IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 4 Issue 6, June 2017
**ISSN (Online) 2348 – 7968 | Impact Factor (2016) – 5.264**
www.ijiset.com

# Adaptive Algorithms Performance Analysis for Noise Cancellations using Neural Networks employing real time signals

**A.M.Prasanna Kumar**
Research Scholar, ECE Department
ACS College of Engineering
Bengaluru, India

**Ramesha K**
Professor in ECE Department
Dr. Ambedkar Institute of Technology
Bengaluru, India

**Abstract**

Noise problems in signals have gained huge attention due to the need of noise-free output signal in mobile communication systems. The principal of adaptive noise cancellation is to acquire an estimation of the unwanted interfering signal and subtract it from the corrupted signal. We proposed signal extraction using Artificial Neural Network Hybrid Back Propagation Adaptive Algorithm (ANNHBPAA) for mobile Applications. ANNHBPAA exploits correlation between the pure speech signal and noise corrupted signal, to generate an estimate of the noise, which in turn subtracts the noise from the noise corrupted signal. The performance analysis of hybrid adaptive algorithms is done based on convergence behaviour, convergence time, correlation coefficients and signal to noise ratio. By taking into considerations of existing algorithms performance proposed hybrid adaptive algorithms gives better convergence behaviour, convergence time, correlation coefficients and signal to noise ratio. Proposed algorithm with only small training sets and it yields good results. Noise cancellation operation is controlled adaptively with the target of achieving improved signal to noise ratio. This paper concentrates upon the analysis of adaptive noise canceller using Recursive Least Square (RLS), Fast Transversal Recursive Least Square (FTRLS) and Gradient Adaptive Lattice (GAL) algorithms. The performance analysis of the algorithms is done based on convergence behaviour, convergence time, correlation coefficients and signal to noise ratio. After comparing all the simulated results we observed that GAL performs the best in noise cancellation in terms of Correlation Coefficient, SNR and Convergence Time. Here RLS Signal to noise ratio is made use as input to generate FTF and GAL signal to noise ratio at 30 DB, at 10 DB and also at -10 DB. Correlation coefficient and convergence time of RLS, FTF and GAL is also generated using neural network techniques.

***Keywords:*** *Adaptive Filter, Noise, Mean Square Error, RLS, FTF, GAL Neural network.*

## I Introduction

A digital communication system consists of a transmitter, channel and Receiver connected together. Typically the channel suffers from two major kinds of impairments: inter symbol interference and noise. The principle of noise cancellation is to obtain interfering signal estimation to subtract it from the corrupted signal. Adaptive noise cancellation [1, 2] using adaptive filter information is the right way. The primary signal serves as the desired response for the adaptive filter that uses reference signal as input. We have taken simulations result of three adaptive algorithms noise cancellation process. Efforts are made use to emphasize real time signals as input and audio signals of practical use. Audio files were read and microphones connected real audio signals. The analysis of the results give guidelines to further analysis**.**

At the end of this paper, a performance study has been done between these algorithms based on their parameters and also discussed the effect of epochs on converging error level.

## II Literature Review

Bernard Windrow, et al., [1] presented experimental analysis of classification of adaptive noise cancelling principles and applications. In this paper authors describes the concept of adaptive noise cancelling, an alternative method of estimating signals corrupted by additive noise or intererance. The method uses a primary input containing the corrupted Signal and a "reference" input containiug noise corre-lated in some unknown way with the primary noise. The reference input is adaptively filtered and subtracted from the primary input to obtain the signal estimate Adaptive filtering before subtraction allows the treatment of inputs that are deterministic or stochastic, stationary or time variable. Wiener sdutions are developed to describe asymptotic adaptive performance and output signal-to-noise ratio for stationary stochastic inputs, including single and multiple reference inputs. These solutions show that when the reference input is free of signal and certain other conditions are met noise in the primary input can be essentialIy eliminated without signal distortion. It is further shown that in treating periodic intererence the adaptive noise canceller acts as a notch filter with narrow bandwidth, infinite null, and the capability of tracking the exact frequency of the interference; in this case the canceller behaves as a linear, time invariant system.

Shubhra Dixit et al., [2] proposed Neural Network-based performance study on the recursive least squares algorithm. Authors recommended a novel method in which analog coefficients are varied for satisfactory performance. Analysis is done on performance factors depicted results showed that as the order of the filter increases, execution time of the recursive least square algorithm decreases. Reduction in processing time is achieved. This method works better for speech signal.

Hyun-Chool Shin et al., [3] developed an algorithm to estimate averaging analysis to study the mean-square performance of adaptive filters, not only in terms of stability conditions but also in terms of expressions for the mean-square error and the mean-square deviation of the filters, as well as in terms of the transient performance of the corresponding partially averaged systems. The treatment relies on energy con-servation arguments. Simulation results illustrate the analysis and the derived performance expressions.

Sayed. A. Hadei et al.,[5] presented a neural network-based application for noise cancellation, the changes in signal characteristics could be quite fast. This requires the utilization of adaptive algorithms, which converge rapidly.Least Mean Squares (LMS) and Normalized Least Mean Squares (NLMS) adaptive filters have been used in a wide range of signal processing application because of its simplicity in computation and implementation. The Recursive Least Squares (RLS) algorithm has established itself as the "ultimate" adaptive filtering algorithm in the sense that it is the adaptive filter exhibiting the best convergence behavior. Unfortunately, practical implementations of the algorithm are often associated with high computational complexity and/or poor numerical properties. Recently adaptive filtering was presented, have a nice tradeoff between complexity and the convergence speed. This paper describes a new approach for noise cancellation in speech enhancement using the two new adaptive filtering algorithms named fast affine projection algorithm and fast Euclidean direction search algorithms for attenuating noise in speech signals. The simulation results demonstrate the good performance of the two new algorithms in attenuating noise.

