NON-LINEAR DYNAMICAL SYSTEMS CONTROL USING NEURAL NETWORKS

Nguyễn Việt Hùng
maihung@hcmc.netnam.vn
Faculty of Electronics Engineering, HCMUTE

Đậu Trọng Hiển
dautronghien@yahoo.com
Faculty of Electronics Engineering, HCMUTE

Trần Thanh Mai
ttmai@hcmut.edu.vn
Faculty of E & Electronics Engineering, HCMUT

ABSTRACT

Non-linear dynamical systems are difficult to control due to the model uncertainties and external disturbances that may occur in these systems. This paper addresses the problem of modeling and on-line direct inverse controlling method using neural networks for a given class of nonlinear systems. The design procedure involves the integration of identification and neural control for an imprecisely known plant. A new recurrent neural network approach for on-line adaptive control is presented. A dynamic learning algorithm for the recurrent neutral network has been developed with faster convergence and improved steady-state performance for two neural networks structure. On-line simulation studies for selected process with the proposed control are presented and discussed using two-link robot arm to demonstrate the performance of the proposed strategy.

1. INTRODUCTION

There are several ways to cope with nonlinear control design for plants subject to uncertainty and disturbances; among them are robust and adaptive control methods. While neural adaptive control is being intensively developed [1], [3], [4], so as the applications of neural networks to nonlinear control. For example, Narendra et al. [2], [7] introduced multilayer neural networks for identification and adaptive control of nonlinear systems. A number of studies such as [6], [10] for adaptive control of unknown feedback linearizable system for achieving guaranteed performance of the neural network has excellent capabilities of nonlinear mapping, learning ability, and parallel computations [5], [12].

In this paper, a general framework for neural controller design is introduced using a new method for cost function implementation, to derive a learning scheme for the neural-based modeling and controlling networks. The control of a system with unknown dynamical and partially know states is accomplished using dynamical neural networks. Novel updating algorithm is developed. On-line tracking performance and robustness properties are proven with performance verified through simulation examples.

2. PROBLEM FORMULATION

2.1 Notation

Definition 1:

Ordered derivative: $\partial^*(.)/\partial v$ a special notation is used to distinguish between two forms of partial derivative [9], [11]. For any variable v, $\partial^* / \partial v$ account for the direct and indirect effects on the interaction between the other variables.

Assumptions 1

Given a positive constant ε_0 and a continuous function $f : C \to R$, where $C \subset R^{m_r}$ is a compact set, there exists a weight matrix $W = W^*$ and a positive integer n^* such that the output $\hat{f}(x,W)$ of the MLP recurrent neural network with n^* nodes satisfies

$$\max_{x\in C}\left\| \hat{f}(u,W^*) - f(x)\right\| \le \varepsilon_0$$

where u: the inputs and n^* may depend on precision parameter ε_0 and the function f

Assumptions 2

The output $\hat{f}(u,W)$ of the MLP recurrent neural network is continuous with respect to its arguments for all finite (u,W)

2.2 Objectives

The control objective: given a desired output $y_d(t)$, find a control u such the output of the system tracks the desired trajectory with an acceptable accuracy, while all the states and control remain bounded.

In this paper, we investigate on the formulation of a new type of cost function, to derive the training rules for the neural-based modeling (Fig. 2.1) and neural-based controller (Fig. 2.2) using the identification error, defined by $e_I = y - \hat{y}$, and the tracking error $e_C = y - d$

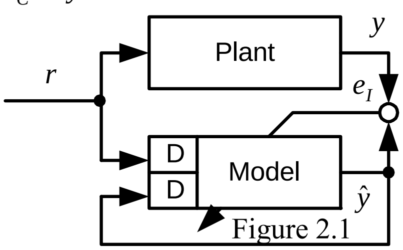


Figure 2.1

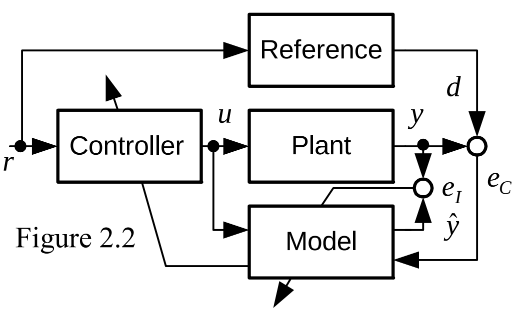


Figure 2.2

Proposition 1:

In a reference neural control scheme using recurrent multi-layer perceptron (MLP), satisfied the Assumption 1 and 2, then the information from control environment can be introduced to the neural networks training process by formulating the cost function with two terms:

$J = J_1 + J_2$

$J_1 = J(e)$: error cost function

$J_2 = J(u,W)$: tuning function, related to the conditions of control environment and the weight regulation

Proof:

Consider the system controlled with the desired control behavior is specified by a cost function J over the interval $[0,T]$: $J = \int_0^T l_C(x(t), y(t), u(t), r(t), t, T)dt$ in which $x(t)$: the state vector, $y(t)$: output vector, $u(t)$: input vector, $r(t)$: a reference vector. Discretize the system dynamics, and then we have $J = \sum_{k=0}^{N} l(x(k), y(k), r(k), k, N)$. Using the definition of ordered derivative, then:

$$\frac{\partial^* J}{\partial x_i(k)} = \frac{\partial l(k)}{\partial x_i(k)} + \sum_{q=1}^{p} \frac{\partial l(k)}{\partial y_q(k)} \frac{\partial^* y_q(k)}{\partial x_i(k)} + \sum_{q=1}^{n} \frac{\partial J}{\partial x_q(k+1)} \frac{\partial^* x_q(k+1)}{\partial x_i(k)};$$
$$k = N-1,...,1 \quad i = 1,..,n \qquad \text{(P.1)}$$

$$\frac{\partial^* J}{\partial u_j(k)} = \frac{\partial l(k)}{\partial u_j(k)} + \sum_{q=1}^{p} \frac{\partial l(k)}{\partial y_q(k)} \frac{\partial^* y_q(k)}{\partial u_j(k)} + \sum_{q=1}^{n} \frac{\partial J}{\partial x_q(k+1)} \frac{\partial^* x_q(k+1)}{\partial u_j(k)};$$
$$k = N-1,...,1\,;\, i = 1,..,n \qquad \text{(P.2)}$$

The equations above shows the direct and indirect effects on the interaction between the other variables, and suggest that we should formulate:

The cost function $J = J_1 + J_2$ (P.3)

In which: $J_1 = \sum_{k=0}^{N} l_1(x(k), y(k), r(k), k, N)$

and $J_2 = \sum_{k=0}^{N} l_2(x(k), y(k), r(k), k, N)$

and when using the ordered derivative notion for J in (P.3), the proposition 1 is proven □

Assumption 3:

The weights sub-optimal value w_j^{lc} which is assumed to be stuck in to local minima, can be defined from the statistical methods [10]

Lemma 1:

The neural-based identification networks, using Figure 2.1, satisfied the proposition 1 and under the assumptions A1-A4., the weights are adjusted as follow:

$$\Delta W_I(k) = 2\mu e^T(k)\left[\frac{\partial y(k)}{\partial W} + \nabla_X Y(k).\nabla_W Y(k)\right]^T + \mu\nabla_W J_2(W,k)$$

in which $J_2 = J_2(W) = p(W)$

and $W_I(k+1) = W_I(k) + \Delta W_I^T(k)$

With a proper choice of: learning rate $\mu < \mu_M$ and weight regulating term $\eta < \eta_M$, then the identification error converges and has local minima-free.

Proof:

<u>Derivation of the training rule</u>: from Fig. 2.1, then:

$$y(k) = f(x(k), x(k-1), \ldots, x(k-n), y(k-1), \ldots, y(k-m), W) \quad \text{(L1.1)}$$

$x(k)$: input vector, $y(k)$: output vector, W: the weights of the identification network

With

$$J(k) = J_1(k) + J_2(W) = \left\{e_j^T(k)e_j(k) + J_2(W)\right\} \quad \text{(L1. 2)}$$

Using gradient descent method, then

$$W(k+1) = W(k) - \eta\frac{\partial J^T(k)}{\partial W} \quad \text{(L1.3)}$$

$$\nabla *_W J = \frac{\partial * (J_1 + J_2(W,k))}{\partial W} = -2e^T(k)\frac{\partial * y(k)}{\partial W} + \frac{\partial * J_2(W,k)}{\partial W} \quad \text{(L1.4)}$$

