Echo Cancellation in Audio Signal using LMS Algorithm

Sanjay K. Nagendra
Dept of Electronics & Communication
Sri Bhagawan Mahaveer Jain College of Engineering,
Bangalore (Rural) District, Karnataka (India) - 562 112
Email: sanjayankenkare@gmail.com

Vinay Kumar.S.B
Dept of Electronics & Communication
Sri Bhagawan Mahaveer Jain College of Engineering
Bangalore (Rural) District, Karnataka (India) - 562 112
Email: vinayvinaysb@gmail.com

Abstract—Acoustic echo cancellation is a common occurrence in today’s telecommunication systems. It occurs when an audio source and sink operate in full duplex mode; an example of this is a hands-free loudspeaker telephone. The signal interference caused by acoustic echo is distracting to both users and causes a reduction in the quality of the communication. This paper focuses on the use of adaptive filtering techniques to reduce this unwanted echo, thus increasing communication quality. Adaptive filters are a class of filters that iteratively alter their parameters in order to minimize a function of the difference between a desired target output and their output. In the case of acoustic echo in telecommunications, the optimal output is an echoed signal that accurately emulates the unwanted echo signal. This is then used to negate the echo in the return signal. This paper examines various techniques and algorithms of adaptive filtering, employing discrete signal processing in MATLAB.

Keywords: Echo cancellation, LMS, Audio Signal

I. INTRODUCTION

Acoustic echo occurs when an audio signal is reverberated in a real environment, resulting in the original intended signal plus attenuated, time delayed images of this signal. This project will focus on the occurrence of acoustic echo in telecommunication systems. Such a system consists of coupled acoustic input and output devices, both of which are active concurrently. An example of this is a hands-free telephony system. In this scenario the system has both an active loudspeaker and microphone input operating simultaneously. The system then acts as both a receiver and transmitter in full duplex mode. When a signal is received by the system, it is output through the loudspeaker into an acoustic environment. This signal is reverberated within the environment and returned to the system via the microphone input. These reverberated signals contain time delayed images of the original signal, which are then returned to the original sender (Figure 1, $a_k$ is the attenuation, $t_k$ is time delay). The occurrence of acoustic echo in speech transmission causes signal interference and reduced quality of communication.

The method used to cancel the echo signal is known as adaptive filtering. Adaptive filters are dynamic filters which iteratively alter their characteristics in order to achieve an optimal desired output. An adaptive filter algorithmically alters its parameters in order to minimize a function of the difference between the desired output $d(n)$ and its actual output $y(n)$. This function is known as the cost function of the adaptive algorithm. Figure 2 shows a block diagram of the adaptive echo cancellation system implemented throughout this paper. Here the filter $H(n)$ represents the impulse response of the acoustic environment, $W(n)$ represents the adaptive filter used to cancel the echo signal. The adaptive filter aims to equate its output $y(n)$ to the desired output $d(n)$ (the signal reverberated within the acoustic environment). At each iteration the error signal, $e(n)=d(n)-y(n)$, is fed back into the filter, where the filter characteristics are altered accordingly.
This project deals with acoustic echo as applies to audio signals, although the techniques will be applicable to a variety of other disciplines.

The goals of this project are as follows:
• To examine adaptive filtering LMS technique as they apply to acoustic echo cancellation and audio signals.
• To simulate LMS algorithm using Matlab.

II. ADAPTIVE FILTERS

A. Introduction

Figure 3 shows the block diagram for the adaptive filter method utilized in this project.

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 minimize 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 audio signals returned to them.

This section examines adaptive filters and various algorithms utilized. The various methods used in this paper 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 \( y(n) \) (equation 1).

\[
\xi(n) = E[e^2(n)] = E[(d(n) - y(n))^2] \tag{1}
\]

The second class are known as Recursive Least Squares (RLS) adaptive filters and they aim to minimize 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 2, the parameter \( \lambda \) 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 \( 1/(1-\lambda) \) is a measure of the memory of the algorithm, this paper will primarily deal with infinite memory, i.e. \( \lambda =1 \). The cost function for RLS algorithm, \( \zeta(n) \), is stated in equation.

\[
\zeta(n) = \sum_{k=1}^{N} \rho_n(k)e_n^2(k) \tag{2}
\]

Where \( k = 1, 2, 3.n. \), \( k = 1 \) corresponds to the time at which the RLS algorithm commences. Later we will see that in practice not all previous values are considered; rather 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. However, for the purposes of deriving these algorithms, the expectation notation shall still be used.

