International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:2, No:8, 2008

Neural Adaptive Switching Control of Robotic Systems

A. Denker, and U. Akıncıoğlu

***Abstract*—**In this paper a neural adaptive control method has been developed and applied to robot control. Simulation results are presented to verify the effectiveness of the controller. These results show that the performance by using this controller is better than those which just use either direct inverse control or predictive control. In addition, they show that the resulting is a useful method which combines the advantages of both direct inverse control and predictive control.

***Keywords*—**Neural networks, robotics, direct inverse control, predictive control.

I. Introduction

CLASSICAL control systems have long been found to be inadequate for robotic systems whose mathematical models are not fully available. The widespread robot control technique has been the model based computed torque control which is subjected to performance degradation due to model uncertainties. Driven by the desire for a high degree of automation, fast speed operation, and high performance requirement from industry, there has been considerable research interest in neural network control of robots and satisfactory results have been obtained in solving some of the special issues associated with the problems of uncertainty. Increasingly sophisticated neural network applications have been developed for better industrial performance. [1,2]. Concurrent advances in microprocessor technology have made the implementation of them practically realizable.

It is known that when they are used as controllers neural networks must be able to realize the dynamics. However, while it is true that neural networks have learning capability and can be used to approximate any dynamic function to any desired accuracy as long as the size of the network is large enough, one should not abuse their capabilities and let them learn all the characteristics of the system without any discrimination.

Manuscript received June 30, 2006.

A. Denker is with the Department of Electronic Engineering, Ankara University, Besevler, Ankara, Turkey, and Faculty of Engineering, Cyprus International University, Nicosia, N. Cyprus (e-mail: adenker@ciu.edu.tr).

U. Akıncıoğlu is with Aselsan, Ankara, Turkey (e-mail: uakincioglu@aselsan.com.tr).

Otherwise the sizes of the required networks are known to grow to uncontrollable magnitudes. This is, obviously, very undesirable for an actual implementation which requires less computational power and accordingly less power and cost.

The concept of switching between direct inverse control and predictive control applications is put forward in this paper in order to eliminate those problems which are associated with the approximation nature of neural networks. These two employed control approaches are conceptually different. In direct inverse control applications, artificial neural networks are used as the controller, on the other hand, in predictive control applications artificial neural networks are used for identification of the process. It is shown in this paper how these two well known but different methods can be unified and generalized in a straightforward way.

The neural adaptive switching technique between direct inverse control and predictive control combines the advantages of predictive control, neural networks and inverse direct control. By attenuating the effects of both parametric uncertainties and uncertain non-linearities this approach can achieve asymptotic tracking. Simulation results are presented to verify the effectiveness of the controller.

II. Neural Adaptive Switching Control

Neural networks can be used in control directly or indirectly. While in direct applications neural networks are used in the controller, in indirect applications they are used in modeling the system. Both of these uses will be implemented in this paper.

A. Indirect Use of Neural Networks

The robot model plays a decisive role in the controller; the model must be able to capture the robot dynamics in order to enable the precise prediction of the control action. Benefits are affected by the discrepancies existing between the real process and the model used. The controller performance degrades quickly as discrepancies rise due to increase in speed or variation in load. The learning capability of neural networks and especially their ability to approximate to nonlinear functions makes them useful for modeling purpose. Neural networks are implemented here for constructing the model of the robotic system by observing the input and the output variables online and offline. Let's consider the system governed by the following nonlinear discrete-time difference equation:

$$y(t+1) = f[y(t),...,y(t-n+1);u(t),...,u(t-m+1)] \qquad (1)$$

As shown in this equation the output of the system at time t+1 depends on the past n output values (y) and the past m input values (u) in the sense defined by the nonlinear function *f*.
The output of the neural network model is denoted by y^m then;

$$y^m(t+1) = f^m[y(t),...,y(t-n+1);u(t),...,u(t-m+1)] \quad (2)$$

f^m is the nonlinear function of the neural network i.e. approximation to function f. In this equation the inputs of the neural network depend on the past outputs of the system. After the required training process we can assume $y^m \approx y$ then, the model becomes independent from the real plant, and the equation becomes;

$$y^m(t+1) = f^m[y^m(t),...,y^m(t-n+1);u(t),...,u(t-m+1)] \quad (3)$$

Subsequent to off-line training of the neural networks to obtain sufficiently accurate approximations of the input-output relationships of the robot they can be utilized to construct controllers.

