International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

Research Article

ANN Control of Non-Linear and Unstable System and its Implementation on Inverted Pendulum

Nitin S Bhattathiri[Ȧ*] and P Anitha Saraswathi[Ȧ]

[Ȧ]Department of Instrumentation & Control Engineering, SRM University, Chennai, India

Accepted 01 April 2014, Available online 10 April 2014, **Vol.4, No.2 (April 2014)**

Abstract

The inverted pendulum is a classical problem for control problem, therefore it is used to benchmark new control techniques This process is best suited for control engineers working on developing new controllers as it is highly nonlinear and unstable. The inability of normal controllers like PID to adapt to changes in the process forces our hand in developing controller with intelligence. Controllers with intelligence means the ability of the controller to react or adapt to uncertainty in the unknown process. Neural Network Controller has the ability to learn and also ability to tolerate incorrect or noisy data. Moreover developing a controller using Adaptive Linear Element (ADALINE) and Radial Basis Function based (RBF) based controllers doesn't require mathematical modelling of the system. The main aim of the project is to keep the pendulum erect. The use of Neural Network based controller reduces the error. The Adaptive Neural Network toolbox is used to do a comparative study of the two types of controllers.

Keywords: *Inverted Pendulum, Adaptive Neural Networks, ADALINE, RBF*

1. Introduction

The process used for this research the inverted pendulum is a typical non-linear and unstable system. The problem faced with these kind of systems is that it cannot be controlled by linear control methods like PI, PID. The characteristics of inverted pendulum make it difficult to identify as well as control. Adaptive Control is a set of techniques for automatic adjustment in real time of controllers in order to achieve or maintain a desired level of performance of a control system when the process parameters are unknown or change with time.

Neural Networks can be used for system identification as well as adaptive control of non-linear systems. Neural Networks are used for approximating complex control problems where the exact mathematical model of the system cannot be determined as in the case of Inverted Pendulum. The Adaptive Neural Network (ANN) controller behaves in such a way that it counteracts any disturbance in the system.

System identification is the process in which a dynamic model of the present state of the system is developed by the input and output signals received from the system. The output data from unstable systems lacks the proper information nor the dynamics of the system. This scenario leads us to develop a feedback controller which stabilizes the system before identification. Neural Network control is classified into 3 types- supervised, direct inverse and unsupervised. In this research we develop the neural network is trained to model the inverse of the process. Theoretically speaking the nonlinearities in the ANN will counter effect the nonlinearities in the process if inverse model developed is accurate.

The study in neural network controller has been an area of great interest over the recent years. There are two factors that mainly contribute to this popularity. First is the capability of neural networks to approximate nonlinear functions (S.S.Sastry and A.Isidori,1989; A.Packard and M.Kantner, 1996) . This is an important reason as in most of the cases control of the process can be achieved through nonlinear control. The second is the ability of neural networks to adapt (R.M.Dressler,1967; V.M.Popov ,1973). The way in which neural networks adapt is natural it is not required to build a model or parameter identification. Two learning algorithms has been proposed, the two being supervised learning and unsupervised learning (S.S.Sastry and A.Isidori,1989) and (K.S.Narendra, Y.H.Lin, and L.S.Valavani,1980; T.Callinan, 2003). For each of this algorithms there exists feed-forward as well as feedback networks.

2. Methodology

2.1 Backpropogation Algorithm

Back propagation was created by generalizing Windrow-Hoff learning rule to multilayer neural networks and nonlinear differential transfer functions. Input vector and corresponding target vectors are used to train the neural networks till the approximation of the function is obtained. Neural Networks with biases are capable of handling any

*Corresponding author: **Nitin S Bhattathiri**

functions with a finite number of discontinuities. Standard backpropagation is a gradient descent algorithm, as is the Widrow-Hoff learning rule, in which the network weights are moved along the negative of the gradient of the performance function (Hagan,M and Demuth,H ,1996). The term backpropagation refers to the manner in which the gradient is computed for nonlinear multilayer networks.

