Study of System Identification method Using Adaptive Filter and Neural Network

A Thesis submitted in partial fulfillment of the requirements for the degree of
Bachelor of Technology in "Electrical Engineering"

By

Nishith Nirvan Tripathy
(109EE0312)

Under supervision of
Prof. Susmita Das

Department of Electrical Engineering
National Institute of Technology
Rourkela-769008 (ODISHA)

May-2013

DEPARTMENT OF ELECTRICAL ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

ODISHA, INDIA-769008

CERTIFICATE

This is to certify that the thesis entitled "**Study of System Identification Method using Adaptive Filter and Neural Network** ", submitted by **Nishith Nirvan Tripathy (109EE0312)** in partial fulfilment of the requirements for the award of **Bachelor of Technology** in **Electrical Engineering** during session 2012-2013 at National Institute of Technology, Rourkela. A bonafide record of research work carried out by them under my supervision and guidance.

The candidates have fulfilled all the prescribed requirements.

The Thesis which is based on candidates' own work, have not submitted elsewhere for a degree/diploma.

In my opinion, the thesis is of standard required for the award of a bachelor of technology degree in Electrical Engineering.

Place: Rourkela

Dept. of Electrical Engineering
National Institute of Technology
Rourkela-769008

Prof. Susmita Das
Professor

ACKNOWLEDGEMENTS

I have been highly indebted in the preparation of this report to my supervisor, Prof.Susmita Das, whose patience and kindness, as well as her academic experience, has been invaluable to me. I could not have asked for a better role model, inspirational, supportive, and patient guide. I could not be prouder of my academic roots and hope that I can in turn pass on the research values and the dreams that she has given to me.

The informal support and encouragement of many friends has been indispensable, and I would like particularly to acknowledge the contribution of all the students working under Prof.Susmita Das.

I would not have contemplated this road if not for my parents, who instilled within me a love of creative pursuits, science and language, all of which finds a place in this report.

Nishith Nirvan Tripathy
109EE0312

ABSTRACT

System Identification is an important way of investigating the world around with proper understanding.This paper deals with the System Identification of a given Black box in which the inputs and outputs are known.It is a method of deriving a mathematical model of a pre-defined part the world,using observations. We aim at reducing the error of the system which is also the cost function. We deal with various methods of training the system according to the given inputs and outputs. System identification include mathematical tools and algorithms that build dynamic models from measured data. The learning paradigm for a given system allows a system to emulate the functions of the environment it is embedded in.We come across a number of neural networks and how it functions in the process.Simulation results exhibits least mean square algorithm using an adaptive filter system modelled and simulated in the MATLAB/Simulink environment.

CONTENTS

CHAPTER 1

INTRODUCTION

CHAPTER 2

NEED OF SYSTEM IDENTIFICATION AND ITS METHODOLOGY

CHAPTER 3

NEURAL NETWORK AND LEARNING PARADIGM

CHAPTER 4

RESULTS AND DISCUSSIONS

CHAPTER 5

CONCLUSIONS

LIST OF FIGURES

LIST OF PLOTS

Chapter 1

INTRODUCTION

INTRODUCTION TO SYSTEM IDENTIFICATION

If a system and an input of a system is given, one can be able to find the output of the system. Its a problem that could be difficult, but it has a straight forward approach that will get you an answer. The approach might be different depending upon how the system is described, but the approach is always straight forward. It's not very likely that one will have problems in the job. If one does , there will probably be some sort of complication like a nonlinearity in the system - that makes the linear systems approach inapplicable. However, there are many times when one has an inverse problem. One may not know the system description, and need to figure out a description of the transfer function, its impulse response, its differential equations, state space equations, etc knowing only the input and output .

In case of linear systems, one can probably get the idea that one could always figure out a transfer function for any linear circuit. But there are many systems for which one cannot get a proper transfer function. An aircraft might have transfer function that differ widely with conditions such as airspeed, altitude, fuel load, atmospheric conditions and it may not be easy to compute those transfer functions from physical data. Again different plants are another example of something that one has to control but where one can't get a good handle on the transfer function of the system. In situations like that one may need to have some tools that will let oneself get a system description from a record of input and output signals.

The organization of the report is as follows. Chapter 2 depicts the background and literature review of the system identification and study of different neural networks. Chapter 3 explains System Identification,its need and modelling. Neural networks and the learning paradigm of the networks along with detailed mathematical analysis is provided in Chapter 4. Chapter 5 describes simulation results of different algorithms for reducing the cost function. Chapter 6 concludes the report followed by the references.

1.1 : BACKGROUND AND LITERATURE REVIEW

In control systems engineering, the branch of system identification deals with statistical methods to build mathematical and analytical models of dynamic systems from the already inferred data. System identification also includes the optimal design of a number of experiments for efficiently generating informative data which suitably fits such models as well as helps in model reduction. System Identification has its roots in standard statistical techniques and many of the basic routines have direct interpretetions as well known statistical methods such as least mean squares and maximum likelihood.Control community has taken an active part in developing and applying these basic technologies to dynamic systems right after the birth of modern control theory in the early 1960s. Maximum likelihood estimation was applied to difference equations and thereafter a wide range of estimation techniques and model para metrizations flourished.

Quality of system identification depends upon the quality of inputs, which are under control of a system engineer. Therefore, system engineers have long used the principles of design of experiments. In recent times, engineers are increasingly using the theory of optimal experimental design to specify the inputs that yield estimators with a high degree of accuracy and precision.

The literature on System Identification is extensive.In addition there are reports on neural network , its working. In any system the major aim is to to reduce the error function(cost function).

Research on adaptive filter started earlier than adaptive noise cancellation,that is around 1950s. The Least Mean Square algorithm (LMS) was one of the adaptive filter devised by Windrow and Holf in their study of pattern recognition scheme known as the adaptive filter element.Robbins and Monroe (1951) highlighted that the LMS algorithm was closely related to the concept of stochastic approximation.The difference between LMS and stochastic approximation was the usage of step size.LMS algorithm uses an appropriate step size estimator to control the correction which has been applied to each of the tap weights for each iteration , but in case of stochastic approximation methods the step size parameter is inversely proportional to time or to a power.

Neural Networks are general black-box structures.So,they can be used in system identification.However,using neural networks for system modeling is one of the many algorithms available for system identification.In Neural Network modeling,most of the work has been carried out on different dynamic systems as most real life problems are dynamic(logistic) in sense .Single layer neural network for linear system identification using gradient descent technique has been reported by by Bhama and Singh.The problem of non-linear system identification using multi-layer feed forward network technique trained by back propagation algorithm was proposed by Narendra and Parthasarathy.

In this thesis,a study is provide on the adaptive filter theory and neural network modeling for system identification.

CHAPTER 2

NEED OF SYSTEM IDENTIFICATION AND ITS METHODOLOGY

NEED OF SYSTEM IDENTIFICATION

System identification can be defined as a method which describes mathematical tools and algorithms that build dynamic models from predefined data. It can be done by adjusting parameters within a given system until its output matches possibly with the measured output. Applying system identification to structural models and engineering mechanics offers an effective means of validating structural models and design assumptions and many other more . in the context of engineering,a system refers to a structure or a part of a structure. Inputs and outputs are dynamic excitations and structural responses respectively and they are sampled at discrete time instants from real world when they can be contaminated with unwanted disturbances i.e,noise.[2]

2.1 : THE MODEL

Generally we assume,

system's input at time t = $u(t)$

system's output at time t = $y(t)$

basic equation between input and output is =

$y(t) + a_1\, y(t\text{-}1) + \ldots\ldots\ldots\ldots + a_n\, y(t\text{-}n) = b_1\, u(t\text{-}1) + \ldots\ldots\ldots\ldots + b_m\, u(t\text{-}m)$ ……(1)

The system is represented in discrete time primarily because observed data are collected by sampling.

