NEURAL SELF-TUNING ADAPTIVE CONTROL OF NON-MINIMUM PHASE SYSTEM

Long T. Ho, Jan T. Bialasiewicz, Hai T. Ho
University of Colorado at Denver
Electrical Engineering and Computer Science Department
1200 Larimer Street, Campus Box 110
Denver, Colorado 80204-5300

Abstract

The motivation of this *research came about when a neural network direct adaptive control scheme was applied to control the tip position of a flexible robotic arm. Satisfactory control performance was not attainable due to the inherent non-minimum phase characteristics of the flexible robotic arm tip. Most of the existing neural network control algorithms are based on the direct method and exhibit very high sensitivity if not unstable closed-loop behavior. Therefore a neural self-tuning control (NSTC) algorithm is developed and applied to this problem and showed promising results. Simulation results of the NSTC scheme and the conventional self-tuning (STR) control scheme are used to examine performance factors such as control tracking mean square error, estimation mean square error, transient response, and steady state response.

1. Introduction

Self-tuning adaptive control used for controlling unknown ARMA plants has traditionally been based on the minimum variance control law and a recursive identification algorithm (Astrom and Wittenmark, 1973; Clark and Gawthrop, 1979). Although the advancement in VLSI has made it more possible to implement real-time recursive algorithms but it is still computationally intensive due to the recursive nature of the algorithm. On the other hand, neural networks VLSI has been made available commercially with extreme processing capability due to its parallel architecture. With this in mind the possibility of formulating neural networks to perform functions of conventional recursive algorithms becomes important. Hence, in this paper we propose the neural self tuning control (NSTC) scheme where the implicit identification is performed by a multilayer neural network (MNN) and the control is based on the generalized minimum variance (GMV) control law.

Neural networks have undoubtedly demonstrated its effectiveness in controlling nonlinear systems with known/unknown dynamics and uncertainties (Narendra and Parthasrathy, 1990; Levin and Narendra, 1993; Werbos et al. 1990; Hunt et al., 1992). In addition, neural network

* This research was funded by NASA Langley Research Center Grant No. NAG-1-1444

adaptive control algorithms have also been developed for specific linear system model such as the state space model (Ho et al., 91a) and the ARMA model (Ho et al., 1991b). It was shown in the simulation results that neural network controllers produced comparable results to conventional adaptive controllers. In this paper, we investigate the performance of the NSTC and compare it to the conventional adaptive STR.

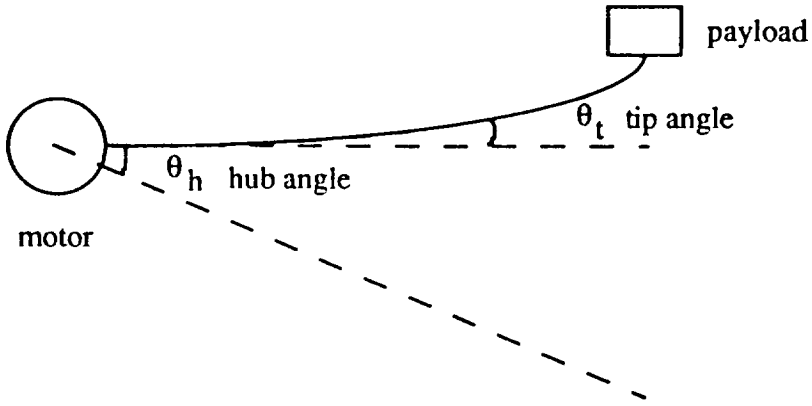

Figure 1.1. Flexible arm system

The flexible arm to be controlled is shown in Figure 1.1. There are two system outputs that are of interest, one is the hub angle $\theta_h(t)$ and the other is the tip angle $\theta_t(t)$ of the arm. The goal is to apply a neural network control scheme to control these outputs to track the command angle. The neural controller will generate a control voltage signal u(t) that will feed the power amplifier in which will force current through the motor and cause the arm position to react. The dynamical transfer function of the hub angle is a linear minimum phase system in which will be shown readily controllable by a neural network. In fact, the direct adaptive neural control scheme in Figure 1.2. can be used to control the hub. This control scheme belongs to the type called specialized learning control (Psaltis et al., 1988; Ho et al., 1991c). However, the tip of the arm, being in a different location from the actuator point, therefore making the system to be of the type non-collocated system. The effect of this dynamically is that there is a zero on the right half side of the s-plane. In other words, the transfer function of the tip angle is of the type non-minimum phase which presents itself to be very difficult to control when direct adaptive control methodology is applied. This difficulty may be due to the controller trying to emulate the inverse dynamics of the non-minimun plant and results in an unstable behavior. According to simulation studies the specialized learning control algorithm diverges when applied to control the tip angle. Most other neural control schemes are also based on the inverse dynamics including the indirect learning method by (Psaltis et al., 1988), the feedback error learning by (Kawato et al., 1988), and the methods presented by (Narendra and Parthasarathy, 1990).

