ICAR '97
Monterey, CA, July 7-9, 1997

NEURO-SLIDING MODE CONTROL OF ROBOTIC MANIPULATORS

Melikşah Ertuğrul * and **Okyay Kaynak** +

* TÜBİTAK Marmara Research Center, Robotics and Automation Group, Gebze, TURKEY
+ UNESCO Chair on Mechatronics and Mechatronics R & A Center, Boğaziçi University, 80815 Istanbul, TURKEY

ABSTRACT: In this paper, a synergistic combination of neural networks with sliding mode control (SMC) is proposed. As a result, the chattering is eliminated and error performance of SMC is improved. In such an approach, two parallel NNs are proposed to realize SMC. The equivalent control and the corrective term of SMC are the outputs of the NNs. Gradient Decent method is used for the weight adaptation. This novel approach is applied to control of a scara type robot manipulator and simulation results are given.

Keywords: Neural Networks, Sliding Mode Control.

1. INTRODUCTION

Variable Structure Controllers with Sliding Mode Control (SMC) was first proposed in early 1950's. After seventies, SMC has become more popular and nowadays it enjoys a wide variety of application areas. The main reason of this popularity is the attractive superior properties of SMC, such as good control performance even in the case of nonlinear systems, applicability to MIMO systems, design criteria for discrete time systems, etc. The best property of the SMC is its robustness. Loosely speaking, a system with a SMC is insensitive to parameter changes or external disturbances [1].

The classical SMC suffers mainly from two disadvantages. The first one is the high frequency oscillations of the controller output, termed "chattering". The second is that a complete knowledge of the plant dynamics is needed in the calculation of the equivalent control [2]. In literature, there are some suggestions to solve these problems. The most popular technique for the elimination of chattering technique is the use of a saturation function [3-4]. On the other hand, one way of avoiding the computational burden involved in the calculation of the equivalent control is the use of an estimation technique as suggested in [5].

SMC design comes up with two parts: the equivalent control and the corrective term. In this paper, two parallel NNs are proposed to realize the both parts. The equivalent control has a role similar to the inverse dynamics. When the system states are on the sliding surface, the equivalent control is enough to keep the system there and the corrective term is zero. The corrective term is necessary when the system states deviate from the surface. A two layer feed-forward NN is designed to compute the equivalent control and weights are adapted to minimize the square of the corrective term.

The chattering comes up as a result of the corrective term. In this paper, a neural network gain adaptation scheme is proposed which directly results in chattering-free control action for the corrective term. The gains of SMC are accepted as the weights of the NN and weights are adapted to minimize a cost function. One of the main problems of NN design is how to select the layers, number of neurons for each layer and the connections between layers. In the structure which is proposed in this paper, this problem is not met since the network topology of the NN is well determined from SMC design. NN has a two layer structure; a hidden and an output layer. The number of neurons for each layer and connections between neurons are directly established through the design of the SMC.

The paper concludes with the presentation of some simulation results obtained for the control of a direct drive scara type robot.

2. VARIABLE STRUCTURE SYSTEMS

In the application of Variable Structure System theory to the control of nonlinear processes it is argued that one only needs to drive the error to a "switching" or "sliding" surface, after which the system is in "sliding mode" and will not be affected by any modeling uncertainties and/or disturbances. Intuitively, VSS with a sliding mode is based on the argument that the control of 1st-order systems (i.e., systems described by 1st-order differential equations) is much easier, even when they are nonlinear or uncertain, than the control general nth-order systems [6].

