MMAE540 Robotics- Class Project Paper

Neural Network Controller for Robotic Manipulator

Kai Qian[1]

[1]Department of Biomedical Engineering, Illinois Institute of Technology, Chicago, IL 60616 USA

1. Introduction

Artificial neural network is parallel computation structure inspired from the understanding of biological nervous system(Lippmann, 1987). It consists of many interconnected computational element through weights which keep adapting to achieve better system response. Two major capacities of neural network are classification (Ripley, 1994), a simple example is just perceptron, and function fitting (can map from space $R^m toR^n$), such as multilayer neural network. It has been shown a two-layer networks with sigmoid function activation function for hidden layer and linear activation function for output layer, can approximate any continuous function to any degree of accuracy with sufficient large number of hidden layer nodes(Hornik et al., 1989). Based on its universal approximation capacity neural network has been naturally and successfully applied to identification and control of complex system dynamics(Hagan et al., 2002).

For robotic manipulator control, the adaptive controller(Slotine and Li, 1991) we studied in class has shown impressive asymptotic trajectory tracking performance in face of parameters uncertainty such as loads changes. Without knowing the mass, link length and load information, the controller estimates those parameters and makes them converge to its real value, therefore provide perfect estimation of system dynamic to achieve the great system performance. The adaptive control scheme(Slotine and Weiping Li, 1987) is shown in figure 1.

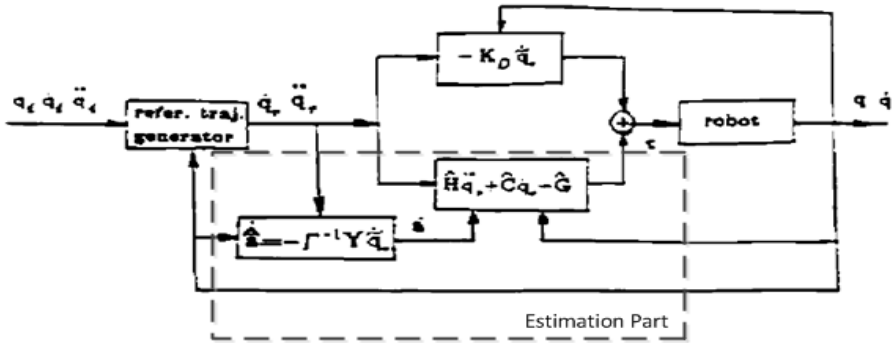


Figure 1. Adaptive control scheme for robotic manipulator(Slotine and Weiping Li, 1987).

In deriving adaptive controller, it requires substantial work to compute the Y matrix, such that $Y(q,\dot{q},\dot{q}_r,\ddot{q}_r)\hat{a} = \hat{H}(q)\ddot{q}_r + \hat{C}(q,\dot{q})\dot{q}_r + G(q)$. System performance depends on the accuracy of regression matrix Y and the knowledge of complete system dynamics. However the estimation of the adaptive controller reminded me the function approximation ability of neural network. It can give great estimation only by training on input and output. If a neural network can replace the role of Y matrix, then good system performance will be achieved as adaptive controller without preliminary dynamics analysis to compute Y. This is also the motivation for this class project paper. From the literature search, the paper(Lewis et al., 1996) was about the same objective with formal stability proof of the neural network controller. The control scheme was shown in figure 2.

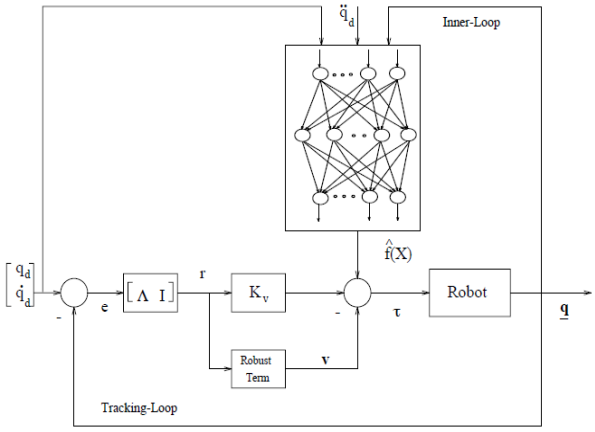


Figure 2. Neural network control Scheme for robotic manipulator(Lewis et al., 1996)

So, the neural network controller developed in this paper was based on this paper. The detailed derivation was given in section 2 method part. And a 2-link manipulator tracking task simulation was used to demonstrate the performance of the neural network controller. The properties and some considerations on above neural network controller was followed in final discussion part.

