CHAPTER 3

ADAPTIVE NOISE CANCELLATION BASED ON
PROPOSED VARIABLE STEP SIZE NORMALIZED
DIFFERENTIAL LEAST MEAN SQUARE ALGORITHM

3.1 INTRODUCTION

In the recent days, interest in adaptive systems has increased, leading to widespread use of adaptive techniques in the fields such as Communications, Signal Processing, speech processing, and control systems. Adaptive systems adapt to the environmental changes and search for the optimum system parameters based on a reference signal. In the case of a filter, the system parameters are the tap weights of the filter. The performance of an adaptive algorithm is highly dependent on the reference input and additive noise statistics. In the digital communication systems, efficient bandwidth utilization is economically important to maximizing profits, while at the same time maintaining performance and reliability.

More importantly, the adaptive filter solution has to be relatively simple and not complicated, which often leads to the use of the conventional LMS algorithm. However, the performance of the LMS algorithm is often sub-optimal and the convergence rate are small. This, therefore, provides the motivation to explore and study VSSLMS adaptive algorithms for various applications.
STT model is proposed for Hearing impaired Persons by removing Noise from Speech signal using Various Adaptive Algorithms. LMS Algorithm is generally used for Noise Cancellation. Among Various LMS Algorithms Normalized Differential LMS and VSSLMS are analyzed. By combining NDLMS and VSSLMS, Novel Adaptive Algorithm is Proposed to remove the Noise From the Speech Signal. A proposed Algorithm Improved Adaptive Filter Based Noise Cancellation technique for Speech signals is used for Noise Cancellation.

3.2 WIENER FILTER

One of the important applications of wiener filter is noise suppression. These are a class of linear optimum discrete time filters known collectively as Wiener filters (Widrow 1975). Wiener filters are a special class of transversal FIR filters that builds upon the MSE cost function to arrive at an optimal filter tap weight vector, which reduces the MSE to a minimum. So Weiner filter reduces the MSE between the desired process and estimated random process.

3.2.1 Mean Square Error Criterion

Figure 3.1 illustrates a linear filter with the aim of estimating the desired signal d (n) from input x (n). Assume that d (n) and x (n) are samples of infinite length, random processes (Rekha 2010). In optimum filter design, input signal and noise signal are viewed as stochastic processes. The filter is based on minimization of the mean square value of the difference between the actual filter output and some desired output, as shown in Figure 3.1.
The requirement of the filter is to make the estimation error $e(n)$ as minimum as possible in some statistical sense by controlling the impulse response coefficients $w_0, w_1, \ldots w_{N-1}$.

Two basic restrictions are:

1. The filter is linear, which makes mathematical analysis easy to handle.

2. The filter is an FIR (symmetrical and odd ordered) filter.

The wiener filter output is $y(n)$ and the estimation error is given by $e(n)$. The performance of the filter is determined by the size of the estimation error, that is, a smaller estimation error indicates a better filter performance. As the estimation error approaches zero, the filter output $y(n)$ approaches the desired signal $d(n)$. So it is clear that the estimation error is required to be as minimum as possible. In other words, in the design of the filter parameters, an appropriate function of this estimation error as cost function is chosen and the set of filter parameters is selected, which optimizes the cost function of the filter.
In Wiener filters, the cost function is chosen to be

\[ \tilde{z} = E[e(n)^2] \]  

(3.1)

Where \( E[.] \) denotes the expectation average since both \( d(n) \) and \( x(n) \) are random processes.

### 3.2.2 Wiener Filter: Transversal, Real Valued Case

Consider an adaptive transversal filter structure as shown in Figure 3.2. Assume that the filter input \( x(n) \) and the desired response \( d(n) \) are real valued stationary processes. The tap weights of the filter \( w_0, w_1, \ldots, w_{N-1} \) are also assumed to be real valued, where \( N \) equals the number of delay units or tap weights.

The filter input \( x(n) \) and tap weight vectors, \( w \), can be defined as column vectors,

\[ x(n) = [x(n), x(n-1), \ldots, x(n-N+1)]' \]  

(3.2)

\[ w = [w_0, w_1, \ldots, w_{N-1}]' \]  

(3.3)

The Wiener filter output is defined as,

\[ y(n) = \sum_{i=0}^{N-1} w_i x(n-i) = w' x(n) = x'(n) w \]  

(3.4)

Subsequently, the error signal of the filter can be written as,

\[ e(n) = d(n) - y(n) = d(n) - w' x(n) = x'(n) w \]  

(3.5)
Substituting (3.5) into (3.1), the cost function is obtained as,

\[ z = E[e(n)^2] = E[(d(n) - w \cdot x(n))(d(n) - x'(n)w)] \]  
(3.6)

By expanding the expression (3.6), the expression becomes,

\[ z = E[d(n)^2] - E[d(n)x'(n)w] - E[d(n)w'x(n)] + E[w'x(n)x'(n)w] \]  
(3.7)

Since, the variable ‘w’ is not a random variable, we obtain

\[ z = E[d(n)^2] - E[d(n)x'(n)]w - w'E[d(n)x(n)] + w'E[x(n)x'(n)]w \]  
(3.8)

\[ \]