After some arrangements, then:

$$\frac{\partial * J_2(W,k)}{\partial W} = \frac{\partial J_2(W,k)}{\partial W} + \sum_{j=0}^{h}\frac{\partial J_2(W,k)}{\partial x(k-j)}\frac{\partial * x(k-j)}{\partial W} + \sum_{j=1}^{n}\frac{\partial J_2(W,k)}{\partial y(k-j)}\frac{\partial * y(k-j)}{\partial W} \quad \text{(L1.5)}$$

$$\Delta W(k) = -\mu\nabla *_W J(k) = 2\mu e^T(k)\left[\frac{\partial y(k)}{\partial W} + \nabla_X Y(k).\nabla_W Y(k)\right]^T + \mu\frac{\partial J_2(W,k)}{\partial W} \quad \text{(L1.6)}$$

Finally

$$\Delta W(k) = 2\mu e^T(k)\left[\frac{\partial y(k)}{\partial W} + \nabla_X Y(k).\nabla_W Y(k)\right]^T + \mu\nabla_W J_2(W,k) \quad \text{(L1.7)}$$

In which, the gradient vector $\nabla_W Y(k)$ and $\nabla_X Y(k)$ are:

$$\nabla_W Y(k) = \left[\left(\frac{\partial^d y(k-1)}{\partial W}\right)^T \quad \cdots \quad \left(\frac{\partial y^d(k-n)}{\partial W}\right)^T\right]^T$$

$$\nabla_X Y(k) = \left[\left(\frac{\partial y(k)}{\partial y(k-1)}\right) \quad \cdots \quad \left(\frac{\partial y(k)}{\partial y(k-m)}\right)\right]$$

<u>Convergence and the proper choice of μ</u>

Chose the Lyapunov function as

$$V(k) = \frac{1}{2}e^T(k)e(k) \quad \text{(L1.8)}$$

The error in the MLP neural networks is:

$$e_i(k) = \frac{-\delta_i(k)}{f'(s_i(k))} \quad \text{(L1.9)}$$

In which:

$\delta_i(k)$: Equivalent sensitivity of the output with respect to weight; $X_i(k)$: inputs and $f'(s_i(k))$: derivative of output activation soft function

$$\Delta e(k) = e(k+1) - e(k) \cong \cong \mu_i f'(s_i(k))\delta_i(k)\|X_i(k)\|^2 \quad \text{(L1.10)}$$

$$\Delta V(k) = V(k+1) - V(k) = = \mu_i\delta_i^2(k)\|X_i(k)\|^2\left(1 + \frac{1}{2}\mu_i f'(s_i(k))^2\|X_{i,}(k)\|^2\right) \quad \text{(L1.11)}$$

$\Delta V(k) < 0$, then

$$0 < \mu < \mu_M = \frac{2}{[f'(s(k))]^2 \|X(k)\|^2} \quad \text{(L1.12)}$$

Local minima-free and the choice of $\lambda < \lambda_M$

Chose the weight regulation function as

$$J_2(W) = \eta \sum_{j=1}^{n} \sum_{i=1}^{m} exp\left(- \frac{\left| w_{ji}(n-1) - w_j^{lc} \right|^2}{\eta^2} \right) \quad \text{(L1.13)}$$

$$\frac{\partial J}{\partial W} \le 0$$

$$\frac{\partial J_1}{\partial W} - \frac{2}{\eta} \left| w_{ji} - w_{ji}^{lc} \right| exp\left(- \frac{(w_{ji}(n-1) - w_{ji}^{lc})^T (w_{ji}(n-1) - w_{ji}^{lc})}{\eta^2} \right) \le 0 \quad \text{(L1.14)}$$

$$\frac{\partial J_1}{\partial W} \ge \frac{2}{\eta} \left| w_{ji} - w_{ji}^{lc} \right| exp\left(- \frac{(w_{ji}(n-1) - w_{ji}^{lc})^T (w_{ji}(n-1) - w_{ji}^{lc})}{\eta^2} \right) \quad \text{(L1.15)}$$