B. Direct Use in Direct Inverse Control

In direct use neural networks are employed in the controller of the system. Direct inverse control utilizes an inverse system model. The inverse model is simply cascaded with the controlled system in order that the composed system results in an identity mapping between desired response (i.e. network inputs) and the control system output. [3]
Let the system to be controlled is expressed as;

$$y(t+1) = g\left[y(t),...,y(t-n+1,u(t),...,u(t-m)\right] \quad (4)$$

Control signal produced by the neural network;

$$\hat{u}(t) = \hat{g}^{-1}\left[y(t+1), y(t),...,y(t-n+1),u(t),...,u(t-m)\right] \quad (5)$$

Network can be used to control the system by substituting the output at time t+1 by the desired output, the reference, r(t+1). If the network represents the exact inverse, the control input produced by it will thus drive the system output at time t+1 to r(t+1). [10]

In the case of nonadaptive neural control, since the quality of their performance relies heavily on the fidelity of the model used for the design of the controller, they can only be used if the model is very accurate and the effect of disturbances is small. [11]

C. Indirect Use in Predictive Control

The success of the predictive control approach is attributed first and foremost to the existence of a reliable model. .It requires an explicit system model to predict the future response of the robot. In the formulation of the controller the predicted output, output and the reference signal are used.
The objective of the predictive control strategy using neural networks is twofold: (i) to estimate the future output of the plant and (ii) to minimize a cost function based on the error between the predicted output of the processes and the reference trajectory. [12, 13]
Predictive control minimizes the cost function J. Consider the system with the cost function given:

$$\begin{aligned} J(t,U(t)) &= \left[R(t)-\hat{Y}(t)\right]^T\left[R(t)-\hat{Y}(t)\right] + \rho\tilde{U}(t)\tilde{U}(t) \\ &= E^T(t)E(t) + \rho\tilde{U}(t)\tilde{U}(t) \end{aligned} \quad (6)$$

where

$$\begin{aligned} R(t) &= \left[r(t+N_1)....\ r(t+N_2)\right]^T \\ \hat{Y}(t) &= \left[\hat{y}(t+N_1|t)...\ \hat{y}(t+N_2|t)\right]^T \\ E(t) &= \left[e(t+N_1|t)...\ e(t+N_2|t)\right]^T \\ \tilde{U}(t) &= \left[\Delta u(t)...\ \Delta u(t+N_u-1)\right] \end{aligned} \quad (7)$$

and

$$e(t) = r(t+k) - y(t+k|t) \quad k=N_1,\ldots N_2 \quad (8)$$

$$\Delta u(k+i-1) = 0 \quad 1 \le N_u < i \le N_2 \quad (9)$$

In these equations:

N_1: minimum prediction

N_2: prediction horizon

N_u: control horizon.

ρ: weighting factor penalizing changes in controls

provided that the system to be controlled is deterministic, the one-step ahead prediction is given by;

$$\begin{aligned} \hat{y}(t) &= \hat{y}(t|t-1) \\ &= \hat{g}_1\left[y(t-1),...,y(t-n),u(t-d),...,u(t-d-m)\right] \end{aligned} \quad (10)$$

g is the function realized by the neural network, d is the time delay. K-step ahead output can be calculated by shifting the equation in time while substituting the predictions for actual measurements as they do not exist;

$$\begin{aligned} &\hat{y}(t+k) = \hat{y}(t+k|t) \\ &= g[\hat{y}(t+k-1),..\hat{y}(t+k-\min[k,n]),y(t-1),...,y(t-\max[n-k,0]), \\ &u(t+k-d),..u(t+k-d-m)] \end{aligned} \quad (11)$$

Prediction signal $\hat{y}$ enters to equation as output signal after time t-1 does not exist. [10]

D. Adaptive Switching Algorithm

We have proposed a switching between the direct inverse control and the predictive control algorithms. The criterion used for the switching is

$$RY = ((R(t)-Y(t-1))^T(R(t)-Y(t-1)) \quad (12)$$

Predictive control is used where the RY criterion is less than the defined value and direct inverse control is when greater than the defined value.

We have also introduced another term RY_e to overcome the disturbance effects seen on direct inverse control.

$$RY_e = (R(t) - Y_M(t))^T (R(t) - Y_M(t)) \quad (13)$$

Where Y_M is the expected output calculated by the model according to the torque values calculated by neural network of the direct inverse control controller.

If RY_e value is greater than the previous RY value then predictive control is used.

Suitable RY value is found with trials. If the RY value is too small or big switching could not be done at the right time.