A useful way is to view it as a way of determining the next output value given previous observations :-

$y(t) = -a_1\, y(t\text{-}1) - \ldots\ldots\ldots\ldots - a_n\, y(t\text{-}n) + b_1\, u(t\text{-}1) + \ldots\ldots\ldots\ldots + b_m\, u(t\text{-}m)$ ……(2)

In vector form'

$\theta = [a_1, a_2 \ldots\ldots a_n, b_1, b_2 \ldots\ldots\ldots b_m]$ and

$Ø(t) = [-y(t\text{-}1)\ldots\ldots\ldots\ldots -y(t\text{-}n)u(t\text{-}1)\ldots\ldots\ldots\ldots\ldots u(t\text{-}m)]$

So eqn 2 :-

$Y(t) = Ø(t)\, \theta$

2.2 : ADAPTIVE FILTER METHODOLOGY

Adaptive filter can be defined as a filter that self adjusts its transfer function according to an optimization algorithm. This section presents a brief description of how adaptive filters work and some of applications where they can be useful.Adaptive filters can self learn. As the signal into the filter is continued, adaptive filter coefficients adjust themselves to achieve the desired result, such as identifying an unknown system or canceling noise in input signal. In the figure below, the box represents an adaptive filter, comprising the adaptive filter itself and the adaptive recursive least squares (RLS) algorithm is the adaptive algorithm.[2,11]

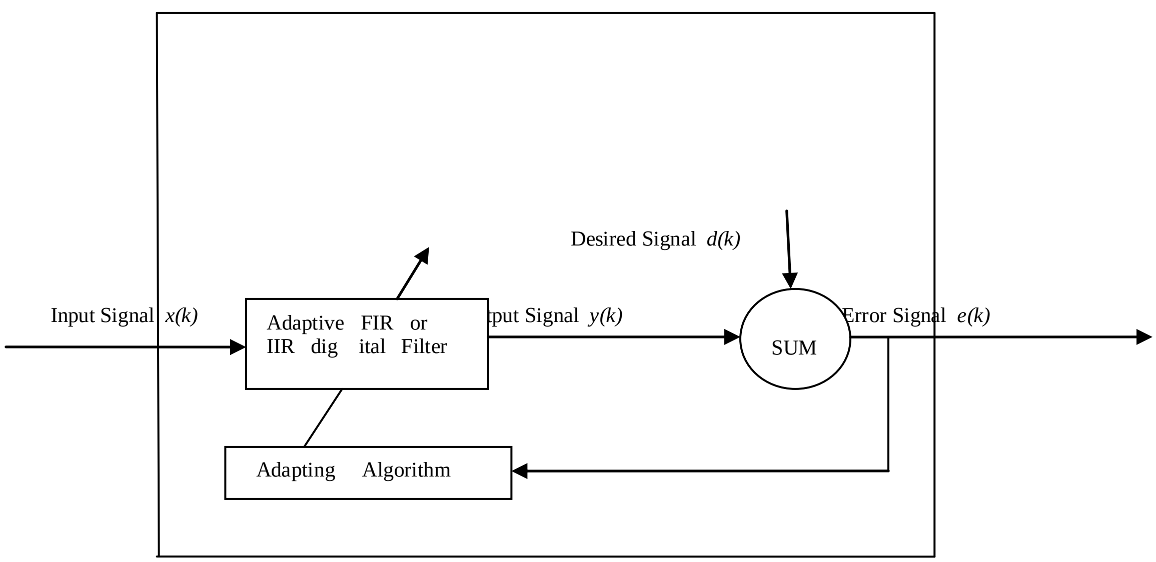


Fig 1 : Block Diagram Defining General Adaptive Filter Algorithm Inputs and Outputs

DSP System Toolbox software in MATLAB includes adaptive filters of a wide range of forms, all of which can be used for specific needs. Some of the them are as follows:

-Adaptive filters that are based on the least mean squares (LMS) algorithm, such as adaptfilt.lms,adaptfilt.filtxlms, and adaptfilt.nlms etc

-Adaptive filters that are based on the recursive least squares (RLS) techniques.Such as adaptfilt.rls, adaptfilt.swrls

An adaptive filter can design itself basing on characteristics of input signal to filter and a signal that shows the expected behavior of filter on its input.

Designing a filter doesn't require any other frequency response information/ specification. To describe the self-learning process a filter uses, one can select an adaptive algorithm that is used to reduce error between the output signal $y(k)$ and the desired signal $d(k)$.When a LMS performance criterion for $e(k)$ has reached its minimum possible value through a number of iterations of adapting algorithm used, the adaptive filter is finished and its coefficients have reached a solution. Now the output from the adaptive filter converges closely with the desired signal $d(k)$. When one changes the input data,called *filter environment*, the filter adapts the new environment by generating a set of different coefficients for new data. One can notice that when $e(k)$ tends towards zero and remains there one can achieve perfect adaptation, an ideal result which may not likely in the real world.Adaptive filter functions in this toolbox implements by replacing the adaptive algorithm with appropriate techniques. To utilize one of the functions, one needs to provide an input signal or signals and initial values for the filter.[6,2]

2.2.1 : Adaptive Algorithms

Recursive Least Square Filter

Recursive least squares (RLS)for an adaptive filter can be defined as an algorithm that recursively finds filter coefficients which minimize a weighted least squares cost function relating to input signals. This is in variation to the other algorithms such as the least mean squares (LMS) which aim at reducing mean square error. In derivation of RLS, input signals are considered deterministic, while in LMS and similar algorithm they are considered to be stochastic.When compared to most of its fellow competitors, the RLS shows an extremely fast convergence. However, this advantage comes at a cost of very high computational complexity.

Motivation

In general, the RLS can be used to solve any problem that can be solved by adaptive filters. Now let us take an example, suppose that signal d(n) is traversed over an echoey , noisy channel that causes it to be received as

$$x(n) = \sum_{k=0}^{q} b_n(k)\, d(n-k) + v(n)$$

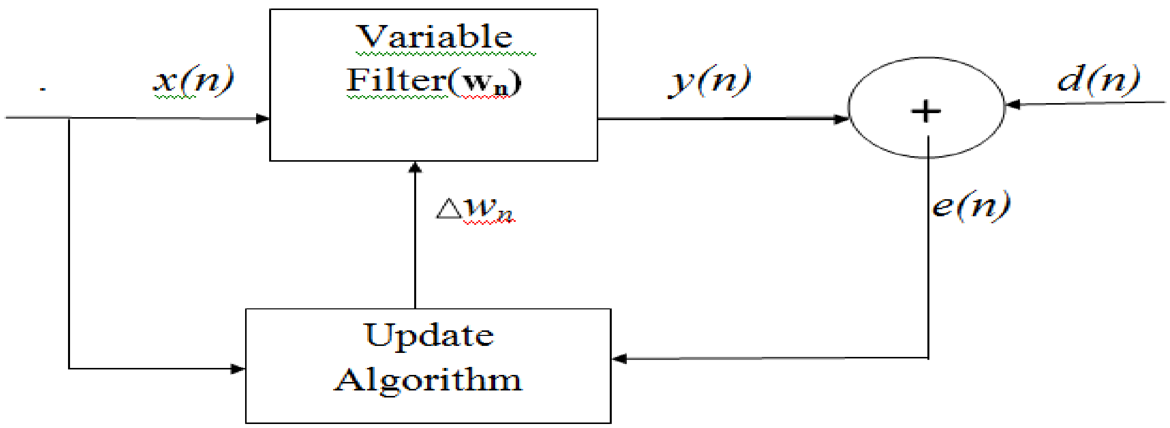
where v(n) represents the additive noise.We shall attempt to recover d(n) with the help of a p+1 tap filter,w :