V.R.Vijaykumar et al.,[6] presented Multi-view classification considering speech is a very basic way for humans to convey information to one another with a bandwidth of only 4 kHz; The study shows the comparison of adaptive algorithms can be extended to the use of multi-dimensional adaptive filtering techniques; for applications like noise cancellation in images. Further, the modified algorithms proposed for optimized to have lower complexity. Speech convey information with the emotion of a human voice. The speech signal has certain properties. It is a one-dimensional signal, with time as its independent variable, it is random in nature, it is non-stationary and i.e. the frequency spectrum is not constant in time. Although human beings have an audible frequency range of 20Hz to 20 kHz, the human speech has significant frequency components only up to 4 kHz.

The most common problem addressed speech processing effect of interference noise in speech signals. Interference noise masks the speech signal and reduces its intelligibility. Interference noise can come from acoustical sources such as ventilation equipment, traffic, crowds and commonly, reverberation and echoes. It can also arise electronically from thermal noise, tape hiss or distortion products. If the sound system has unusually large peaks in its frequency response, the speech signal can even end up masking itself.

One relationship between the strength of the speech signal and the masking sound is called the signal-to-noise ratio, expressed in decibels. Ideally, the S/N ratio is greater than 0dB, indicating that the speech is louder than the noise. Just how much louder the speech needs to be in order to be understood varies with, among other things, the type and spectral content of the masking noise. The most uniformly effective mask is broadband noise. Although, narrow-band noise is less effective at masking speech than broadband noise, the degree of masking varies with frequency. High-frequency noise masks only the consonants, and its effectiveness as a mask decreases as the noise gets louder. But low-frequency noise is a much more effective mask when the noise is louder than the speech signal, and at high sound pressure levels it masks both vowels and consonants.

Lilatul Ferdouse et al., [7] presented noise problems in signals have gained huge attention due to the need of noise-free output signal in numerous communication systems. The principal of adaptive noise cancellation is to acquire an estimation of the unwanted interfering signal and subtract it from the corrupted signal. Noise cancellation operation is controlled adaptively with the target of achieving improved signal to noise ratio. This paper concentrates upon the analysis of adaptive noise canceller using Recursive Least Square (RLS), Fast Transversal Recursive Least Square (FTRLS) and Gradient Adaptive Lattice (GAL) algorithms. The performance analysis of the algorithms is done based on convergence behavior, convergence time, correlation coefficients and signal to noise ratio.

Vartika Anand et al., [8] proposed a system in current scenario of modern technology, we are facing a necessity of noise removal in signal processing. Various approaches are used for the same. This paper describes an intelligent adaptive filtering for noise cancellation. Here ANFIS method is being used for removal of noise from audio speech signals. An audio signal contaminated with noise is taken and inspected with eight types of membership functions: bell MF, triangle MF, Gaussian MF, two -sided MF, pi-shaped MF, product of two sigmoid MF, difference of two sigmoid MF and trapezoidal MF. Finally using ANFIS, the original audio speech signal is restored. The major advantage of this system is its ease of implementation and faster convergence rate.In this paper, adaptive noise cancellation using ANFIS has been implemented on audio speech signal. ANFIS is a intelligent method for real-time noise cancellation which is based on fuzzy logic and neural networks. This method is more efficient to eliminate noise and has faster convergence time, low computation load and fewer memory requirements.

## III Adaptive Algorithms

Adaptive filtering constitutes one of the core technologies in digital signal processing and finds numerous application areas in science as well as in industry. Adaptive filtering techniques are used in a wide range of applications such as noise cancellation. Noise cancellation is a common occurrence in today telecommunication systems. Recursive Least Squares (RLS) algorithm is capable of realising a rate of convergence that is much faster than LMS algorithm, because the RLS algorithm utilises all the information contained in the input data from the start of the adaptation up to the present. The standard RLS Algorithms, Fast Transversal RLS Algorithm Gradient Adaptive Lattice are some common algorithms [8].

Soft computing is a new approach to construct intelligent systems. The complex real world problems require intelligent systems that combine knowledge, techniques and methodologies from various sources. Neural networks recognize patterns and adapt themselves to cope with changing environments. Neural networks inference systems incorporate human knowledge and perform inferencing and decision making. Noise is an unwanted energy, which interferes with the desired signal. It can be suppressed with adaptive filters using signal processing. But if the noise frequency is same as the original signal then sometimes it also eliminates the desired signal. Therefore, noise cancellation is used which will not affect the desired signal. The basic principle of noise cancellation using Adaptive neural algorithms is to filter out an interference component by identifying the nonlinear model between a measurable noise source and the corresponding immeasurable interference.

This paper concentrates upon the analysis of adaptive noise canceller using new proposed hybrid adaptive algorits comprising all Recursive Least Square (RLS), Fast Transversal Recursive Least Square (FTRLS) and Gradient Adaptive Lattice (GAL) algorithms. The performance analysis of hybrid adaptive algorithms is done based on convergence behaviour, convergence time, correlation coefficients and signal to noise ratio. After comparing all the simulated results we observed that GAL performs the best in noise cancellation in terms of Correlation Coefficient, SNR and Convergence Time. RLS, FTRLS and GAL were never evaluated and compared before on their performance in noise cancellation in terms of the criteria considered here [6].

The basic adaptive algorithms which are widely used for performing weight updation of an adaptive filter are: the LMS (Least Mean Square), NLMS (Normalized Least Mean Square) and the RLS (Recursive Least Square) algorithm [1]. The performance of these adaptive algorithms is highly dependent on their filter order and signal condition [8].