2. Method

2.1. Neural Network

A three layer neural network as figure 3 with sigmoid activation function for input layer and linear activation function for output layer is defined by

$$y = W^T \sigma(V^T x) \tag{1}$$

$$\sigma(z) = \frac{1}{1+e^{\alpha z}} \qquad \text{sigmoid}$$

where V is NN input layer weights matrix, and W is output layer weights matrix. Output vector y is in m dimensions, which is also corresponding to the output layer neuron number N3, N3=m, and input layer vector x is in n dimensions, input layer number N1 = n, A general function f(x) can be written as

$$f(x) = W^T \sigma(V^T x) + \varepsilon \tag{2}$$

ε is NN function reconstruction error. Since any continuous smooth function can be approximated by a large multilayer net based on various activation function, such as sigmoid and radial basis functions(Cybenko, 1989, Hornik et al., 1989). There exist finite hidden neuron number N2 , W and V to make the function reconstruction error ε very small. $f(x)$ will provide a best fit for target function.

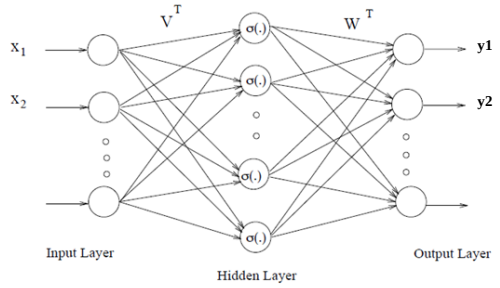


Figure 3. Three Layers Neural Network Structure

2.2 Robotic Manipulator Dynamics

A general n-link robotic manipulator dynamic equation is given by

$$M(q)\ddot{q} + V_m(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d = \tau \tag{3}$$

$$e(t) = q_d(t) - q(t) \tag{4}$$

$$r = \dot{e} + \Lambda e \tag{5}$$

$q_d(t)$ is the desired trajectory input, e(t) is the tracking error and r(t) is the filtered tracking error. And rearrange the system equation (3) with (4) (5), the manipulator dynamic equation can be expressed in term of filtered tracking error r as

$$M\dot{r} = -V_m r - \tau + f + \tau_d \tag{6}$$

$$f(x) = M(q)(\ddot{q}_d + \Lambda e) + V_m(q,\dot{q})(\dot{q}_d + \Lambda e) + G(q) + F(\dot{q}) \tag{7}$$

$$x = [e^T, \dot{e}^T, {q_d}^T, {\dot{q}_d}^T, {\ddot{q}_d}^T]^T \tag{8}$$

We choose control torque input to be

$$\tau = \hat{f} + K_v r \tag{9}$$

where $\hat{f}$ is an estimation of f, then the system dynamic equation in term of filtered tracking error can be written as

$$M\dot{r} = -(K_v + V_m)r + \tilde{f} + \tau_d \tag{10}$$

$$\tilde{f} = f - \hat{f} \tag{11}$$

Based the feedback filtered tracking error dynamic equation (10), if function estimates error of f(t) is small and system disturbance is small, then a good tracking performance will be achieved(Dawson et al.,

1998). So, the essential part of the neural network controller in this paper is using neural network to approximate function f(t) to achieve better tracking performance.

2.3 Neural Network Controller A

Given neural network estimation of f as in (7) by

$$\hat{f}(x) = \hat{\hat{W}}^T \sigma(V^T x) \tag{12}$$

where $\hat{W}$ and $\hat{V}$ are weights estimates and let W and V be the "ideal weights" with which the neural network reconstruction error will be minimal.