$$y(n) = \sum_{k=0}^{p} w_n(k) x(n-k) = \mathbf{w_n}^{\mathbf{T}} \mathbf{x_n}$$

where $\mathbf{x_n} = [\ x(n)\ \ x(n-1)\ \ \ldots\ldots\ldots.x(n-p)]^T$ is the vector containing the *p* most recent samples of *x(n)*. Our goal is to estimate the parameters of the filter **w**, and at each time n we refer to the new least squares estimate by $\mathbf{w_n}$. As time evolves, we would like to avoid completely redoing the least squares algorithm to find the new estimate for $\mathbf{w_{n+1}}$, in terms of $\mathbf{w_n}$

Discussion

The idea behind RLS filters is to minimize a cost function *C* by appropriately selecting the filter coefficients $\mathbf{w_n}$, updating the filter as new data arrives.Error signal *e(n)* and the desired signal *d(n)* are defined in the feedback diagram below:



Negative Feedback Diagram

Error implicitly depends on filter coefficients through the *y(n) as follows* :

$e(n) = d(n) - y(n)$

Weighted least squares error function *C* is the cost function we desire to minimize . Being a function of e(n),C is therefore also dependent on filter coefficients:

$C(w_n) = \sum_{i=0}^{n} \lambda^{n-i} e^2(i)$

where $0 < \lambda \leq 1$ is "forgetting factor" which gives exponentially lesser weight to the older error samples.

The cost function can be minimized by taking partial derivatives for all entries k of coefficient vector $\mathbf{w_n}$ and setting results to zero :

$$\frac{\partial C(w_n)}{\partial w_n(k)} = \sum_{i=0}^{n} 2\lambda^{n-i} e(i) \frac{\partial e(i)}{\partial w_n(k)} = \sum_{i=0}^{n} 2\lambda^{n-i} e(i)x(i-k)=0$$

Rearranging the above equation yields

$$\sum_{l=0}^{p} w_n(l) \left[\sum_{i=0}^{n} \lambda^{n-i} x(i-l)x(i-k)\right] = \sum_{i=0}^{n} \lambda^{n-i}\, d(i)x(i-k)$$

This form is expressed in terms of matrices :

$$R_x(n)w_n = r_{dx}(n)$$

Where $R_x(n)$ is weighted sample co-relation matrix for x(n) and $r_{dx}(n)$ is equivalent

estimate for cross correlation between d(n) and x(n) .Based on this we can find the co-efficients that minimize cost function as :

$$w_n = R_x^{-1}(n)r_{dx}(n)$$

This is the main result of discussion.

Choosing λ

Smaller λ is, the smaller is the contribution of previous samples. This makes filter more sensitive to the most recent samples, which means more deviations in the filter co-efficients. λ = 1 case can be referred to as growing window RLS algorithm

2.3 : System Identification using adaptive filter methodology

One common application of adaptive filter is to use adaptive filters to identify unknown system, such as response of an unknown communications channel or frequency response of an auditorium,an utilization in system identification.Other applications include noise cancellation ,channel identification.In the given figure,unknown system is placed in parallel with adaptive filter. This layout represents one of the many possible structures. The shaded area is adaptive filter system.[1,4]

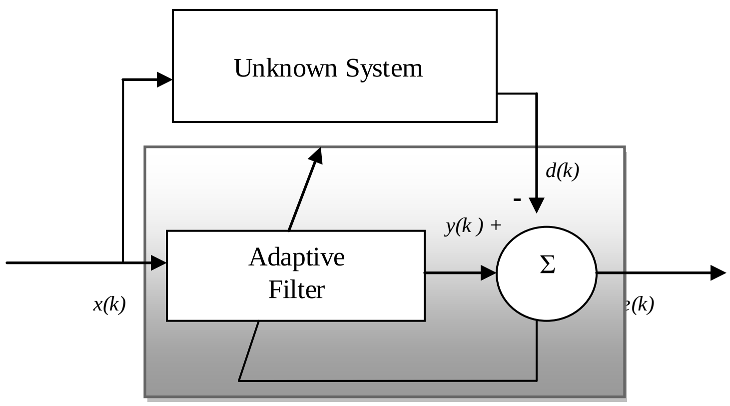


fig 3 : Using an Adaptive Filter to Identify an Unknown System

When *e(k)* is very small,adaptive filter response is closer to response of unknown system. In this case the same input is feeded to both adaptive filter and the unknown. If the unknown system is a modem, the input represents white noise, and is a part of the sound one hear from the modem when one logs into his Internet service provider.

2.2.3 : Inverse System Identification

By placing the unknown system in series with the adaptive filter, the filter adapts to become inverse of the unknown system as *e(k)* gradually becomes very small. As shown the process requires a delay which is inserted in the path of the desired signal *d(k)* to keep data at summation synchronized. Adding the delay helps in keeping the system causal.

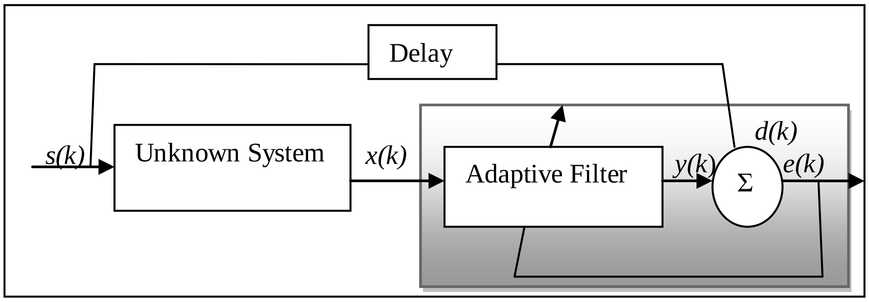


Fig 4 : Determining an Inverse Response to an Unknown System

Including delay in the system to account for delay caused by unknown system prevents this condition.Plain old telephone systems (POTS) are examples that commonly use inverse system identification method to compensate for copper transmission medium. When one sends data or voice over the telephone lines,copper wires behave as a filter, which has a response that rolls off at higher frequencies and have other anomalies as well.Adding adaptive filter which has a response that is inverse of wire response, and configures the filter to adapt, lets the filter to compensate for rolloff and other anomalies,increases available frequency output range and the data rate for telephone system(POTS).[11]

2.3.1 : Noise Cancellation

In noise cancellation, adaptive filters lets one remove noise from the signal in real time. Here, the desired signal,which one has to clean up, combines noise(echo) and desired information. To remove noise, a signal *n'(k)* is feeded to the adaptive filter that is correlated to the noise to be removed from desired signal.So as long as input noise to the filter remains correlated to the noise accompanying the desired signal, adaptive filter adjusts its coefficients accordingly to reduce value of the difference between *y(k)* and *d(k)*,thereby removing noise and resulting in a cleaning signal in *e(k)*. One can notice that in this application,error signal actually tends to match to the input data signal, but doesn't converges to zero.[1,4]

2.3.2 : Prediction

Predicting signals requires that one makes key assumptions. Such as assuming that the signal is either steadily or slowly varying with time, and periodic with time as well.

