2008-03-12

Improving Performance of the Filtered-X Least Mean Square Algorithm for Active Control of Noise Contatining Multiple Quasi-Stationary Tones

Stephan P. Lovstedt
Brigham Young University - Provo

IMPROVING PERFORMANCE OF THE FILTERED-X LEAST MEAN SQUARE ALGORITHM FOR ACTIVE CONTROL OF NOISE CONTAINING MULTIPLE QUASI-STATIONARY TONES

by

Stephan Paul Lovstedt

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Physics and Astronomy

Brigham Young University

April 2008

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Stephan Paul Lovstedt

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Scott D. Sommerfeldt, Chair
Date	Kent L. Gee
Date	Jonathon D. Blotter

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Stephan P. Lovstedt in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Scott D. Sommerfeldt
Chair, Graduate Committee

Accepted for the Department

Date

J. Ward Moody, Graduate Coordinator
Department of Physics and Astronomy

Accepted for the College

Date

Thomas W. Sederberg, Associate Dean
College of Physical and Mathematical Sciences

ABSTRACT

IMPROVING PERFORMANCE OF THE FILTERED-X LEAST MEAN SQUARE ALGORITHM FOR ACTIVE CONTROL OF NOISE CONTAINING MULTIPLE QUASI-STATIONARY TONES

Stephan Paul Lovstedt

Department of Physics and Astronomy

Master of Science

The Filtered-X Least-Mean-Square (FXLMS) algorithm is widely used in active noise control due to its robustness, simplicity, and ability to be implemented in real time. In a feedforward implementation of the FXLMS algorithm, a reference signal that is highly correlated with the noise to be controlled is filtered with an estimate of the transfer function of the secondary path. The convergence characteristics of the FXLMS algorithm have been well studied. A convergence parameter is used to optimize the convergence of the algorithm. However, the optimal value for the convergence parameter is frequency dependent. Thus for noise containing multiple tones at different frequencies the convergence parameter can only be optimized for one of those tones. Other tones will have slower convergence rates and in general less attenuation than they would have if

they were treated singly and parameters could be optimized for those frequencies separately. A method is developed to modify the magnitude response of the secondary path estimate while maintaining the original phase response, which equalizes the convergence characteristics over multiple frequencies, giving more uniform convergence and attenuation for all tones being controlled. Stability of the algorithm is not compromised. The modification to the FXLMS algorithm is relatively simple to implement and has been shown to increase overall attenuation of a signal containing multiple tones by an additional 6-9 dB.

ACKNOWLEDGMENTS

I would like to thank the following people:

- Scott Sommerfeldt for being my graduate advisor
- Jon Blotter and Kent Gee for serving on my graduate committee
- Tim Leishman for helping me get involved in acoustics research
- Jared Thomas who collaborated closely with me for most of my research
- Ben Faber for developing much of the ANC resources I needed for my research
- BYU Acoustics Research Group for general support and camaraderie
- My wife and daughter Marnie and Penelope for their love, support and patience.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1- INTRODUCTION

1.1 Active Noise Control Basics

Active Noise Control (ANC) uses the principle of superposition of waves. The net displacement of the medium through which two or more waves are traveling is the sum of their individual wave displacements. Consider the addition of two identical waves (same frequency and amplitude) shown in Figure 1.1. The net amplitude of the resulting wave depends on the relative phase of the two waves. When the positive antinodes of one wave align with negative antinodes of the other (exactly out of phase) the combined wave displacement will be zero. The same principle holds for more complicated waveforms and sound fields.

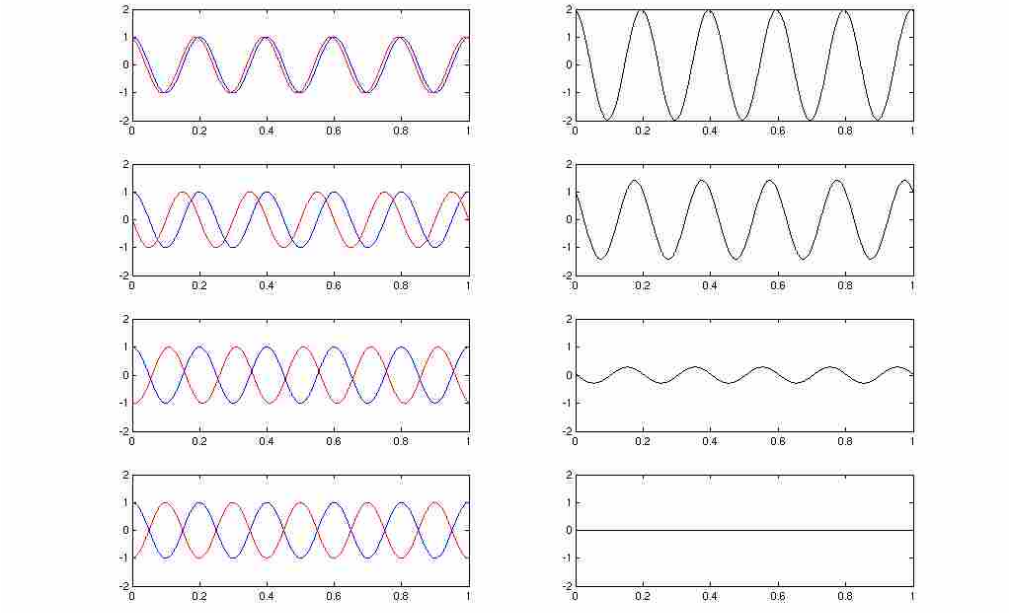

Figure 1.1 This figure demonstrates the principle of superposition. In the left column two identical sine waves with some phase difference between them are plotted together. The right column shows the result of adding the two waves on the left. When the waves are exactly out of phase (bottom plots) the addition of the two waves results in no wave motion.

ANC then works by electroacoustically creating a wave disturbance in a medium that attenuates, through superposition, an unwanted wave disturbance or noise. A feedforward ANC system (the type considered in this thesis) requires four basic components[1]:

(1) Reference Sensor

(2) Control System

(3) Control Loudspeaker(s)

(4) Error Sensor(s)

Figure 1.2 shows how these basic components can be combined in an ANC system. The reference sensor measures the unwanted noise and produces a reference signal that is correlated to and characterizes the spectral content of the unwanted noise. This reference signal is "fed forward" to the controller so that it can determine the proper control signal before the noise has reached the listening location. The control system uses this reference signal to create a control signal to drive a loudspeaker which creates a canceling noise that will attenuate the unwanted noise when combined at the error sensor. The error sensor measures the residual noise after the unwanted noise and control signal have combined and sends an error signal back to the controller to adapt the control system in an attempt to further reduce the error. The control system adaptively modifies the control signal to minimize the residual error.

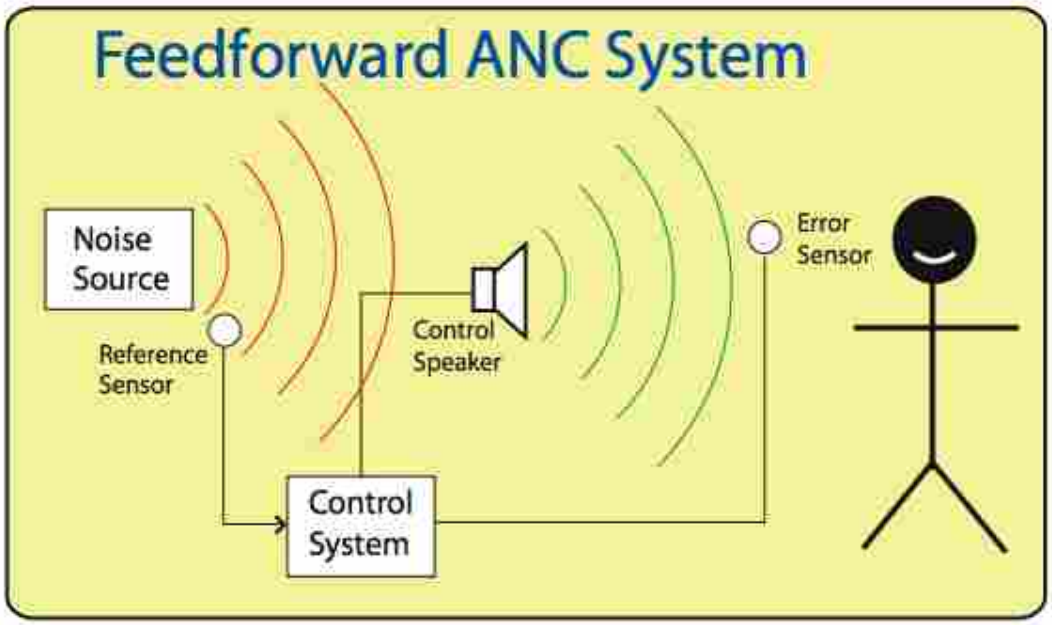

Figure 1.2 Basic setup of a feedforward ANC system

1.2 FXLMS Algorithm

One of the most popular feedforward control algorithms for acoustic noise is the Filtered-X Least Mean Square (FXLMS) algorithm[2] due to its robustness and relative ease of implementation. The FXLMS algorithm creates a control signal by filtering the reference signal with an adaptive control filter. The control filter is updated via a gradient descent search process until an ideal filter that minimizes the residual noise is found. In the FXLMS algorithm, the reference signal is filtered by an estimate of the secondary path transfer function (usually designated $\hat{H}$), which is the propagation path from the controller to the error sensor giving the filtered-x signal. This gives an estimate of the gradient for the search and ensures that the algorithm will be stable. Without this the controller would not be able to compensate for changes that happen to the control output between the controller and the error signal (specifically the phase change). Rather than cancelling the noise the controller may amplify it.

The convergence properties of the FXLMS algorithm have been well studied. They are dependent on the distribution of the eigenvalues of the filtered-x autocorrelation matrix [3]. It has been further shown that convergence and stability of the algorithm can be evaluated for individual frequency bins and are dependent on the power gain of the secondary path, the power spectrum of the reference signal, and the convergence parameter, or step size the algorithm takes with each iteration of the search for the optimum control filter. Since neither the power gain of the secondary path nor the power of the reference signal are, in general, uniform over frequency, convergence will be poor (slow) for some frequencies compared to others. The algorithm performance is degraded if the power gain of the secondary path is not flat over the entire frequency range targeted for control [4]. This is a drawback of the FXLMS algorithm for ANC applications where multiple tones need to be controlled simultaneously or where a single tone that sweeps through a range of frequencies is to be controlled. These two types of noise will be called "multiple tone noise" and "swept tone noise" respectively in this thesis. A derivation of the FXLMS algorithm and a detailed discussion of its convergence and stability properties will be given in Chapter 2.

1.3 Previous Work: Other Algorithms- Improvements for Frequency Dependent Convergence of FXLMS

Various adaptations to the FXLMS algorithm have been developed in an effort to overcome the performance loss due to its frequency dependent convergence behavior. One way to do this for swept tone noise is to make the convergence coefficient, μ, vary with frequency so that the step size is always optimal for the current operating frequency of the system. The normalized FXLMS[5] adjusts μ according to the power of the filtered-x

input signal. This requires real time estimates of the power of the input signal to be calculated. This is applicable to swept tone noise applications where a single tone is to be controlled since the value for μ can then be optimized as the frequency and hence the reference signal power changes. However it is not applicable for multiple tone noise as there will always be varying convergence when more than one tone is present.

The Higher Harmonic Least Mean Square (HLMS) algorithm[6] uses a separate reference signal and controller that is run in parallel for each tone being controlled. This essentially reduces the problem of controlling multiple tones with different convergence properties to multiple single tone ANC problems. This allows using a separate convergence parameter that is optimized for each individual frequency. More uniform convergence and increased overall attenuation of all tones is achieved at the expense of more computational complexity. Splitting the tonal components of a reference signal into individual references used by each controller requires additional signal processing and conditioning. The need for multiple controllers increases computational load and cost. Lee et al.[7] also developed a method to process tonal components of a multiple tone noise problem separately while addressing some of the increased computation issues. These apply to multiple tone noise.

Kuo et al.[8] discuss how the eigenvalue spread over frequency for a system is a function of the reference signal amplitude multiplied by the secondary path magnitude at a given frequency. Where the reference signal is internally generated the amplitude of the reference can vary with frequency to compensate for the gain modulation of the secondary path to give uniform power of the filtered-x signal at specified frequencies resulting in uniform eigenvalues for the system. This gives uniform convergence over

frequency with a single value for μ. This was demonstrated in computer simulations by optimizing the magnitude of internally generated reference signals as the inverse of the secondary path magnitude [9]. This approach requires that the user have control over the reference tone amplitudes. This type of modification applies to swept tone noise and multiple tone noise.

Other algorithms such as the FxGAL [10], ALE+FxLMS [11], and Modified FXLMS [12-14] show improved convergence properties over the FXLMS algorithm. The drawback of most of these approaches is that they increase the computational burden of the algorithm, increase the algorithm's complexity, or are not applicable to one of the two types of noise considered here.

Work by Thomas[15] showed that flattening the magnitude response of the secondary path estimate used to make the filtered-x signal gives more uniform eigenvalues, better tracking, and better overall performance for control of swept tone noise. In his work, the finite impulse response (FIR) filter representing the secondary path transfer function (implemented as a vector of coefficients modeling the response of a system to an impulse) is modified in the frequency domain to give a new estimate with flat magnitude response and phase equal to the original phase response of the secondary path. This method was termed the "Eigenvalue Equalization FXLMS algorithm" (EE-FXLMS). This method improved performance of the algorithm in a way that was simple to implement and did not increase the computational burden of the ANC system. The specific focus of his work was tractor noise and is applicable to any swept tone noise application.

The work in this thesis is an extension of Thomas's work, applying the idea of modifying the magnitude response of the secondary path estimate while preserving the phase, to a different type of noise problem—multiple tone noise. In Thomas's work the power of the reference signal was assumed to be independent of frequency. Here, no such assumption is made for the multitone reference signal. A genetic algorithm is used to find optimum magnitude coefficients for a modified FIR model of the secondary path for a given multi-tone reference. As before, the eigenvalue disparity is reduced and the performance of the algorithm is improved over the FXLMS algorithm in both rate of convergence and overall attenuation achieved. Issues arising from finite frequency resolution of sampled systems are addressed in the optimization.

1.4 Why Use Active Noise Control?

Many acoustic noise problems are dominated by low-frequency noise. Passive noise control techniques such as noise barriers and absorbers are ineffective at these low frequencies. Barriers must be made very heavy and absorbers very large to attenuate low-frequency noise. Active noise control (ANC) works best for low-frequency noise problems where the cost in terms of weight and bulk make passive noise control methods inadequate for several important applications[16].

ANC has been used successfully to attenuate noise in ducts, automobiles, earthmoving equipment [17], aircraft[18-19], and other applications [20].

ANC is of special interest to the aircraft industry. The high noise levels inside an aircraft cabin limit flight time due to noise exposure restrictions [21]. Passive control of

interior noise for aircraft has reached its limit under weight and cost considerations, but there are demands for higher performance (therefore lighter weight) aircraft with reduced interior noise levels [22]. ANC provides the potential to reduce noise levels below that which can be achieved using only passive means, or achieve reductions comparable to passive means with a much lower weight penalty, especially at low frequencies.

In helicopters, low frequency noise masks speech and makes communication difficult, even with the use of intercom systems. To compensate, pilots increase the volume in their headsets in order to hear others talking to them. Increased sound pressure levels (SPL) inside of headsets are required for intelligibility at speech frequencies. Exposure to these high SPLs can cause hearing loss [23]. An ANC system has the potential to alleviate these problems by reducing the low frequency noise, allowing flight crews to communicate effectively at lower headset SPLs.

1.5 Motivation For and Objectives of This Thesis

While many algorithms and control schemes that improve on certain drawbacks of the FXLMS algorithm have been presented in dozens if not hundreds of technical papers, there is still room for improvement. Specifically, multiple tone noise continues to be a difficult ANC problem. Helicopter noise is one such ANC application and was the original motivation for the need to improve the convergence properties of the FXLMS algorithm. An algorithm that improves the performance of an ANC system while keeping cost and complexity low will open new applications for ANC and improve the benefit of ANC in some applications where it is currently being used.

The primary goal then of this research effort is to develop a new way of implementing the EE-FXLMS algorithm and equalizing the eigenvalues of an ANC system for multiple tone noise such as is found in the interior of helicopter cabins. Methods of altering the magnitude of the secondary estimate to optimize an ANC system for control of multiple stationary sinusoids will be explored and their benefit will be demonstrated using computer simulations as well as experimental measurements in a mock cabin enclosure.

1.6 Organization of This Thesis

In Chapter 2, the FXLMS algorithm will be derived and its convergence properties and drawbacks for swept tone and multiple tone noise discussed in detail. This will include discussion of quadratic error surfaces, steepest descent search methods, eigenvalues, and time and frequency domain analysis of convergence of the error signal. Additionally, the structure of reference signals for multiple tone noise and its implication in applying the FXLMS algorithm to that type of noise will be discussed. Chapter 3 will introduce the EE-FXLMS algorithm. The method of equalizing the eigenvalues by altering the magnitude of the secondary path estimate will be explained as well as the genetic algorithm used for the optimization. The effect of the modification of the magnitude on the eigenvalue disparity will be presented. Chapter 4 gives experimental verification to the theory and results in Chapter 3. It is shown that the eigenvalue equalization performed in Chapter 3 gives better performance in ANC tests. In Chapter 5 conclusions based on the experimental results are made.

CHAPTER 2-FILTERED-X LEAST MEAN SQUARES (FXLMS) ALGORITHM

2.1 FXLMS Algorithm (FXLMS)

The FXLMS algorithm is shown in block diagram form in Figure 2.1. In the figure, and in all equations presented, the variable *t* is a discrete time index and the variable *z* is a discrete frequency-domain index. The discussion of the FXLMS algorithm and its properties given here follows closely the development of Widrow and Stearns[24].

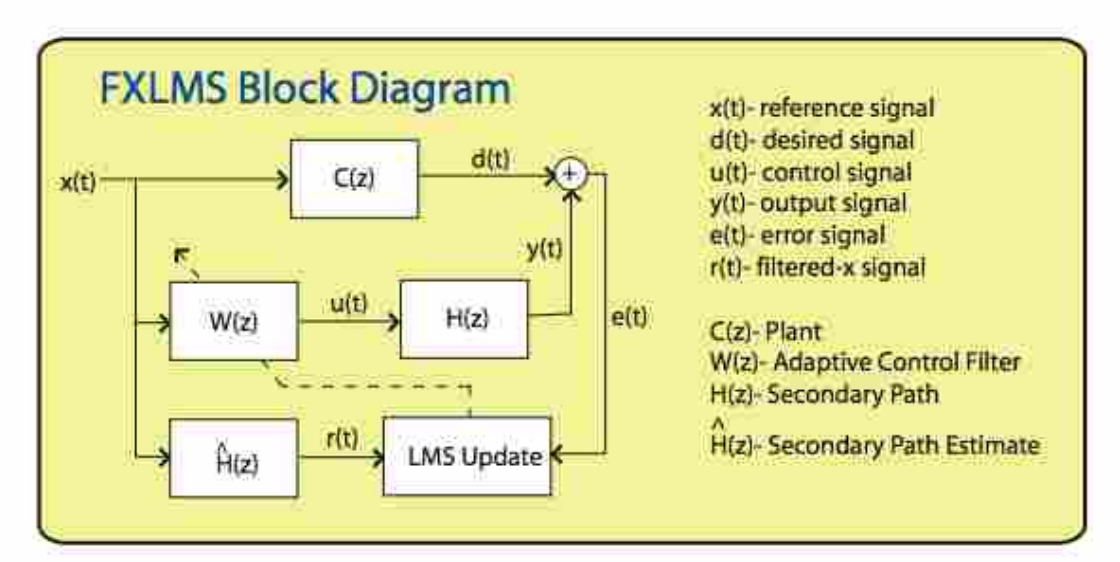

Figure 2.1 Block diagram of the filtered-x least mean square algorithm.

As before, the reference signal, x(t), is obtained that characterizes the unwanted noise. The noise measured at the location of the reference sensor propagates through the primary path, or plant, represented by **C**(z) and arrives at the listening location as the signal, d(t). This is the unwanted noise to be cancelled. It is sometimes referred to as the "desired" signal, meaning the signal that the controller is trying to duplicate (with opposite phase) and hence attenuate. Neither the transfer function, **C**(z), nor d(t) are known. All that is known is that d(t) will be correlated with the noise characterized by the

reference. The FXLMS algorithm creates a control signal by simply filtering the reference signal, x(t), with an adaptive FIR control filter, **w**(t) given by

$$\mathbf{w}(t) = [w_0, w_1, w_2, \ldots w_{L-1}]^T \tag{2.1}$$

where **W**(z) is the frequency response of **w**(t) given by

$$\mathbf{W}(z) = w_0 + w_1 z^{-1} + w_2 z^{-2} + \ldots + w_{L-1} z^{-L+1} \tag{2.2}$$

The control signal, u(t), is the convolution of x(t) with **w**(t). The control signal is filtered by the secondary path transfer function, **H**(z), and arrives at the error sensor as the output signal, y(t). The secondary path transfer function includes the effects of digital-to-analog and analog-to-digital converters, filters, audio power amplifiers, loudspeakers, the acoustical transmission path between the control source and error microphone, error sensor response, and other signal conditioning. The output signal combines with the unwanted noise to give the residual error signal, e(t), measured by the error sensor. An adaptive process searches for the optimal coefficients for the control filter, **w**(t), which will minimize the residual error. This optimal filter is designated **w***.

