Sliding-Mode Learning of a Neuro-Adaptive Robust Control Configuration

Marco Antonio de Oliveira Alves Jr
EMBRAER
Empresa Brasileira de Aeronáutica S.A.
Av. Faria Lima 2170, CEP 12227-901, S. J. C. - Brazil
marco.alves@embraer.com.br

Euripedes G. O. Nobrega
Faculdade de Engenharia Mecânica
Universidade Estadual de Campinas
CP 6122 , 13083-970 Campinas SP - Brazil
egon@fem.unicamp.br

Abstract: ***Performance and robustness are highly desirable characteristics for any control method. But they are not found in general simultaneously in the same configuration due to its opposite nature. Recently, the compromise between robustness and performance has motivated new studies, mixing adaptive and robust methods, which are complementary in dealing with uncertainties and parameter variation. Variable structure control is a very successful adaptive method which has attracted much attention recently due to its inherent robustness, and also because it is equally applied to linear and nonlinear systems. The basic idea is to restrict the state space of a given plant through a so called sliding surface, whose dynamics is simpler than the original plant dynamics. Enforcing a state-space trajectory from the initial condition of the plant to reach the surface in finite time, once there the plant remains on the surface and its dynamics is substituted by the surface dynamics. For adequately designed surfaces, they present the invariance property, guaranteeing an intrinsic robustness because the new dynamics does not depend on the plant parameters. Associating sliding-mode algorithms to artificial neural networks, some of the recently proposed configurations may present simultaneously good performance and robustness. In this work, a new configuration is proposed, implementing a neuro-adaptive control method using the variable structure approach to adjust the neural network weights, and presenting also robustness. The main idea is to add to a regular controller signal, a second control signal generated by an artificial neural network, in order to compensate for perturbations of the plant. An adaptive online learning is adopted whose transient signals are expected do not disturb the main control loop. It is expected also that the controller performance will be maintained through a wide variation of operational conditions of the plant, independently of perturbations caused by structural and parametric variations or nonlinearities not considered in the model. The configuration is explored through two different cases, when there is an acceptable linear model and when such model is not available. Numerical simulations presenting good results justify the expectations for the configuration.***

1. INTRODUCTION

High performance and good robustness are two highly valuable characteristics essential to any control method, but in general it is necessary to achieve a compromise between them. Any plant has time-variable parameters, nonlinearities and more degree-of-freedom than it is convenient to consider in the mathematical model, implying that an uncertain model is indicated in general to prevent unexpected perturbations. This means to relieve the performance goals, which are directly related to the accuracy of the model. However, designing a good controller faces the challenging task of managing uncertainty and model accuracy. Two of the most successful control strategies are based on robust and adaptive methods, but each one alone cannot assure high performance and good robustness simultaneously. Adaptive control and robust control methods are complementary on dealing with model uncertainty. The adaptive approaches use online identification to adjust the controller to plant model errors, and are suitable to a wide range of parameter variations. Robust control methods aim to make the system insensitive to uncertainties, but they present in general a fixed-structure controller, which cannot respond efficiently to parameter variation. Recently, hybrid configurations using both approaches have been presented as capable of presenting the best characteristics of each one [1]. A new hybrid configuration using an adaptive neural network in a double loop is proposed here, intended to perform robustly to uncertain linear and nonlinear plant control.

Variable structure control (VSC), also known as sliding-mode control, is probably the most successful adaptive method, distinguished because of its remarkable simplicity. Despite being originally studied in the decade of 1950, it did not receive wide acceptance at the beginning, due to the lack of more systematical design procedure and the presence of chattering in the control signals [2]. But its inherent robustness has attracted much attention in the last two decades [3][4]. The method is based on the design of a restraining sliding-mode surface, reducing the order of the plant dynamics and indeed substituting it, but causing chattering to the control input due to the switching to remain at the predefined state-space subspace. Furthermore, a conventional VSC needs the assumption of known uncertainty bounds [5]. Nevertheless, the switching mode presents invariance properties to the plant model uncertainties, making VSC a good option to control uncertain nonlinear systems.