Properly trained backpropagation networks give reasonably favourable output when it is presented with unfamiliar inputs. When a new input is given to the network it produces the output similar to the correct output for input vectors used in training that are similar to the new input being presented. This property makes it possible to train a network on a representative set of input target pairs and get a suitable response without training the network for all input/output pairs (Saerens M. and Soquet A ,1991).

The backpropagation algorithm is used to find a local minimum of the error function. The network is initialized with randomly chosen weights. The gradient of the error function is computed and used to correct the initial weights. Our task is to compute this gradient recursively.
After choosing the weights of the network randomly, the backpropagation algorithm is used to compute the necessary corrections. The algorithm can be decomposed in the following four steps:
i) Feed-forward computation
ii) Backpropagation to the output layer
iii) Backpropagation to the hidden layer
iv) Weight updates

The algorithm is stopped when the value of the error function has become sufficiently small. A two-layer network is used to learn the system and the standard backpropagation algorithm (J.J.Slotine and W.Li, 1991; P.A.Ioannou and J.Sun, 1996) is employed to train the weights.

2.2 ADALINE Based Algorith

Essentially, the introductory weights of ADALINE system need to be set to little arbitrary qualities and not to zero as in Hebb or perceptron networks, in light of the fact that this may impact the error factor to be acknowledged. After the starting weights are accepted, the activations for the input unit are set. The net input is figured dependent upon the training input patterns and the weights. The training methodology is proceeded until the error, which is the contrast between the target and the net input gets to be least. The step based training algorithm for an ADALINE is as accompanies:
Step 1: Initialize the weights (not zero but small random values are used). Set the learning rate α.
Step 2: While stopping condition is false, do step 3-7.
Step 3: For each bipolar training pair s:t, perform steps 4-6.
Step 4: Set activations of input units $x_i = s_i$, for i= 1 to n.
Step 5: Compute net input to output unit

$$y_{\_in} = b + \sum_i x_i w_i$$

Step 6: Update bias and weights, i=1 to n.

$$w_i(new) = w_i(old) + \alpha(t - y_{\_in})x_i$$

$$b(new) = b(old) + \alpha(t - y_{\_in})$$

Step 7: Test for stopping condition.
The stopping condition may be when the weight change reaches small level or number of iterations etc.

3. Modelling Of Inverted Pendulum

The reason for developing a Simulink model for inverted pendulum is that the model developed will have the same characteristics of that of the real time process. The initial step in developing a Simulink model for inverted pendulum is to derive the system dynamics equation (S.S.Sastry and A.Isidori, 1989) using 'Lagrange Equations'. There are many other methods for developing the system equations but here in this paper we use Lagrange Equation method for developing the system equations. Fig. 1 is a free body diagram of the inverted pendulum.

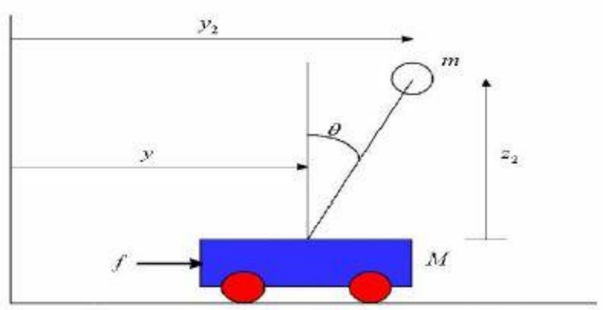

Fig.1 Free body diagram of the inverted pendulum.