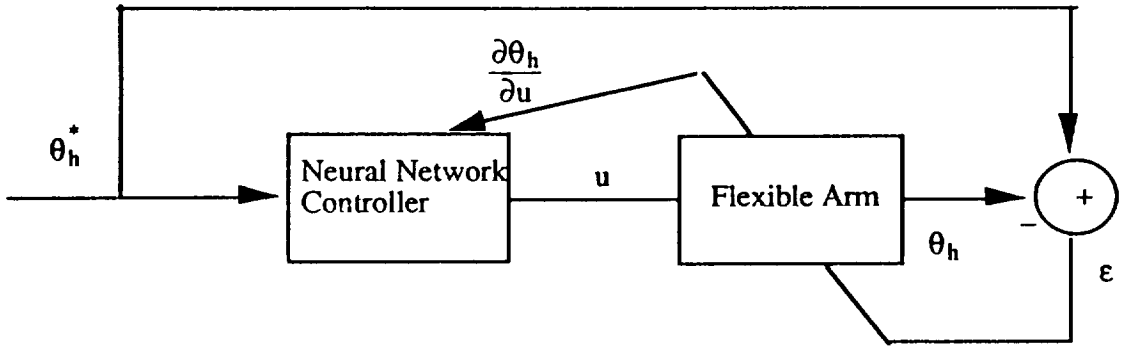

Figure 1.2. specialized learning control of hub

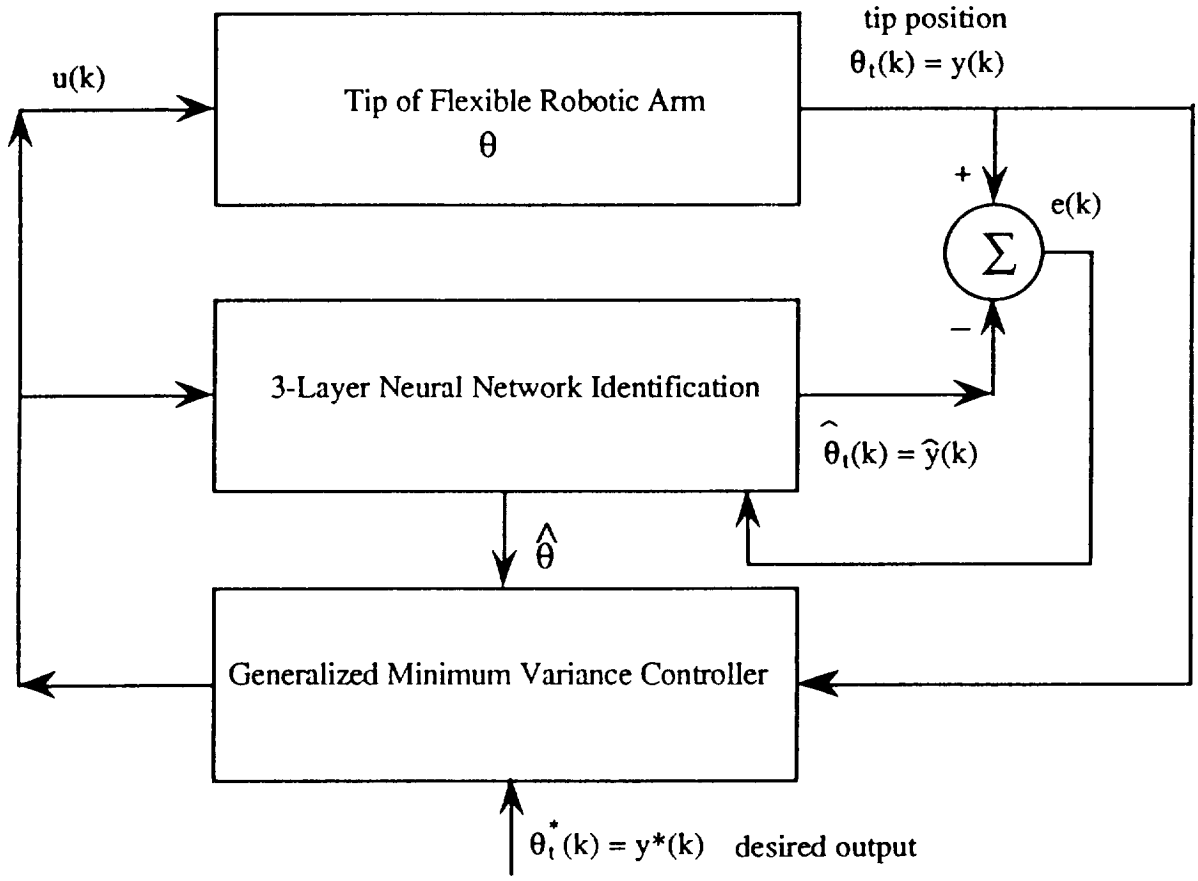

Figure 1.3 Indirect neural adaptive control scheme

In this report, we propose to use the neural self tuning control scheme which is based on an indirect control method (Ho et al., 1991c) to control the tip angle. This scheme is shown in Figure 1.3 where the identification is performed by the MNN and the control is performed by the generalized minimum variance (GMV) controller. The GMV control algorithm has a dynamic weighting function $Q(q^{-1})$ applied to the plant control signal u(k) in the cost function to limit and

condition the control energy. Thus, upon selecting the proper weighting function the controller can be input/output stable and effective in controlling the non-minimum phase plant. In section 2 the neural self-tuning control (NSTC) which consists of the minimum variance control algorithm and the neural identification is presented. Section 3 presents a comparative simulation study of the adaptive STR scheme and the NSTC scheme. And section 4 gives the conclusion of the results found in this study and address the advantages and disadvantages of the neural control scheme used for treating linear system.

2. Stochastic neural self-tuning adaptive control (NSTC)

The NSTC consists of the minimum variance control law and the neural identification algorithm. The model assumed for the plant is of ARMA input/output type having the form

$$y(k) = q^{-d}\frac{B(q^{-1})}{A(q^{-1})}u(k) + \frac{C(q^{-1})}{A(q^{-1})}\xi(k) \tag{2.1}$$

where u(k), y(k), ξ(k), and d are system input, output, uncertainty, and delay, respectively. A, B, and C are unknown system dynamics defined as

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + \ldots + a_{na}q^{-na} \tag{2.2}$$

$$B(q^{-1}) = b_0 + b_1q^{-1} + b_2q^{-2} + \ldots + b_{nb}q^{-nb} \tag{2.3}$$

