International Conference on the Contribution of Engineering, Sciences and Humanities for the Development of the Society
Genba Spoanrao Moze College of Engineering, Pune, India ESH-18
13th-14th October 2018 www.conferenceworld.in ISBN : 978-93-87793-50-7

# IMPROVED FAST BLOCK LMS ADAPTIVE ALGORITHM USING LABVIEW

**Akanksha Parihar[1], Jyotsna V Ogale[2], Aman Sharma[3]**

[1,3]*Electronics & Communication Engineering*

[2]*Electronics & Instrumentation Engineering*

*Samrat Ashok Technological Institute, Vidisha (M.P.)-464001, India*

**ABSTRACT-** *Adaptive filters are widely used for the filtration of noisy signal in a communication system. Due to lack of any unique solution to Adaptive Filters, many types of Adaptive Algorithms exits. The works presented in this paper aims towards improved implementation of Fast Block LMS Adaptive Algorithm (FBLMS) with some modifications to reduce the impact of noise on received signal. The proposed Algorithm is implemented using Virtual Instrumentation Tool LabVIEW. The simulation are performed on Chirp Signal and Audio signal (Acquired in real time such as human voice and Environmental Sources), the results obtained are compared to standard Fast Block LMS Algorithm, showing certain improvements.*

***Keywords***-*Chirp Signal, Audio signal (Acquired in real-time), Adaptive Filter, Fast Block LMS Algorithm, LabVIEW, LMS Algorithm.*

## 1. INTRODUCTION

### 1.1 Communication System

The purpose of a Communication Systemis that of transferring information from one point (Source/Transmitter) to another(Intended point/Receiver).As shown in Fig. 1 below, A Basic Communication System may consist of Transmitter, Channel and Receiver. Any stage of the system can be affected by Noise, but the most sensitive part is channel being used to transmit the information.

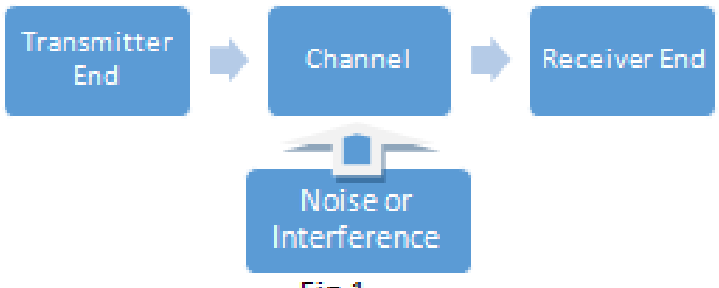



Fig.1

Noise or Interference is the unwanted signal which can distort the information present in the transmitted signal. This can happen during transmission over a channel. Therefore the analysis of received noisy signal and Noise Removal is a very important process, so that noise free signal can be obtained at the receiver end.

Noise can be removed with the help of filtering at receiver end. A Filter is a device that passes the useful components and removes the unwanted components from the input signal. The components can be in terms of frequency or amplitude. The input signal can be in any form i.e. audio, video, speech and image. In our work, Chirp signal is used as basis for developing the algorithm, and then actual Audio signal obtained from varied sources like human voice and sound produced from household equipment are used as an input signal for determining the performance of the developed algorithm.

### 1.2 Adaptive Filters

Adaptive filtering is a process where filter coefficients are adjusted/adapted in such a manner, that we can obtain the best possible information from the received noisy signal, with minimal changes to the characteristics of original signal. An adaptive filter is a device which maintain a relationship between input and output signal then self-adjust the coefficients of the filter according to its adaptive algorithm. Fig. 2 shows the diagram of a linear adaptive filter.