Assume that $|\eta| > 0$ and is a quite large number, so $\frac{1}{\eta^2} \to 0$, $\Rightarrow \frac{2}{\eta_M} \left| w_{ji} - w_{ji}^{lc} \right| \ge \frac{\partial J_1}{\partial W}$,

$$\Rightarrow \quad \eta_M \le 2 \left| w_{ji} - w_{ji}^{lc} \right| . \frac{1}{\frac{\partial J_1}{\partial W}} \quad \text{(L1.16)}$$

□

Assumption 4:

The control inputs u of the MLP recurrent neural network are bounded and be continuous with respect to its arguments for all finite (u, W)

Lemma 2:

The neural-based controller, using Figure 2.2, satisfied the proposition 1, under the assumptions A1-A4., the weights are adjusted as follow:

$$\Delta W^T(k) = 2\mu e^T(k) Q.(\nabla *_W Y(k)) - \mu.(\nabla_U J_{hc}^T(k))(\nabla *_W U(k))$$

and $W(k+1) = W(k) + \Delta W^T(k)$

$J_2 = J_2(u)$: The tuning function, related to the conditions of control environment, Q: weighting factor case of MIMO system.

With the proper choice of control parameter, then the tracking error converges

Proof:

Derivation of the training rule

$$J(k) = J_1 + J_2(u, W) = J_1 + J_2(u) \quad \text{(L2.1)}$$

From Fig. 2.1,

$$e_I(k) = d(k) - y(k) \quad \text{(L2.2)}$$

$$u(k) = f_C(u(k),..,u(k-m), r(k),...,r(k-q), W_C)$$

$$y(k) = f_I(y(k-1),...,y(k-n), u(k),..,u(k-p), W_I)$$

The model is matching, then $y(k) \cong \hat{y}(k)$

$$y(k) = f_I(y(k-1),...,y(k-n), u(k)...,u(k-p), W_I)$$

$$\frac{\Delta W^T(k)}{\mu} = 2e^T(k)\left(\frac{\partial * y(k)}{\partial W_C} \right) - \left[\sum_{j=0}^{r} \left(\frac{\partial * J_2(u)}{\partial u(k-j)} \right)^T \right] \left(\frac{\partial * u(k-j)}{\partial W_C} \right) \quad \text{(L2.3)}$$

$$\frac{\partial * u(k)}{\partial W} = \frac{\partial u(k)}{\partial W} + \sum_{j=1}^{m} \frac{\partial u(k)}{\partial u(k-j)} \frac{\partial * u(k-j)}{\partial W} \quad \text{(L2.4)}$$

$$\frac{\partial * y(k)}{\partial W} = \sum_{j=0}^{h} \frac{\partial y(k)}{\partial u(k-j)} \frac{\partial * u(k-j)}{\partial W} + \sum_{j=1}^{n} \frac{\partial y(k)}{\partial y(k-j)} \frac{\partial * y(k-j)}{\partial W} \quad \text{(L2.5)}$$

Define the input, output vectors *U(k)*, *Y(k)*:

The gradient vectors $\nabla_W U(k)$, $\nabla_U Y(k)$, $\nabla_Y Y(k)$ and $\nabla_W Y(k)$, then

$$\Delta W^T(k) = 2\mu e^T(k) Q.(\nabla *_W Y(k)) - \mu.(\nabla_U J_2^T(k))(\nabla *_W U(k)) \quad \text{(L2.6)}$$

Tracking convergence: $\int_0^t |e_C(\tau)|^2 d\tau < M_e$

Chose P, Q, Λ: satisfied the Lyapunov function [8]:

$\Lambda^T P + P\Lambda = -Q$ Define $\dot{e} = \Lambda e + \omega$

$\omega = min(e_I) =$ Minimum identification error

Chose the Lyapunov function candidate

$$V = \frac{1}{2} e^T P e \qquad \text{(L2.7)}$$

$$\dot{V} = \frac{1}{2} \dot{e}^T P e + \frac{1}{2} e^T P \dot{e} \qquad \text{(L2.8)}$$