2.1 The System (Plant)

Consider a nonlinear, non-autonomous multi-input multi-output system of the form,

$$x_i^{(k_i)} = f_i(X) + \sum_{j=1}^{m} b_{ij} u_j \qquad i=1..m \tag{1}$$

where $x_i^{(k_i)}$ means the $k_i{}^{th}$ derivative of x_i. Also, the vector U of components u_j is the control input vector and the state X is composed of the x_i's and their first (k_i-1) derivatives. Such systems are called square systems since they have as many control inputs as outputs to be controlled x_i [6]. The system can be written in a more compact form as letting

$$X = \begin{bmatrix} x_1 & \dot{x}_1 & \cdots & x_1^{k_1} & \cdots & x_m & \dot{x}_m & \cdots & x_m^{k_m} \end{bmatrix}^T \quad (2)$$

$$U = \begin{bmatrix} u_1 & \cdots & u_m \end{bmatrix}^T \quad (3)$$

The system equation becomes,

$$\dot{X}(t) = F(X) + BU(t) \quad (4)$$

where X is (nx1) and B is (nxm) input gain matrix.

2.2 Sliding Surface

For the systems given in (4), generally, sliding surface, S (mx1) is selected [5] as given below,

$$S(X,t) = G(X^r(t) - X(t)) = GE = \phi(t) - S_a(X) \quad (5)$$

where, $$\phi(t) = G\,X^r(t), \quad S_a(X) = G\,X(t) \quad (6)$$

the time and state dependent parts. Also X^r represents the reference state vector and G is (mxn) slope matrix of the sliding surface. Generally, the G matrix is selected such that the sliding surface function becomes,

$$S_i = \left(\frac{d}{dt} + \lambda_i\right)^{k_i - 1} e_i \quad (7)$$

where e_i is the error for x_i ($e_i = x_i^r - x_i$). λ_i's are selected as positive constants. Therefore e_i goes to zero when S_i equals to zero.

The aim in SMC is to force the system states to the sliding surface. Once the states are on the sliding surface, the system errors converge to zero with an error dynamics dictated by the design of the matrix G.

2.3 Sliding Mode Controller Design

The method described in this section is based on the selection of a Lyapunov function [5]. The control should be chosen such that the candidate Lyapunov function satisfies Lyapunov stability criteria.

Firstly, a Lyapunov function is selected as given below,

$$V(S) = \frac{S^T S}{2} \quad (8)$$

It can be noted that this function is positive definite. ($V(S=0)=0$ and $V(S)>0 \;\forall S \neq 0$)

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

$$\frac{dV(S)}{dt} = -S^T D\, sign(S) \quad (9)$$

D is (mxm) positive definite diagonal gain matrix. *sign(S)* means signum function is applied to each element of S, i.e.

$$sign(S) = \begin{bmatrix} sign(S_1) & \ldots & sign(S_m) \end{bmatrix}^T \quad (10)$$

Taking the derivative of (8), and equating this to (9), one will obtain the following equation,

$$S^T \frac{dS}{dt} = -S^T D\, sign(S) \quad (11)$$

By taking the time derivative of (5) and using the plant equation,

$$\frac{dS}{dt} = \frac{d\phi}{dt} - \frac{\partial S_a}{\partial X}\frac{dX}{dt} = \frac{d\phi}{dt} - G(F(X) + BU) \quad (12)$$

is obtained. By putting (12) into (11), the control input signal can be obtained as,

$$U(t) = U_{eq}(t) + (GB)^{-1} D sign(S) = U_{eq}(t) + \Delta U(t) \quad (13)$$

where $U_{eq}(t)$ is the equivalent control and written as,

$$U_{eq}(t) = -(GB)^{-1}\left(GF(X) - \frac{d\phi(t)}{dt}\right) \quad (14)$$

The solution for chattering elimination is making the second term of the controller of (13) proportional to S or a smooth function of S around zero. A good example is the use of a saturation function instead of the sign function.

3. NEURO-SLIDING MODE CONTROL

A basic Artificial Neural Network consists of "neurons", "weights" and "activation functions". The weights are adapted such that the overall system performance is improved. Nowadays, a variety of Neuro-controllers are successfully used for control applications [7].