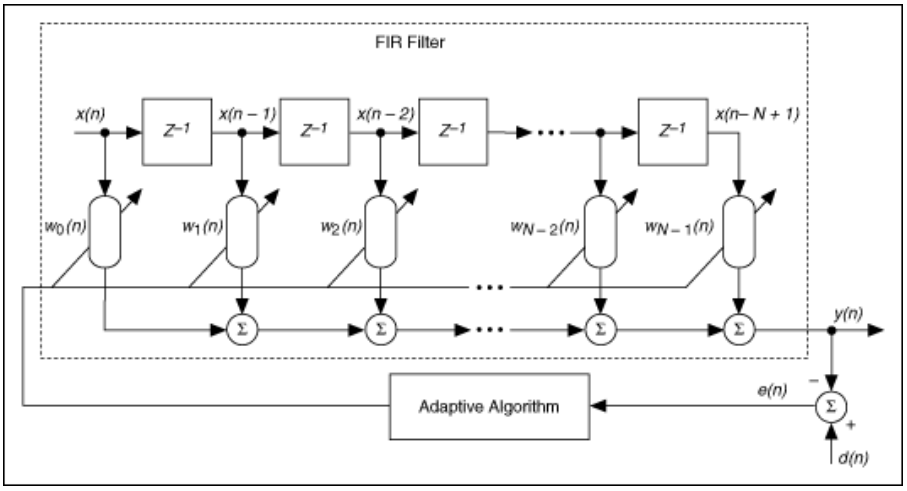



**Fig.2 - Linear Adaptive Filter**

Where,

x(n) = input signal, y(n) = corresponding output signal

d(n) = actual signal, e(n) = d(n) - y(n), intermediate error signal

$z^{-1}$ = unit delay, $w_i(n)$ = initial weight vector or filter coefficient i=[0, 1,2,…… N-1]

The aim of Adaptive Algorithm is to adjust the value of $w_i(n)$ in such a manner that the power of the error signal e(n) is minimized.

### 1.3 Adaptive Filter Algorithms

There is no unique solution of adaptive filters therefore based on parameter to be analyzed for performance, there are various types of adaptive algorithms, some of them are as follows-

### 1.3.1 LMS Algorithm

The Least Mean Square adaptive algorithm is given by Widrow and Hoff(1960). The simplicity of the LMS algorithm made it as a standard algorithm against other linear adaptive filter algorithms. In conventional LMS, the filter coefficients for each sample is updated as :-

**(Adjusted weight vector) = (step size parameter)*(input vector)*(error signal)**

**w^(n+1) = w^(n) + mu*u(n)*e(n)---(1)**

Where,

w^(n+1) = updated weight vector, w^(n) = weight vector of previous sample

mu= step-size parameter, u(n) = input vector

e(n) = error signal,

### 1.3.2 Recursive Least Squares (RLS) Algorithms

This Algorithm updates the filter coefficients by using the following equation:

$$w(n+1) = w(n) + e(n).K(n) \text{---(2)}$$

Where,

w(n)= filter coefficients vector and K(n) is the gain vector.

K(n) is defined by the following equation:

$$K(n) = \frac{P(n).u(n)}{[\alpha + u^T P(n).u(n)]} \text{---(3)}$$

$\alpha = contant$

When compared to LMS algorithms, Recursive Least Squares (RLS) algorithms have a faster convergence speed but involve more complicated mathematical operations and require more computational resources than LMS algorithms. Therefore, In this paper, Fast Block Algorithm is used which is a type of LMS Algorithm and some improvisations has also done over it.

### 1.4. LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a graphical programming language that uses icons known as Virtual Instruments VIs instead of lines of text to create applications. In contrast to text-based programming languages that use instructions to determine the order of program execution, LabVIEW uses dataflow programming. In data flow programming, the flow of data through the nodes in the block diagram determines the execution order of the VIs and functions. VIs, or virtual instruments, are LabVIEW programs that imitate physical instruments. The contents block diagram window can be compared to flow chart in which

the flow of instruction of designed code can be observed, whereas in front panel, is intended for interaction between user and VI. This panel enables user to give input values and observe the output data.

Since LabVIEW is graphical based programming instead of typical text-based programming, this makes it easier to design, observe, and analyze the simulation at each and every point of the code simultaneously.

## 2. PROPOSED ALGORITHM

In conventional LMS algorithm the weight vector of a finite impulse response are updated in time domain. By using Fourier transform in the adaptive algorithm the weight vector can be map in frequency domain and by taking it's inverse Fourier transform the weight vector can be map back in time domain. This process is known as frequency-domain adaptive filtering (FDAF).The Fast Block Adaptive Algorithm is an algorithm in which filter parameters are adapted in frequency domain. In this algorithm Fast Fourier Transform (FFT) is used as it is a powerful tool for performing fast convolution and fast correlation. The flow chart of Fast Block LMS algorithm is shown in Fig 3.

