AN ALGORITHM FOR CONTROLLING DYNAMICAL SYSTEM[+]

Fouad Sh.Tahir [*] **Shereen F.Abd-Alkarim**[**]

Abstract :

In this work, the feasibility of using neural adaptive control with finite recurrent back propagation algorithm for controlling DC motor is investigated to capture the nonlinearity which is better than that of using back propagation algorithm. This was obtained from the results of the simulation adopted in this work .

The build-up of the simulation procedure is implemented by MATLAB.

المستخلص :

تم في هذا البحث توضيح إمكانية استخدام المسيطر العصبي المتكيف باستخدام خوارزمية الانتشار المعاكس المحددة للسيطرة على محرك التيار المستمر للتخلص من الخصائص اللاخطية ، حيث أثبتت نتائج المحاكاة بان استخدام هذه الخوارزمية افضل من خوارزمية الانتشار المعاكس .

وقد تم بناء نموذج المحاكاة باستخدام برامجية الـ MATLAB .

7. List of Symbols

R_a : resistance of armature of motor.
L_a : inductance of armature of motor.
K_m : torque constant of motor.
K_b : back e.m.f constant of motor.
J_m :inertia of motor.
J_p : pendulum inertia of motor.
g :acceleration.
N : gear ratio.
ω_m : shaft velocity of motor.
. shaft position of motor : θ_m

Introduction:

Artificial neural networks have been shown to be effective as computational processors for adaptive control. They exhibit a number of desirable properties not found in conventional symbolic computation systems including robust performance, high degree of fault-tolerance, high parallel computation rates, the ability to

[+]**Received on 24/1/2007Accepted on 1/10/2007**

Lecturer Energy and Fuel Research Center University of Technology [*]

Lecturer Dept.of Educationa Technology University of Techonlogy [**]

generalize, and adaptive learning[1]. The challenges for future research in the field of control is to develop robust, adaptive, and fault-tolerant controls [2]. Over the past ten years, significant progress has been made in theory and applications of adaptive control , it has become a promising approach to achieve high performance of advanced control systems [3].

The idea of using neural networks for controlling physical systems has recently been explored.However, Psaltis et al.[4] only showed an application that convert polars coordinates to cartesian coordinates in which no dynamics of the plant is considered. Although Kawato et al.[5] have applied neural networks to the control of a complex robotic manipulator, they used neural network only as a part of the inverse dynamics model of the manipulator, sub-systems using the conventional computational method were used to compute various nonlinear transformations of input signal.

In this work we use neural adaptive control which has been developed for controlling dynamical system with finite recurrent back propagation (FRBP).

Structure Of Finite Recurrent Back Propagation:

Let us suppose that the network has (I) input units, (J) hidden units and (K) output units. Fig(1) describes the construction of the network [6,7].

To express easily and clearly , let (w) be the weight matrix which consists of WI_{jxi} ,WH_{jxj} ,WO_{kxj} .

WI_{jxi} : the weight matrix from input layer to hidden layer .

WH_{jxj} : the weight matrix from hidden layer to hidden layer .

WO_{kxj} : the weight matrix from hidden layer to output layer .

The characteristic of the network is that the units in hidden layer are connected to themselves and each other . However the information in such units at one time can only be transferred until being replaced by new data after finite steps , i.e. the units in hidden layer only memorize the information for finite time which is the key difference from the fully recurrent BP. For the sake of this , the network here is called Finite Recurrent Back Propagation network (FRBP). Here let the number of steps be integer, Hence at the time (t) the input to the hidden units is :

$$AH_i(t) = \sum_{i=1}^{I} I_i W_{ij} + \sum_{i \neq j} W_{ij} f(AH_j(t-1)) \qquad \begin{array}{l} i=1,2,\ldots,I \\ J=1,2,\ldots,J \end{array} \qquad \ldots\ldots\ldots(1)$$

The activation function (f) is a sigmoid , that is :

$$F(x) = \frac{1}{2} + \frac{1}{(1+EXP(-X))} \qquad \ldots\ldots\ldots(2)$$

The output of the network is a weighted sum of the hidden unit outputs :

$$O_k(t) = f\left[\sum_{k=1}^{K} W_{jk} f(AH_j(t)\right] \qquad \text{k =1,2,…,K} \qquad \text{……(3)}$$

The net is trained by minimization of the total error which is evaluated as that :

$$E(t) = \frac{1}{2}\sum_{p=1}^{P}\ \sum_{k=1}^{K}(T_{kp}(t) - O_{kp}(t))^2) \qquad \text{……(4)}$$

where :

h_j : the actual output of hidden neuron j for input signal i .

I_i : input signal of input neuron i .

W_{ij} :synaptic weight between input neuron i and hidden neuron j .