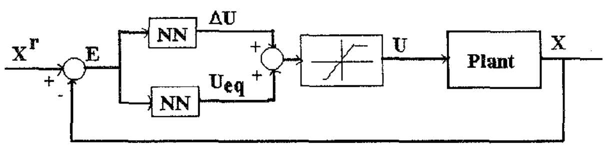

Figure 1. Overall control system.

3.1 NN for the Corrective Control Term Computation

In this NN structure, the gains of SMC are taken as the weights of NN. The aim is the use of NN weight adaptation techniques to adapt the gains of SMC. The structure of NN is thus well determined from the SMC design. NN has a two layer structure; a hidden and an output layer. The number of neurons for each layer and connections between neurons are obviously established.

The inputs for the neural network are selected as the state errors. In the hidden layer, the number of neurons is equal to the number of sliding functions (i.e. the number of plant inputs). Each input is not connected to all neurons at the hidden layer, instead, it is connected to only one neuron at the hidden layer so that an appropriate sliding surface is formed at that neuron. The outputs of the hidden layer are passed through an activation function too, such as the sign function or sign like continuous functions (e.g. shifted sigmoid).

The hidden layer is fully connected to the output layer at which the number of neurons is equal to the number of the plant inputs. There is no activation function for this layer. The output of the neurons is the corrective term to be added to equivalent control as shown in (13).

To make the Controller in (13) to fit better into an NN form, one can also write it as,

$$U(t) = \tilde{U}_{eq}(t) + \Delta U(t) \tag{15}$$

where,

$$\Delta U(t) = K\, h(S) \tag{16}$$

and also $K = (GB)^{-1}D$ and $h(S) = sign(S)$.

A simple NN structure is given in Fig. 2 for a two degrees of freedom robot manipulator. The overall system with NN is shown in Fig. 1. The controller output is passed through a limiter because the plant naturally has limits in its inputs.

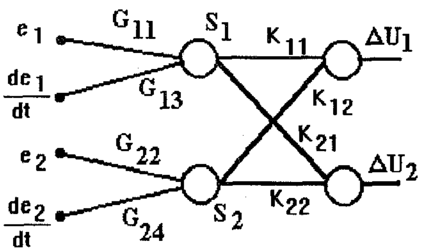

Figure 2. NN Structure for Δu

An adaptation scheme to minimize the control effort and sliding function is proposed using the well-known MIT rule (Gradient Decent). The criterion (cost) which is to be minimized is chosen as,

$$J = \frac{1}{2}\left(S^T S + U^T U\right) \tag{17}$$

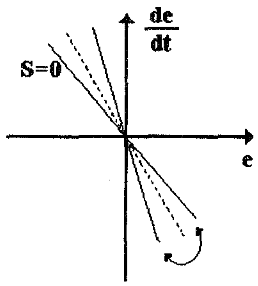

Figure 3 The Effect of G-Adaptation

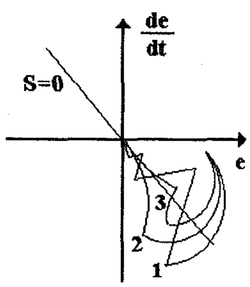

Figure 4 K-Adaptation

To make J small it is reasonable to change the parameters (weights) in the direction of the negative gradient of it i.e.,

$$\frac{dK_{ji}}{dt} = -\gamma \frac{\partial J}{\partial K_{ji}} \quad \text{and} \quad \frac{dG_{ji}}{dt} = -\gamma_2 \frac{\partial J}{\partial G_{ji}} \tag{18}$$

3.1.1 Weight Adaptation for the Second Layer

Weight Adaptation for second layer also means the adaptation of K. The effect of K-adaptation is presented in Fig. 4. The gradient decent as in (18) for K can be derived as,

$$\frac{dK_{ji}}{dt} = -\gamma_1 \frac{\partial J}{\partial S_j}\frac{\partial S_j}{\partial K_{ji}} - \gamma_1 \frac{\partial J}{\partial U_j}\frac{\partial U_j}{\partial K_{ji}} \tag{19}$$

Where $j=1..m$ and $i=1..m$.