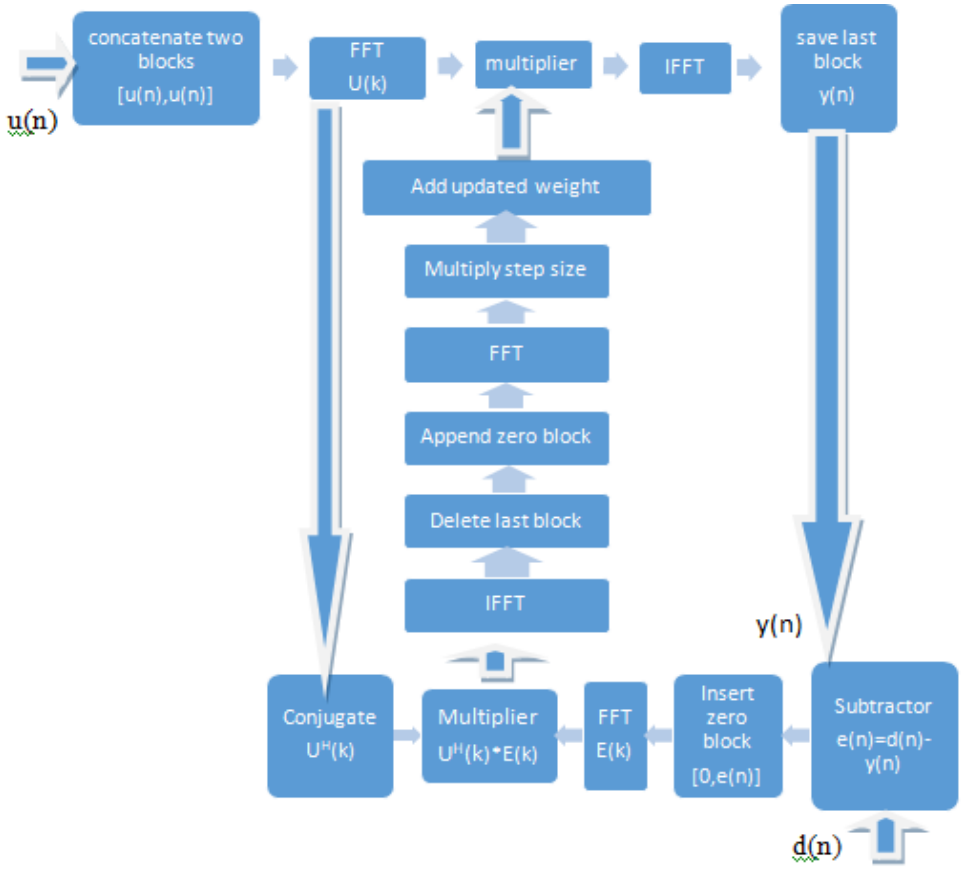



Fig. 3 Flow Chart of Fast Block LMS Adaptive Algorithm

Steps involved in Applied Fast Block Algorithm are as follows:-

**Step 1:** Take an input signal of N-samples represented as d(n) and select 100 elements of the input signal from i*100th index value( where, i= number of iterations the algorithm runs starting from zero)

**Step 2:** Add Additive White Gaussian Noise in the input signal

$$\mathbf{u(n) = d(n) + g(n)}\text{---}\mathbf{(4)}$$

where, d(n) = Desired signal, g(n) = Gaussian noise, u(n) = Noisy signal

**Step 3:** Concatenate two blocks [u(n),u(n)] and take it's FFT

$$\boldsymbol{U(k) = FFT[u(n), u(n)]}\text{---}\mathbf{(5)}$$

**Step 4:** Multiply U(k) with the updated weight vector, In Fast Block LMS Algorithm the initial weight is given as-

$$\boldsymbol{W(k) = FFT}\begin{bmatrix}\boldsymbol{w(k)}\\ \boldsymbol{0_{M\times 1}}\end{bmatrix}\text{---}\mathbf{(6)}$$Where,

w(k) should be taken as zero if its value is not known.
M = number of elements of weight matrix.

