Acoustic Echo Cancellation using Normalised Least Mean Square(NLMS) Adaptive Algorithm

Sindhura Chilakapati, Jyothirmai Mamidala, Venkata Manoj Manda
Department of Signal Processing
Blekinge Institute of Technology
Karlskrona, Sweden
{sich15, srma15, vema15}@student.bth.se

Dr. Benny Sällberg
Sällberg Technologies e.U.,
Friedrich Schiller-Str. 11,
A-4840 Vöcklabruck, Austria
benny@sallberg.at

***Abstract* — Acoustic echo occurs when an audio source and sink operate in full duplex mode, an example of this is a hands-free loudspeaker telephone. The received signal from loud speaker is reverberated through the physical environment and picked up by the microphone. This causes delayed and attenuated images of the original speech signal, which is called as acoustic echo. The presence of a large acoustic coupling between microphone and loudspeaker causes this effect. Our aim here, is to provide a good free voice quality when two or more people communicate from different places. This report examines a technique of echo canceller system designed by making use of NLMS algorithm of adaptive filtering, employing discrete signal processing in MATLAB. Also a real-time implementation of an adaptive echo cancellation system has been developed using the analog device 21262 SHARC processor development kit.**

Keywords: reverberation, adaptive-filtering, analog device SHARC processor.

I. Introduction

ECHO is the phenomenon where distorted and delayed version of an original sound or electrical signal is reflected back to the original source. There are two types of echo; Acoustic echo, which is caused by the reflection of sound waves and acoustics coupling between the loudspeaker and the microphone and Electrical echo, caused by the impedance mismatch at the hybrids transformer which the subscriber two-wire lines are connected to telephone exchange four wire lines in the telecommunication systems.

In the case of teleconference system, the speech signal from far-end which is generated from loud speaker. This after directing and reflecting from the floor, wall, other objects inside the room is receipt by microphone of near-end, as the result, this makes the echo that is sent back to the far-end. This called as acoustic echo problem will disturb the conversation of the people and reduce the quality of system. This is a common problem of the Communication networks.

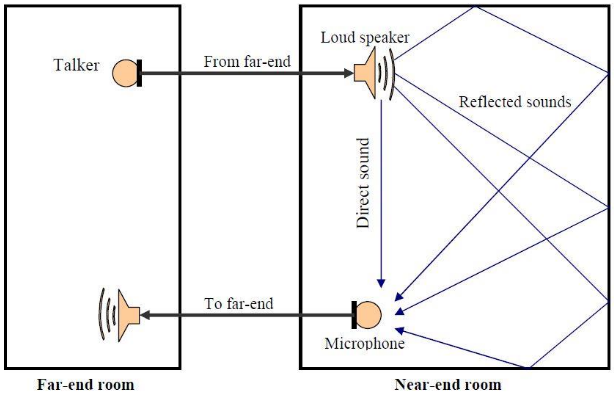

Fig. 1. Figure 1: Teleconference system with echo paths

To handle with the acoustic echo problem in teleconference systems, one can use voice switches and directional microphones but these methods have placed physical restriction on the speaker. The common and more perfective method is implementing the Acoustic Echo Cancellation (AEC) to remove the echo. AEC enhances greatly the quality of the audio signal of the hands-free communication system. Due to their assistance, the conferences will work more smoothly and naturally, keep the participants more comfortable. In the modern digital communication system such as: Public Switched Telephone Network (PSTN), Voice over IP (VoIP), Voice over Packet (VoP) and cell phone networks; the application of Acoustic echo canceller is very important and necessary because it brings the better quality of service and obtains the main purpose of the communication service providers.

The main objective is to use NLMS algorithm as a technique in designing acoustic echo canceller system. Thereby, the cancellation of acoustic echo is done and this provides an echo free environment for speakers during conversation.

II. Background

In order to understand the content presented in this paper it is first necessary to provide some background information regarding digital signal theory. We will start out elementarily and then progress to more complicated theoretical basis in the derivation and implementation of the adaptive filtering technique used in acoustic echo cancellation.