A neural network is an artificial representation of the human brain that tries to simulate its learning process. An artificial neural network is often called a neural network or simply neural net.[] The network uses three layers. Input layer meant for input, and output layer is meant for required output and hidden layer is in between the two layers. Number of nodes in input layer and output layer depend on number of respective variable. Number of nodes in hidden layer depends on the nature of problem data.[1,3,5,7]. The Paper analyses taking signal to noise ratio (30DB) as input and generates five different cases. FIVE different calculations Cases (A, B, C, D & E) are explained here.
The paper tries to generate Signal to noise ratio (30 DB) of FTF, GAL, using RLS signal to noise ratio of 30DB real time signals as input.

### (i) General Adaptive Filter

The general set up of an adaptive-filtering environment is illustrated in Fig1. Where $k$ is the iteration number, $x(k)$ denotes the input signal, $y(k)$ is the adaptive-filter output signal, and $d(k)$ defines the desired signal. The error signal $e(k)$ is calculated as $d(k) - y(k)$. The error signal is then used to form a performance (or objective) function that is required by the adaptation algorithm in order to determine the appropriate updating of the filter coefficients. The minimization of the objective function implies that the adaptive-filter output signal is matching the desired signal in some sense.
The complete specification of an adaptive system, as shown in Fig.1,

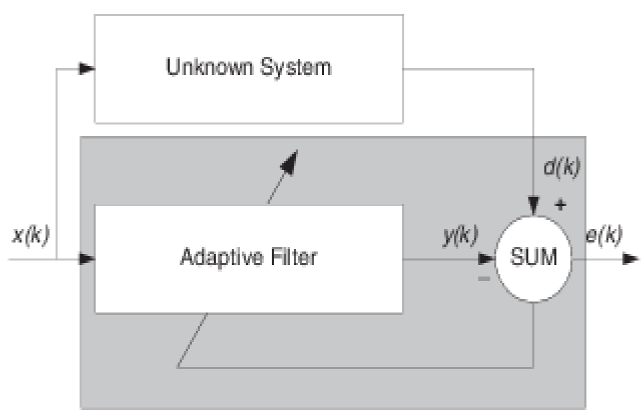


**Fig.1 General Adaptive-Filter Configuration [8]**

## (ii) Adaptive Filtering

Digital signal processing systems are attractive due to their low cost, reliability, accuracy, small physical sizes, and flexibility. One such digital signal processing system is called filter. Filtering is a signal processing operation whose objective is to process a signal in order to manipulate the information contained in the signal. For time-invariant filters the internal parameters and the structure of the filter are fixed, and if the filter is linear the output signal is a linear function of the input signal. If models are available for the signal and noise, it is possible to design a filter to enhance the signal relative to the noise. Many kinds of non-adaptive filters have been proposed. These include stack filters which are defined based on threshold decomposition and Boolean operations, and Volterra filters which are linear combinations of order stochastics. When the signal and noise models are not completely specified, it seems appropriate models could be estimated by analyzing the actual data. Especially, when the models are ill-defined or are time-varying. This leads to adaptive filters [1] [3].The general set up of an adaptive-filtering environment is illustrated in Fig1. Where $k$ is the iteration number, $x(k)$ denotes the input signal, $y(k)$ is the adaptive-filter output signal, and $d(k)$ defines the desired signal. The error signal $e(k)$ is calculated as $d(k) - y(k)$. The error signal is then used to form a performance.

## (iii) Hybrid Learning Back Propagation Algorithm

The process of determining the error rates of each neuron that impact the output is called back propagation. Figure 2 shows the typical structure of a back propagation Network. The neurons of the input layer are fully connected to the hidden layer and the outputs of the hidden layer are fully connected to the output layer. The error rate of one neuron affects the entire network. This procedure is crucial to the error minimization process, a process that will eventually lead to identification of the desired value. In back propagation a face image sample is propagated through the network producing an output. This is compared to the desired output, giving an error rate for the output layer. The error rates of a neuron being the function of the errors of units in the output layer, effects the layers below it. Due to this, the error propagates back to input layer through the hidden layer until the output reaches the desired value. Each neuron will make slight weight adjustments in order to minimize its error signal. The process is repeated for all the input values to be processed. In addition to the layered structure, BPN are also characterized as being feed-forward, which means that all the links in the network are unidirectional and the network is acyclic.

*Training Method:* The training of ANN is carried out in two parts. First, the feed-forward path is trained using the standard back propagation algorithm, until the feed-forward path is trained. The feedback path serves to alter the contribution that a given section of input makes to the outcome. This, in turn means that the feedback path must be taught to produce different signals depending on the initial output from the feed-forward algorithm. The feedback signals will vary depending on the stability of the sample input. Second, the training of the feedback path is conducted using a set of pairs consist of two face images. The use of these pairs facilitates the adjustment of the weights in the feedback path. The training phase is complete as soon as the feed-forward and feedback paths both have been trained. It should be noted that the two different parts of the network were trained separately.

Because the basic learning algorithm, backpropagation method, which we presented before, is based on the gradient method gives slow convergence and tendency to be trapped in local minima, a hybrid learning algorithm is introduced to speed up the learning process substantially.
Now the gradient method (also called backpropagation method) is combined with the least squares method to update the parameters of MFs in an adaptive inference system. Each epoch in the hybrid learning algorithm includes a forward pass and a backward pass. In the forward pass, the input data and functional signals go forward to calculate each node output. The functional signals still go forward until the error measure is calculated. In the backward pass, the error rates propagate from output end toward the input end, and the parameters are updated by the gradient method. The traditional feedforward neural network is a static structure which simply maps input to output.By modeling each synapse as a linear filter, the neural network as a whole thought of as an adaptive system with its own internal dynamics. An adaptive filtering procedure that could be used in the application of the neural network techniques is considered and a control system which adjusts parameters of the adaptive filter by means of a multilayered neural network designed.