The mean-square error (MSE), expressed as

$$\xi(t) = E\{|e(t)|^2\}, \tag{2.3}$$

is a quadratic performance function of the filter coefficients with a unique global minimum. The operator E{} is the expectation value. For a control filter with L-coefficients, the error surface is a hyperparaboloid in L+1 dimensions with the coordinate axes corresponding to the filter weights $w_0, w_1, w_2, \ldots, w_{L-1}$. This is easily visualized for a 2-coefficient control filter for which the error surface is a paraboloid as shown in Figure 2.2. Cutting the paraboloid with planes parallel to the w_0w_1-plane we get ellipses

(hyperellipses for higher dimensions) as contours of equal mean-square error, also shown in Figure 2.2.

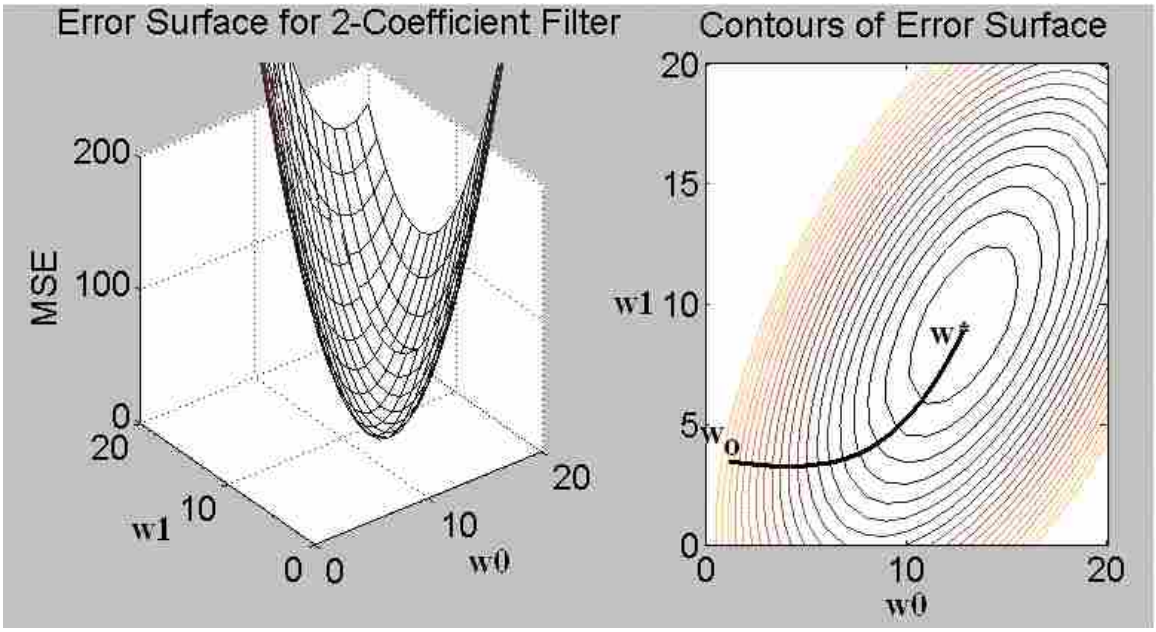

Figure 2.2 Three-dimensional quadratic error surface for the mean square error. The plot on the left shows the error surface as a paraboloid in 3-dimensions with MSE as a function of the two filter weights. The plot on the right shows the same surface as equal MSE contours. The path of steepest descent from some starting point $\mathbf{w_0}$ to the optimum w* is shown on the contour plot.

The FXLMS algorithm uses a gradient descent search method to adapt the control filter coefficients. The algorithm starts at some point, $\mathbf{w_0}$, on the error surface and follows the path of steepest descent (direction of the negative gradient) along the error surface toward the optimal filter weights at the bottom of the paraboloid. This is not, in general, the most direct path from the starting point to the optimum. This is shown on the contour plot of Figure 2.2. The control coefficients at each iteration of the algorithm then are given by

$$\mathbf{W}(t+1) = \mathbf{w}(t) + \mu\,(-\nabla\,\xi\,) \qquad (2.4)$$

The convergence coefficient, μ, is a step size that dictates how far in the direction of the negative gradient the algorithm can move with each iteration of the search for the set of filter coefficients that minimize $\xi(t)$.

The FXLMS algorithm uses the square of the error signal as an approximation of the mean square error. Minimizing the instantaneous squared error signal can be shown to converge in the mean to the solution obtained when minimizing the mean square error. The error signal e(t) is the desired signal, d(t), plus the output signal y(t). The output signal is the control signal u(t) filtered by $\mathbf{H}(z)$, and u(t) is x(t) filtered by the control filter $\mathbf{W}(z)$. These relationships are summarized showing both signals and filters (or transfer functions) in the frequency domain as:

$$\mathbf{E}(z) = \mathbf{D}(z) + \mathbf{Y}(z) = \mathbf{D}(z) + \mathbf{U}(z)\mathbf{H}(z) = \mathbf{D}(z) + \mathbf{X}(z)\mathbf{W}(z)\mathbf{H}(z) \tag{2.5a}$$

Because convolution (or multiplication in the frequency domain) is commutative

$$\mathbf{E}(z) = \mathbf{D}(z) + \mathbf{W}(z)\mathbf{X}(z)\mathbf{H}(z) \tag{2.5b}$$

and defining $\mathbf{R}(z) = \mathbf{X}(z)\mathbf{H}(z)$ we arrive at

$$\mathbf{E}(z) = \mathbf{D}(z) + \mathbf{W}(z)\mathbf{R}(z) \tag{2.5c}$$

In the time domain the instantaneous error signal can then be written as

$$e(t) = d(t) + \mathbf{w}(t)^T\mathbf{r}(t) \tag{2.6}$$

where $\mathbf{w}(t)$ is the vector containing the current set of filter coefficients and $\mathbf{r}(t)$ is a vector of previous time samples of the filtered-x signal the same size as the control filter vector. $\mathbf{W}(z)$ and $\mathbf{R}(z)$ in Eq. 2.5c are the frequency responses of these vectors. The instantaneous squared error can then be written as

$$\varepsilon(t) = e^2(t) = d^2(t) + 2d(t)\mathbf{w}^T\mathbf{r}(t) + \mathbf{w}^T\mathbf{r}(t)\mathbf{r}^T(t)\mathbf{w} \quad (2.7)$$

The gradient of the squared error can be expressed as

$$\frac{\partial e^2(t)}{\partial \mathbf{w}} = 2d(t)\mathbf{r}(t) + 2\mathbf{r}(t)\mathbf{r}^T(t)\mathbf{w}$$

$$= 2\mathbf{r}(t)\,[d(t) + \mathbf{r}^T(t)\mathbf{w}]$$

$$= 2e(t)\mathbf{r}(t) \quad (2.8)$$

The factor of two in Eq. 2.8 is sometimes omitted from this expression for the gradient since it can be included in the convergence parameter, but is left here.

The FXLMS algorithm derives its name from the filtered-x signal, $\mathbf{r}(t)$, which, along with the instantaneous error, gives an approximation of the gradient and hence, the direction of steepest descent on the error surface. The control filter update equation can then be written according to Eq. 2.4 as

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \mu 2e(t)\mathbf{r}(t), \quad (2.9)$$

which is called the least mean square (LMS) update.

The fact that the instantaneous error can be used in an estimate of the gradient makes the FXLMS algorithm simple and easy to implement. It is a noisy estimate, but statistically a good one. Because it is a noisy estimate, the FXLMS algorithm does not exactly follow the path of steepest descent but is erratic. Figure 2.3 shows this erratic path that follows the general trend of the path of steepest descent.

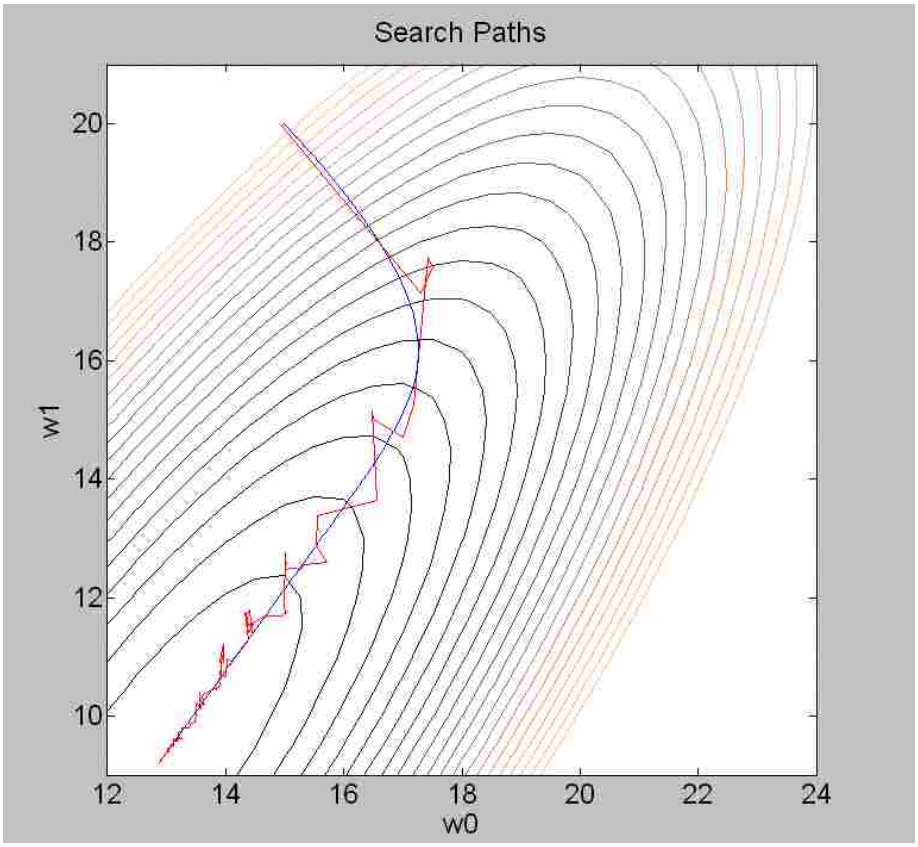

Figure 2.3 Search paths on the contour plot of the error surface. The blue line is true path of steepest descent, and the red line is the path the FXLMS algorithm follows due to the noisy gradient estimate.

2.2 Secondary path transfer function

The reference signal filtered by an FIR estimate of the secondary path transfer function gives the filtered-x signal used in the control filter update. $\mathbf{H}(z)$ is estimated through a process called system identification (Sys ID). For this research, the Sys ID process is performed offline (before ANC is started). This is the preferred method for applications where the secondary path does not change significantly during operation of the ANC system due to its relative simplicity. Band-limited white noise is played through the control speaker(s) and the output is measured at the error sensor. The measured impulse response is obtained as a FIR filter, $\hat{\mathbf{h}}(t)$, which represents $\mathbf{H}(z)$ in the time

domain. The coefficients of $\hat{\mathbf{h}}(t)$ are stored and used to pre-filter the reference signal and give the input signal to the LMS update.

The FXLMS algorithm is robust to errors in the estimation of $\mathbf{H}(z)$. As long as the poles of the transfer function $\mathbf{H}(z)$ are within the unit circle in the complex plane the algorithm will converge slowly. The distance from the origin to the poles is given by

$$d = [1 - \beta\cos(\Phi)]^{1/2} \tag{2.10}$$

where β is assumed to be a small positive number for slow convergence and Φ is the phase difference between the actual secondary path transfer function, $\mathbf{H}(z)$, and the secondary path estimate. The distance from the poles to the origin can only be greater than one if $\cos(\Phi)$ is negative. This puts a stability limit for phase errors of

$$\cos(\Phi) > 0, \tag{2.11}$$

or equivalently

$$-90^{o} < \Phi < 90^{o} \tag{2.12}$$

This means that the algorithm will converge (slowly) as long as phase errors in the secondary path estimate are less than $+/-90^{o}$ [2]. Convergence time of the algorithm increases as the inverse of $\cos(\Phi)$. Computer simulations show that phase errors less than 45^{o} do not significantly affect performance of the algorithm[25].

The gain applied to the reference signal by filtering it with $\hat{\mathbf{h}}(t)$ does not affect the stability of the algorithm and is usually compensated for by modifying the convergence parameter, μ.

2.3 FXLMS Convergence and Eigenvalues of Filtered-x Autocorrelation Matrix

The shape and orientation of the error surface, and hence the path of steepest descent the algorithm will follow toward the optimum filter, are a function of the filtered-x autocorrelation matrix defined as

$$\mathbf{R} = E\{\mathbf{r}(t)\mathbf{r}^T(t)\}, \tag{2.13}$$

where $E\{\}$ denotes the expected value of the operand which is the filtered-x signal vector, $\mathbf{r}(t)$, multiplied by the filtered-x signal vector transposed, $\mathbf{r}^T(t)$. It is a square matrix with each dimension equal to the number of coefficients in the control filter (LxL). Expressing **R** in normal form, in terms of eigenvalues and eigenvectors, is helpful in describing the error surface. The eigenvalues of **R** are associated with the homogeneous equation

$$[\mathbf{R} - \lambda\mathbf{I}]\mathbf{Q}_n = \mathbf{0} \tag{2.14}$$

where **I** is the identity matrix, $\mathbf{Q}_n$ is a column vector (eigenvector), and λ is the eigenvalue which is a scalar variable. The eigenvalues are the values of λ that render the bracketed matrix in Eq. 2.14 singular. These values are typically determined by setting the determinant of the bracketed matrix in Eq. 2.14 equal to zero and then solving for the eigenvalues, as shown by Eq. 2.15.

$$|[\mathbf{R} - \lambda\mathbf{I}]| = 0 \tag{2.15}$$

Since the autocorrelation matrix, **R**, is positive semidefinite, the L eigenvalues are all real and non-negative, though some of them may be zero. The rank of the autocorrelation matrix gives the number of nonzero eigenvalues. The number of nonzero eigenvalues is twice the number of tones in the reference signal. For each eigenvalue, λ_n, there is a corresponding eigenvector, $\mathbf{Q}_n$.

The eigenvectors of **R** define the principal axes of the error surface. In three dimensions, this corresponds to the major and minor axes of the elliptical mean-square error contours shown in Figure 2.2. The eigenvalues (multiplied by two) give the second derivative (curvature) of the error surface along the principal axes defined by their corresponding eigenvectors[26]. Convergence of the algorithm happens independently along each of the principal axes of the error surface and is governed by a unique geometric ratio. These ratios are

$$\begin{aligned} \gamma_0 &= 1-\mu 2\lambda_0 \\ \gamma_1 &= 1-\mu 2\lambda_1 \\ &\vdots \\ \gamma_{L-1} &= 1-\mu 2\lambda_{L-1} \end{aligned} \tag{2.16}$$

Movement along a principal axis for which the curvature is zero makes no difference in the mean-square error and hence eigenvalues that are zero are not significant to the convergence of the algorithm toward the optimum. The difference between the projections of **w** and **w*** onto the principal axes of the error surface decreases by these geometric ratios with each iteration of the algorithm (if the true gradient is known at each iteration). It is apparent then that for the algorithm to converge to the optimal solution and remain stable, $|\gamma| < 1$. This puts the following stability limits on the convergence parameter:

$$0 < \mu < \frac{1}{\lambda_{max}} \tag{2.17}$$

where λ_{max} is the maximum eigenvalue of the autocorrelation matrix. The larger the curvature in any dimension the smaller the step size needs to be to avoid overshooting the optimum and causing the algorithm to diverge.

In practice, it is computationally demanding to obtain a real-time estimate of the autocorrelation matrix, so the optimal μ is often selected through experimentation. In this work, the structure of the eigenvalues of a given ANC problem is explored using an offline estimate of the autocorrelation matrix. This is done in a numerical analysis program by taking an actual $\hat{\mathbf{h}}(t)$ model from a mock cabin enclosure, convolving this with a reference signal for the given noise application, computing the autocorrelation matrix, and getting the eigenvalues by solving Eq. 2.15. The "eig" function in Matlab was used to compute these eigenvalues from the autocorrelation matrix, **R**.

2.4 Limitations of the FXLMS—Eigenvalue Disparity

One of the limitations of the FXLMS algorithm is that it exhibits frequency-dependent convergence behavior that can lead to a significant degradation in the overall performance of the control system. Two types of noise will be discussed as they relate to this limitation:

(1) A single tone with time-varying frequency, such as engine noise, where the engine firing frequency changes along with the speed of the engine in revolutions per minute (rpm) during operation. It is assumed that the signal power of the tone in the reference remains the same, independent of frequency. This type of noise will be referred to as "swept tone noise".

(2) Noise containing multiple quasi-stationary tones, such as helicopter cabin noise, where multiple rotating parts contribute strong tones that do not vary significantly in frequency during normal operation. This type of noise will be referred to as "multiple tone noise".

Swept tone noise is treated by Thomas[15] and though not a main focus of this thesis, a discussion of it here provides a useful background in explaining multiple tone noise.

For a single tone input there will be two nonzero eigenvalues of the input autocorrelation matrix. The eigenvalues will change as the input changes in frequency since the gain applied by filtering the input with the secondary path estimate is frequency dependent, and/or the amplitude of the reference changes with frequency. If we assume that the power of the reference signal is constant, the gain (which varies with frequency) applied to the reference signal in filtering it with $\hat{\mathbf{h}}(t)$ makes the power of the filtered-x signal vary with frequency. This makes the eigenvalues of the input autocorrelation matrix and the optimal step size vary with the frequency of the tone in the noise.

If a monofrequency reference signal is used, λ_{max} can be computed for that frequency. If the simulation is repeated over a range of frequencies, λ_{max} for a single tone at each frequency in that range can be found. For control of a single tone, λ_{max} is the only eigenvalue of interest since it will determine the convergence of the algorithm for that frequency. Figure 2.4 shows an offline simulation of the maximum eigenvalues over frequency using an actual $\hat{\mathbf{h}}(t)$ from a mock cabin enclosure, and equal amplitude tonal inputs from 0-300 Hz along with a plot of the magnitude of the secondary path estimate. The maximum eigenvalue at each frequency is plotted in red and the magnitude of the secondary path estimate is plotted in blue. Both curves in the figure have been normalized to the largest value in the range. The maximum eigenvalues at each frequency follow the trend in the magnitude response of H(z). The disparity in λ_{max} over frequency shows how the convergence of the algorithm will change as it controls a single tone swept through this range.

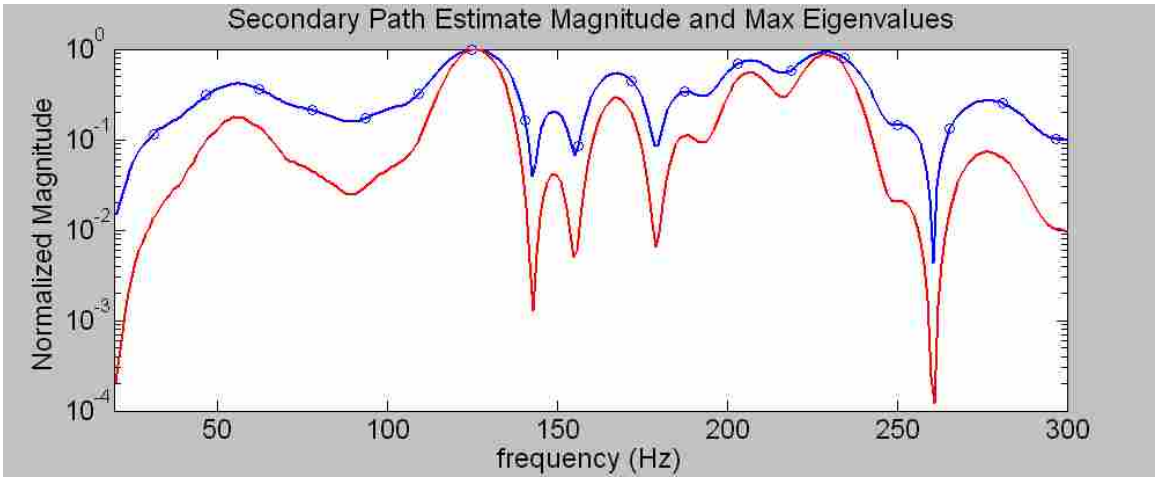

Figure 2.4 Maximum eigenvalues for single tone inputs at each frequency (red) and secondary path estimate magnitude (blue).

The largest eigenvalue for a single tone occurs at about 125 Hz. This location corresponds to the largest μ that is stable for the entire frequency range from 0-400 Hz as given by Equation 2.17 All other frequencies have a smaller maximum eigenvalue and could use a larger μ, and still be stable, if just that particular frequency was targeted for control. Frequencies at the valleys of Figure 2.4 have the smallest eigenvalues and could use the largest μ's and still be stable, again if they were the only frequencies targeted for control. The larger μ's are especially desirable for non-stationary noise as they lead to faster convergence and increased attenuation. When the tone shifts in frequency faster than the controller can reduce the error to its steady state level, overall attenuation will be increased for larger values of μ.

Since a single μ is used by the algorithm, the largest of the maximum eigenvalues for the range of frequencies must determine the maximum μ that can be used while maintaining system stability for the entire range. The convergence parameter is then optimized for one frequency (that corresponding to the largest eigenvalue) in the range of the sweep, but not for all frequencies in the range. When the algorithm is controlling a

tone in a frequency range other than that frequency for which μ was optimized, convergence will be slower. Tracking and convergence at some frequencies will be slow leading to reduced performance of the control system.

There is a similar limitation for noise with multiple quasi-stationary tones. For multiple tone noise, the eigenvalues are not computed for individual tones as before, but for the composite reference signal containing all tones to be controlled. In this case, the disparity among all of the nonzero eigenvalues, not just λ_{max} at each frequency, gives information about how the algorithm will converge. The number of nonzero eigenvalues is the same as the rank of the filtered-x autocorrelation matrix, or twice the number of tones in the reference. Figure 2.5 shows the composite eigenvalues for a multiple tone reference signal (two per tone) compared to the maximum eigenvalue over frequency computed for single tone inputs as in Figure 2.4. The eigenvalues turn out to be slightly different when calculated for single tone inputs and for a composite reference with all tones combined, though the latter closely follow the trends in the eigenvalues computed for single tone inputs. Circles mark the two eigenvalues per tone computed from the composite reference signal for each tonal frequency. The two circles are close enough to each other on the logarithmic ordinate that they appear as one.