B. Wiener Filters

Wiener filters are a special class of transversal FIR filters which build upon the mean square error cost function of equation 3 to arrive at an optimal filter tap weight vector which reduces the MSE signal to a minimum. They will be used in the derivation of adaptive filtering algorithms in later sections.

Consider the output of the transversal FIR filter as given below, for a filter tap weight vector, \( w(n) \), and input vector, \( x(n) \).

\[
y(n) = \sum_{i=0}^{N-1} w(n)x(n-i) = w^T(n)x(n) \tag{3}
\]
The mean square error cost function can be expressed in terms of the cross-correlation vector between the desired and input signals, \( p(n) = E[x(n) d(n)] \), and the autocorrelation matrix of the input signal, \( R(n) = E[x(n)x'(n)] \).

\[
\xi(n) = E[(d(n) - y(n))^2] = E[(d(n) - x(n)w^n(n))^2] = E[d^2(n) - 2d(n)x(n) + x^2(n)]w^T(n)w(n) + \]
\[
E[w^T(n)x(n)x^T(n)x(n)] + E[w^T(n)x(n)x^T(n)]w(n) = E[d^2(n)] - 2E[w^T(n)x(n)] + E[w^T(n)x(n)]w(n) \tag{4}
\]

When applied to FIR filtering the above cost function is an \( N \)-dimensional quadratic function. The minimum value of \( \xi(n) \) can be found by calculating its gradient vector related to the filter tap weights and equating it to 0.

\[
\frac{\partial}{\partial w_0} = 0 \text{ for } 0, 1, \ldots, N - 1
\]

\[
V = \left[ \frac{\partial}{\partial w_0}, \frac{\partial}{\partial w_1}, \ldots, \frac{\partial}{\partial w_{N-1}} \right]^T
\]

\[
\nabla \xi = 0 \tag{5}
\]

By finding the gradient of equation 4, equating it to zero and rearranging gives us the optimal wiener solution for the filter tap weights, \( w_o \).

\[
\begin{align*}
2Rw_0 - 2p &= 0 \\
w_0 &= R^{-1}p
\end{align*} \tag{6}
\]

The optimal wiener solution is the set of filter tap weights which reduce the cost function to zero. This vector can be found as the product of the inverse of the input vector autocorrelation matrix and the cross correlation vector between the desired signal and the input vector. The Least Mean Square algorithm of adaptive filtering attempts to find the optimal wiener solution using estimations based on instantaneous values.

C. Mean Square Error (MSE) Adaptive Filters

C1. Least Mean Square (LMS) Algorithm

The Least Mean Square (LMS) algorithm was first developed by Widrow and Hoff in 1959 through their studies of pattern recognition (Haykin 1991, p. 67). From there it has become one of the most widely used algorithms in adaptive filtering. The LMS algorithm is a type of adaptive filter known as stochastic gradient-based algorithms as it utilizes the gradient vector of the filter tap weights to converge on the optimal wiener solution. It is well known and widely used due to its computational simplicity. It is this simplicity that has made it the benchmark against which all other adaptive filtering algorithms are judged.

With each iteration of the LMS algorithm, the filter tap weights of the adaptive filter are updated according to the following formula.

\[
w(n+1) = w(n) + 2\mu e(n)x(n) \tag{7}
\]

Here \( x(n) \) is the input vector of time delayed input values, \( x(n) = [x(n) x(n-1) x(n-2) \ldots x(n-N+1)]^T \).

The vector \( w(n) = [w0(n) w1(n) w2(n) \ldots w_{N-1}(n)]^T \) represents the coefficients of the adaptive FIR filter tap weight vector at time \( n \). The parameter \( \mu \) is known as the step size parameter and is a small positive constant. This step size parameter controls the influence of the updating factor. Selection of a suitable value for \( \mu \) is imperative to the performance of the LMS algorithm, if the value is too small the time the 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.