III. Implementation Platform

The simulations were carried out on a two joint direct drive SCARA type robot (shown in Fig. 1) which was chosen as implementation platform on earlier work, the parameters and the equations of the process can be found in detail in Denker [14], Efe [9], and Cılız [15]. Robot is modeled by:

$$M(Q)\ddot{Q} + V(Q,\dot{Q}) = \tau - f_c \quad (14)$$

where
M(Q): inertia matrix
V(Q, Q́): Coriolis terms
τ: torque
f_c: friction force

The behavior of the system can be express with four differential equations

$$M(Q) = \begin{bmatrix} p_1 + 2p_3\cos(Q_2) & p_2 + p_3\cos(Q_2) \\ p_2 + p_3\cos(Q_2) & p_2 \end{bmatrix} \quad (15)$$

$$V(Q,\dot{Q}) = \begin{bmatrix} -Q_2(2\dot{Q}_1 + \dot{Q}_2)p_3\sin(Q_2) \\ \dot{Q}_1^2 p_3 \sin(Q_2) \end{bmatrix} \quad (16)$$

p parameters can be expressed as:

$$\begin{aligned} p_1 &= 2.0857 + 0.0576M_p \\ p_2 &= 0.1168 + 0.0576M_p \\ p_3 &= 0.1630 + 0.0862M_p \end{aligned} \quad (17)$$

IV. Simulation Results

In this section, we compare the neural adaptive switching method with the direct inverse control and predictive control methos. Figs. 2, 3, and 4 depict the position and velocity ouputs of base and elbow joints. In direct inverse control neural networks are used as controller directly and in predictive control they are used as the model of the process.

In direct inverse control input of the controller is the error signal; the output of the controller is the torque values. At the same time past torque values and the systems past outputs were used to adapt the neural network. The simulation results are given in Fig. 2.

In predictive control, the inputs of the model are the previous state and the applied torque. The output of the model is the position and the velocity values of the base and elbow joints. The neural network is adapted with the past values.

The controller applies algorithm which optimizes the cost function given by;

$$J = \sum_{k=N_1}^{N_2} (R(t+k) - Y(t+k)) + \rho \sum_{k=1}^{N_u} \Delta u(t+k-1) \quad (18)$$

$$\Delta u(t) = u(t) - u(t-1) \quad (19)$$

$N_1 = 1xT_s$
$N_2 = 4\ xT_s$
$N_u = 2\ xT_s$

T_s: Sampling Period
ρ= 0,008

On the other hand the weighting factors of the all of the state variables are same in the cost function. By changing these weights we can increase the effect of variable or variables on the cost function. Simulation results are given in the Fig. 3.

In the switching control scheme, the speedy characteristics of the direct inverse control are utilized when a sudden change in the reference occurs. After this sudden change is negotiated, we switch to the predictive control algorithm in order to decrease the overshoot. We have also introduced another switching condition to overcome the disturbance effects seen on direct inverse control, whenever output performance starts to detoriates in direct inverse control then transition is made to predictive control. If the RY value is too small or big switching could not be done at the right time. In this application the suitable RY value was found as 0.05. The simulation results for the adaptive switching method are given in Fig. 4.

In direct inverse control it is observed that the response time is smaller but the overshoot value is higher. On the contrary, in predictive control, the response time is higher but the overshoot value is smaller. The main difference can be observed in Fig. 3 that shows the effect of switching between two methods. For this case, the output signals display sudden jumps due to discontinues translation from one to other. It is clear that for the specific example we considered here the neural adaptive switching method is the obvious choice. Compared to the other two this approach presents better performance in following the reference.