$$\boldsymbol{y(k) = U(k) \times W\hat{}(k)}\text{---}\mathbf{(6)}$$

**Step 5:** Take IFFT (Inverse Fast Fourier Transform) of y(k) and save last M-elements. The output will be represented as y(n).Subtract this output y(n) from the desired signal d(n) to get the error signal e(n)

$$\boldsymbol{e(n) = d(n) - y(n)}\text{---}\mathbf{(7)}$$

$$\boldsymbol{E(k) = FFT\,[0, e(n)]}\text{---}\mathbf{(8)}$$

**Step 6:** $\emptyset(k)$ is first M element of IFFT[U*(k) + E(k)]

**Step 7:**The updated weight vector is given by :-

$$\boldsymbol{W(k+1) = W(k) + \mu \times FFT}\begin{bmatrix}\boldsymbol{w(k)}\\ \boldsymbol{0_{M\times 1}}\end{bmatrix}\text{---}\mathbf{(9)}$$

Where,

$\boldsymbol{\mu}$= Step size, which was fixed in Standard Fast Block LMS Algorithm. But in our proposed Algorithm, it is given as-

$$\boldsymbol{\mu = \{mu/(Power\ of\ input\ signal)\}}\text{----}\mathbf{(10)}$$

mu = constant

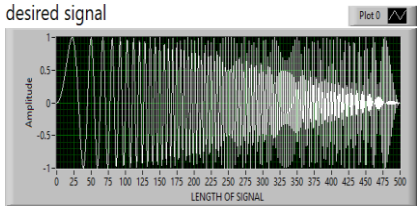
**Step 8:** This process runs for M-number of iterations and every time the value of y(n) is appended after it's previous value. The final value of y(n) is taken for the considerations to find the Mean Square Error between d(n) and y(n)

Although, the size of the weight vector always remain same for any value of N. The input signal is divided into subgroups having M number of elements and then processed. Then the output of each M-element are added in the end of the process. So that there will be no requirement of varying the size of weight vector which was not done in Standard Fast Block LMS Algorithm.

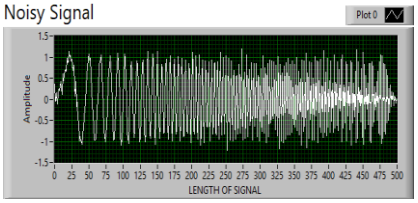
## 3. SIMULATION AND RESULTS

Simulation is done in graphical programming language LabVIEW.The Chirp Signal is an in-built signal in LabVIEW. The Audio signals are acquired in LabVIEW through headphones by using an in-built VI (Virtual Instrument) Acquire Sound. Two types of sound are acquired in our work. One of human voice and other is sound created by household equipments. The actual Chirp and Audio Signals are represented as desired signal. When these signal are added with Guassian Noise then it is represented as Noisy Signal which are further processed through both the Algorithms.