Where
M = Mass of the cart
m = Mass of the pole
l= Length of the pole
f=control force
The Lagrange equations use the kinetic and potential energy in the system to determine the dynamical equations of the cart-pole system. The kinetic energy of the system is the sum of the kinetic energies of each mass. The kinetic energy, T_1 of the cart is given by

$$T_1 = \frac{1}{2} M \dot{y}^2 \tag{1}$$

The pole above the cart can move along the vertical as well as the horizontal plane so the kinetic energy, T_2 is given by

$$T_2 = \frac{1}{2} m(\dot{y}_2^2 + \dot{z}_2^2) \tag{2}$$

From the free body diagram we find the values of y_2 and z_2

$$y_2 = y + l\sin\theta \tag{3}$$

$$z_2 = l\cos\theta \tag{4}$$

$$\dot{y}_2 = \dot{y} + l\theta\dot{\cos}\theta \tag{5}$$

$$\dot{z}_2 = -l\dot{\theta}\sin\theta \quad (6)$$

The total energy, T of the inverted pendulum is therefore the sum of T_1 and T_2 given by

$$T = T_1 + T_2 = \frac{1}{2}\left[M\dot{y}^2 + m(\dot{y}_2^2 + \dot{z}_2^2)\right] \quad (7)$$

Equations (3) and (4) are substituted in (7) and we get

$$T = \frac{1}{2}M\dot{y}^2 + \frac{1}{2}m\left[\dot{y}^2 + 2\dot{y}\dot{\theta}l\cos\theta + l^2\theta^2\right] \quad (8)$$

The potential energy V of the inverted pendulum system is

$$V = mgz_2 = mgl\cos\theta \quad (9)$$

The Lagrangian function i

$$\mathrm{L} = T - V \quad (10)$$

$$L = \frac{1}{2}[M\dot{y}^2 + m[\dot{y}^2 + 2\dot{y}\dot{\theta}l\cos\theta + l^2\theta^2]] - mgl\cos\theta \quad (11)$$

The state space variables are y and θ, so the Lagragian equations are

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{y}}\right) - \frac{\partial L}{\partial y} = 0 \quad (12)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\theta}}\right) - \frac{\partial L}{\partial\theta} = 0 \quad (13)$$

$$\frac{\partial L}{\partial y} = (M+m)\dot{y} + ml\cos\theta\,\dot{\theta} \quad (14)$$

$$\frac{\partial L}{\partial y} = 0 \quad (15)$$

$$\frac{\partial L}{\partial\dot{\theta}} = ml\cos\theta\,\dot{y} + ml^2\dot{\theta} \quad (16)$$

$$\frac{\partial L}{\partial\theta} = mgl\sin\theta \quad (17)$$

Equations 14-17 are substituted in the Lagrangian equation and we get the dynamic equation for the nonlinear inverted pendulum which are

$$(M+m)\ddot{y} + ml\cos\theta\,\ddot{\theta} - ml\dot{\theta}^2\sin\theta = f \quad (18)$$

$$ml\cos\theta\, yml\sin\theta\,\dot{y}\dot{\theta} - mgl\sin\theta = 0 \quad (19)$$

Some of the modelling as well as the control technique used in this paper are linear so the equations 18 and 19 should be linearized. In order to do this we substitute the value of $\sin\theta = 0$ and $\cos\theta = 1$. This is because the value of θ is very small. The quadratic terms are also negligible. Thus we form two linear equations given below

$$\ddot{y} = \frac{f}{M} - \frac{mg}{M}\theta \quad (20)$$

$$\ddot{\theta} = -\frac{f}{Ml} + \left(\frac{M+m}{Ml}\right)g\theta \quad 21)$$

Now that we have obtained dynamic equations related to the inverted pendulum system we can design the Simulink model of the system. The process of developing the Simulink model of inverted pendulum is by using the various gain, integrator blocks in MATLAB.

The linear inverted pendulum model is as shown in Fig.2. The mass of the cart, M is set to 1.2 Kg, the mass of the pendulum is set to 0.11 Kg and length of the pendulum is 0.4 meters.

FF

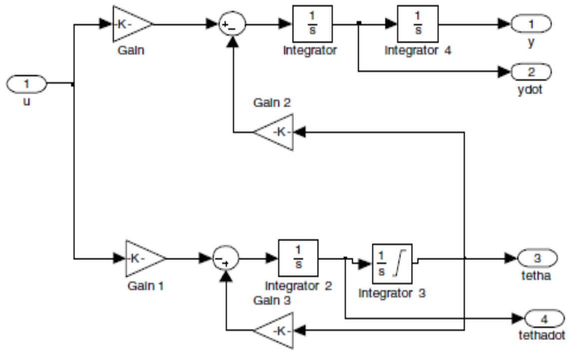

Fig.2 Simulink model of linear inverted pendulum.

