I.J. Intelligent Systems and Applications, **2012, 8, 53-61**
Published Online July 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2012.08.07

Supervised Online Adaptive Control of Inverted Pendulum System Using ADALINE Artificial Neural Network with Varying System Parameters and External Disturbance

Sudeep Sharma, Vijay Kumar, Raj Kumar
Electronics & Computer Engineering Department, IIT Roorkee, Roorkee, Hardwar
Sudeep_61186@yahoomail.co.in, vijecfec@iitr.ernet.in, rajkumar1chak@gmail.com

***Abstract*—** Generalized Adaptive Linear Element (GADALINE) Artificial Neural Network (ANN) as an Artificial Intelligence (AI) technique is used in this paper to online adaptive control of a Non-linear Inverted Pendulum (IP) system. The ANN controller is designed with specifications as: network type is three (Input, Hidden and Output) layered Feed-Forward Network (FFN), training is done by Widrow-Hoffs delta rule or Least Mean Square algorithm (LMS), that updates weight and bias states to minimize the error function. The research is focused on how to adapt the control actions to solve the problem of "parameter variations". The method is applied to the Nonlinear IP model with the application of some uncertainties, and the experimental results show that the system responds very well to handle those uncertainties.

***Index Terms*—** Generalized Adaptive Linear Element (GADALINE) Artificial Intelligence (AI) Inverted pendulum (IP); Artificial Neural Network (ANN); Feed-Forword Network (FFN); Least Mean Square algorithm (LMS).

I. Introduction

The inverted pendulum problem is a classical example of an unstable nonlinear dynamic system. Therefore, much attention has been given to explore a better solution for it and further, to solve other similar control problems. Nonlinear system control now occupies an increasingly important position in the area of process control engineering as reflected by the tremendous increase in the number of research papers published in this area. Artificial neural network proved to be a useful tool in dealing with applications such as pattern recognition, signal processing, image processing and various complex controls and mapping problems etc. Neural networks have been applied successfully to identify and control of dynamic systems because of their learning capacity and ability to tolerate incorrect or noisy data [1]. The universal approximation capabilities of the multilayer perceptron make it a popular choice for modeling nonlinear systems and for implementing general-purpose nonlinear controllers. A conventional controller which is discussed in [2] is used to collect the training data to develop Neural Network controller for a inverted pendulum system. Multilayer neural network consists of single layer each for inputs and outputs and one or more hidden layers, each layer can have one or more no's of neurons. The linking weights are updated by the back-propagation algorithm. It can control different systems through quick learning process and has perfect performances. Ni (1996) proposed a method for identification and control of nonlinear dynamic system a recurrent model was applied as identifier [3]. Gupata (1999) presented an improvement to back-propagation algorithm based on the use of an independent, adaptive learning rate parameter for each weight with adaptive nonlinear function [4]. LM algorithm is an effective optimization technique that can guide the weight optimization [5]. Neural network has been widely applied for state feedback controller design, nonlinear system control, nonlinear dynamical system identification, optimal control synthesis and three-dimensional medical image [6-10]. The ADALINE neural network is revisited and generalized in [11] by introducing a momentum term in the LMS learning algorithm. The momentum term helps to speed up the learning process and reduce the zigzag effect during convergence period of learning. This paper uses the regularized LM algorithm to offline train the connecting weights of neural network and for online training of ADALINE ANN LSM algorithm is used. The objective of this paper is to develop both online and offline neural network-based nonlinear controller for IP system and provide a rapid, reliable solution for the control algorithm.

II. ADALINE Artificial Neural Network

ADALINE is one of the basic models used for data prediction. This method has not found much application to predict nonlinear system dynamics because it is slow in convergence therefore multiple ADALINE units are combined together and named as MADALINE in [11] to capture the dynamic behavior of the feedback control law and to control the IP system. A generalized (ADALINE) neural network, called GADALINE, for online identification of linear time varying systems which can be described by a discrete time model. GADALINE is a linear recurrent type of neural network. In such networks, the current system output is dependent on past outputs and on both the current and past inputs. So the GADALINE needs to have output feedback and to remember past input/output data. The author further generalizes the adaptive learning by adding a momentum term to the GADALINE weight adjustment. This momentum term is turned on only during convergence period and the GADALINE's learning curve is therefore smoothed by turning off the momentum once the learning error is within a given small number. The low computational complexity makes this method suitable for online system identification and real time adaptive control applications. Simulation results showed that GADALINE provides a much faster convergence speed and better tracking of control law than the generalized Radial Basis Function (GRBF) which is discussed in [12].The structure of an ADALINE is shown in Fig.1.

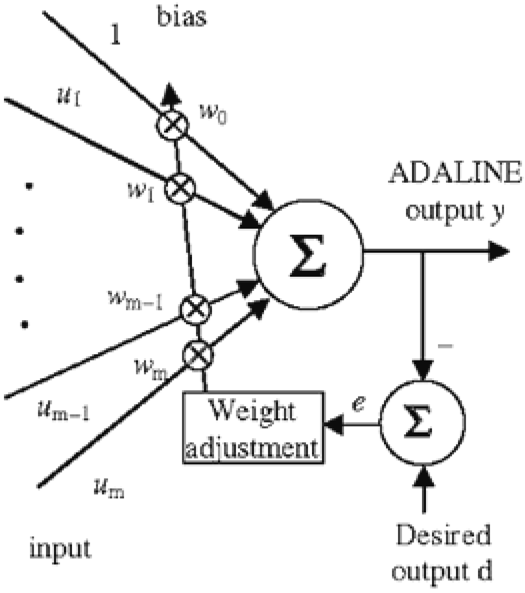


Fig.1. A single ADALINE unit

If we consider the notation for inputs as u_1-u_m, then the output y can be found by,

$$y = \sum_{1}^{m} w_i u_i$$
$$= u^T w \qquad (1)$$

Where u is the input vector and w is the weight vector,

$$u = [u_1 \ u_2 \dots u_m]^T \qquad (2)$$

$$w = [w_1 \ w_2 \dots w_m]^T \qquad (3)$$

The GADALINE learning algorithm is based on the Widrow-Hoff rule with a momentum added to the weight adjustment stage. The implication on addition of the momentum term to learning algorithm is discussed in [11] as follows.

