Fall 2015

A subband Kalman filter for echo cancellation

Rakesh Vijayakumar

Department:

A SUBBAND KALMAN FILTER FOR ECHO CANCELLATION

by

RAKESH VIJAYAKUMAR

A THESIS

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

2015

Approved by

Dr. Steven Grant, Advisor
Dr. Kurt Kosbar
Dr. Randy Moss

ABSTRACT

This thesis involves the implementation of a Kalman filter for the application of echo cancellation. This particular Kalman filter is implemented in the frequency domain, in subbands, so as to make it faster and of lesser calculational complexity for real time applications. To evaluate the functioning of this subband Kalman filter, comparison of the performance of the subband Kalman filter is done with respect to the original time domain Kalman filter, and also with other subband adaptive filters for echo cancellation such as the NLMS filter. Additionally, since background noise affects the working of any adaptive filter, the newly developed subband Kalman filters performance at different noise conditions is compared, and an attempt to keep track of and predict this noise is also performed.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr Steven Grant, for all the wonderful insight, knowledge, help and support provided to me during the course of this thesis. I would also like to thank my other committee members, Dr Randy Moss and Dr Kurt Kosbar, for their patience, inputs and feedbacks. I would like to thank the peers in my research group, for all the ideas brainstorming with them has provided. Finally, I would like to thank my family, especially my parents, for never giving up on me and supporting me through all my decisions.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

1. INTRODUCTION

1.1. THE ECHO CANCELLATION PROBLEM

In telephony, the near end speakers echo is the time delayed version of the speakers signal transmitted to the far end speaker, which is received along with the far end speakers signal as noise. There are two major types of echo:

1. Hybrid echo: This type is due to different electrical connections of the telephonic system between the two speakers- for all purposes of this thesis, it is assumed that this type of echo does not exist.

2. Acoustic echo: The more common type of echo in a phone call between two speakers is the acoustic echo, which occurs due to the near end speakers voice getting reflected across the receiving far end speakers surrounding and then returning back to the near end speaker as unwanted noise. The work of this thesis focuses on this type of echo.

The illustration of acoustic echo formation is as shown in Figure 1.1.

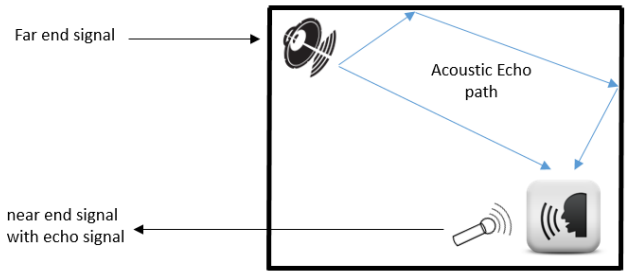

Figure 1.1. Illustration of acoustic echo

1.2. MMSE ADAPTIVE FILTER THEORY

The process of acoustic echo cancellation involves the determination of the acoustic echo path.. This section will give a brief overview of adaptive filter theory, focusing on the filters that use the minimization of the mean square error (MMSE) criterion as the condition for their optimality, since MMSE based linear filtering is simple, elegant, involves only second order statistics, and is useful in many practical applications[1]. The least square error approach of the Wiener filter is studied first, then its solution is extended to the normalized mean square error (NLMS) and Kalman filter algorithms.

1.2.1. MMSE Estimators. The block diagram for estimation using adaptive filters is as shown in Figure 1.2. The definition of the signals used in figure 1.2 is done in the context of a general estimator being used for the application of echo cancellation.

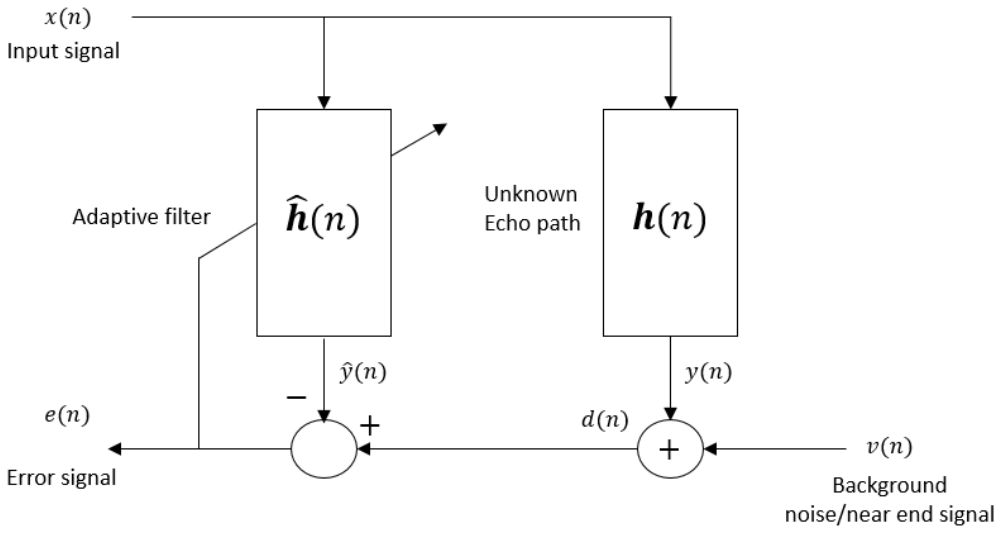

Figure 1.2. Block diagram for MMSE estimator adaptive filters

With reference to Figure 1.2, the following can be defined at any given point of time n:

1. Far end signal: $x(n)$ is the far end signal,which when reflected around the acoustic environment of the near end speaker then becomes the echo signal.

2. Echo path: $\mathbf{h}(\mathbf{n})$ is the echo path of the near end speaker and effectively acts as a tapped filter, which delays the far end signal $x(n)$ to produce the echo signal. The echo path is the unknown system to be estimated.

3. Echo signal: $y(n)$ is the echo signal produced when the far end speaker's signal goes through the echo path.

4. Background noise: $v(n)$ is the background noise, which is usually the near end speakers' signal, which gets added to the echo signal and needs to be taken into consideration when calculating the estimate of the echo path.

5. Measured signal: $d(n)$ is the measured signal, which is the addition of the echo signal and the background noise.

6. Echo path estimate: $\hat{\mathbf{h}}(\mathbf{n})$ is the estimate of the echo path that is calculated by the MMSE filter.

7. Estimate signal: $\hat{\mathbf{y}}(\mathbf{n})$ is the estimate signal obtained by passing the far end speaker's signal through the estimate of the echo path.

8. Error signal: $e(n)$ is the difference between the actual measured signal and the estimated signal.

Optimal signal estimation involves the successful estimation of the unknown quantity or signal $h(n)$ (in this case the echo path) using data from other related signals of interest, namely input signal $x(n)$ and unknown systems output or echo

signal $y(n)$. All these algorithms work on minimizing the square of the error signal $e(n)$ i.e. to minimize the difference between the signal estimate and the actual signal. Another important factor that influences the performance of the estimator is the measurement or background noise that is added to the output signal, $v(n)$.

MMSE estimators, such as the Wiener filter, are single value estimators,which looks at the entire history of the present and past samples of input data to make a single ensemble estimation at the present instance of time.

MMSE estimators such as the least mean squares filter(LMS), are time value estimators, which means that at every given instant of time, the estimate of the unknown variable is estimated and updated on the existing present and past values of input data. Such estimators are also known as recursive mean square estimators.

Depending on whether the estimators are ensemble estimators or recursive estimators, $\hat{h}$ as the estimate of $h(n)$, is either predicted once based on the causal history of the data, or is estimated at every time value iteration by a two-step process of prediction and update based on certain parameters. The difference in the update equations is what gives rise to the different types of MMSE adaptive filters, which are further discussed in detail in the following sections.

1.2.2. Echo Cancellation Signal Equations. This section looks at the context and usage of the defining equations for the various signals that were defined in the previous section. With reference to Figure 1.2, at any given dicrete time index n, the acoustic echo path of the near end speaker, of length L is given by

$$\mathbf{h} = [h_0 \; h_1 \; h_2 \; h_3 ... \; h_{L-1}]^T \tag{1.1}$$

The latest L number of samples of the input signal which then go through the echo path or the far end signal vector are then given by 1.2 below.

$$\mathbf{x}(n) = [x(n)\ x(n-1)\ x(n-2)\ x(n-3)\ ...\ x(n-L+1)]^T \tag{1.2}$$

The echo signal of the far end speaker is given by

$$y(n) = \mathbf{x}^T(n).\mathbf{h} \tag{1.3}$$

Let $v(n)$ be the background noise or the near end speaker's signal accordingly(during the scenario of double talk, $v(n)$ will be the near end speaker's signal [2], else $v(n)$ is background noise, generally white Gaussian noise) at the time instant n, whose variance is given by $\sigma_v^2(n)$. Then the total microphone signal is given by

$$d(n) = y(n) + v(n) \tag{1.4}$$

The goal of the echo canceller MMSE adaptive filter is to estimate the echo path $\mathbf{h}$ as

$$\hat{\mathbf{h}} = [\hat{h_0}\ \hat{h_1}\ \hat{h_2}\ \hat{h_3}\ \hat{h_{L-1}}]^T \tag{1.5}$$

The estimated echo signal is then calculated as

$$\hat{y}(n) = \mathbf{x}^T(n).\hat{\mathbf{h}}(n) \tag{1.6}$$

The error between the estimated echo and the actual echo is the residual error, given by

$$e(n) = d(n) - \hat{y}(n) \tag{1.7}$$

This error needs to be subject to the MMSE condition, which will be discussed in the next subsection.

1.2.3. Principle of Orthogonality. the cost function for the optimization of the MMSE estimator, J, is chosen to minimize the mean square error, and with reference to 1.7, is given by

$$J = E[|e(n)^2|] \tag{1.8}$$

where E denotes the expectation operator, and $||$ refers to the magnitude.

In order to minimize this cost function, a gradient operator ∇ is defined on the basis of the coefficients of the filter. Hence, for the ith filter coefficient, h_i,

$$\nabla_i = \frac{\partial}{\partial h_i} \tag{1.9}$$

Applying the gradient vector ∇ to the cost function J, the gradient vector $\nabla.J$ is obtained, and is given by

$$\nabla_i J = \frac{\partial J}{\partial h_i} \tag{1.10}$$

To minimize the cost function, the elements of the vector in 1.10 must all be zero at once, ie

$$\nabla_i J = 0 \epsilon i = 0, 1, 2, 3... \tag{1.11}$$

Hence, to minimize the cost function J, with regards to 1.7, simplifying the partial derivatives results in the following equation solution [3] at any given time instant n:

$$\nabla_i J = -2E[x(n-i)e^*(n)] \tag{1.12}$$

which when equated to 0, specifies the operating conditions required for the minimization of the cost function J as

$$E[x(n-i)e^*(n)] = 0 \epsilon i = 0, 1, 2, 3... \tag{1.13}$$

This is known as the principle of orthogonality, since two signals are known as orthogonal signals when the correlation between them is zero, and here the condition for optimal linear filtering is that the estimated error is orthogonal to all L input samples that are involved in the estimation of the echo path at any given instant of time n.

1.2.4. Assumptions and Limitations. Before looking into the different MMSE estimators, it is important to note some of the assumptions and limitations being taken into account during the development of the Kalman filter.

1. The measurement noise or background noise is assumed to be zero mean, white gaussian noise (except in the case of double-talk scenario, which will be discussed later on in this thesis).

2. The study of estimators is limited to discrete time, linear, causal finite impulse response response (FIR) estimators.

3. The echo paths that are being estimated by the MMSE estimators in this thesis are wide sense stationary(WSS) as well as slowly changing non stationary echo paths.

1.2.5. Wiener Filter. Expanding on the equation for the principle of orthogonality, 1.13, with $R_{ex}(n) = E[x(n-i)e^*(n)]$ as the cross correlation between the error signal and the input signal at any given time instant n and for a given filter coefficient m, the following can be deduced with the help of equation 1.7.

Hence, the deduced equation at any given instant of n for the cross correlation now looks as shown below:

$R_{ex}(n) = E[x(n-i)e^*(n)] = E[x(n-i)(\hat{y}(n) - y(n)]$

Simplifying the above solution results in the expression shown in 1.14 given below:

$$R_{\hat{y}(n)x}(n) = R_{yx}(n) \tag{1.14}$$

where $R_{y(\hat{n})x}(n)$ is the correlation between the input signal and the estimated echo signal.

From observing equation 1.14 and noting that the estimated output $\hat{y}(n)$ is calculated by passing the input signal $x(n)$ through the estimated echo path impulse response $\hat{h}(n)$, the following relationship can be obtained for a particular filter coefficient at any given instant of time n [4] :

$\hat{h}(n) * R_{xx}(n) = R_{\hat{y}(n)x}(n)$

which for all L coefficients of $\hat{h}$ becomes

$$\sum_{i=1}^{L} \hat{h}_i(n) * R_{xx}(n) = R_{yx}(n) \tag{1.15}$$

Solving for $\hat{h}(n)$, equation 1.15 then becomes

$$\sum_{i=1}^{L} \hat{h}_i(n) = \sum_{i=1}^{L} R_{xx}(n-i)^{-1} R_{yx}(n) \tag{1.16}$$

Hence, it is seen from equation 1.16 that the estimated echo path $\hat{h}$ at any given instance of time n depends on the cumulative correlation matrices R_{xx} and R_{yx} until that instance of time n.

1.2.6. The LMS Algorithm. The Wiener filter, as seen in the previous subsection, gives an estimate of the echo path at any given instance of time n, not as a recursively updating estimate value but a single 'true' estimate based on the

ensemble values of the correlation matrices of the input signal and the estimated output up until the present instance of time n.

At any given instant of time n, therefore,the equation 1.16 can then be rewritten as an equation of vectors, as shown in 1.17.

$$\hat{\mathbf{h}}(n) = \mathbf{R_{xx}}^{-1}.\mathbf{R_{yx}}. \tag{1.17}$$

The LMS algorithm (and any recursive adaptive filter for that matter) starts with an initial guess of the echo path estimate, usually with the initial value $\hat{\mathbf{h}}(n) = 0$, and then obtaining better approximations by taking steps in the direction of the negative gradient of the mean square error [1]. This means that the LMS algorithm uses the method of steepest descent to iteratively optimize the cost function,ie, to minimize the mean square error. To make this an adaptive procedure, the ensemble values of $\mathbf{R_{xx}}$ and $\mathbf{R_{yx}}$ are replaced with their instantaneous estimates, given by $\mathbf{x}(n).\mathbf{x}^H(n)$ and $\mathbf{x}(n).\mathbf{y}^*(n)$ respectively.

The illustration of the steepest descent algorithm is as given in Figure 1.3.

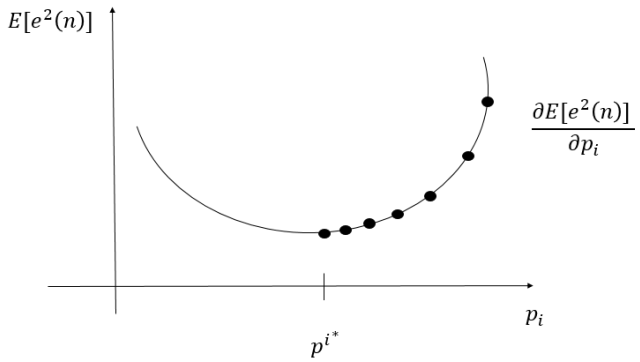

Figure 1.3. Steepest descent illustration

Hence, using the concept of the steepest descent to the cost function J from 1.10,The update equation of the echo path estimate simplifies to 1.18.

$$\hat{\mathbf{h}}_{n+1} = \hat{\mathbf{h}}_n - \mu \frac{\partial J}{\partial \mathbf{h_n}} \tag{1.18}$$

Substituting from 1.12 into 1.18, the equation for the iterative echo path estimate is obtained as shown in eq 1.19 as:

$$\hat{\mathbf{h}}_{n+1} = \hat{\mathbf{h}}_n + 2\mu \mathbf{x}(n) e^*(n) \tag{1.19}$$

where μ is the adaptation rate or adaptive step size.