The figure shown below Fig. 3 is a Simulink model of nonlinear inverted pendulum using equations (18) and (19). The following equations give the control law developed for the inverted pendulum. The initial four equations are substituted in the main equation. The main equation is the one which calculates the required force, U to keep the inverted pendulum erect or stable.

$$h_1 = \frac{3}{4l}\mathrm{g}\sin\theta \quad (22)$$

$$h_2 = \frac{3}{4l}\cos\theta \quad (23)$$

$$f_1 = m\left(l\sin\theta\,\dot{\theta}^2 - \frac{3}{8}g\sin\theta\right) - f\dot{x} \quad (24)$$

$$f_2 = M + m\left(1 - \frac{3}{4}cos^2\theta\right) \quad (25)$$

$$u = \frac{f_2}{h_2}[h_1 + k_1(\theta - \theta_d) + k_2\theta + c_1(x - x_d) + c_2x] \dot{-} f_1 \quad (26)$$

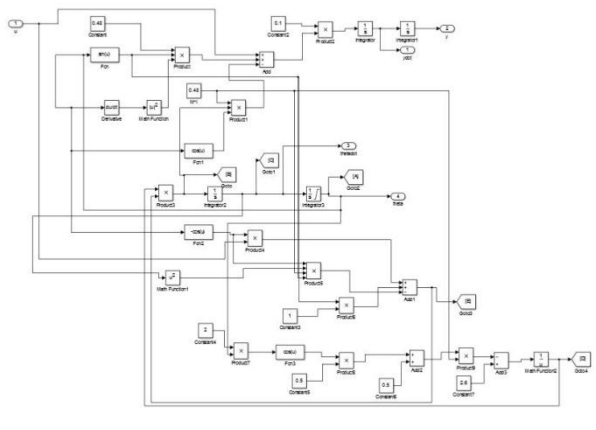

Fig.3 Simulink model of nonlinear inverted pendulum

For simulation purpose we take the following numerical values: M= 1.2 Kg, m= 0.1 Kg, l=0.4 m, g=9.81 m/s, k_1 =25, k_2 =10, c_1 =1, c_2 = 2.6 and x_d= 0 meters and$\theta_d = 0$ rad, which are the desired condition and the angle of pendulum. The Simulink model developed using the above control law was developed and is shown below in Fig.4.

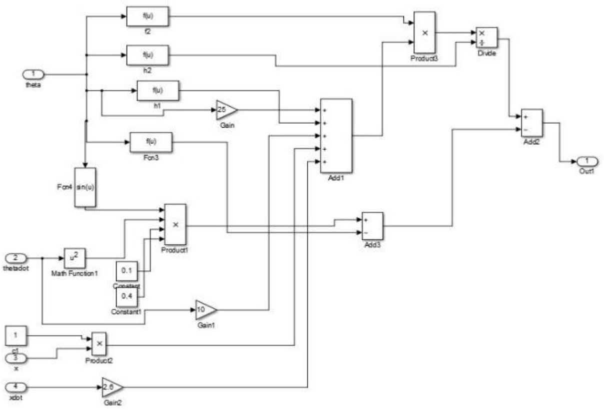

Fig.4 Simulink model of linear control law.

The figure below Fig.5 shows the model of nonlinear pendulum with control law.