Assume ideal weights are bounded, desired trajectory input is bounded and the input vector x to neural network is bounded.

$$\|Z\|_F \le Z_M, \|Z\|_F^2 = trace(Z^T Z) \tag{13}$$

$$\left\| \begin{matrix} q_d \\ \dot{q}_d \\ \ddot{q}_d \end{matrix} \right\| \le Q_d \tag{14}$$

$$\|x\| \le c_1 Q_d + c_2 \|r\| \tag{15}$$

c_1, c_2, Q_d, Z_m are constants. Define the hidden-layer output error by

$$\tilde{\sigma} = \sigma - \hat{\sigma} = \sigma(V^T x) - \sigma(\hat{V}^T x) \tag{16}$$

and with Taylor series expansion expressed by

$$\sigma(V^T x) = \sigma(\hat{\hat{V}}^T x) + \sigma'(V^T x)\tilde{V}^T x + \mathrm{O}(\tilde{V}^T x)^2 \tag{17}$$

Since the activation function is sigmoid function, the higher order term in (17) is bounded by

$$\left\| \mathrm{O}(\tilde{V}^T x)^2 \right\| \le c_3 + c_4 Q_d \left\| \tilde{V} \right\|_F + c_5 \left\| \tilde{V} \right\|_F \|r\| \tag{18}$$

Let control input be

$$\tau = \hat{\hat{W}}^T \sigma(V^T x) + K_v r - v \tag{19}$$

Substitute (12)(17)(19) into (10) finally we will get

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}^T \hat{\sigma} + \hat{W}^T \sigma' \tilde{V}^T x + w_1 + v \tag{20}$$

$$w_1 = \tilde{W}^T \hat{\sigma}' \tilde{V}^T x + W^T \mathrm{O}(\tilde{V}^T x)^2 + (\varepsilon + \tau_d) \tag{21}$$

w_1 is the disturbance term containing neural network reconstruction error, higher order term in Taylor expansion of sigmoid function, and system disturbance.

Assume w_1 and v is zero, given the weight updated law by

$$\dot{\hat{W}} = F\hat{\sigma}r^T$$
$$\dot{\hat{V}} = Gx(diag(\hat{\sigma}')W^T r)^T \tag{22}$$

F and G are positive definite Matrice. And choose Lyapunov function to be

$$L = \frac{1}{2}r^T Mr + \frac{1}{2}trace(\tilde{W}^T F^{-1}\tilde{W}) + \frac{1}{2}trace(\tilde{V}^T G^{-1}\tilde{V}) \tag{23}$$

$$\dot{L} = r^T M\dot{r} + \frac{1}{2}r^T \dot{M}r + trace(\tilde{W}^T F^{-1}\dot{\tilde{W}}) + trace(\tilde{V}^T G^{-1}\dot{\tilde{V}}) \tag{24}$$

Using (20) and let w1 and v to be zero, with the weights updating law, finally we will get

$$\dot{L} = -r^T K_v r \le 0 \tag{25}$$

As t increases, tracking error will approach zero. Formula (19) and (22) defined the Neural Controller A, however, the error tracking performance is based on three strong assumptions 1) no neural network estimation error 2) No unmodeled disturbance 3) No higher-order Taylor series term. Furthermore, no information is provided on weights updating stability. These limitations make the neural network controller A less appealing.

Neural controller B was proposed to overcome above limitations with weights updating law by

$$\dot{\hat{W}} = F\hat{\sigma}\hat{r}^T - Fdiag(\sigma')V^T xr^T - \kappa F\|r\|W$$
$$\dot{\hat{V}} = Gx(diag(\hat{\sigma}')W^T r)^T - \kappa G\|r\|V \tag{26}$$

where κ is constant. Add one robustifying term v(t)

$$v(t) = -K_z(\|\hat{Z}\|_F + Z_M)r \tag{27}$$

K_z is another design constant. Let the control law of neural controller B be

$$\tau = \hat{W}^T\sigma(V^T x) + K_v r - v \tag{28}$$

Choosing the same Lyapunov function as (23), differentiating and substituting system dynamic equation (20) without assuming w1 = 0, yields

$$\begin{aligned}\dot{L} = &-r^T K_v r \\ &+\frac{1}{2}r^T(\dot{M} - 2V_m)r + trace\tilde{W}^T(F^{-1}\dot{\tilde{W}} + \hat{\sigma}\hat{r}^T - \sigma'\hat{V}^T xr^T) \\ &+\text{trace}\tilde{V}^T(G^{-1}\dot{\tilde{V}} + xr^T\hat{W}^T\hat{\sigma}') + r^T(w+v)\end{aligned} \tag{29}$$