1.2.7. NLMS Adaption Step Size Considerations. The step size μ generally can take values in the range $0 < \mu < 2/max(\mathbf{x}(n).\mathbf{x}^H(n))$. This means that the correlation of the input signal x plays an important role in the performance of the LMS algorithm. The step size μ is chosen, taking into account the following:

1. A smaller step size decreases the convergence speed of the adaptive filter, but ensures lower values of residual steady state error.

2. A larger step size will reduce the time taken by the adaptive filter to converge, but too large a step size can cause the adaptive filter to become unstable and result in large residual steady state error.

It is also to be noted that μ is generally a constant, for all the L coefficients of the echo path estimate $\hat{\mathbf{h}}$, and in general, for all the iterative updates that the filter performs to obtain the optimal soultion of the echo path estimate, μ is a constant unless the adaptive filter is designed to be a variable step size adaptive filter.

1.2.8. Normalized LMS. As seen in the previous subsection, the performance of the regular LMS algorithm is highly sensitive to the power of the input signal. This can be avoided by normalizing the correlation matrix of the input signal and the error signal, with respect to the energy of the input signal. Thus, when normalized, the equation of the normalized LMS (NLMS) algorithm becomes:

$$\hat{\mathbf{h}}_{n+1} = \hat{\mathbf{h}}_n + \frac{\tilde{\mu}}{\| \mathbf{x}(n) \|)^2}\mathbf{x}(n)e^*(n) \tag{1.20}$$

where || denotes the L2 norm.

In order to avoid division by zero, the above equation 1.20 is altered slightly to become

$$\hat{\mathbf{h}}_{n+1} = \hat{\mathbf{h}}_n + \frac{\tilde{\mu}}{\delta + \| \mathbf{x}(n))^2 \|)}\mathbf{x}(n)e^*(n) \tag{1.21}$$

where δ is a small positive constant. It is also important to note that the new adaptive step size $\tilde{\mu}$ is now independent of the input signal eigen values, and hence, has a range between $0 < \tilde{\mu} < 2$.

2. TIME DOMAIN KALMAN FILTER

2.1. NEED FOR KALMAN FILTERING IN ECHO CANCELLATION

The MMSE estimators discussed in the previous sections perform well in the scenario of stationary problems, which means that for the previously described estimators, the echo path is modelled as a stationary process, whose mean and variance do not change with time. Unfortunately, an echo path in reality, does tend to change over time, due to various factors such as the movement of the speaker, temperature changes,etc.

Since these echo paths, usually modelled as the coefficients of an FIR filter, are dependent on various factors of the acoustic environment, are better modelled as slowly varying non stationary processes. The Kalman filter, which can track trajectories of moving objects such as missiles etc [5], is more suited for a non stationary process such as an acoustic echo path. The Kalman filter, when derived with the appropriate simplifications and approximations,also offers good convergence features and moderate computational complexity. [6]

2.2. STATE VARIABLE MODEL

This section outlines the state variable equations for the general Kalman filter (GKF), as described in [6].

For the general Kalman filter, if the P most recent time samples of the microphone signal are considered, then the far end signal is a matrix of size LXP, given by

$$\mathbf{X}(n) = [\mathbf{x}_n \ \mathbf{x}_{n-1} \ \mathbf{x}_{n-2} \ \mathbf{x}_{n-3} ... \ \mathbf{x}_{n-P+1}]^T \tag{2.1}$$

However, to reduce the calculational complexity, P is assumed to be 1, and hence, 2.1 becomes equivalent to equation 1.2. The equations for the microphone signal, near end noise signal and the error signal remain the same as discussed in section 1.2.1.

As mentioned earlier, the echo path is modelled as a slowly varying process, following a first order Markov model. A first order Markov model is a model of a probabilistic process that, at any given instance of time, depends on its penultimate time instance value. Putting this in terms of the current time instance n, the echo path equation is then modelled as

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \mathbf{w}(n) \tag{2.2}$$

where $\mathbf{w}(n)$ is, for all practical purposes, white zero mean gaussian noise, called the process noise. This process noise, with variance $\sigma_w^2(n)$, is important in capturing the uncertainties of $\mathbf{h}(n)$ [6], and will be discussed in future chapters.

2.3. GENERAL KALMAN FILTER DESIGN EQUATIONS

This section derives the equations of the general Kalman filter, as described in [6]. Using all the same signal notations defined in section 1.2.1,the optimum estimate of the echo path, $\hat{\mathbf{h}}(n)$ using the Bayesian approach is given as follows [1]:

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mathbf{K}(n).e(n) \tag{2.3}$$

where $\mathbf{K}(n)$ is the update factor known as the Kalman gain matrix, and $e(n)$ is the apriori error signal, given by

$$e(n) = d(n) - \mathbf{x}^T(n).\hat{\mathbf{h}}(n-1) \tag{2.4}$$

It is to be noted that the error $e(n)$ in equation 1.7 is called the aposteriori error-the apriori error at any instant of time n is calculated using the estimate of the previous time instant $n-1$.

The state estimation error or aposteriori misalignment is given by

$$\mu(n) = (h)(n) - (\hat{h})(n) \tag{2.5}$$

The correlation of the state estimation error $\mathbf{R}_\mu$ at any given instant of time n is given by

$$\mathbf{R}_\mu(n) = E[\mu(n).\mu^T(n)] \tag{2.6}$$

The state estimation error ie the difference between the actual echo path and the estimated echo path is required to derive the Kalman gain.The apriori state estimation error is hence given by

$$\mathbf{m}(n) = (h)(n) - (\hat{h})(n-1) \tag{2.7}$$

and the correlation of the apriori state estimation error $\mathbf{m}(n)$ is given by

$$\mathbf{R}_\mathbf{m}(n) = E[\mathbf{m}(n).\mathbf{m}^T(n)] \tag{2.8}$$

The relation between the apriori misalignment correlation at any given instant of time n with the aposteriori misalignment correlation at the previous time instant $n-1$ is given by

$$\mathbf{R_m}(n) = \mathbf{R}_\mu(n-1) + \sigma_w^2(n) \tag{2.9}$$

The solution of the Kalman gain matrix involves minimizing the below cost function $J(n)$ with respect to $\mathbf{K}(n)$, which is nothing but the MMSE optimal estimator criterion.

$$J(n) = \frac{1}{L}.tr(\mathbf{R}_\mu(n)) \tag{2.10}$$

where $.tr$ is the trace. Hence, the Kalman filter gain matrix is found to be

$$\mathbf{K}(n) = \mathbf{R_m}(n).\mathbf{x}(n)[\mathbf{x}^T(n).\mathbf{R_m}(n).\mathbf{x}(n) + \sigma_v^2(n)]^{-1} \tag{2.11}$$

However, the correlation of the apriori error signal defined in equation 2.4 at any instant of time n, can be expressed in terms of the correlation of the apriori misalignment defined in equation 2.9 at the previous time instant $n-1$, as

$$\mathbf{R_e}(n) = \mathbf{x}^T(n).\mathbf{R_m}(n).\mathbf{x}(n) + \sigma_v^2(n) \tag{2.12}$$

Hence, substituting equation 2.12 into the equation for Kalman gain matrix 2.11 yields

$$\mathbf{K}(n) = \mathbf{R_m}(n).\mathbf{x}(n).{\mathbf{R_e}}^{-1} \tag{2.13}$$

After calculating the Kalman gain for the current iteration,the aposteriori misalignment is then updated as

$$\mathbf{R}_{\mu}(n) = [I_L - \mathbf{K}(n).\mathbf{x}^T(n)].\mathbf{R_m}(n) \tag{2.14}$$

2.4. SUMMARY OF EQUATIONS

The following steps, followed in order, summarize the operation of the Kalman adaptive filter algorithm for echo cancellation, going through the iterative process of prediction and update.

1. for the time instant $n = 0$, initialise $(\hat{h})(0) = 0$, $\mathbf{R}_{\mu}(0) = 0$ and $\mathbf{R_m}(0) = \Delta.I_L$, where $\Delta = .00001$ is a small normalizing factor, and I_L is an identity matrix of size LXL where L is the length of the echo path.

2. Update $\mathbf{R_m}$ as per the below equation:

$$\mathbf{R_m}(n) = \mathbf{R_m}(n-1) + \sigma_w I_L \tag{2.15}$$

   where σ_w is the process noise variance.

3. for the next instant of time n, calculate the correlation of the apriori error signal, $\mathbf{R_e}(n)$, given by equation 2.12.

$$\mathbf{R_e}(n) = \mathbf{x}^T(n).\mathbf{R_m}(n).\mathbf{x}(n) + \sigma_v^2(n) \tag{2.16}$$

4. The residual error after passing the far end signal through the echo path estimate is then calculated as

$$e(n) = \mathbf{d}(n) - \mathbf{x}^T(n).(\hat{h})(n-1) \tag{2.17}$$

5. Using the value of $\mathbf{R_e}(n)$ calculated in the first step, the Kalman gain matrix is calculated as

$$\mathbf{K}(n) = \mathbf{R_m}(n).\mathbf{x}(n).{\mathbf{R_e}}^{-1} \tag{2.18}$$

6. The estimate of the echo path is then updated as follows:

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mathbf{K}(n).e(n) \tag{2.19}$$

7. The aposteriori error correlation is then updated for use in the next iteration $n+1$ as

$$\mathbf{R}_\mu(n) = [I_L - \mathbf{K}(n).\mathbf{x}^T(n)].\mathbf{R_m}(n) \tag{2.20}$$

8. Go back to step 1 and repeat the prediction update iteration process for the next time instant $n+1$ with the updated values of $\hat{\mathbf{h}}$ and $\mathbf{R}_\mu$ calculated in the previous steps as the initial values for the following iterations.

3. THE THEORY OF SUBBANDING

As discussed earlier in previous chapters, the features of an acoustic echo path in a dynamically changing environment are not so easy to model- many challenges involved include the fact that sometimes, the echo paths are of really large lengths(typically of length $125ms$ or more). Also, as mentioned earlier, the echo path is not a stationary statistic process, but has often and sometimes drastic changes in its coefficients.

These long and fast changing echo paths make the traditional echo cancellation algorithms such as the NLMS algorithm ineffective in a dynamically changing acoustic echo path environment. The Kalman filter is an upgrade to the NLMS, and is much more capable at handling the issues caused by long and fast changing echo paths. The concept of subbanding, [7] further helps ameliorate the problem of a non stationary acoustic echo path scenario, and this chapter will discuss the concept of frequency subbanding for the purpose of echo cancellation. The subband acoustic echo canceller shown in figure Figure 3.1 is based on the subband approach proposed by Kellermann in [8].

The subband echo canceller shown in figure Figure 3.1 involves splitting the input and desired signals into several smaller signals, and then applying echo cancellation techniques on each of the smaller signals. The input and desired signals are first filtered into subbands, using an analysis system. The echo cancellation is then performed on each of the subbanded signals individually, after which the individual residual errors of each subband are then reconstructed into a single residual error signal using the synthesis system. The entire sequence of events in the process is referred to as the analysis-synthesis system (ASS) [7].

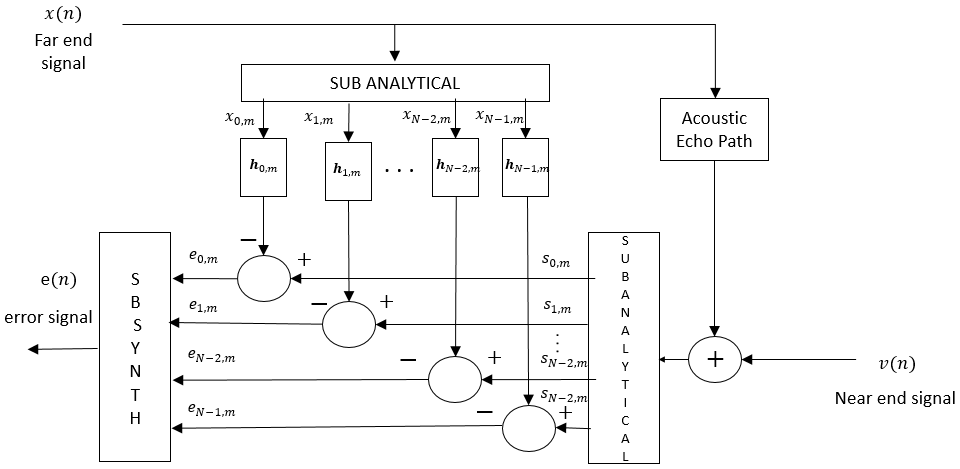

Figure 3.1. Illustration of subbanding operation

The analysis or subbanding system consists of a bank of filters, which are modulated versions of a low pass prototype filter [9]. The input and desired signals are subsampled or downsampled and filtered using these analysis system filter banks, after which the echo cancellation would take place. The filter banks work by dividing the frequency spectrum of the signal into K subbands between 0 and 2π radians, with a subsampling factor of R. Since the frequency bands above π radians are complex conjugates of those below π radians, only $K/2$ frequency bands need adaptive filters, since the remaining bands can be calculated by taking the respective complex conjugates. K and R are selected based on a number of criteria. Increasing the value of K will increase the overall delay, and downsampling will lead to aliasing, which is why the value of the downsampling factor R chosen is also important. R is preferred to be as large as possible, since the higher the downsampling, lower is the complexity of the subband processing (echo cancellation), and downsampling with subbands of bandwidth $2\pi/R$ reduces the spectral dynamics of the input speech signal, making

the eigen spread of the input signal similar to that of white noise [8], which helps improve convergence of the subband adaptive filters. The synthesis or reconstruction system involves recombination of the residual error signal for each subband into a single whole residual signal, by a summation of an interpolated and filtered version of each of the subband residual error signals. The working of the analysis and synthesis filterbank will be discussed in more detail in the following chapters.

The primary advantage of subbanding is that the operation of each of the adaptive filters in each subband are independent of each other. Hence, subbands which have low energy signal components converge at approximately the same rate as those with high energy signal components. Hence, the overall convergence speed increases. Another factor that helps increase convergence speed is that since the number of taps of the adaptive filters to be estimated in the subbands is reduced by a factor of R, the time taken to converge also reduces. However, the effect of this factor is negated by the fact that the adaptive filters are now updated only $1/R$ times as often, reducing the overall speed of convergence.The other main advantage of the subbanding is reduced computational complexity, since the number of taps to be estimated reduces from quite a large number (1024 or greater) to much smaller number of taps depending on the number of subbands chosen. Another secondary advantage of subbanding is that acoustic echo paths of rooms will have different lengths for different subbands, and hence, the length of the adaptive filters used to estimate the subbands can be varied to match the expected echo path of that subband, further leading to computational efficiency.

As mentioned earlier, the major disadvantage of subbanding is the additional delay caused by the filtering operations performed by the analysis filters and the synthesis filters, which add some overhead to the delay due to the process of adaptive filtering itself. The amount of delay is propotional to the number of subbands. Another disadvantage of subbanding is that the stopbands of different subbands can alias

and leak into other subbands,interfering with the convergence of that subband, by reducing the attenuation of the stop band in question, which introduces aliased signals as well as noise into the subband and hinders convergence of the error. Analytical filter design is required to overcome this problem. Another important but overlooked issue with subbanding is that of the introduction of leading taps. As mentioned in [8], downsampling and low pass filtering to subbands can cause the adaptive filters in the subbands to become non-causal, and in order to avoid this, inducing delays into the echo path may become necessary, which further add to the overhead in the delay caused by the subbanding system. However, the acoustic echo path usually gains this delay due to the distance between the far end loudspeaker and the near end microphone. Hence, these effects can be overlooked if there is enough distance and hence delay between the far end and near end.

4. SUBBAND ANALYSIS AND SYNTHESIS FILTERS

This chapter of the thesis looks at the equations and working of the filters used in the process of subbanding, the analysis filters and the synthesis filters.