This paper presents the use of Hybrid Back Propagation Algorithm for Adaptive Filter Applications.
The Artificial Neural Network (ANN) exploits correlation between the pure speech signal and echo corrupted signal, to generate an estimate of the echo, which in turn subtracts the noise from the echo corrupted signal.The LMS algorithm which is one of the most efficient criteria for determining the values of the adaptive noise cancellation coefficients are very important in communication systems, but the LMS adaptive noise cancellation suffers response degrades and slow convergence rate under low Signal-to-Noise ratio

(SNR) condition. This paper presents an adaptive noise canceller algorithm based on neural network. The major advantage of the proposed system is its ease of implementation and fast convergence. The proposed algorithm is applied to noise canceling problem of long distance communication channel. The simulation results showed its effectiveness.Filtering is a signal processing operation whose objective is to process a signal in order to manipulate the information contained in the signal. A digital filter is the one that processes discrete-time signals represented in digital format. For time-invariant filters the internal parameters and the structure of the filter are fixed, and if the filter is linear the output signal is a linear function of the input signal. Once prescribed specifications are given, the design of timeinvariant linear filters entails three basic steps, namely: the approximation of the specifications by a rational transfer function, the choice of an appropriate structure defining the algorithm, and the choice of the form of implementation for the algorithm [1].An adaptive filter is required when either the fixed specifications are unknown or the specifications cannot be satisfied by time-invariant filters. The adaptive filters are time-varying since their parameters are continually changing in order to meet a performance requirement. Adaptive filters are considered nonlinear systems; therefore their behavior analysis is more complicated than for fixed filters.

## IV Artificial Neural Network

Neural networks are composed of simple elements operating in parallel. The network function is determined largely by the connections between elements. Neural network can be trained to perform a particular function by adjusting the value of the connections (weights) between elements. Artificial Neural Network has preserved three basic characteristics. Neural network learns from experience; generalize from learned responses, and abstract essential pattern from inputs. Neural networks have been trained to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, vision and control systems [2].

ANNs are mathematical modeling tools which are particularly useful for predicting and forecasting in complex settings. The ANN accomplishes this through a large number of highly interconnected processing elements (neurons), working in unison to solve specific problems. Each neuron is connected to some of its neighbors with varying coefficients or weights which represent the relative influence of the different neuron inputs on other neurons [7]. ANN is the interconnection between the basic units called artificial neurons. An artificial neuron takes two inputs, multiplies them by a weight and adds them together. Each input link has an independent weight associated with it. If the sum of the weighted input's value is greater or equal to the threshold value, the output is equal to 1. If the sum is less than the threshold value, the output is 0. This is important, because it allows artificial neurons to compute the logical functions ANN relies on its ability to adjust its weights in order to associate each piece of input to the corresponding desired output. This ability to adapt proves especially helpful for problems where there is a finite set of outcomes, but no realistic way to represent all possible inputs. In the context of ANN is created by combining artificial neurons into a structure containing three layers. The first layer consists of neurons that are responsible for inputting a face image sample into the ANN. The second layer is a hidden layer which allows an ANN to perform the error reduction necessary to successfully achieve the desired output. Final layer is the output layer wherein the number of neurons in this layer is determined by the size of the set of desired outputs, with each possible output being represented by separate neuron.

## V The Adaptive LMS Algorithm

The LMS algorithm [2] is a stochastic gradient-based algorithm as it utilizes the gradient vector of the filter tap weights to converge on the optimal wiener solution. With each iteration of the LMS algorithm, the filter tap weights of the adaptive filter are updated according to the following formula:

$$w(n+1) = w(n) + 2\mu\, e(n)\, x(n) \quad (1)$$

Where, $x(n)$ is the input vector of time delayed input values, $w(n)$ represents the coefficients of the adaptive FIR filter tap weight vector at time n and $\mu$ is known as the step size. Selection of a suitable value for $\mu$ is imperative to the performance of the LMS algorithm, if the value is too small, the time adaptive filter takes to converge on the optimal solution will be too long; if $\mu$ is too large the adaptive filter becomes unstable and its output diverges.
Fig.2 represents a model for Adaptive Noise Cancellation.

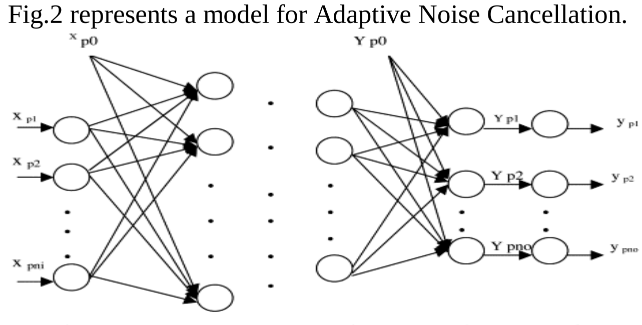


**Fig. 2. Schematic Diagram of Hybrid Learning Back Propagation Algorithm Adaptive Neural Network Model.**

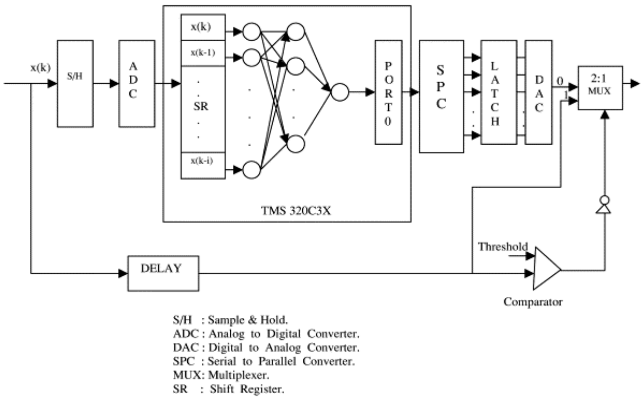


**Fig. 3. Block Diagram of Instrument Model for Adaptive Filtering Noisy Signals.**

## VI Adaptive Algorithm for Backpropagation

This method is used to find instantaneous value of gradient vector. Mean square error *e(n)* can be minimized by varying the weights of the filter. Optimal weiner solution is obtain for every iteration of the adaptive filter weights given in Equations (1) and (2)