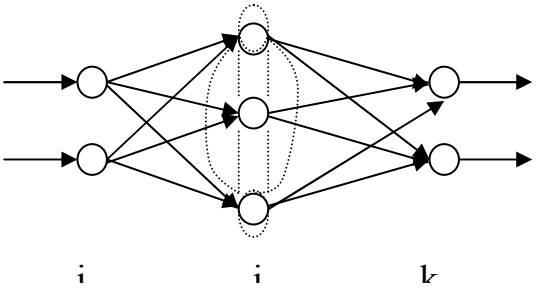
f : the activation function .

O_k : the actual output of output neuron k .

$AH_i(t)$: total input to neuron i .

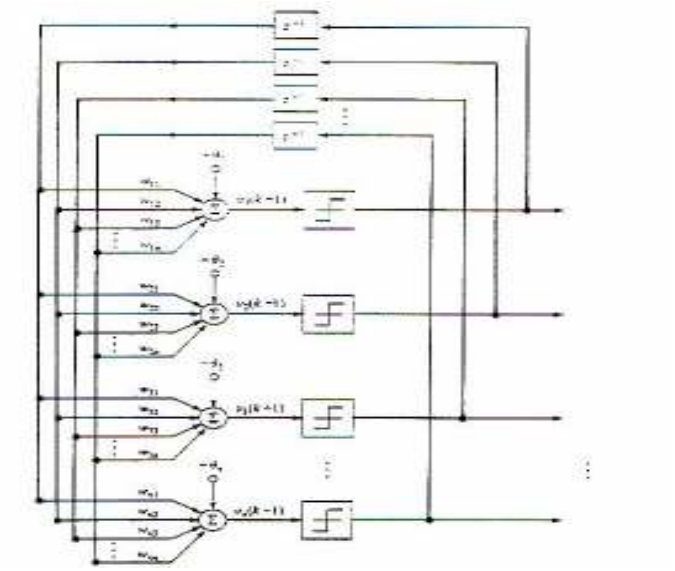
P : the sample length .

$T_{kp}(t)$: the target values that the output of the Kth unit in output layer while inputting the Pth sample match at time (t) . This can be fulfilled by a gradient desent procedure adjusting (W) along the negative of $\nabla_w E(t)$.The weight change for any particular in the network by FRBP algorithm.



i i k

(a)



(b)

(a)Finite recurrent back propagation network.(b)Details of Jth nodes structure

.

Figure(1) Network construction

Case Study:

The inverted pendulum (IP) system is essential in the evaluation and comparison of various control theories. The inverted pendulum is used in simulations and experiments to show the performance of different controllers (e.g. PID, State Space and Intelligence Controllers).[8]

We choose an inverted pendulum controlled by a dc motor via a gear train as our plant which is illustrated in Fig (2).

We make the following assumptions: [9]

1. The dc motor is armature-controlled.
2. The motor inertia J_m is negligible compared to the inverted pendulum inertia J_p .
3. $J_p = mL^2$,i.e., the inverted pendulum inertia is calculated by considering mass m as a point mass at the end of a massless shaft of length L .
4. The gear train has no backlash , and all connecting shafts are rigid

The armature-controlled dc motor is depicted schematically in Fig (3).

We choose the state variables:

$$x_1 = \theta_p \ , \qquad x_2 = \frac{d\theta_p}{dt} = \omega_p \ , \qquad x_3 = I_a$$

For the plant output , we take $y = x_1$. In vector-matrix form the state equations describing our plant are [10]:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{g}{L}\sin x_1 + \frac{NK_m}{mL^2}x_3 \\ -\frac{K_b N}{L_a}x_2 - \frac{R_a}{L_a}x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L_a} \end{bmatrix} u \qquad \text{…….(5)}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

where K_m is the motor torque constant , K_b is the back emf constant , and N is the gear ratio . Reasonable parameters for the plant are $g = 9.8 m/s^2$, L=1m , m = 1 kg , N =10 , K_m =0.1 Nm/A , K_b = 0.1 Vs/rad , R_a =1Ω , L_a =100 mH . These parameters lead to

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ 9.8\sin x_1 + x_3 \\ -10x_2 - 10x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix} u$$

…….(6)

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

The neural adaptive control configuration uses two neural networks: [11] a controller network and a model network as shown in Fig (4). The model network can be trained off-line using historical plant measurements.

The controller is adaptively trained to force the plant output to track a reference model output .The model network is used to predict the effect of controller changes on plant output , which allows the updating of controller parameters.