A. Speech signals:

A speech signal consists of three classes of sounds. They are voiced, fricative and plosive sounds. Voiced sounds are caused by excitation of the vocal tract with quasi-periodic pulses of airflow. Fricative sounds are formed by constricting the vocal tract and passing air through it, causing turbulence that result in a noise-like sound. Plosive sounds are created by closing up the vocal tract, building up air behind it then suddenly releasing it, this is heard in the sound made by the letter .p.

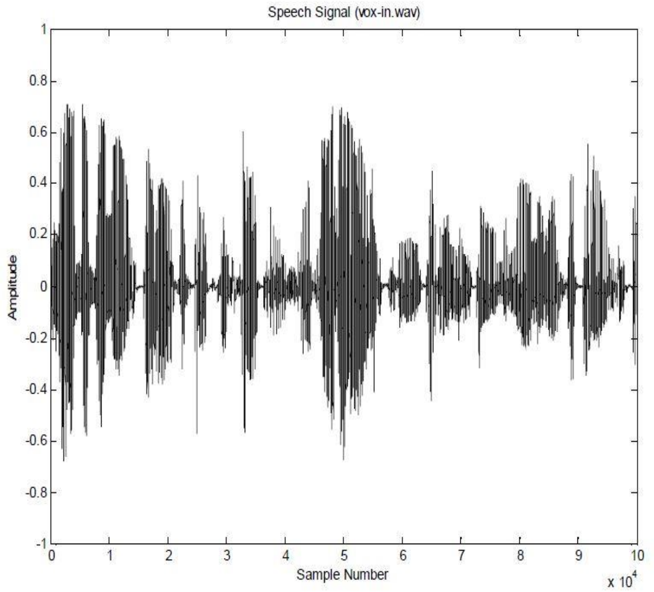

Fig. 2. Figure 2: Speech Signal

Figure shows a discrete time representation of a speech signal. By looking at it as a whole we can tell that it is non-stationary. That is, its mean values vary with time and cannot be predicted using the above mathematical models for random processes. However, a speech signal can be considered as a linear composite of the above three classes of sound, each of these sounds are stationary and remain fairly constant over intervals of the order of 30 to 40 milli-seconds. The theory behind the derivations of many adaptive filtering algorithms usually requires the input signal to be stationary.

B. AEC System Overview:

Acoustic echo cancellation is required in different fields of communication for removing the echo of the coupling between the loudspeaker and the microphone. In case of not doing this, then this coupling results in an undesired acoustic echo which degrades the quality of sound. We describe a block diagram of an AEC system as in Figure. This system consists of following three components: 1. Adaptive filter. 2. Doubletalk detector. 3. Nonlinear processor.

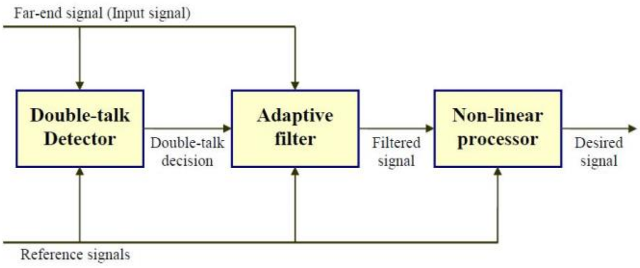

Fig. 3. Figure 3: Acoustic echo canceller system

Adaptive filter is the most important component of acoustic echo canceller and it plays a key role in acoustic echo cancellation. It performs the work of estimating the echo path of the room for getting a replica of echo signal. It requires an adaptive update to adapt to the environmental change. Another important thing is the convergence rate of the adaptive filter which measures that how fast the filter converges for best estimation of the room acoustic path.

It is rather difficult to predict when the adaptation of the filter should stop or slow down and it is also important to know that the near-end speech signal exists or not in the presence of far-end signal. In the situation when both ends talk (near-end and far-end), this is known as double-talk. In case of double-talk, the error signal will contain both echo estimation error and near-end speech signal. When we use this signal for updating the filter coefficient then it diverges. As the result, the adaptive filter will work incorrectly and finally the bad sound signal was issued. So to overcome this problem, one uses Double-talk Detector.

The nonlinear processor (NLP) is required for completely or partly cancels the residual signal in the absence of near-end speech signal. By removing the residual signal will cancel any occurring acoustic echo. The NLP will gradually cancel the signal and insert a form of comfort noise to give the impression to far-end. The NLP as well as the adaptive filter need an accurate estimation from the DTD to operate efficiently.