$$w(n+1) = w(n) + 2\mu e(n)x(n) \quad (1)$$

$\mu$ represents step size, n is the time, *w(n)* gives the coefficients of adaptive filter and *x(n)* represents the input vector. Optimal value of $\mu$ is chosen to avoid large convergence time, instability and divergence of output.

$$W(n+1) = W(n) + \mu(n)e(n)X(n) \quad (2)$$

Mean square error *e(n)* is given in Equation (3)

$$e(n) = d(n) - W^T(n)X(n) \quad (3)$$

Input signal vector X*(n)* is given by

$$X(n) = [x(n)x(n-1) \dots\dots\dots.. x(n-L+1)]^T$$

LMS algorithms as follows

$$y(n) = \sum_{i=0}^{M-1} w_i(n)^* x(n-1) \quad (4)$$

$$e(n) = d(n) - y(n) \quad (5)$$

$$For \quad M = 1,$$

$$w_i(n+1) = w_i(n) + \mu^* e(n)^* x(n-1) \quad (6)$$

$$y(n) = w_0(n)^* x(n) \quad (7)$$

$$e(n) = d(n) - y(n) \quad (8)$$

$$w_0(n+1) = w_0(n) + \mu^* e(n)^* x(n) \quad (9)$$

Where,
$W_{(n)} = [w_0(n) w_1(n) \dots\dots w_{L-1}(n)]^T$ is the coefficient vector.

PROPOSED ADAPTIVE ALGORITHMS FOR NEURAL NETWORK HYBRID BACK PROPAGATION

Step 1
Normalize inputs and outputs for maximum values. Neural network perform better for input output values varies within 0-1.
For every training pair L inputs and n output is given by
$(I)_1 \div (I \times 1)$
$(O)_0 \div (n \times 1)$

Step 2
Assess neurons present in hidden layer between
$1 \times m < 2L$

Step 3
[V] Gives synapses weights into input neurons along with neurons of hidden layer and [W] gives weights of synapses linking hidden layer neurons with output neurons. Weights initialized to small values between -1 to and1. Threshold value taken as zero
$\lambda$ assumed as 1.
$[V]^0 = [random\ weights]$
$[W]^0 = [random\ weights]$
$[\lambda V]^0 = [\lambda W]^0 = [O]$

Step 4
Output of the input layer can be evaluated for one set of input and output as
$[O]_1 = [I]_1$
$L \times 1 = L \times 1$

Step 5
By computing multiplying corresponding weights of inputs to the hidden layer is given by
$[I]_H = [V]^T [O]_1$
$m \times 1 = m \times L = L \times 1$

Step 6
Using sigmoidal function compute output of hidden layer as
$[O]_H = 1 \div (1 + e^{-Im})$

Step 7
Determine the comparable weights of synapses as

$$[I]_O = [W]^T[O]_H$$
$$n \times 1 = (n \times m)(m \times 1)$$

Step 8
Using sigmoidal function enumerate network output
$$[O]_O = 1 \div (1 + e^{-Im})$$

Step 9
Quantify the error and subtract between desired output and the output of the network for $i^{th}$ training set

$$E^P = \sqrt{\Sigma(T_j - O_{oj})^2 \div n}$$

Step 10
Evaluate [d]as
$$[d] = (T_k - O_{pk})O_{ok}(1 - O_{ok})$$
$$N \times 1$$

Step 11
Assess[Y]matrix
$$[Y] = [O]_h(d)$$

Step 12
$$\lambda W = \alpha[\lambda W]^t + \eta[Y]$$
$$m \times 1 = m \times n = n \times 1$$

Step 13
Compute
$$[e] = [w][d]$$
$$d^* = e_i(O_{Hi})(1 - O_{Hi})$$
Calculate[x]matrix
$$[X] = [O]_I(d^*) = (I)_I(d^*)$$

Step 14
Enumerate
$$[\lambda V]^{t+1} = \alpha[\lambda V]^t + \eta[X]$$

Step 15
Find
$$[V] = [V]^t + [\lambda V]^{t+1}$$
$$[W]^{t+1} = [W]^t + [\lambda W]^{t+1}$$

Step 16
$$Error\ rate = \Sigma ep \div nset$$

Step 17
Repeat 4-16 steps until error rate converges within tolerance.

## VII Simulation Results

The Simulation based on four different types of signals mixed with various types of noise. Signals are periodic signal, audio signal, chirp signal and saw-tooth signal. Each signal has been subjected to some noise. Then the convergence behaviours of the RLS, FTF and GAL algorithms for these signals have been analyzed. Audio files were read and microphones connected real audio signals. The signals were then polluted by white, pink, grey and burst noise. We also apply AWGN channel model and take low, moderate and high signal-to-noise ratio. The signals were then passed through the simulation of the adaptive filter, and their error recovery rate, correlation coefficient and time were calculated. The analysis of the results offered useful insight into the characteristics of the algorithms. For the RLS algorithm, two parameters were varied to find their effect on the performance. One of them is the filter length, and the other is the forgetting factor. In Fast RLS algorithms, FTF and GAL, the performances were analyzed by varying filter related different parameters.