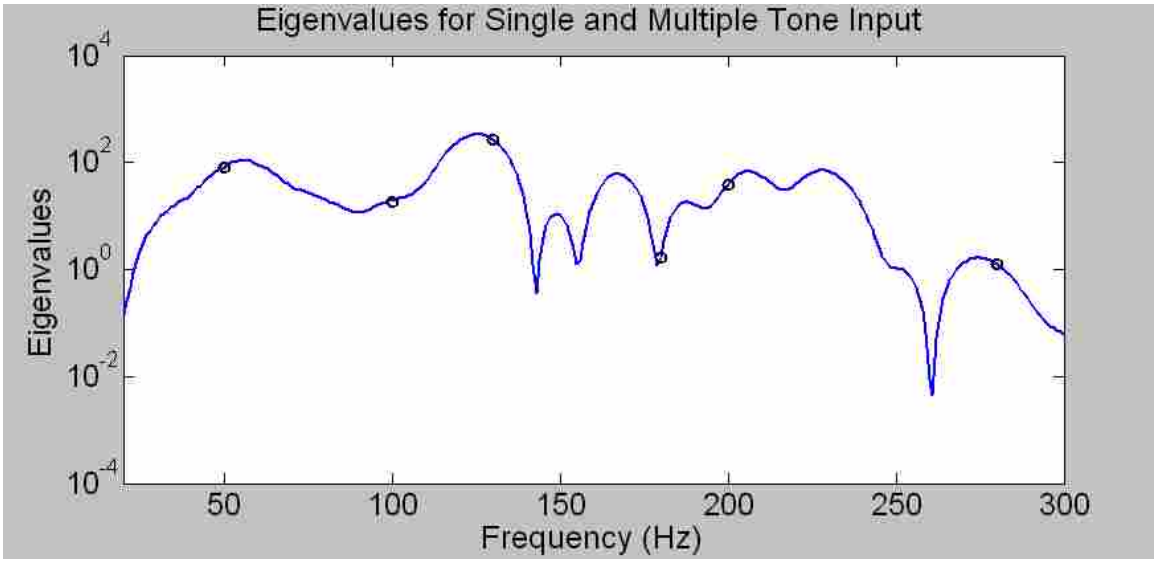

Figure 2.5 Maximum eigenvalues calculated for single tone inputs at each frequency in the range 20-300 Hz (blue line) and all 12 nonzero eigenvalues (black circles) for composite reference with six tones at frequencies marked by black circles.

If there is noise present in the reference signal the autocorrelation matrix may become full rank; however the number of significant eigenvalues remains twice the number of tones in the reference. All other eigenvalues will be very nearly zero. The greater the disparity in the nonzero eigenvalues the more eccentric the contours of the error surface become. Again, μ must be chosen based on the largest of the nonzero eigenvalues to keep the algorithm from diverging. The convergence parameter can be optimized within stability limits set by λ_{max} of the nonzero eigenvalues to give fast (optimal) convergence along one of the principal axes (the one with the largest eigenvalue), while convergence along others will be slower according to Eq. 2.16, since the step size used will be significantly smaller than the optimum dictated by the curvature for that axis. Variable convergence along different principal axes means that tonal components of the noise will converge at different rates. The algorithm will be able to

attenuate portions of the total noise quickly while other tones in the noise will linger and take longer to converge making the overall error signal converge more slowly.

2.5 Excess Mean-Square Error (MSE) and Misadjustment

In addition to the convergence rate of tonal components of the error signal, "misadjustment" errors that prevent the algorithm from converging to the true optimal solution depend on the eigenvalues of the autocorrelation matrix.

The FXLMS algorithm uses the method of steepest descent to arrive at the optimum filter coefficients. As the algorithm converges and **w** approaches **w*** the gradient approaches zero. This means that if the true gradient was known at each iteration the algorithm would stop when it reached the optimum since according to Eq. 2.4 each subsequent set of filter coefficients would be identical to the previous ones and a steady state solution would be reached. Since a noisy gradient estimate is used, even at or very near the optimum the gradient will not be zero and the algorithm continues to search. This causes the algorithm to randomly vary around the optimum, sometimes moving away from the optimum and increasing the mean square error of the error signal. Figure 2.6 shows the random searching process near the optimum.

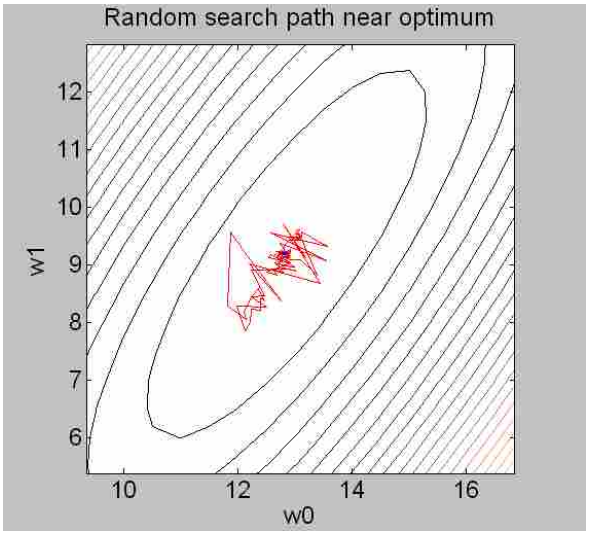

Figure 2.6 Random searching of FXLMS algorithm near the optimum due to gradient search noise.

The increase in mean square error due to this random searching near the optimum is known as misadjustment error. Increasing μ increases the misadjustment error because the algorithm will randomly step farther from the optimum, "climbing" higher up the sides of the hyperparaboloid where the MSE is larger. Decreasing μ improves misadjustment, but slows down convergence of the error signal. Generally μ is chosen to compromise between fast convergence and acceptable misadjustment error. Misadjustment error also increases as the disparity in the eigenvalues increases[27].

2.6 Multiple Tone Noise and Reference Signals

A feedforward implementation of ANC requires that a reference signal be obtained from the noise source(s). The control signal is derived from this reference signal through a filtering process which makes adjustments to the magnitude and phase of tonal components of the reference signal; there is no signal generation performed by the controller or control filter, only filtering. Attenuation can only be achieved for frequency content that is present in the reference signal. If only the fundamental frequency of a

disturbance is in the reference signal only the fundamental can be controlled and not the harmonics. To control harmonics, they must either be generated from the fundamental or the reference signal must be obtained in such a way that the harmonics are also present. Thus, the more closely the frequency content of the reference signal matches that in the noise, the more attenuation is possible.

In some cases, an acoustic reference signal is obtained by directly sampling the acoustic field near the noise source. Because it is directly sampled from the sound field, this type of reference will, in general, have all of the frequency content necessary to cancel all of the unwanted noise. In practice, however, an acoustic reference can be problematic. Open feedback paths from the control signal can lead to corruption of the reference signal, poor performance, and algorithm instability [28].

These problems can be circumvented for periodic noise by using a non-acoustic reference signal. There are various ways to do this. For rotating machinery, an optical sensor or a Hall (magnetic) sensor may be used to obtain a reference signal. For noise associated with the engine firing frequency, a reference signal can be obtained either from a rotating part of the engine (and multiplied by the appropriate ratio to get the firing frequency) or directly from a magneto. In any case, the reference signal would likely be some sort of periodic non-sinusoidal wave— something resembling a square pulse or sawtooth wave or something in between. Rarely would the reference signal be a pure sinusoid at the fundamental frequency without first conditioning the signal. This is actually advantageous. These types of signals contain not only the fundamental, but also higher harmonics which will, in general, be present in the noise as well and can then be controlled. The relative weights of the harmonic components depend on the shape of the

reference signal; typically the fundamental will be the strongest with the harmonics decreasing as frequency increases. There may be some periodicity to the weights of the harmonic components.

The shape of the spectrum for a square pulse depends on the length of the duty cycle. For example, a perfect square wave (50% duty cycle) will have only odd harmonics that decrease as frequency increases. However, tachometers rarely produce a perfectly symmetric square pulse and even harmonics will also be present. A shorter duty cycle will in general give a flatter spectrum with significant harmonics higher in frequency. Figure 2.7 shows time waveforms and spectra typical of these types of tachometer signals. Figure 2.8 shows an actual unprocessed reference signal containing tonal components from the engine of a Robinson R44 helicopter. This reference signal was taken from the magneto and thus had the engine firing frequency and harmonics.

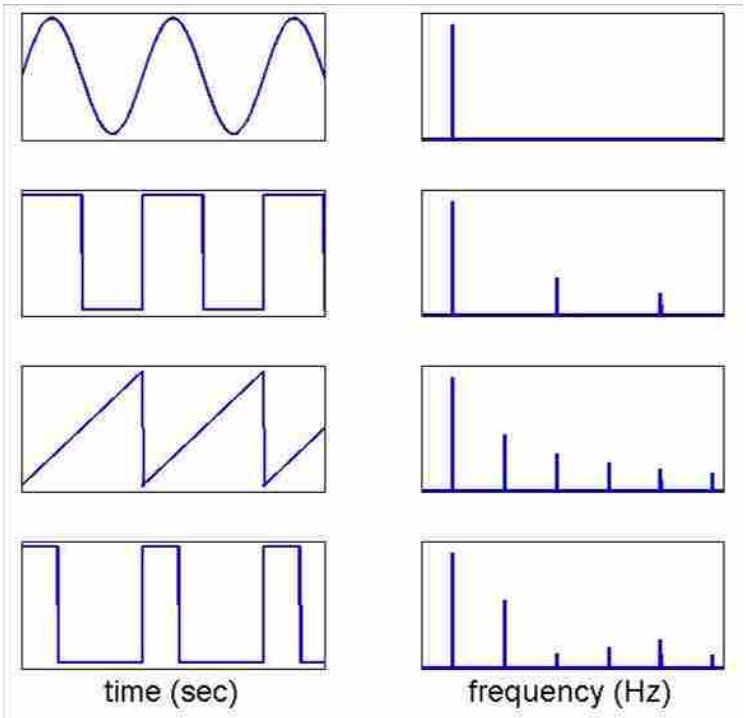

Figure 2.7 Waveforms (left column) and spectra (right column) for signals generated by typical non acoustic reference sensors. All plots in the figure are linear amplitude.

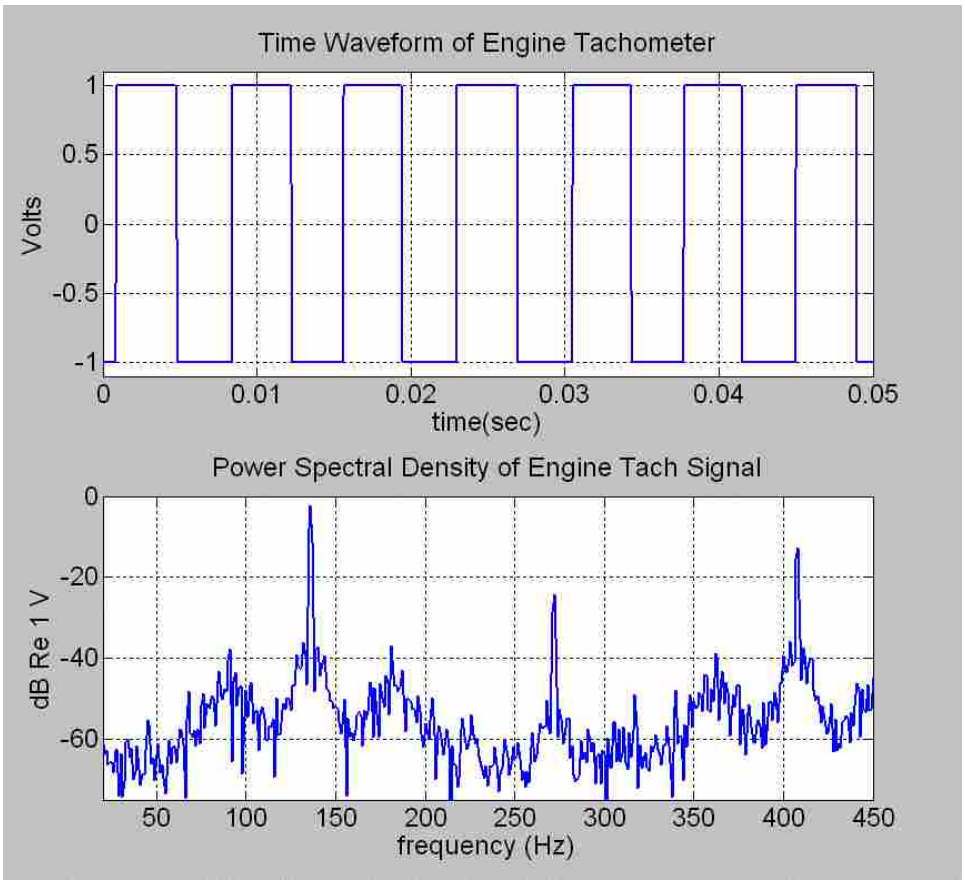

Figure 2.8 Tachometer signal from engine magneto on a Robinson R44 helicopter, giving engine firing frequency fundamental (136 Hz) and harmonics (272 and 408 Hz).

Several methods have been proposed to generate a reference signal with the appropriate frequency content and relative tonal amplitudes from these various types of tachometer signals. These are outlined in block diagram form in Figure 2.9. In waveform synthesis[29-30], the tachometer signal is only used to get an estimate of the fundamental frequency through FFT or autocorrelation methods. From this information a pure sinusoidal reference can be generated that has the same frequency as the tachometer. Harmonics can also be generated and added to the fundamental to get a multi-tonal reference signal. The advantage to this method is that the user can control the number of harmonics generated and their amplitudes. The difficulty in practice of waveform

synthesis is that high sampling frequencies and long time records are required to get high enough frequency resolution to get a good estimate of the fundamental frequency.

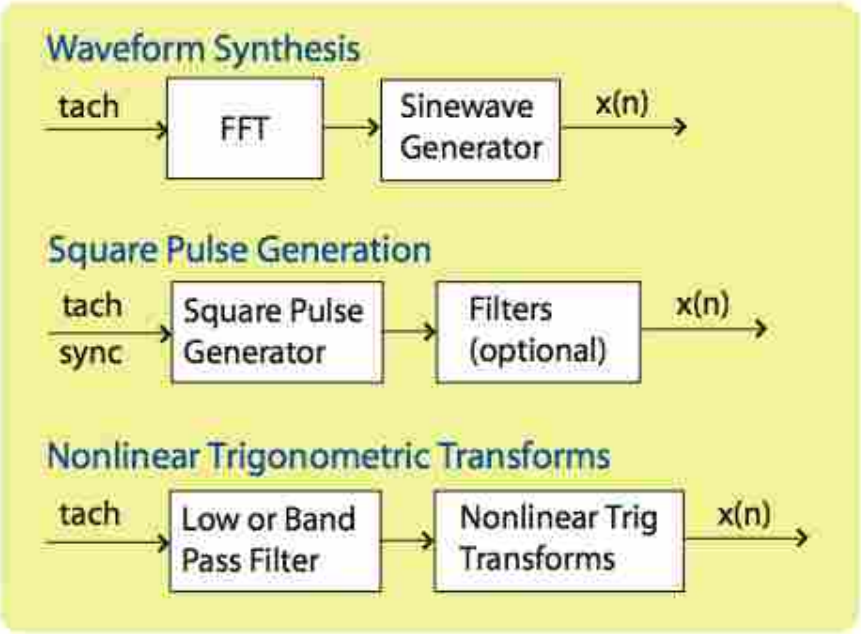

Figure 2.9 Block diagrams of different reference generation schemes.

Alternatively, the tachometer signal can be used to synchronize a square pulse generator. The user specifies the length of the duty cycle and hence has some control over how many harmonics are present and their relative amplitudes. Often a low pass filter is used in conjunction with the square pulse generator where harmonics higher than those desired in the reference are present. This method does not require long sample records since no frequency domain analysis is necessary. With significant additional filtering, the amplitudes of the tones present in the resulting reference signal can be set to desired values.

One other method of generating a multitone reference is using nonlinear trigonometric transforms. To do this the tachometer signal is first processed to get a sinusoid at the fundamental frequency. This can be done as described before with a sine

wave generator or using low-pass or band-pass filters to isolate the fundamental from the tachometer signal. Harmonics can be generated by performing nonlinear trigonometric transforms on this fundamental signal (Table 2.1). The user can define the amplitudes of the harmonics (A_2, A_3, … , A_N) in relation to the fundamental with this method.

Table 2.1 Nonlinear trigonometric transform equations

$$A_2\cos(2x) = A_2[2\cos^2(x) - 1]$$
$$A_3\cos(3x) = A_3[3\sin(x) - 4\sin^3(x)]$$
$$A_4\cos(4x) = A_4[1 - 8\cos^2(x) + 8\cos^4(x)]$$

When several noise sources are present, reference signals from each containing a fundamental and harmonics can be obtained and treated individually by separate controllers each creating a control signal that cancels the noise contribution from its source. The control outputs can be combined and use the same control sources, though this does require multiple control filters. Alternatively, the reference signals from each noise source can also be combined to produce a single reference signal containing tonal contributions from all noise sources and sent to a single control filter. This is simpler in that a single control filter can be used; however, combining multiple reference signals into a single composite reference signal further complicates the trends in tonal weights over frequency. As an example of such a case, a helicopter has three main noise sources: the engine, main rotor and tail rotor. Figure 2.10 shows what a combined reference signal from all three sources might look like. Figure 2.11 shows a power spectrum of the interior cabin noise measured at the pilot's position for the Robinson R44 helicopter. Prominent tones are marked with colors that indicate the source of that tonal component in the noise.

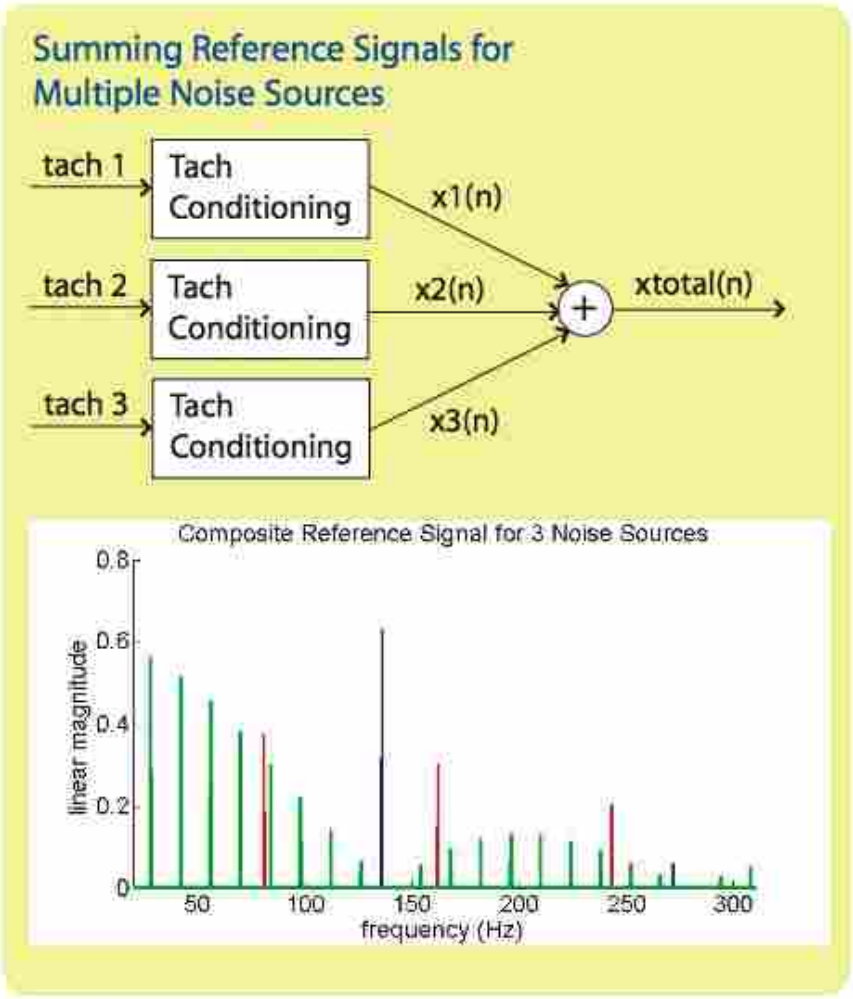

Figure 2.10 Composite reference signal for three major noise sources (engine, main rotor and tail rotor) on a Robinson R44 helicopter. Blue lines are the engine tones, red are tail rotor tones, and green are the main rotor tones.