Widrow-Hoffs delta rule or Least Mean Square algorithm (LMS), is based on minimizing the error function,

$$E(w) = 1/2\, e^2(t) \qquad (4)$$

where t is the iteration index and $e(t)$ is given by,

$$e(t) = d(t) - y(t) \qquad (5)$$

where the desired output d(t) at each iteration t is constant.

The weight adjustment is,

$$\Delta w(t) = \eta e(t) u(t) \qquad (6)$$

and

$$w(t+1) = w(t) + \Delta w(t) \qquad (7)$$

or

$$w(t+1) = w(t) + \eta e(t) u(t) \qquad (8)$$

Here the learning-rate parameter is usually in the range of 0< η <2. To speed up the convergence, a momentum term is added into the weight adjustment equation (6).

The new weight adjustment now becomes,

$$\Delta w = \eta e(t) u(t) + \alpha \Delta w(t-1) \qquad (9)$$

Where $1 > \alpha \geq 0$ is a small non- negative number, to see the effect of the momentum, we can expand the equation starting from t = 0,

$$\Delta w(0) = \eta e(0) u(0)$$

$$\Delta w(1) = \eta e(1) u(1) + \alpha \Delta w(0)$$
$$= \eta e(1) u(1) + \alpha\, \eta e(0) u(0)$$

$$\Delta w(2) = \eta e(2) u(2) + \alpha \Delta w(1)$$
$$= \eta e(2) u(2) + \alpha \eta e(1) u(1) + \alpha^2 \eta e(0) u(0)$$

$$\Delta w(t) = \eta \sum_{\tau=0}^{t} \alpha^{t-\tau} e(t) u(t) \qquad (10)$$

Remember $e(t)u(t)$ is just $-\partial E(t)/\partial w(t)$, then

$$\Delta w(t) = -\eta \sum_{\tau=0}^{t} \alpha^{t-\tau} \partial E(\tau)/\partial w(\tau) \qquad (11)$$

So, we have an exponentially weighted time series. We can then make the following observations,

a) If the derivative $\partial E(t)/\partial w(t)$ has opposite sign on consecutive iterations, the exponentially weighted sum $\Delta w(t)$ reduces in magnitude, so the weight $w(t+1)$ is adjusted by a small amount, which reduces the zigzag effect in weight adjustment.

b) On the other hand, if the derivative $\partial E(t)/\partial w(t)$ has same sign on consecutive iterations, the exponentially weighted sum $\Delta w(t)$ grows in magnitude, so the weight $w(t+1)$ is adjusted by a large amount. The effect of the momentum term is to accelerate descent in steady downhill region.

Because of this added momentum term, the weight adjustment continues after convergence, that is, when even the magnitude of the error term e is small. For linear system identification purpose, these weights correspond to system parameters. It is desirable to keep these parameters fixed once convergence is seen. So we should turn off the momentum term after convergence is detected, say $|e| < \varepsilon$, $\varepsilon > 0$, is a small number. However, when the system parameters change at a later time, another convergence period will begin, and then it is better to turn the momentum back on.

GADALINE training is performed by presenting the training pattern set, a sequence of input-output pairs. During training, each input pattern from the training set is presented to the GADALINE; the GADALINE computes its output y. This value and the target output from the training set are used to generate the error and the error is then used to adjust all the weights [11].

The step based training algorithm for an GADALINE is as follows:

Step1: Initialize the weights (non-zero i.e. some random small values are used). Set learning rate η and momentum term α.

Step2: While stopping condition is false, do step 3-7.

Step3: For each bipolar training pair *u: d*, perform steps4-6.

Step4: Set activations of input units $u_i=s_i$, for $i=1$ to *m.*

Step5: Compute net input to output unit as:

$$y = \sum\nolimits_1^m w_i u_i = u^T w$$

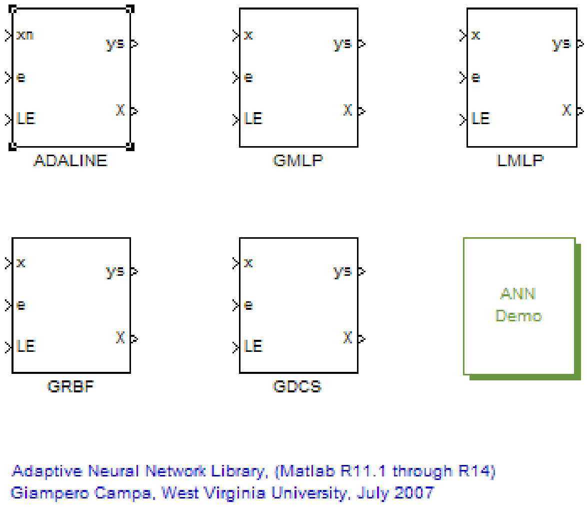
Step 6: Update network weights, as:

$$w(t+1) = w(t) + \eta e(t) u(t) + \alpha \Delta w(t-1)$$

Step 7: Test for stopping condition.

The stopping condition may be when the weight change reaches small level or number of iterations etc [13].

This problem is solved by using an adaptive neural toolbox which is an add-on for MATLAB simulink [15]. This toolbox basically allows for online neural learning to occur. The block diagram of the toolbox is shown in Fig.2. This toolbox contains an ADALINE and RBF (Radial Basis Function) simulink blocks.