### (a) Comparison based

Mean square error with number of samples, error convergence characteristics, The paper discusses comparison of signal to noise ratio of RLS, FTF, and GAL algorithms when given SNR=30 DB considering fourdifferent type of signals. Here noise cancellation performance of the three algorithms.[8]Gradient Adaptive Algorithms (GAL) GAL shows the best performance Almost in every case of noise cancellation [4], RLS and FTF algorithm shows worse performance than GAL algorithms. For both of the algorithms, the output of SNR value are decreased.

All the simulations presented in this paper were performed using Matlab programming environment. The task has been accomplished using the new hybrid adaptive algorithms for adaptive noise cancellation process.

## VIII Analysis

A neural network is an artificial representation of the human brain that tries to simulate its learning process. An artificial neural network is often called a neural network or simply neural net.[2] The network uses three layers. Input layer meant for input, and output layer is meant for required output and hidden layer is in between the two layers. Number of nodes in input layer and output layer depend on number of respective variable. Number of nodes in hidden layer depends on the nature of problem data.[1,3,5,7]. The Paper analyses taking signal to noise ratio (30DB) as input and generates five different cases. FIVE different calculations Cases (A, B, C, D & E) are explained here.

The paper tries to generate Signal to noise ratio (30 DB) of FTF, GAL, using RLS signal to noise ratio of 30DB as input.

Simulated results of FTF and GAL (30 DB) are given in actual FTF and Actual GAL Column. Calculated resulted is given in respective predicted column & shown in Fig.4. [8]

Difference between actual and predicted is given in corresponding error Column. Given in case A shown in Table.1

| Case A : Hidden Layer 17 Neurons, 57,600 Iterations | | | | | | |
|---|---|---|---|---|---|---|
| | Actual FTF | Predicted | Error | Actual GAL | Predicted | Error |
| Chirp | 24.7287 | 25.8828 | 1.1541 | 68.74 | 69.5689 | 0.8289 |
| Sinusoidal | 14.513 | 16.7142 | 2.2012 | 72.7454 | 72.8571 | 0.1117 |
| Saw Tooth | 12.788 | 15.5497 | 2.7617 | 71.474 | 71.4839 | 0.0099 |
| Audio | 25.513 | 25.6120 | 0.0990 | 46.6549 | 46.6586 | 0.0037 |
| | | | | | | Total Error: 7.1702 |

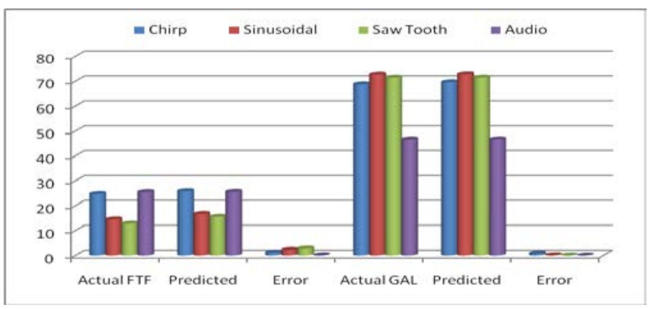


Fig.4: Hidden Layer 17 Neurons, 57,600 Iterations

Case **B:**

**The paper tries to generate signal to noise ratio(10 DB) of RLS, FTF, and GAL at 10 DB**

Using RLS signal to noise ratio of 30DB as input.

Simulated results of RLS, FTF, GAL at 10DB is given in the respective actual column

Calculated data is given in respective predicted column & shown in Fig.5

Difference between actual and predicted is given in corresponding error Column. Given in case B. shown in Table.2

| Case B : Hidden Layer 11 Neurons 85601 Iterations | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Actual RLS | Predicted | Error | Actual FTF | Predicted | Error | Actual GAL | Predicted | Error |
| Chirp | 9.7591 | 9.7740 | 0.0149 | 8.355 | 8.3525 | 0.0025 | 19.3497 | 19.3 | 0.0497 |
| Sinusoidal | 7.7829 | 7.7828 | -0.0001 | 7.7703 | 7.7702 | 1E-04 | 18.1702 | 18.0666 | 0.1036 |
| Saw Tooth | 7.5157 | 7.5005 | -0.0152 | 7.5087 | 7.5066 | 0.0021 | 20.4481 | 20.4321 | 0.016 |
| Audio | 8.2704 | 8.2898 | 0.0194 | 9.3365 | 9.3355 | 0.001 | 10.0652 | 10.0555 | 0.0097 |
| | | | | | | | | | Total error : 0.019 |

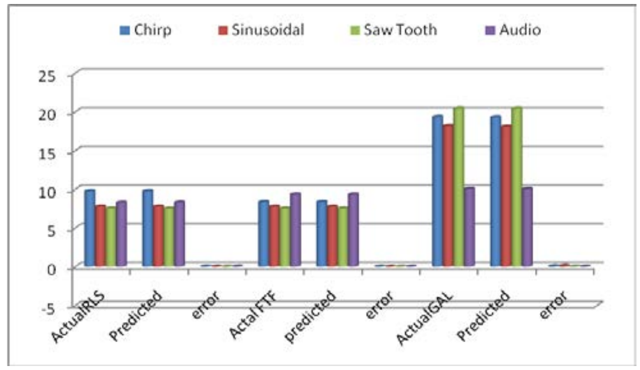


Fig.5: Hidden Layer 11 Neurons 85601 Iterations

Case **C:**

**The paper tries to generate signal to noise ratio (-10 DB) of RLS, FTF, and GAL at 10 DB**

Using RLS signal to noise ratio of 30DB as input.

Simulated results of RLS, FTF, and GAL at -10DB is given in the respective actual column & shown in Fig.6.Calculated data is given in respective predicted column.