Through several inequality equations, it gives

$$\dot{L} \le -\|r\|[K_{v\min}\|r\| + \kappa\|\tilde{Z}\|_F(\|\tilde{Z}\|_F - Z_M) - C_0 - C_1\|\tilde{Z}\|_F] \tag{30}$$

$$\begin{aligned} &if \quad [K_{v\min}\|r\| + \kappa\left\|\tilde{Z}\right\|_F(\left\|\tilde{Z}\right\|_F - Z_M) - C_0 - C_1\left\|\tilde{Z}\right\|_F] \geq 0 \\ &then \quad \dot{L} \leq 0 \end{aligned} \tag{31}$$

Rearranging (31),

$$\|r\| > \frac{\kappa C_3^2/4 + C_0}{K_{v\min}} \equiv b_r \tag{32}$$

Or

$$\left\|\tilde{Z}\right\|_F > C_3/2 + \sqrt{C_3^2/4 + C_0/\kappa} \equiv b_z \tag{33}$$

where the $K_{v\min}$ is the minimal element in the diagonal gain matrix, and constants in (32)(33) are from bounded input, bounded ideal weight, and bonded higher order term in σ function Taylor series expansion assumptions as in (14)(15)(18) . (32)(33)(31) state that based on neural controller B (defined by (26)(27(28)), $\dot{L}$ is negative outside a compact set. If tracking error is outside of the compact set, $\dot{L}$ will become less than zero, system energy decreases and tends to drive $\|r\|$ back to the compact set. And the same explanation applies to the weights matrix. Therefore, neural controller B will guaranty the tracking error is bounded and weights updating is bounded. And the compact set range or tracking error range is adjustable by changing gain K_v and those bounded constants as in (32)(33).

Since neural network controller B can provide bounded tracking error and weights tuning, it was adopted for the following simulation study section.

3. Simulation Results

Neural network controller B was implemented and simulated for a 2-link planar arm that is the same as homework 6, with m1 = m2 = 5, l1 = l2 = 1, lc1=lc2=0.5 as shown in figure 4.

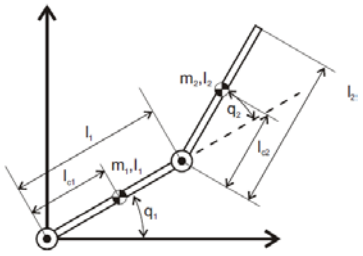


Figure 4. two-link planar

3.1 Neural Network Controller B vs. PD controller

The task is joints angle tracking as the desired trajectory given by $q_d1 = 2\sin(2t), q_d2 = \sin(t)$, with initial condition $q_1(0) = -45^\circ, q_2(0) = 45^\circ$. The tracking performance of neural controller B is shown in figure 5. The parameters for the controller are Kv = 20, Kz = 0.2, Zm = 400, Λ=[5 0;0 5], Initial Weights W , V sets to zero, F=diag(100*ones(80,1)), G=diag(100*ones(11,1)), neural network input layer neuron number N1=11, as the input vector given by

$$x = [1, e^T, \dot{e}^T, {q_d}^T, {\dot{q}_d}^T, {\ddot{q}_d}^T]^T \tag{34}$$

Constant 1 in x vector is corresponding to the threshold vector which is the first column of weights matrix V. As shown in figure 5, neural network controller B achieved good performance for the tracking task; only very small tracking error was observed (less than 0.08 rad).