$$C(q^{-1}) = 1 + c_1q^{-1} + c_2q^{-2} + \ldots + c_{nc}q^{-nc} \tag{2.4}$$

where q is the shift operator. For the above unknown plant, in Figure 1.3, the objective is to control its output to track a command signal y*(k) based on the generalized minimum variance control index (Clark and Gawthrop, 1979)

$$\begin{aligned} J(k+d) &= E\{\phi^2(k+d)\} \\ &= E\{[P(q^{-1})y(k+d)+Q(q^{-1})u(k)-R(q^{-1})y^*(k)]^2\} \\ &= E\{[\phi_y(k+d)+Q(q^{-1})u(k)-R(q^{-1})y^*(k)]^2\} \end{aligned} \tag{2.5}$$

where E is the expectation operator, ϕ_y(k+d) is the auxiliary output, and P,Q, and R are the weighting dynamics which can be chosen depending on the required response characteristics.

2.1. Generalized minimum variance control

In this section, the generalized minimum variance self-tuning control algorithm for the above problem statement is summarized (Clark and Gawthrop, 1979). To obtain the optimal control u(k) which minimizes the performance index (2.5), the predictive auxiliary output ϕ_y(k+d) in terms of the system dynamics must be determined. Consider the following identity

$$\frac{P(q^{-1})C(q^{-1})}{A(q^{-1})} = F(q^{-1}) + q^{-d}\frac{G(q^{-1})}{A(q^{-1})} \tag{2.1.1}$$

where the order of $F(q^{-1})$ and $G(q^{-1})$ are nf=d-1, ng=na-1, respectively. The output prediction can be shown to have the form

$$\phi_y(k+d) = \hat{\phi}_y(k+d) + \tilde{\phi}_y(k+d) \tag{2.1.2}$$

where

$$\hat{\phi}_y(k+d) = C(q^{-1})^{-1}[G(q^{-1})y(k) + F(q^{-1})B(q^{-1})u(k)]$$
$$= C(q^{-1})^{-1}[G(q^{-1})y(k) + E(q^{-1})u(k)] \tag{2.1.3}$$

and

$$\tilde{\phi}_y(k+d) = F(q^{-1})\xi(k+d) \tag{2.1.4}$$

$\hat{\phi}_y(k+d)$ and $\tilde{\phi}_y(k+d)$ are the deterministic and uncorrelated random components of $\phi_y(k+d)$. Next, substituting (2.1.2) into (2.5), there results

$$J(k+d) = E\{[\hat{\phi}_y(k+d)+Q(q^{-1})u(k)-R(q^{-1})y^*(k)]^2\}+E[\tilde{\phi}_y(k+d)]^2\} \tag{2.1.5}$$