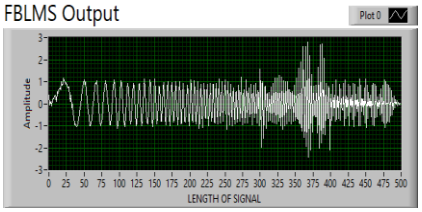
### 3.1 Chirp Signal :-



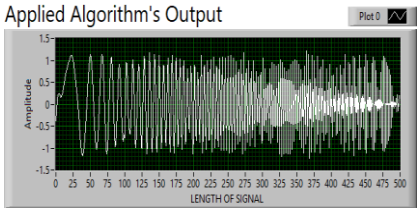

**(a1)**




**(b1)**

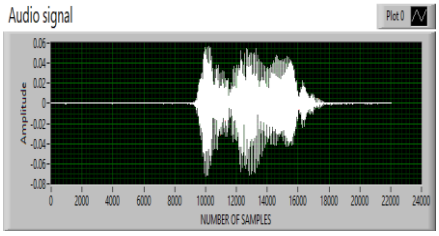



**(c1)**

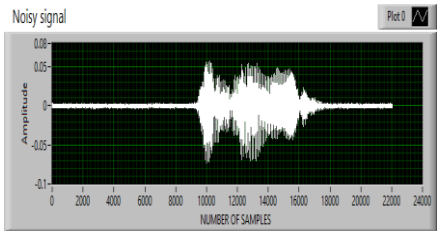



**(d1)**

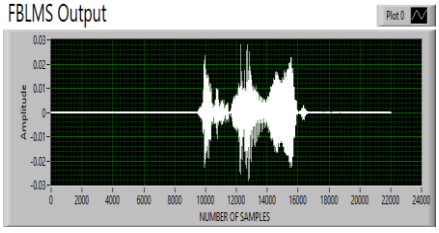
### 3.2 AUDIO SIGNAL(Human voice "Hello")



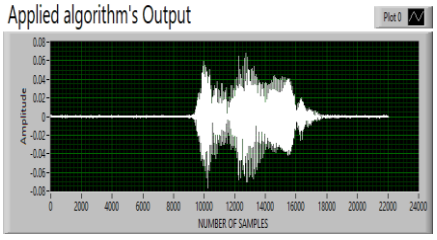

**(a2)**




**(b2)**

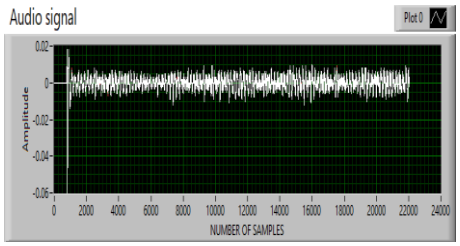



**(c2 )**

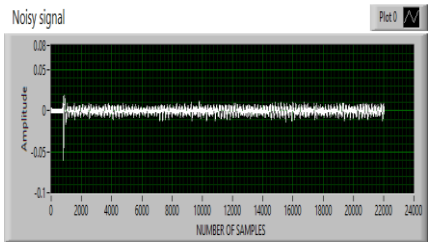



**(d2)**

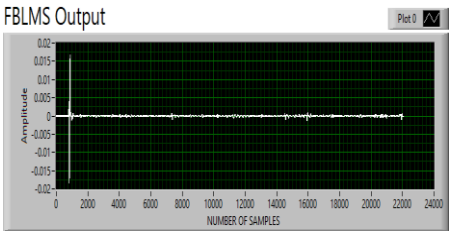
### 3.3 Audio Signal (Electronic or Electrical equipment's Sound)



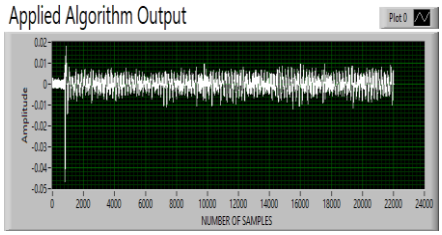

**(a3)**




**(b3)**




**(c3)**



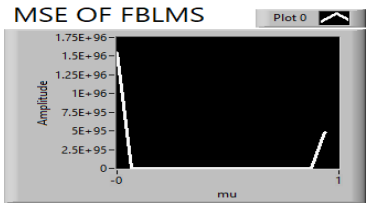

**(d3)**

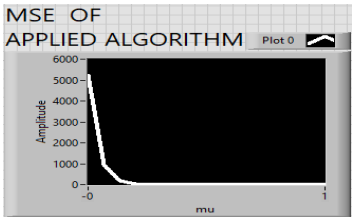
**Fig. 6 (a1),(a2) and (a3) represents actual signal,**
**(b1),(b2) and (b3) represents signal affected by Guassian noise**
**(c1),(c2) and (c3) represents output of Standard Fast Block Algorithm**
**(d1),(d2) and (d3) represents output of Proposed Fast Block Algorithm**

It can be seen that the Graph obtained from Proposed/Applied Fast Block Algorithm are better than Standard Fast Block Algorithm.