Considering that artificial neural networks (ANN) can approximate nonlinear mappings with reasonable accuracy, several authors adopted the design philosophy of approximating the nonlinearities with a neural network, compensating the approximation errors and external disturbances with a robust controller [6] [7] [8] [9]. The

ANN modules are used to learn nonlinear functions representing the plant direct or inverse dynamics, or some other desired specific function. The ANN may be trained offline, presenting a fixed input-output relation, as in [10]. When the training is otherwise explored online, it is appropriately called an adaptive artificial neural network (AANN). Barambones and Etxebarria [9] present a configuration to control a generic robot, implementing an AANN-based feedback linearization and a robust sliding-mode controller compensating for the neural approximation error.

Robotic manipulators are hard to control nonlinear systems, with time-varying inertia and gravitational loads, and joint friction model uncertainties. These characteristics make robot control a good candidate to test control strategies. Jung and Hsia [7] propose a robust impedance control scheme that uses an ANN to cancel out the uncertainties of the robot dynamic model. Sun et al [6] present an approach for robot trajectory tracking, dropping the common assumption of the known bound on the ANN reconstruction error. Following this trend, a planar two-link robotic arm is here adopted to numerically simulate and test the proposed configuration, using two different combinations of controller design methods.

The following sections present the proposed configuration, the sliding-mode adaptive control method for the neural network controller, the adopted residue generator techniques, the robot modeling, the numerical results obtained through simulations and the conclusions. A stability study is been formulated and will be presented in a future work.

2. PROPOSED CONFIGURATION

The proposed configuration is represented through the block diagram depicted in Fig. 1. The adopted control methodology is based on a robust configuration proposed by Zhou [1]. However, a neuro-adaptive control concept is here introduced. The methodology description may be followed based on the block diagram in Fig. 1.

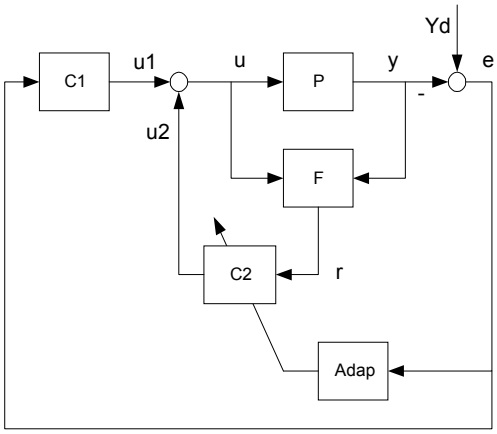

Figure 1 – The proposed controller configuration

The configuration is based on using two different controllers simultaneously. The first controller, here called the main or direct controller and represented in the block diagram as C_1, may be designed based on any control method, and produces the main control signal vector $u_1(t)$. The second one is a neural-network-based adaptive controller, producing the control signal vector $u_2(t)$, which adds to the signal $u_1(t)$ to produce the control signal $u(t)$. The control signal $u_2(t)$ is a complementing signal to compensate for errors between an assumed plant model and the actual plant. The block F is a residue generator, whose output is the residual vector $r(t)$. The mathematical model of the plant is used to design the residue generator, and any difference in the estimated output signals obtained based on this model and the actual measured signals is reflected at the output of module F as a residue. If there is a match between the implicit model and the experimental online results, the residues are very small and the direct controller is sufficient to govern the plant. The residue vector is the input to the neural controller C_2 and if there is a significant residue, the respectively generated control signal will try to correct it. This difference may be due to any modeling error, so uncertainties and time-variability of the parameters are automatically corrected. So far, the configuration is very similar to the scheme proposed by Zhou. However there are two important differences: using an adaptive neural network as the second controller, and also the adopted sliding-mode weight adjustment (represented in Fig. 1 as the module named *Adap*). The objective here is to guarantee the overall performance through transients, commonly found when there is perturbation on the neural network weights, which could occur in a case of a conventional on-line adaptation, due to a sudden parameter variation. Using a sliding-mode learning algorithm for the weight adjustment avoids these abrupt transients. The difference between the response of the plant y and the desired response y_d is the error vector e. This error signal vector is used on-line to adjust the neural network weights, through a designed sliding-mode law.