Adaptive Neural Network Library, (Matlab R11.1 through R14)
Giampero Campa, West Virginia University, July 2007

Fig.2. Adaptive neural network toolbox

The inputs to each block are:

x: The input vector to the neural network.
e: The error between the real output and the network approximation.
LE: A logic signal that enables or disables the learning.

The outputs of each block are:

Ys: The value of the approximate function.
X: All the "states" of the network, namely the weights and all the parameters that change during the learning process.

There is an interface for each block so that the user can set the network parameters such as learning rate, number of the neurons in each layer, etc.

A. *Modeling of Inverted Pendulum and Controllers*

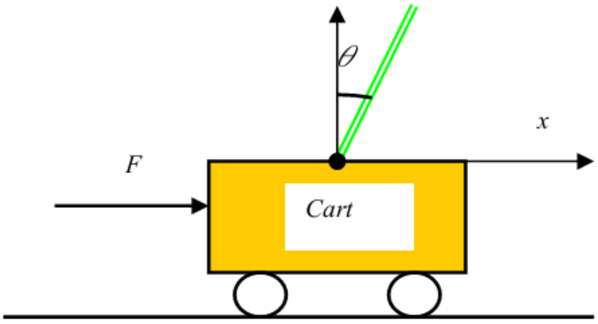


Fig.3. Inverted pendulum system sketch

The inverted pendulum system, as shown in Fig.4, is composed of a rigid pole and a cart on which the pole is hinged [14]. The cart moves on the rail track to its right or left, depending on the force exerted on the cart. The pole is hinged to the car through a frictionless free joint such that is has only one degree of freedom. The control goal is to balance the pole starting from nonzero conditions by supplying appropriate force to the cart. The displacement of the cart can be measured by a sensor installed on both sides of the rail track, and the angle signal can be measured by a coaxial angle sensor installed in the bearing which articulates the pendulum to the cart. On one side of the rail track a DC permanent magnetic direct torque motor is mounted which drives the cart to move on the rail track using a driving belt. When the cart moves left and right, torque act on the pendulum to keep the whole system in stable condition. Masked values of system plant parameters are used so that it can be easily varied to check the performance of the proposed controller with different system parameters. The cart mass is *M*, the pendulum mass is *m*, its length is *2L*. The cart is restricted to move within a fixed range. The reference position for x is 0 meter, when cart is in the center of the rail track and for θ is π radians, when the pendulum is at a natural stable downward position.

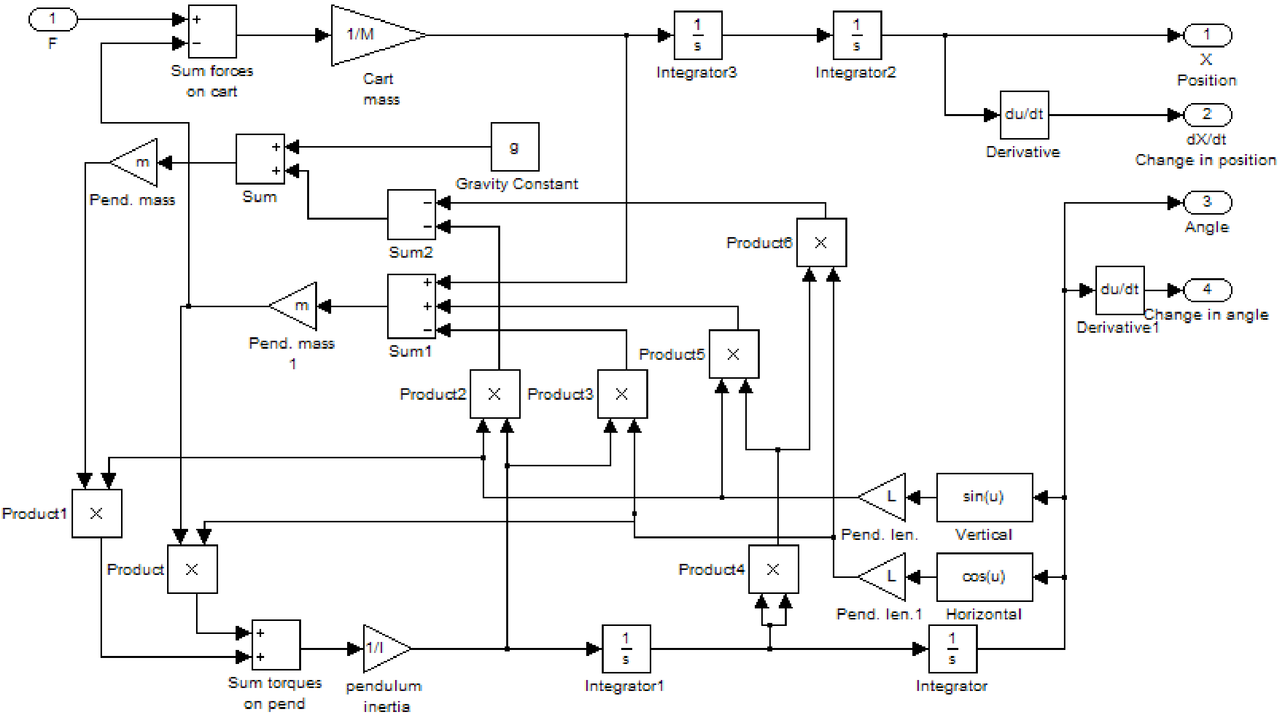


Fig.4. simulation model of nonlinear inverted pendulum system

B. Mathematical model of inverted pendulum

The dynamic equations of the system can be found with help of the Euler-Lagrange equation as in [2]:

$$(M + m\ddot{x} + b\dot{x} + mL\ddot{\theta}\cos\theta - mL\dot{\theta}^2\sin\theta) = F$$

$$(I + mL^2)\ddot{\theta} + mgL\sin\theta = -mL\ddot{x}\cos\theta \qquad (1)$$

Now by using the above equations the nonlinear model of IP system is developed in MATLAB simulink [15] which is shown in Fig.4.