Difference between actual and predicted is given in corresponding error Column. Given in case C shown in Table.3

| Case C : Hidden Layer 11 Neurons 85601 Iterations | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | RLS | Predicted | Error | Actual FTF | predicted | Error | Actual GAL | Predicted | Error |
| Chirp | 9.2383 | 9.2111 | 0.0272 | 9.3945 | 9.2214 | 0.1731 | 8.7799 | 8.7777 | 0.0022 |
| Sinusoidal | 9.7834 | 9.7688 | 0.0146 | 9.7822 | 9.6652 | 0.117 | 9.4333 | 9.333 | 0.1003 |
| Saw Tooth | 9.2232 | 9.227 | -0.0038 | 9.2027 | 9.222 | 0.0193 | 9.0513 | 9.0111 | 0.0402 |
| Audio | 9.8538 | 9.777 | 0.0768 | 10.0622 | 10.0521 | 0.0101 | 9.3343 | 9.2343 | 0.1 |
| | | | | | | | | | Total Error : 0.9765 |

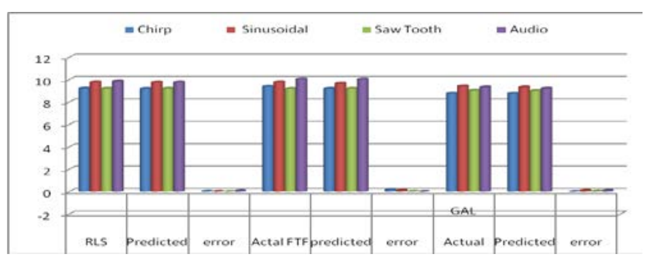


Fig.6: Hidden Layer 11 Neurons 85601 Iterations

Case **D:**

**The paper tries to generate signal to noise ratio Correlations Coefficient of RLS, FTF, and GAL**

Using RLS signal to noise ratio of 30DB as input.

Simulated results of Correlations Coefficient RLS, FTF, GAL is given in the respective actual column shown in Fig.7

Difference between actual and predicted is given in corresponding error Column. Given in case D.
Calculated data is given in respective predicted column shown in Table.4

| Case D : 6 Neurons in Hidden Layer, 60000 Iterations | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | RLS | Predicted | Error | Actual FTF | Predicted | Error | Actual GAL | Predicted | Error |
| Chirp | 0.8401 | 0.8392 | 0.0009 | 0.8501 | 0.8742 | 0.0241 | 0.9218 | 0.9213 | -0.0005 |
| Sinusoidal | 0.9464 | 0.9461 | 0.0003 | 0.9459 | 0.9734 | 0.0275 | 0.9422 | 0.9436 | 0.0014 |
| Saw Tooth | 0.8935 | 0.8934 | 0.0001 | 0.8909 | 0.8742 | 0.0167 | 0.9021 | 0.9024 | 0.0003 |
| Audio | 0.9798 | 0.9794 | 0.0004 | 0.9988 | 0.9734 | 0.0254 | 0.9989 | 0.9992 | 0.0003 |
| | | | | | | | | | Total Error : 0.0213 |

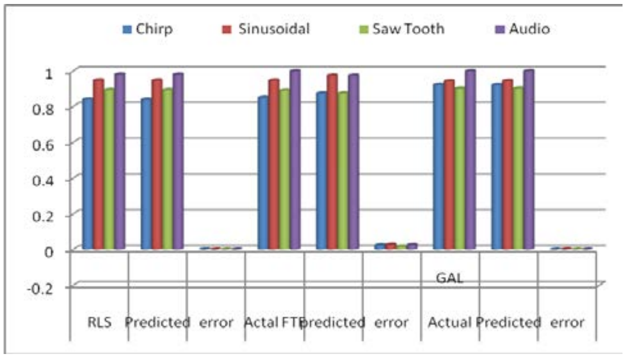


Fig.7: 6 neurons in hidden layer, 60000 iterations

Case E:

**The paper tries to generate Convergence Time of RLS, FTF, and GAL**

Using RLS signal to noise ratio of 30DB as input.

Simulated results of Convergence Time of RLS, FTF, GAL is given in the respective actual column

Calculated data is given in respective predicted column Fig.8
Difference between actual and predicted is given in corresponding error Column. Given in case E shown in Table.5.

| Case E : 6 Neurons in Hidden Layer, 60,000 Iterations | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | RLS | Predicted | Error | Actual FTF | Predicted | Error | Actual GAL | Predicted | Error |
| Chirp | 0.453 | 0.4403 | 0.0127 | 1.797 | 1.2211 | -0.5759 | 0.672 | 0.6725 | 0.0005 |
| Sinusoidal | 0.641 | 1.2211 | 0.5801 | 2.438 | 2.4380 | 0.0000 | 0.844 | 0.8434 | -0.0006 |
| Saw Tooth | 0.422 | 1.2211 | 0.5801 | 1.453 | 1.4534 | 0.0000 | 1.453 | 1.4530 | 0.0000 |
| | | | | | | | | | Total Error : 0.0615 |

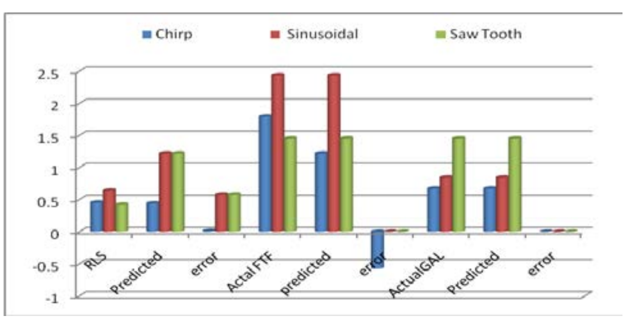


Fig.8: 6 Neurons in Hidden Layer 60,000 Iterations

## IX Conclusion

The proposed hybrid adaptive algorithms is an adaptive filter recursively finds the coefficients that minimize a weighted linear least squares cost function relating to the input signals. This is in contrast to other algorithms such as the least mean squares (LMS), Recursive least squares (RLS) that aim to reduce the mean square error. In the derivation of the hybrid adaptive algorithms the input signals are considered deterministic, while for the LMS, RLS and GAL similar algorithm they are considered stochastic.