The tracking converge when $V > 0$ and $\dot{V} < 0$

$$\dot{V} \le -\frac{1}{2} e^T Q e + e^T P \omega \le$$

$$\le -\frac{\lambda_{Q\,min}}{2} |e|^2 + e^T P \omega \qquad \text{(L2.9)}$$

$\lambda_{Q\min}$: The minimum eigenvalue of Q

$$\dot{V} \le -\frac{\lambda_{Q\,min} - 1}{2} |e|^2 - \frac{1}{2}\left[|e|^2 - 2e^T P\omega + |P\omega|^2\right] + $$

$$+ \frac{1}{2}[P\omega]^2 \qquad \text{(L2.10)}$$

$$\dot{V} \le -\frac{\lambda_{Q\min} - 1}{2} |e|^2 + \frac{1}{2}[P\omega]^2 \qquad \text{(L2.11)}$$

$$\int_0^t |e(\tau)|^2 d\tau \le \frac{2}{\lambda_{Q\,min} - 1}\left(|V(0)| + |V(t)|\right) +$$

$$+ \frac{1}{\lambda_{Q\,min} - 1} |P|^2 \int_0^t |\omega(\tau)|^2 d\tau \quad \text{(L2.12)}$$

Define $A = \frac{2}{\lambda_{Q\min} - 1}\left(|V(0)| + \sup_{t \succ 0} |V(t)|\right)$;

and $B = \frac{1}{\lambda_{Qm\min} - 1} |P|^2 \qquad \text{(L2.13)}$

$$\Rightarrow \int_0^t |e(\tau)|^2 d\tau \le A + B\int_0^t |\omega(\tau)|^2 d\tau \qquad \text{(L2.14)}$$

With A, B, positive, and the identification is matching, $\Rightarrow \int_0^t |e(\tau)|^2 d\tau \le M_e < \infty$

3. CONTROL SCHEME

We propose a general scheme of neural reference model control systems consist neural model and neural controller as show in Fig. 3.1. The neural training process uses two updating laws from lemma 1, and lemma 2. The overall scheme also include a disturbance canceller consist of a disturbance detector and disturbance rejector. The first component detects the mismatching error due to the system disturbances. Then, the disturbance rejector is a neural-based controller to produce a supplement control signal to compensate the control action by performing disturbance cancellation in plant control.

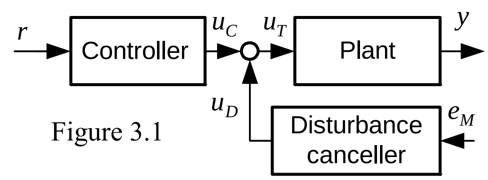
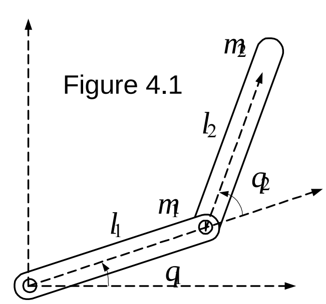


Figure 3.1

4. SIMULATION RESULTS

A planar two-link arm using DC electric drive is used for illustration purpose appears in fig 4.1, the dynamic are given in Lewis et al. [4], [11]. Here we select $l_1=1m$, $l_2=1m$, $m_1=0,8kg$, $m_2=2,3kg$ and $g_0=9,8m/s^2$, no friction term was used in this examples. We would like to illustrate the neural network tracking control scheme derived herein which will require no knowledge of the dynamics, nor even their structure, which is needed for adaptive control. The controlled system was tested by simulated experiments and the response of the controlled system to a tracking control is illustrated in Figure a, b. The control system was also tested for disturbances in the case of change in load and/or parameter variations. This is shown in Figure 4.2 with q_1, q_2 are outputs trajectory and qd_1, qd_2 are the references trajectory, respectively.