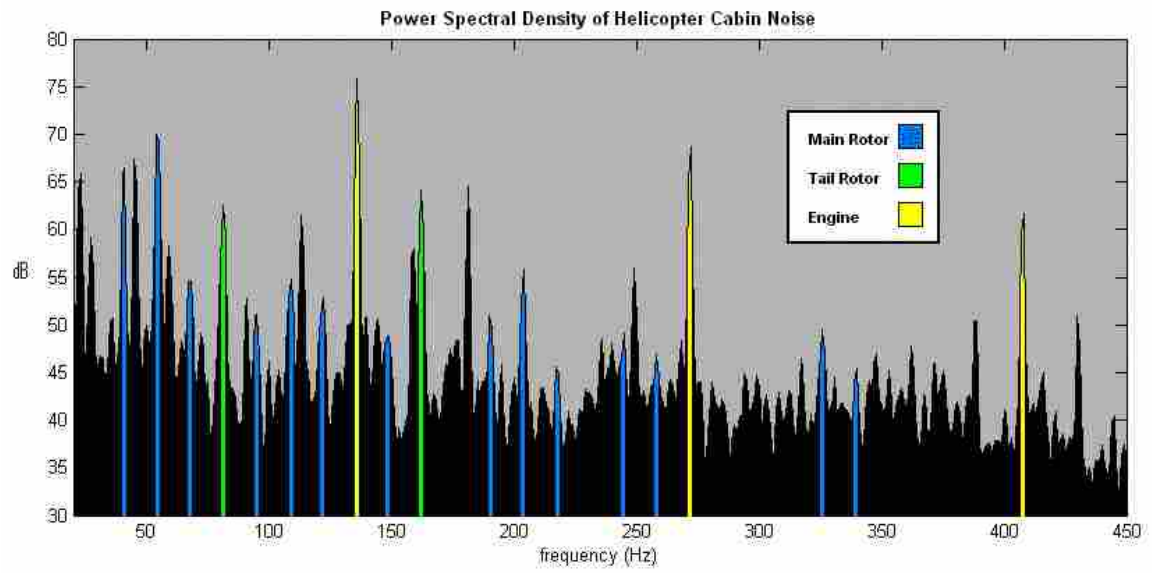

Figure 2.11 Interior cabin noise at pilot's position with noise source of prominent peaks identified

In summary, for feedforward ANC to be effective, a good reference signal that properly characterizes the noise to be cancelled is important. Non-acoustic references from tachometer signals allow for simpler and more robust implementation of the FXLMS algorithm and generally require some conditioning. Unless significant conditioning of the signal is done, multi-tone reference signals have non-uniform tonal amplitudes over frequency. When these types of reference signals are used, disparity in the eigenvalues comes not only from the gain of the secondary path estimate, but also from the non-uniform amplitudes of tones in the reference.

CHAPTER 3- EIGENVALUE EQUALIZATION FILTERED-X LEAST MEAN SQUARE (EE-FXLMS) ALGORITHM

3.1 Equalization of Eigenvalues

In Section 2.3, the frequency dependent convergence behavior of the FXLMS algorithm was explained by looking at the eigenvalues of the filtered-x autocorrelation matrix. The largest eigenvalue dictated the upper stability limit for the convergence parameter. The frequency corresponding to the smallest eigenvalue(s) will converge at the slowest rate of all the tones for that constrained value of μ. This slowest rate will determine the overall convergence rate of the error signal.

It has been shown that the FXLMS algorithm reaches a limit for maximum efficiency when all of the eigenvalues are equal. This corresponds to an error surface whose mean-square error contours are circular since the curvature is the same along any of the principal axes (see Figure 3.1). On this type of error surface the path of steepest descent is a straight line from any point on the surface to the optimum[31]. This is the shortest path to the optimum. Furthermore the geometric ratios governing convergence rates along the principal axes will all be equal and convergence at all frequencies will be uniform.

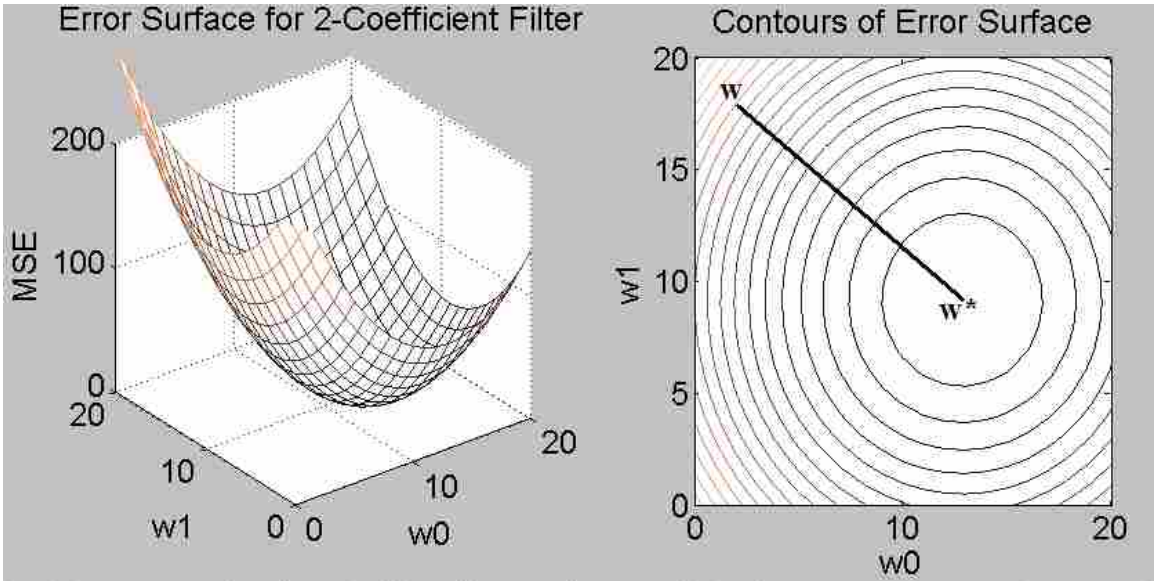

Figure 3.1 Normalized error surface with circular contours. The path of steepest descent (black line) is a straight line from the starting point, w, to the optimum, w*.

If the variance in the eigenvalues of the autocorrelation matrix is minimized, a single convergence parameter could then be chosen that would be nearly optimal for all frequencies targeted for control and the algorithm would converge at nearly the same rate at all frequencies or for all modes of convergence. Additional misadjustment error due to eigenvalue disparity will also be reduced. The primary goal of this research then is to introduce a method to equalize the eigenvalues of the input autocorrelation matrix. The resulting algorithm is called the Eigenvalue Equalization FXLMS (EE-FXLMS) algorithm. This algorithm was previously introduced for single tone noise and shown to improve convergence and tracking for single tone noise[15]. The current work extends this idea by further applying eigenvalue equalization techniques to multiple tone noise.

The eigenvalue span, defined as λ_{max} divided by λ_{min}, is used as a metric to quantify any improvement in the eigenvalue disparity. This ratio is the most important property, as any change in the actual magnitude of the eigenvalues is compensated for by making a complementary adjustment to the magnitude of the convergence parameter, μ.

3.2 Methods of Equalization

The autocorrelation matrix is directly dependent on the filtered-x signal, which is computed by filtering the input reference signal, x(t), with $\hat{\mathbf{H}}(z)$. Thus, any attempt at equalizing the eigenvalues must be done by altering either the reference signal or the secondary path model. Adjusting the power of the reference signal has been shown to be an effective way of doing this[8-9]; however, in many applications this amount of control over the reference signal is not feasible. We focus on making changes to $\hat{\mathbf{H}}(z)$ only, using the reference taken more or less directly from the tachometer. The reference is minimally conditioned so that all desired frequency content is represented, though the amplitude of each tone is arbitrary. For this research we assume that the reference contains all necessary frequency content to control tones present in the noise, but that the user has no control over the amplitudes of the tones in the reference.

For the EE-FXLMS algorithm, the FIR estimate of the secondary path is replaced with a new FIR estimate that has the same phase response as the original estimate with a different magnitude or gain response. The magnitude response of the new FIR estimate is the inverse of the amplitude of tones in the reference. This is intended to equalize the power of tonal components in the filtered-x signal, which in turn would equalize the eigenvalues of the filtered-x autocorrelation matrix. These adjustments to the secondary path estimate are made in the frequency domain. This change means that the $\hat{\mathbf{H}}(z)$ used to give the filtered-x signal is no longer a good estimate of the true secondary path. The errors introduced to the magnitude of the secondary path estimate only affect the eigenvalues and hence the rates of convergence of the algorithm, but do not affect

stability. This is all done offline as part of the set up of the ANC system and does not increase the computational load on the ANC system during operation.

First, the FFT of the FIR filter obtained in the System Identification (Sys ID) process is separated into magnitude and phase. The phase of the original secondary path transfer function estimate is preserved while the magnitude coefficients are adjusted to have the inverse trend of tones in the reference signal. The new magnitude coefficients are combined with the original phase response and transformed back into the time domain via an inverse FFT, giving a new FIR filter estimate of the secondary path.

3.2.1 Eigenvalue equalization applied to swept tone noise

For swept tone noise, the amplitude of the tone in the reference is assumed to be independent of frequency and hence the magnitude response of the new FIR filter estimate is made to be flat. To accomplish this, the secondary path estimate is made via the offline process described in Section 2.2. The coefficients of the FIR filter representing $\mathbf{H}(z)$ are transformed into the frequency domain using the Fast Fourier Transform. The complex coefficients are divided by their magnitude and multiplied by the mean magnitude over all frequency. These new coefficients are inverse Fourier transformed back into the time domain to get a new FIR filter with flat magnitude response (at data points) and a phase response equal to the original (at data points).

Figure 3.2 shows the original eigenvalues and the modified eigenvalues when the magnitude coefficients of $\hat{\mathbf{H}}(z)$ are flattened. In the figure, the eigenvalues for both the original and modified cases have been normalized by the largest of the maximum eigenvalues in the range. The span for the original eigenvalues in this range (0-300 Hz on

the plot) is 1.385 x 10^5 and the span for the flattened magnitude $\hat{\mathbf{H}}(z)$ is 162.3. These modifications to $\hat{\mathbf{H}}(z)$ make a noticeable improvement in the performance of the algorithm. The more uniform rate of convergence of all modes of the system is beneficial as it speeds up the overall convergence of the error signal. For dynamic signals, this increased rate of convergence often equates to greater attenuation, as it also results in more rapid tracking.

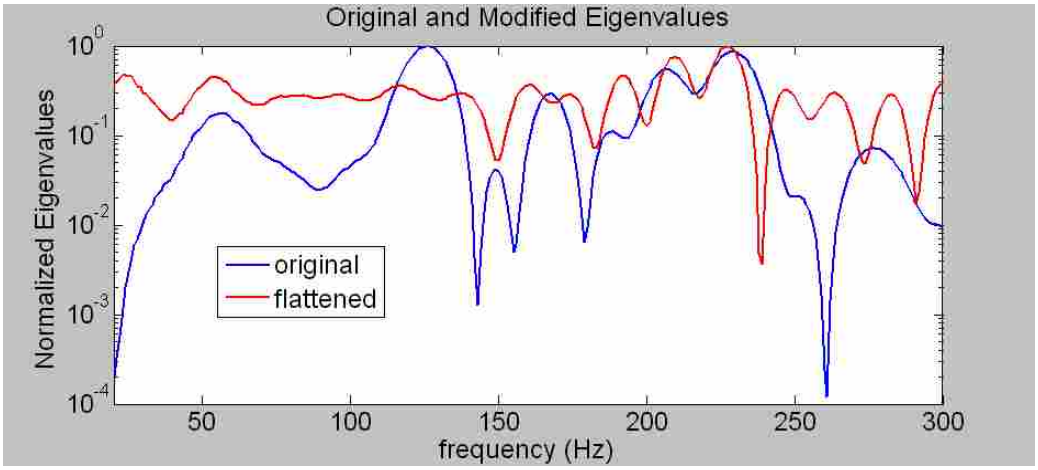

Figure 3.2 Normalized original and modified eigenvalues by frequency for single tonal inputs.

The eigenvalues are much more uniform, but still not perfectly uniform. This is due to the finite resolution of the digital system and of the sampled secondary path estimate. The shape of the magnitude response, $\hat{\mathbf{H}}(z)$, can only be constrained to a given value at its respective frequency bins; there is no guarantee that the response of $\hat{\mathbf{H}}(z)$ is also flat between frequency bins. This variation between bins is not random but cannot be predicted. The same is true for the phase response of the filter. As an example, a 128 coefficient $\hat{\mathbf{H}}(z)$ model sampled at 2000 Hz will have a frequency resolution of 15.625 Hz. For swept tone noise, the system may be excited at any frequency in the range of the application.

An estimate of the "analog" or continuous response of $\hat{\mathbf{H}}(z)$ between frequency bins can be made by zero padding the 128-coefficient model before computing the Fast Fourier Transform (FFT). The 128-coefficient frequency response (circles) and the zero padded frequency response (solid line) for the original (blue), and flattened (red) magnitude coefficients of $\hat{\mathbf{H}}(z)$ from a mock cabin are shown in Figure 3.3.

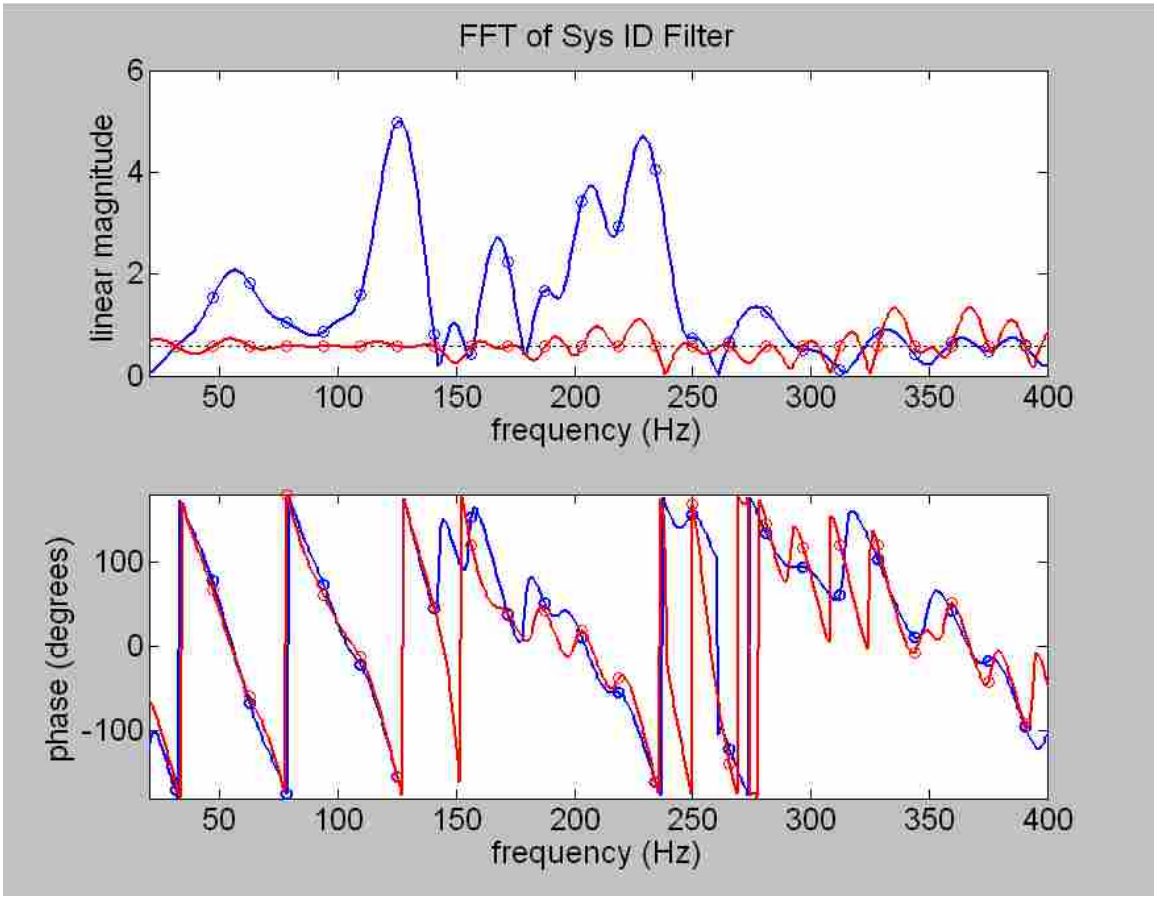

Figure 3.3 Plot of the 128-coefficient frequency response (circles) and the zero padded frequency response (solid line) for the original (blue), and flattened (red) magnitude coefficients of $\hat{\mathbf{H}}(z)$ from a mock cabin. The dashed line shows that drawing a line through the 128-coefficient frequency response points gives a flat line.

The length of the zero padded filter was 2048 coefficients giving a frequency resolution of 0.98 Hz. The discrete magnitude response is indeed flat at the frequency bin values, but the zero-padded model shows that the true response deviates from flat in

between bins. This magnitude variation between frequency bins in $\hat{\mathbf{H}}(z)$ contributes to the residual variation seen in the modified eigenvalues for the range.

Another source of variation may come from frequency leakage when the reference signal gets down-sampled before being convolved with $\hat{\mathbf{h}}(t)$. Before being convolved with $\hat{\mathbf{h}}(t)$, the reference signal is down-sampled with the same sampling frequency as was used to find $\hat{\mathbf{h}}(t)$; for this example, 2000 Hz was used. In addition, only *n* number of samples of the reference signal are kept at a given time, where *n* is the number of coefficients in $\hat{\mathbf{h}}(t)$; for this example, 128 was used. This down-sampling process causes amplitude estimation error in the frequency domain due to leakage. Thus if the original reference signal is assumed to be equally weighted at each frequency, as was done to create the eigenvalue simulations shown in Figures 3.2 and 3.3, the actual reference signal used in those simulations is no longer equally weighted over frequency. This also contributes to residual variation in the eigenvalues.

Similarly, Figure 3.3 also shows the original, flattened, and zero padded flattened phase response of $\hat{\mathbf{H}}(z)$ from a mock cabin. The modified phase response matches the original phase response at bin values, but deviates from it between bins. Figure 3.4 plots the difference between the original and modified phase response. If flattening the magnitude response of the secondary path estimate introduces errors in excess of 45º in the range, performance will be significantly compromised, and errors exceeding 90º will make the algorithm have instability problems at those frequencies[2,25]. A slightly more conservative limit of 40º phase error is marked by a green line and the 90 degree limit for stability is marked by red lines on the plot.

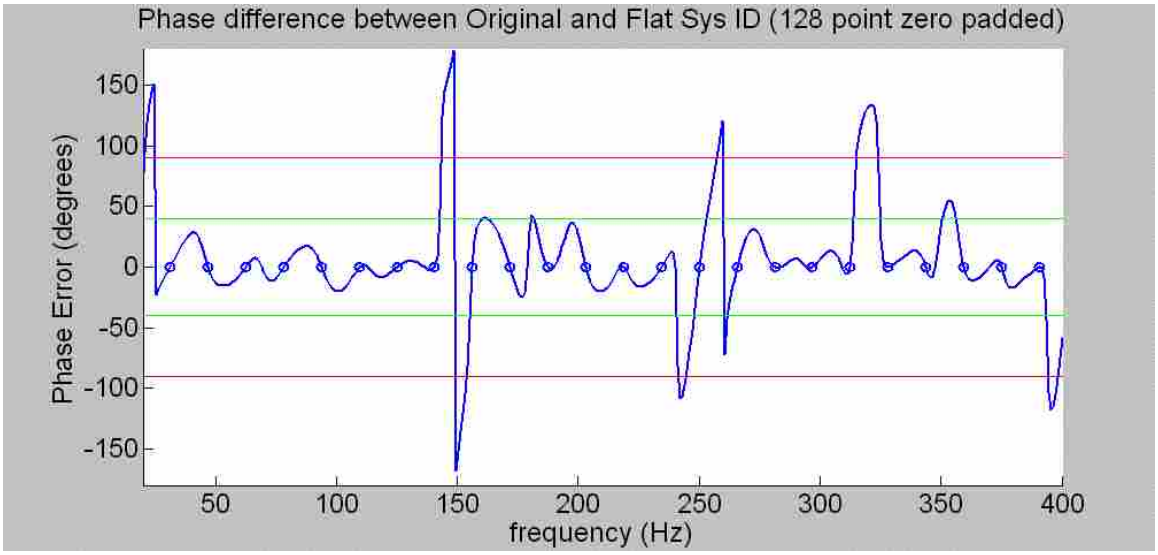

Figure 3.4 Phase errors introduced by modifying the magnitude response of the secondary path estimate.

In some places the error between the zero padded responses of the original and modified phase response exceeds 40º and 90º. While the errors in the original model of the secondary path are not known, it is assumed that it is a better estimate than the modified secondary path model.

3.2.2 Eigenvalue equalization applied to multiple tone noise

When multiple noise sources are present, a reference signal may be obtained from each and combined into a single reference signal. In some cases, the reference signal will contain a fundamental frequency and harmonics from a single noise source. In either case, the combined tones in the reference signal will in general have different amplitudes unless conditioned as discussed in Section 2.6. This weighting of the reference tones will be specific to each application and depends on how the reference signals are conditioned and combined. This frequency-dependent weighting of the reference tones, as well as the gain applied by the secondary path estimate contribute to the eigenvalue disparity for multiple tone noise and simply flattening the magnitude may not necessarily reduce

eigenvalue disparity. For multiple tone noise in this research, an arbitrary (but specific, for consistency) weighting was applied to the tones in the reference signal. The amplitude of the reference signal tones was defined by:

$$\text{Tonal Amplitude} = -0.0036 * f + 1.18 \tag{3.1}$$

This gave a decreasing trend in amplitude for increasing frequency that ranged from 1.0 at 50 Hz to 0.1 at 300 Hz. All tones included in the multiple tone noise test signals were in this range. Reference signals that had tonal amplitudes defined by Eq. 3.1 are termed "weighted" reference signals to distinguish them from reference signals that have tones with equal amplitudes which are termed "equal" reference signals.

To equalize the eigenvalues when the tones in the reference are arbitrarily weighted, a trend line connecting the peaks of the tones (on a power spectrum plot) in the reference is drawn. The inverse of this line gives the desired trend for the magnitude coefficients in $\hat{\mathbf{H}}(z)$, which here corresponds to the inverse of Eq. 3.1. Since the tonal amplitudes for the test case were specified, obtaining the inverse trend line was straight forward. In actual implementation, an offline "Ref ID" process would also be required. This would entail recording the reference signal under normal operating conditions for the system at the sampling frequency used by the controller. The desired magnitude trend for the modified Sys ID filter could be obtained from the Fast Fourier Transform (FFT) or power spectrum plot of the reference.