Using (17), the partial derivatives of the cost function can be calculated as,

$$\frac{\partial J}{\partial S_j} = S_j \qquad \text{and} \qquad \frac{\partial J}{\partial U_j} = U_j \tag{20}$$

Using (15) and (16),

$$\frac{\partial U_j}{\partial K_{ji}} = \frac{\partial U_j}{\partial \Delta U_j}\frac{\partial \Delta U_j}{\partial K_{ji}} = 1.h(S_i) \tag{21}$$

The sliding function in (5) can be rewritten with using (15) and the integral of (4) as,

$$S = G\left(X^r - X\right) = GX^r - G\int\left(F(X) + B\left(U_{eq} + Kh(S)\right)\right) \tag{22}$$

Taking the partial derivative of (22) and assigning the constants to γ which are obtained by multiplication of elements of G and B,

$$\frac{\partial S_j}{\partial K_{ji}} = -\gamma \int h(S_i(\xi))d\xi \tag{23}$$

The last form of K adaptation is obtained as,

$$\frac{dK_{ji}}{dt} = \gamma^3_{ji} S_j \int h(S_i(\xi))d\xi \; - \; \gamma^1_{ji} U_j h(S_i) \tag{24}$$

3.1.2 Weight Adaptation for the First Layer

Weight Adaptation for the first layer also means the adaptation of G. The effect of G-adaptation is presented in Fig. 3. Similar to the derivation of (19), the gradient descent for G can be derived as,

$$\frac{dG_{ji}}{dt} = -\gamma_2 \frac{\partial J}{\partial S_j}\frac{\partial S_j}{\partial G_{ji}} - \gamma_2 \sum_{k=1}^{m} \frac{\partial J}{\partial U_k}\frac{\partial U_k}{\partial G_{ji}} \qquad (25)$$

Where $j=1..m$ and $i=1..n$. Using (5),

$$\frac{\partial S_j}{\partial G_{ji}} = E_i \qquad (26)$$

Taking the partial derivative of U,

$$\frac{\partial U_k}{\partial G_{ji}} = \frac{\partial U_k}{\partial \Delta U_k}\frac{\partial \Delta U_k}{\partial h(S_j)}\frac{\partial h(S_j)}{\partial S_j}\frac{\partial S_j}{\partial G_{ji}} = 1 K_{kj}\, h'(S_j) E_i \qquad (27)$$

where,

$$h'(S_j) = \frac{\partial h(S_j)}{\partial S_j} \qquad (28)$$

The last form of G adaptation is obtained as,

$$\frac{dG_{ji}}{dt} = -\gamma_{ji}^2 S_j E_i - \gamma_{ji}^2 h'(S_j) E_i \left(\sum_{k=1}^{m} U_k K_{kj}\right) \qquad (29)$$

3.2 NN for the Equivalent Control Computation

If the knowledge of $F(X)$ and B matrices is very poor, then the equivalent control calculated will be too far off from the actual equivalent control. In this paper, a neural network is proposed to compute the equivalent control.

The aim of the network is to minimize the error between the desired equivalent control and the net output. The difference is actually the corrective term (ΔU), because of the equivalent control is enough to keep the system states on the sliding surface. The corrective term is necessary when the system is not on the sliding surface.