![Figure 3.2 Structure of an Adaptive Transversal filter](image-url)
Next, it can express \( E \left[ d \left( n \right) \times \left( n \right) \right] \) as a \( N \times 1 \) cross correlation vector

\[
P = E \left[ d \left( n \right) \times \left( n \right) \right] = \left[ p_0, p_1, \ldots, \ldots, p_{N-1} \right]'
\] (3.9)

And \( E \left[ x \left( n \right) \times \left( n \right) \right] \) as a \( N \times N \) autocorrelation matrix \( R \)

\[
R = E \left[ x \left( n \right) \times \left( n \right) \right] = \begin{bmatrix}
    r_{00} & r_{01} & r_{02} & \cdots & r_{0,N-1} \\
    r_{10} & r_{11} & r_{12} & \cdots & r_{1,N-1} \\
    r_{20} & r_{21} & r_{22} & \cdots & r_{2,N-1} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    r_{N-1,0} & r_{N-1,1} & r_{N-1,2} & \cdots & r_{N-1,N-1}
\end{bmatrix}
\] (3.10)

From (3.9), \( p' = E \left[ d \left( n \right) \times \left( n \right) \right] \) and hence \( p'w = w'p \)

This implies that \( E \left[ d \left( n \right) x \left( n \right) \right] w = E \left[ d \left( n \right) x \left( n \right) \right] w' \)

Subsequently, we get

\[
\xi = E\left[ d \left( n \right)^2 \right] - E\left[ d \left( n \right) x \left( n \right) \right] w - w' E\left[ d \left( n \right) x \left( n \right) \right] + w' E\left[ x \left( n \right) x \left( n \right) \right] w
\] (3.11)

\[
= E\left[ d \left( n \right)^2 \right] - 2p'w + w' R w
\] (3.12)

This is a quadratic function of tap weight vector ‘\( w \)’ with a single global minimum. To obtain the set of filter tap weights that minimize the cost function, \( \xi \), solve the system of equations that results from setting the partial derivatives of \( \xi \) with respect to every tap weight of the filter i.e. the gradient vector to zero. That is

\[
\frac{\partial \xi}{\partial w_j} = 0
\]
For \( i = 0, 1, \ldots, N-1 \) where \( N \) = Number of tap weights

The gradient vector in (3.12) can also be expressed as

\[
\nabla \xi = 0
\]

(3.13)

Where, \( \nabla \) is the gradient operator defined as a column vector

\[
\nabla = \left[ \frac{\partial}{\partial w_0} \frac{\partial}{\partial w_1} \frac{\partial}{\partial w_{N-1}} \right]
\]

(3.14)

and zero on the right hand side of (3.13) denotes the column vector consisting of \( N \) zero. It has been further proved that the partial derivatives of \( \xi \) with respect to the filter tap weights can be solved such that,

\[
\nabla \xi = 2Rw - 2p
\]

(3.15)

By assuming \( \nabla \xi = 0 \), the following equation is obtained, in which the optimum set of Wiener filter tap weights can be obtained,

\[
Rw = p
\]

This implies that,

\[
w = R^{-1} p = w_0
\]

(3.16)

Where \( w_0 \) indicates the optimum tap weight vector. This equation is known as the Wiener’s equation and can be solved to obtain the tap weights vector, which corresponds to the minimum point of the cost function.
3.2.3 Iterative Search Algorithm

It has been illustrated clearly in the previous section that the Wiener-Hopf expression can be solved to obtain the optimum filter tap weights by minimizing the cost function (Rekha 2010), if the required statistics of the signals ‘R’ and ‘p’ are available. Although, it presents computational complexity, when the filter contains a large number of tap weights and when the input data rate is high. An alternative is to use an iterative search algorithm that starts at some arbitrary initial point in the tap-weight vector space and moves progressively towards the optimum filter tap weight vector in steps. Each step is chosen with the aim of reducing the cost function of the filter.

The principle of finding the optimum filter tap weight vector by the progressive reduction of the underlying cost function by means of an iterative algorithm is essential to the development of adaptive algorithms. Adaptive algorithms are actually iterative search algorithms derived for minimizing the cost function.

3.2.4 Method of Steepest Descent

The steepest descent method is also called as Gradient Descent method. Assume that the cost function to be reduced is convex; this can begin with an arbitrary point on the performance surface and take a small step in the direction in which the cost function reduces fastest. This results to a step along the steepest descent slope of the performance at that point. Repeating this consecutively, convergence towards the base of the performance surface is guaranteed.

The method of steepest descent (Rekha 2010) is an alternate iterative search method to find \( w_0 \). This method belongs to a family of
iterative methods of optimization. It is a general scheme that performs an iterative search for a minimum point of any convex function of a set of parameters. Here, this method is implemented in transversal filter with the convex function referring to the cost function and the set of parameters referring to the filter tap weights. It uses the following procedures to search the minimum point of the cost function of a set of filter tap weights.

Step 1. Start with an initial condition of the adaptive filter tap weights whose optimum values are to be found for reducing the cost function. Set all the tap weights to zero Unless some prior knowledge is available, i.e. the initial weight is \( w(0) \).

Step 2. Use this initial condition to compute the gradient vector of the cost function with respect to the tap weights at the present input point.

Step 3. Update the tap weights by taking steps in the reverse direction of the gradient vector obtained in step 2. This corresponds to step in the direction of the steepest descent in the cost function at the present input. Also, the size of the step is chosen proportional to the size of the gradient vector.