Two cases are considered to explore the configuration, applied to a two-link robotic arm, a nonlinear plant with variable parameters. The first case is a model-based design, except for the neural controller, and the second case corresponds to a totally unknown model. For the first case the nonlinear model was linearized around a central point of the robot workspace, yielding a linear time-invariant state-space model. A PID direct controller is adopted and a residue generator is then designed, both modules using the linearized model of the plant. For the second case, neural-network-based modules are adopted for both controllers and also for the residual generator, trained with input/output data obtained from the plant.

For the first case a PID control law was used, according to:

$$v = \ddot{y}_d + K_d \dot{e} + K_p e + K_i \int e dt \qquad (1)$$

where the tracking error is defined as

$$e = y_d(t) - y(t) \tag{2}$$

and the nonlinear state-space model of the robot is

$$\dot{x}(t) = f(x) + g(x)u_1(t)\,. \tag{3}$$

Adopting

$$u_1(t) = \frac{1}{\hat{g}(x)}\left(-\hat{f}(x) + v(t)\right), \tag{4}$$

based on the linearization of state-space model ($\hat{f}(x)$, $\hat{g}(x)$), the control law is calculated as

$$u_1(t) = \frac{1}{\hat{g}(x)}\left(-\hat{f}(x) + \ddot{y}_d + K_d\dot{e} + K_p e + K_i\int e dt\right) \tag{5}$$

Notice that using the nonlinear functions *f(x)* and *g(x)*, this approach would result in a PID feedback linearization direct controller.

3. NEURAL ADAPTIVE CONTROL

For the proposed configurations, the second controller is always based on a neural network adaptive control approach. For this controller, a configuration of two neuron layers is used, with a linear layer at the output and a nonlinear as the input layer. The ANN output is found according to

$$u_2(t) = W_2\sigma_1\left[W_1\phi(t)\right], \tag{6}$$

where W_1 and W_2 are respectively the weights of the input and output layers, $\phi(t)$ is the neural network input vector, and σ_1 is the activation nonlinear function of the input neuron layer. Training of the network is accomplished on-line, based on a sliding-mode surface [6] and adopting the error defined as the difference between the desired plant response and the measured one. The weight adaptation is according to the following expression:

$$W_1(t) = W_1(t-1) + \eta\left(W_1(t-1) - W_1(t-2)\right) + \frac{\alpha\,|e_1|\,X_s\,\mathrm{sign}s_1}{\varepsilon + X_s^T X_s}, \tag{7}$$

where W_1 is the weight for the output layer, X_s is the output of the hidden layer, e_1 is the performance error defined by (2). The error represented by the product $sign(s_1)|e_1|$ is then backpropagated to the hidden layer [10]. The parameters α and η are respectively the learning rate and the momentum, and ε is a small valued parameter to avoid a singularity problem. The sliding surface is imposed by the variable s_1, defined below, and $sign(s_1)$ is the signum function. This formulation guarantees convergence if the system is persistently excited [11], a condition fulfilled due to the on-line continuous training of the ANN. The sliding surface is defined according to

$$s_1 = \dot{e}_1 + \lambda e_1\,, \tag{8}$$

where e_1 is the performance error and $\lambda > 0$ is a scalar parameter. A complete survey of sliding-mode control can be found in [4].