4.1. ANALYSIS FILTERBANK

Following the equations in [9], the signal in a subband m, denoted as $x_m(k)$, is calculated by filtering and downsampling of the full length signal $x(n)$, given by

$$x_m(k) = \sum_{l=0}^{K-1} f_m(l).x(rk-l) \tag{4.1}$$

with the subband filter for the m^{th} subband denoted by $f_m(n)$ of length K and the downfactoring or subsampling factor is R. These filters $f_m(n)$ are modulated versions of the analytically designed prototype filter $f(n)$. The vector representation of the subband filters can be expressed as $\mathbf{f}_m^T = \mathbf{w}_m^T.\mathbf{F}$, where

$$\mathbf{f}_m = [f_m(0)\ f_m(1)\ f_m(2)\ ...\ f_m(K-1)]^T \tag{4.2}$$

$$\mathbf{w}_m = [1\ e^{\frac{j.2\pi.m}{M}(1)}\ e^{\frac{j.2\pi.m}{M}(2)}\ ...\ e^{\frac{j.2\pi.m}{M}(M-1)}]^T \tag{4.3}$$

$$\mathbf{F} = [diag(\tilde{\mathbf{f}}_0)\ diag(\tilde{\mathbf{f}}_M)\ ...\ diag(\tilde{\mathbf{f}}_{K-M})] \tag{4.4}$$

where *diag* denotes a diagonal matrix. The representation of the diagonal matrix is given as per 4.5 below.

$$diag(\tilde{\mathbf{f}}_i) = \begin{bmatrix} f(i) & ... & 0 \\ : & ... & 0 \\ : & ... & f(i+M-1) \end{bmatrix} \tag{4.5}$$

The signal can then be subsampled by a factor of R, and only one sample every R sample periods would be required to obtain the subband signals [10]. This R number of samples constitute what is called a frame [9]. The fullband input vector signal for a frame k is defined as

$$\mathbf{x}(k) = [x(rk)\ x(rk-1)\ ...\ x(rk_K+1)]^T \tag{4.6}$$

Now, the set of modulation vectors for the M subbands can be cumulatively represented by a modulation matrix, $\mathbf{W}$, given by

$$\mathbf{W} = [\mathbf{w}_0\ \mathbf{w}_1 ...\ \mathbf{w}_{M-1}]^T \tag{4.7}$$

The M subband samples for a given frame k can be calculated as

$$[x(rk)\ x(rk-1)\ ...\ x(rk_K+1)]^T = \mathbf{WFx}(k) \tag{4.8}$$

From 4.5, it can be seen that F is a sparse matrix, hence $\mathbf{Fx}(k)$ from equation 4.8 can be calculated using K real multiplications. Also, with respect to the definition of the inverse discrete Fourier transform (IDFT), $\mathbf{W}/M$ is the inverse DFT matrix. Hence, the computationally efficient way of calculating the subband samples at frame number k according to 4.8 is to first use K multiplications to calculate $\mathbf{Fx}(k)$, then taking the inverse discrete fourier transform of $\mathbf{Fx}(k)$, and dividing by M.

The concept of the subbanding can be better understood by looking at what goes on in any one subband m, depicted in figure Figure 4.1.

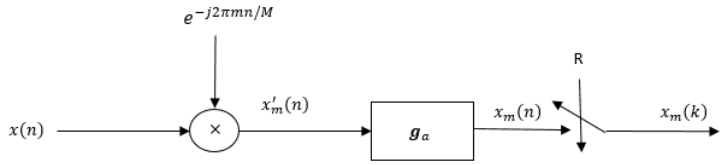

Figure 4.1. Block diagram of mth subband analysis filter

As seen in Figure 4.1, the input signal $x(n)$ is modulated in the mth subband by multiplying it by $e^{-j2\pi mn/M}$, after which the modulated signal is low pass filtered using the subband filter $\mathbf{g}_a$, to produce the subband signal $x_m(n)$, which is then downsampled by a factor of R to give the downsampled signal $x_m(k)$ at frame k.

4.2. SYNTHESIS FILTERBANK

The synthesis filterbank is used to reconstruct the error signals from the various subbands to obtain the cumulative residual error signal. The general structure of the synthesis filterbank is as shown in figure Figure 4.2. The signals being reconstructed are the error signals in each of the subbands obtained after the echo cancellation operation is performed in each of the subbands. As seen in figure Figure 4.2, $e_m(k)$ is upsampled by a factor of R, by the insertion of $R-1$ zeros between each sample of $e_m(k)$, resulting in $\dot{e}_m(n)$. This creates duplicate images of $e_m(k)$ in the frequency domain, which is why the signal $\dot{e}_m(n)$ is then low pass filtered using the reconstruction filter $\mathbf{g}_(s)$ to eliminate these images. Then the modulation of $\dot{e}_m(n)$

by multiplying by $f_m(n)$ to shift it back up by m subbands, to obtain $e'_m(n)$. $f_m(n)$ is given by

$$f_m(n) = e^{j2\pi mn/M} \tag{4.9}$$

Then, all the processed subband signals at that instant of time n are summed together to obtain the reconstructed signal as

$$e_n = \sum_{m=0}^{M-1} e'_m(n) \tag{4.10}$$

The filtering operation can be performed using a form II FIR structure, however, for reducing the amount of delay overhead involved in the subbands, a transpose form II FIR structure is chosen, as shown in figure Figure 4.3.

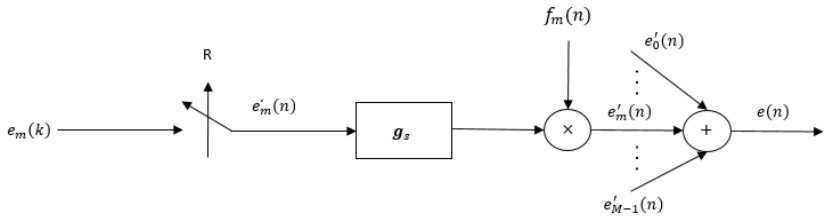

Figure 4.2. Block diagram of mth subband signal's synthesis filter

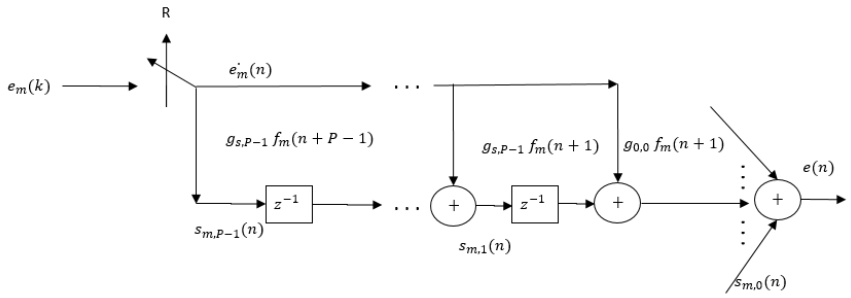

Figure 4.3. Transpose form II FIR structure

The equations of the above configuration are better explained using state vectors, as explained in [9]. The state vector of a subband m,for the above configuration, for a given frame k, $\mathbf{s}_m k$ which is L values long, (L being the length of the reconstruction filter, can be defined for the sake of convenience into two vectors, an upper vector of length R and a lower vector of length $L - r$, written as

$$\mathbf{s}_m(k) = \begin{bmatrix} \mathbf{s}_m^u(k) \\ \mathbf{s}_m^l(k) \end{bmatrix} \tag{4.11}$$

where the upper state vector $\mathbf{s}_m^u(k)$ is given by

$$\mathbf{s}_m^u(k) = [s_{m,0}(kr) \ ... \ s_{m,r-1}(kr)]^T \tag{4.12}$$

and the lower state vector $\mathbf{s}_m^l(k)$ is given by

$$\mathbf{s}_m^l(k) = [s_{m,r}(kr) \ ... \ s_{m,K-1}(kr)]^T \tag{4.13}$$

Now, a frame k consists of R samples, hence, in the fullband, R samples can be calculated at a time, by summing the upper state variables of the state vector, which contain the most recently calculated values. These R values calculated are given by

$$[e(kr-r+1)\ e(kr-r+2)\ ...\ e(kr)]^T = \sum_{m=0}^{M-1}\sum_{u=0}^{r-1}\mathbf{s}_{m,r}^{u}(k) \tag{4.14}$$

Now, due to interpolation, every subband error sample will be followed by $R-1$ zeros, and for these $R-1$ time samples, the state vector will only undergo a time shift, hence, $R-1$ time shifts will occur before the next R fresh samples of the fullband signal are calculated. Similar to the analysis filters, the modulated versions of the synthesis filter, $g(n)$, are given by the filter vector (for a particular subband m) as

$$\mathbf{g}_m = [g_m(0)\ ...g_m(K-1)]^T = \mathbf{G}\mathbf{w}_m \tag{4.15}$$

where G is a sparse matrix defined as

$$\mathbf{G} = [diag(\tilde{\mathbf{g}}_0)\ diag(\tilde{\mathbf{g}}_M)\ ...\ diag(\tilde{\mathbf{g}}_{K-M})] \tag{4.16}$$

and the *diag* function is defined in equation 4.5. As mentioned earlier, the state vector is shifted by $r-1$ positions (one time delay or time shift per time sample) and then added to the modulated input subband sample. The whole process can be represented by the following equation:

$$\mathbf{s}_m(k) = \begin{bmatrix} \mathbf{s}_l^m(k-1) \\ 0_r \end{bmatrix} + \mathbf{G}\mathbf{w}_m e_m(k) \tag{4.17}$$

The sum of state vectors, or vector of all subband state vectors, can similarly be defined as

$$\mathbf{s}(k) = \begin{bmatrix} \mathbf{s}^l(k-1) \\ \\ 0_r \end{bmatrix} + \mathbf{GW}\mathbf{e}^{sub}(k) \tag{4.18}$$

where $\mathbf{W}$ is defined in equation 4.7 and $\mathbf{e}^{sub}(k)$ at any given frame k is given by

$$\mathbf{e}^{sub}(k) = [e_0(k)\ e_1(k)\ ...\ e_{M-1}(k)]^T \tag{4.19}$$

The above equation, eq 4.19, represents the system in Figure 4.3. $\mathbf{e}^{sub}(k)$ will only be non zero every r samples, hence for the remaining $R-1$ samples, there is only time shifting of the state vector $\mathbf{s}(k)$. At the end of every frame k, the first R samples of the state vector $\mathbf{s}(k)$ are obtained as the newly calculated $n-R$ to n samples, where n is the time instant for the full band vector. It is to be noted that the prototype analytical filter and synthetic filter were designed as a minimization problem solved using quadrative programming as described in [9].

5. SUBBAND ADAPTIVE FILTERS

This section of the thesis looks at the design equations of the subbanded adaptive filters, as well as a few special scenarios and practical conditions involved during the process of acoustic echo cancellation.

5.1. SUBBAND NLMS DESIGN EQUATIONS

The equations for the subband NLMS remain similar to those of the fullband NLMS, except that now the NLMS algorithm is applied individually to each of the M subbands. Hence, the equations for the subband NLMS are now expressed in terms of subbanded signals, for any particular subband m. The update equation for $\hat{\mathbf{h}}_{m,n+1}$ hence becomes

$$\hat{\mathbf{h}}_{m,n+1} = \hat{\mathbf{h}}_{m,n} + \frac{\tilde{\mu}}{\delta + \parallel \mathbf{x}_m(n))^2 \parallel)} \mathbf{x}_m(n) e_m^*(n) \tag{5.1}$$

After the estimation of the subband echo paths is performed and the echo cancellation takes place, the residual errors in each of the subbands are then fed back into the synthesis filterbank as discussed in the previous sections to obtain the fullband residual error, which can be used to study the performance of the subband echo canceller.

5.2. SUBBAND KALMAN FILTER DESIGN EQUATIONS

Similar to the NLMS algorithm, the equations for the subband Kalman filter are designed and expressed for each subband, m, and can be summarized in the following equations:

1. Initialise $(\hat{h})_m(0) = 0$ and $\mathbf{R}_{\mu_m}(0) = 0$.

2. Update $\mathbf{R_{m_m}}$ as per the below equation:

$$\mathbf{R_{m_m}}(n) = \mathbf{R_{m_m}}(n-1) + \sigma_w I_l \tag{5.2}$$

   where σ_w is the process noise variance.

3. $\mathbf{R_{e_m}}(n)$ is then calculated as

$$\mathbf{R_{e_m}}(n) = \mathbf{x}_m^T(n).\mathbf{R_{m_m}}(n).\mathbf{x}_m(n) + \sigma_{v_m}^2(n) \tag{5.3}$$

4. The residual error in each subband is then calculated as

$$e_m(n) = \mathbf{d}_m(n) - \mathbf{x}_m^T(n).(\hat{h}_m)(n-1) \tag{5.4}$$

5. Now the Kalman gain matrix is calculated as

$$\mathbf{K_m}(n) = \mathbf{R_{m_m}}(n).\mathbf{x_m}(n).\mathbf{R_{e_m}}^{-1} \tag{5.5}$$

6. The estimate of the echo path is then updated as

$$\hat{\mathbf{h}}_\mathbf{m}(n) = \hat{\mathbf{h}}_\mathbf{m}(n-1) + \mathbf{K}_m(n).e_m(n) \tag{5.6}$$

7. The aposteriori error correlation is then updated for use in the next iteration $n+1$ as

$$\mathbf{R_{m_m}}(n) = [I_l - \mathbf{K_m}(n).\mathbf{x}_m^T(n)].\mathbf{R_{m_m}}(n) \tag{5.7}$$

8. step 1 is then repeated to continue the prediction update process with the calculations of previous iterations as updated values for the variables.

It is to be noted that in the subbands, the adaptive echo cancellation algorithms work on complex valued numbers.

6. NEAR END ENERGY ESTIMATION FOR NOISE POWER

An adaptive filter's estimation of the echo path coefficients depends on a large extent, on the error between the echo path estimate and the actual echo path. This error is estimated accurately by the original design equations assuming the measurement noise $v(n)$ is zero, or whose power is assumed to be known.With reference to Figure 1.2, the measurement noise $v(n)$ is the near end noise that gets added to the echo signal $y(n)$. In case of usual telephonic scenarios, where noise power is an unknown changing variable, it becomes necessary to track the noise power and estimate it as accurately as possible, in order to avoid the coefficients of the adaptive filter diverging and losing track of the true echo path coefficients. There is also the scenario of double talk to be considered, where the near end speaker also speaks, and his signal gets added to the echo signal along with background measurement noise. The equation for the near end noise signal $v(n)$ then becomes

$$v(n) = w(n) + n_x(n) \tag{6.1}$$

where $w(n)$ is the background noise with variance σ_w^2 and $n_x(n)$ is the near end speaker's speech signal, with variance σ_n^2.

Hence, slight modifications are necessary to the Kalman algorithm, to account for noise tracking. This thesis uses the cross correlation between the far end signal $\mathbf{x}(n)$ and the residual error $e(n)$ to estimate the measurement noise variance, as per [2]. From [2],The cross correlation $\mathbf{r}_{e\mathbf{x}}(n)$ between the far end and error signals is given by

$$\mathbf{r}_{e\mathbf{x}} = E[\mathbf{x}(n)e(n)] = R_{\mathbf{xx}}\delta\mathbf{h}(n-1) \tag{6.2}$$

where E is the expectation operation, $R_{\mathbf{xx}} = E[\mathbf{x}\mathbf{x}^{\mathbf{T}}]$ and $\delta\mathbf{h}(n-1) = \mathbf{h} - \hat{\mathbf{h}}(n-1)$.The variance of the error $e(n)$ in terms of the E operation is given as

$$\sigma_e^2(n) = E[e^2(n)] = \delta\mathbf{h}^T(n-1)\mathbf{r}_{e\mathbf{x}} + \sigma_w^2(n) + \sigma_n^2(n) \tag{6.3}$$

Now,approximating $R_{\mathbf{xx}}$ using only the nth latest far end signal sample $x(n)$ (also called the excitation signal) rather than the entire vector $\mathbf{x}(n)$ with variance σ_x^2, and calling the near end energy estimate as $\nu(n) = \sigma_v^2 + \sigma_n^2(n)$, from equation 6.3, the following equation for the near end energy estimate can be written as:

$$\nu(n) \approx \sigma_e^2(n) - \frac{\mathbf{r}_{e\mathbf{x}}(n)^T\mathbf{r}_{e\mathbf{x}}(n)}{\sigma_x^2} \tag{6.4}$$

The values of the variables used in eq 6.4 are exact and not readily available, hence the estimates of the above variables are used instead.

$$\hat{\nu}(n) \approx \hat{\sigma}_e^2(n) - \frac{\hat{\mathbf{r}}_{e\mathbf{x}}(n)^T\hat{\mathbf{r}}_{e\mathbf{x}}(n)}{\hat{\sigma_x}^2} \tag{6.5}$$

The estimates are recursively calculated every iteration as

$$\begin{aligned} \hat{\mathbf{r}}_{e\mathbf{x}}(n) &= \lambda\hat{\mathbf{r}}_{e\mathbf{x}}(n-1) + (1-\lambda)\mathbf{x}(n)e(n) \\ \hat{\sigma_x}^2(n) &= \lambda\hat{\sigma_x}^2(n-1) + (1-\lambda)x^2(n) \\ \hat{\sigma}_e^2(n) &= \lambda\hat{\sigma}_e^2(n-1) + (1-\lambda)e^2(n) \end{aligned} \tag{6.6}$$

In algorithms such as the NLMS algorithm, double talk is usually detected using the Geigel detection algorithm [10], and whenever the criterion for double talk is met, the adaptation of the adaptive filter coefficients is halted so as to prevent further divergence of the coeffcients of the adaptive filter from the true echo path. Other methods for adaptation during the detected double talk include a variable step size for the NLMS update equation, as studied in [2], which ultimately acts like a

Geigel detector, almost halting estimation when double talk is detected, and then varying the learning rate of the NLMS algorithm to ensure the coefficients of the adaptive filter catch up with the changes in the echo path coefficients. The main advantage of the Kalman filter over the NLMS filter, in either fullband or subband, is that double talk is not an issue provided the noise estimation is done accurately enough. This means that following the noise estimation described in eq 6.5, the Kalman filter can still keep actively tracking the true echo path coefficients without the need to halt echo path estimation. This will be shown in the results, which will be discussed in the following chapter.