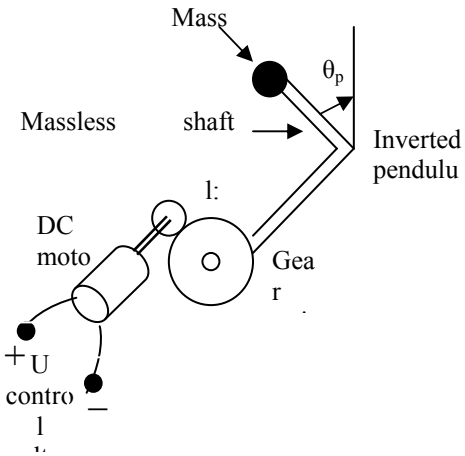


Fig (2) Dc motor controlling an inverted pendulum via a gear train

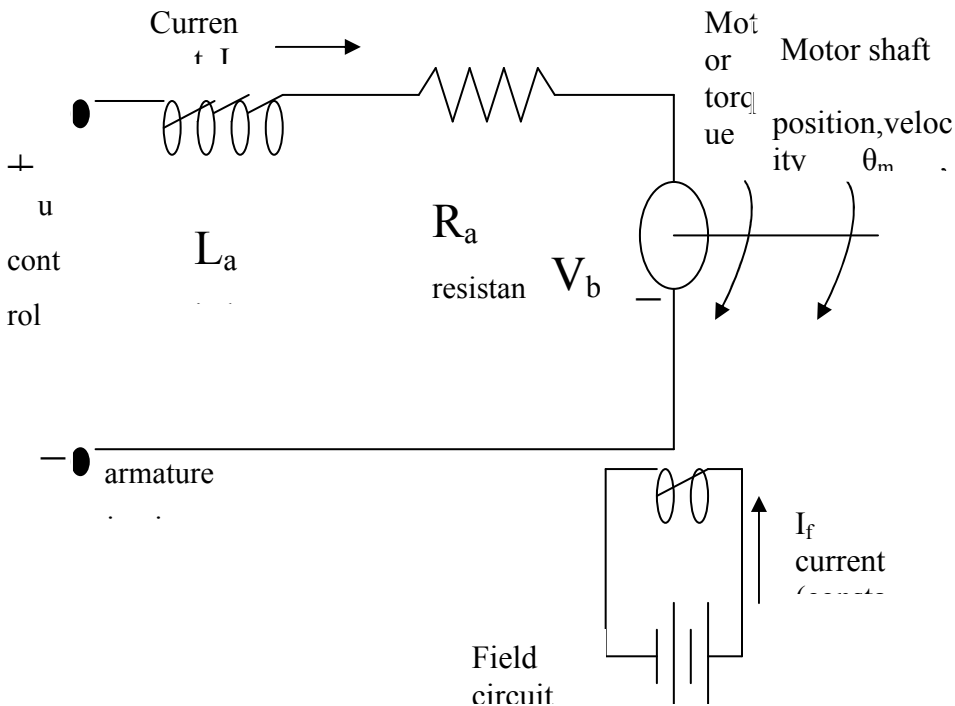


Fig (3) Schematic of an armature- controlled dc motor

o/p signal to correct the error of the

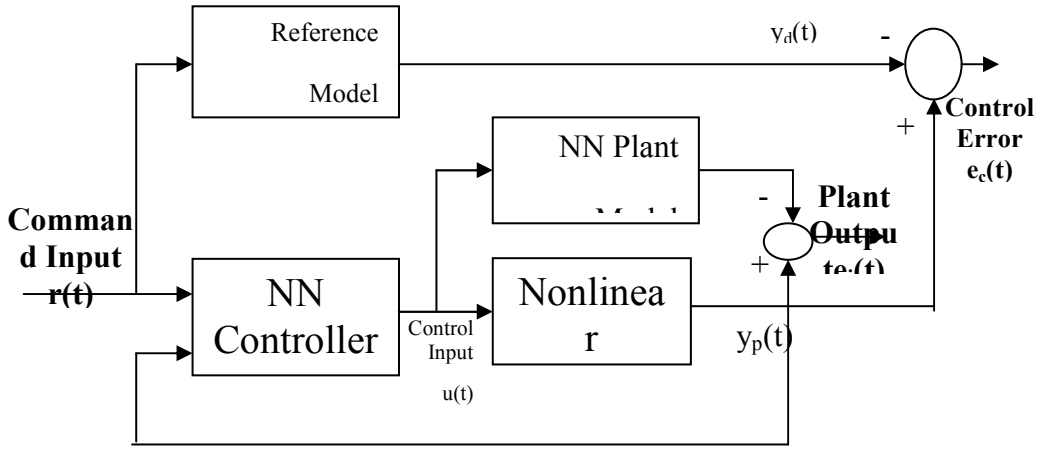


Fig (4) Block diagram of the indirect adaptive control approach

Simulation Results

Fig (5) shows the response of the motor with u = 0 , and initial conditions x_1=0.02 , x_2= 0 and x_3= 0 when the network is trained with both BP and FRBP algorithms .The simulation parameters can be specified as follows:

1.The initial conditions correspond to the pendulums unstable equibrium position : x_1= 0.0 , x_2= 0.0 ,x_3= 0.0 .