C. Feedback control law for Inverted Pendulum System

To provide the supervised training data for the neural network controller we use the direct adaptive control technique described in [2]. Based on the equation (1) the following equations are a control law developed for the inverted pendulum controller. The first four equations (2-5) are entered into the main equation. The main equation (6) calculates the required force, *u* to keep the pendulum stable.

$$h_1 = \frac{3}{4L} gsin\theta \qquad (2)$$

$$h_2 = \frac{3}{4L} cos\theta \qquad (3)$$

$$f_1 = m\left(Lsin\theta\,\dot{\theta}^2 - \frac{3}{8} gsin2\theta\right) \qquad (4)$$

$$f_2 = M + m\left(1 - \frac{3}{4} cos^2\theta\right) \qquad (5)$$

$$u = \frac{f_2}{h_2}\left[h_1 + k_1(\theta - \theta_d) + k_2\dot{\theta} + c_1(x - x_d) + c_2\dot{x}\right] - f_1 \qquad (6)$$

The following numeric values are used: M=1.096 *Kg*, m=0.109 *Kg*, L=0.25 *m*, g=9.81 *m/s*^{2}, k$_1$=25, k$_2$=10, c$_1$=1, c$_2$=2.6. Also x_d = 0 meter and θ_d= 0 radian which are the desired position of the cart and angle of the pendulum respectively. A simulation model of the above control law was developed and is shown in Fig.5.