Figure 4.1



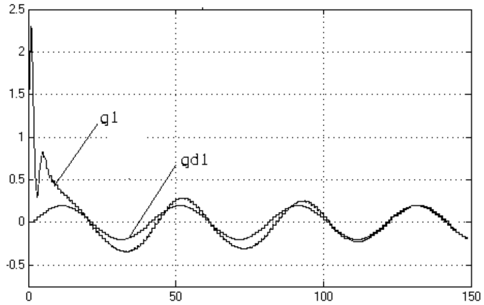


Figure 4.2.a: q_1 when $m_2=m_{2nom}$

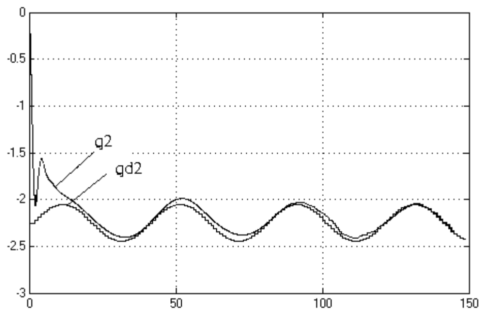


Figure 4.2.b: q_2 when $m_2=m_{2nom}$

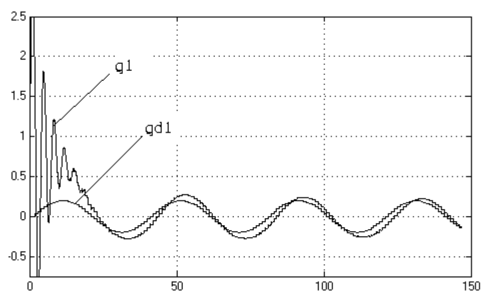


Figure 4.2.c: q_1 when $m_2=1.5m_{2nom}$
Disturbance appears

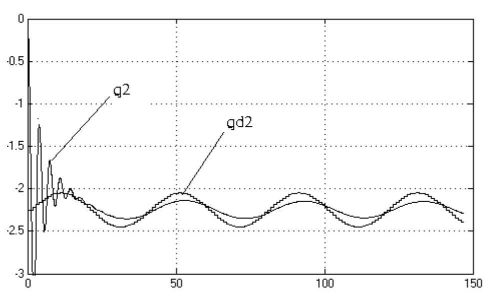


Figure 4.2.d: q_2 when $m_2=1.5m_{2nom}$
Disturbance appears

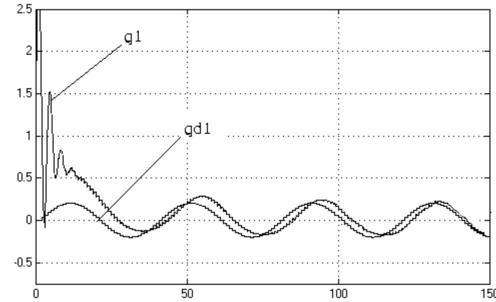


Figure 4.2.e: q_1 when $m_2=1.5m_{2nom}$
Disturbance canceller in action

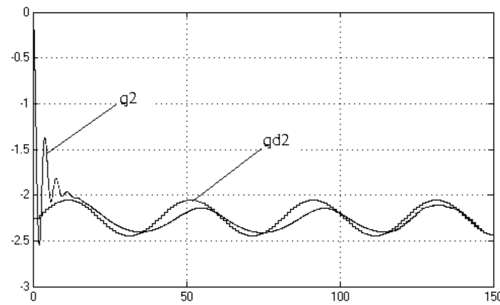


Figure 4.2.f: q_2 when $m_2=1.5m_{2nom}$
Disturbance canceller in action

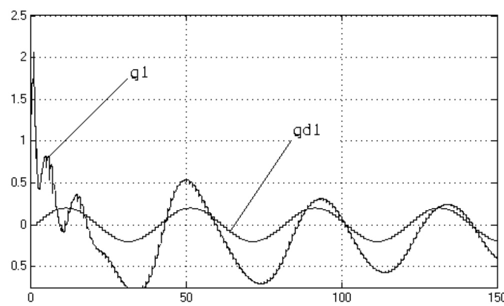


Figure 4.2.g: q_1 when $m_2=0.5m_{2nom}$
Disturbance appears

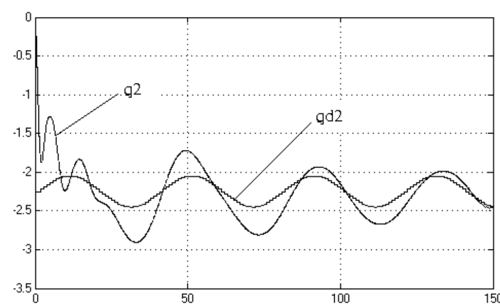


Figure 4.2.h: q_2 when $m_2=0.5m_{2nom}$
Disturbance appears

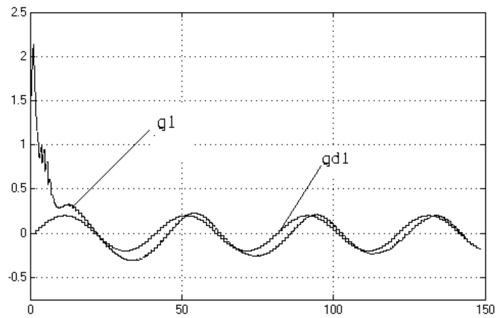


Figure 4.2.i: q_1 when m_2=0.5m_{2nom}
Disturbance canceller in action

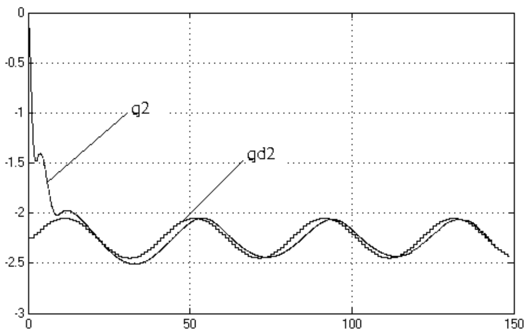


Figure 4.2.j: q_2 when m_2=0.5m_{2nom}
Disturbance canceller in action

5. CONCLUSION

In this paper a control algorithm and control scheme for dynamical nonlinear systems has been presented. It was demonstrated that the control algorithm can control a two-link robot arm. The response to disturbance was also demonstrated. The proposed method do not require the robot dynamics to be exactly know in the system, hence the same neural network controller can be applied to many type of robot or a class of nonlinear systems without modification.

REFERENCES

1. Anders Forsgren, Robert Kling; *An Implementation of Recurrent Neural Networks for Prediction and Control of Nonlinear Dynamic Systems*; Tech Report, Monash Univ., Melbourne, Australia, March 27, 2003.