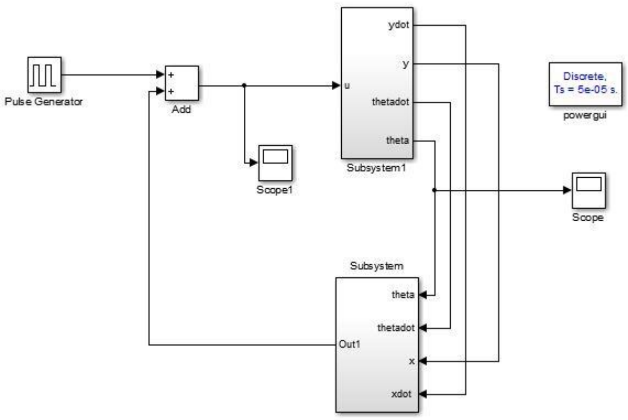

Fig.5 Simulink model of the system with control law.

4. Neural Network Control of Inverted Pendulum

The fundamental undertaking of this project is to design a controller which keeps the pendulum system inverted. There are a couple paramount focuses to recollect when planning a controller for the inverted pendulum. The inverted pendulum is open-loop unstable, nonlinear and a multi-output system. Nonlinear system: Standard straight PID controllers can't be utilized for this system on the grounds that they can't outline complex nonlinearities in the pendulum process. ANN's have indicated that they are equipped for distinguishing complex nonlinear systems. They ought to be appropriate for producing the complex inward mapping from inputs to control movements. Multi-output system: The inverted pendulum has four outputs, keeping in mind the end goal to have full state sentiment control four PID controller might need to be utilized. Neural systems have a huge focal point here because of their parallel nature. One ANN could be utilized rather than four PID's. Open-loop unstable: The inverted pendulum is unstable in open-loop conditions. As soon as the system is simulated the pendulum falls over. Neural networks require some serious energy to prepare so the pendulum system will must be stabilized by one means or another before a neural system could be prepared. When the genuine neuro-controller is produced in matlab, the fundamental sorts of neuro-control are examined. The five sorts of neural system control techniques that have been examined are supervised, model reference control , direct inverse, internal model control and unsupervised.

Supervised Control

It is conceivable to show a neural system the right movements by utilizing an existing controller or human reaction. This sort of control is called supervised taking in. Generally customary controllers (feedback linearization, principle based control) are based around an operating point. This implies that the controller can work accurately if the plant/process works around a certain point. These controllers will come up short if there is any kind of lack of determination or change in the obscure plant. The favourable circumstances of neuro-control is if a questionable matter in the plant happens the ANN will have the ability to adjust its parameters and administer controlling the plant when other vigorous controllers might come up short. In supervised control, an instructor gives right movements to the neural system to take in. In logged off preparing the targets are gave by an existing controller, the neural system modifies its weights until the yield from the ANN is like the controller (R.Marino and P.Tomei,1995). The point when the neural system is prepared, it is put in the criticism circle. Since the ANN is prepared utilizing the existing controller targets, it ought to have the ability to control the procedure. At this stage, there is an ANN which controls the methodology comparative to the existing controller. The true playing point of neuro-control is the capability to be versatile on the web. A mistake indicator (craved sign genuine yield indicator) is computed and used to conform the weights online. If a large disturbance/uncertainty happens in the process- the large error is feedback into the ANN and this changes the weights so the framework remains stable. The solution to this problem is using the adaptive neural networks. Fig. 6 shows a block diagram of ANN based supervisory learning

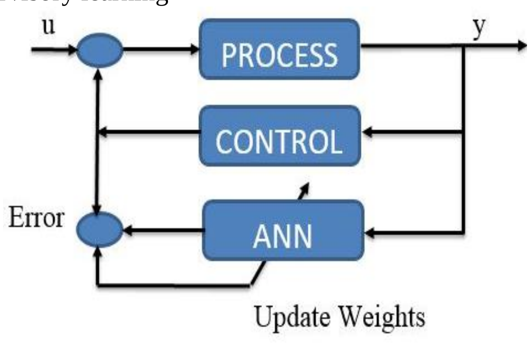

Fig.6 Supervised learning ANN controller

5. Simulation Results and Discussions

Fig.7 shows the Simulink model of supervised learning using ADALINE as controller with step input. Fig.8 shows the closed loop response of the inverted pendulum showing the controller makes it stable. Fig.9 gives us an idea about how the weight changes during training of the ADALINE network.