C. Adaptive filters:

The block diagram for the adaptive filter method utilised is shown in below figure. Here w represents the coefficients of the FIR filter tap weight vector, x(n) is the input vector samples, z-1 is a delay of one sample periods, y(n) is the adaptive filter output, d(n) is the desired echoed signal and e(n) is the estimation error at time n. The aim of an adaptive filter is to calculate the difference between the desired signal and the adaptive filter output, e(n). This error signal is fed back into the adaptive filter and its coefficients are changed algorithmically in order to minimise a function of this difference, known as the cost function.

In the case of acoustic echo cancellation, the optimal output of the adaptive filter is equal in value to the unwanted echoed signal. When the adaptive filter output is equal to desired signal the error signal goes to zero. In this situation the echoed signal would be completely cancelled and the far user would not hear any of their original speech returned to them.

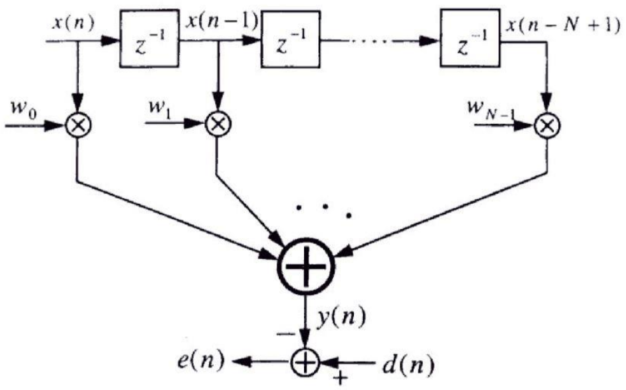

Fig. 4. Figure 4: Adaptive filter block diagram

The methods can be divided into two groups based on their cost functions. The first class are known as Mean Square Error (MSE) adaptive filters, they aim to minimize a cost function equal to the expectation of the square of the difference between the desired signal d(n), and the actual output of the adaptive filter y(n). When the adaptive filter output is equal to desired signal the error signal goes to zero. In this situation the echoed signal would be completely cancelled and the far user would not hear any of their original speech returned to them.

$$\xi(n) = E[e^2(n)] = E[(d(n) - y(n))^2] \qquad (1.)$$

The second class are known as Recursive Least Squares (RLS) adaptive filters and they aim to minimise a cost function equal to the weighted sum of the squares of the difference between the desired and the actual output of the adaptive filter for different time instances. The cost function is recursive in the sense that unlike the MSE cost function, weighted previous values of the estimation error are also considered. The cost function is shown below in equation the parameter λ is in the range of $0<\lambda<1$. It is known as the forgetting factor as for $\lambda<1$ it causes the previous values to have an increasingly negligible effect on updating of the filter tap weights. The value of 1/(1- λ) is a measure of the memory of the algorithm. The cost function for RLS algorithm, ζ(n), is stated in equation.

$$\zeta(n) = \sum_{k=1}^{n} \rho_n(k) e_n^2(k) \qquad (2.)$$

$$\rho_n(k) = \lambda^{n-k} \qquad (3.)$$

Where k=1, 2, 3.n., k=1 corresponds to the time at which the RLS algorithm commences. Only the previous N (corresponding to the filter order) error signals are considered.

As stated previously, considering that the number of processes in our ensemble averages is equal to one, the expectation of an input or output value is equal to that actual value at a unique time instance.

D. Adaptive algorithm-NLMS:

One of the primary disadvantages of the LMS algorithm is having a fixed step size parameter for every iteration. This requires an understanding of the statistics of the input signal prior to commencing the adaptive filtering operation. In practice this is rarely achievable. Even if we assume the only signal to be input to the adaptive echo cancellation system is speech, there are still many factors such as signal input power and amplitude which will affect its performance. As the input signal power changes in time and due to this change the step-size between two adjacent coefficients of the filter will also change and also affects the convergence rate. Due to small signals this convergence rate will slow down and due to loud signals this convergence rate will increase and give an error. So to overcome this problem, we try to adjust the step-size parameter with respect to the input signal power. Therefore the step-size parameter is said to be normalized. by selecting a different step size value, μ(n), for each iteration of the algorithm. This step size is proportional to the inverse of the total expected energy of the instantaneous values of the coefficients of the input vector **x**(n). This sum of the expected energies of the input samples is also equivalent to the dot product of the input vector with itself, and the trace of input vectors auto-correlation matrix,

The step size for computing the update weight vector is,

$$\mu(n) = \frac{\beta}{c + ||X(n)||^2} \qquad (4.)$$

Where, μ (*n*) is step-size parameter at sample n, β is normalized step-size (0 < β < 2), *c* is safety factor (small positive constant).