4. RESIDUAL GENERATORS

Two different approaches are used to implement the residual generator, a model-based observer and a neural network observer. For the first case, an output observer formulation commonly used for model-based fault detection, is adopted. It follows

$$\begin{aligned} r &= \hat{N}u - \hat{M}y \\ \hat{N} &= D + C\left(sI - A + LC\right)^{-1}\left(B - LD\right), \\ \hat{M} &= I - \left(sI - A + LC\right)^{-1}L, \end{aligned} \tag{9}$$

where L is the designed observer gain and $\hat{N}$ and $\hat{M}$ are rational function matrices obtained by left-factorization of the robot transfer matrix based on the linear time-invariant state-space model *[A B C D]*. Details of this residual generator formulation may be found in [12]. Using this approach, the residues result from the differences between this nominal model and the measured plant outputs. Because the plant used for the simulations presented in section 6 is nonlinear, the residues are due to the nonlinearity errors.

For the second case, the neural residual generator is based on an ANN output observer, trained off-line as a nonlinear identification of the plant. In other words, the output of the neural network aims to estimate the output of the plant, and may be trained with experimental data, or as in this paper, simulated output data obtained from the nonlinear model of the plant. The main advantage of this ANN approach is that it is not necessary to know the plant model to design the residual generator. As the neural observer is based on fixed-value parameters of the plant, the observer will be sensitive to any plant perturbation. An exhaustive training covering the robot workspace is indicated. Any variation of the behavior of the plant, due e.g. to parameter drift, different payload, or even an incipient fault, will influence the residual signals.

5. ROBOT MODELLING

The dynamic model for a robotic manipulator with *n* links can be formulated as:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = u\,, \tag{10}$$

where $q, \dot{q}, \ddot{q}$ are the joint position, velocity and acceleration vectors, M is the inertia matrix, C represents the Coriolis and centripetal forces, G is the gravitational terms and $u(t)$ is the vector of joint torques. Considering that the inertia matrix is always symmetrical and positive definite, (10) may be converted to state space and written as:

$$\dot{X}_1 = X_2$$
$$\dot{X}_2 = f(X_1, X_2) + g(X_1, X_2)u \quad (11)$$

where $X_1 = [x_1 \ \cdots \ x_n]^T$, is the joint angular displacement vector, $X_2 = [\dot{x}_1 \ \cdots \ \dot{x}_n]^T$, is the velocity vector, and $f(X_1, X_2) = -M^{-1}G$ and $g(X_1, X_2) = M^{-1}$. For a two link planar robotic arm, the matrices M and G are given in (12) and (13) (see [13]).

$$M = \begin{bmatrix} (b_1 + b_2)a_1^2 + b_2a_2^2 + 2b_2a_1a_2\cos(x_2) & b_2a_2^2 + b_2a_1\cos(x_2) \\ b_2a_2^2 + b_2a_1\cos(x_2) & b_2a_2^2 \end{bmatrix} \quad (12)$$

$$G = \begin{bmatrix} -b_2a_1a_2(2x_3x_4 + x_4^2)\sin(x_2) + 9.8(b_1 + b_2)a_1\cos(x_1) + 9.8b_2a_2\cos(x_1 + x_2) \\ b_2a_1a_2x_1^2\sin(x_2) + 9.8b_2a_2\cos(x_1 + x_2) \end{bmatrix} \quad (13)$$

The respective geometric parameters are adopted for simplicity as $a_1 = a_2 = 1$, $b_1 = b_2 = 1$.

6. NUMERICAL SIMULATIONS

Aiming a better understanding of the role of the second controller and to analyze the configuration performance, two simulation cases are here presented. The case where C_1 is the PID controller is presented together with the results where C_1 is an off-line trained neural controller, for three different pairs of position of the joints. In both cases C_2 is the previous described neuro-adaptive controller. The desired reference signal and results for the PID controller without C_2 are presented also, for comparison.

Three different pair of positions are used as the initial condition for a square wave response analysis, respectively: position 1 corresponds to angles of 45º/45º for the first and second joint, with a desired square wave with amplitude ±25º; position 2 corresponds to the joint positions of 10º/20º and a desired square wave amplitude of ±5º; and the position 3 corresponds to the joint angles 70º/45º and desired square wave amplitude of ±25º. The first position pair corresponds to the point where the model was linearized. These values were chosen to present a general view of the behavior of the controller throughout the robot workspace.