This type of modified $\hat{\mathbf{H}}(z)$ is designated as an "x-inverse" model. Figure 3.5 shows the trend line for the amplitude of tones in the reference as given by Eq. 3.1, the desired magnitude response for $\hat{\mathbf{H}}(z)$, and the zero padded response for a 128-coefficient x-inverse model. All curves have been normalized in the figure.

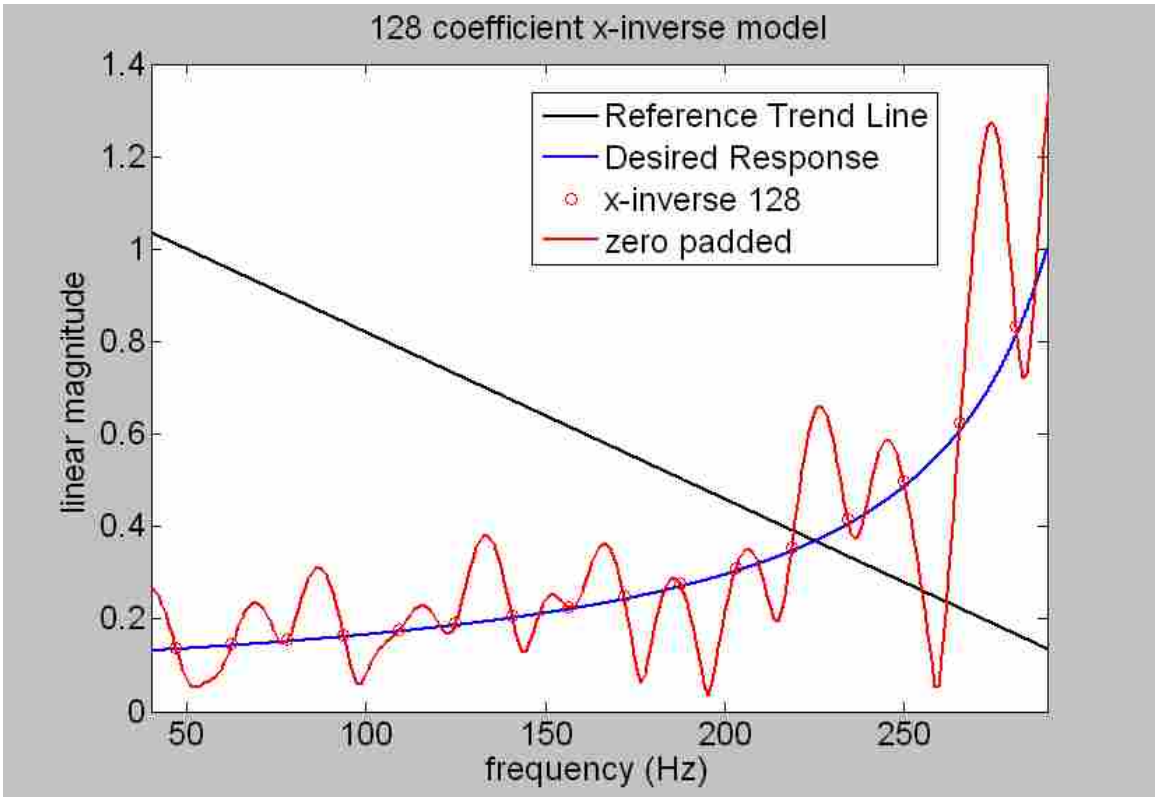

Figure 3.5 Reference tone amplitude trend line (black line) for multiple tone noise signals with desired magnitude response (blue) and 128-coefficient (red circles) and zero padded (red line) x-inverse model magnitude responses.

As before, the response of the filter between bins deviates from the trend assigned to the coefficients. The length of the FIR filter model of $\hat{\mathbf{H}}(z)$ was increased from 128 to 256 to double the resolution in an attempt to constrain the magnitude response between bins to follow more closely the desired trend. The result is shown in Figure 3.6. The result is that the magnitude response matches the desired curve at more points; however, deviation from the desired trend in between these points is not necessarily improved. The same is true for the phase response.

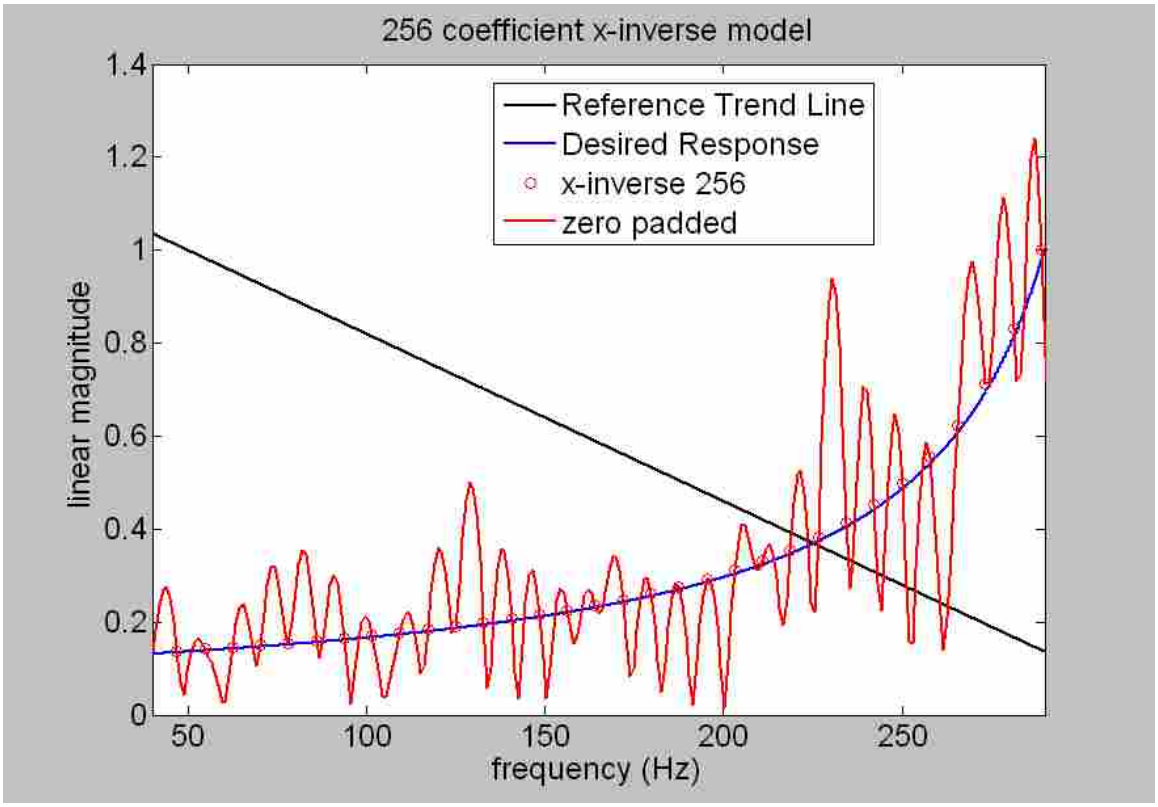

Figure 3.6 Reference tone amplitude trend line for multiple tone noise signals with desired and zero padded x-inverse model magnitude responses.

Thus, an x-inverse model will reduce the eigenvalue variation only for some cases. If the tones in the reference are chosen to correspond exactly to frequency bin values, the eigenvalues are much more uniform using the x-inverse model than using the original model. However if the tones lie off these frequency bin values, the eigenvalue span can be worse than for the unmodified $\hat{\mathbf{H}}(z)$.

Two reference signals containing six tones were made for comparison; one with all six tones on frequency bins (62.5, 93.75, 125, 171.875, 203.125, and 296.875 Hz) and the other with these tones shifted slightly to lie between bin values (50, 100, 130, 180, 200, and 280 Hz). The eigenvalues and eigenvalue span for these weighted reference signals with the original and x-inverse $\hat{\mathbf{H}}(z)$ models of different lengths were calculated and are reported in Table 3.1.

Table 3.1 Comparison of eigenvalue span for original and x-inverse models for 128-coefficient and 256-coefficient filters and tones on and off of frequency bin values.

	On Bins				Off Bins			
	original		x-inverse		original		x-inverse	
Sys ID length	128	256	128	256	128	256	128	256
Eigenvalues for composite signal	340.65	373.87	33.73	33.75	268.54	296.75	73.99	156.91
	323.63	353.49	32.91	32.62	268.69	296.70	73.72	157.69
	107.37	78.24	31.16	30.88	82.83	45.25	34.18	39.41
	96.17	76.10	30.26	30.16	81.92	44.50	34.02	39.25
	68.90	42.42	28.21	28.03	39.29	30.91	12.00	36.10
	67.46	42.08	26.13	26.09	38.61	30.22	11.97	37.48
	57.52	34.77	24.14	24.18	18.41	9.73	10.10	21.34
	55.16	32.32	22.62	22.72	18.37	9.70	10.07	21.18
	13.08	14.40	16.52	16.50	1.68	4.66	6.49	12.11
	12.15	13.27	17.39	17.31	1.65	4.58	6.16	12.72
	0.08	0.17	18.80	19.20	1.25	1.25	5.00	0.16
	0.08	0.17	18.29	18.43	1.24	1.24	4.63	0.16
span	**4328.5**	**2162.4**	**2.0**	**2.0**	**216.8**	**238.5**	**16.0**	**990.5**

When the tones lie on the frequency bins, the x-inverse model gives a significant improvement in the eigenvalue span. For off-bin frequencies the x-inverse model is better than the original for the 128-coefficient filter, but not as good as when the tones are on bins. When the filter length is increased to 256, the span for the x-inverse model was worse than the original model for off bin tones. The span for the x-inverse model with 256 coefficients went from 239 to 991, likely because the magnitude response of the x-inverse model goes almost to zero at 200 Hz (see Figure 3.6). Increasing the resolution by using a longer filter does not (at least in some cases) improve the eigenvalue span. Because it is unlikely that the reference signal will contain frequencies that only correspond to the frequency bins of $\hat{\mathbf{H}}(z)$, the x-inverse method is inadequate since it can result in an eigenvalue span that is worse than the unmodified $\hat{\mathbf{H}}(z)$.

The eigenvalue span for the 128-coeffficient x-inverse model and tones at off-bin frequency values was reduced significantly over the original model from 217 to 16.

However, Figure 3.7 shows that the phase errors introduced exceed stability limits near several tones in the reference. Green and red horizontal lines mark 40 and 90º of phase error between the original and modified phase response and vertical dashed lines show the positions of off-bin reference tones.

Phase errors introduced into regions where no tonal components of the noise are being controlled will not affect the stability or performance of the system. For helicopters the tones in the noise are very stable and do not shift in frequency significantly; however, phase errors very near those tonal frequencies are potentially problematic for both stability and overall performance of the ANC system.

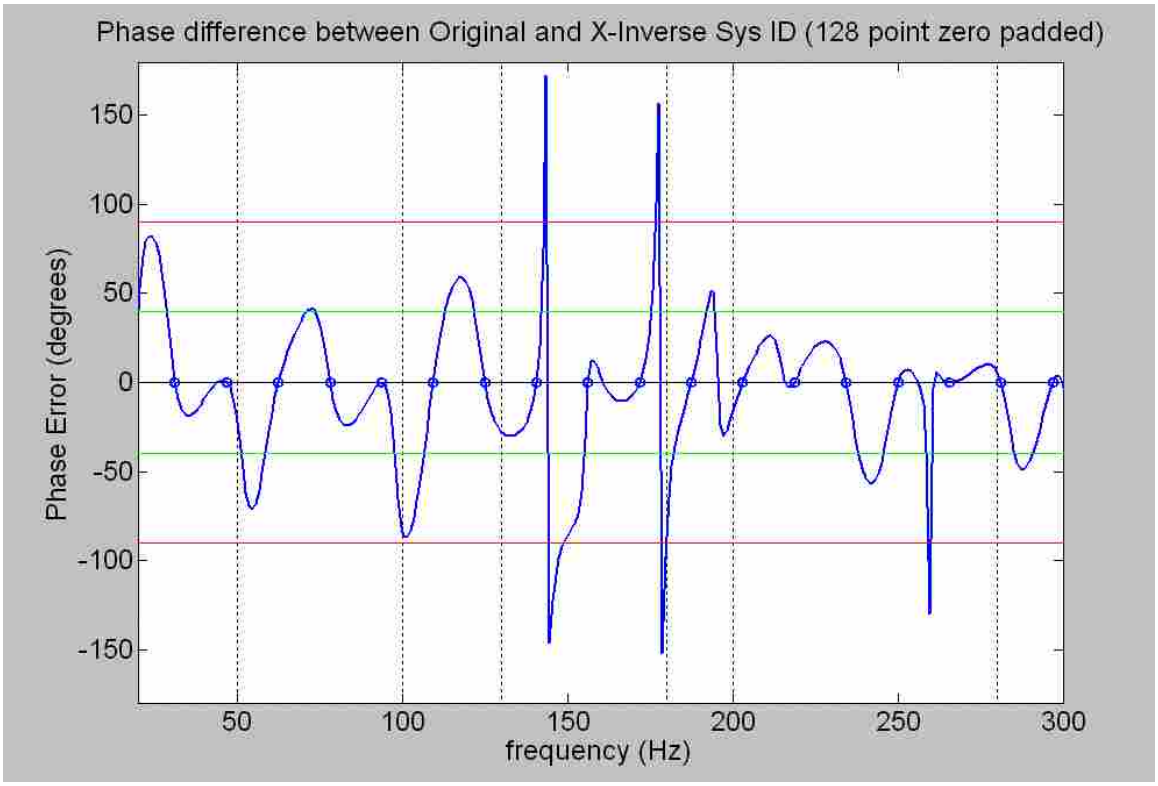

Figure 3.7 Phase difference between 128-coefficient original and x-inverse models with reference tones off frequency bin values.

In this case even though the eigenvalue span was improved, the x-inverse model would not work well if used in ANC since instability and poor performance would result from the phase issues. This is another reason the x-inverse method is inadequate.

The inability to control the magnitude and phase response of the secondary path estimate in between frequency bin values and the unpredictable changes that occur in each when the magnitude coefficients of the original model are modified make the x-inverse method of eigenvalue equalization inadequate. A genetic algorithm approach was developed to optimize the magnitude coefficients, and which can overcome these difficulties for multiple tone noise.

3.3 Genetic Algorithm Optimization of Secondary Path Estimate Magnitude

A genetic algorithm was used to investigate the possibility of getting more uniform eigenvalues for specific multiple tone noise cases. Optimizing the magnitude coefficients of $\hat{\mathbf{H}}(z)$ in ways other than those described previously may lead to improved eigenvalue span, but are not intuitive. Genetic algorithms (GA's) have gained considerable popularity in recent years for their ability to solve problems with a large number of design variables, multiple local minima and maxima, non-differentiable functions, or some combinations of these[32-33]. They can work well for both discrete and continuous variables. GA's mimic the natural selection process in nature. Individuals with the best "fitness" survive and have the opportunity to reproduce. Parents are chosen from the most fit individuals of a population of randomly generated designs. They exchange and pass on their genetic information to offspring which inherit desirable traits from the most fit individuals from the previous generation. As in nature, mutations are

introduced occasionally to provide for random variation. Parents and children compete to be included in the next generation. As the generations progress the probability that the most fit design in the generation is the global optimum for the design space increases.

The genetic algorithm cycle used to optimize the magnitude coefficients of $\hat{\mathbf{H}}(z)$ can be broken down into nine steps. There are multiple ways of implementing each step, each with different advantages and disadvantages. All of these options will not be discussed. Only the methods used in the GA developed for this work will be discussed here. An explanation of all of the various ways of implementing a genetic algorithm, including those used here, is found in the course notes for an optimization based design course offered by the Mechanical Engineering Dept. at BYU[34]. A brief description of each step is now given.

1. Determine a coding for the design

Each design in a GA consists of a number of independent variables chosen by the designer. Each independent variable is called a "gene" and is coded in a "chromosome." Thus, one design, or chromosome, is a set of variables, or genes. As the desired result of the algorithm was to obtain an optimized impulse response model, $\hat{\mathbf{h}}(t)$, that could be used in physical experimentation, a 128, or 256 coefficient $\hat{\mathbf{h}}(t)$ for the mock cabin was obtained by the Sys ID process described in Section 2.2. The FFT of $\hat{\mathbf{h}}(t)$ was then taken, and the magnitude and phase was calculated. Phase information of $\hat{\mathbf{H}}(z)$ was preserved in a vector. The magnitude information of $\hat{\mathbf{H}}(z)$ was discarded, as the GA was implemented to find the optimal magnitude coefficients by making each unknown magnitude coefficient a gene. Each design then contained 64 or 128 genes, representing the

unknown 128 or 256 magnitude coefficients of $\hat{\mathbf{H}}(z)$ (since they are mirrored about the Nyquist frequency).

2. Generate an initial population of size N

Once the coding scheme for a single design was established, a population of N designs was randomly generated. This was done by randomly assigning a value between a specified allowable minimum and maximum value for each gene (magnitude coefficient) in the design. The process was repeated N times to generate the entire population. In general, designs with many genes require large population sizes to maintain adequate diversity. The allowed range for gene values in this case was .01 to 10 and a population size of 500 random designs was used.

3. Calculate fitness for each design

After the initial population was randomly generated, each design was evaluated and assigned a fitness value. Each randomly generated set of magnitude coefficients was recombined with the stored phase information, and the inverse FFT was taken to get a new unique model for the impulse response, $\hat{\mathbf{h}}(t)$. This new model was used to compute the eigenvalues of the filtered-x autocorrelation matrix in the same manner as explained in Section 2.3.

The fitness value was simply the span (λ_{max} divided by λ_{min}) of all nonzero eigenvalues. In addition, a penalty was applied to any design whose phase response was in error by more than 40º in a range of +/- 5 Hz around each of the tonal frequencies in the multiple tone noise. This was done to decrease the design's sensitivity to tonal

frequencies shifting. Constraining the phase in this way ensures that the algorithm will remain stable for small changes in the tonal frequencies. Designs whose performance would be hindered by the phase error introduced by altering the magnitude response were assigned a poor fitness value.

4. Selection of parents

A tournament selection process was used to choose parent designs from the population. A specified number of designs were randomly selected to compete in the tournament. The design with the best fitness wins the tournament and was made a parent design. The smaller the tournament size the more likely a design with poor fitness will be chosen to be a parent. A relatively small tournament size of 5 was used. This process was repeated until enough parents had been selected to make N children; a set of two parent designs producing two children.

5. Perform crossover

A process called crossover exchanged traits from each parent design and created children designs. In this way, new designs were made that had traits from each parent. For this thesis, blend crossover was used. In blend crossover, genes from both parents are blended to make two new children genes. This occurs gene by gene. First a random number between zero and one is chosen for each gene to determine whether crossover will occur. If the random number is larger than the user defined crossover probability no crossover occurs. The genes for the children, c_1 and c_2, are equal to the parent genes, p_1 and p_2 respectively, so that if no crossover occurs for any genes in the design the children

will be identical to the parents. If the random number is less than the user specified crossover probability another random number is chosen. If it is <0.5, the blend parameter, *a*, is calculated by

$$a = \frac{(2r)^{\frac{1}{\eta}}}{2} \tag{3.2}$$

and if the random number is ≥ 0.5 the blend parameter is

$$a = 1 - \frac{(2 - 2r)^{\frac{1}{\eta}}}{2} \tag{3.3}$$

The children genes c_1 and c_2 are created from the parent genes p_1 and p_2 by

$$c_1 = (a)p_1 + (1 - a)p_2 \tag{3.4}$$

$$c_2 = (1 - a)p_1 + (a)p_2$$

The value of η is chosen by the user. As $\eta \rightarrow 0$, the crossover becomes uniform, meaning that $c_1 = p_1$ and $c_2 = p_2$. As $\eta \rightarrow \infty$, $a \rightarrow ½$ and the children's genes are the average of the parent's gene values.

The crossover probability was chosen to be 50% and η was 0.5.

6. Perform mutation

After crossover, some of the genes in the children designs are mutated. Mutation provides for diversity and occasionally introduces new beneficial information into a design. Higher mutation probability maintains more diversity in the designs as the generations progress and can help the algorithm avoid converging on a local optimum in

the design space. Mutation can be made dynamic allowing for high diversity initially, keeping the algorithm from settling prematurely in a local optimum. In later generations mutation is constrained, allowing the algorithm to randomly make fine adjustments to the design once it is near what is hoped to be the global optimum. Initially, mutation can cause the gene to become any value in the allowable range for that gene. By the last generation, when mutation occurs the new value for the gene is only allowed to have a new value that is very close to the original. The probability of mutation occurring does not change, only how different the mutated gene is allowed to be from its pre-mutation value. This is done by introducing a dynamic mutation parameter α,

$$\alpha = \left(1 - \frac{n-1}{N}\right)^{\beta} \tag{3.5}$$

where n is the current generation number and N is the total number of generations. The exponent, β, is a user defined parameter that weights the dynamic function of α. If β = 0, α will always be one and the amount of mutation allowed will be uniform for all generations. If β is greater than zero the amount of mutation allowed decreases as the generation number increases.

A random number is chosen to determine whether mutation will occur for each child gene. If the random number is less than the user-specified mutation probability, another random number, c_{mut}, is chosen within the allowable range for that gene. If c_{mut} is less than the current value for the gene the new gene value is

$$c_{new} = c_{\min} + (c_{mut} - c_{\min})^{\alpha} (c - c_{\min})^{(1-\alpha)} \tag{3.6}$$

and if c_{mut} is greater than the current value for the gene the new gene value is

$$c_{new} = c_{\max} - (c_{\max} - c_{mut})^{\alpha} (c_{\max} - c)^{(1-\alpha)} \quad (3.7)$$

The mutation probability was chosen to be 50% and β was set to 0.5. The values for c_{min} and c_{max} were 0.1 and 10 which were the maximum and minimum values allowed for any gene when the population was originally generated.