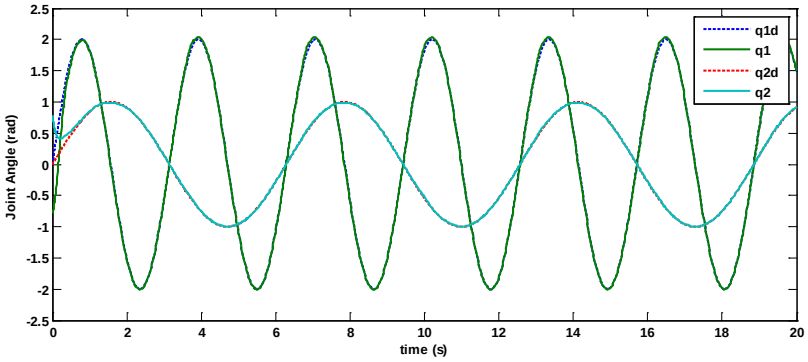


Figure 5. Response of Neural Controller B

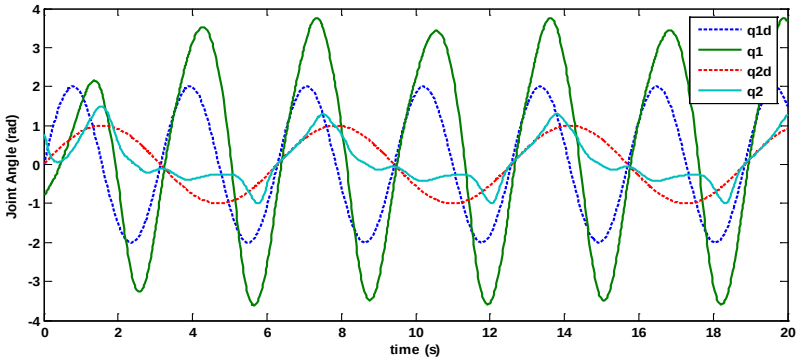


Figure 6. Response of PD Controller without Neural Network Part

For PD controller, the control torque input generated by Neural Network part is removed. All the other parameters left unchanged. The system output was shown in figure 6. The system tracking performance degrades a lot. Large tracking error and phase shift were observed. When PD controller gains Kv increased from 20 to 240, the system response was as in figure 7. Tracking error still could be observed clearly from the figure 7. Control torque input graphs for neural network controller B and PD controller with large gain were shown in figure 8. Oscillations were observed in torques generated by neural network controller B. Magnitude of the torque generated by PD controller was larger than that of neural network controller B.

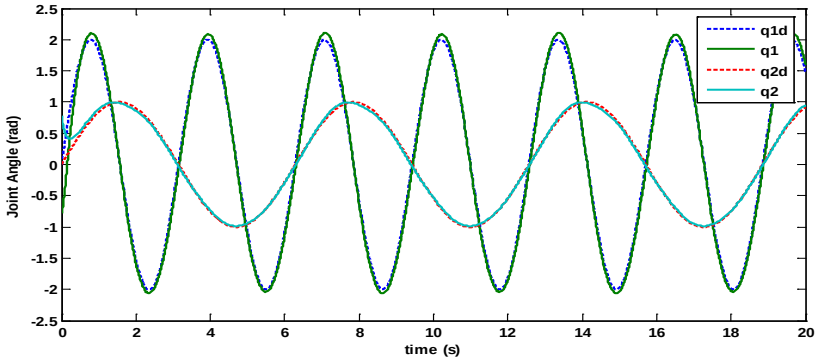
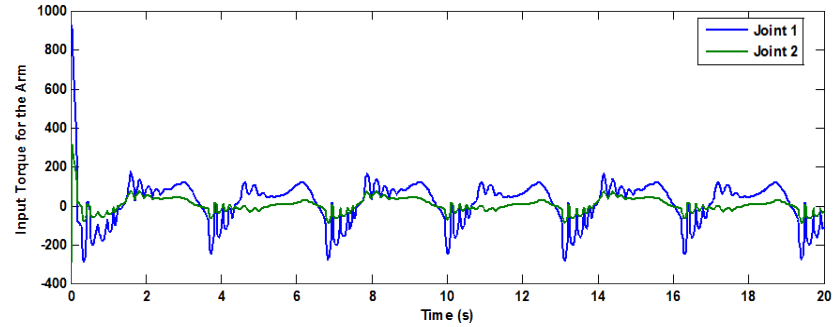
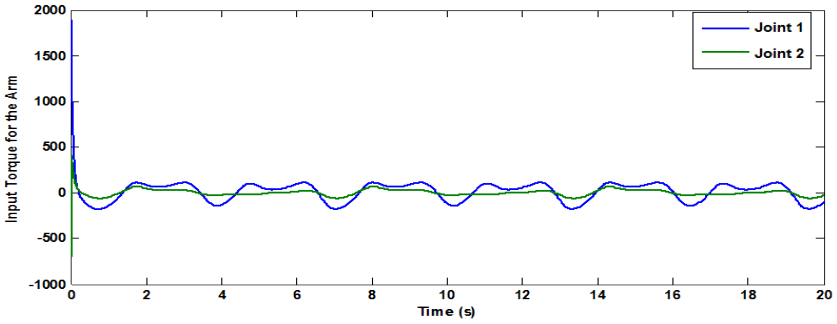