The net has two layer structure. The inputs (designated as Z) to the net consist of desired and actual states. Because, all of them are used for the computation of the equivalent control in (14). It has 8 elements for a 2 DOF scara robot. The hidden layer is selected arbitrarily. It is selected as 8 neurons for a two DOF scara robot and designated as *Ynet* and *Yout*, the net sum and output respectively. The output layer has two neurons and designated as *Unet* and *Ueq*.

$$Ynet_j = \sum_{i=1}^{8} Wz_{j,i} * Z_i \qquad j=1..8 \quad Yout_j = g(Ynet_j) \qquad (30)$$

$$Unet_j = \sum_{i=1}^{M=8} Wy_{j,i} * Yout_i \quad j=1..2 \quad Ueq_j = g(Unet_j) \qquad (31)$$

The cost function is selected as

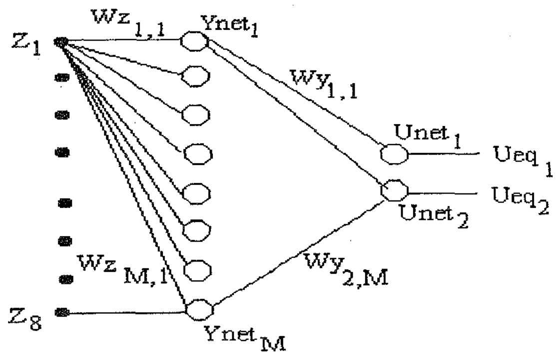

Figure 5. NN structure for U_{eq}

$$E = \frac{1}{2}\sum_{j=1}^{2}\left(Ueq_j^d - Ueq_j\right)^2 \qquad (32)$$

Gradient Descent (or back propagation) is used as

$$\frac{dWz_{j,i}}{dt} = -\mu \frac{\partial E}{\partial Wz_{j,i}} = \mu\, \delta z_j\, Z_i \qquad (33)$$

where,

$$\delta z_j = \left(\sum_{k=1}^{2} \delta y_k\, Wy_{k,j}\right) g'\left(Ynet_j\right) \qquad (34)$$

where g(.) is an activation function. It is selected as a shifted sigmoid function and its derivative is computed as,

$$g'(Ynet) = \left.\frac{dg(x)}{dx}\right|_{x=Ynet} = \frac{1}{2}\left(1 - g^2(Ynet)\right) \qquad (35)$$

Gradient descent for the output layer is computed as,

$$\frac{dWy_{j,i}}{dt} = -\mu \frac{\partial E}{\partial Wy_{j,i}} = \mu\, \delta y_j\, Yout_i \qquad (36)$$

where,

$$\delta y_j = \left(U_{eq,j}^d - U_{eq,j}\right) g'\left(Unet_j\right) = \left(\Delta U_j\right) g'\left(Unet_j\right) \qquad (37)$$

4. ROBOTICS APPLICATION

4.1 Robot Dynamics

The robot considered in this paper is a two degrees of freedom SCARA type experimental direct drive robot manipulator which is manufactured by Integrated Motion Corporation . Its structure is shown in Fig.6. The robot model can be written as,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + f_c = \tau \qquad (38)$$

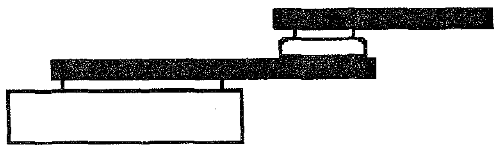

Figure 6. The structure of Direct Drive Scara

The details of the dynamics can be found in [8]. The model in (38) can be written in the state-space form representation as,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -M^{-1}(C x_2 + f_c) \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix} u \qquad (39)$$

where,

$$[x_1 \;\; x_2]^T = [q \;\; \dot{q}]^T \qquad \text{and} \qquad u = \tau$$

The equation (39) is in the form of (4), and the proposed method can be applied.

4.2 Simulation Results

The simulation studies carried out indicate that with a proper selection of gain matrices, proposed Neuro-SMC is capable of achieving a good trajectory following performance.

The desired trajectory used for simulation studies is shown in Fig. 7. It corresponds to the state references shown in Fig. 8. The initial values of G and K matrices are selected,

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \text{ and } K = \begin{bmatrix} 3 & 0.3 \\ 0.3 & 3 \end{bmatrix}$$

In the case of Neuro-SMC, the tracking errors are seen to be smaller as compared to the classical one. Moreover, the controller outputs display some chattering at the beginning and then become smooth when the sliding surfaces are reached. The results are presented in Figs. 11-13. Figures 9 and 10 indicates how the adaptation process evolves.