6.1 Simulations Results

Figures 2, 3 and 4 shows the position results for the three different initial positions of the joints.

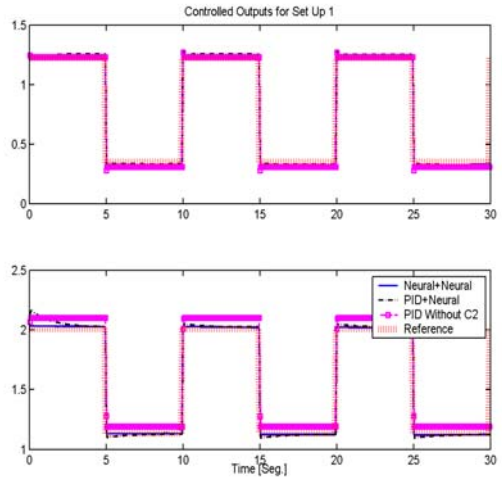

Figure 2) Results for the 45º/45º position

Fig. 2 shows the performance for position 1. The tracking performance is very similar for the three controllers. But it is possible to note that the PID without C_2 presents the worst performance, mainly for the second joint, where it can be seen a small stationary error. Notice that this should be the best PID performance, because it corresponds to the point of linearization. It is possible to see also that the complete neural controller presents the best performance.

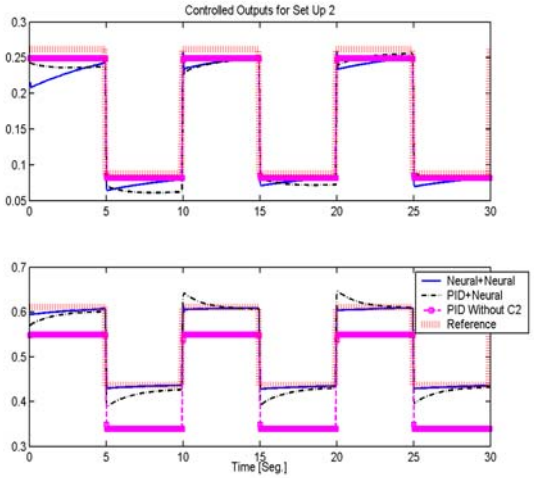

Figure 3) Results for the 10º/20º position

Fig. 3 shows the performance for the position 2. The tracking performance is not similar anymore for the three controllers. The PID without C_2 now presents a significant stationary error, bigger for the second joint. The PID with neural controller shows the better performance for the first joint and the complete neural controller shows the best result for the second joint. It is clear the effect of the weight adjustment for the neural controller trying to correct the stationary error of the PID controller. For the first joint, the complete neural controller seems to be

improving with time, also due to the adaptive training of the neural network.

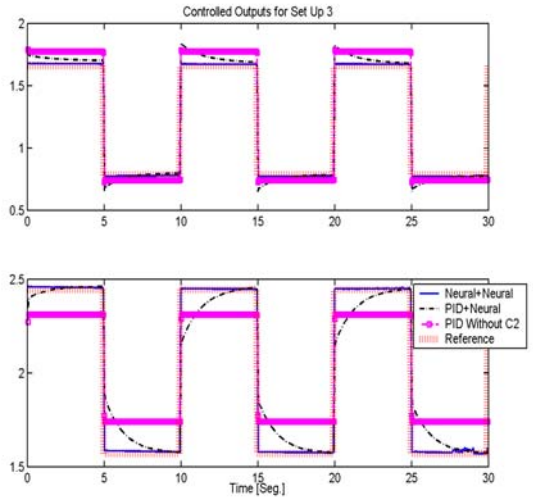

Figure 4) Results for the 70º/45º position

Fig. 4 shows the performance for the position 3. The stationary error is present again for the PID only controller, which is slowly corrected by the inclusion of the neural adaptive controller. A tuning of the adjustment parameters may improve this result. The complete neural configuration presents here an impressive result, tracking the square wave very well.