Step 4. Go back to Step 2, and iterate the same process until no further significant change is observed in the tap weights i.e. the search has converged to an optimal point. According to the above procedures, if \( w(n) \) is the tap weights vector at the \( n^{th} \) iteration , then the following recursive equation is used to update \( w(n) \).

\[
w(n + 1) = w(n) - \mu \nabla_n \zeta
\]  

(3.17)

where \( \mu \) is called step size , and \( \nabla_n \zeta \) denotes the gradient vector evaluated at the point \( w = w(n) \).
### 3.2.5 Error Performance Surface

The estimation error $e(n)$ can be given as:

$$E\{e(n)\} = d(n) - \sum_{i=0}^{N-1} w_i x(n-i)$$

(3.18)

The cost function can be written as:

$$\xi = E\left[\frac{1}{2} (d(n))^2\right] - \sum_{i=0}^{N-1} w_i^* E\{x(n-i)\} - \sum_{i=0}^{N-1} w_i^* E\{x^*(n-i)d(n)\} +$$

$$\sum_{i=0}^{N-1} \sum_{k=0}^{N-1} w_i^* w_j E\{x^*(n-i)x^*(n-k)\}$$

(3.19)

The cost function or the mean squared error is precisely a second order function of the tap weights in the filter. Since ‘w’ can assume a continuum of values in the $N$ dimensional $w$-plane, the dependence of the cost function depends on the tap weights $w_0, w_1, \ldots, w_{N-1}$ may be visualized as a bowl shaped $(N+1)$-dimensional surface with $N$ degrees of freedom represented by the tap weights of the filter. The surface so described is called the error performance surface of the transversal filter. The surface is characterized by a unique minimum, where the cost function $\xi$ attains its minimum value. At this point, the gradient vector $\nabla \xi$ is identically zero. The height $\xi$ corresponds to the physical description of filtering the signal $x(n-i)$ with the fixed filter weight $w$, from which a prediction error signal $e(n)$ with power of $\xi$ is generated. Some filter setting $w_0 = (w_{00}, w_{01})$ will produce the MMSE ($w_0$ is the optimum filter tap weight vector). This theory is the base of basic adaptive algorithms of adaptive signal processing.
3.3 NOISE CANCELLATION USING LMS ALGORITHM

In summary, LMS algorithm for every search iteration, can be expressed in the form of three operations:

1. Filter output: \( y(n) = w \times (n) \)

2. Error Estimation: \( e(n) = d(n) - y(n) \)

3. Tap Weight adaptation: \( w(n+1) = w(n) + \mu e(n) \times (n) \)

The second operation defines the estimation error \( e(n) \), computed based on the current estimate of the tap weights vector when the first operation. \( \mu e(n) \times (n) \) in the third operation refers to the correction that is applied to the previous estimate of the tap weights vector. (Corresponding to step 3 of the method of steepest descent). The desired signal has been obtained by convolving the input signal (vocal wav file) with the appropriately defined impulse response of the acoustic environment. The plot for the output of the adaptive filter shows that an estimate of the desired noise signal is obtained as its output and can thus be subtracted from the noise signal to minimize the error. The noise canceller identifies the transfer function of the acoustic noise path i.e. the impulse response \( h(n) \) between the loudspeaker and the microphone. Since the impulse response variable as a person moves and varies with the environment, an adaptive filter is used to identify \( h(n) \). The desired signal is obtained by convolving the input signal with the impulse response of the acoustic environment and a noise replica is created at the output of the adaptive filter. The noise replica \( y(n) \) is then subtracted from the noise signal \( d(n) \) to give the error \( e(n) = d(n) - y(n) \). The simplicity and robustness of the LMS updating equation are the most important features and lead to many successful developments of other GNDN-based algorithms.
The LMS algorithm is introduced as a way to recursively adjust the parameters of \( w(n) \) of a linear filter with the goal of minimizing the error between a given desired signal and the output of the linear filter.

**Stochastic Gradient Descent**

- **Non Mean Square**
  - Non-Linearity
  - Modified LMS
  - Higher Order
- **Mean Square**
  - Mixed-Order
  - LMS
  - NLMS
  - NDLMS
  - VSSLMS

**Figure 3.3 The Grouping of LMS Algorithms**

Figure 3.3 shows the stochastic GNDN classification, it is classified into mean Square and Non Mean Square.

### 3.4 LMS ALGORITHM

The LMS belongs to the class of GNDN-based algorithms. It is the most widely used algorithm for applications like channel equalization, noise cancellation and echo cancellation. The simplicity and robustness of the LMS updating equation are the most important features and leads to many successful developments of other GNDN-based algorithms. The LMS algorithm is introduced as a way to recursively adjust the parameters of \( w(n) \) of a linear filter with the goal of minimizing the error between a given desired signal and the output of the linear filter. LMS is one of the many related algorithms which are appropriate for the task and whole families of
algorithms have been developed which can address a variety of problem settings, computational restrictions and minimization criteria.

### 3.4.1 LMS updating Equation

The conventional adaptive LMS algorithm is a stochastic implementation of the method of steepest descent algorithm. Substitute the cost function into (3.17), we get

\[ w(n+1) = w(n) - \mu \nabla e^2(n) \quad (3.20) \]

Where \( \nabla \) is the gradient operator, \( w = [w_0, w_1, \ldots, \ldots, w_{N-1}] \) and \( \mu \) is the step size.