2. Asriel U. Levin, Kumpati S. Nadrendra. *Control of Nonlinear Systems Using Neural Networks*, IEEE Trans. on N.Netwoks, pp. 30-42, Jan.1996.
3. Bernard Widrow, Gregory L. Plett; *Intelligent Adaptive Inverse Control*; Department of Electrical Engineering, Stanford University, USA, http://www.ust.edu/~IFAC.pdf
4. Chiman Kwan, Frank L. Lewis, Darren M. Dawaon; *Robust Neural Network Control of Rigid-Link Electrically Driven Robots*; IEEE Transaction on N.N.Vol.9 No. 4, July 1998
5. F.L. Lewis, Aydin Yesildirek,Kai Liu; *Multilayer Neural-Net Robot Controller with Guaranteed Tracking Performance*; IEEE Trans. on N.Nets Vol.7 No.2 March 1996
6. Isabelle Rivals, Léon Personnaz; *Neural Network Construction and Selection in Nonlinear Modeling*; Publ. in IEEE Trans. on Neural Networks Vol. 14, No. 4, 2003, pp. 804-819
7. K.S. Narendra, Parthasarathy K.; *Identification and Control of Dynamical Systems Using Neural Networks*, IEEE Trans.on N.Networks 1,1990, pp .4-27
8. Lin-Xin Wang; *A Course in Fuzzy Systems and Control*, Prent. Hall International, Inc. 1997
9. Paul J. Werbos; *Handbook of Neural Computation*; IOP Publishing Ltd and Oxford University Press, 1997
10. Siu-yeung Cho, Tommy W.S. Chow, *Training multilayer neural networks using fast global learning algorithm – least squares and penalized optimization methods*, Neurocomputing 25 (1999), pp 115-131.
11. Sybert Stroeve; *An Analysis of Learning Control by Backpropagation Through Time*; Neural Networks 11 (1998), pp.709–721
12. Young H. Kim, Frank L. Lewis; *Neural Network Output Feedback Control of Robot Manipulators*; IEEE Trans. on Robotic and Automation Vol.13 No.2 April 1999, pp 213-271.