7. SIMULATION RESULTS

This section details the results of the simulations of the NLMS and Kalman filter algorithms in the fullband and subbands, to evaluate the performance of the newly developed subband Kalman filter. All simulations were carried out in MATLAB. All results are the average of 5 iterations.

7.1. INPUT SPECIFICATIONS

The adaptive filters were tested using both speech and white gaussian noise as the far end speaker's signal $\mathbf{x}(n)$. The subband filters work on complex data, hence, even the fullband signals were tested with complex white gaussian noise, to see how they perform when fed with complex inputs. The speech input is as shown in figure Figure 7.1.

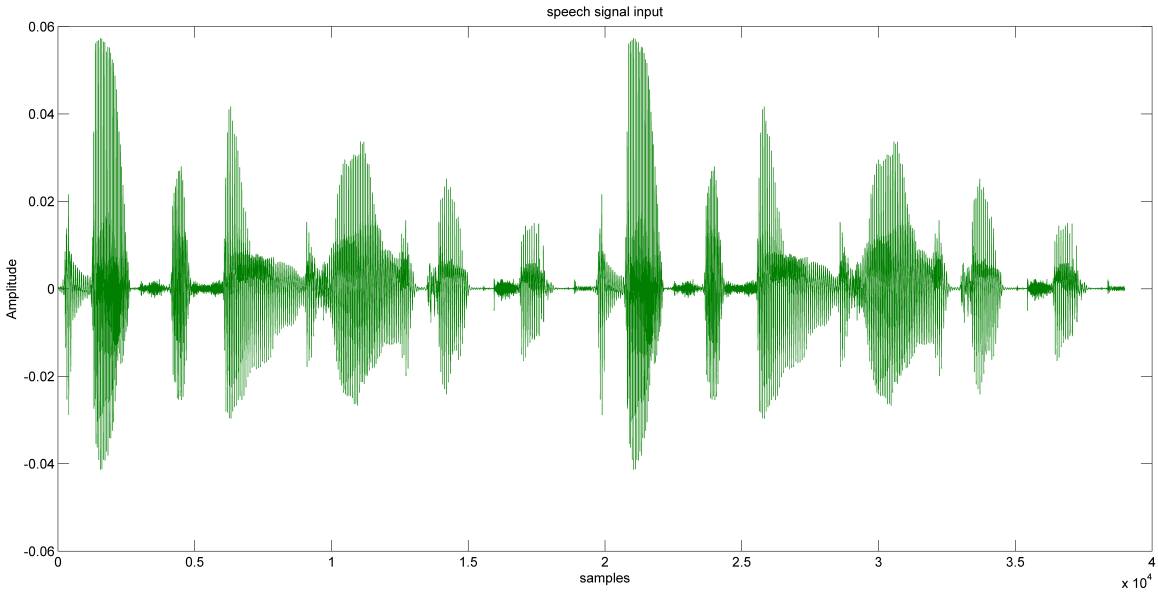

Figure 7.1. Speech input signal

7.2. ECHO PATH SPECIFICATIONS

The fullband and subband filters were tested with two 256 tap length echo paths, an impulse, and a random echo path, as shown in figures Figure 7.2 and Figure 7.3.

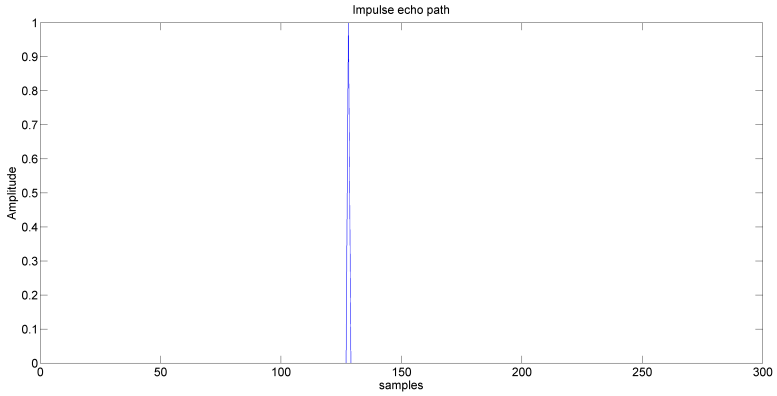

Figure 7.2. Impulse echo path

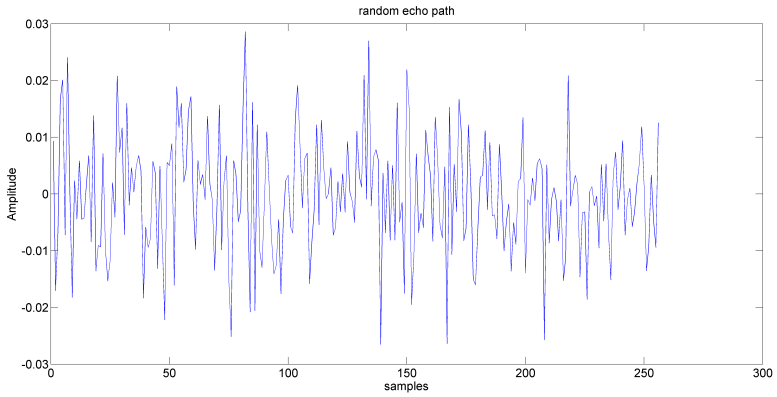

Figure 7.3. Random echo path

7.3. TEST SPECIFICATIONS

The fullband and subband filters were tested under the following conditions:

1. Constant measurement noise power: In this scenario,the measurement noise that gets added to the echo signal is assumed to have constant power, and the signal to noise ratio (SNR) is set at $30dB$.

2. Varying measurement noise power: In this scenario, the noise power is increased halfway through the simulation, to observe how the adaptive filters react to the change in noise power estimation. In these simulations, the SNR was changed from $30dB$ to $20dB$.

3. Varying echo path: To study the converging properties of the adaptive filters, the echo paths were changed midway through the simulations, to see how the adaptive filter coefficients handled the sudden change in the echo path coefficients. The SNR for these simulations remain at $30dB$.

4. Double talk: In this scenario, a near end speaker's signal is introduced halfway through the simulation, and the performance of the adaptive filters is studied to see the effects of double talk.

7.4. TEST CRITERIA SPECIFICATIONS

The following criteria were used to study the performance of the adaptive filters:

1. Residual error: The error signal, which is the difference between the echo signal and the estimated echo signal, can be studied to observe the rate of convergence of the adaptive filters and the steady state error of the filters.

2. Echo to return loss enhancement(ERLE): the echo to return loss enhancement is a measure of the amount of energy of the far end signal with respect to that of the energy signal. A lower error signal will result in a higher ERLE. The ERLE was calculated as an average over 1000 samples and expressed in dB, as expressed in the formula below:

$$ERLE(n) = 10\log(x_{avg}(n)/e_{avg}(n) + 1) \tag{7.1}$$

where x_{avg} and e_{avg} are the averaged values of the far end and error signals respectively.

7.5. CONSTANT SNR SCENARIO

The residual error versus the far end signal for the fullband and subband filters are as shown in Figure 7.4 and Figure 7.5.

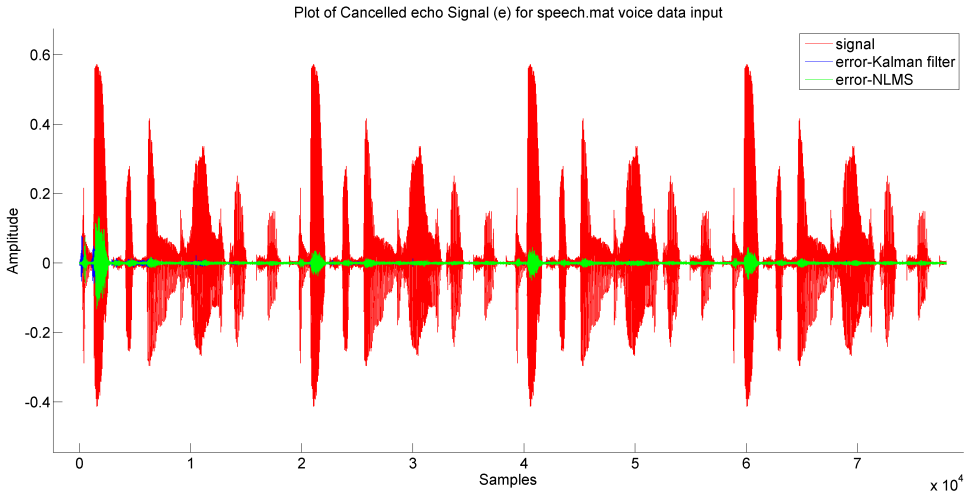

Figure 7.4. Residual error for fullband filters

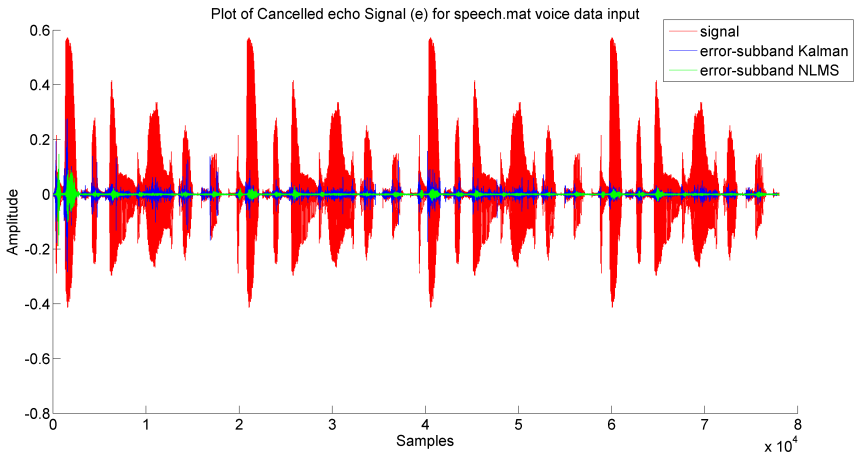

Figure 7.5. Residual error for subband filters

The averaged ERLE for the fullband and subband filters are as shown in Figure 7.6 and Figure 7.7.

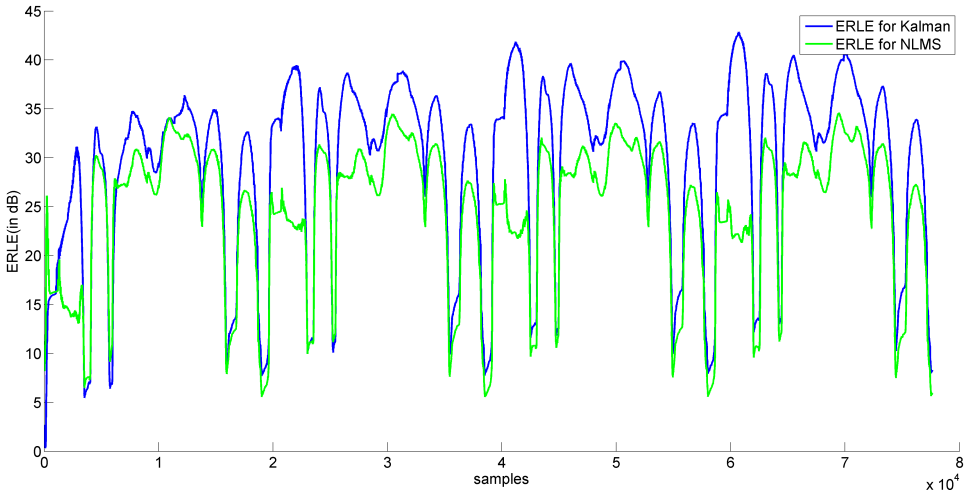

Figure 7.6. ERLE for fullband filters

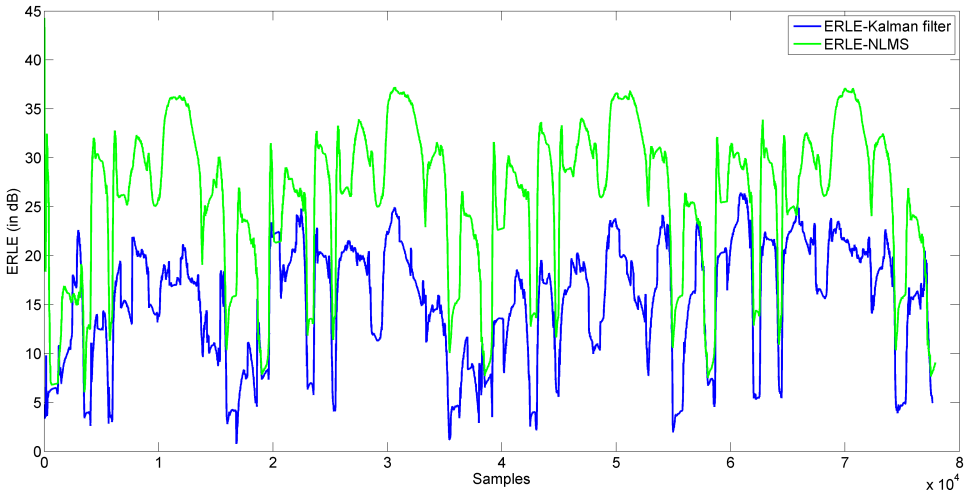

Figure 7.7. ERLE for subband filters

The residual error versus the far end signal for the fullband and subband filters for the random echo path are as shown in Figure 7.8 and Figure 7.9.