Figure 7. Response of PD Controller with large PD gains (Kv=240)



a) Torque generated by neural network controller B



b) Torque generated by PD controller with large gain

Figure 8. Torque generated by neural network controller B and PD controller with large gain

3.2 Comparison to adaptive controller

The adaptive controller used here was the same as in homework 6 with the following parameters, Kd = 20, Lambda = diag([5,5]) P=diag([4, 2, 1, 20, 8] . The response of the adaptive controller was shown in figure 9. The a vector estimation was as figure 10.

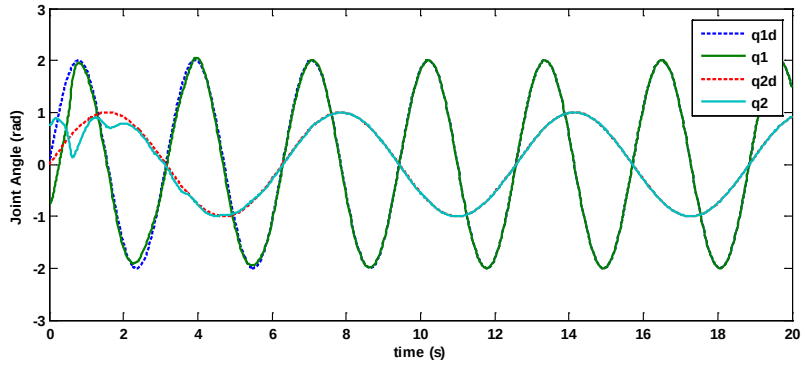


Figure 9. Response of Adaptive controller

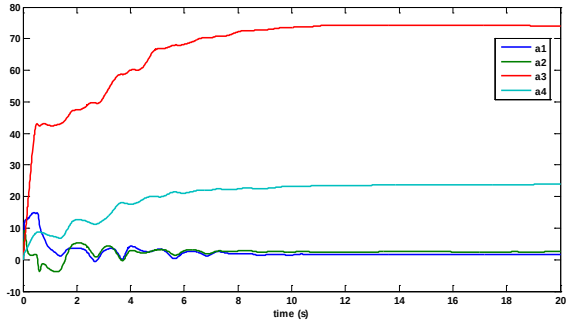


Figure 10. Estimation of parameter vector a in adaptive controller

The tracking performance was perfect for adaptive controller as shown figure 9. To model the unmodeled dynamics effect of adaptive controller, the element of y(1,4)=cos(q(1)) in adaptive controller Y matrix was set to zero. The system output was then shown in figure 11. Significant tracking performance degradation was observed from the figure 11.