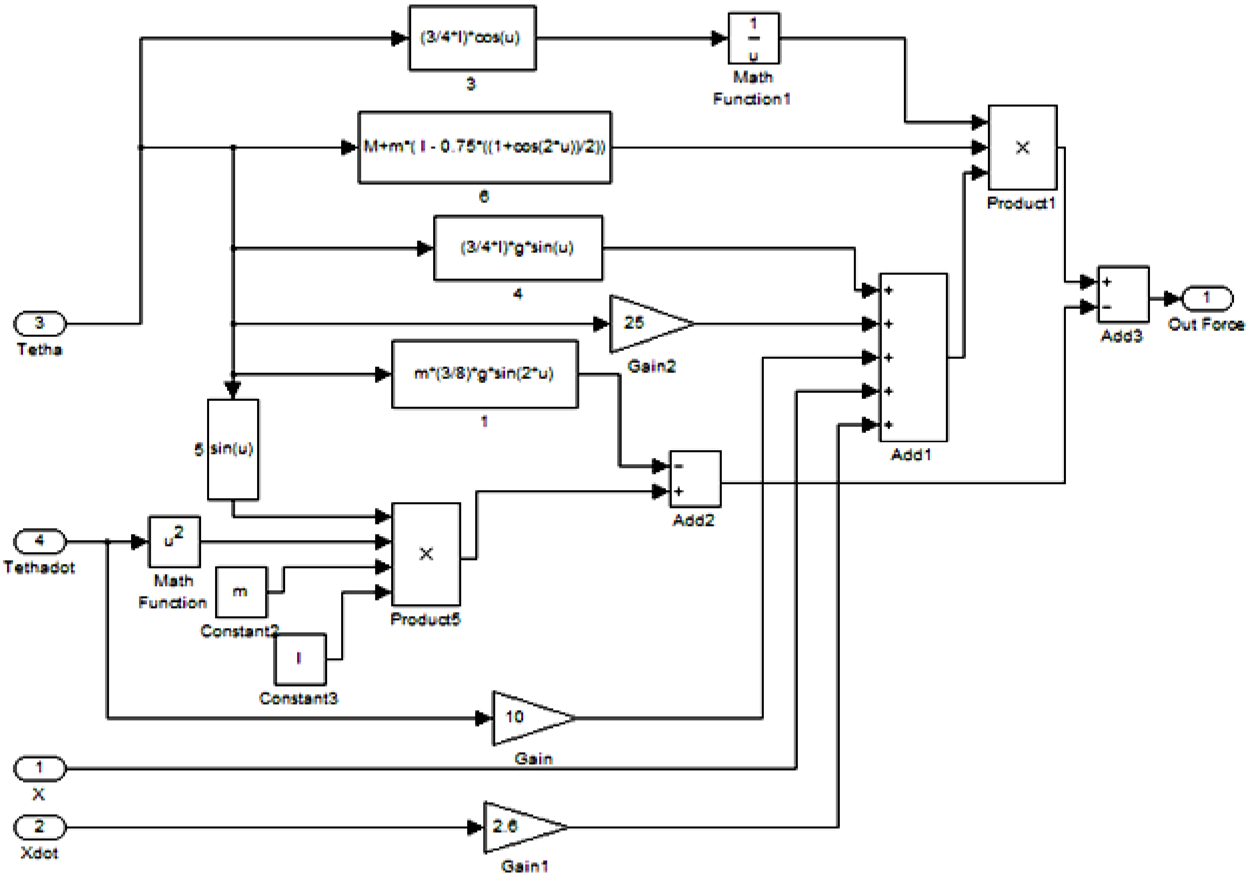


Fig.5. Simulink model of nonlinear control law

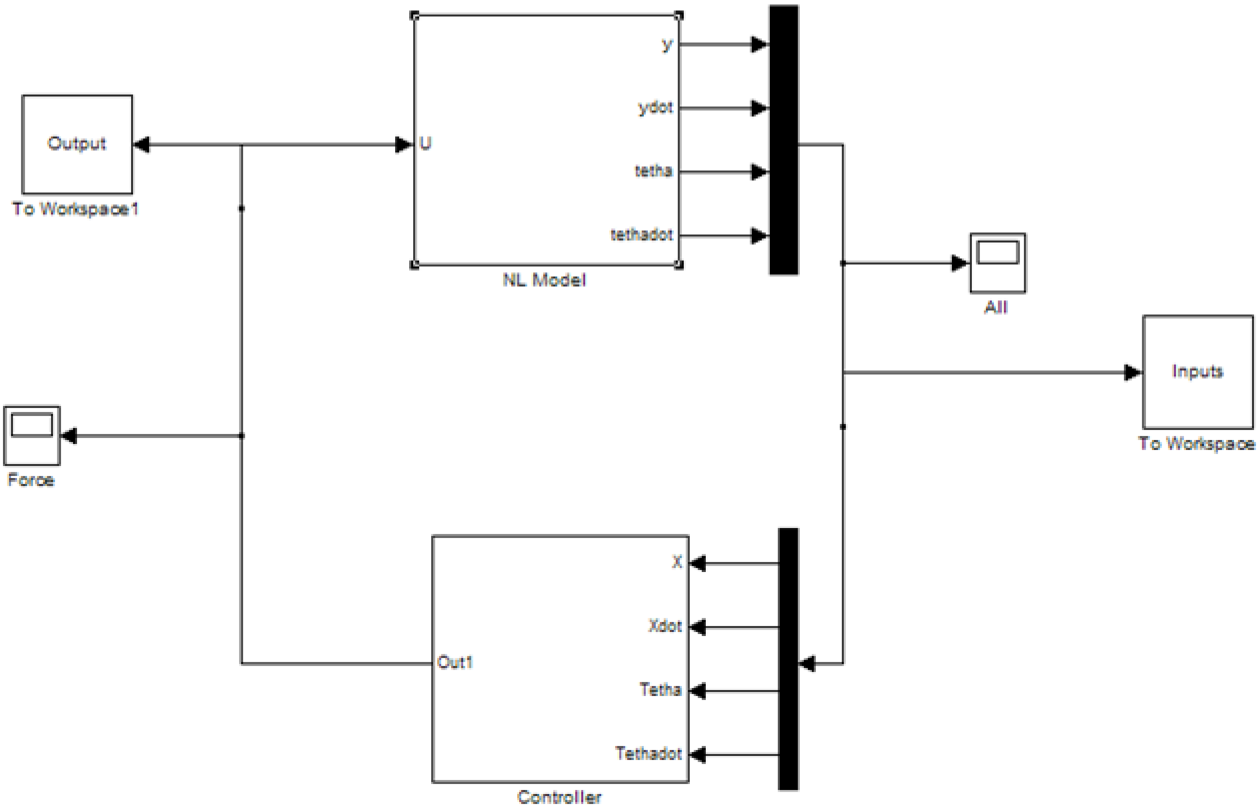


Fig.6.Simulink blocks of non-linear model of inverted pendulum and controller

D. Offline trained ANN Controller for Inverted pendulum

The offline ANN has been trained by using the training data generated by the feedback control law from the setup shown in the Fig.6, the offline ANN training specification is given as: network type is feed-forward, network training function is Levenberg-Marquardt algorithm (trainlm), input activation function sigmoid (tansig), output activation function is linear(purelin), no. of iterations (epochs) is 1500, learning rate is 0.1, momentum control factor to avoid the problem of local minima is 0.1, no. of neurons in input hidden and output layers are 4,13,1 respectively. After training the neural net with the above settings the simulation block has been generated using the command

Gensim (network_name, desired_sample_time) in the MATLAB command prompt. The simulation model of the trained neural network controller is shown in Fig.7.

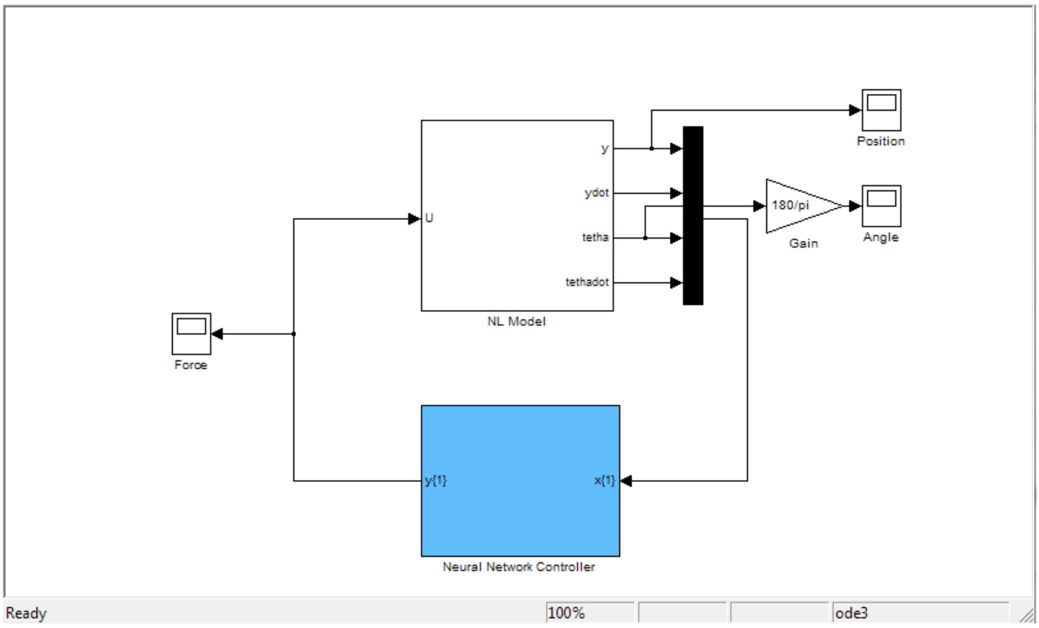


Fig.7. Simulation model of IP & offline ANN controller system

Fig.8 below shows the simulink setup of the adaptive learning. The error signal is equal to the output of the controller minus the output of the network. This signal is fed back into the GADALINE. The weights in the GADALINE are updated using the steepest descent gradient, which minimizes the square error between measurements and estimates. The learning rate of the GADALINE is set to 0.1 and the sample rate to 0.01.