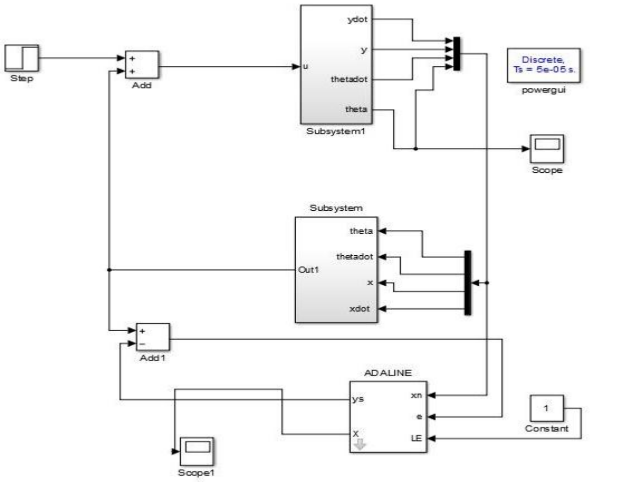

Fig.7 Adaptive training using ADALINE

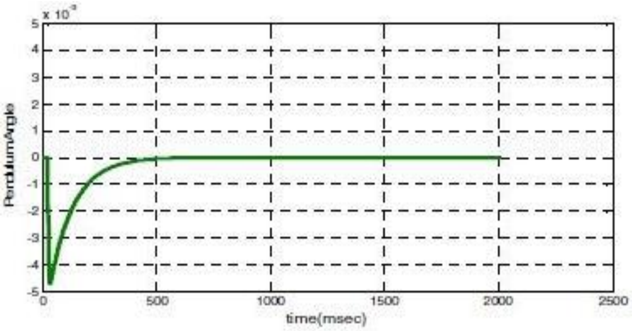

Fig.8 Response of ADALINE controller

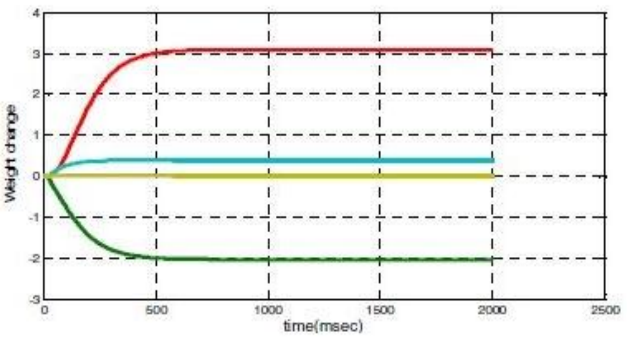

Fig.9 Weight change in ADALINE network for step input.

Fig.10 shows the Simulink model of supervised learning using RBF as controller with step input. Fig.11 shows the closed loop response of the inverted pendulum showing the controller makes it stable. Fig.12 gives us an idea about how the weight changes during training of the RBF network.