Fig. 1 Two Joint SCARA Type Robot

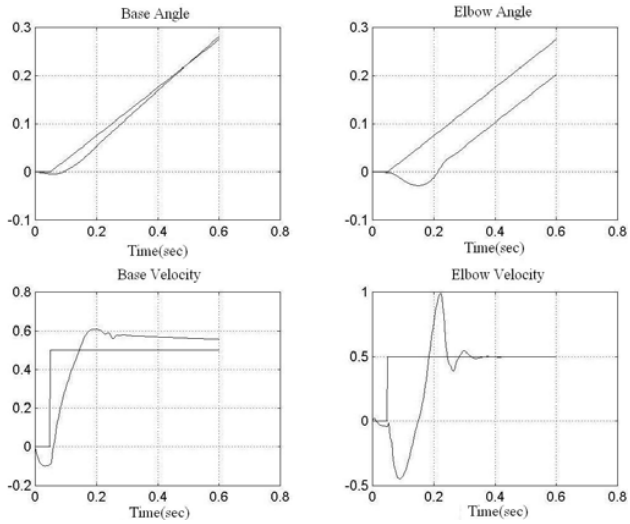

Fig. 2 Direct Inverse Control Application

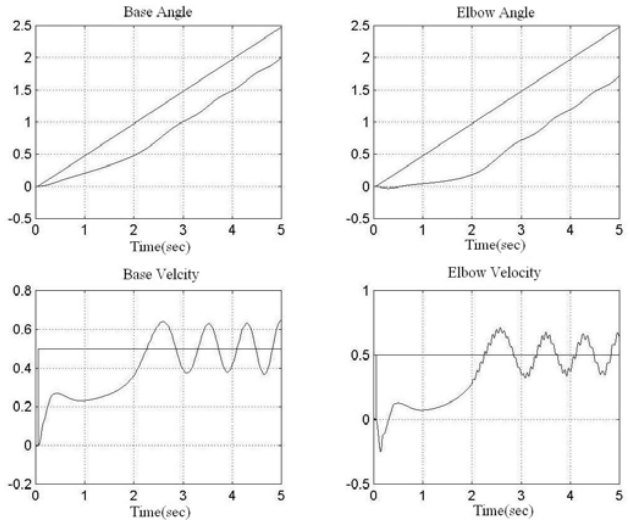

Fig. 3 Predictive Control Application

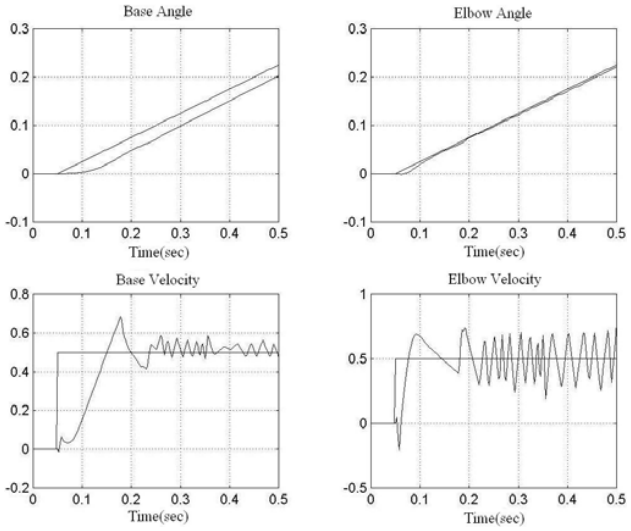

Fig. 4 Switching Application

V. Conclusion

In this paper a neural adaptive controller has been developed and applied to robot control. The results of the simulation experiment show that the performance of the robot system by using the proposed control has more advantages than that by using either pure direct inverse control or predictive control. The control accuracy is much improved over that of using either of the other two methods.

References

[1] Cembrano, G. and Wells, G. 1992. Neural Networks for Control, Boulberg, L. Krıjgsman, A. and Vingehoods, R. A. Application of Artificial Intelligence in Process Control. Pergoman Pres, 388 – 402.

[2] Chen, L. and Narendra K. S. 2001. Nonlinear Adaptive Control Using Neural Netwoks and Multiple Models. Automatica, 1245-1255.

[3] Hunt, K. J. Sbarbaro, D. Zbikowski , R. and Gawthrop, P. J. 1992. Neural Networks for Control Systems – A Survey. Automatica, 28(6) 1083-1112

[4] Cichocki, A. Unbehauen, R. 1993. Neural Networks for Optimization and Signal Processing. WILEY. Chichester .

[5] Freeman, L. A. Skapura, D. M. 1991.Neural Networks Algorithms Applications and Programing Techniques Addison-Wesley.

[6] Noriega, J. R. and Wang, H. 1998. A Direct Adaptive Neural-Network

[7] Efe, M. Ö. ve Kaynak O. 2004. Yapay Sinir Ağları ve Uygulamaları. Boğaziçi Üniversitesi, 148s., İstanbul.

[8] Rivals, I. Personnaz, L. 2000. Nonlinear Internal Model Control sing Neural Networks: Application to Process with Delay and Design Issues, IEEE Transactions on Neural Networks, 11(1) pp 80-90.

[9] Wang, L. Wan, F. 2001. Structured Neural Networks for Constrained Model Predictive Control. Automatica, 1235-1243.

[10] Lazar, M. and Pastavanu, O. 2002. A neural predictive controller for non-linear systems, Mathematics and Computers in Simulation, 60 315-324.

[11] Denker, A. and Ohnishi, K. 1996. Robust Tracking Control of Mechatronic Arms. IEEE/Asme Transactıons on Mechatronıcs. 1(2), 181-188.

[12] Cılız, M. K. 2005. Adaptive Control of Robot Manipulators with Neural Network Based Compensation of Frictional Uncertainties. Robotica, 23, 159-167.

[13] Hagan, M. T. Demuth, H. B. and Beale, M. H. 1996. Neural Network Design, University of Colorado, Colorado.