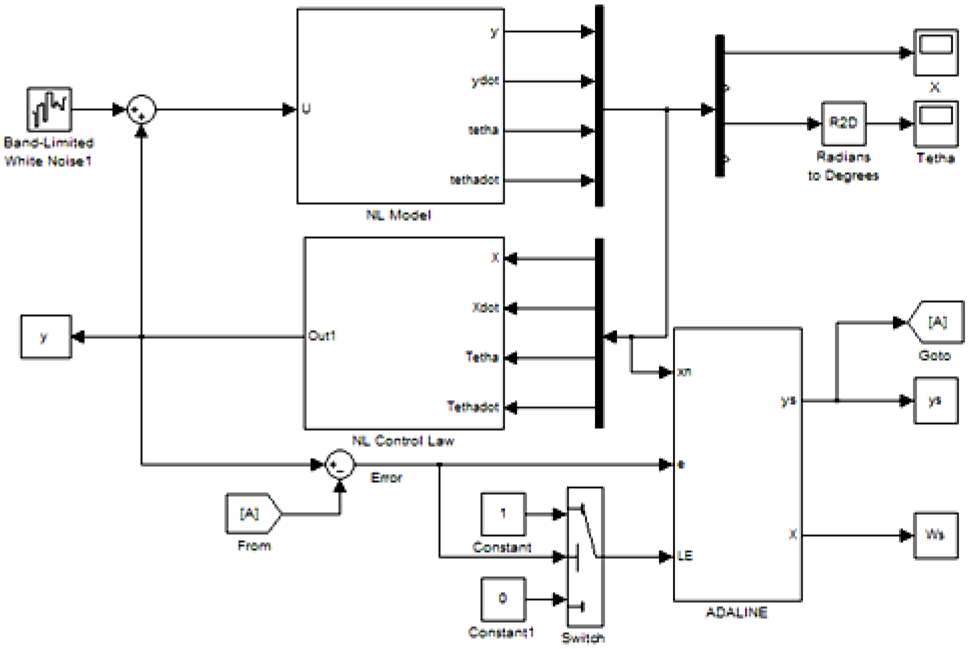


Fig.8 Simulation model of IP & Adaptive (ADALINE) ANN controller system

III. SIMULATION Results and Analysis

The simulation results show the closed loop response of inverted pendulum with online and offline controller systems. The simulation time is 10 seconds for all simulations with fixed step size solver of sample time 0.01 second. Three experiments are performed to demonstrate the online adaptability and controlling ability of the proposed neural network controller for the inverted pendulum system under the application of initial condition, band-limited noise and for some parameter variations of the inverted pendulum system.

In the first experiment the initial parameters of the inverted pendulum system are listed as follows: $x(0)$=0.0m, $\dot{x}(0) = 0.0m/s$, $\theta(0)$=0.0rad, $\dot{\theta}(0) = 0.0rad/s$ These values are putted in the initial condition column of the integrator blocks of nonlinear inverted- pendulum model corresponding to cart position, cart-velocity, pendulum angle and pendulum angler velocity respectively. The disturbance is introduced externally in the form of band limited white noise with noise power 0.1 sample time 0.01 and seed value of {16}.

The simulation setup for the first experiment is shown in Fig.8 with ADALINE block parameters are as follows: 4- inputs (cart position, cart-velocity, pendulum angle, and pendulum angular velocity), 1-output (force on cart), 1.0- learning rate, 1e-6 -stabilizing factor, Inf –weight limiter, 0-initial condition, 0.01-sample time. The corresponding Simulink results are shown in Fig.9. The experiment result shows that the ADALINE ANN controller adapts its weights to respond similar to the desired value.