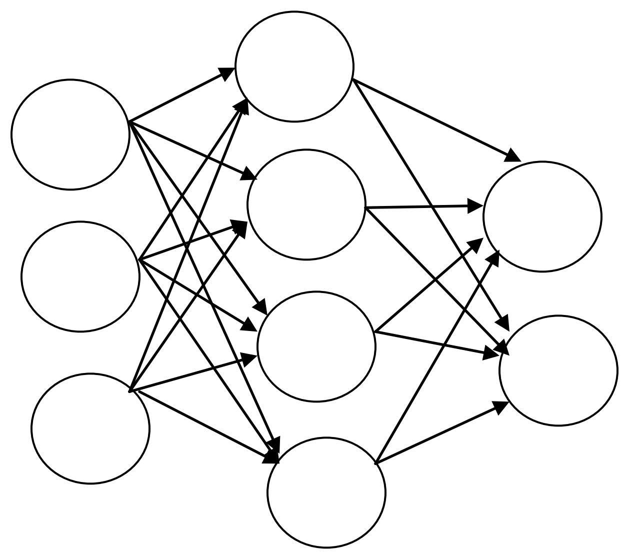
CHAPTER 3

NEURAL NETWORK AND LEARNING PARADIGM

3.1 : ARTIFICIAL NEURAL NETWORK

The Non-Linear System Identification of complex dynamic systems has potential applications ina broad range of areas such as control and communication.Because of its function approximation properties and learning capability,Artificial Neural Networks(ANNs) have become a useful tool for a wide variety of applications.The term neural network refers to a network of biological neurons.Its modern usage refers to a network or circuit called artificial neural network . An **Artificial Neural Network**, many times just called neural network, is mathematical modeling which inspired by the biological neural networks. A neural network is comprised of interconnected group of artificial neurons and it processes information(required data) using a connection approach in computation. In many cases a neural network structure is an adaptive filter system that changes its structure during a learning paradigm. Neural networks can be used to model complex relationships between inputs , outputs inorder to find patterns in data..[3] An artificial neural network thus basically contains a number of computing elements, which are called as neurons which perform a weighted sum of input signals and connecting weights. The sum is then added with a bias or threshold and the resultant is then passed through a non-linear activation function such as $\tanh(.)$ type,which is a logistic function. Each neuron is related with three different parameters on whose basis learning of neuron can be adjusted; these are connecting weights, bias and slope of non-linear function. From structural point of view, neural network (NN) may be single layer or it might be multi-layer. In a multi-layer structure, there might be more than one hidden layers and there is one/ many artificial neurons in each layer and for practical cases there might be a number of layers. Each neuron of a single layer is connected to each and every neuron of next layer.[9-10]

ANNs are thus capable of generating complex mapping between input and output space and arbitrarily complex non-linear decision making boundaries can be formed by these networks.



INPUT LAYER HIDDEN LAYER OUTPUT LAYER

Fig 5:Artificial Neural network

Neural network is charaterised by 4 properties :-

1.each neuron is represented by a set of linear synaptic links,a bias term non-linear activation link.

2.the synaptic weight represent the respective input signals.

3.the weighted sum of the input signals is the induced local field of the neuron.

4.the activation link squares the induced local field of neuron to produce an output.

3.2 : NEURON STRUCTURE

In 1958, Rosenblatt demonstrated some practical applications using the perceptron.The perceptron is a single level connection of McCulloh-Pitts neurons sometimes called single layer feed forward networks. The network is capable of linearly separating the input vectors into pattern of classes by a hyper plane.

A linear associative memory is an example of a single layered network. In such an application the network associates an output pattern(vector) and information is stored in network by virtue of modifications made to the synaptic weights of the network.[7]

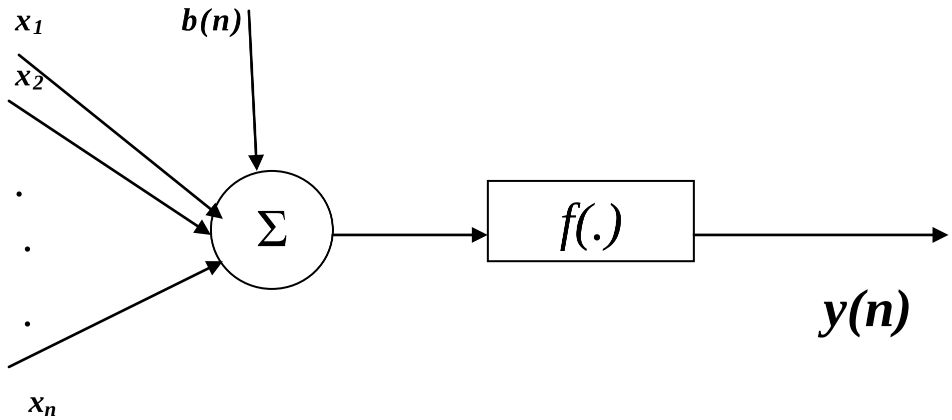


fig 6 : Neuron Structure

The structure of single neuron is represented in Fig. 3.1. An artificial neuron involves passing through a non-linear activation function. The output of the neuron can be represented as,

$y(n) = f[\Sigma w_j(n)\, x_j(n) + b(n)]$

here, $b(n)$ = threshold to neuron known as bias, $w_j(n)$ = weight which is associated with j_{th} input, and N = no. of inputs of neuron.[10]

Activation Function and Bias

The internal sum of inputs is passed through the activation function, which can be a monotonic function. Linear functions are used but these don't contribute to the non-linear transformations within a layered structure, which ultimately defeats the purpose of using a neural filter. A function which limits amplitude range and limits output strength of each neuron of a layered network to the defined range in non-linear manner will finally contribute to the nonlinear transformation. There are many activation functions, which are selected according to the specific needs of the problem. All the neural network structures employ an activation function which is defined as the output of a neuron in terms of activity level at its input (it ranges from -1 to 1 or 0 to 1). Table 3.1 shows the basic activation functions. The mostly used activation functions are the sigmoid functions and the hyperbolic tangent functions as they are differentiable.

NAME	MATHEMATICAL REPRESENTATION
Linear	$f(x) = kx$
Step	$f(x) = \alpha, \quad if\ x \geq k$ $\beta, \quad if\ x < k$
Sigmoid	$f(x) = 1/1+e^{-\alpha x}, \quad \alpha > 0$
Hyperbolic Tangent	$f(x) = 1-e^{-\gamma x} / 1+e^{-\gamma x}, \quad \gamma > 0$

Fig 7 : different activation functions

This bias gives network an extra variable and networks with bias are more powerful than those without bias. The neuron without a bias always gives a total input of zero to activation function when network inputs are zero. This is normally not desirable and is avoided by the use of a bias.[11,12]

3.3 : LEARNING PARADIGM

This property is of primary importance for a neural network modelling as its ability of network to learn from its neighbouring environment, and to improve its performance through learning accordingly. The betterment in performance takes place with time in accordance with some presdefined measures. A neural network learns about its surrounding environment through a process of adjustments that is applied to its synaptic weights and at bias levels. Ideally, the network becomes more intelligent about its environment after each iteration of the learning process. Hence one can define define learning as:

"A process by which the free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded."