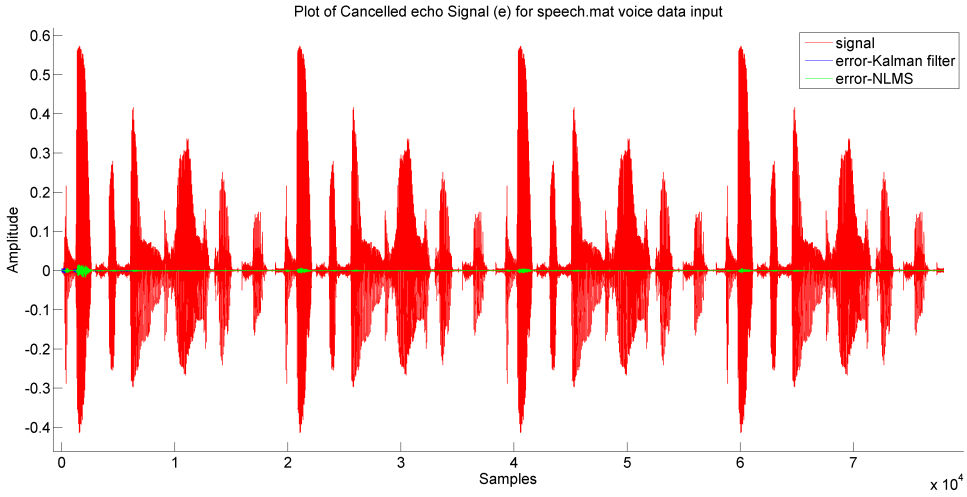

Figure 7.8. Residual error for fullband filters for random echo path

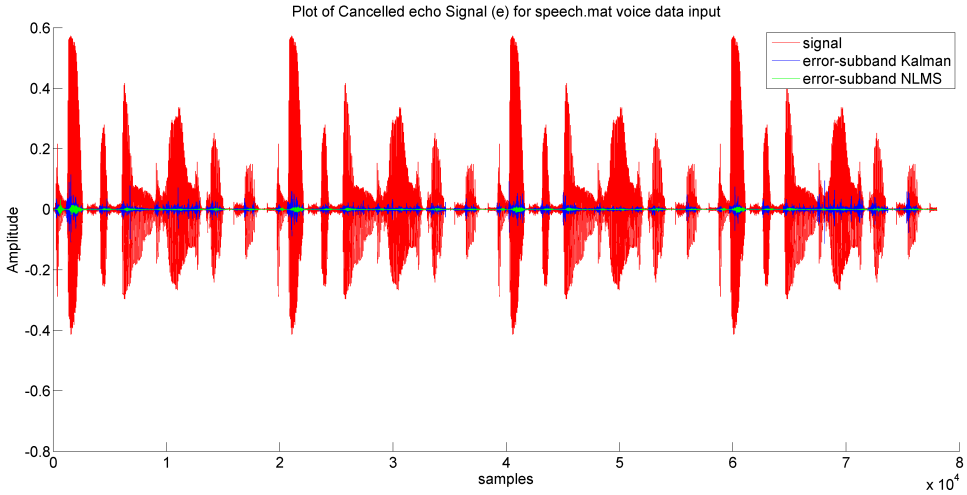

Figure 7.9. Residual error for subband filters for random echo path

The averaged ERLE for the fullband and subband filters are as shown in Figure 7.10 and Figure 7.11.

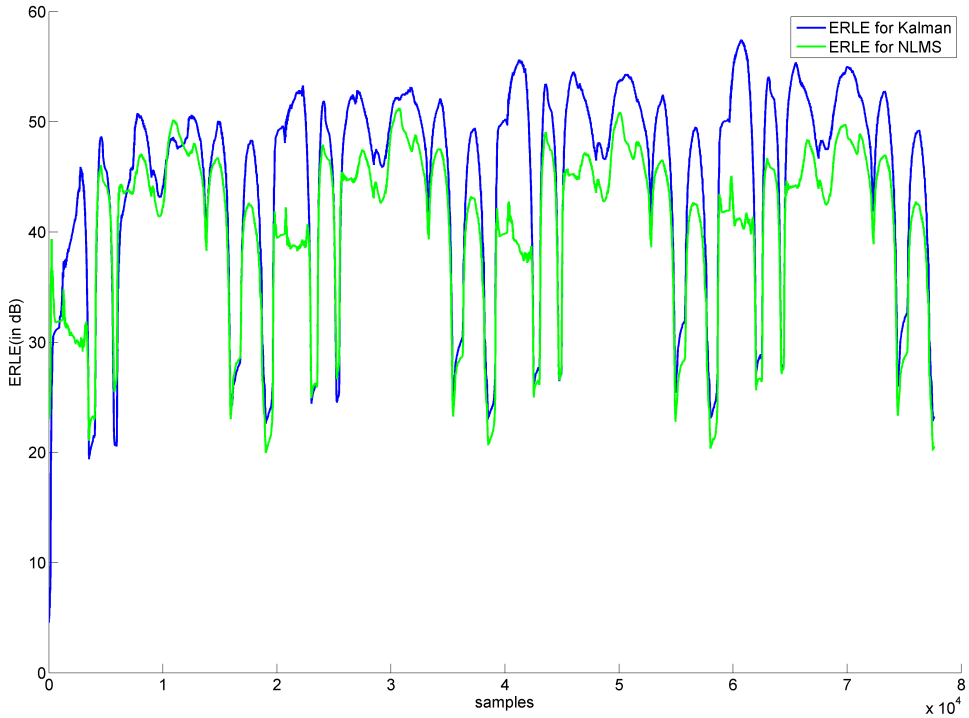

Figure 7.10. ERLE for fullband filters for random echo path

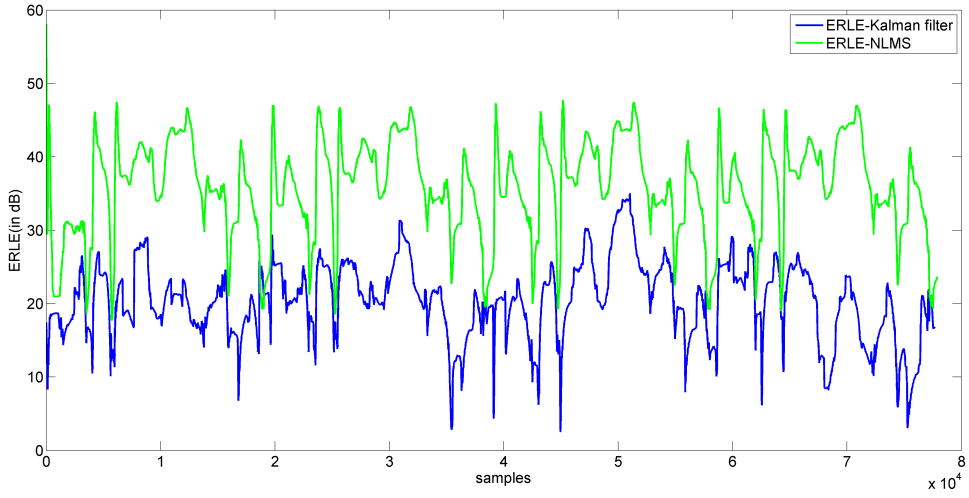

Figure 7.11. ERLE for subband filters for random echo path

As seen in the above results, the subband Kalman filter is not performing as effectively as the fullband Kalman filter, while the subband NLMS filter shows almost the same performance in both fullband and subbands. This is because of the parameter called the process noise variance, which, as mentioned earlier, captures the uncertainities of the echo path. Varying this parameter slightly affects the performance of the Kalman filter in the subbands, but not to the extent that it outperforms the NLMS. The process noise is attempted to be tracked recursively using the correlation between the current estimate of the echo path $\hat{\mathbf{h}}(n)$ and the previous estimate $\hat{\mathbf{h}}(n-1)$ using the following equation:

$$\sigma_w^2(n) = \lambda(\sigma_w^2(n-1)) + (1-\lambda).(\mathbf{deltah'deltah}) \tag{7.2}$$

where $\mathbf{deltah} = \hat{\mathbf{h}}(n) - \hat{\mathbf{h}}(n-1)$.

This equation helps improve the performance of the subband kalman filter marginally, but still not to the extent that the subband filter matches the fullband filter's performance.

7.6. VARYING SNR SCENARIO

The residual error versus the far end signal for the fullband and subband filters for varying SNR are as shown in Figure 7.12 and Figure 7.13. The SNR drops from $30dB$ to $20dB$ halfway through the simulation.

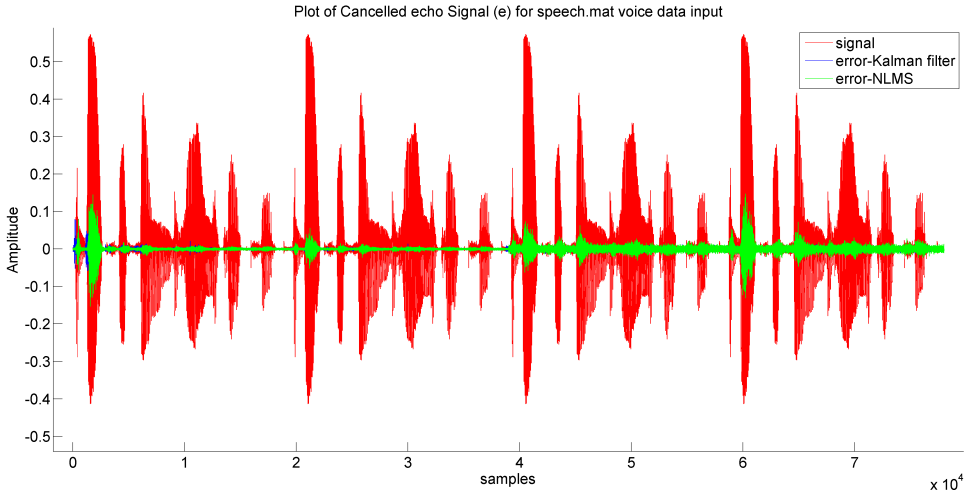

Figure 7.12. Residual error for fullband filters with change in SNR

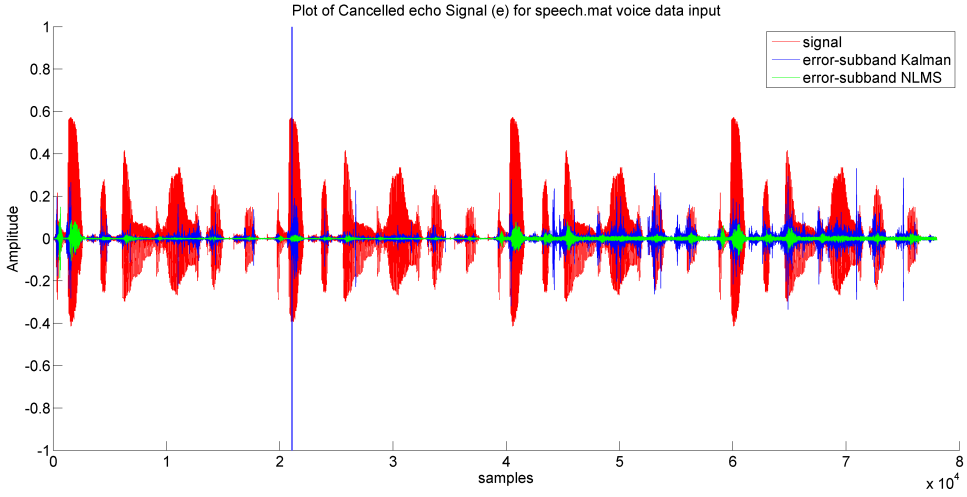

Figure 7.13. Residual error for subband filters with change in SNR

The averaged ERLE for the fullband and subband filters are as shown in Figure 7.14 and Figure 7.15.

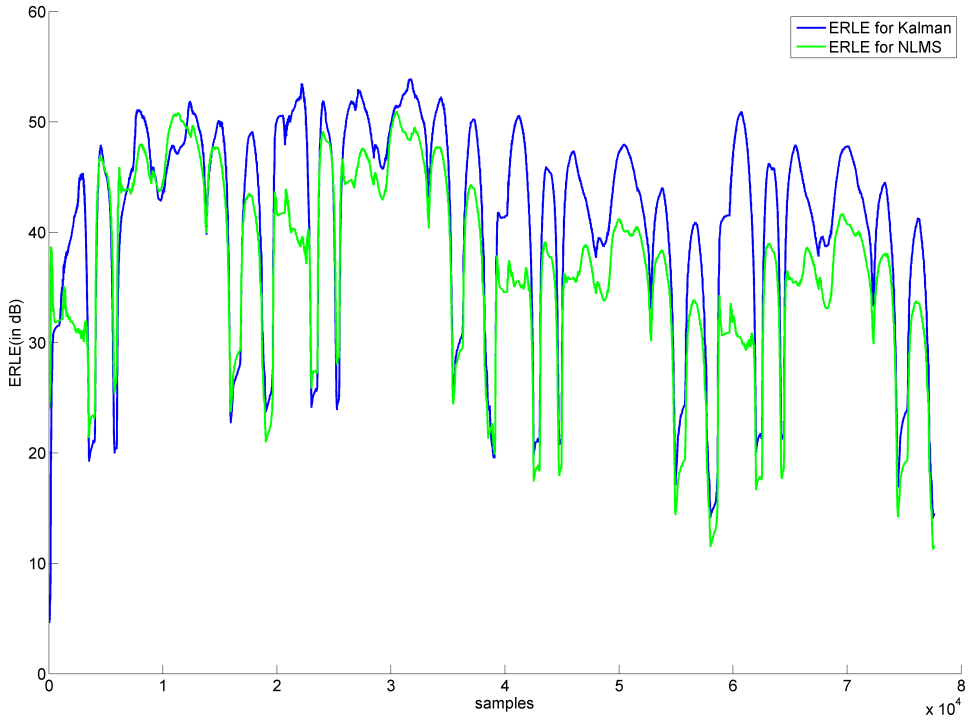

Figure 7.14. ERLE for fullband filters with change in SNR

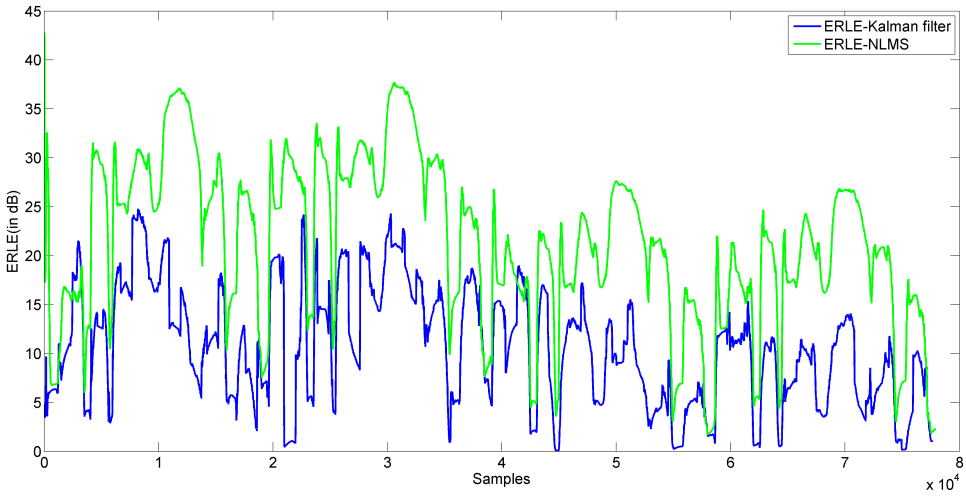

Figure 7.15. ERLE for subband filters with change in SNR

The error vs far end signal plot for a random echo path subject to a change in SNR is as shown in Figure 7.16 and Figure 7.17.