Compared to most of its competitors, the RLS exhibits extremely fast convergence. However, this benefit comes at the cost of high computational complexity.Using RLS signal to noise ratio of 30DB as input. Neural network is used to generate FTF, GAL Signal to noise ratio at 30dB, 10dB and -10dB.

The performance analysis of hybrid adaptive algorithms is done based on convergence behaviour, convergence time, correlation coefficients and signal to noise ratio. After comparing all the simulated results tabulated.we observed that GAL performs the best in noise cancellation in terms of Correlation Coefficient, SNR and Convergence Time. RLS, FTRLS and GAL were never evaluated and compared before on their performance in noise cancellation in terms of the criteria considered

With persistent improvement of the adaptive hybrid algorithms and the rapid development of signal processing chip it will be more widely used in mobile telecommunication system, and signal processing fields. The simulation perception analysis of hybrid adaptive algorithms is carried out on the convergence behavior, correlation coefficient and convergence time. After comparing, simulated results were tabulated. By taking into considerations of existing algorithms performance of hybrid adaptive algorithms gives better convergence time, convergence behavior, correlation coefficients. Adaptive noise cancellation using hybrid adaptive algorithms implemented. This method is more systematic in eliminating noise from corrupted signal and has less time to converge, faster response and reduction in memory.

## X Acknowledgement

The author is deeply grateful to Hon.A.C.Shanmugam Chairman MCET and Hon.S.Vijayanand Executive Director, and Dr.M.S.Murali Principal, ACS College of Engineering Bangalore for their kind encouragement and help during the progress of this work.

## References


[1] Bernard Widrow, John R Glover, John M Mccool, John Kaunitz, Charles S Williams, Robert H Hearn, James R Zeidler, Eugene Dong Jr and Robert C Goodlin, "Adaptive Noise Cancelling Principles and Applications," *Proceedings of IEEE Spectrum*, vol.63, issue 12, pp. 1692-1716, December 1975.

[2] Shubhra Dixit and Deepak Nagaria "Neural Network Implementation of Least-Mean-Square Adaptive Noise Cancellation," *International Conference on Issues and Challenges in Intelligent Computing Techniques*, vol.1, pp. 134 – 139, Ghaziabad, India, February 2014.

[3] Hyun-Chool Shin, Ali H Sayed, "Mean Square Performance of Adaptive Filters using Averaging Theory," *Thirty-Eighth Asilomer IEEE International Conference on Signals, Systems and Computers*, vol.1, pp.229-234, 2004.

[4] Richard C North, James R Zeidier,Terence R Albert and Walter H Ku, "Comparison of Adaptive Lattice Filters to LMS Transversal filters for Sinusoidal Cancellation," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*,vol.4,pp. 33 – 36, San Francisco, CA, March 1992.

[5] Syed A Hadei and M Lotfizad, "A Family of Adaptive Filter Algorithms in Noise Cancellation for Speech Enhancement," *International Journal of Computer and Electrical Engineering*, vol.2 (2), pp.307-315, April 2010.

[6] V R Vijay Kumar, P T Vanathi and P Kanaga sapabathi, "Modified Adaptive Filtering Algorithm for Noise Cancellation in Speech Signals," *Electronics and Electrical Engineering*, Kanus Technologija, vol. 2 (74), pp.17-20, 2007.

[7] Lilatul Ferdouse, Nasrin Akhter, Tamanna Haque Nipa and Fariha Tasmin Jaigirdar, "Simulation and Performance Analysis of Adaptive Filtering Algorithms in Noise Cancellation," *International Journal of Computer Science Issues*, vol.8, issue 1, January 2011.

[8] Vartika Anand, Shalini Shah and Sunil Kumar, "Intelligent Adaptive Filtering for Noise Cancellation," *International Journal of Advanced Research in Electrical, Electronics & Instrumentation Engineering*, vol.2, issue 5, pp. 2029-2039, May 2013.



First Author

Prasanna Kumar A.M. obtained the B.E degree in E & C from Mysore University and M.E degree in Electronics from Gulbarga University. He is pursuing his PhD in E&C Engineering at Visvesvaraya Technological University Belgaum, under the guidance of Dr. K. Ramesha Professor, Department of Electronics and Communication Engineering, Dr. Ambedkar Institute of Technology, Bangalore. He has 2 research publication in Journal refereed International Journals and Conference Proceedings. He is currently working as Professor in Department of Electronics and Communication Engineering at ACS College of Engineering, Bangalore. His research interests include Adaptive Algorithms, Neural Networks and Mobile Applications. He is a life member of Indian Society for Technical Education, New Delhi. He is a Fellow member of IETE.

Second Author

Dr.K Ramesha is Professor in Department of Electronics and Communication Engineering at Dr.Ambedkar Institute of Technology, Bangalore. B.E degree in E & C awarded from Gulbarga University and M.Tech degree obtained in Electronics from Visvesvaraya Technological University Belgaum, He is awarded Ph.D in E&C Engineering at JNTU Hyderabad. His research interests include Image Processing, Biometrics, VLSI Signal Processing, Computer Networks and Computing. 18 papers published in international journals and conferences.