Since the second term in (2.1.5) is unpredictable random noise which is uncompensatable by the control input u(k), and the first term is a linear function of u(k), J(k+d) can be minimized by setting

$$[\hat{\phi}_y(k+d)+Q(q^{-1})u(k)-R(q^{-1})y^*(k)] = 0 \tag{2.1.6}$$

Solving for the generalized minimum variance (GMVC) control in (2.1.6) gives

$$u(k) = \frac{R(q^{-1})y^*(k)-\hat{\phi}_y(k+d)}{Q(q^{-1})} \tag{2.1.7}$$

using (2.1.3), (2.1.7) can also be written as

$$u(k) = \frac{C(q^{-1})R(q^{-1})y^*(k)-G(q^{-1})y(k)}{E(q^{-1})+C(q^{-1})Q(q^{-1})} \tag{2.1.8}$$

Remarks : Recall that $E(q^{-1})$ is equal to $F(q^{-1})B(q^{-1})$ where $B(q^{-1})$ contains the zeros of the plant. Notice that having the weighting function $Q(q^{-1})$ additive to $E(q^{-1})$ in (2.1.8) gives the designer the ability to alter the poles of the controller. Thus with a non-minimum phase plant $B(q^{-1})$ shall have unstable roots but proper selection of $Q(q^{-1})$ in (2.1.8) can assure the control signal u(k) to be bounded.

2.2. Neural system identification

In this section, the framework of the neural identification algorithm for the self-tuning control scheme in Figure 1.3 is presented. Recall the predicted auxiliary output in (2.1.3) which can also be written as

$$\phi_y(k+d) = C(q^{-1})^{-1}[G(q^{-1})y(k) + E(q^{-1})u(k)] + F(q^{-1})\xi(k+d)$$
$$= C(q^{-1})^{-1}[G(q^{-1})y(k) + E(q^{-1})u(k)] + v(k) \tag{2.2.1}$$

where the uncorrelated noise sequence $F(q^{-1})\xi(k+d)$ is replaced by v(k). Also (2.2.1) can be written as

$$\phi_y(k+d) = \sum_{i=0}^{ng} g_i y(k-i) + \sum_{i=0}^{ne} e_i u(k-i) - \sum_{i=1}^{nc} c_i \phi_y(k+d-i) + v(k) \tag{2.2.2}$$

$$\phi_y(k+d) = \psi'(k)\theta(k) + v(k) \tag{2.2.3}$$

where

$$\psi'(k) = [y(k)...y(k-ng); u(k)...u(k-ne); \phi_y(k+d-1)...\phi_y(k+d-nc)] \tag{2.2.4}$$

$$\theta'(k) = [g_0\ g_1 \cdots g_{ng}; e_0\ e_1 \cdots e_{ne}; -c_1\ -c_2 \cdots -c_{nc}] \tag{2.2.5}$$

since the parameter vector θ is unknown, the estimated form of $\phi_y(k+d)$ is given as

$$\hat{\phi}_y(k+d) = \hat{\psi}'(k)\hat{\theta}(k) \tag{2.2.6}$$

where

$$\hat{\psi}'(k) = [y(k)...y(k-ng); u(k)...u(k-ne); \hat{\phi}_y(k+d-1)...\hat{\phi}(k+d-nc)] \tag{2.2.7}$$

$$\hat{\theta}'(k) = [\hat{g}_0\ \hat{g}_1 \cdots \hat{g}_{ng}; \hat{e}_0\ \hat{e}_1 \cdots \hat{e}_{ne}; -\hat{c}_1\ -\hat{c}_2 \cdots -\hat{c}_{nc}] \tag{2.2.8}$$

The unknown parameter vector in (2.2.8) (Figure 2.1), is taken from the output of the neural network

$$\begin{aligned}\hat{\theta}(k) &= [\hat{\theta}_1(k)\ \hat{\theta}_2(k) \cdots \hat{\theta}_j(k) \cdots \hat{\theta}_{n3}(k)]' \\ &= [O_1(k)\ O_2(k) \cdots O_j(k) \cdots O_{n3}(k)]'\end{aligned} \tag{2.2.9}$$

Where n3 is the number of neurons at the output layer. Consider the system identification cost function

$$\begin{aligned}V(k) &= \frac{1}{2} E\{\varepsilon'(k)\Lambda^{-1}(k)\varepsilon(k)\} \\ &= \frac{1}{2} E\{[\phi_y(k)-\hat{\phi}_y(k)]'\Lambda^{-1}(k)[\phi_y(k)-\hat{\phi}_y(k)]\}\end{aligned} \tag{2.2.10}$$

where $\Lambda(k)$ is a symmetric positive definite weighting matrix, and $V(k)$ is minimized by adjusting the weights of the neural identifier.