There are three major learning paradigms . They are supervised learning, unsupervised learning and reinforcement learning.

3.3.1 : Supervised learning

One might think of a teacher as having knowledge of the surrounding environment, with that knowledge being shown by a set of input output system examples. The environment is, however unknown to neural network under consideration. Suppose now the teacher and the neural network are both exposed to a training algorithm, by virtue of already present knowledge, the teacher is able to provide neural network with a desired response for that training algorithm. In supervised learning, one is given a set of example pairs and the aim is to find the function present in the allowed class of functions that matches with the examples. In other words one wishes to *i*nfer the mapping that is implied by the data; the cost function relates the mismatch between mapping and the data and it implicitly contains prior knowledge about the problem. The network parameters which includes the weights and the thresholds that are chosen arbitrarily and are gradually updated during the training procedure inorder to minimize the difference between the desired and the measured signal. This updation can be carried out iteratively in a step-by-step procedure with an aim of eventually making the neural network emulate teacher. In this way knowledge of the surrounding environment that is available teacher is transferred to neural network. When this condition is reached, one may then dispense with teacher and let the neural network present deal with environment completely by itself. This is the way supervised learning acts.[8]

3.2.2 : Unsupervised learning

In unsupervised learning or which can be called as self-supervised learning there is no teacher present to over-see the learning process, rather steps are taken a task independent measure of quantity of representation that network is then required to learn, and free parameters of network are gradually optimized with respect to that measure. Once the network has turned to the statistical regularities of the input, it develops the ability to form internal representations for encoding features of input and thereby creates new classes by itself . In this learning weights and biases are updated in response to network input only. There are no desired outputs that are available. Most of these type of algorithms perform some kind of clustering operations. They learn themselves to categorize the input patterns into some classes.[12]

3.2.3 : Reinforcement learning

In reinforcement learning, data are usually not available, but generated through an agent's interactions with the surrounding environment. At each point of time the agent performs an action and the environment then generates an observation and an instantaneous cost function ,according to some (usually unknown) processes.