It should be pointed out that the initial errors in the angular positions are especially introduced to show the system behavior when the system is not on the sliding surface.

In the design of the Neuro-SMC, shifted sigmoid functions of the form

$$h(S_j) = \frac{2}{1+e^{-100 S_j}} - 1 \quad \text{and} \quad g(Ynet_j) = \frac{2}{1+e^{-Ynet_j}} - 1$$

are used instead of the signum function to be able to obtain differentiable functions.

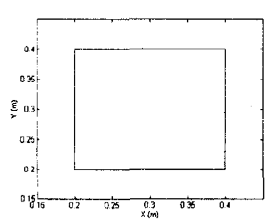

Figure 7. End effector Reference

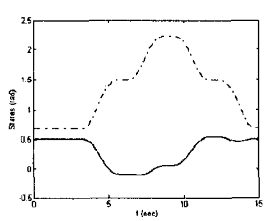

Figure 8. Angular References

The essential tuning parameters in adaptation are γ_i's in the equation (18). When selecting a value for γ_i, there are two criteria; adaptation capability and stability. While a low value causes to low adaptation capability, a high value may cause to instability. As a result, a sufficiently large value that does not make the system unstable should be chosen. The results presented here are obtained by,

$$\gamma^1_{ij} = \begin{bmatrix} 1.0 & 0.2 \\ 0.2 & 1.0 \end{bmatrix} \quad \gamma^3_{ij} = \begin{bmatrix} 0.7 & 0.2 \\ 0.2 & 0.7 \end{bmatrix} \quad \gamma^2_{ij} = 0.05 * \begin{bmatrix} 1.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 1.0 \end{bmatrix}$$

5. CONCLUSIONS

In this paper, a Neuro-Sliding Mode Controller is proposed and simulation results are presented. Two parallel NN are used to realize the Neuro-SMC.

The structure of the neural network for corrective term is such that its weights are the gains of SMC and its outputs are the corrective terms that are to be added to the equivalent control. An adaptation scheme using gradient decent is used to adapt the weights of the NN. The aim of the adaptation is to eliminate the chattering and to reduce the error. Therefore, the cost function is selected as sum of squares of the control signal and the sliding function.

In the design of a classical SMC, the controller output is obtained as the equivalent control plus a corrective term. The corrective term is necessary when the system deviates from the sliding surface. This term pushes the system back to the sliding surface. When the system on the sliding surface, the equivalent control is sufficient to keep the system on it. Therefore, the corrective term should be minimized when the system is in the vicinity of the sliding surface to minimize chattering. This can be achieved by minimizing the multiplicative gain K for the second layer. The G adaptation effects the slope of the sliding surface.

The structure of NN to compute the equivalent control is a standard two layer feedforward NN with back propagation adaptation algorithm and the error is accepted as the corrective term.

The Neuro-SMC has mainly five advantages;

1. Chattering is eliminated,
2. Error performance is improved,
3. The NN structure determination problem is solved for corrective term. In other words, number of layers, number of neurons and connections are well defined from SMC design,
4. There is no need to compute the inertia (or inverse) matrix to compute the equivalent control,
5. It is a robust Neuro-Controller.

The simulation results presented in this paper indicate that the suggested approach has considerable advantages compared to the classical one and is capable of achieving a good chatter-free trajectory following performance without an exact knowledge of plant parameters. These characteristics make it a promising approach for motion control applications.