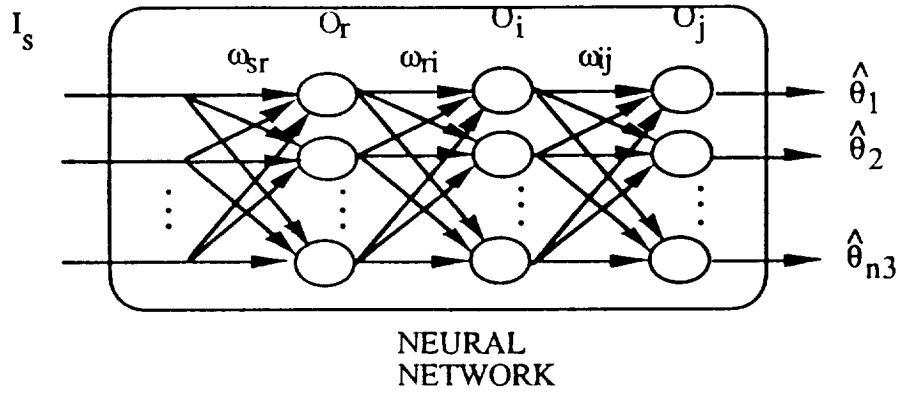

Figure 2.1. Neural network structure

In Figure 2.1, the weights interconnecting the layers will be based on the back propagation approach (Rumelhart and McClelland, 1986). Consider a general weight vector of n weights

$$\omega = [\omega(1)\ \omega(2)\ \ldots\ \omega(n)]'$$

The update equation of this weight vector can be expressed as

$$\omega(k+1) = \omega(k) - \eta\frac{\partial V(k)}{\partial \omega(k)}$$

$$= \omega(k) + \eta\frac{\partial \hat{\theta}'(k)}{\partial \omega(k)}\frac{\partial \hat{\phi}_y'}{\partial \hat{\theta}(k)}\Lambda^{-1}\varepsilon(k) \qquad (2.1.11)$$

where η is the learning step size. From (2.2.6) we see that the second partial derivative term in (2.1.11) is

$$\frac{\partial \hat{\phi}_y'}{\partial \hat{\theta}(k)} = \hat{\psi}(k+d) \qquad (2.1.12)$$

and the first partial derivative term in (2.1.11) can be derived using the differentiation chain rule. Hence, for the individual weights in each of the neural network layer in Figure 2.1. can be formulated by using (2.1.12) and the back propagation derivation (Rumelhart and McClelland, 1986). The complete algorithm can be found in (Ho et al., 1991b). Once the estimate of θ is available, $\hat{\phi}_y(k+d)$ in (2.2.6) can be computed, and then the control signal can be generated using (2.1.7).

3. Empirical studies

In this section we examine some simulation results of the indirect neural control scheme for controlling the flexible arm tip. An attemp to use the direct adaptive control scheme to control the tip position gave unstable results even after numerous controller parameter changes (Ho et al., 1993). Next, the NSTC scheme in section 2 was applied to control the tip position and produced encouraging results. Lastly, the neural identifier in the NSTC algorithm is compared with the recursive least square identifier.

Recall that this scheme (Figure 1.3.) has two distinct functions, identification and control, which are done by the neural network and the (GMV) control, respectively. In this section we perform the simulations of two schemes which are: The adaptive STR using recursive least square identification, and the NSTC using the neural identification. This is so that a comparative study can be done to assess the performance of the developed NSTC.

Simulation: The transfer fucntion of the arm tip position that was derived theoretically and measured experimentally in (Fraser and Daniel, 1991) is

$$\frac{\theta_t(s)}{U(s)} = \frac{3.6\,(1-\frac{s^2}{48.4^2})}{s(s+0.16)} \qquad (3.1)$$

Here, the resonant modes are excluded and assumed to be completely filtered out because we are mainly interested in the effects of the unstable zero. This model is then discretized with a sampling period of 6 msec. Recall the control index defined in section 2

$$J(k+d) = E\{\phi^2(k+d)\}$$
$$= E\{[P(q^{-1})y(k+d)+Q(q^{-1})u(k)-R(q^{-1})y^*(k)]^2\} \qquad (3.5)$$