When design the LMS adaptive filter, one difficulty we meet is the selection of the step-size parameter μ. For stationary processes, this algorithm converts in the limits:

$$0 < \mu < \frac{2}{\lambda_{max}} \qquad (5.)$$

$$0 < \mu < \frac{2}{trace(\boldsymbol{R}_{x})} \tag{6.}$$

However, the auto-correlation ($\boldsymbol{R}_{x}$) generally is unknown, for this reason, the maximum lambda max λ and ($\boldsymbol{R}_{x}$) are estimated in order to use the bounds. To solve this problem, one introduces new estimate of trace($\boldsymbol{R}_{x}$)

$$trace(\boldsymbol{R}_{x}) = (p+1)\hat{E}\{|x(n)|^2\} \tag{7.}$$

Where,
p = 0,1,2,... and
$\hat{E}\{|x(n)|^2\}$ is the power of input signal. It can be estimated by estimator:

$$\hat{E}\{|x(n)|^2\} = \frac{1}{p+1}\sum_{k=0}^{p} |x(n-k)|^2 \tag{8.}$$

Therefore, the limits of step-size parameter will become,

$$0 < \mu < \frac{2}{(p+1)E\{[x(n)]^2\}} \tag{9.}$$

By substituting one-size parameter:

$$0 < \mu < \frac{2}{X^H(n)X(n)} \tag{10.}$$

For time-varying processes, one computes the step-size parameter in time (sample n),

$$\mu\,(\,n\,) = \frac{\beta}{X^H(n)X(n)} = \frac{\beta}{||X(n)||^2} \tag{11.}$$

Where, β is normalized step-size (0 < β < 2)
By replaced μ by μ (n) into the Equation for updating the weight vector in LMS algorithm, we achieve a new algorithm was known as Normalized Least Mean Square (NLMS). The weight vector update now is,

$$W\,(n+1) = w\,(n) + \mu(n)e(n)X^*(n) \tag{12.}$$

(or)

$$W\,(n+1) = w\,(n) + \frac{\beta}{||X(n)||^2}e(n)X^*(n) \tag{13.}$$

In the LMS algorithm, because the weight vector **w**(n) changes depending on the input signal **x**(n) . Thus it will get the problem is called as *gradient noise amplification.* when **x**(n) is too large. However, by using NLMS algorithm we can avoid this problem. Take a look the Equation when **x**(n) is very small the calculation of weight vector updating equation will be the big problem. For this reason, one implements the safety factor as,

$$W\,(n+1) = w\,(n) + \frac{\beta}{c+||X(n)||^2}e(n)X^*(n) \tag{13.}$$

Where c is safety factor (small positive constant)
Finally, the Equation (2.54) is the weight vector updating equation for NLMS.

E. Echo Return Loss Enhancement (ERLE):

Echo Return Loss Enhancement ERLE is one of the most important parameters is commonly used to evaluate the performance of the echo cancellation algorithm. This quantity measures how much echo attenuation the echo canceller removed from the microphone signal. ERLE, measures in dB, is defined as the ratio of the microphone signal's power (*d[n]*) and the residual error signal's power (*e[n]*).

$$ERE = 10 * \log_{10}\big[[mean(power(echo)))]\big] / \big[[mean(power(error)))]\big] \tag{14.}$$