Using (3.14) and (3.18), we note that the \( i \) th element of \( \nabla e^2(n) \) can be given by,

\[ \frac{\partial e^2(n)}{\partial w_i} = 2e(n) \frac{\partial e(n)}{\partial w_i} \quad (3.21) \]

The derivative of the first equality in (3.5) is

\[ \frac{\partial e(n)}{\partial w_i} = 0 - \frac{\partial y(n)}{\partial w_i} \quad (3.22) \]

Since \( d(n) \) is independent of \( w_i \)

Substituting (3.22) into (3.21)

\[ \frac{\partial e^2(n)}{\partial w_i} = -2e(n) \frac{\partial y(n)}{\partial w_i} \quad (3.23) \]
Then take the derivative of \( y(n) \) in the first equality of (3.4) and substituting the result in (3.21)

\[
\frac{\hat{e}^2(n)}{\hat{w}_i} = -2e(n)x(n - i)
\]  \hspace{1cm} (3.24)

Then using (3.14) and (3.22), we get,

\[

\nabla e^2(n) = -2e(n)x(n)
\]  \hspace{1cm} (3.25)

Where \( x(n) = [x(n) x(n-1) \ldots x(n-N+1)]' \) as stated in (3.2)

Substituting (3.23) into (3.18) gives

\[
W(n+1) = w(n) + 2\mu e(n)x(n)
\]  \hspace{1cm} (3.26)

This equation is known as the LMS tap weight adaptation. During every iteration, the tap weight adaptation updates the tap weight vector with the direction of minimizing the cost function to find \( w_0 \). It is noteworthy that the factor 2 in the last term of (3.24) is left out by some authors. If \( \frac{1}{2} \) is multiplied to both sides by (3.15) to cancel out the factor of 2 on the right hand side, we get \( (\nabla e^2)/2 = Rw - p \). Substituting this into (3.17) leads to

\[
w(n + 1) = w(n) - \frac{1}{2} \mu \nabla e^2(n)
\]  \hspace{1cm} (3.27)

Consequently,

\[
w(n + 1) = w(n) - \frac{1}{2} \mu \nabla e^2(n)
\]  \hspace{1cm} (3.28)
3.4.2 Step Size Optimization

The step size plays a significant role in controlling the performance of the Adaptive filter. The problem of the conventional LMS algorithm is that the fixed step size governs the trade-off between the convergence rate and the steady state error. A large step reduces the transient time but will result in a larger steady state MSE. On the other hand, to achieve a smaller steady state error, a small step size has to be used which will cause a slower convergence rate. Selection of the step size is application specific to the priority requirements such as fast convergence, robustness, tracking capability and adaptation accuracy. To achieve overall high performance, a sequence of optimum step sizes can be used to imitate the RLS performance; known as the time-varying step size algorithm. It is possible that the optimum step size can be determined by solving the difference equation for the MMSE at each iteration. A number of time varying step sizes have been proposed to improve the step size trade-off effect, which are a vital tool in achieving desired adaptive performance.

To ensure stability (or convergence) of the LMS algorithm, the step size parameter is bounded by the following equation:

$$0 < \mu < \left( \frac{2}{\text{tap weight power}} \right)$$  \hspace{1cm} (3.29)

where tap weight input power is the sum of the mean squared values of all the tap inputs in the transversal filter and is given by

$$\sum_{k=0}^{\infty} E[|x(n-k)|^2]$$

Note that the upper bound is dependent on the statistics of filter input signals. Intuitively, we may interpret from this equation that when the power of the input signals varies greatly, a smaller step size is required to avoid instability or gradient noise amplification.
3.4.3 Effect of Power Spectral Density of the Input Signal

The convergence rate of the LMS algorithm deteriorates with higher input correlation levels due to a greater interaction among the adaptive tap coefficients. To ensure improved convergence rate, the LMS algorithm requires input signals to have equal excitation over the whole range of frequency.

3.4.4 Effect of Filter Tap Length

In order to ensure better asymptotic performance, the length of the LMS adaptive FIR filter must be sufficient to cover the impulse response of the unknown channel. However, this may lead to increased computational complexity when the impulse response of the unknown channel is long. Moreover, this may lead to poor convergence rates when the input signals are highly correlated.

3.4.5 Effect of Input Signal Power

The correction term $\mu e(n) x(n)$ in the LMS tap weight adaptation is directly proportional to the tap input $x(n)$. In other words, the convergence rate and stability of the LMS algorithm is directly dependent on the value of $\mu \sigma_x^2$, where $\sigma_x^2$ is the variance of power of the input signal. When the power of the input signal $x(n)$ is huge or varies greatly, the LMS algorithm demonstrates an uneven behavior known as gradient noise amplification. Consequently, the LMS algorithm becomes unstable and therefore will not lead to the optimal solution. To deal with this problem, a modified version of LMS known as Normalized algorithm can be implemented.
3.5 NORMALIZED LMS (NLMS) ALGORITHM

Determining the upper bound step size is a problem for the variable step size algorithm if the input signal to the adaptive filter is non-stationary. The fastest convergence is achieved by the choice of step size as follows:

\[ \mu_{\text{max}} = \frac{1}{(\lambda_{\text{max}} + \lambda_{\text{min}})} \]  

(3.30)