7. Measure fitness of children

Once all of the children were created through crossover and mutation, the fitness value of each child was computed in the same way as described in Step 3.

8. Perform elitism

Once each child design has a fitness value, parents are made to compete with children in a process called elitism. All of the parents and children are sorted by their fitness value and the N number of designs with the best fitness value become the starting generation for the next iteration of the algorithm. Elitism increases the evolutionary pressure on the population.

9. Repeat Steps 4 – 8 for M number of generations

Steps 4-8 were repeated for M number of generations. The number of generations needed to be large enough to allow the algorithm to converge on an optimum design. This was determined experimentally. The GA was run for 50 generations. If the fitness of the best design in each generation was still decreasing when the algorithm stopped, it was continued for another 50 generations using the final generation from the first run as the

first generation of the next run. Since the GA was restarted, dynamic mutation was reset introducing higher mutation again in the early generations of the second GA run.

3.4 Genetic Algorithm Results

The genetic algorithm was only run for the reference signal containing six off bin tones as this is the more general case, for 128 and 256 filter coefficients. Figures 3.8-3.10 show the fitness of the best design in each generation of the genetic algorithm. The 128-coefficient model was only run once. The 256-coefficient model was run twice since after the first run it appeared that it might continue to improve the fitness. In Figure 3.10 the first data point is the fitness of the original model, not the last generation of the first run.

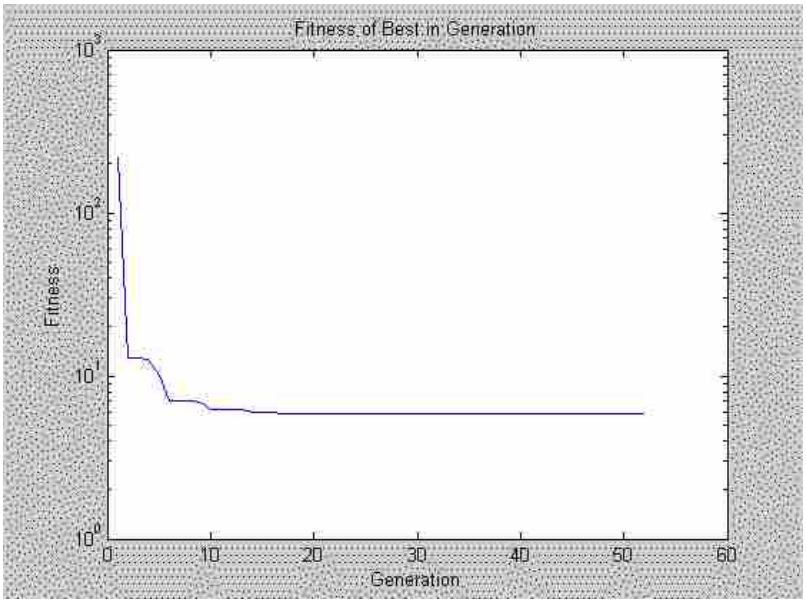

Figure 3.8 Fitness history for genetic optimization of 128-coefficient Sys ID model.

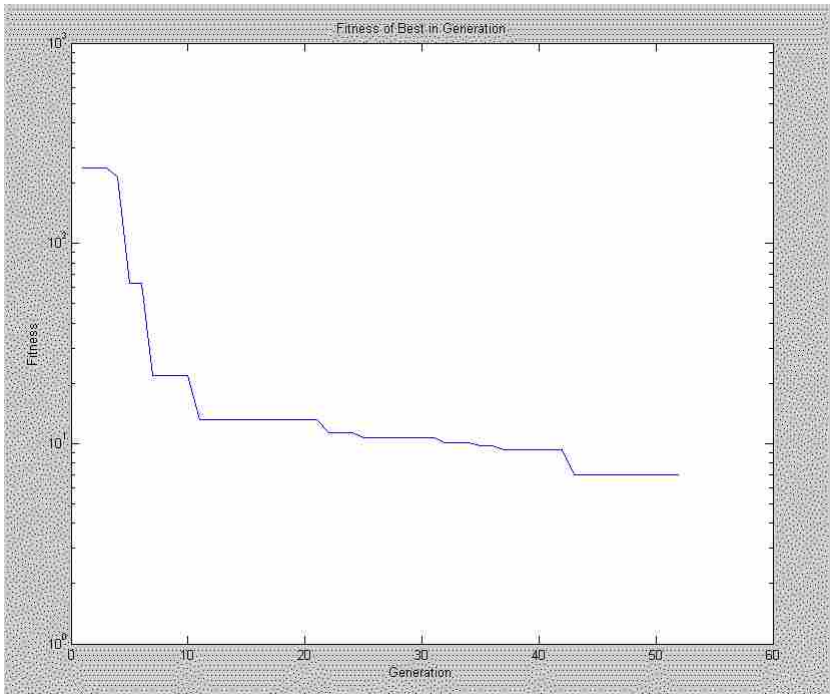

Figure 3.9 Fitness history for first run of genetic optimization of 256-coefficient Sys ID model

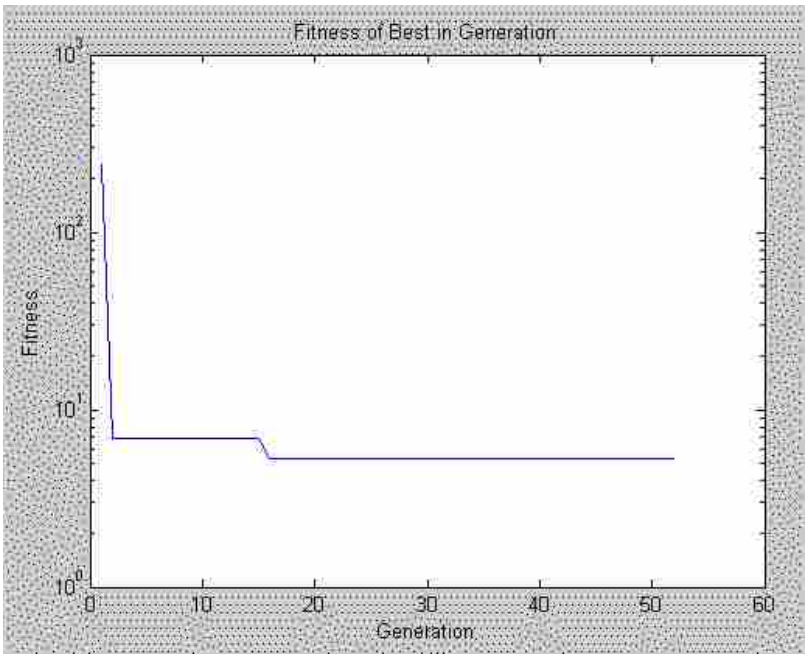

Figure 3.10 Fitness history for second run of genetic optimization of 256-coefficient Sys ID model

The results of the genetic optimization are shown in Table 3.2. The genetic algorithm was able to find a FIR filter with magnitude response that is unlike the x-inverse model and the original model and reduces the eigenvalue span to 5.8 for 128 coefficients and 5.3 for 256 coefficients while keeping the phase response close (within the 40 degree tolerance) to the original near the tonal frequencies. As mentioned in Section 3.2.1, the original or unmodified Sys ID is assumed to be the best model available with a phase response closest to the true phase response of the secondary path. These composite eigenvalues and span are shown in Table 3.2. The values from Table 3.1 for the original and x-inverse models are repeated in Table 3.2 for comparison. Both genetic models have eigenvalue span several orders of magnitude better than the original models and the x-inverse model for 256-coefficients. The 128-coefficient genetic model has better span than the 128-coefficient x-inverse model without the phase error problems shown in Figure 3.6.

Table 3.2 Comparison of eigenvalue span for original, x-inverse and GA magnitude coefficients for off bin tones.

	Off Bins					
	original		x-inverse		genetic	
Sys ID length	128	256	128	256	128	256
Eigenvalues for composite signal	268.54	296.75	73.99	156.91	350.88	15.85
	268.69	296.70	73.72	157.69	332.05	15.39
	82.83	45.25	34.18	39.41	321.39	12.01
	81.92	44.50	34.02	39.25	314.55	11.66
	39.29	30.91	12.00	36.10	283.49	9.17
	38.61	30.22	11.97	37.48	253.61	8.83
	18.41	9.73	10.10	21.34	191.17	4.60
	18.37	9.70	10.07	21.18	185.19	4.55
	1.68	4.66	6.49	12.11	60.34	2.97
	1.65	4.58	6.16	12.72	61.47	3.15
	1.25	1.25	5.00	0.16	63.24	3.28
	1.24	1.24	4.63	0.16	62.59	3.28
span	**216.8**	**238.5**	**16.0**	**990.5**	**5.8**	**5.3**

The 128-coefficient and 256-coefficient designs are shown in Figures 3.11 and 3.13. The magnitude and phase response of the original Sys ID model (blue curve) and the genetic model (red curve) are plotted on the same graph as well as the difference between the original and genetic phase responses in Figures 3.12 and 3.14. The shaded regions mark +/- 5 Hz around each of the tonal frequencies with a vertical dashed line at the tonal frequencies. Here an assumption is made about the amount of frequency shift in the tonal frequencies that may be expected. For helicopter noise, under normal operation the engine always operates at the same rpm and thus the main and tail rotors have the same blade passage frequency. The tonal noise from these sources is very stable and a 10 Hz window is a reasonable constraint to place on the phase error. In these regions, the phase response does not exceed the 40 degree tolerance used by the GA and marked by the green lines.

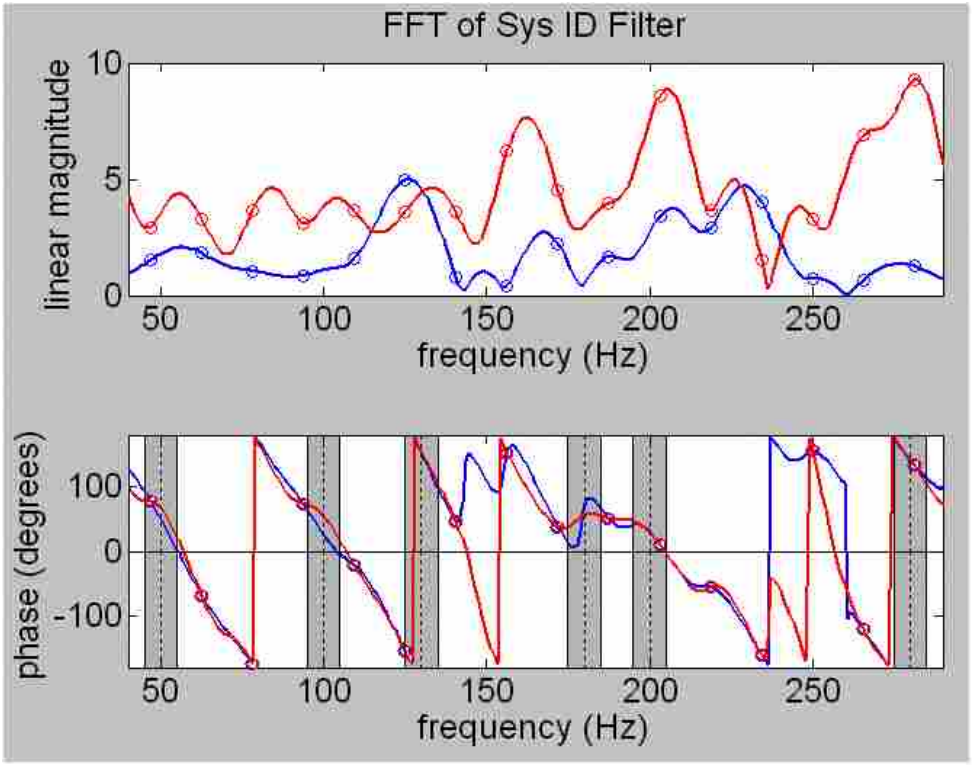

Figure 3.11 The magnitude and phase response of the 128-coefficient (circles) and 128-coefficient zero padded (solid line) original Sys ID model (blue) and the genetic model (red). The shaded regions mark +/- 5 Hz around each of the tonal frequencies with a vertical dashed line at the tonal frequencies.

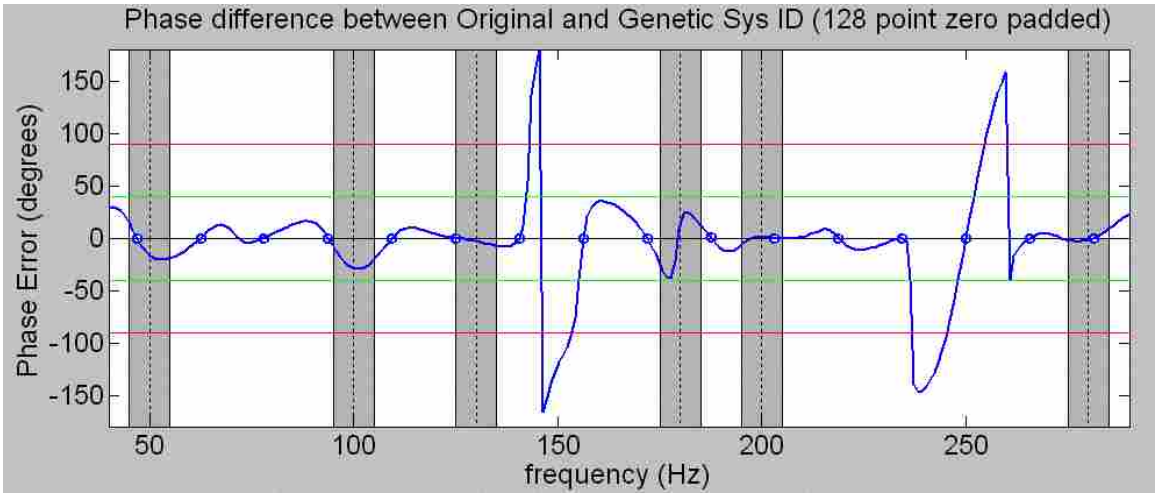

Figure 3.12 Difference between the 128-coefficient zero padded phase response of the original and genetic Sys ID models. The shaded regions mark +/- 5 Hz around each of the tonal frequencies with a vertical dashed line at the tonal frequencies. In these regions the phase response does not exceed the 40 degree tolerance used by the GA and marked by the green lines.

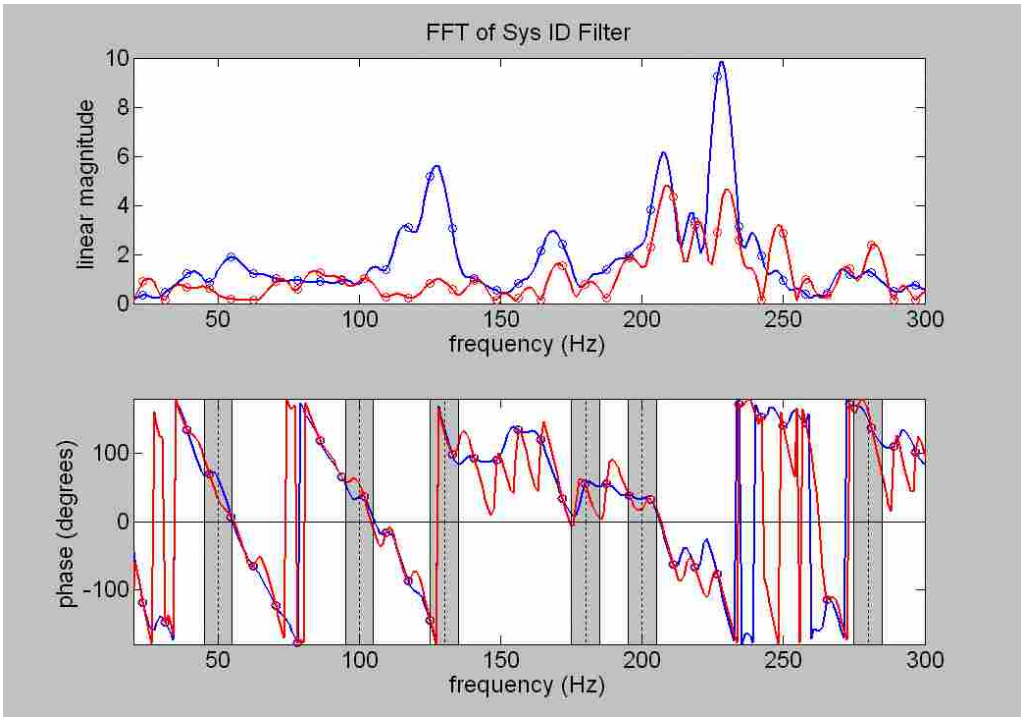

Figure 3.13 The magnitude and phase response of the 256-coefficient (circles) and 256-coefficient zero padded (solid line) original Sys ID model (blue) and the genetic model (red). The shaded regions mark +/- 5 Hz around each of the tonal frequencies with a vertical dashed line at the tonal frequencies.

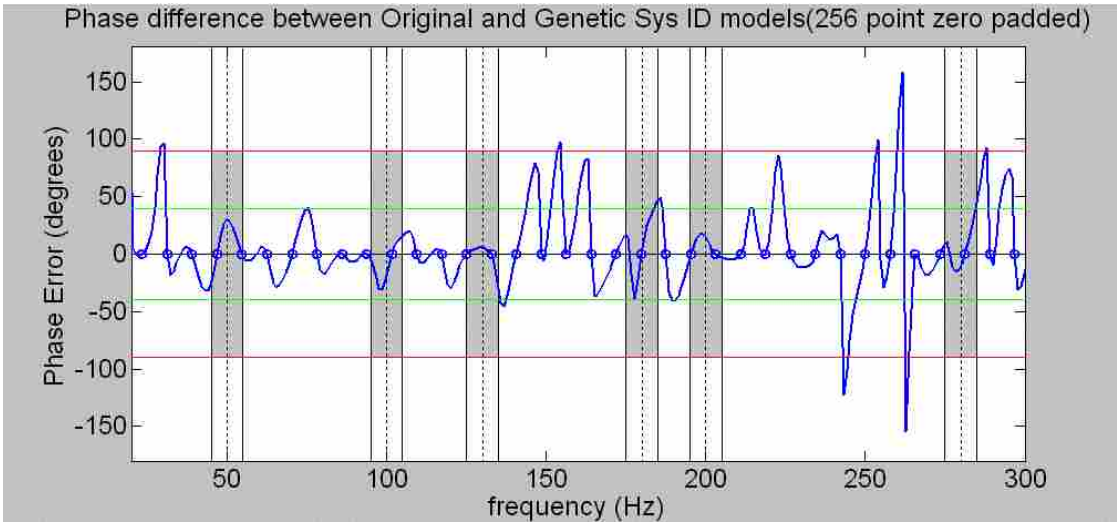

Figure 3.14 Difference between the 256-coefficient zero padded phase response of the original and genetic Sys ID models. The shaded regions mark +/- 5 Hz around each of the tonal frequencies with a vertical dashed line at the tonal frequencies. In these regions the phase response does not exceed the 40 degree tolerance used by the GA and marked by the green lines.

The genetic algorithm was also run without the constraint that the phase be within 40º of the original phase response. This further reduced the eigenvalue span to 2.3 and 2.4 for the 128-coeffficient and 256-coefficient models respectively. These genetic designs had a phase response that was not in error by more than 40º at the tonal frequencies, but in some cases exceeded 40º error within the +/- 5 Hz regions adjacent to the tonal frequencies.

3.4.1 Sensitivity of genetic design to changes in reference tones

The optimized magnitude coefficients found by the genetic algorithm are specific to the noise problem given to the genetic algorithm. If the tones shift in frequency or change in amplitude, the result is no longer guaranteed to be an optimum result. To see how sensitive the genetic algorithm model is to shifts in the tonal frequencies, the

eigenvalue span for reference signals with all tones shifted by +/- 2 Hz up to +/- 10 Hz was calculated. This is compared to the original model in Table 3.3.

Table 3.3 Eigenvalue span for original and genetic Sys ID models and off bin tone reference signals with frequencies of all tones shifted by various amounts.

	f shift	-10	-6	-4	-2	0	2	4	6	10
Original Model	**128 span**	193.5	209.3	217.5	220.7	216.8	213	201.1	202.2	392.7
	256 span	64.4	265.5	444.0	361.7	238.5	169.6	174.8	500.0	1240.1
Genetic Model	**128 span**	9.0	5.1	4.3	5.0	5.8	7.6	9.9	12.8	24.4
	256 span	23.9	11.3	28.2	13.0	5.3	12.9	51.4	132.0	1309.2

In general, the farther the tones get from the frequencies for which the magnitude was optimized the worse the span gets. For this case, they were on average better than the original Sys ID model; however, it is difficult to predict how sensitive a genetic model will be for any given application without first performing the optimization. The sensitivity will depend on how much the magnitude response of the genetic model varies near the frequencies for which it was optimized. Because the genetic algorithm was constrained to have phase errors less than 40º in this range the phase optimization is not affected by shifting the frequencies of the tones in the reference as shown in Figures 3.12 and 3.14. If the tonal frequencies in the reference shift from the values for which they were optimized, degraded performance may result as a consequence of possible increases in the eigenvalue span; however, the phase constraints on the genetic model ensure that the algorithm will remain stable for small shifts in frequency.

CHAPTER 4- EXPERIMENTAL SET-UP

It was shown in Section 3.4 that the genetic Sys ID models had a better eigenvalue span and a phase response close to the original phase response. As explained in Sections 2.4 and 3.1, these improvements should give better performance when the genetic model is used in an ANC system. The benefit of these improvements in an actual ANC test was verified experimentally. The experimental set up used for the ANC tests is described here.