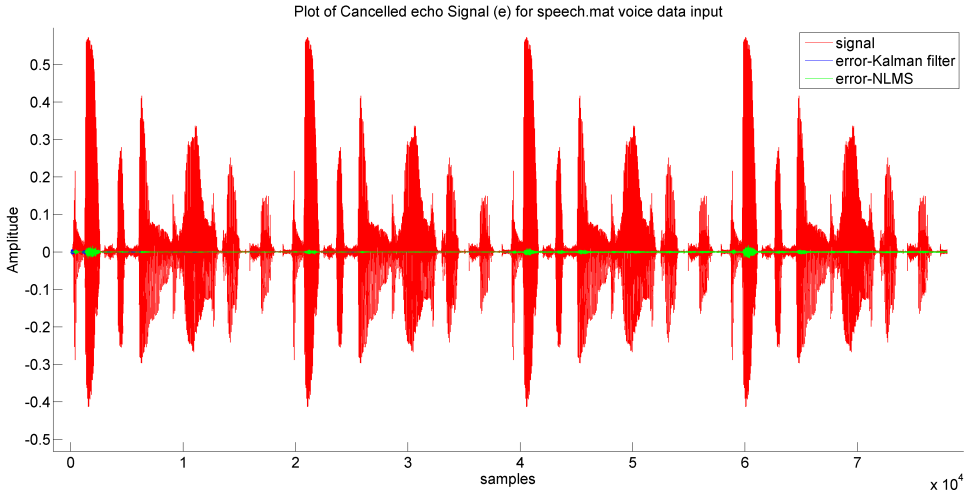

Figure 7.16. Residual error for fullband filters for a random echo path for SNR change

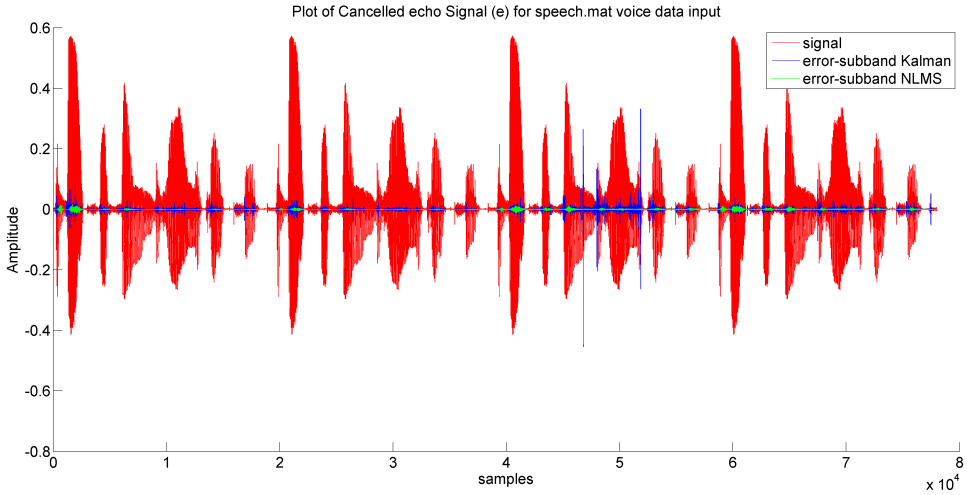

Figure 7.17. Residual error for subband filters for a random echo path for SNR change

The ERLE plot for the random echo path scenario with a change in the SNR midway through the simulations is as shown in Figure 7.18 and Figure 7.19.

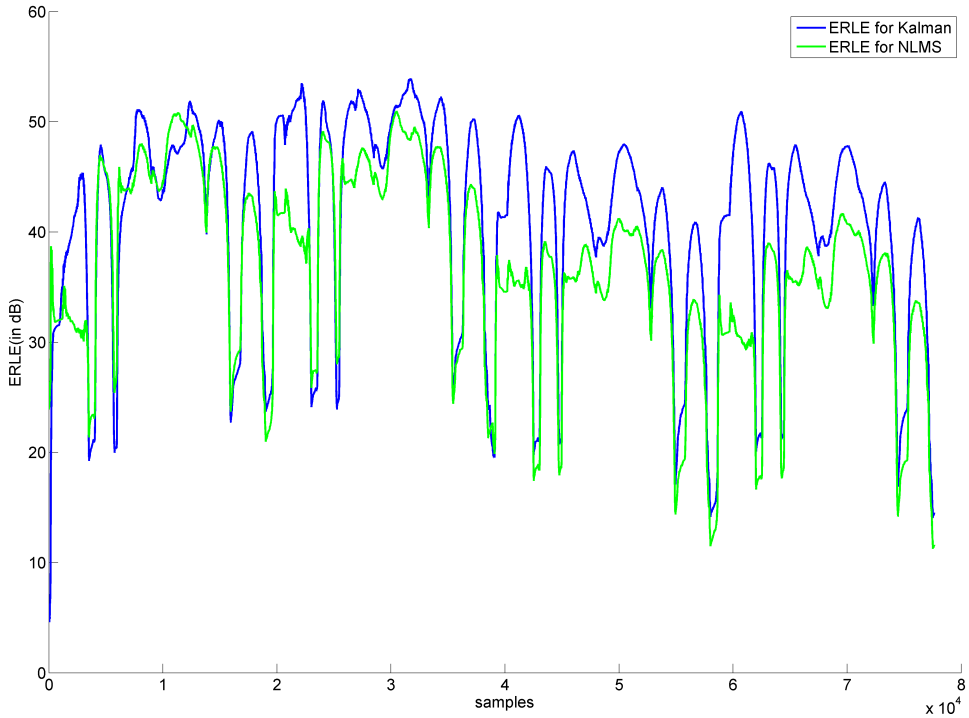

Figure 7.18. ERLE for fullband filters for a random echo path for SNR change

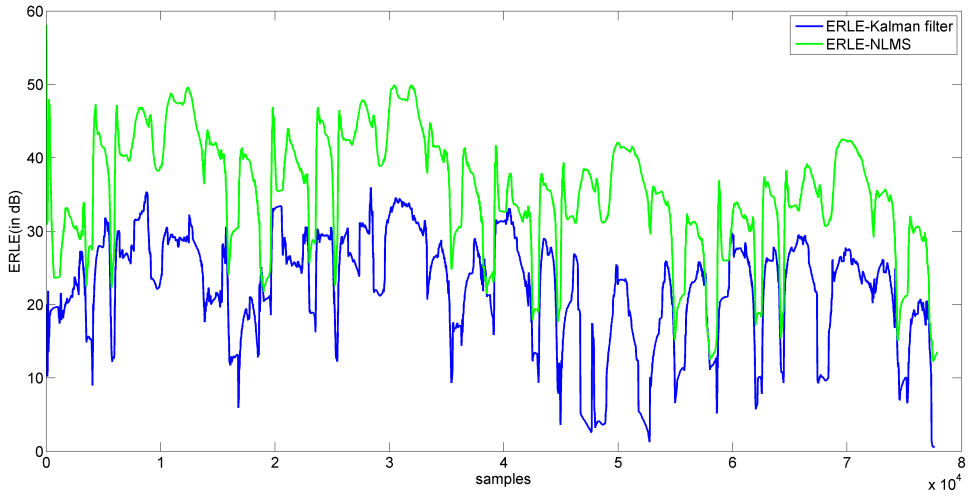

Figure 7.19. ERLE for subband filters for a random echo path for SNR change

7.7. VARYING ECHO PATH SCENARIO

Here, the echo path is changed suddenly midway through the simulation. The residual error vs far end signal plot for an echo path change is as shown in Figure 7.20 and Figure 7.21.

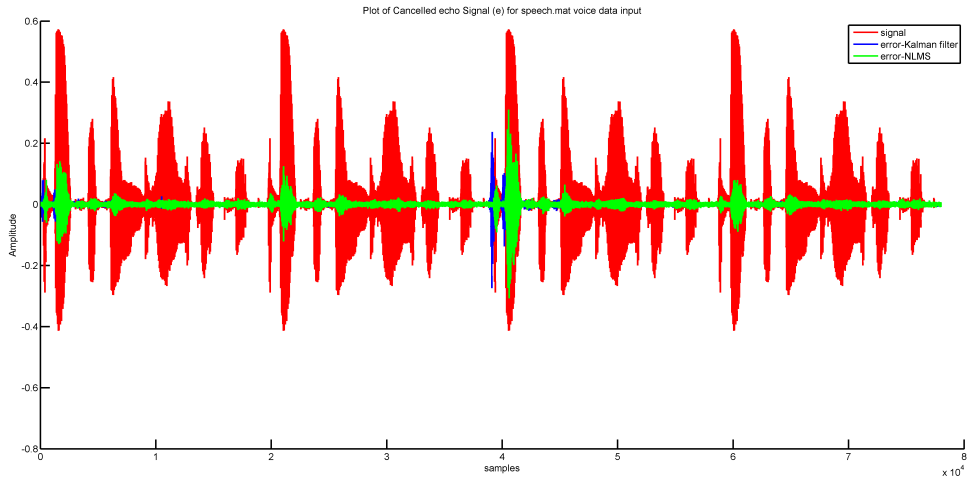

Figure 7.20. Residual error for fullband filters with echo path change

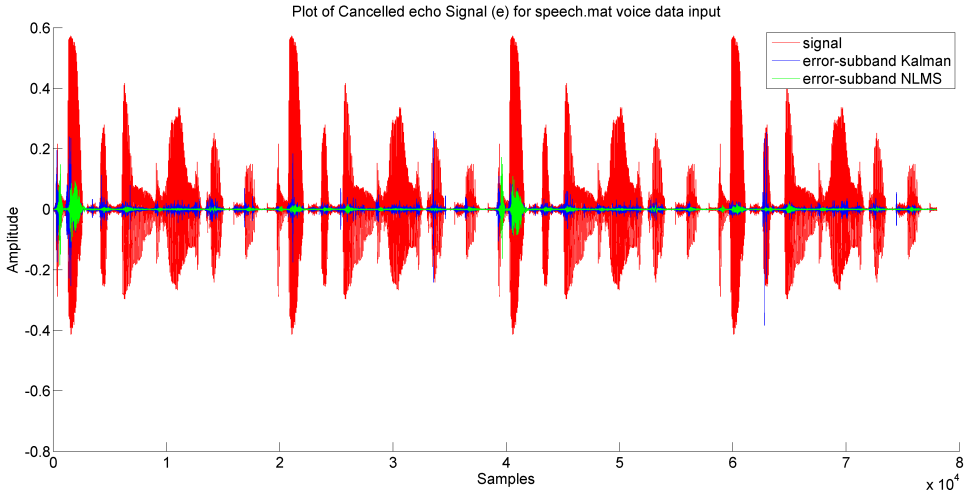

Figure 7.21. Residual error for subband filters with echo path change

The ERLE for the above scenario is shown in Figure 7.22 and Figure 7.23.

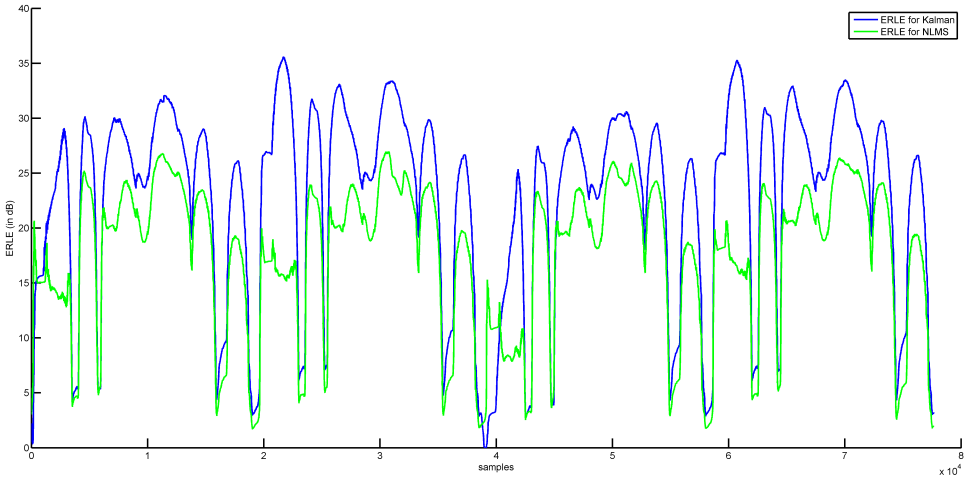

Figure 7.22. ERLE for fullband filters with echo path change

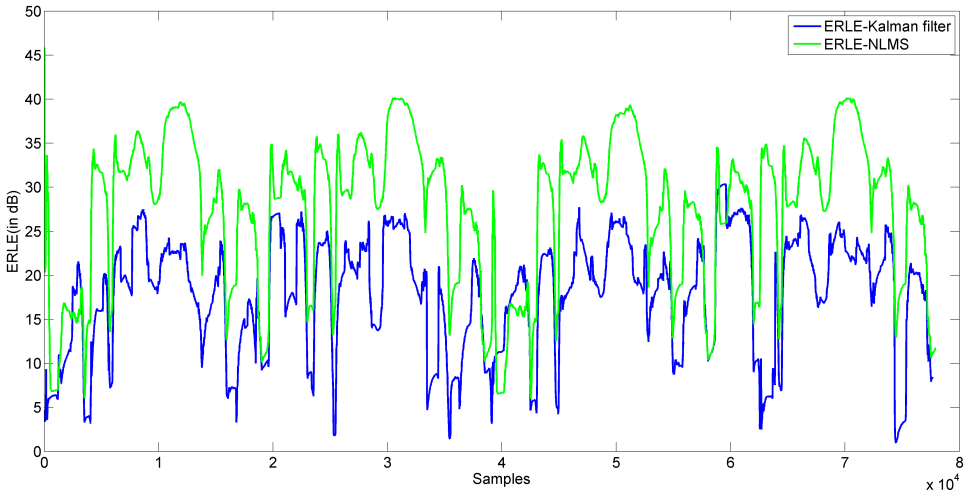

Figure 7.23. ERLE for subband filters with echo path change

The residual error for a random echo path, which changes midway through the simulation into another random echo path, is as shown in Figure 7.24 and Figure 7.25.

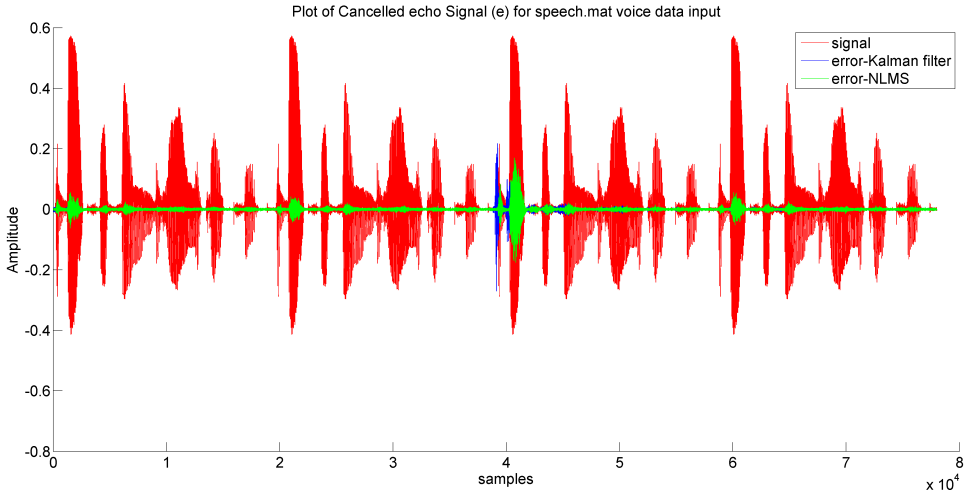

Figure 7.24. Fullband filter error for random echo path with path changes

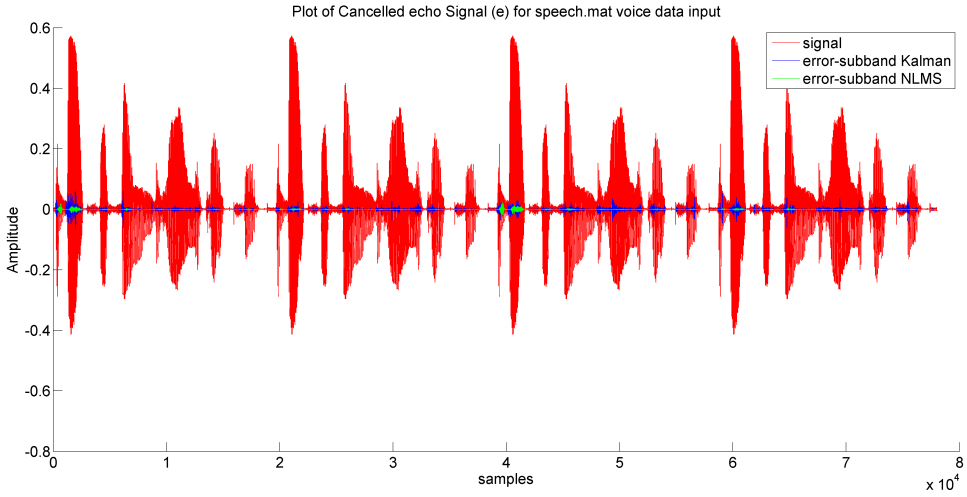

Figure 7.25. Subband filter error for random echo path with path changes

The ERLE for the above described scenario is shown in Figure 7.26 and Figure 7.27.