However, experimental results have shown that the maximum step size in Equation (3.30) does not always produce the stable and fast convergence, according to Raymond Kwong & Edward Johnston (1992), (2/3) \( \mu_{\text{max}} \) is a rule of thumb for LMS algorithm. To increase the convergence speed Normalized LMS (NLMS) algorithm is a natural choice, because the step size is normalized by the input signal power. The NLMS has been always the favored choice of algorithm for faster convergence speed and for non-stationary input. The value of \( \mu \sigma_x^2 \) directly affects the convergence rate and stability of the LMS filter. As the name implies, the NMLS algorithm is an effective approach to overcome this dependence, particularly when the variation of the input signal power is large, by normalizing the update step size with an estimate of the input signal variance. In practice, the correction term applied to the estimated tap weight vector \( w(n) \) at the \( n^{\text{th}} \) iteration is normalized with respect to the squared Euclidean norm of the tap input \( x(n) \) at the \( (n-1)^{\text{th}} \) iteration,

\[ w(n + 1) = w(n) + \frac{\epsilon}{\| x(n) \|^2} e(n) x(n) \]  

(3.31)

where

\[ \| x(n) \| = \text{Euclidean Norm} \]
Apparently, the convergence rate of the NMLS algorithm is directly proportional to the NLMS adaptation constant $\alpha$, i.e. the NLMS algorithm is independent of the input signal power. By choosing $\alpha$ so as to optimize the convergence rates of the algorithms, the NLMS algorithm converges more rapidly than the LMS algorithm. It can also be stated that the NLMS is convergent in mean square if the adaptation constant $\alpha$ is from $0$ to $2$ (however a more practical step size for NLMS is always less than unity).

$$0 < \alpha < 2$$

Despite this particular edge that the NLMS exhibits, it does have a slight problem of its own. When the input vector $x(n)$ is small, instability may occur since we are trying to perform numerical division by small value of the Euclidean Norm. However, this can be easily overcome by appending a positive constant to the denominator in (3.28) such that

$$w(n+1) = w(n) + \frac{\alpha}{c + \|x(n)\|^2} e(n)x(n)$$

(3.32)

Where $c, \|x(n)\|^2$ is the normalization factor. With this, more robust and reliable implementation of the NLMS algorithm is obtained.

There is an immense amount of literature on variable step size methods, which make the most of the trade-off between fast convergence and lower steady state error. Gear shifting is a popular approach by Widrow, which is based on using large step size values when the filter weights are far from the optimal solution and small step size values when near to the optimum solution. A number of time-varying step-size algorithms have been proposed to enhance the performance of the conventional LMS algorithm. Several criteria have been used: squared instantaneous error, sign changes of successive samples of the gradient, attempting to reduce the squared error at
each instant, or cross correlation of input and error. The variable step size algorithm given by Raymond Kwong & Edward Johnston (1992) has been discussed in the next section and its modification has also been proposed. The modified VSS LMS algorithm gives a better performance as compared to other variable step size algorithms.

3.6 VARIABLE STEP SIZE LMS ALGORITHM

ANC using VSSLMS Algorithm is used as an Adaptive Filter to remove the noise from the speech signal. For STT conversion model, noiseless speech is required for STT Transformation.

Based on the error squared power, Raymond Kwong & Edward Johnston (1992) proposed a simpler VSSLMS algorithm. The error power reflects the convergence state of the adaptive filter, where a converging system has a higher error power while the converged system has a smaller error power. Therefore, scalar step size increases or decreases as the squared error increases or decreases, thereby allowing the adaptive filter to track changes in the system and produces a smaller steady state error.

In VSSLMS algorithm the step size adjustment is controlled by the square of the prediction error. A large prediction error will cause the step size to increase to provide faster tracking while a small prediction error will result in a decrease in the step size to yield smaller maladjustment. By considering the adaptive filtering or the system identification problem, a set of filter weights is adjusted so that the system output tracks a desired signal. Let the input vector of the system be denoted by \( X_k \), and the desired scalar output be \( d_k \). These processes are related by the equation

\[
d_k = X_k^T W_k^* - e_k
\]  

(3.33)
where $e_k$ is a zero mean Gaussian independent sequence, independent of the input process $X_k$.

Two cases will be considered:

1. $W^*_k$ equals a constant $W^*$.

2. $W^*_k$ is randomly varying according to the equation.

Here the first case will be referred to as a stationary system or environment, the second a nonstationary system or environment.

$$W^*_{k+1} = a W^*_k + Z_k$$ (3.34)

Here $a$ is less than but close to 1, and $Z_k$, is an independent zero mean sequence, independent of $X_k$ and $e_k$ with covariance $E(Z_k Z_j^T) = \sigma^2 \delta_{kj}$, $\delta_{kj}$ being the Kronecker delta function. The input process $X_k$, is assumed to be a zero mean independent sequence with covariance $E(X_k X_k^T = R)$, a positive definite matrix.