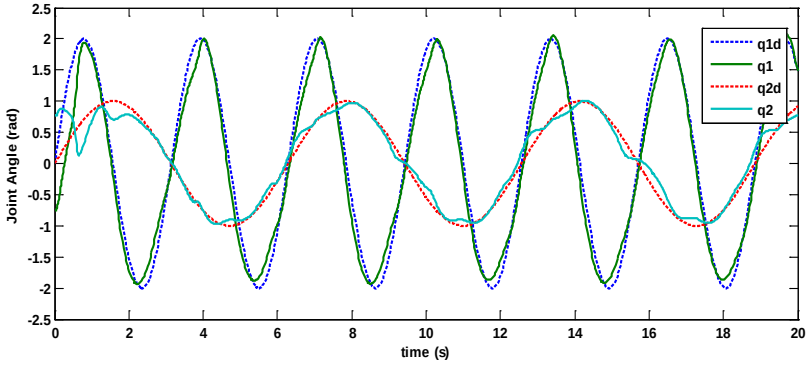


Figure 11. Response of adaptive controller with unmodeled system dynamics

3.3 Neural Network Controller B with less number of nodes in hidden layer

The neural network controller B with 80 nodes in hidden layer gave good tracking performance as figure 5. To reduce the number of nodes in hidden layer will increase the estimation error of the neural network, which then increases the bound of the tracking error as (32). A neural network controller with the same parameters as in section 3.1 but the number of nodes in hidden layer was reduced to 20. The response was shown in figure 12. As expected, larger tracking error was observed compared with figure 5.

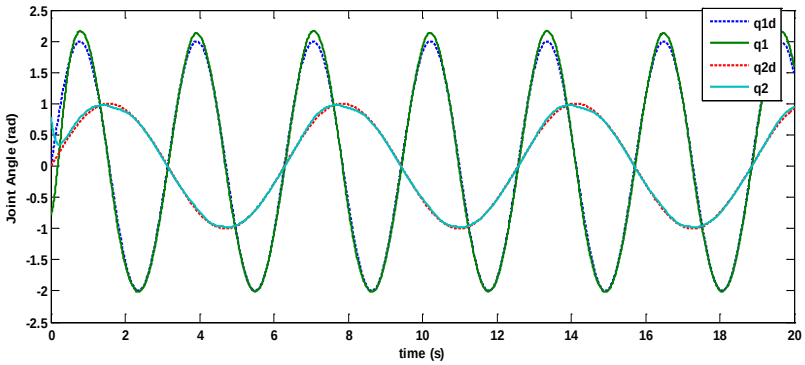


Figure 12. Neural Network Controller B with fewer nodes in hidden layer

3.4 Neural Network Controller A

Neural network controller A based on (19)(22) with w1(t)=0,v(t)=0, Kv = 20, Λ=[5 0;0 5], Initial Weights W , V sets to zero, F=diag(100*ones(80,1)), G=diag(100*ones(11,1)) was also simulated to compare with NN controller B. The response and torque plot were as figure 13 and figure 14. Since the strong assumptions of neural network controller A are easily violated, the tracking performance and stability cannot be guaranteed, just as shown in figure 13 and 14, the tracking performance was very poor and very large torque spikes.

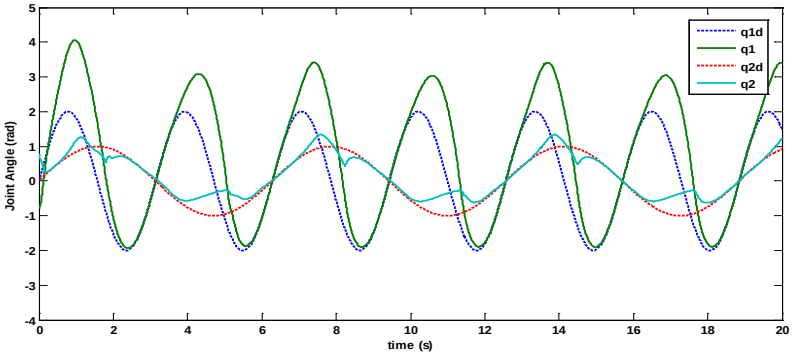


Figure 13. Response of neural network controller A

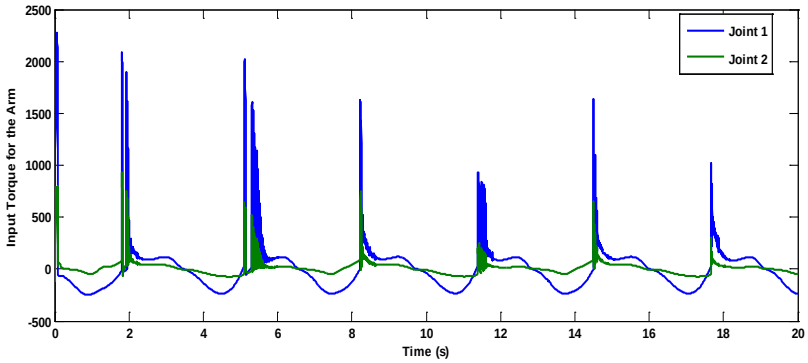


Figure 14. Torque generated by network controller A

4. Discussion

The neural network controller B provided good tracking performance which is comparable to the output by adaptive controller, while no preliminary analysis of system dynamics to derive the controller is required. With bounded input and sufficient large net, the system tracking error and weights will be bounded by neural controller B.