ERLE depends on the algorithm we use for the adaptive filter, two quantities are considered with ERLE are the convergence time and near-end attenuation will be different relative to different algorithms. In the simulation, to fulfil the MATLAB code to the computation of ERLE is done as follow, to obtain an estimate of convergence or the Echo Return Loss Enhancement (ERLE), first the coupling factor or the Echo Return Loss (ERL) of the loudspeaker-microphone enclosure is estimated. An estimate of the ERL is required to determine how much attenuation can be attributed to the echo path and how much can be attributed to the echo canceller. The coupling factor determines the attenuation or possible gain in the path. There are two main approaches to estimating the coupling factor of an echo canceller. The first method is amplitude based while the second is cross-spectrum based. The amplitude based method to estimate ERL is the average spectral energy of the near-end signal over the average spectral energy of the far-end signal. This approach should only be updated during periods of known far-end signal energy and should not be updated during periods of double-talk. In the cross-spectrum based method, the far-end and near-end spectrum signals

Then it is normalized by the far-end signal energy. This method is unaffected by double-talk of the near-end speaker and far-end speaker as long as they are uncorrelated. The downside to this method is the echo path changes are not followed accurately due to the long averaging period. Using a combination of the two methods will allow for quick and accurate estimation of the ERL, and hence provides proper control of the entire echo cancellation system.

III. PROBLEM STATEMENT AND MAIN CONTRIBUTION

In reverberant environments the speech signal is mixed with the acoustic echoes. To avoid these disturbances Acoustic Echo Cancellation technique is used which removes the echoes caused due to reverberation.

The occurrence of acoustic echo in speech transmission causes signal interference and also reduces the quality of communication.

Main contribution is application of NLMS technique for echo canceller system, implementing in MATLAB software, visual DSP code verification by using DSP kit.

IV. PROBLEM SOLUTION

A. *Modelling:*

Adaptive filter is a good supplement to achieve a good replica because of the echo path is usually unknown and time-varying.

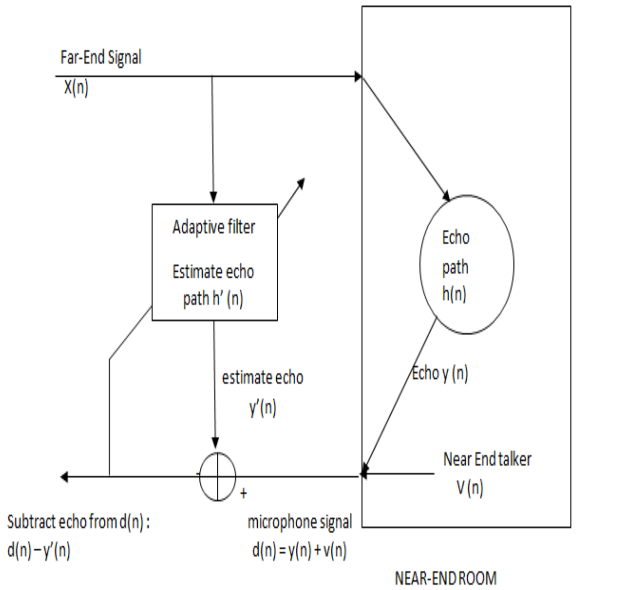

Fig. 5. Figure 5: Modelling of acoustic echo canceller

The figure illustrates about three step of the AEC using adaptive filter. In the Figure, by using adaptive filter for AEC follows three basic steps: Estimate the characteristics of echo path $h(n)$ represented by $h\hat{}(n)$; Create a replica of the echo signal represented by $y\hat{}(n)$; Echo is then subtracted from microphone signal (includes near-end and echo signals) to obtain the desired signal or clear signal $d(n) - y\hat{}(n)$.

B. *Implementation:*

The Acoustic Echo Canceller is implemented in both MATLAB and on Digital Signal Processor. The output signal free from Echo, is obtained using NLMS algorithm. The Echo Return Loss Enhancement (ERLE) is also calculated.

i. Implementation on MATLAB:

MATLAB is a numerical computing environment that is especially effective to calculate and simulate the technical problems. This programming language is very powerful allows to record audio signals, matrix calculation and to hear the resultant sounds. The structure of the commands is suitable to compute with signal processing.