The LMS type adaptive algorithm is a gradient search algorithm which computes a set of weights $W_k$ that seeks to minimize $E(d_k - X_k^T W_k)^2$. The algorithm is of the form

$$W_{k+1} = W_k + \mu_k X_k \in_k$$ (3.35)

where

$$\in_k = d_k - X_k^T W_k$$ (3.36)
and $\mu_k$ is the step size. In the standard LMS algorithm $\mu_k$ is a constant. In some cases $\mu_k$ is time varying with its value determined by the number of sign changes of an error surface gradient estimate. In variable step size algorithm, for adjusting the step size $\mu_k$,

$$\dot{\mu}_{k+1} = \alpha \mu_k + \gamma \epsilon_k^2$$

with

$$0 < \alpha < 1, \quad \gamma > 0$$

and

$$\mu_{k+1} = \begin{cases} 
\mu_{\text{max}} \quad \text{if } \dot{\mu}_{k+1} > \mu_{\text{max}} \\
\mu_{\text{min}} \quad \text{if } \dot{\mu}_{k+1} < \mu_{\text{min}} \\
\mu_{k+1} \quad \text{otherwise}
\end{cases}$$

(3.37)

$$\mu_{\text{min}}, \mu_{\text{max}}$$

where $0 < \mu_{\text{min}} < \mu_{\text{max}}$. The initial step size $\mu_0$ is usually taken to be $\mu_{\text{max}}$, although the algorithm is not sensitive to the choice. From (3.38), it can be seen that, the step size $\mu_k$ is always positive and is controlled by the size of the prediction error and the parameters $\alpha$ and $\gamma$. A large prediction error increases the step size to provide faster tracking. If the prediction error decreases, the step size will be decreased to reduce the maladjustment. A constant $\mu_{\text{max}}$ constant to ensure that the MSE (MSE) of the algorithm remains bounded. A sufficient condition for $\mu_{\text{max}}$, to guarantee bounded MSE is

$$\mu_{\text{max}} \leq \frac{2}{3\text{tr}(R)}$$

(3.39)
is chosen to provide a minimum level of tracking ability. Usually, $\mu_{min}$ will be near the value of $\mu$ that would be chosen for the fixed step size (FSS) algorithm. $\alpha$ must be chosen in the range $(0, 1)$. The parameter $\gamma$ is usually small and may be chosen in conjunction with $\alpha$ to meet the maladjustment requirements.

3.6.1 Experimental set up of Noise cancellation using VSSLMS Algorithm

STT model is proposed for Hearing impaired Persons by removing noise from Speech signal using VSSLMS Algorithm. Speech Signal is tested with VSSLMS Algorithm. Figure 3.4 shows the block diagram of ANC model using Adaptive Filter to remove Noise from Speech Signal. Here VSSLMS Algorithm is used in Adaptive filters to remove Noise from Speech signal. The speaker produces the output of the speech signal and simulation results are performed using MATLAB, which are discussed in the Next Chapter. Figure 3.5 shows the simulated MATLAB input, output and MSE of the speech signal.

![Experimental Setup of Adaptive Filter using VSSLMS Algorithm](image)

Figure 3.4 Experimental Setup of Adaptive Filter using VSSLMS Algorithm
Figure 3.5  Input speech signal corrupted with Noise and VSSLMS Output of Speech Signal

3.7  NORMALIZED DIFFERENTIAL LMS ALGORITHM

ANC using Normalized Differential LMS Algorithm is used as an Adaptive Filter to remove the noise from the speech signal. For STT conversion model, noiseless speech required for STT Transformation.

In NDLMS Algorithm a different approach is considered for weight adjustment. The motivation is to design an LMS algorithm that can handle both the strong and the weak target signals. Hence whenever the filter inputs and outputs fluctuate more or less, the weights should be adjusted accordingly. Some adverse effects may occur when we use an LMS algorithm as described in (1). Under the circumstance that the weight $W(k)$ has been approaching the optimal value $W_{opt}$, when $e(k)$ are relatively large for a while,
Equation (1) implies that $W(k)$ will deviate away from the optimal weight in a relatively large manner. This causes the adaptive weights to fluctuate around their optimal values for a relatively longer period; and it is not possible to make the steady-state EMSE arbitrarily small by reducing $\mu$.

Later Joshua Zhang and Heng-Ming Tai (2007) made a slight modification in LMS algorithm equations and they proposes the Normalized Differential LMS Algorithm. It is mainly used for finding the error signal and the filter weights. In this case the algorithm improves the steady state performance for cancelling noise in speech processing. NDLMS Algorithm proposes to adjust the weight according to the difference of the signals, instead of the signals themselves. Similar to the normalized LMS scheme, the weight update equation becomes

$$w(n+1) = w(n) + \frac{\mu}{\varepsilon + \| \nabla y(k) \|^2} \nabla x(n) * \nabla e(n)$$

(3.40)

This algorithm is suited for speech processing where the magnitudes of speech signals generally vary slowly. The performance criterion behind this proposal is the MSE difference criterion

$$J(k) = E\{ |e(k) - e(k-1)|^2 \}$$

(3.41)

If $\{ \nabla x(k) \}$ is an independent random sequence and the elements of $\nabla x(k)$ are independent and identically distributed satisfying the following conditions,

$$E\{ \nabla x(i) \} = 0 \text{ and } E\{ \nabla x(i) \nabla x(j) \} = \sigma^2_{\nabla x} \delta(i-j)$$
Then the optimal adjustment gain is \( \mu^* = 12 \) and the optimal convergence rate is \( \eta^* = 1 - 1/N \). NDLMS algorithm is used for both the strong and the weak target signals, but the step size \( \mu \) value is not variable, so convergence rate is low.