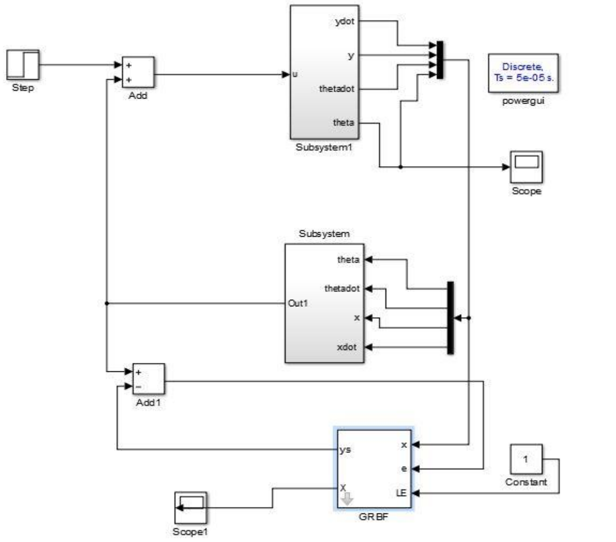

Fig.10 Adaptive training using RBF

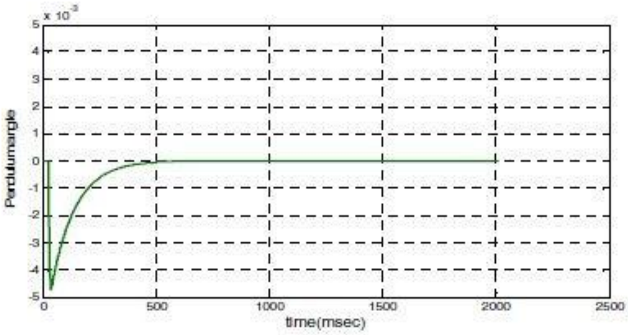

Fig.11 Response of RBF controller

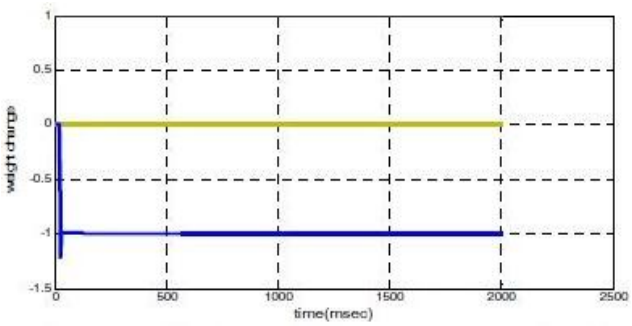

Fig.12 Weight change in RBF network for step input.

6. Conclusion

The fundamental objective of this research work is to develop ADALINE and RBF based neural network for exceedingly nonlinear inverted pendulum to keep it stable. The supervised control system is the most productive way to implement as seen from the outputs. On account of ADALINE neural network it is seen that the error minimize as the amplitude decreases but not in the case of RBF neural network. In both cases the shut circle reaction of the transformed pendulum is stable this shows that the supervised control make the framework stable if the framework is non-linear i.e. the instance of the altered pendulum. On correlation between ADALINE and RBF, ADALINE gives better come about and additionally estimates the control law exactly. The neural network controller created in this research is in based on the supervised control.

References

S.S.Sastry and A.Isidori (1989), Adaptive control of linearizable systems, IEEE Trans.Automat. Contr. **34,** no.11, pp 1123-1131.

A.Packard and M.Kantner (1996), Gain scheduling the LPV way, Proc. IEEEConf. Dec.Contr., pp 3938-3941.

R.M.Dressler (1967), An Approach to Model-Reference Adaptive Control System, IEEE Trans. Autom. Contr. 12, no. 1, pp 75-80.

V.M.Popov (1973), Hyperstability of control systems, Springer-Verlag.

J.J.Slotine and W.Li (1991), Applied Nonlinear control, Prentice Hall, New Jersey.

P.A.Ioannou and J.Sun (1996), Control Robust Adaptive, Prentice Hall, New Jersey.

R.Marino and P.Tomei (1995), Nonlinear Control Design Geometric, Adaptive, & Robust, Prentice Hall, New Jersey.

K.S.Narendra, Y.H.Lin, and L.S.Valavani (1980), Stable Adaptive Controller Design, Part II: Proof of Stability, IEEE Trans. Autom. Contr. 25, no. 3,pp440-448.

T.Callinan (2003) , Artificial neural network identification and control of the inverted pendulum

Hagan,M and Demuth,H (1996), Neural Network Design, Boston PWS.

Saerens M. and Soquet A (1991), Neural Controller based on back-propagation algorithm, IEE Proceedings –F, Vol. 138, No.1, pp 55-62.

Narendra K.S. and Parthasarathy K. (1993), Identification an control of dynamical systems using neural networks, IEEE transactions on neural networks, Vol.1, No.1, pp 4-27.

Marco,P and Raul,L (1995), Application of several neuro control schemes to a 2 DOF manipulator.