4.1 Mock Cabin Enclosure

ANC tests were conducted to compare the performance of the FXLMS and EE-FXLMS algorithms for various multiple tone noise test signals in a mock cabin enclosure. The mock cabin is made of a metal angle iron frame with ½" plywood panels on five sides and $^1/_8$" acrylic for the front panel. It has nominal dimensions of 45"x40"x60". A chair was placed in the back center of the cabin. A Mackie HR824 studio monitor placed on the floor underneath the chair in the back of the cab was used as a primary noise source.

The cab was placed in a variable acoustics chamber for these tests. The variable acoustics chamber has removable absorptive panels on the walls and the ceiling. All of these panels were installed during these tests. It was not however a hemi anechoic environment. This is partly due to the fact that other equipment was permanently housed in the chamber and also because the panels were not absorptive at the low frequencies

used in the test signals. The main purpose for using this chamber was that it was very quiet. The enclosure is shown in Figure 4.1.

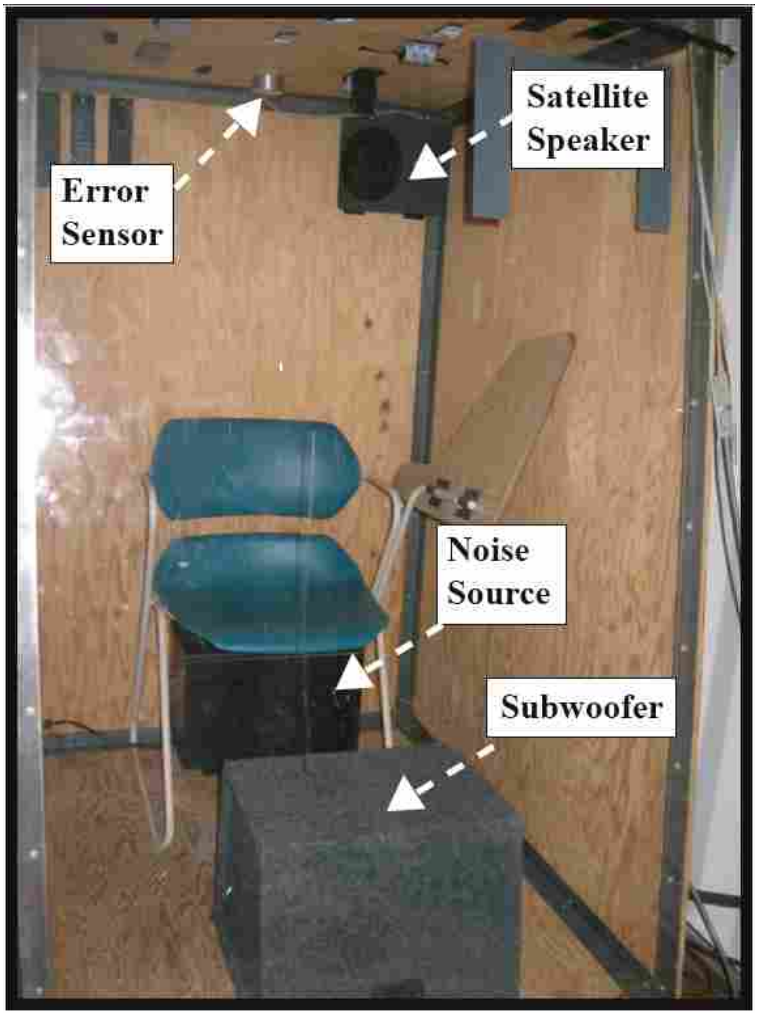

Figure 4.1 Mock cabin enclosure with control system

4.2 Control System

Control sources consisted of a 4" satellite loudspeaker placed in the top back corner of the enclosure behind where an operator's head would be positioned and a 10" subwoofer placed at the front center of the cab on the floor. These were in off-the-shelf car audio speaker enclosures. The control sources were powered by a Clarion APA450 400W stereo amplifier. Even though there were two control sources, it remained a single input, single output system. A crossover circuit with a crossover frequency at 90 Hz split

the single control signal and routed it to either the subwoofer or satellite speaker accordingly.

A small microphone array was placed on the ceiling of the cab above where a person sitting in the chair's head would be. The controller was configured to use one of these microphones as a single squared pressure error sensor.

A single input, single output (SISO) control scheme was used for simplicity in evaluating the performance differences of the FXLMS and EE-FXLMS algorithms. The locations of the control sources and error sensor were driven by convenience and placed where they would be most unobtrusive to someone sitting in the chair. No attempt was made to optimize their placement beyond that.

The control algorithms were implemented on a Texas Instruments TMS320C6713 digital signal processor with a Traquair Data Systems, Inc. I/O interface card, memory, and power supply all housed in the small enclosure shown in Figure 4.2. A separate module containing variable filters and gains was used to condition the inputs and outputs independently for the control system (see Figure 4.3). This second module had an overall cutoff frequency of 300 Hz (low pass), consequently only tonal components below 300 Hz were targeted for control.

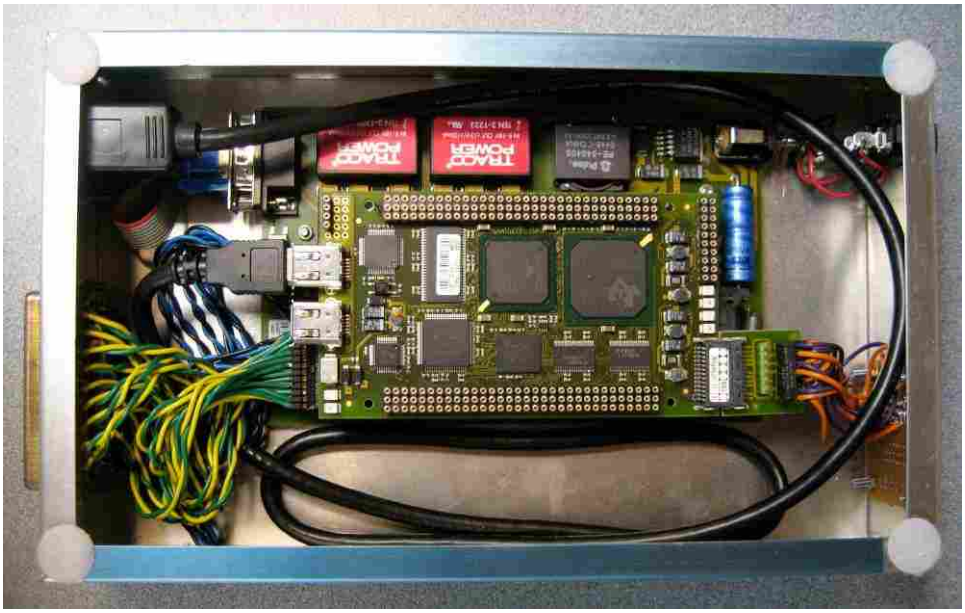

Figure 4.2 DSP board and I/O interface card in enclosure used for ANC tests

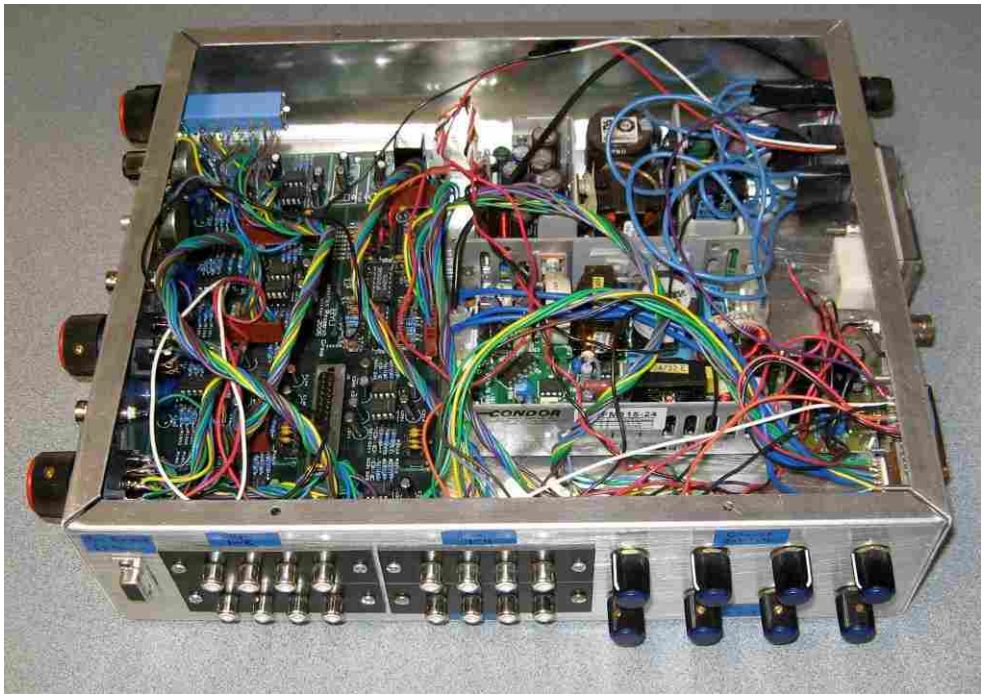

Figure 4.3 "Silver Box"- module containing variable filters and gains used to condition the inputs and outputs independently for the control system

Benjamin Faber at BYU developed a PC based ANC application called "ANC Remote" as an interface to control the parameters of the ANC system on the DSP board and monitor the inputs and outputs of the control system. With this application, the user

can set sampling frequency, convergence parameter size, control channel configurations, Sys ID parameters, control filter length etc.

4.3 Test Signals

The test signals used in the ANC tests were the same as described in Section 3.2.2 and used in the eigenvalue calculations in Sections 3.2.2 and 3.4. Test signals were generated in MATLAB®. The primary noise signal contained tonal components at user specified frequencies with a broadband noise floor. Six tones were combined into a composite signal. For some test cases, the frequencies were chosen to exactly coincide with frequency bin values for the sampling frequency and length of the Sys ID filter. The amplitudes of the individual tonal components were chosen such that when the noise was played through the primary speaker in the cab the tonal components had similar amplitudes at the error sensor.

The reference signal was a summation of these same tones without the added noise floor. As discussed previously, this research assumes that conditioning of the reference signal is undesirable or infeasible so the tonal components in a multi-tone reference have an arbitrary weighting assigned to them. The amplitude of the reference tones decreased with increasing frequency, as defined by Eq. 3.1. This made reference tones that varied in amplitude by up to a factor of ten over the frequency range of interest. Reference signals with the same tones with equal amplitudes in the reference were also used in some ANC tests.

The primary noise and reference signals were written to left and right channels of a stereo wav file with a 25 kHz sampling frequency. These were played back to the

primary source and to the reference input of the DSP using the soundcard on the laptop used to run ANC remote.

4.4 Algorithm Parameters

The ANC system operated at a sampling frequency of 2000 Hz. This is somewhere in the recommended range of 4-10 times the highest frequency to be controlled, which was 300 Hz. The Sys ID filter length was either 128 or 256 coefficients. The convergence parameter value was $1/10^{th}$ the maximum stable value found experimentally for each reference signal and Sys ID model used (original, flat, x-inverse, or genetic). The length of the control filter was 100 for all test cases.

CHAPTER 5- EXPERIMENTAL RESULTS

5.1 ANC Measurements in Mock Cabin Enclosure

To evaluate the performance of the ANC system with various algorithms and Sys ID models, time records of the error signal were taken using a B&K PULSE Data Acquisition System. Ten second time captures sampled at 2048 Hz were taken for three different cases:

(1) Stationary error signal with control off

(2) Converging error signal from the time control was enabled

(3) Stationary error signal after the algorithm had converged to its eventual steady state level.

It should be noted that the wave files used to play noise in the mock cabin were 60 seconds long and played on a loop in the cab. Every time the wav file repeated there was a momentary discontinuity in the signal. This caused the error signal to increase in level slightly and the algorithm to re-converge to its final level again. The third time capture was taken during the last 20 seconds of the wav file to allow for this to happen. For some test cases where convergence of certain tones in the noise was very slow the error signal never reached its steady state level and was always converging.

These data were post processed in MATLAB. The three time captures were plotted on the same graph for each Sys ID model (original, x-inverse, flat, and genetic) to show the convergence of the total error signal. To observe the convergence behavior of the various tonal components, a spectrogram of the converging error signal was made.

Slices from this spectrogram were also plotted for individual frequencies giving the learning curves for those tones overlaid for comparison of each control case.

It should be noted on the spectrogram plot that the tones in the noise do not all start at the same level. So even though some tones may drop below 50dB on the spectrogram which corresponds to black regions on the plot, those tones did not necessarily converge at the same rate since one may have started at a lower level. For this reason comparing convergence of individual tones is best done by looking at the learning curves (Figures 5.2 and 5.4 for example), whereas the spectrogram plots (Figures 5.1 and 5.3 for example) are best for comparing one ANC measurement to another.

The magnitude axes for the plots of the three time captures go from 60 dB to 95 dB while the magnitude on the spectrogram and plots of the learning curves go from 45-75 dB and 20-80 dB respectively. This is intentional and consistent with all similar plots shown here. The combined error signal has greater overall magnitude than the magnitude of the individual tones which are added in the combined error signal. The color axis on the spectrogram is further limited at the lower end to make convergence of the tones easier to compare as they drop into darker regions more quickly.

Along with these plots, tables showing the eigenvalue pairs corresponding to each tone in the noise as well as the eigenvalue span for all of the eigenvalues are included for each case. Color coded phase errors are also reported in the tables. Green corresponds to a phase error at the tonal frequency of less than 40º, yellow to a phase error between 40º and 90º, and red to phase errors 90º and greater.

Two other metrics are used to quantify the ANC performance. Overall eventual attenuation achieved was calculated by comparing the average level of the error signal

with control off and that of the error signal with control on for a long time (first and third time captures). Convergence time was taken to be a measure of how long it took the error signal, from the time that control was enabled, to reach 1/*e* of its initial value (about 9 dB attenuation), where *e* is the base of the natural logarithm. The reason for choosing this was that the convergence time essentially becomes a measure of the rate of attenuation, which was felt to be useful when comparing cases where the overall level of attenuation may be significantly different. When a signal did not converge to 1/*e* of its initial value during the second time capture it is reported as 10+ seconds. The actual convergence time for these measurements was not calculated.

In the remainder of this chapter, first results from each test case will be presented individually in Sections 5.2-5.3. Next some test cases are plotted together for direct comparison in Section 5.4. Eventual reduction and convergence time for all test cases are reported together in Table 5.11 of Section 5.4 also.

5.2 FXLMS ANC Results- Original Sys ID with Weighted Reference

Figures 5.1-5.2 and 5.3-5.4 show ANC results for the FXLMS algorithm with a weighted reference signal (reference tone amplitudes defined by Eq. 3.1) and an unmodified (original) secondary path estimate (128 and 256-coefficients respectively). These measurements serve as a baseline to which other arrangements are compared. The overall reduction of the error signal and convergence time (time to drop to 1/*e* of initial level) was 19.5 dB and 3.9 seconds for the 128-coefficient model and 16.9 dB and 6.3 seconds for the 256-coefficient model. The spectrograms (Figures 5.1 and 5.3) and the individual learning curves (Figures 5.2 and 5.4) show the non-uniform convergence of

the various tones in the noise resulting from eigenvalue disparity. Table 5.1 shows the eigenvalue pairs corresponding to each tone in the noise as well as the eigenvalue span for all of the eigenvalues. Color coded phase errors are also reported in the table as colors.

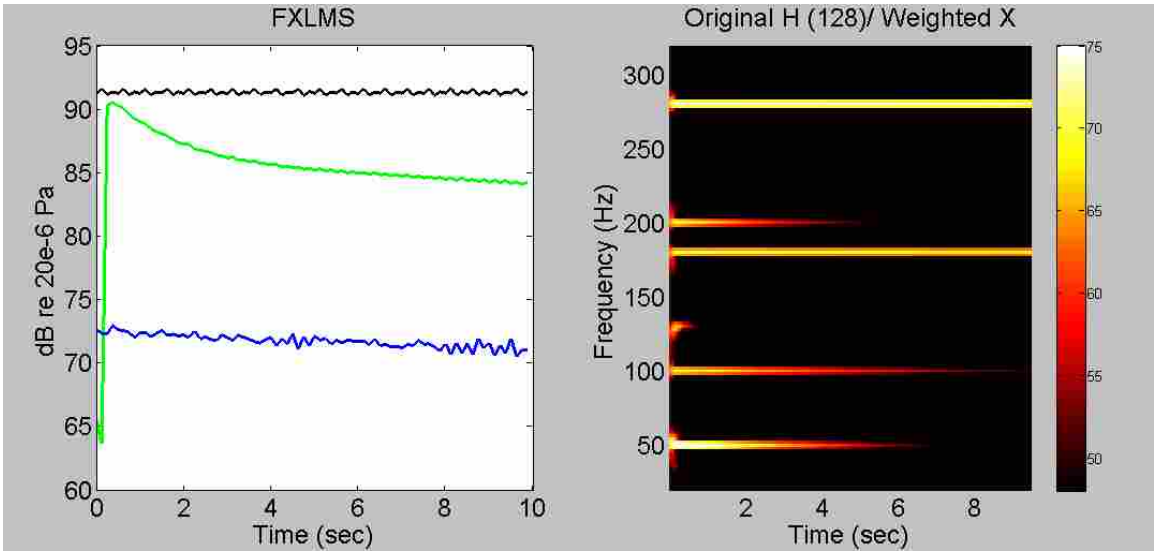

Figure 5.1 The three time captures (ANC off in black, ANC on in green, and steady state ANC in blue) for ANC test with FXLMS algorithm using original 128-coefficient Sys ID model and weighted reference are plotted together on the left. The spectrogram of converging error signal (green curve on the left) from the time ANC was enabled is plotted on the right.

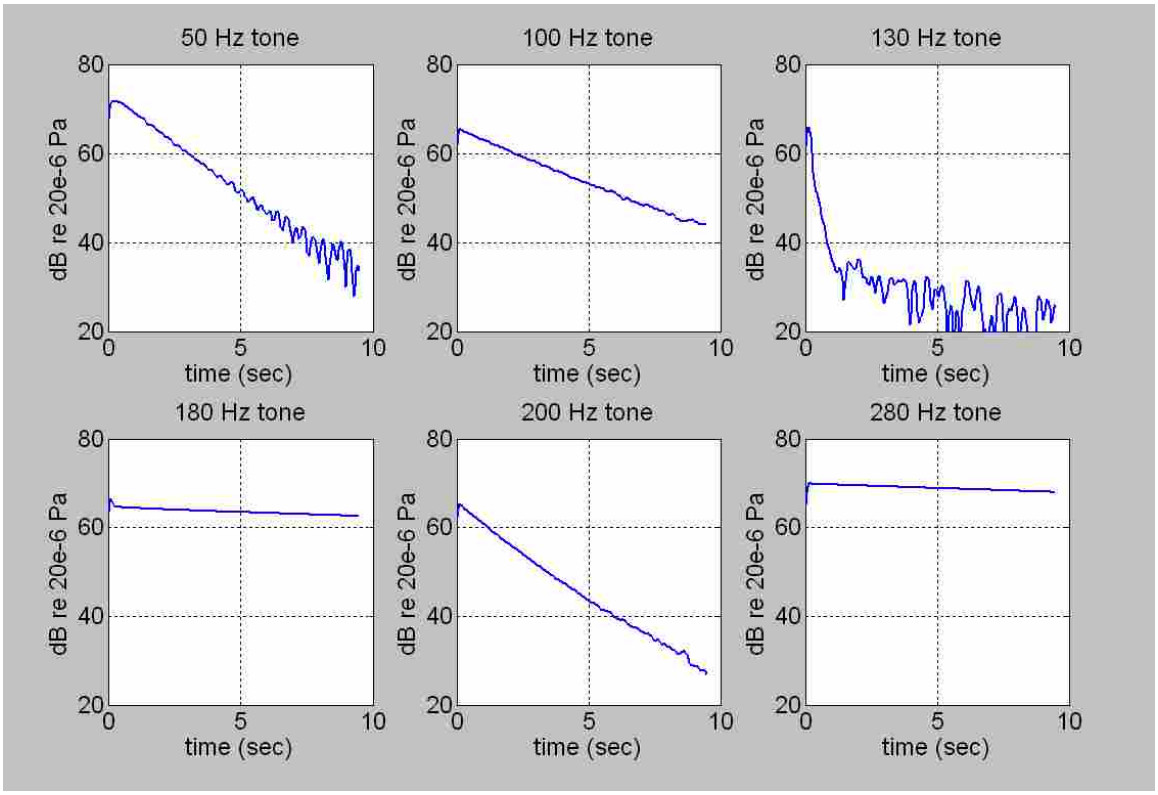

Figure 5.2 Individual learning curves for tones in noise signal for ANC with FXLMS algorithm using original 128-coefficient Sys ID model and weighted reference.

Table 5.1 Eigenvalue pairs for each tone and eigenvalue span for all tones for 128-coefficient original Sys ID model and weighted reference. Color coded phase error at each tone given by: Green $\Phi<40^{o}$, Yellow $40^{o} \leq \Phi < 90^{o}$, Red $\Phi \geq 90^{o}$.

Tone (Hz)	50	100	130	180	200	280	Eigenvalue Span
Eigenvalues	82.83 81.92	18.41 18.37	268.54 268.69	1.68 1.65	39.29 38.61	1.25 1.24	216.8
Phase Error							

The 130 Hz tone converges fastest and corresponds to the largest eigenvalue pair shown in Table 5.1. Slopes of the learning curves for other tones follow the trends in the eigenvalue pairs in Table 5.1. There are no known phase errors in this Sys ID model. The same observations describe the result for the 256-coefficient model shown in Figures 5.3-5.4 and Table 5.2.