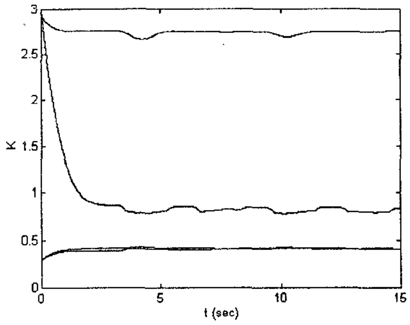

Figure 9. The Adapted Parameter K

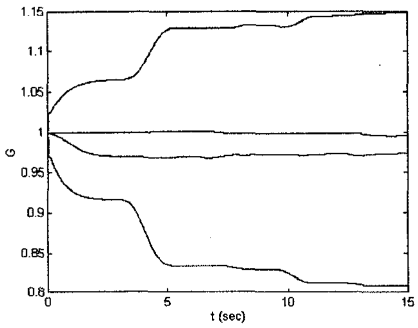

Figure 10. The Adapted Parameter G

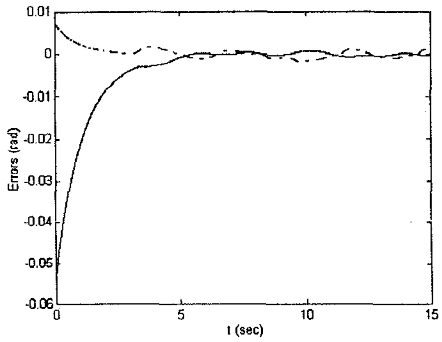

Figure 11. Angle errors for NN SMC

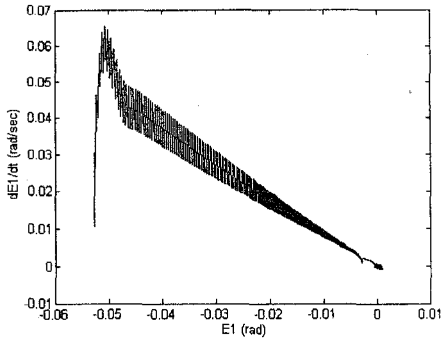

Figure 12. Phase Plane 1 for NN SMC

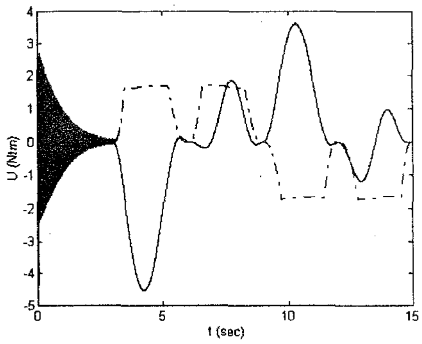

Figure 13. Controller Outputs for NN SMC

REFERENCES

[1] Hung J. Y., "Variable Structure Control: A survey", *IEEE Transaction on Industrial Electronics,* v.40, pp. 2-22, 1993.

[2] Utkin, V.I.,"Sliding Modes in Control Optimization", *Springer-Verlag*, 1981.

[3] Slotine, J.J. ans S.S. Sastry, "Tracking Control of Nonlinear Systems Using Sliding Surfaces with Applications to Robot Manipulators", *International Journal of Control,* v.38. pp.465-492, 1983.

[4] Tzafestas S. G., "Neural Networks in Robotics: State of the Art", *IEEE Int. Conf. On Industrial Electronics*, 1995.

[5] Ertuğrul, M., O. Kaynak, A. Sabanovic and K. Ohnishi, "A Generalized Approach for Lyapunov Design of Sliding Mode Controllers for Motion Control Applications", *IEEE 4th International Workshop on Advanced Motion Control, AMC'96,* pp.407-412, Mie University, Japan, 1996.

[6] J. J. Slotine and W. Li, "Applied Nonlinear Control", *Prentice Hall*, 1991.

[7] T. Fukuda et al, "Neural Servo Controller; Adaptation and Learning", *Proc. IEEE International Workshop on Advanced Motion Control,* pp.107-115, 1990.

[8] Direct Drive Manipulator R&D package User Guide. Integrated Motion Incorporated, Berkeley CA.