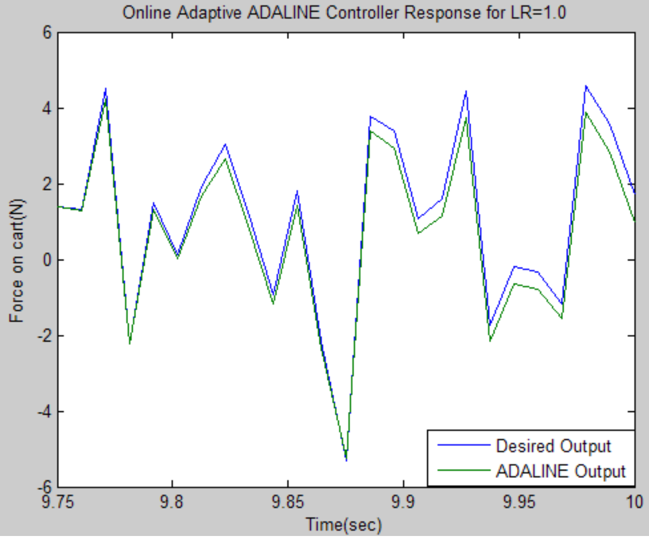


Fig.9 Simulation results of online adaptive ADALINE controller

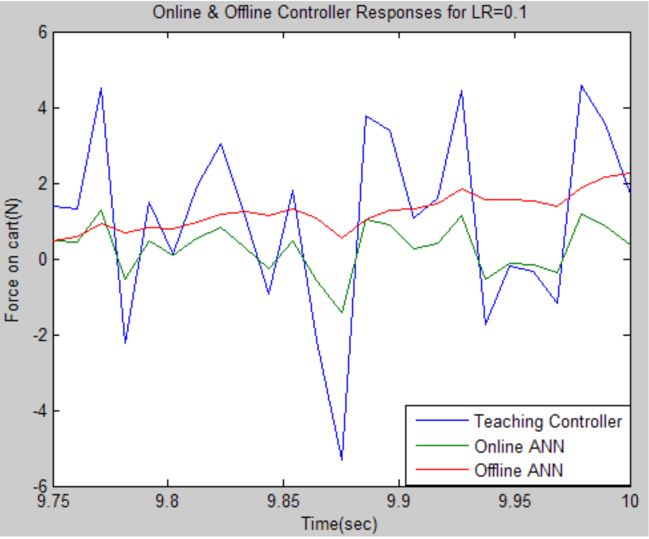


Fig.10 Simulation results for online and offline controllers

Now in second experiment the parameters are set to zero initial values as $x(0)$=0.0m, $\dot{x}(0) = 0.0m/s$, $\theta(0)$=0.0rad, $\dot{\theta}(0) = 0.0rad/s$, and the disturbance is introduced externally in the form of band limited white noise with noise power 0.1 sample time 0.01 and seed value of 16. The ADALINE block parameters are as follows: 4- inputs (cart position, cart-velocity, pendulum angle, and pendulum angular velocity), 1-output (force on cart), 0.1- learning rate, 1e-6 -stabilizing factor, Inf –weight limiter, 0-initial condition, 0.01-sample time. The simulation setup for the second experiment is shown in Fig.8 and results are shown in Fig.10 and Fig.11 from the results in Fig.10 it is clear that for learning rate of 0.1 online ANN responds better than the offline trained ANN and the performance of online trained ANN can be further improved if we increase the learning rate which is shown in Fig.11 that for learning rate of 0.5 and 1.0 the adaptive ADALINE ANN controllers output is very close to the reference controller.

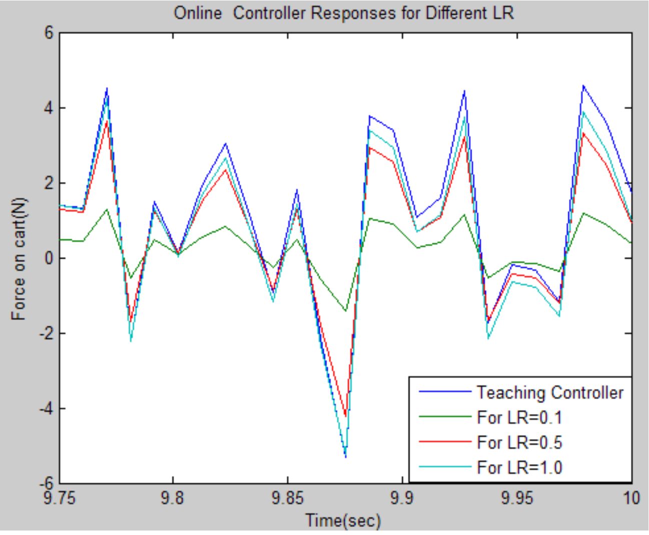


Fig.11. Simulation results of ADALINE for different learning rates

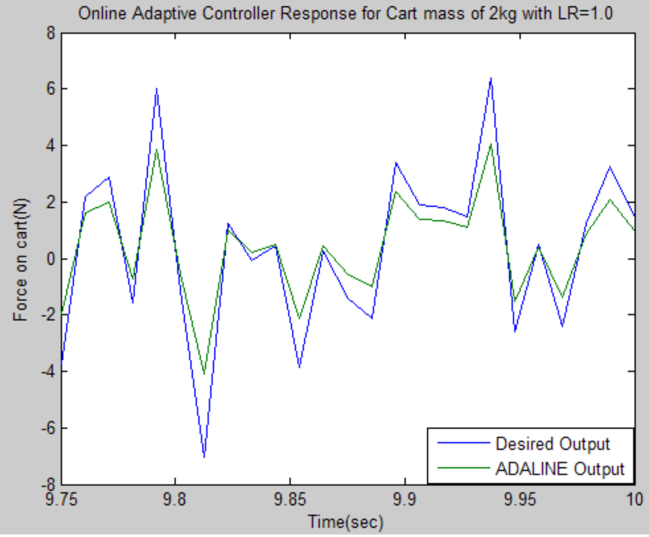


Fig.12. Simulation results for changed cart mass of 2kg with simulation time 10 seconds

Finally in third experiment the parameters are set to initial values given as $x(0)=0.0m$, $\dot{x}(0) = 0.0m/s$, $\theta(0)=0.0rad$, $\dot{\theta}(0) = 0.0rad/s$.In this experiment the value of cart mass M is increased nearly 100% and it is changed to $2kg$ the simulation setup is same as that of first experiment except that now the cart mass is $2kg$. The disturbance is introduced externally in the form of band limited white noise with noise power 0.1 sample time 0.01 and seed value of {16}. The ADALINE block parameters are as follows: 4- inputs (cart position, cart-velocity, pendulum angle, and pendulum angular velocity), 1-output (force on cart), 1.0- learning rate, 1e-6 -stabilizing factor, Inf –weight limiter, 0-initial condition, 0.01-sample time. Simulation result for the changed mass of cart is shown in Fig.12. Results shows that control action by online adaptive ADALINE is similar to the teaching controller and it can be improved if we increase the simulation time 20sec as shown in the Fig.13, the ADALINE output is now more close to reference output.