CHAPTER 4

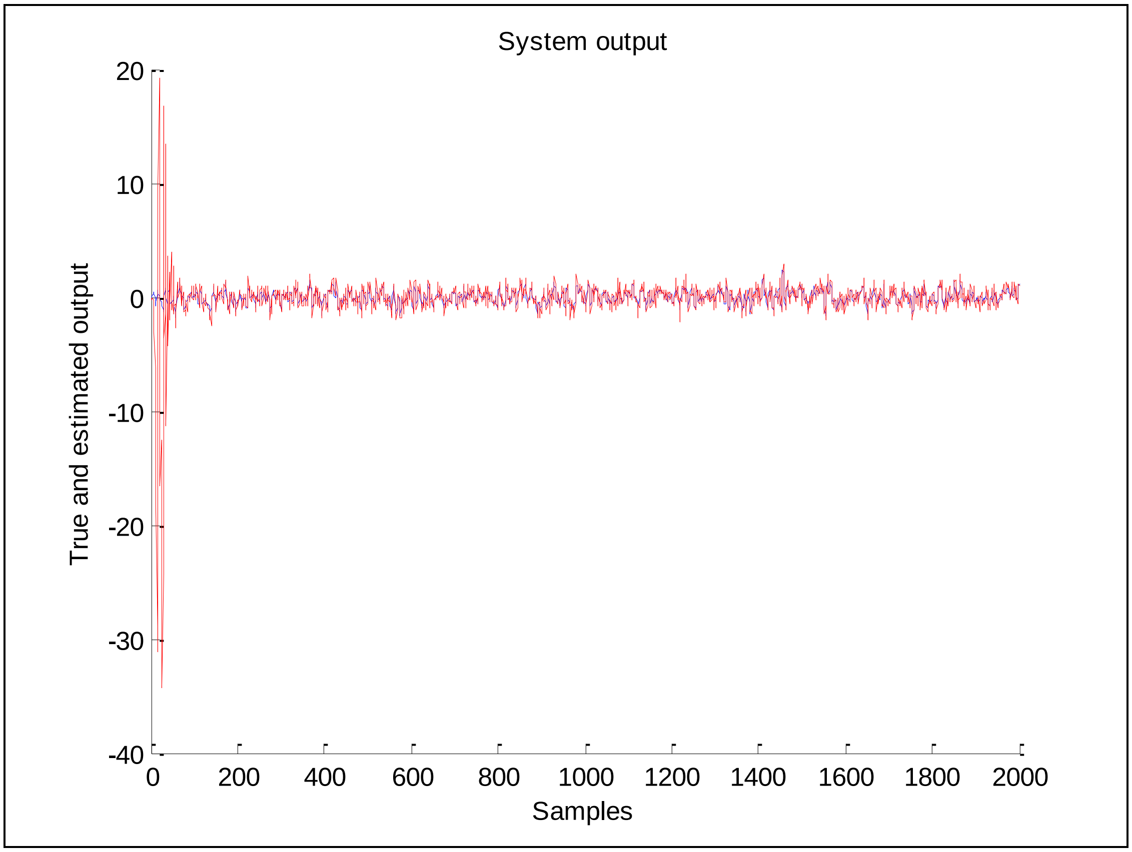
RESULTS AND DISCUSSIONS

CHAPTER 4: RESULTS AND DISCUSSIONS

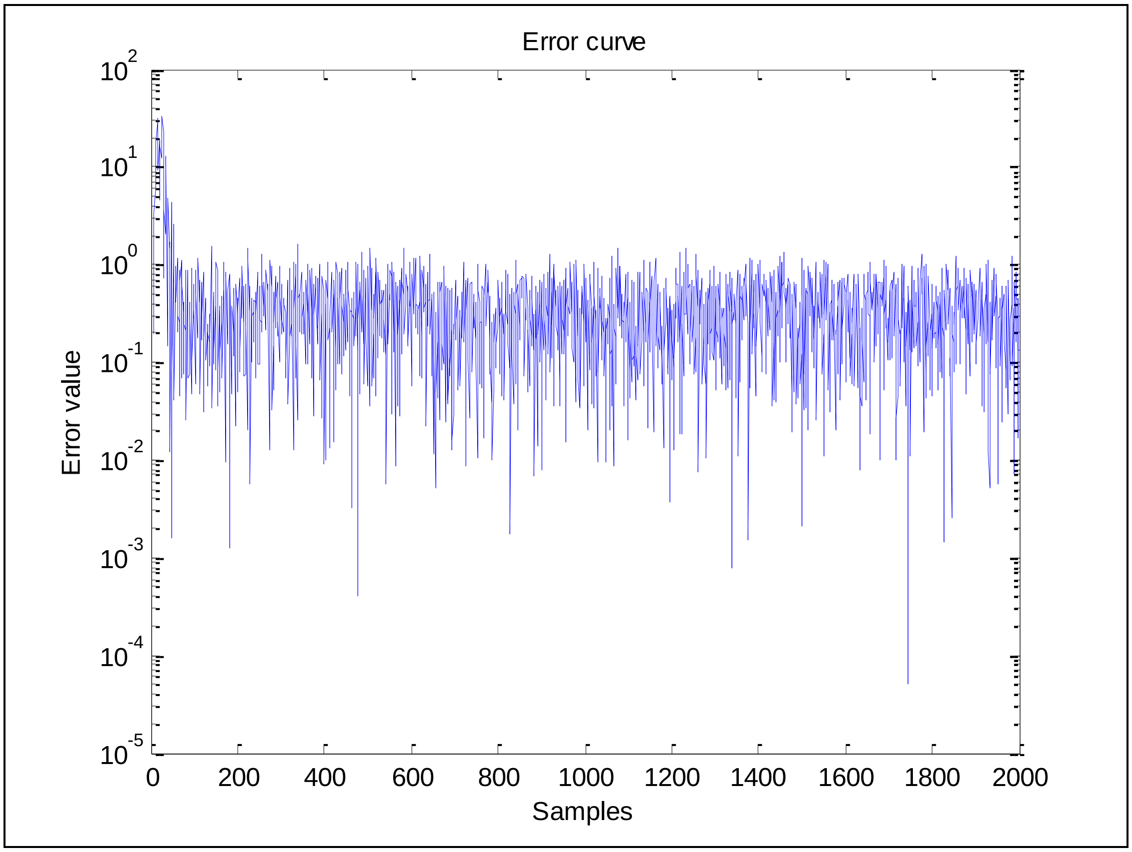
The toolbox provides many adaptive filter design functions for system identification of lms algorithm using adaptive filters.one such MATLAB program is written below and the simulation results as well.[1].The system is a fifth order transfer function.

```
clear all
close all
hold off
sysorder = 5 ;
N=2000;
inp = randn(N,1);
n = randn(N,1);
[b,a] = butter(2,0.25);
Gz = tf(b,a,-1);
h=  [0.0976;
    0.2873;
    0.3360;
    0.2210;
    0.0964;];
y = lsim(Gz,inp);
n = n * std(y)/(10*std(n));
d = y + n;
totallength=size(d,1);
N=60 ;
w = zeros ( sysorder  , 1 ) ;
for n = sysorder : N
    u = inp(n:-1:n-sysorder+1) ;
y(n)= w' * u;
e(n) = d(n) - y(n) ;
```

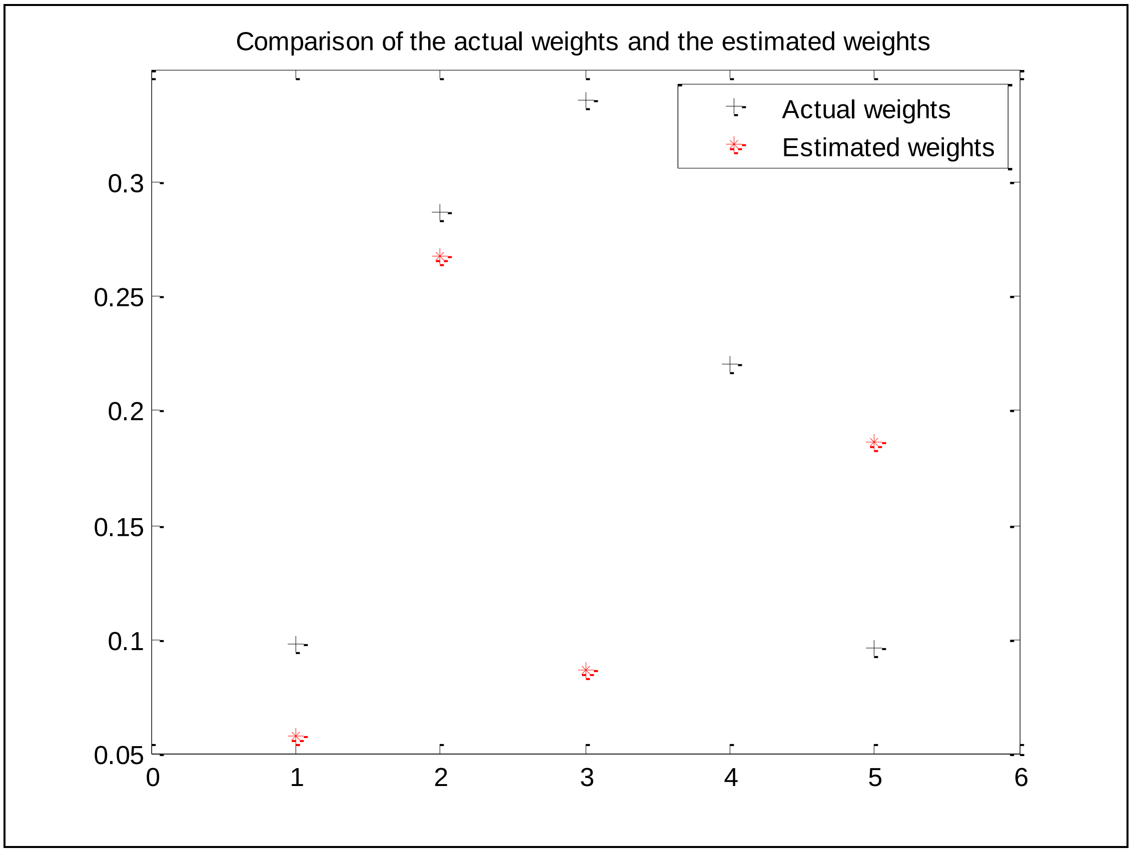
```
if n < 20
mu=0.32;
else
mu=0.15;
end
    w = w + mu * u * e(n) ;
end
for n =  N+1 : totallength
    u = inp(n:-1:n-sysorder+1) ;
y(n) = w' * u ;
e(n) = d(n) - y(n) ;
end
hold on
plot(d)
plot(y,'r');
title('System output') ;
xlabel('Samples')
ylabel('True and estimated output')
figure
semilogy((abs(e))) ;
title('Error curve') ;
xlabel('Samples')
ylabel('Error value')
figure
plot(h, 'k+')
holdon
plot(w, 'r*')
legend('Actual weights','Estimated weights')
title('Comparison of the actual weights and the estimated weights') ;
axis([0 6 0.05 0.35])
```



plot 1: the plot of the true and estimated output vs samples (no.of iterations)



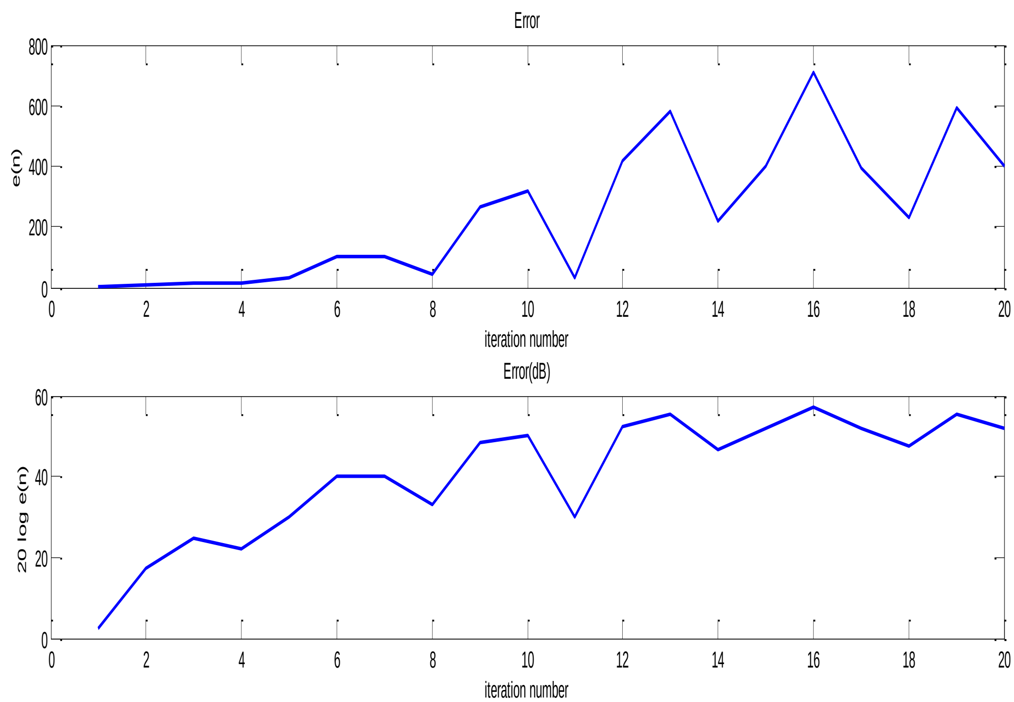
Plot 2 : the error curve for the given no of iterations