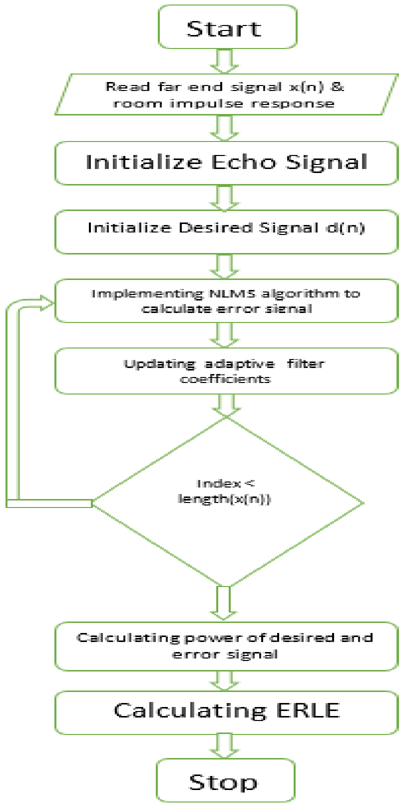

Fig. 6. Figure 6: flow chart of implementation in matlab

Figure shows the flowchart for the implementation of acoustic echo canceller in MATLAB. The input signal x(n) is of 160000 samples with sampling frequency (Fs) 8 KHz. The reference signal i.e. room impulse response h(n) is of 4096 samples.

ii. Implementation on DSP:

DSP processors are concerned primarily with real time signal processing which means that the processing must be in pace with some external event, in this case it is continuous audio input. DSP is especially designed for audio and image signal processing which makes it efficient in handling applications like generating instrument effects. In this project. The Acoustic Echo Canceller is implemented on ADSP 21262 SHARC processor. The ADSP-21262 EZ-KIT Lite includes an ADSP-21262 processor desktop evaluation board along with an evaluation suite of the VisualDSP++ development. It also includes a debugging environment with the C/C++ compiler, assembler, and linker.

The SHARC processors are based on a 32-bit super Harvard architecture that includes a unique memory architecture comprised of two large on-chip, dual-ported SRAM blocks coupled with a sophisticated IO processor, which gives SHARC the bandwidth for sustained high-speed computations. SHARC represents today's de facto standard for floating- point DSP targeted for premium audio applications.

The continuous audio input is given using the function "DSP_get_audio". It is used to return a pointer to the current audio block. Then the samples from audio block are copied to a buffer. The output is copied to the same physical block buffer.

As the input is real time audio, programming the DSP is a bit difficult task. While programming, a standard configuration provided by the software framework is used. The software framework consists of one header file i.e. framework.h and also .c class file i.e. framework.c. This software framework defines the DSP sample rate, DSP block size, DSP input gain and DSP output attenuation.

The implementation of Acoustic Echo Canceller with N=128 samples (input signal) and L = 64 samples (room impulse response) on DSP required the following amount of memory resources:

- Memory (in bytes) : 116 bytes.
- Instructions per cycle : 929.28 million

iii. Problems faced with MATLAB:

- Longer execution time while working with real time applications. This MATLAB constraint restricts the length of audio signal.
- Block processing had to be used to synchronize the MATLAB code with the actual DSP application. Implementation of block processing posed a challenge in the case of time-varied delay effects.
- Longer execution time was another problem which is undesirable while working with real time applications. This MATLAB constraint restricted the length of audio signal to be considered while observing the results.

iv. Problems faced with DSP:

- Printing the data in real time is difficult as the processing rate is very fast compared to the rate at which values are printed. This eliminates part of data from printing as the data is over-written.
- Reducing the number of iterations was a challenge since it may increase the processing time of each block.
- Block processing was also a challenge while programming the DSP.

C. Validation:

MATLAB is used for the offline implementation and Visual DSP++ Environment is used for the real time implementation. There are two modes in Visual DSP++ Environment:

- Simulation.
- Emulation.

Under simulation mode, the input is given from the PC (MATLAB or direct audio file), and the output obtained is compared with the output of MATLAB.