2.The input control signal (control voltage) is u(t)=step(t) , the unit step function.

3.The initial weights in the network are small random numbers.

It seems from Table (1) that the number of iterations and the time needed for training the network in FRBP algorithm are less than BP algorithm and this is because when using FRBP algorithm the guarantee that the parameters will converge or that the output error will tend to zero takes less time than using BP algorithm ,that is because of the finite recurrent BP which stors the information for finite time .

Conclusions

The FRBP which is proposed in this work has a simple structure and training algorithm .

This network has proved its ability to capture the nonlinearity .It can be seen from the results obtained that using FRBP algorithm has a better response than that when using BP algorithm

Also the FRBP algorithm reaches the required error goal much faster , with better accuracy and in a more stable condition than the BP algorithm and it is seen that FRBP provides faster response with settling time =8.1 sec while for BP=10.4sec and with less overshoot .

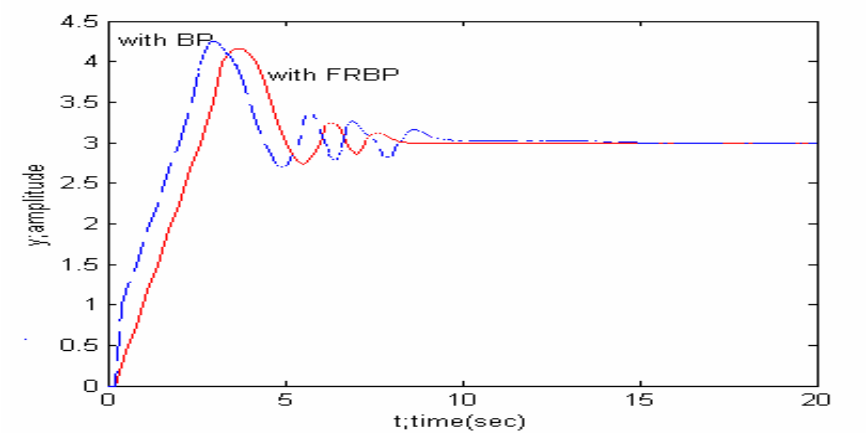


Fig (5) Response for motor with FRBP and BP algorithms

Table (1) Number of iterations and time for FRBP and BP algorithms

Type of algorithm	No. of iteration	Time (sec)
FRBP	22	10
BP	68	31

References:

1. Phil, P., Neural Networks, Antony Rowe Ltd , Great Britain ,2000 .
2. Mohammed, M.F. A New Approach For Synchronous Machine Control Using Artificial Neural Network , M.Sc. Military College of Engineering ,Iraq. 2002.
3. Lin, F. & Saikalis, G. "Adaptive Neural Network Control by Adaption Interaction ", (internet search) , : // www.emsl.pnl.gov: 2080/proj/neuron-ann.htm1.2001.
4. Psaltis, D. Sideris, A. and Yamamura, A. "A Multilayered Neural Network Controller", IEEE Control Systems Magazine,Vol.8, No.2, pp. 17-21, 1988.
5. Kawato, M. Furukawa, K. and Suzuki, R. "A Hierarchical Nrural-Network Model for Control and Learning of Voluntary Movement " ,Biological Cybernerics, vol.57, No.3, pp.169-185, 1987.

6. Narendra, K.S. Mukhopadhyay, S. “Adaptive Control Using Neural Networks and Approximate Models”, IEEE Transaction on Neural Networks, vol.8.,No.3 , pp.475-485, 2000.
7. Fredric, M. Princilples of Neuro Computing for Science and Engineering, Mc. Graw-Hill, 2001.
8. Ogata, K. Modern Control Engineering, Prentice Hall, International Edition ,New Jersey, 2002.
9. Consoli, K. Marchesoni, S. “An Adaptive Control of Unknown Dynamical Systems via Neural Networks” *IEEE Control Systems Magazine*, vol.4, No.3, pp.25-30, 1996.
10. Dorsey, J. Continuous and Discrete Control Systems.Modeling, Dentifaction,Design,and Implementation, Mc Graw-Hill, International Edition, 2002.
11. Patifio,H. and Derong, L. “Neural Network based model reference adaptive Control System”, *IEEE Transaction on System,,Man and Cybernetic ,par B*,Vol.30, No.1, pp.122-128, 2000.