Figures 5, 6 and 7 present the corresponding control signals for each of the three positions. There are four panels for each figure. The two upper panels correspond to the C_1 controllers, for the first and second joint respectively, and the two lower panels correspond to the C_2 controllers also for the two joints.

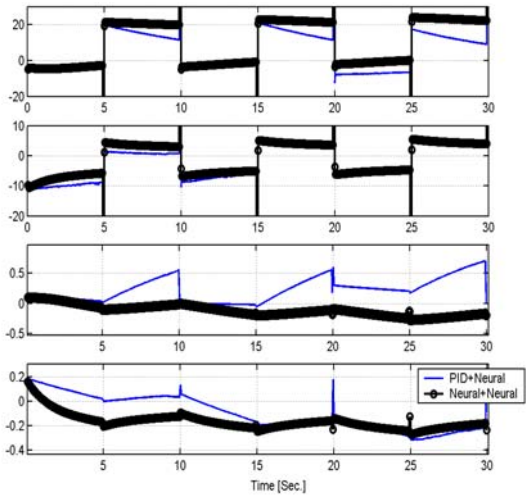

Figure 5) Control signals for the 45º/45º Position

Fig. 5 presents the control signals for the position 1. The control effort for the second controller is one order of magnitude lower then the effort for the direct controller, which is sensible because it is a complementing signal. But it is interesting to notice how a small signal can make a difference in the performance of the system. It is visible also that the PID with the neural configuration has an increasing contribution for the second controller, while the complete neural configuration does not change significantly.

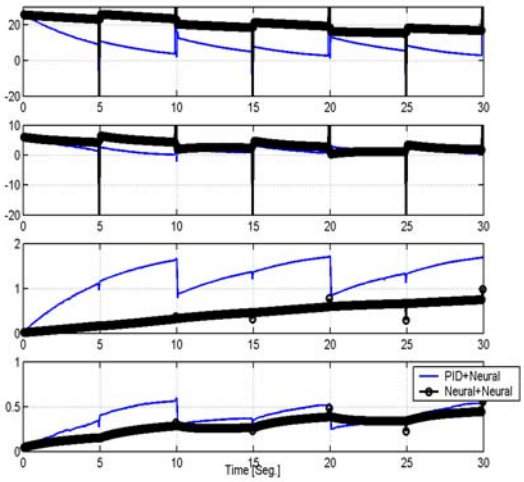

Figure 6) Control signals for the 10º/20º position

Fig. 6 presents the control signals for the position 2. The signals behavior are different from the previous case. However the control effort for the second controller is again one order of magnitude lower then the effort for the direct controller. But now for both cases the second controller is slowly increasing the signal level.

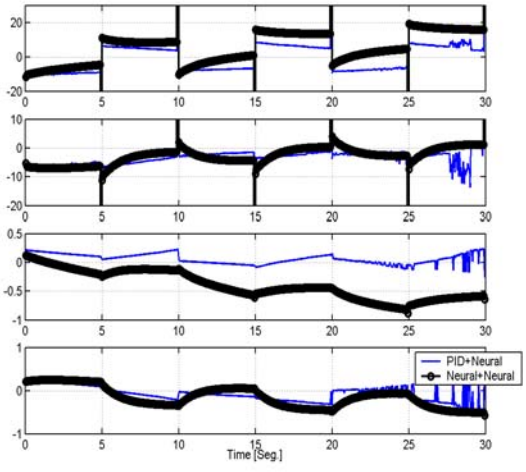

Figure 7) Control signals for the 70º/45º position

Fig. 7 presents the control signals for the position 3. The signals are again different from the previous cases, but

maintaining essentially the same relative level proportion. The main difference is that now it is possible to notice a chattering in the signals of both configurations, not seen in the previous results.