The control torque generated by neural controller was not very smooth compared with PD controller and Adaptive controller. Since weights were only proofed to be bounded, not to exponentially converge. It may keep adjusting weights to keep the tracking error to be in the small compact set as in (33). The oscillation of control signal by adaptive neural network controller was also mention and shown in (Cao and Hovakimyan, 2007).

The neural network learning rules developed in(Lewis et al., 1996) as (22)(26) were not the standard training or learning rule for multilayer neural network as in(Gurney and Gurney, 1997). In (22), weights update did the backpropagation part, not backpropagated the neural network estimation errors ($f - \hat{f}$), but the errors were tracking error $\|r\|$, since the real value of f was unknown. It follows into unsupervised learning. The author (Lewis et al., 1996) changed the name of (22) 'standard backpropagation rule' to 'unsupervised backpropagation rule in his later book(Lewis et al., 1999). It also gave a hint in applying neural network to control system. Bring in the matrix expression of the neural network, assuming it gives ideal fit for certain part of control system to be estimated, the weights updated law are then derived based on carefully selected Lyapunov function. The stability conditions also followed.

Two drawbacks of neural network controller also raised in implementing the controller. 1) too many parameters to adjust to guaranty the performance of NN controller, such as Kv, Kz , Zm , Λ, F, G and

neural network hidden nodes number N2. All these add to the complex of the controller in trading off the benefit from no preliminary system analysis as in adaptive controller which has fewer parameters to adjust. Although 10 hidden layer nodes were used in simulation examples in (Lewis et al., 1996) , but the dimension of its G and F matrix did not match the weights updating law (22)(26). 2) Computation load. Since the hidden layer nodes number is large, this then corresponds to a large weight matrix, and each element in it has to be updated each step. It seems impossible to implement it in realtime control situation.

References

CAO, C. & HOVAKIMYAN, N. 2007. Novel L1 neural network adaptive control architecture with guaranteed transient performance. *IEEE Trans Neural Netw,* 18**,** 1160-71.

CYBENKO, G. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS),* 2**,** 303-314.

DAWSON, D. M., HU, J. & BURG, T. C. 1998. *Nonlinear control of electric machinery*, Dekker.

GURNEY, K. & GURNEY, K. N. 1997. *An introduction to neural networks*, UCL Press.

HAGAN, M. T., DEMUTH, H. B. & JESÚS, O. D. 2002. An introduction to the use of neural networks in control systems. *International Journal of Robust and Nonlinear Control,* 12**,** 959-985.

HORNIK, K., STINCHCOMBE, M. & WHITE, H. 1989. Multilayer feedforward networks are universal approximators. *Neural Netw.,* 2**,** 359-366.

LEWIS, F. L., JAGANNATHAN, S. & YEŞILDIREK, A. 1999. *Neural network control of robot manipulators and nonlinear systems*, Taylor & Francis.

LEWIS, F. L., YEGILDIREK, A. & LIU, K. 1996. Multilayer neural-net robot controller with guaranteed tracking performance. *IEEE Trans Neural Netw,* 7**,** 388-99.

LIPPMANN, R. 1987. An introduction to computing with neural nets. *ASSP Magazine, IEEE,* 4**,** 4-22.

RIPLEY, B. D. 1994. Neural Networks and Related Methods for Classification. *Journal of the Royal Statistical Society. Series B (Methodological),* 56**,** 409-456.

SLOTINE, J.-J. E. & WEIPING LI 1987. On the Adaptive Control of Robot Manipulators. *The International Journal of Robotics Research,* 6**,** 49-59.

SLOTINE, J. J. E. & LI, W. 1991. *Applied nonlinear control*, Prentice Hall.