where the weighting functions were chosen as

$$P(q^{-1})=1;\ Q(q^{-1})=.1+.06q^{-1};\ R(q^{-1})=1 \qquad (3.6)$$

and the desired hub position $\theta_t^*(k)$ was a step command. Beginning with Figure 3.4. showing the desired step tip response, the controlled tip response based on the adaptive STR and the tip response from the NSTC. Obviously both controllers manage to track the command signal. However, the NSTC seems to have a slower settling time. Figure 3.5. shows the converging tracking control index (2.1.5) where both schemes seem very comparable to each other. Figure 3.6. displays the comparable control energy produced by these controllers. Note that the transient control energy was affected by two factors: one is the initial condition of the estimated parameter vector $\hat{\theta}_0$ (which was set as $\hat{\theta}_0$ = [1 1 ... 1]' for both control schemes), the further $\hat{\theta}_0$ is away from the optimum $\hat{\theta}^*$ in the parameter state space, the longer the convergence of the tracking control index (2.1.5). The other factor is the selection of the input weighting function $Q(q^{-1})$ which has the effect of limiting the control energy with the tradeoff of slower tracking convergence. Lastly, we compare the recursive least square identification with the neural network identification. The two identifiers estimate the parameter vector θ in (2.2.5) so that the predictive output term $\hat{\phi}_y(k+d)$ in (2.2.2) can be computed. Figure 3.7. shows the estimation cost function V(k) in (2.2.10) response of the RLS and the neural network. V(k) of the RLS has a slightly faster convergence than the neural network but not by a significant degree. Again, this indicates that the identification performance of the two algorithms are comparable to each other. The convergence of the neural network in this kind of control application is serveral orders of magnitude faster than other applications. In this case it took less than 300 iterations for the 3-layer nerual network to be maturely trained. This indicates that it is practical for real-time implementation. For completeness, the time response of the true output $\theta_t(k)$ and the estimated output $\hat{\theta}_t(k)$ produced by the neural network is shown in Figure 3.8.

Neural Network: The three layer neural network $N^3_{2,5,15,P0}$ used in this scheme consists of one input layer, two hidden layers, and one output layer with the number of neurons as 2, 5, 15, and P0, respectively. P0 is the length of the vector defined in (2.2.8) which is (ng+1)+(ne+1)+nc, and

is 11 for the case of the arm tip plant. The input of the neural network was a selected as constant vector Is = [1 1]' because it was desired that the output of the neural network to be correlated to the its input. The parameters of the sigmoidal activation function at the output node was found to be most influential on the tracking error convergent rate. Predominantly the slope of the activation function was observed to be proportional to the estimation convergent rate V(k). Also the bipolar sigmoidal saturation levels of the output neuron needed to be set equal to or greater than the maximum component of the parameter vector θ. The tuning of the sigmoidal functions was done manually by trial and error. Autotuning of the sigmoidal function parameters can also be applied to obtain statiscally better results (Yamada and Yabuta, 1992; Proano, 1989). However, the optimal dimension of the neural network in terms of number of layers and nodes was not known and therefore an initial pick of $N^3_{2.5.15.P0}$ was used throughout the simulation.

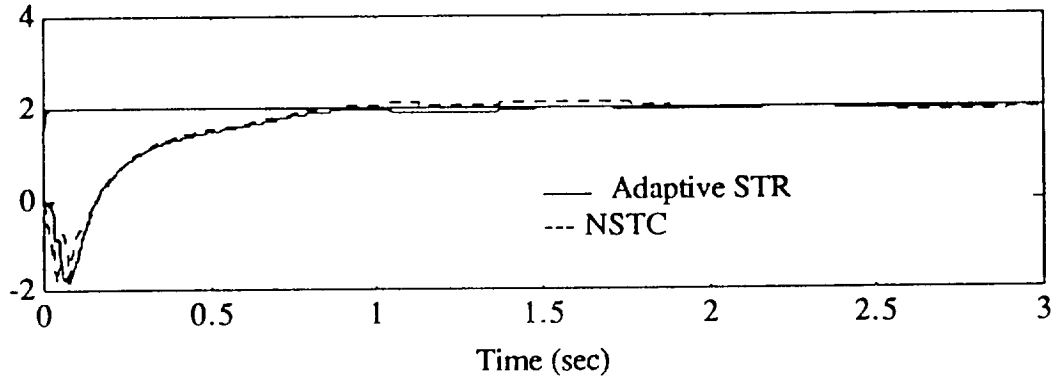

Figure 3.4. Tip position response: $\theta_t^*(k)$ & $\theta_t(k)$ of the adaptive STR and the NSTC