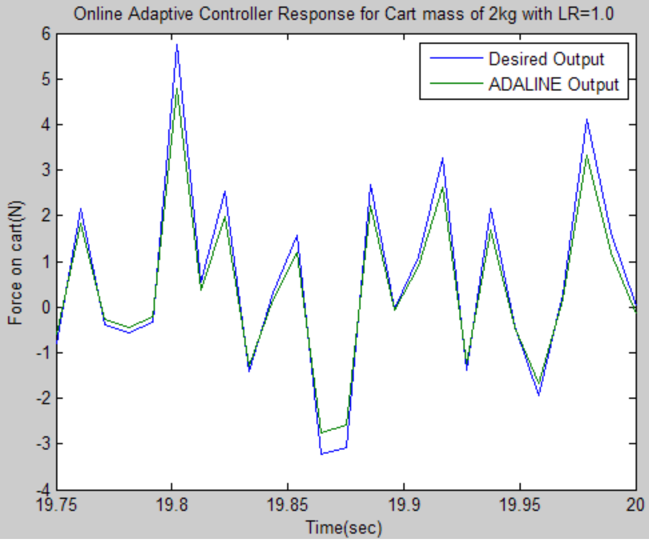


Fig.13. Simulation results for changed cart mass of 2kg with simulation time 20 seconds

IV. Conclusion

This paper introduced Online and offline artificial neural networks to control the inverted pendulum system. The advantage of neural network is its nonlinear nature to predict the nonlinearity in the system. To generate supervised data to train the ANN controller the IP system is stabilized by the control law. The data generated is used train both kind of neural network controllers so that they can mimic the output of control law by optimizing the linking weights and bias values. These trained neural network controllers are tested in MATLAB Simulink with nonlinear inverted pendulum model. Simulation results show that both online and offline ANN controller stabilizes the IP system under uncertain parameters and it is also able to handle external disturbances. ADALINE ANN proved to be very fast and accurate in online prediction and control. Based on the simulation results proposed ANN controllers can be used to control the real time IP system.

References

[1] C.S. Huang, "A neural network approach for structural identification and diagnosis of a building from seismic response data", *Earthquake Engineering and Structural Dynamics* , February 2003, pp. 187-20.

[2] T.Callinan , "Artificial neural network identification and control of the inverted pendulum", August 2003.

[3] X F. Ni, "A new method for identification and control of nonlinear dynamic systems", *Engng. Applic. Artif. Intell* , vol. 9, no. 3, March 1996, pp. 231-243.

[4] P.Gupta, "An improved approach for nonlinear system identification using neural networks". *Journal of the Franklin Institute*, May 1999, pp. 721-734.

[5] F. Meulenkamp, "Application of neural networks for the prediction of the unconfined compressive strength from Equotip hardness", *Int. J. of Rock mechanics and mining Sciences,* 36(1999),pp.29-39

[6] Xuemei Zhu "State feedback controller design of networked control systems with time delay in the plant", *International Journal of Innovative Computing, Information and Control* ,2008,4(2): pp.283-290.

[7] Li Shouju, Liu Yingxi "An improved approach to nonlinear dynamical system identification using PID neural networks" *International Journal Of Nonlinear Sciences And Numerical Simulation* 7 (2),2006:pp. 177-182.

[8] Radhakant Padhi, Prashant Prabhat and S. N. Balakrishnan "Reduced order optimal control

synthesis of a class of nonlinear distributed parameter systems using single network adaptive critic design" *International Journal of Innovative Computing, Information and Control*, 2008, 4(2): pp.457-469.

[9] Li Shouju, Liu Yingxi, "Application of improved neural network to nonlinear system control" *Inter. Transactions on Computer Science and Engineering*, 2006, 28(1):pp.27-36.

[10] Tadashi Kondo and Junji Ueno, "Multi-layered gmdh-type neural network self-selecting optimum neural network architecture and its application to 3-dimensional medical image, recognition of blood vessels" *International Journal of Innovative Computing, Information and Control*, 2008,4(1):pp. 175-187

[11] Wenle Zhang, "A generalized ADALINE neural network for system identification"*IEEE International Conference on Control and Automation*, Guangzhou,CHINA, 2007 pp. 2705-2709.

[12] Ching-Chih Tsai, Cheng-Kai Chan, Sen-Chung Shih, "Adaptive Nonlinear Control Using RBFNN for an Electric Unicycle". *IEEE International Conference on Systems, Man and Cybernetics*, 2008, pp. 2343-2348.

[13] Amit K. Singh, Dr.Prerna Gaur," Adaptive control for non-linear systems using Artificial Neural Network and its application applied on Inverted Pendulum" *IEEE Conference on Power Electronics (IICPE)*, 2011 , pp. 1 – 8.

[14] S.K. Oh, and W. Pedrycz, "Parameter estimation of fuzzy controller and its application to inverted pendulum", *Engineering Applications of Artificial Intelligence* , January 2004, pp. 37-60.

[15] The Mathworks, Inc. Version 7.8.0.347 (R2009a) License Number: 161051 February 12, 2009.

Sudeep Sharma was born in Bhopal, India in1986. He received bachelor of Engineering degree in Electronics and Communication Engineering from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal 2009. Currently he is pursuing M.Tech with specialization in System Modelling & Control in Electronics and Computer Dept. from Indian Institute of Technology Roorkee. His current areas of research includes Reinforcement learning, Neural Network, Fuzzy control.

Vijay Kumar was born in Biharshariff, India in 1954. He received B.E degree from M.M.M. College of engineering Gorakhpur, Masters and Ph.D. degree from University of Roorkee (with control and guidance specialization) in 1983 and 2001 respectively. He was lecturer in the department of Electronics and Computer engineering Indian Institute of Technology, Roorkee (formerly University of Roorkee) during 1983 to 2002, where currently he is working as Associate Professor. His area of research includes Robust Control, fuzzy control, order reductions.

Raj kumar was born in Agra, India in 1987. He received bachelor of Engineering degree in Electronics and Communication Engineering from G.L.A. Institute of Tech & Management, Mathura. Currently he is pursuing M.Tech with specialization in System Modeling & Control in Electronics and Computer Department from Indian Institute of Technology Roorkee. His area of research includes Fuzzy logic Control, Reinforcement Learning.

How to cite this paper: Sudeep Sharma,Vijay Kumar,Raj Kumar,"Supervised Online Adaptive Control of Inverted Pendulum System Using ADALINE Artificial Neural Network with Varying System Parameters and External Disturbance", International Journal of Intelligent Systems and Applications(IJISA), vol.4, no.8, pp.53-61, 2012. DOI: 10.5815/ijisa.2012.08.07