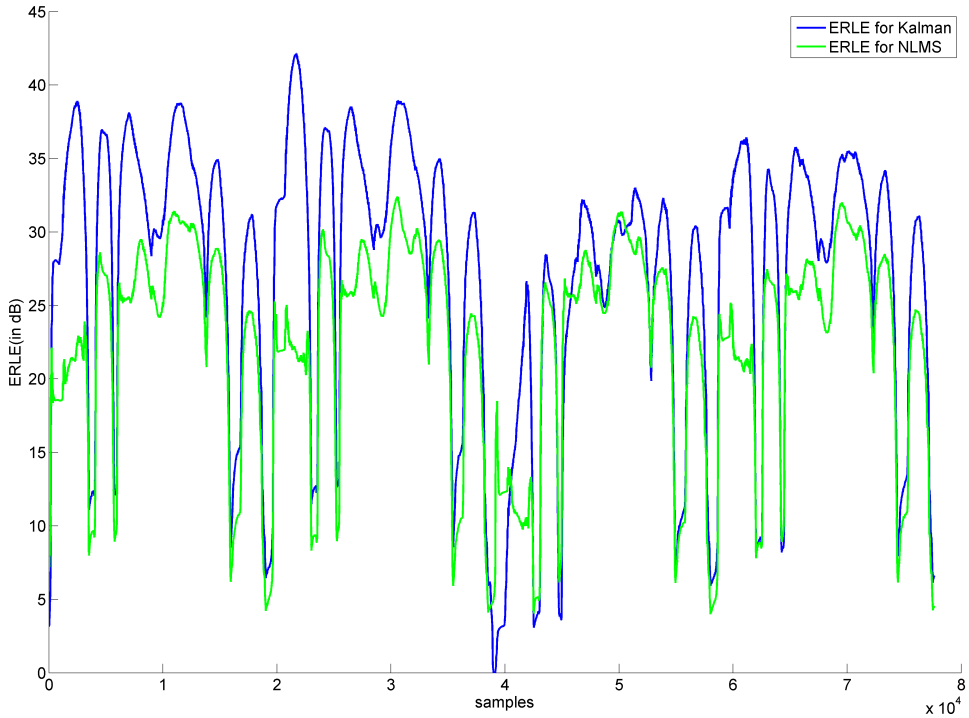

Figure 7.26. Fullband filter ERLE for random echo path with path changes

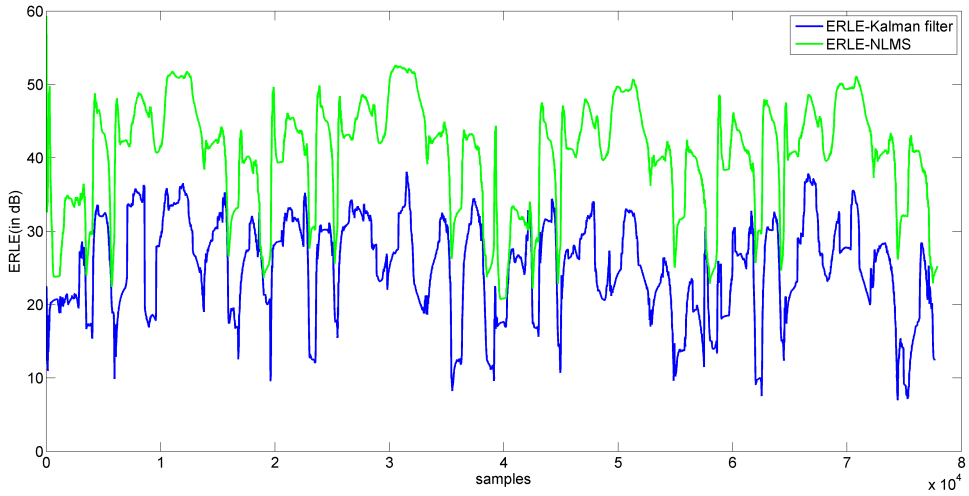

Figure 7.27. Subband filter ERLE for random echo path with path changes

7.8. DOUBLE TALK

In this scenario, halfway through the simulation, a near end speaker's signal was added to the echo signal, and the performance of the adaptive filters at distinguishing between the far end and near end signal was studied.

The residual error for the fullband and subband filters is as shown in Figure 7.28 Since near end talk was introduced mid way through the simulations, The residual error halfway through the simulations will now contain the near end speaker's speech, as observed in Figure 7.28.

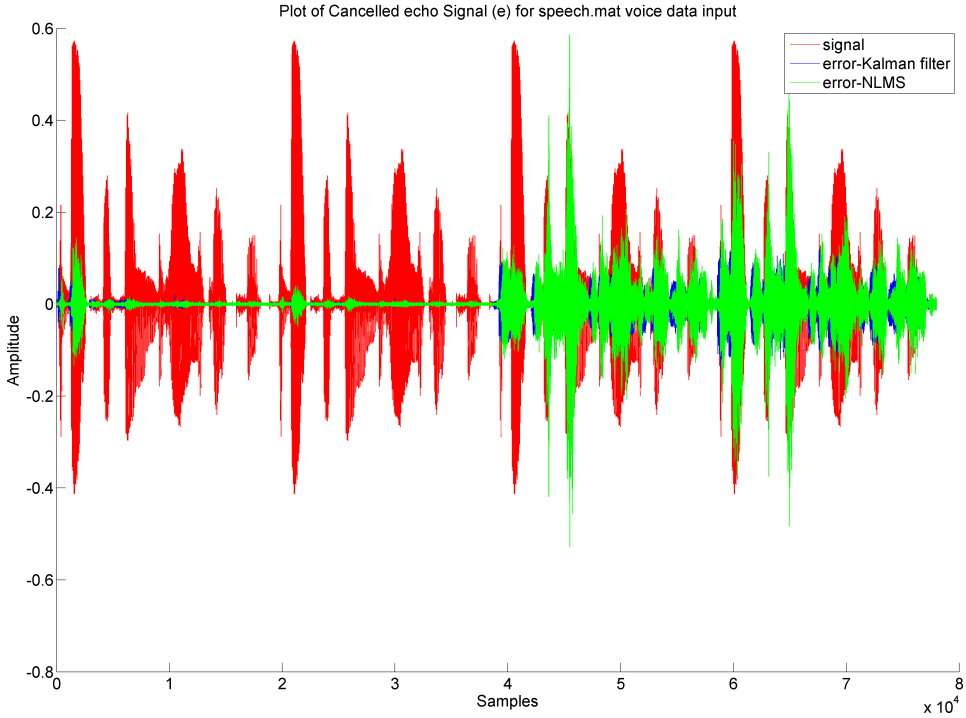

Figure 7.28. Fullband filter error for double talk

Figure 7.28 however, does not make it clear the amount of near end signal present in the residual error. Hence, only the second half of the residual error signal for each of the adaptive filters is plotted against the original near end signal to see how much of the near end signal is identified.

The near end signal vs the residual error for the fullband NLMS is as shown in Figure 7.29. The near end signal vs the residual error for the subband NLMS is as shown in Figure 7.30.

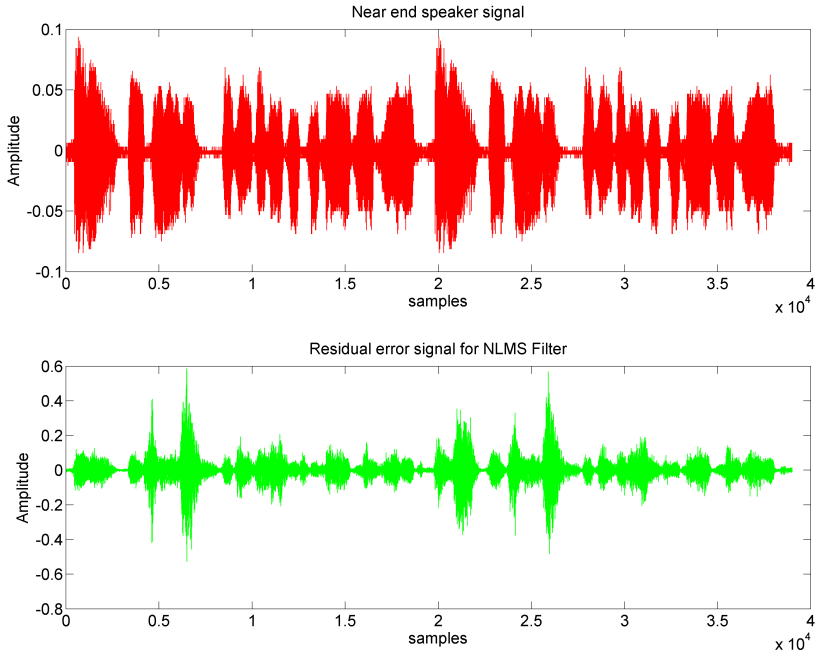

Figure 7.29. Fullband NLMS double talk signal

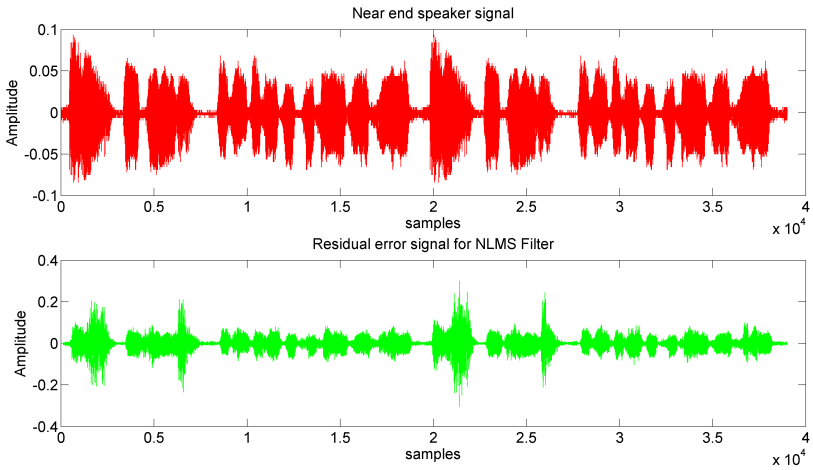

Figure 7.30. Subband NLMLS double talk signal

As seen in the above figures, in both the fullband and the subband, the double talk signal is not obtained, since the echo path estimate diverges instead of converging, adding far end signal to the double talk signal. Figure 7.31 and Figure 7.32 show the double talk signal for fullband and subband kalman adaptive filters respectively.

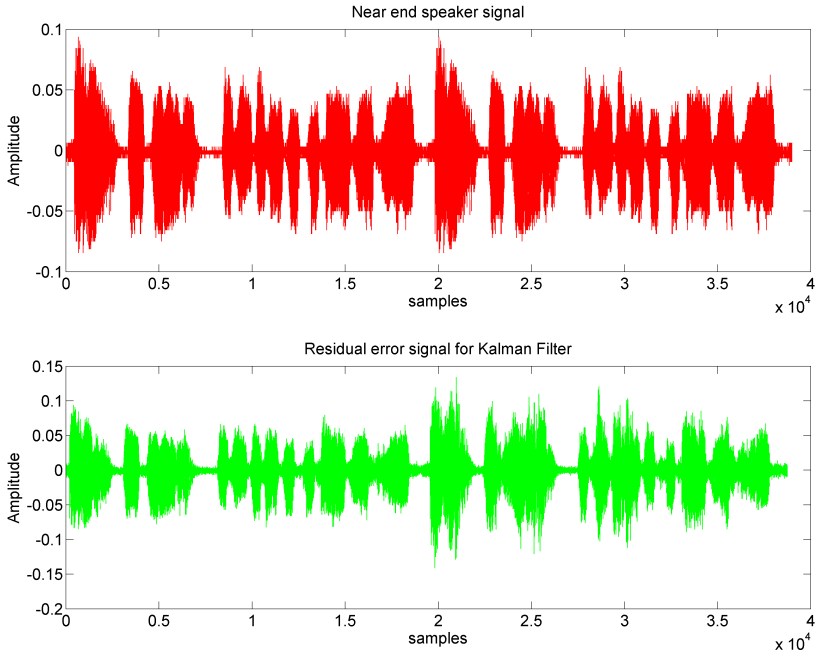

Figure 7.31. Fullband kalman double talk signal

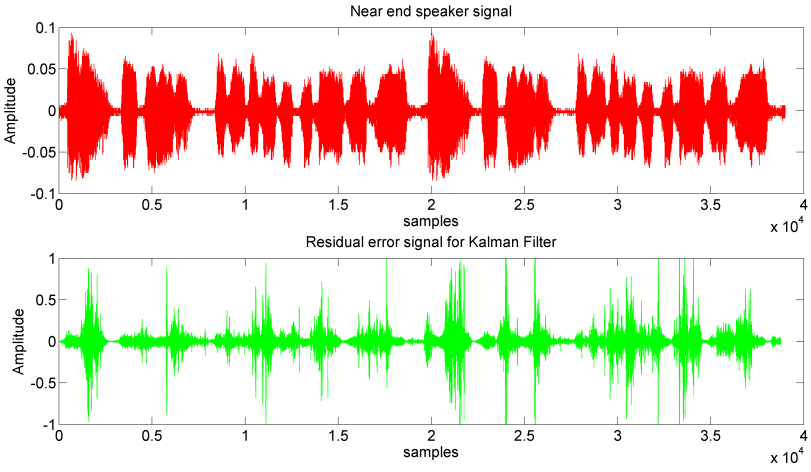

Figure 7.32. Subband kalman double talk signal

Although it can be seen that the fullband Kalman filter performs much better than the subband Kalman filter in terms of double talk detection, there is a slight improvement in the double talk detection of the subband Kalman over the subband NLMS. As discussed earlier, this is due to the process noise variance determination, and double talk detection in the subband Kalman filter can definitely be improved by focusing on estimating the process noise better. From all the above plots, it can be clearly seen that the fullband Kalman filter is superior in performance to its currently developed subband counterpart. However, the main advantage of the subbanding is the computational compexity, which will be discussed in the next part of the results.

7.9. COMPUTATIONAL COMPLEXITY AND TIME STUDIES

The Computational complexity $C_{adaptive}$ of an adaptive filter is propotional to the length of the adaptive filter, L, as $C_{adaptive} \propto L^2$. In the subbands, due to the fact that the signals are broken into smaller lengths, there is a reduction in the computational complexity which leads to the computational complexity in the subbands being $C_{subband} \propto \frac{L^2}{N^2}$. There is further computational complexity reduction since the filters are updated only every $\frac{1}{N}$ intervals, however, the effect of this reduction is negated by the fact that the subbands have to now perform N calculations for every update. A reduction in computational complexity by $N.^2$ is very profitable, especially when dealing with long echo paths. For a 1000 length echo path, the computational complexity of a fullband adaptive filter is 1000000 while for a subband adaptive filter for the same echo path, the complexity reduces to ≈ 977. This is a huge gain, hence, subband adaptive filters are favored for real time implementations, where memory is limited and the adaptation needs to be performed quickly.

To notice just how fast subband filters are compared to fullband filters, Table 7.1 can be referred, where the simulation time for fullband and subband filters are noted for a single iteration.

Filter	Fullband Kalman	Subband Kalman	Fullband NLMS	Subband NLMS
Time in seconds	661.531	20.816802	3.03125	6.099183

Table 7.1. Simulation time of the studied adaptive filters

As seen in the above table, the time of simulation is reduced greatly for the subband Kalman filter compared to the fullband Kalman filter, which makes it favorable for use in real time scenarios.

8. CONCLUSION AND FUTURE WORK

In this thesis, a subband Kalman filter was developed and studied against existing adaptive filter algorithms to evaluate its performance and potential for real time applications. Future work would involve study of the estimation of process noise in the subbands, which is a major performance factor as discussed in the results. The measurement noise variance update equation also needs investigation, since the assumption of both measurement and process noise being white, gaussian and uncorrelated will not hold in the subbands. Finally, a real time implementation of the developed kalman filter could help to see if actual real time performance matches MATLAB simulated results.