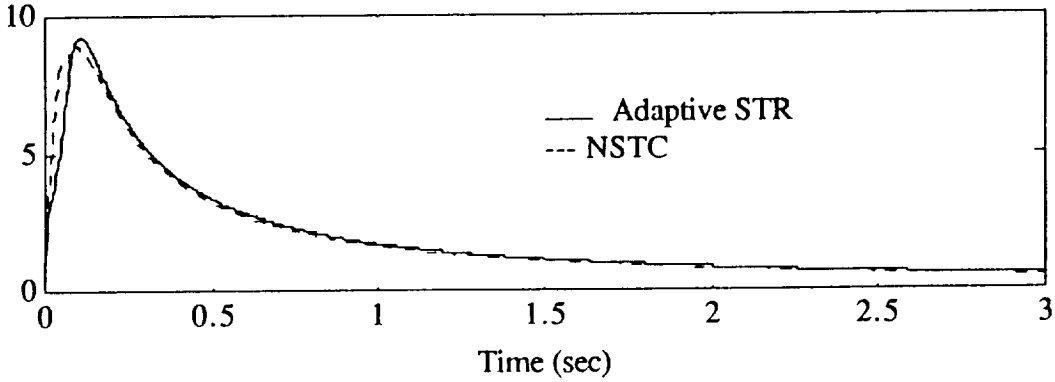

Figure 3.5. Control performance index J(k) of the adaptive STR and the NSTC

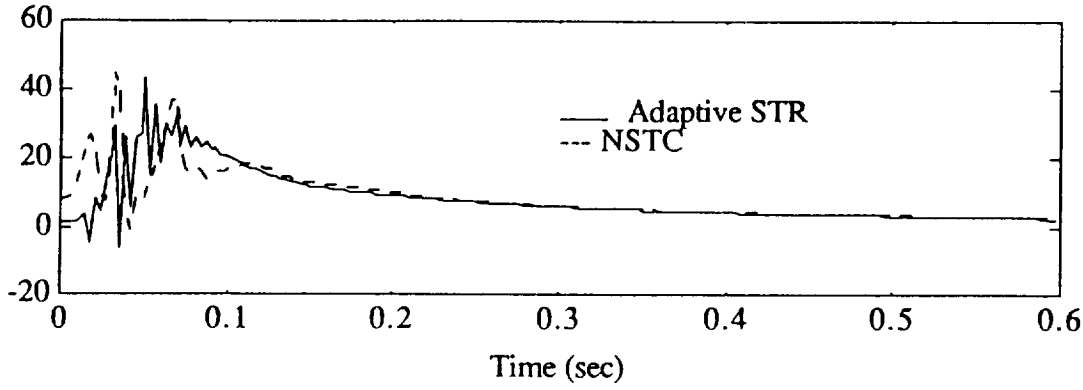

Figure 3.6. Control signal u(k) of the adaptive STR and the NSTC

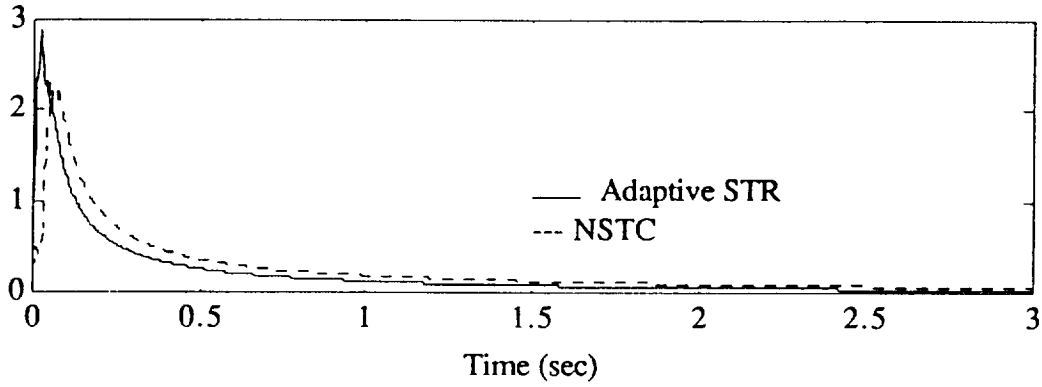

Figure 3.7. Identification cost index V(k) of the adaptive STR and the NSTC

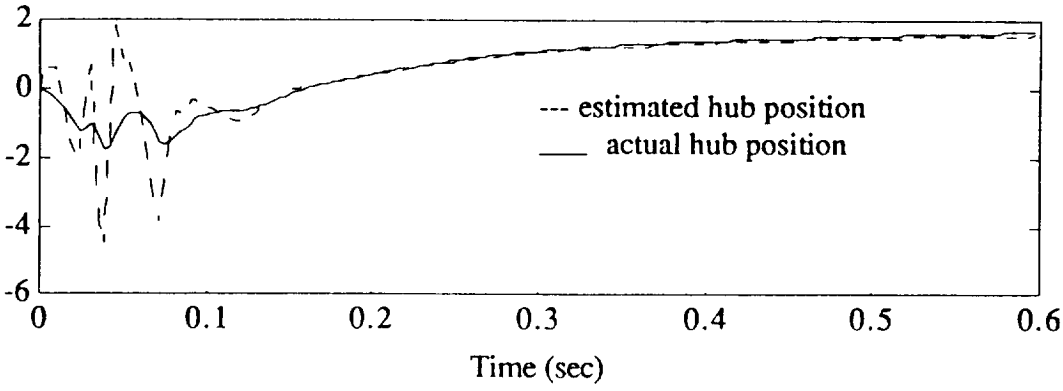

Figure 3.8. True and neural network estimated tip position: $\theta_t(k)$ & $\hat{\theta}_t(k)$

4. Conclusion

The neural self-tuning control (NSTC) algorithm was developed and applied to control the tip of a flexible arm system. The dynamics of the flexible arm tip involves an unstable zero and therefore making the system non-minimum phase. most of the existing neural adaptive control are based on the inverse dynamics and therefore would not be able to control this type of plant. The NSTC was based on an indirect control method where the identification is performed by the neural network and the control was based on the generalized minimum variance (GMV) control law. The performance of the NSTC was investigated and was compared to the adaptive STR by means of simulation.