### 3.7.1 Experimental set up of Noise Cancellation using NDLMS Algorithm

The speech signal is tested with NDLMS Algorithm. Figure 3.6 shows the block diagram of ANC model using Adaptive Filter to remove Noise from Speech Signal. Here NDLMS Algorithm is used in Adaptive filters to remove Noise from Speech signal. The speaker produces the output of the speech signal and simulation results are performed using MATLAB, which are discussed in the Next Chapter. Figure 3.7 shows the simulated MATLAB input, output and MSE of the speech signal.

![Block Diagram](image)

**Figure 3.6  Experimental Setup of Adaptive Filter using NDLMS Algorithm**
3.8 PROPOSED ADAPTIVE ALGORITHM

ANC using Variable Step Size Normalized Differential LMS Algorithm is used as an Adaptive Filter to remove the noise from the speech signal. For STT conversion model, noiseless speech required for STT Transformation.

The NDLMS and VSSLMS are combined together and an improved efficient algorithm called Variable Step Size Normalized Differential LMS (VSSNDLMS) algorithm is proposed to enhance speech processing. The objective of proposing this algorithm is to design an effective adaptive filter to remove the noise and to improve the quality of speech signals. Figure 3.4 shows the proposed algorithm using VSSNDLMS method.
The Normalized Differential LMS Algorithm is particularly suited for slowly varying signals and is less sensitive to the desire signal power variation compared to the existing Algorithms. Moreover, the excess error and maladjustment by NDLMS are much less than that of existing Algorithms. VSSLMS Algorithm is used to reduce the trade off between maladjustment and tracking ability of the fixed step size LMS Algorithm. The VSSLMS also reduces sensitivity of the maladjustment to the level of non-stationary. The features of NDLMS and VSSLMS are combined, VSSNDLMS Algorithm is proposed.

In case of LMS algorithm under non-stationary environment, errors occur which leads to deviation of filter weights from the optimal weight of the filter. The proposed algorithm satisfies this criterion by adjusting the step size. The VSSLMS algorithm converges faster and NDLMS algorithm has minimal MSE. By combining the VSS and NDLMS, the VSSNDLMS algorithm converges fast with MMSE.

Figure 3.8 Shows a typical Adaptive Noise Canceller has two inputs: 1) a primary input, \(d(k)\), composed of the desired signal, \(s(k)\), corrupted by a filtered additive noise signal \(v(k)\), and 2) a reference input, \(x(k)\), which is a different filter noise and the input to the adaptive filter. \(X(k)\) is assumed to be correlated with \(v(k)\) and uncorrelated with \(s(k)\). The output of Adaptive Noise Canceller is \(e(k)\), the Difference between the primary input and the output of the Adaptive filter. The noise signal is represented by \(n(k)\).

The difference in input \(x(k)\) and difference in output \(e(k)\) is given to VSSNDLMS Algorithm.
From Equation 3.35, 3.36 and 3.40 VSSNLMS algorithm is proposed, the equation for updating the coefficient is given by,
\[
w(n+1) = w(n) + \frac{\mu_{\text{var}}}{\mu + \| \nabla X(k) \| ^2} \times \nabla x(n) \times e(n)
\]  
(3.42)

Where,
\[
\nabla e(k) = e(k) - e(k - 1)
\]  
(3.43)

\[
\nabla x(k) = x(k) - x(k - 1)
\]  
(3.44)

and the \(\mu_{\text{var}}\) is the variable step size which is given by,
\[
\mu_{\text{var}} = \begin{cases} 
\mu_{\text{max}} \text{ if } \mu_i > \mu_{\text{max}} \\
\mu_{\text{min}} \text{ if } \mu_i < \mu_{\text{min}}
\end{cases}
\]  
(3.45)

Where, \(\mu_i = \alpha \cdot \mu - \gamma \frac{1}{k}\) and by selecting the proper value of \(\alpha\) and \(\gamma\) we can get better performance. Depending upon the speech signal, the value of \(\alpha\) and \(\gamma\) vary. The constant \(\mu_{\text{max}}\) is chosen to ensure that the mean-square error (MSE) of the algorithm remains bounded. \(\mu_{\text{min}}\) is chosen to provide a minimum level of tracking ability. Usually, \(\mu_{\text{min}}\) will be near the value of \(\mu\) that would be chosen for the fixed step size (FSS) algorithm. The purpose of the adaptive filter is to minimize the MSE (MSE). Therefore, the MSE quantity is essential to evaluate the performance of the adaptive filter.

\[
\text{MSE} = \mathbb{E}\{\| d(n) - \hat{d}(n) \| ^2\}
\]  
(3.46)

The quality of adaptation can be measured by the maladjustment, \(M\), which is the dimensionless ratio of the EMSE to the MMSE in the steady – state environment.

\[
M = \frac{\text{EMSE}_{ss}}{\text{MMSE}}
\]  
(3.47)
The MMSE is the optimum filtering error, which represents the portion of primary signal that cannot be cancelled by optimal weight. It can be defined by averaging the output signal power over samples after which the algorithm reaches the steady state, that is,

$$\text{MMSE} = \frac{1}{K - P} \sum_{k=p}^{K} |s(k)|^2$$

(3.48)

Where \( K \) is the total number of the speech samples

\( P \) is the number of samples at which the algorithm reaches steady state.

