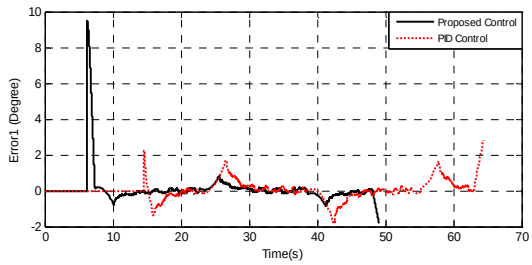


(a)

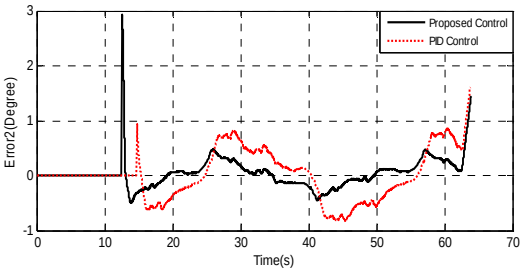


(b)

Figure 6. Desired and actual states positions of two axes for PID control (a) for axis 1 and (b) for axis 2

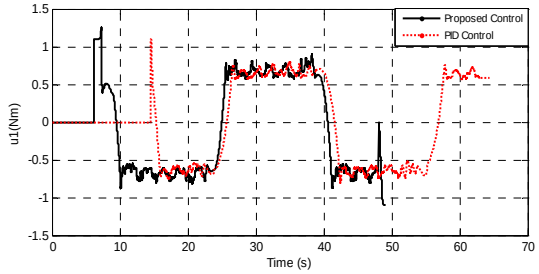


(a)

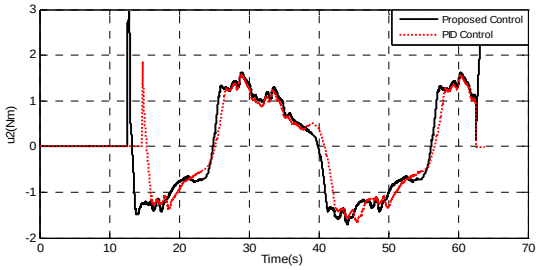


(b)

Figure 7. Position errors for two axes for adaptive RBFNN based fuzzy sliding mode control and PID control (a) for axis 1 and (b) for axis 2



(a)



(b)

Figure 8. Control Torque Inputs for two axes adaptive RBFNN based fuzzy sliding mode control and PID control (a) for axis 1 and (b) for axis 2

5. Conclusions

Adaptive RBFNN-based fuzzy sliding mode control and application results of the proposed method to the industrial robot are presented in this paper.

Using adaptive RBFNN-based fuzzy sliding mode control, whole knowledge of the system dynamics and system parameters aren't required to compute the equivalent control. An adaptive rule is utilized for on-line adjusting the weights of RBFNN, which is used to compute the equivalent control. Adaptive training algorithm was derived in the sense of Lyapunov stability analysis, so the stability of the closed-loop system can be guaranteed. Using fuzzy controller to adjust the slope of the sliding surface control gain in SMC reduces the reaching time and chattering with respect to classical SMC.

Experimental results presented in this paper indicate that tracking responses trace closely the desired

trajectory. Tracking errors of the proposed controller are smaller than those of classical control. Experimental results demonstrate that the adaptive RBFNN-based fuzzy sliding mode control is applicable control scheme for trajectory tracking applications of robotic manipulators.

Acknowledgment

This research has been supported by Yildiz Technical University Scientific Research Projects Coordination Department. Project number: 25-04-02-03.

References

[1] **D. Katic, M. Vukobratovic.** Intelligent Control of Robotic Systems. *Kluwer Academic Publishers, London*, 2003.

[2] **Y. Guo, P. Woo.** An Adaptive Fuzzy Sliding Mode Controller for Robotic Manipulators. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, Vol*.33, *No*.2, 2003, 149-159.

[3] **S.B. Choi, J. Kim.** A Fuzzy–Sliding Mode Controller for Robust Tracking of Robotic Manipulators. *Mechatronics, Vol*.7 *No*.2, 1997, 199-216.

[4] **M. Ertugrul, O. Kaynak.** Neural Computation of the Equivalent Control in Sliding Mode for Robot Trajectory Control Applications. *IEEE International Conference on Robotics&Automation*, 1998, 2042-2047.

[**5]** **V. Mkrttchian, A. Lazaryan.** Application of Neural Network in Sliding Mode Control. *IEEE International Conference on Control Applications*, 2000, 653 – 657.

[**6]** **C. Tsai, H. Chung, F. Yu.** Neuro-Sliding Mode Control with Its Applications to Seesaw Systems. IEEE Trans. on Neural Networks, Vol.15, No.1, 2004, 124-134.

[7] **M. Ertugrul and O. Kaynak.** Neuro Sliding Mode Control of Robotic Manipulators. *Mechatronics* 10, 2000, 239-263.

[8] **N. Derbel, A.M. Alimi.** Design of a Fuzzy Sliding mode Controller by Fuzzy Logic. *International Journal of Robotics and Automation, Vol*. 21, *No.* 4, 2006, 359-367.

[9] **M. Abdelhameed.** Adaptive Neural Network Based Controller for Robots. *Mechatronics*, 1999, 147-162.

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

[11] **H. Javaheri and G.R. Vossoughi.** Sliding Mode Control with Online Fuzzy Tuning: Application to a Robot Manipulator. *Proceeding of the IEEE International Conference on Mechatronics & Automation Canada*, 2005, 1357-1362.

[12] **O. Kaynak, K. Erbatur, M. Ertugrul.** The Fusion of Computationally Intelligent Methodologies and Sliding-Mode Control—A Survey. *IEEE Trans. Ind. Electronics, Vol*.48, 2001, 4- 7.

[**13]** **B. W. Bekit, J.F. Whidborne, L.D. Seneviratne.** Fuzzy sliding mode control for a robot manipulator. *Computational Intelligence in Robotics and Automation*, 1997, 320-325.

[14] **F.M. Ham, I. Kostanic.** Neurocomputing for Science &Engineering. *Mc Graw-Hill Inc*., 2001.

[15] **S. Huang, K. Huang, K. Chiou.** Development and Application of a Novel Radial Basis Function Sliding Mode Controller. *Mechatronics* 13, 2001, 313-329.

Received November 2010.