APPENDIX A

FULLBAND NLMS AND KALMAN FILTERS-MATLAB CODE

```
tic
clc
clear all
close all
load speech.mat
% h1 = randn(256,1); %random echo path scenario
% h2 = h1;
h1=zeros(256,1); %impulse response with one
   sample with value 1
h1(128)=1;
h2=zeros(256,1);
h2(128)=1;
% h2 = randn(256,1); %change in echo path
   scenario
h=h1;
n = length(h1);
x = 10*speech; %input signal
x_near_end = 0.1*wavread('nineoneone.wav');
x_near_end = [x_near_end;x_near_end;x_near_end;
   x_near_end;x_near_end];
[y1,state] = filter(h,1,x); %filtered output
y2 = filter(h2,1,x,state); %filtered output
y=[y1;y2];
x=[x;x];
iter = 5;
delta3 = 0.03;
mu = 1;
for ii=1:iter
    clear y1
% y1 = y; %ideal no noise scenario
 y1 = awgn(y,30,'measured'); %constant SNR
    scenario

%SNR changing scenario
%     y1 = y;
%     y2 = y1*0;
%     y2(1:round(length(y1)/3)) = awgn(y1(1:
   round(length(y1)/3)),25,'measured');
%     y2(round(length(y1)/3)+1:round(2*length(y1
   )/3)) = awgn(y1(round(length(y1)/3)+1:round(2*
   length(y1)/3)),18,'measured');
%     y2(round(2*length(y1)/3)+1:end) = awgn(y1(
   round(2*length(y1)/3)+1:end),10,'measured');
%     y1 = y2;
```

```
%       y1 = y1+ x_near_end(1:length(y1)); %
   double talk scenario

Il = eye(n); %identity matrix of size nXn
 Ip = eye(1); %identity matrix of size 1X1
delta=.00001;
Rmu = delta*Il; %apriori misalignment
   initialization
w_cov_const = 0.01; %process noise variance
v_cov_const = 0.1; %observation noise variance
hhat = zeros(length(h),1);
lambda=.9;
lambda_v = 0.999;
sigmae = 0.001;
sigmax = 0.001;
Rex = 0.001*ones(n,1);
for i=1:length(x)-n+1
    X = (x(i:i+n-1));
    Rm = Rmu + w_cov_const*Il;
    Re = X.'*Rm*X + (v_cov_const)*Ip;
    K = Rm*X/(Re+delta3);
    e(i) = (y1(i+n-1) - X.'*hhat);
    hhold=hhat;
    hhat = hhat + K*e(i);
    Rmu = (Il-K*X.')*Rm;
    deltah=hhat-hhold;
    temp = deltah.*deltah;


    w_cov_const=lambda*w_cov_const+(1-lambda)*(
       deltah.'*deltah);
%     background noise tracking implemented here
   ...
        Rex = lambda_v*Rex + (1-lambda_v)*X*e(i);
            %correlation between far end signal
           and residual error
         sigmax = lambda_v*sigmax + (1-lambda_v)*
            X(end).^2; %excitation signal variance
        sigmae = lambda_v*sigmae + (1-lambda_v)*e
           (i).^2; %cancellation error variance
        v_cov_const = (sigmae - (1/(sigmax+delta3
           )*(Rex.'*Rex)));
```

```
end
%NLMS algorithm implemented here
    hhat_nlms = zeros(length(h),1);
    X_nlms=hhat_nlms*0;

for i=1:1:length(x)
        X_nlms(2:n)=X_nlms(1:n-1);
        X_nlms(1)=x(i);
        e_nlms(i)=y1(i)-hhat_nlms'*X_nlms;
        hhat_nlms=hhat_nlms+ mu.*conj(e_nlms(i))
           .*X_nlms./(X_nlms'*X_nlms+delta);


end
    error1(ii,:) = e;
    error1_nlms(ii,:) = e_nlms;
end

average_error1 = sum(error1)/iter;
average_error1_nlms = sum(error1_nlms)/iter;

figure(1);
hold on;
plot(real(x),'r');
plot(real(average_error1));
plot(real(average_error1_nlms),'g');
title('Plot of Cancelled echo Signal (e) for
   speech.mat voice data input');
legend('signal','error-Kalman filter','error-
   NLMS');


x1=x(1:length(average_error1));
x2 = x(1:length(average_error1_nlms));
xx=filter(ones(1000,1)./1000,1,x1.^2);
xx2 = filter(ones(1000,1)./1000,1,x2.^2);
xx = xx.';
xx2 = xx2.';
ee=filter(ones(1000,1)./1000,1,average_error1
   .^2);
ee_nlms = filter(ones(1000,1)./1000,1,
   average_error1_nlms.^2);
erle=10.*log10(((xx)./(ee))+1);
erle_nlms = 10.*log10(((xx2)./(ee_nlms))+1);
```

```
figure(2);
hold on;
plot(erle(100:end),'linewidth',2)
plot(erle_nlms(100:length(x)-n+1),'g','linewidth
   ',2);
legend('ERLE for Kalman','ERLE for NLMS');
toc

%DT plots in figures (3) and (4)
figure(3);
subplot(2,1,1);
plot(x_near_end(1:length(y1(round(length(y1)/2):
   end))),'r')
xlabel('samples')
ylabel('Amplitude')
title('near end signal')
subplot(2,1,2);
plot(average_error1((round(length(y1)/2)):end),'
   g')
ylim([min(y11) max(y11)])
xlabel('samples')
ylabel('Amplitude')
title('error signal for subband kalman filter')
figure(4);
subplot(2,1,1)
plot(x_near_end(1:length(y1(round(length(y1)/2):
   end))),'r')
xlabel('samples')
ylabel('Amplitude')
title('near end signal')
subplot(2,1,2)
plot(average_error1_nlms((round(length(y11)/2)):
   end),'g')
ylim([min(y11) max(y11)])
xlabel('samples')
ylabel('Amplitude')
title('error signal for subband NLMS')
```

APPENDIX B

SUBBAND NLMS AND KALMAN FILTERS-MATLAB CODE

```
tic
clc
clear all
close all
rng('shuffle');
load EE341_filters; %load the filters
load speech; % load the speech file
h1 = 0.01*randn(256,1);
h2 = h1;
% h1=zeros(256,1); %impulse response with one
   sample with value 1
% h1(128)=1;
% h2=zeros(256,1);
% h2(128)=1;
% h2 = 0.03*randn(256,1); %change in echo path
   scenario
h=h1;
n = length(h);
x1 = 10*speech; %input signal
x_near_end = 0.1*wavread('nineoneone.wav');
x_near_end = [x_near_end;x_near_end;x_near_end;
   x_near_end;x_near_end];
delta2 = 0.01;
delta3 = 0.035;
[y1,state] = filter(h,1,x1); %filtered output
y2 = filter(h2,1,x1,state); %filtered output
y1=[y1;y2];
x1=[x1;x1];
iter = 5;

for ii=1:iter
    clear y11;
    %      y11 = y1; %ideal no noise scenario
    y11 = awgn(y1,30,'measured');  %constant SNR
        scanario


    %  changing SNR scanario below
    %         y11 = y1;
    %         y11(1:round(length(y1)/2)) = awgn(
       y11(1:round(length(y11)/2)),30,'measured')
       ;
    %         y11(round(length(y11)/2)+1:end) =
       awgn(y11(round(length(y11)/2)+1:end),20,'
```

```
   measured');



%double talk scenario
%       y11(round(length(y11)/2):end) = y11(
   round(length(y11)/2):end) + x_near_end(1:
   length(y11(round(length(y11)/2):end)));

Ip = eye(1); %identity matrix of size 1X1
delta=.00001; %normalizing factor
N=32; % Number of subbands between 0 and 2*
   pi
R=16; % Downsampling factor in subbands
w_cov_const = 1.5/R; %process noise variance
v_cov_const = 1.5/R; %observation noise
   variance
Lsub=ceil(n./R);
hhat = zeros(Lsub,1); %initialising estimate
    of h as hhat
Il = eye(Lsub); %identity matrix of size
   LsubXLsub
Rmu = delta*Il; %apriori misalignment
   initialisation
[xsub] = sba_local(x1,anal_sbflt,N,R); %
   converting input signal into subbands
[ysub] = sba_local(y11,anal_sbflt,N,R); %
   converting filtered signal into subbands


[rows,cols] = size(xsub); %determining
   number of rows
lambda=.999;
lambda_v = 0.999;
e=zeros(rows,cols-Lsub+1);

for jj =1:rows %accessing all columns of a
   particular row at once
    hhat = zeros(Lsub,1); %initialising
       estimate of h as hhat for every
       iteration
    Rmu = delta*Il; %aposteriori
       misalignment initialisation
    x = xsub(jj,:); %accessing one subband
```

```
   of input signal at a time
y = ysub(jj,:); %accessing one subband
   of filtered signal at a time
w_cov_var=w_cov_const;
v_cov_var = v_cov_const;
mu = 1;
Rex = 0.001*ones(Lsub,1);
sigmax = 0.001;
sigmae = 0.001;

for i=1:cols-Lsub+1
    X = (x(i:i+Lsub-1).');
    Rmu = Rmu + abs(w_cov_var)*Il; %
       apriori Misalignment update
    Re = X.'*Rmu*X + (v_cov_var)*Ip; %
       apriopri error signal correlation
    K = (Rmu*X)/(Re+ delta3); %Kalman
       Gain calculation
    e(jj,i) = (y(i+Lsub-1) - X.'*hhat);
       %error signal calculation
    hhold=hhat;
    hhat = hhat + K*e(jj,i); %h estimate
        update equation
    deltah=hhat-hhold;
    Rmu = (Il-K*X.')*Rmu; %aposteriori
       Misalignment update
    w_cov_var=lambda*w_cov_var+(1-lambda
       )*(deltah.'*deltah);
    %background noise tracking
       implemented here...
    Rex = lambda_v*Rex + (1-lambda_v)*X*
       e(jj,i); %correlation between far
       end signal and residual error
    sigmax = lambda_v*sigmax + (1-
       lambda_v)*(abs(X(end))).^2; %
       excitation signal variance

    sigmae = lambda_v*sigmae + (1-
       lambda_v)*(abs(e(jj,i))).^2; %
       cancellation error variance
    v_cov_var = (sigmae - (1/(sigmax+
       delta3))*abs(Rex.'*Rex));
```

```
        end
        %NLMS performed here
        hhat_nlms = zeros(Lsub,1);
        X_nlms=hhat_nlms*0;

        for i=1:1:cols
            X_nlms(2:Lsub)=X_nlms(1:Lsub-1);
            X_nlms(1)=x(i);
            e_nlms(jj,i)=y(i)-hhat_nlms'*X_nlms;
            hhat_nlms=hhat_nlms+ mu.*conj(e_nlms
               (jj,i)).*X_nlms./(X_nlms'*X_nlms+
               delta);


        end

    end
    % e_10(ii,:,:) = e;
    error1(ii,:) = sbs_local(e,synth_sbflt,N,R);
        %reconstructing the error signal using
       synth_sbflt
    error1_nlms(ii,:) = sbs_local(e_nlms,
       synth_sbflt,N,R); %reconstructing the
       error signal using synth_sbflt
end


average_error1 = sum(error1)/iter;
average_error1_nlms = sum(error1_nlms)/iter;
figure(1);
hold on;
plot(real(x1),'r');
plot(real(average_error1));
plot(real(average_error1_nlms),'g');
title('Plot of Cancelled echo Signal (e) for
   speech.mat voice data input');
legend('signal','error-subband Kalman','error-
   subband NLMS');
hold off;
xlabel(['Samples'])
ylabel(['Amplitude'])
figure(2)
```

```
x11=x1(1:length(average_error1));
x12 = x1(1:length(average_error1_nlms));
xx=filter(ones(1000,1)./1000,1,x11.^2);
xx1 = filter(ones(1000,1)./1000,1,x12.^2);
ee=filter(ones(1000,1)./1000,1,real(
   average_error1.^2));
ee_nlms = filter(ones(1000,1)./1000,1,real(
   average_error1_nlms.^2));

erle=10.*log10(((xx)./(ee).')+1);
erle_nlms=10.*log10(((xx1)./(ee_nlms).')+1);
plot(erle(100:end),'linewidth',2);
hold on;
plot(erle_nlms(100:end),'g','linewidth',2);
legend('ERLE-Kalman filter','ERLE-NLMS')
xlabel(['Samples']);
ylabel(['ERLE(in dB)']);
toc
average_error1_n = 0.8*(average_error1/max(abs(
   average_error1)));

%DT plots in figures (3) and (4)
figure(3);
subplot(2,1,1);
plot(x_near_end(1:length(y11(round(length(y11)
   /2):end))),'r')
xlabel('samples')
ylabel('Amplitude')
title('near end signal')
subplot(2,1,2);
plot(average_error1_n((round(length(y11)/2)):end
   ),'g')
ylim([min(y11) max(y11)])
xlabel('samples')
ylabel('Amplitude')
title('error signal for subband kalman filter')
figure(4);
subplot(2,1,1)
plot(x_near_end(1:length(y11(round(length(y11)
   /2):end))),'r')
xlabel('samples')
ylabel('Amplitude')
title('near end signal')
subplot(2,1,2)
```

```
plot(average_error1_nNLMS((round(length(y11)/2))
   :end),'g')
ylim([min(y11) max(y11)])
xlabel('samples')
ylabel('Amplitude')
title('error signal for subband NLMS')
```

APPENDIX C

ANALYSIS FILTERBANK-MATLAB CODE

```
function [xsub] = sba_local2(xlong,anal_sbflt,N,
   R);

L=length(anal_sbflt);

LN=L/N;

H=diag(anal_sbflt(1:N));
for n=2:LN
    H=[H, diag(anal_sbflt((n-1)*N+1:n*N))];
end

r=N/2;
lngexp=r*(floor(length(xlong)/r));

x=zeros(L,1);

m=0;
for n=1:r:lngexp

 %    x(2:L)=x(1:L-1);
 %    x(1)=xlong(n)


    x(r+1:L)=x(1:L-r);
    x(1:r)=flipud(xlong(n:n+r-1));

    m=m+1;
    xsub(:,m)=N*ifft(H*x);
end
```

APPENDIX D

SYNTHESIS FILTERBANK-MATLAB CODE

```
function xhat = sbs_local2(xsub,synth_sbflt,N,R)
   ; % use your own synthesis filterbank here

L=length(synth_sbflt);

LN=L/N;

G=diag(synth_sbflt(1:N));
for n=2:LN
    G=[G; diag(synth_sbflt((n-1)*N+1:n*N))];
end

r=N/2;
lngexp=r*size(xsub,2);

s=zeros(L,1);

m=0;
xhat=zeros(lngexp,1);
for n=1:lngexp

   s(1:L-1)=s(2:L);
   s(L)=0;

    if rem(n,r)==0
        y=N*ifft(xsub(:,n/r));
        s=G*y+s;
    end

  xhat(n)=s(1);
end
```

BIBLIOGRAPHY

[1] D. Manolakis, V. Ingle, and S. Kogon, *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering, and Array Processing*. Artech House signal processing library, Artech House, 2005.

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

[3] S. Haykin, *Adaptive Filter Theory (3rd Ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.

[4] A. V. Oppenheim and G. C. Verghese, "Signals, systems, and inference," *Class notes for*, vol. 6, 2010.

[5] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[6] C. Paleologu, J. Benesty, and S. Ciochina, "Study of the general kalman filter for echo cancellation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, pp. 1539–1549, Aug 2013.

[7] A. Gilloire, "Experiments with sub-band acoustic echo cancellers for teleconferencing," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '87.*, vol. 12, pp. 2141–2144, Apr 1987.

[8] W. Kellermann, "Analysis and design of multirate systems for cancellation of acoustical echoes," in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pp. 2570–2573 vol.5, Apr 1988.

[9] P. Eneroth, S. Gay, T. Gansler, and J. Benesty, "A real-time implementation of a stereophonic acoustic echo canceler," *Speech and Audio Processing, IEEE Transactions on*, vol. 9, pp. 513–523, Jul 2001.

[10] J. Benesty, T. Gänsler, D. R. Morgan, M. M. Sondhi, S. L. Gay, *et al.*, *Advances in network and acoustic echo cancellation*. Springer, 2001.

VITA

Rakesh Vijayakumar was born in Bangalore, India. He obtained his Bachelor of Engineering (B.E) in Electrical Engineering from Sir M.Visvehwaraih Institute of Technology, Bangalore, in 2012. In December 2015, he obtained his MS in Electrical Engineering, from Missouri University of Science and Technology, Rolla.