In summary, the NSTC has a very comparable performance to the adaptive STR shown by simulation results in section 3. Unlike other applications of neural networks where thousands of iterations were required before the network can be maturely trained, in this application the neural network identification had a convergent rate comparable to that of the RLS. Another advantage of the NSTC is due to the availability of neural network VLSI and the massive parallel architecture of the neural network there will be a computation advantage over conventional recursive algorithms. This will enable real-time implementation with faster sampling rate for system with high bandwidth. Also another advantage of the NSTC is that because the identification is done by the neural network it inherits the decentralize property, meaning if there is a failure in a node or connection the impact on the performance will be minimal. Whereas with the conventional digital filter a failure in one of the coefficients will have a major impact on the output. With all the above encouraging characteristics there is one important disadvantage of using the neural network and that the the lack of understanding how the dimension and activation characteristics of a network is related to its accuracy and stability. Whereas, these issues of the recursive algorithms have been addressed and elaborately analysed (Kumar, 1990).

References

Astrom, K. , and Wittenmark, B., "On self-tuning regulators", Automatica, 9, pp.185-199, 1973.

Clark, D., and Gawthrop, P., "Self-tuning controller", in proceedings IEE, 126, pp.633-640, 1979.

Fraser, A., and Daniel, R., Perturbation Techniques for Flexible Manipulators, Kluwer Academic Publishers, Norwell, Massachusetts, 1991.

Ho, T., Ho, H. "Stochastic state space neural adaptive control", in proceedings the Third International Conference on Advances in Communication and Control Systems, Victoria, B.C., Canada, Oct. 16-18, 1991a.

Ho, H., Ho, T., Wall, E., and Bialasiewicz, J., "Stochastic neural self-tuning adaptive control", in proceedings of the Third International Conference on Advances in Communication and Control Systems, Victoria, B.C., Canada, Oct. 16-18, 1991b.

Ho, T., Ho, H., Wall, E., and Bialasiewicz, J., "Stochastic Neural Direct Adaptive Control", in proceedings of the 1991 IEEE International Symposium on Intelligent Control, Arlington, Virginia, Aug. 13-15, 1991c.

Ho L., and Bialasiewicz, J., "Neural Self-Tuning Adaptive Control of Non-Minimum Phase System", Independent Study Report, University of Colorado at Denver, Department of Electrical Engineering, May 1993.

Hunt K., Sbarbaro, D., Zbikkowski, R., and Gawthrop, P., "Neural Networks for Control System - A survey", Automatica, Vo. 28, No. 6, 1992.

Kawato, M., Setoyama, T. and Suzuki, R. "Feedback-error-learning of movement by multi-layer neural network", in proceedings of the International Neural Networks Society First Annual Meeting, 1988.

Kumar, P., "Convergence of Adaptive Control SchemesUsing Least-Squares Parameter Estimates", IEEE Transactions on Automatic Control, Vol. 35, NO 4, pp. 416-424, April 1990.

Levin. A., and Narendra, K., "Control of Nonlinear SystemsUsing Neural Networks: Controllability and Stablilization", IEEE Transactions on Neural Networks, Vol 4. NO. 2, March 1993.

Miller, T., Sutton, R., and Werbos, P., Neural Networks for Control, The MIT press, Cambridge, Massachusetts, 1990.

Narendra, K., and Parthasarathy, K., "Identifciation and control of dynamical systems using neural networks" IEEE Transactions on Neural Networks, Vol 1. NO. 1, March 1990.

Proano, J., Neurodynamic Adaptive Control Systems, Ph.D Dissertation, University of Colorado, Boulder, 1989.

Psaltis, D., Sideris, A., and Yamamura, A., "A multilayered neural network controller" IEEE Control Systems Magazine, April 1988.

Rumelhart, D., and McClelland, J., Parallel Distributed Processing: Vol 1, Foundations, The MIT Press, 1986.

Yamada, T., and Yabuta, T., "Neural Network Controller Using Autotuning Method for Nonlinear Functions", IEEE Transactions on Neural Networks, Vo 3, No. 4, July 1992.