The EMSE, caused by the deviation of the weights from their optimal values, is defined as,

$$E\text{MSE}(k) = \frac{1}{N} \sum_{j=0}^{N-1} |e_1(k - j)|$$

(3.49)

Where \( e_1(k) = e(k) - s(k) \) is the residual error and \( N \) is the number of samples used in the estimation. The steady – state EMSE (\( \text{EMSE}_{SS} \)) is defined by taking the average of the EMSE over certain durations,

$$\text{EMSE}_{SS} = \frac{1}{K - P} \sum_{k=p}^{K} E\text{MSE}(k)$$

(3.50)

Thus improved Adaptive Filter Based Noise Cancellation techniques for speech signals is proposed.
3.8.1 Experimental set up of Noise cancellation using VSSNDLMS Algorithm

Speech Signal is tested with VSSNDLMS Algorithm. Figure 3.10 shows the block diagram of ANC using Improved Adaptive filter Based Noise cancellation Techniques for speech signals. Here Proposed VSSNDLMS Algorithm is used in Adaptive filters to remove Noise from Speech signal. The speaker produces the output of the speech signal and simulation results are performed using MATLAB, which are discussed in the Next Chapter. Figure 3.11 shows the simulated MATLAB input , output and MSE of the speech signal.

![Diagram of Adaptive Filter using VSSNDLMS Algorithm](image)

**Figure 3.10** Experimental Setup of Adaptive Filter using VSSNDLMS Algorithm
Figure 3.11 Input speech signal corrupted with Noise and NDLMS Output of Speech Signal
### 3.8.2 Summary of Proposed Algorithm

#### Table 3.1 Summary of VSSNDLMS Algorithm

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update weight</td>
<td>$w(n+1) = w(n) + \frac{\mu_{\text{var}}}{\varepsilon + | \nabla X(k) |^2} \times \nabla x(n) \times \nabla e(n)$</td>
</tr>
<tr>
<td></td>
<td>Where,</td>
</tr>
<tr>
<td></td>
<td>$\nabla e(k) = e(k) - e(k-1)$</td>
</tr>
<tr>
<td></td>
<td>$\nabla x(k) = x(k) - x(k-1)$</td>
</tr>
<tr>
<td></td>
<td>$\mu_{\text{var}}$ is the variable step size</td>
</tr>
<tr>
<td></td>
<td>$x(n)$ is input signal</td>
</tr>
<tr>
<td></td>
<td>$e(n)$ is error signal</td>
</tr>
<tr>
<td></td>
<td>$w(n)$ is weight vector</td>
</tr>
<tr>
<td>Step-size</td>
<td>$\mu_{\text{var}} = \begin{cases} \mu_{\text{max}} &amp; \text{if } \mu_i &gt; \mu_{\text{max}} \ \mu_{\text{min}} &amp; \text{if } \mu_i &lt; \mu_{\text{min}} \end{cases}$</td>
</tr>
<tr>
<td></td>
<td>Where,</td>
</tr>
<tr>
<td></td>
<td>$\mu_i = \alpha \times \mu + \gamma e_k^2$</td>
</tr>
<tr>
<td>MSE</td>
<td>$\text{MSE} = E{| d(n) - \hat{d}(n) |^2 }$</td>
</tr>
<tr>
<td></td>
<td>Where,</td>
</tr>
<tr>
<td></td>
<td>$d(n)$ is desire response</td>
</tr>
<tr>
<td>Mis-adjustment</td>
<td>$M = \frac{EMSE_{ss}}{MMSE}$</td>
</tr>
<tr>
<td></td>
<td>$MMSE = \frac{1}{K-P} \sum_{k=P}^{K}</td>
</tr>
<tr>
<td></td>
<td>$EMSE(k) = \frac{1}{N} \sum_{j=0}^{N-1}</td>
</tr>
<tr>
<td></td>
<td>$EMSE_{ss} = \frac{K}{K-P} \sum_{k=P}^{K} EMSE(k)$</td>
</tr>
</tbody>
</table>
3.8.3 Flowchart

Figure 3.12 shows the flow chart of the improved ANC algorithm using VSSNDLMS. Noisy speech signal is given as input, proposed VSSNDLMS Algorithm is selected, step size is chosen and filter operation is performed, finally Error and output are computed.

Step 1: Input speech signal in (*.wav) format

Step 2: Select the VSSNDLMS Algorithm

Step 3: Select step size

Step 4: Update weight vector

Step 5: Compute input and output

Step 6: Compute MSE

Step 7: Check error e (n) = 0, if no go to step 3

Step 8: Stop
Figure 3.12 Flowchart of improved Adaptive Filter Based Noise Cancellation Algorithm for Speech Signals
3.9 SUMMARY

The proposed algorithm aimed at designing an effective adaptive filter to remove the noise and to improve the quality of speech signals. The noiseless speech signal is required in the process of conversion of speech signals into text which is to be used by hearing impaired people. The Variable Step Size Normalized Differential LMS algorithm proposed which combines the features of NDLMS and VSSLMS algorithm. The property of Variable Step Size Normalized Differential LMS algorithm is faster convergence and MMSE. Among various LMS Algorithms like Normalized Differential LMS, VSSLMS the proposed algorithm proved to be of greater efficiency in noise cancellation. Performance analysis of Various Adaptive Algorithms is discussed in the Next Chapter with Simulation Results.