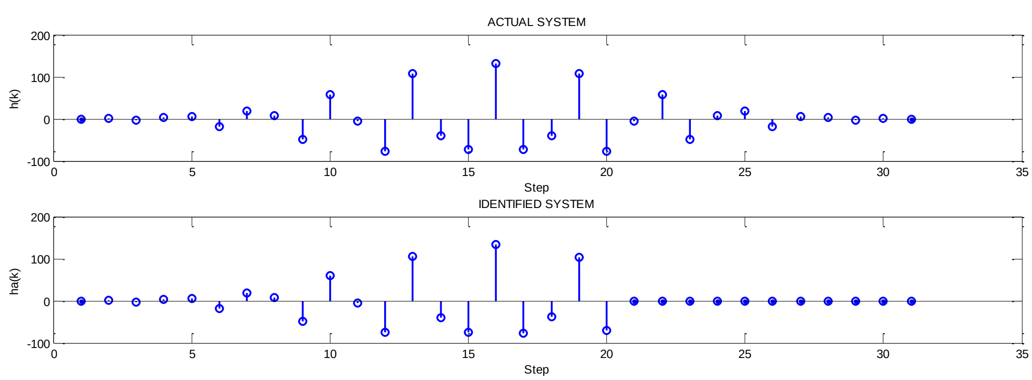
Plot 3 : comparision between actual weights and estimated weights

Now again when a particular system is taken with number of samples ,N=20, step size=0.1 and input signal=*5u(t)*,the two different adaptive algorithms give the following results :

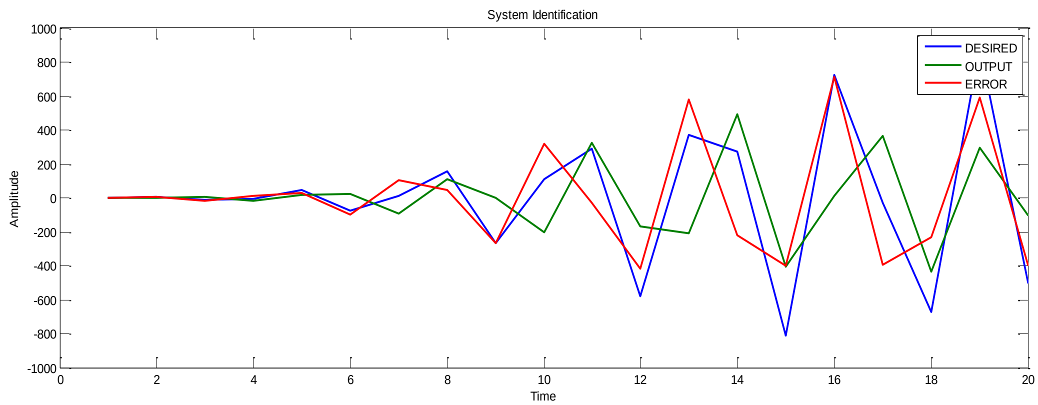
1 RLS Algorithm :



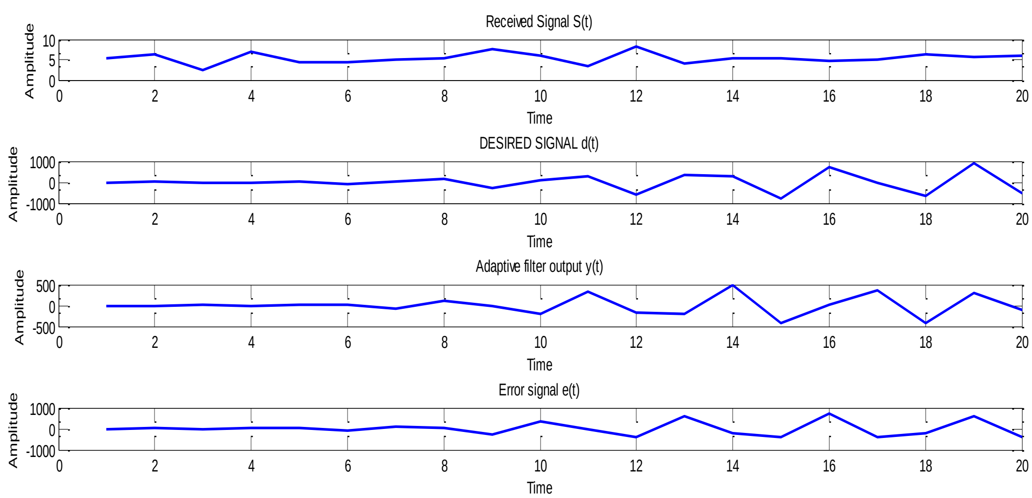
Plot 4 : Error signal response at different iterations