7. CONCLUSIONS

The proposed configuration simulations presented good results for the studied cases. Including the neuro-adaptive controller associated to a linear PID controller and residual generator, improved the results comparing to the operation of the PID controller only. An adequate compensation for modeling errors is provided by the adaptive neural controller, implying a better reference tracking of the two robot links. A fine-tuning of the neural network learning parameters may be necessary to achieve the best possible results. For the completely neural configuration the results are similar, with the adaptive neural controller improving the performance of the direct neural controller, and also accomplishing a better performance than in the first case. The robustness of the configuration is apparent from the overview of the simulations, but more specific analysis and tests must be conducted to confirm this, and published in the future. Further studies are necessary also to investigate the effect of the switching on the residues and control signals, and to confirm theoretically the stability of the configuration.

Two different approaches may be considered for the application of the configuration, based on the availability of a reliable mathematical model of the plant. If such a model is available, a simple controller may be designed and the second controller easily implemented without previous training of the neural networks. In either case, the designer may choose between a linear output observer and a neural observer, exercising engineering judgement. If the model is not available or if it is too complex to be used, a complete neural network configuration may be adopted, with the neural output observer and the direct controller fixed and trained off-line, and the second controller being adaptively adjusted online through the sliding-mode learning law.

8. REFERENCES

[1] Zhou, K. and Ren, Z., A New Controller Architecture for High Performance, Robust, and Fault-Tolerant Control, IEEE Transaction on Automatic Control, Vol. 46, No. 10, October 2001, pp. 1613 - 1618

[2] Gao, W. and Hung, J. C., Variable Structure Control of Nonlinear Systems: A New Approach, IEEE Transaction on Industrial Eletronics, Vol. 40, No. 1, February, 1993, pp. 45 – 56

[3] DeCarlo, R. A., Zak, S. H. and Matthews, G. P., Variable structure control of nonlinear multivariable systems: a tutorial, Proceedings of the IEEE, V. 76, no 3, 1988, pp. 212-232

[4] Hung, Y. J. and Hung, J. C., Variable structure control: a survey, IEEE Trans. on Industrial Electronics, V. 40, no 1, 1993, pp. 2-22

[5] Yu, H. and Lloyd, S., Variable structure adaptive control of robot manipulators, IEE Proc.-Control Theory Appl. V. 144, no 2, 1997, p. 167-176

[6] Sun, F. C., Sun, Z. Q., Zhang, R. J. and Chen, Y. B., Neural adaptive tracking controller for robot manipulators with unknown dynamics, IEE Proc.-Control Theory Appl. V. 147, no 3, 2000, p. 366-370

[7] Jung, S. and Hsia, T. C., Robust neural force control scheme under uncertainties in robot dynamics and unknown environments, IEEE Trans. On Industrial Electronics, V. 47, no 2, 2000, pp. 403-412

[8] Topalov, A. V. and Kaynak, O., Online learning in adaptive neurocontrol schemes with sliding mode algorithm, IEEE Trans. On Systems, Man and Cybernetics, V. 31, no 3, 2001, pp. 445-450

[9] Barambones, O. and Etxebarria, V., Robust neural control for robotic manipulator, Automatica, V 38, 2002, pp. 235-242

[10] Haykin, Simon, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1998, 2nd Edition.

[11] Khanmohammadi, S; et. al., Modified Adaptive Discrete Control System Containing Neural Estimator and Neural Controller, Artificial Intelligence in Engineering, 14 , 2000, 31-38.

[12] Frank, P. M. and X. Ding, Survey of Robust Residual Generation and Evaluation Methods in Observer-Based Fault Detection Systems, Elsevier Science Ltd, 1997.

[13] Lewis, F. L.; Jagannathan and Yesildirek, Neural Network Control of Robot Manipulators and Nonlinear Systems, Taylor & Francis, 1999.