The MSE Graph for both Fast Block LMS and Applied Fast Block LMS Algorithms are shown for various input signals :-

- **CHIRP SIGNAL**

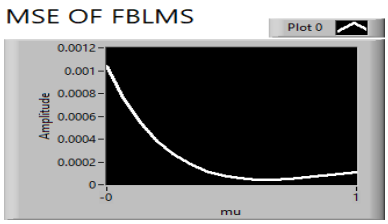


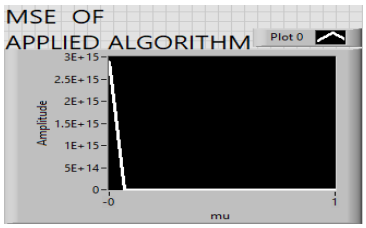




**MSE(min) = 6.32356E-3(at mu = 0.1)** **MSE(min) = 5.8975E-3(at mu = 2.7)**

Fig. 5 MSE curves of both Algorithm using Chirp signal as input

- **AUDIO SIGNAL(Human voice "Hello")**

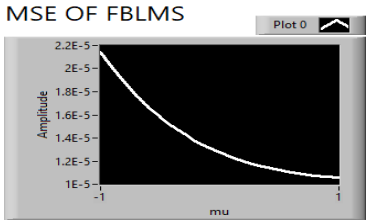


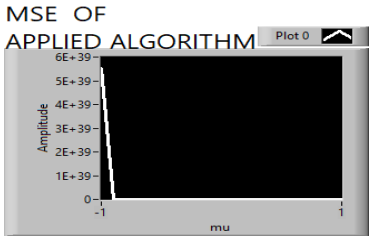




**MSE(min) = 4.56661E-5(at mu = 0.5)** **MSE(min) = 2.57526E-5(at mu = 3.8)**

Fig. 6 MSE curves of both Algorithm using Audio signal(human voice of "hello") as input

- **AUDIO SIGNAL(Electronic or Electrical equipment'sSound)**




**MSE(min approx.) = 8.13857E-7(at mu = 158)** **MSE(min) = 6.52097E-7(at mu = 0.8)**

Fig. 7 MSE curves of both Algorithm using Audio signal(Electronic or Electrical equipment's Sound) as input

## 4. Conclusion & Future Work

In Standard Fast Block LMS Algorithm, $\mu$ is taken as constant. But in this paper $\mu$ is a variable which is dependent on characteristics of input signal, this gives a certain control over performance of the Algorithm. This can also be observed through the graph, the value of MSE is decreased using the proposed Algorithm. Also the size of the weight vector is made constant using buffers of fixed size, irrespective of the input data size. This feature of implemented algorithm make it possible to analyze the signal obtained in real time, which was not possible with Standard Fast Block Algorithm, thus the complexity decreases. The performance of this work can

be further improved by designing and adopting a variable size weight vector depending on characteristic of received noisy signal.

## 5. REFERENCES

[1] Shrikant J. Honade, Prashant V. Ingole, Sanjay V. Dudul and Nitin A. Shelke,"Performance Analysis of Various LMS Adaptive Filtering Algorithms",IEEE WiSPNET 2017 conference.

[2] BERNARD WIDROW, JOHN M. McCOOL, MICHAEL G. LARIMORE AND C. RICHARD JOHNSON," Stationary and Non-stationary Learning Characteristics of the LMS Adaptive Filter", Proceedings of the IEEE, VOL. 64, NO. 8, AUGUST 1976

[3] Boio KrstajiC, Zdravko UskokoviC and LJubiSa Stankovic,"Adaptive Channel Equalizer with New VSS LMS Algorithm", Seventh International Symposium on Signal Processing and Its Applications, 2003,IEEE.

[4] AMAN SHARMA AND SANJEEV NARAYAN SHARMA," Adaptive Filtering Of LFM Signals Using FrFT", ICSP Paper ID: ICSP-AUL-1410

[5] SIDDAPPAJI and K L SUDHA," Performance Analysis of New Time Varying LMS (NTVLMS) Adaptive Filtering Algorithm in Noise Cancellation System", 2015 International Conference on Communication, Information & Computing Technology (ICCICT), Jan. 16-17, Mumbai, India