Plot 5 :Difference between the actual system and identified System

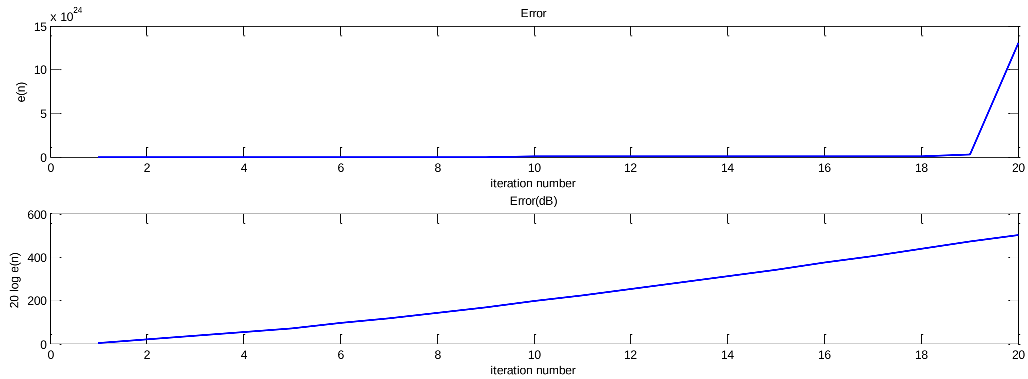


Plot 6 : System Identified along with time gradually

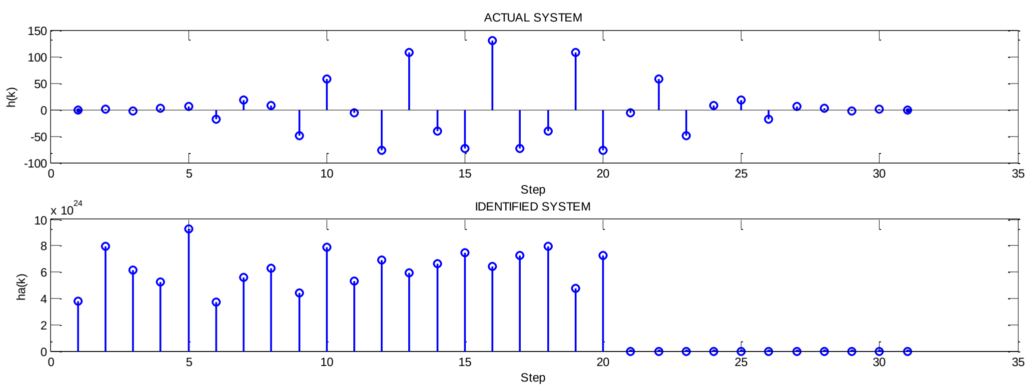


Plot 7 :Amplitudes of different signals along with time

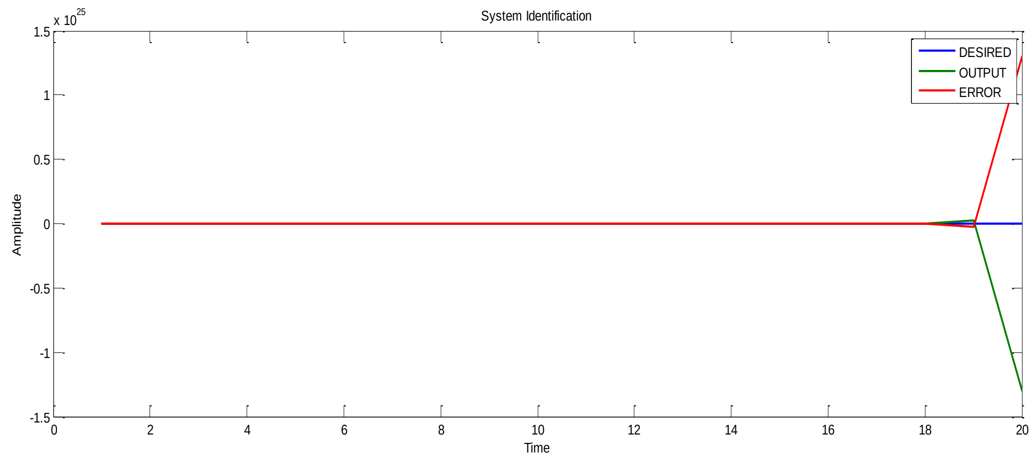
2.LMS Algorithm



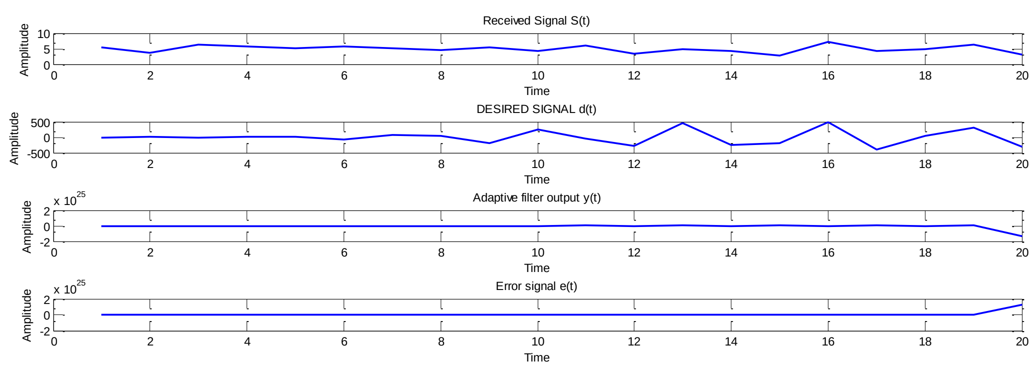
Plot 8 : Error signal response at different iterations



Plot 9:Difference between the actual system and identified System



Plot 10 : System Identified along with time gradually



Plot 11 :Amplitudes of different signals along with time

CHAPTER 5: CONCLUSIONS

This report provides an analytical background for system identification. We have studied and analyzed different adaptive algorithms for system identification. LMS algorithm is useful for practical implementation.RLS method is much more faster than the LMS methods but require larger number of floating point operationand has complexity in calculations.The quality of system identification depends on a various factors such as the quality of the inputs, which are under control of a systems engineer and the correctness of the output provided. Therefore, systems engineers have long used the principles of design of experiments. In recent times,the engineers are increasingly using the theory of optimal experimental design inorder to specify inputs which yield maximally precise estimators. It also describes about the artificial neural network and its learning paradigm.

The utility of artificial neural network models lies in the fact that they can be used to interpret a function from observations. This is particularly helpful in applications where the complexity of data or task makes the design of such a function or system by hand impractical.

Hence,Neural network modeling ,now-a-days is used to calculate, research, develop and apply artificial neural networks, in some cases even a wider array of adaptive systems.The utility of Neural Network modeling in system identification is on a rise these days and in the coming years it is bound to take over the field of system modeling and system identification in various fields of applications such as control,communications etc.

REFERENCES

[1]. Mamadou Mboup, Madeleine Bonnet, and Neil Bershad,Fellow. " *LMS Coupled Adaptive Prediction and System Identification: A Statistical Model and Transient Mean Analysis"*,.IEEE :10th oct,1994

[2]. lennart ljung. *"the control handbook"*. Department of electrical engineering Linkoping university Sweden

[3]. Forssell, Lindskog. " *Combining Semi-Physical and Neural Network Modeling: An Example of Its Usefulness"*

[4]. Daniel Graupe: " *Identification of Systems"*,Van Nostrand Reinhold, New York,1972 (2nd ed., Krieger Publ. Co., Malabar, FL, 1976)

[5]. J.-S. R. and Sun, C.-T. Jang,., " *Functional Equivalence between Radial basis Function Networks and Fuzzy Inference System*." IEEE Transaction on Neural networks, Vol. 4, pp. 156-159 :1993

[6].*International journal of adaptive control and signal processing,* Vol. 13, pp. 487 – 505, 1999

[7] .Sathyam and Bonala. *"Functional Equivalence between Radial basis Function Networks and Fuzzy Inference system."*Final Thesis Report :2009

[8]. K. S. Narendra and A. M. Annaswamy, "*Stable Adaptive Systems"*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[9] K. S. Narendra and K. Parthasarathy, *"Gradient methods for the optimization of dynamical systems containing neural networks,"* IEEE Trans. Neural Networks, vol. 2, pp. 252–262, 1991.

[10]. S. Bhama and H. Singh, "*Single layer neural networks for linear system identification using gradient descent technique,"* IEEE Trans. Neural Networks, vol. 4, pp. 884–888, 1993.

[11]. I. Kanellakopoulos, P. V. Kokotovi´c, and A. S. Morse, *"Systematic design of adaptive controllers for feedback linearizable systems,"* IEEE Trans. Automat. Contr., vol. 36, pp. 1241–1253, 1991.

[12]. K. S. Narendra and K. Parthasarathy, *"Identification and control of dynamical systems using neural networks,"* IEEE Trans. Neural Networks, vol. 1, pp. 4–26, 1990.