C2. Derivation of the LMS algorithm

The derivation of the LMS algorithm builds upon the theory of the wiener solution for the optimal filter tap weights, \( w_o \), as outlined in section B. It also depends on the steepest-descent algorithm as stated in equation 8, this is a formula which updates the filter coefficients using the current tap weight vector and the current gradient of the cost function with respect to the filter tap weight coefficient vector, \( \nabla \xi(n) \).

\[
w(n+1) = w(n) - \mu \nabla \xi(n) \tag{8}
\]

Where \( \xi(n) = E[e^2(n)] \)

As the negative gradient vector points in the direction of steepest descent for the \( N \)-dimensional quadratic cost function, each recursion shifts the value of the filter coefficients closer toward their optimum value, which corresponds to the minimum achievable value of the cost function, \( \xi(n) \). The LMS algorithm is a random process implementation of the steepest descent algorithm, from equation 8. Here the expectation for the error signal is not known so the instantaneous value is used as an estimate. The steepest descent algorithm then becomes equation 9.

\[
w(n+1) = w(n) - \mu \xi(n) \tag{9}
\]

where \( \xi(n) = e^2(n) \)
The gradient of the cost function, $\nabla \xi(n)$, can alternatively be expressed in the following form.

$$\nabla \xi(n) = \nabla \hat{y}^2(n)$$

$$= \frac{\partial \hat{y}^2(n)}{\partial w}$$

$$= 2\hat{y}(n) \frac{\partial \hat{y}(n)}{\partial w}$$

$$= 2\hat{y}(n) \frac{\partial (d(n) - y(n))}{\partial w}$$

$$= -2\hat{y}(n)x(n)$$

Substituting this into the steepest descent algorithm of equation 8, we arrive at the recursion for the LMS adaptive algorithm.

$$w(n+1) = w(n) + 2\mu \hat{y}(n)x(n)$$

C3. Implementation of the LMS Algorithm

Each iteration of the LMS algorithm requires 3 distinct steps in this order:

1. The output of the FIR filter, $y(n)$ is calculated using equation 12.

$$y(n) = \sum_{i=0}^{N-1} \hat{y}(n)x(n-i) = w^T(n)x(n)$$

2. The value of the error estimation is calculated using equation 13.

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

3. The tap weights of the FIR vector are updated in preparation for the next iteration, by equation 14.

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

The main reason for the LMS algorithms popularity in adaptive filtering is its computational simplicity, making it easier to implement than all other commonly used adaptive algorithms. For each iteration the LMS algorithm requires $2N$ additions and $2N+1$ multiplications ($N$ for calculating the output, $y(n)$, one for $2\mu e(n)$ and an additional $N$ for the scalar by vector multiplication).

III. MATLAB SIMULATIONS OF ADAPTIVE FILTERING ALGORITHMS

LMS adaptive filtering algorithm outlined in Section 2 is implemented using Matlab. In the simulation the echoed signal was generated by defining an appropriate impulse response then convolving this with an audio input wav file. and the results are shown below.
In the above figure 4. Shows the magnitude response of a Chebyshev Type2 filter, Figure 5. shows the Room impulse response, Figure 6 shows the input audio signal, then the echo signal is noting but the convolution between the sequence of impulse response and audio input is as shown in figure 7. Then the LMS algorithm outputs for audio signal input, with N=1000, µ=0.007 is as shown in figure 8 and Figure 9 shows the Adaptive filter output.

IV. SUMMARY AND FUTURE WORK

The LMS algorithm for a particular audio file is considered and the corresponding desired signal, adaptive filter output signal, estimation error and mean square error are plotted.

The future work to be done is about the same to input the audio file as a input for different algorithms namely NLMS (Normalized Least Mean Squares) Algorithm, VSSLMS (Variable Step Size Least Mean Squares) Algorithm, VSSNLMS (Variable Step Size Normalized Least Mean Squares) Algorithm and also with RLS Algorithm.

Once determined the above mentioned parameters using different algorithms a comparison is done with respect to the calculations and choosing the best for the above mentioned project and implementing the same using processor or at the gate level implementation.

V. REFERENCES