Under emulator mode, the real time implementation is done. The outputs obtained in both the modes are similar. The outputs obtained from MATLAB are shown below:

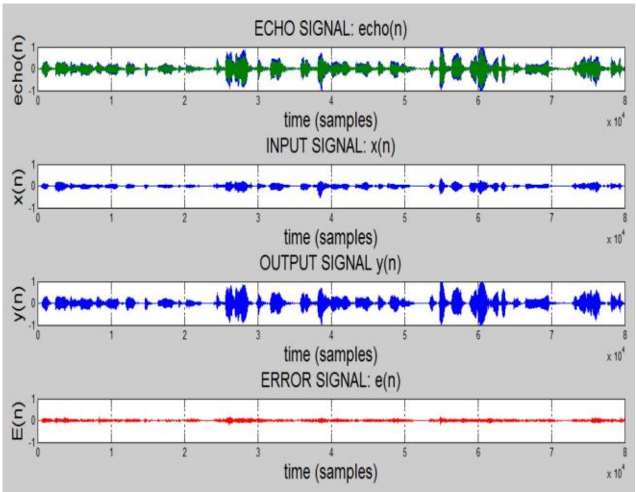

Fig. 7. Figure 7: plot of different signals

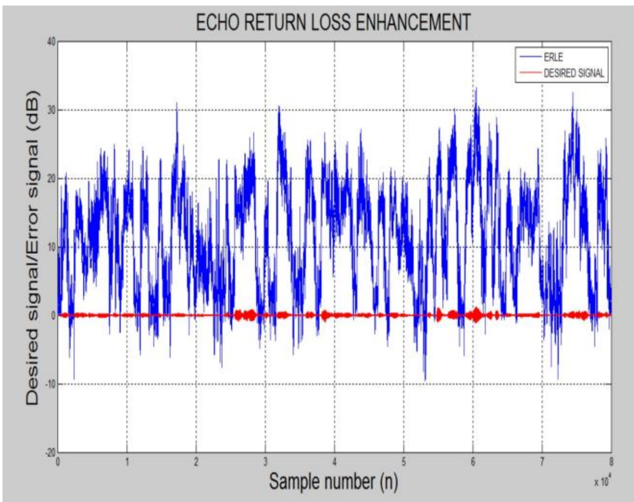

Fig. 8. Figure 8: plot of ERLE

The maximum value of ERLE can be varied depending on the value of step size.

TABLE 1: VARIATION OF MAX VALUE OF ERLE W.R.T STEP-SIZE

a	c	μ	Max ERLE
0.34	0.1	3.3840	32.4788
0.20	0.1	1.9906	30.8631
0.10	0.2	0.4988	28.4057

The maximum value of ERLE is decreased when the step size is decreased.

The results from Visual DSP++ are compared with the results from MATLAB i.e. the samples of ERLE and output signal y(n) from Visual DSP++ are compared with the samples obtained from MATLAB. They are observed to be approximately equal.

V. CONCLUSION

- The echo from the signal is removed using NLMS, when the algorithm is triggered.
- The results obtained from MATLAB are approximately equal to the results obtained from the C-code in simulator mode.
- The ERLE plot obtained is also efficient since the range of ERLE is within -40dB to 40dB.
- The audio of the output speech signals were highly satisfactory and validated the goals of this research.

VI. FUTURE WORKS

- The Acoustic Echo cancelation can be further implemented using other adaptive algorithms like Recursive Least Square (RLS) algorithm and Variable Step Size Normalized LMS (VSNLMS).
- The Acoustic Echo cancelation can also be further implemented on a real time system with multiple channel echoes.

REFERENCES

[1] M. Rages and K. C. Ho, “Limits on echo return loss enhancement on a voice coded speech signal,” in *The 2002 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS-2002*, 2002, vol. 2, pp. II–152–II–155 vol.2.

[2] M. A. Iqbal and S. L. Grant, “Novel variable step size nlms algorithms for echo cancellation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008*, 2008, pp. 241–244.

[3] M. Hamidia and A. Amrouche, “Improved variable step-size NLMS adaptive filtering algorithm for acoustic echo cancellation,” *Digit. Signal Process.*, vol. 49, pp. 44–55, Feb. 2016.

[4] V. Garre and S. K. Mannem, *An Acoustic Echo Cancellation System based on Adaptive Algorithm*. 2012.

[5] C. Schüldt, *Low-complexity adaptive filtering for acoustic echo cancellation in audio conferencing systems*. Karlskrona: Department of Signal Processing, School of Engineering, Blekinge Institute of Technology, 2009.

[6] G. G. Chowdary, *Acoustic Echo cancellation inside a Conference Room using Adaptive Algorithms*. 2012.