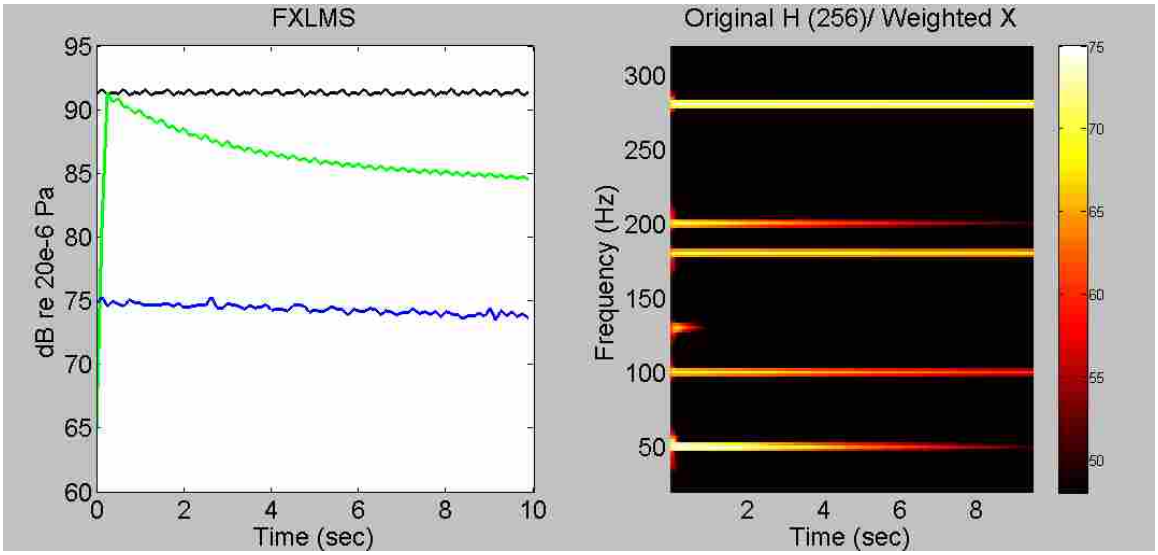

Figure 5.3 The three time captures (ANC off in black, ANC on in green, and steady state ANC in blue) for ANC test with FXLMS algorithm using original 256-coefficient Sys ID model and weighted reference are plotted together on the left. The spectrogram of converging error signal (green curve on the left) from the time ANC was enabled is plotted on the right.

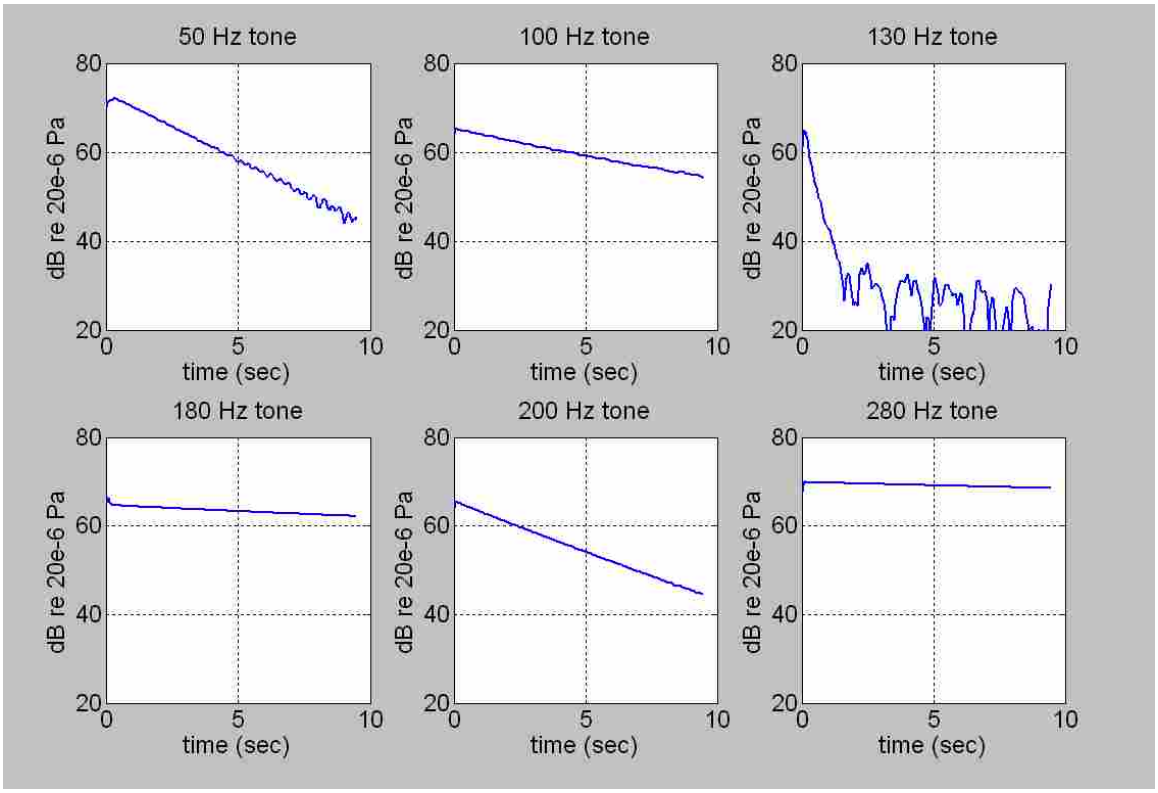

Figure 5.4 Individual learning curves for tones in noise signal for ANC with FXLMS algorithm using original 256-coefficient Sys ID model and weighted reference.

Table 5.2 Eigenvalue pairs for each tone and eigenvalue span for all tones for 256-coefficient original Sys ID model and weighted reference. Color coded phase error at each tone given by: Green Φ<40º, Yellow 40º ≤ Φ < 90º, Red Φ ≥ 90º.

Tone (Hz)	50	100	130	180	200	280	Eigenvalue Span
Eigenvalues	45.3 44.5	9.7 9.7	296.8 296.7	4.7 4.6	30.9 30.2	1.3 1.2	238.5
Phase Error							

5.3 EE-FXLMS ANC Results

This section shows ANC results for various arrangements of the EE-FXLMS algorithm as described in Chapter 3. Secondary path estimate models of various types (flat, x-inverse and genetic) for 128 and 256 coefficients are paired with either a weighted reference signal or a reference with all tones having equal amplitude.

5.3.1 Flat Magnitude Sys ID Model

Previously, the EE_FXLMS algorithm was implemented by flattening the magnitude coefficients of the secondary path estimate[15]. This section shows ANC tests using a flattened magnitude secondary path model and both weighted reference and equal amplitude reference. The flat H/weighted x combination is shown in Figures 5.5-5.6 and Figures 5.8-5.9. While flattening the magnitude may improve eigenvalue span and performance some, it is not matched to the amplitude of the tones in the reference which are not flat. The overall reduction and convergence times were 26.0 dB and 1.3 seconds for the 128-coefficient model and 16.9 dB and 9.0 seconds for the 256-coefficient model. The 128-coefficient flat Sys ID model paired with a weighted reference performed better than the FXLMS case in both measures of performance. Phase errors for the 128-coeffficient model are shown in Figure 5.7. Phase errors in the 128-coefficient flat model

exceeded 40º at the tonal frequency of 180 Hz. Table 5.3 shows the eigenvalue pairs corresponding to each tone in the noise, the eigenvalue span for all of the eigenvalues, and the phase errors as before. Based on the eigenvalues for the 180 Hz tone it is expected that this tone would converge at least as fast as the 200 Hz tone, but in fact converges slower. The phase error at 180 Hz slowed convergence but was not so great as to cause instability.

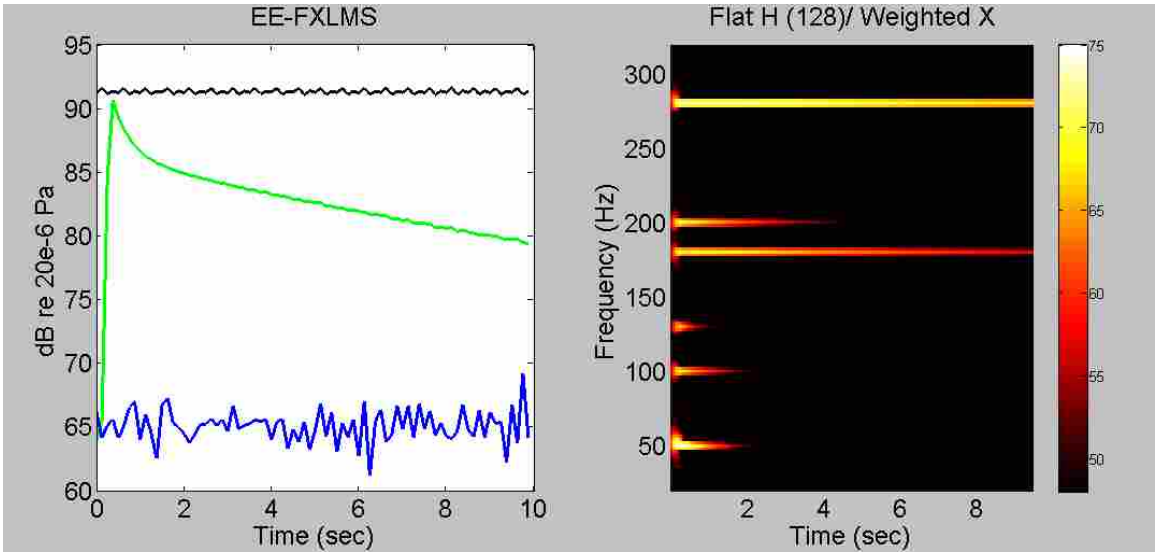

Figure 5.5 Plot of three time captures (left) and spectrogram of converging error signal from the time ANC was enabled (right) for ANC test with EE-FXLMS algorithm using 128-coefficient flat Sys ID model and weighted reference.

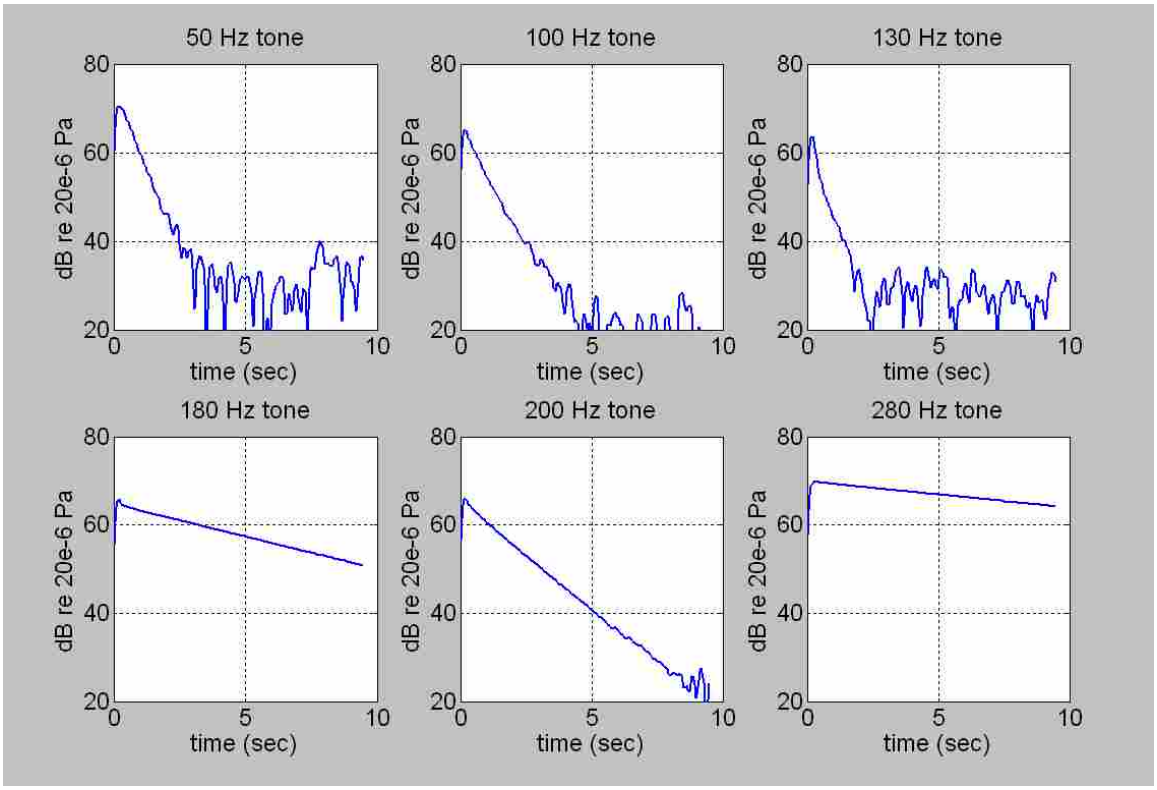

Figure 5.6 Individual learning curves for tones in noise signal for ANC with EE-FXLMS algorithm using 128-coefficient flat Sys ID model and weighted reference.

Table 5.3 Eigenvalue pairs for each tone and eigenvalue span for all tones for 128-coefficient flat Sys ID model and weighted reference. Color coded phase error at each tone given by: Green Φ<40º, Yellow 40º ≤ Φ < 90º, Red Φ ≥ 90º.

Tone (Hz)	50	100	130	180	200	280	Span
Eigenvalues	11.6 11.3	6.0 5.7	3.5 3.4	1.1 1.1	0.8 0.8	0.2 0.2	53.7
Phase Error							

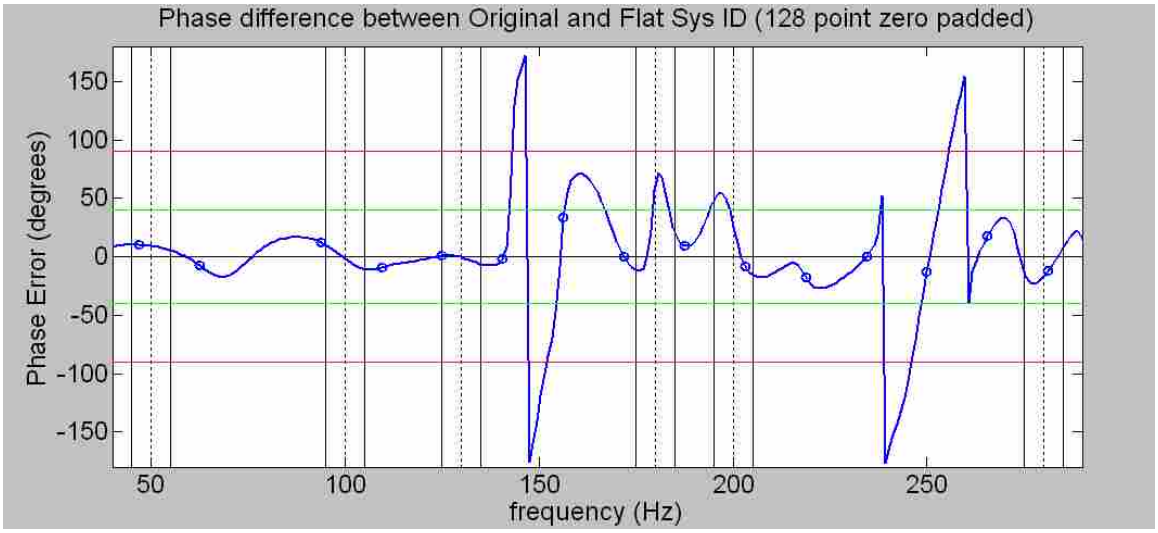

Figure 5.7 Phase Errors (difference from original) for the 128-coefficient flat Sys ID model. Dashed vertical lines mark tonal frequencies and solid vertical lines mark +/- 5 Hz around those tonal frequencies. Note that the phase error exceeds 40º (green horizontal lines) at the 180 Hz tone.

The 256-coefficient flat Sys ID model paired with a weighted reference performed worse with the same overall reduction as the FXLMS case but slower convergence. Phase errors for the 256-coeffficient model are shown in Figure 5.10 and did not exceed 40º at any of the tonal frequencies. Slopes of the learning curves for all of the tones follow the trends in the eigenvalue pairs in Table 5.4.

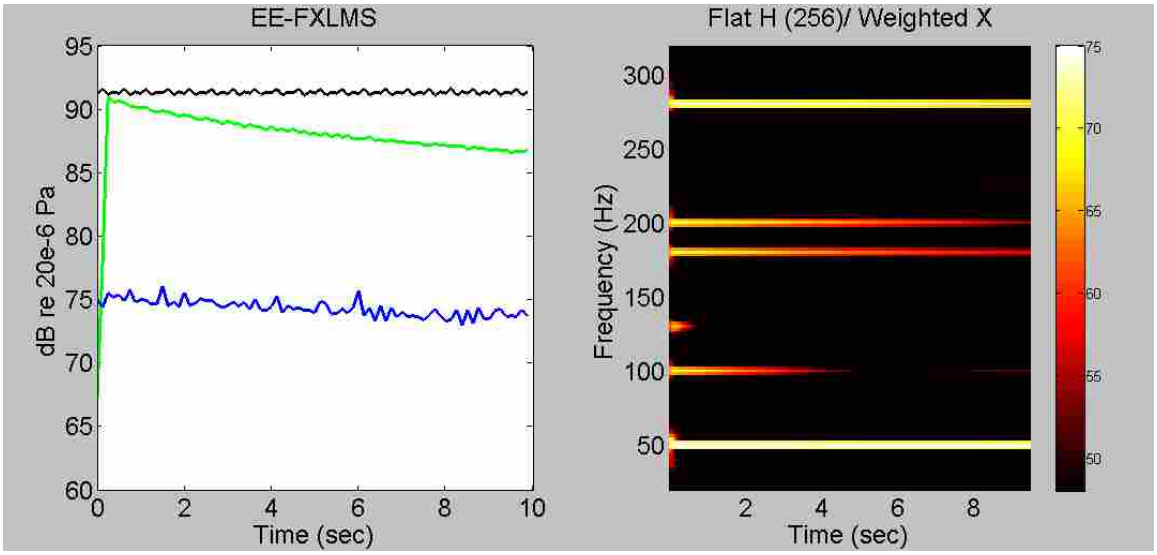

Figure 5.8 Plot of three time captures (left) and spectrogram of converging error signal from the time ANC was enabled (right) for ANC test with EE-FXLMS algorithm using 256-coefficient flat Sys ID model and weighted reference.

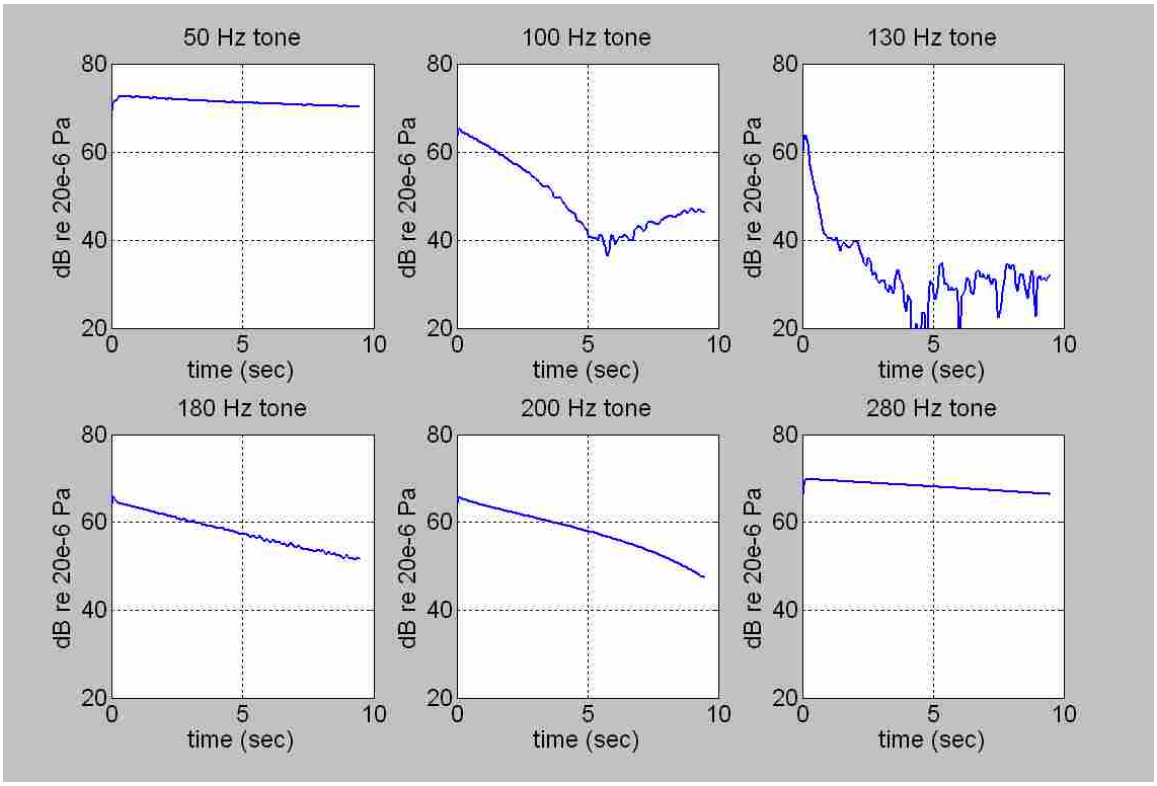

Figure 5.9 Individual learning curves for tones in noise signal for ANC with EE-FXLMS algorithm using 256-coefficient flat Sys ID model and weighted reference.

Table 5.4 Eigenvalue pairs for each tone and eigenvalue span for all tones for 256-coefficient flat Sys ID model and weighted reference. Color coded phase error at each tone given by: Green Φ<40º, Yellow 40º ≤ Φ < 90º, Red Φ ≥ 90º.

Tone (Hz)	50	100	130	180	200	280	Span
Eigenvalues	2.1 2.1	6.5 6.2	19.6 19.8	1.9 1.8	0.1 0.1	0.4 0.4	134.2
Phase Error							

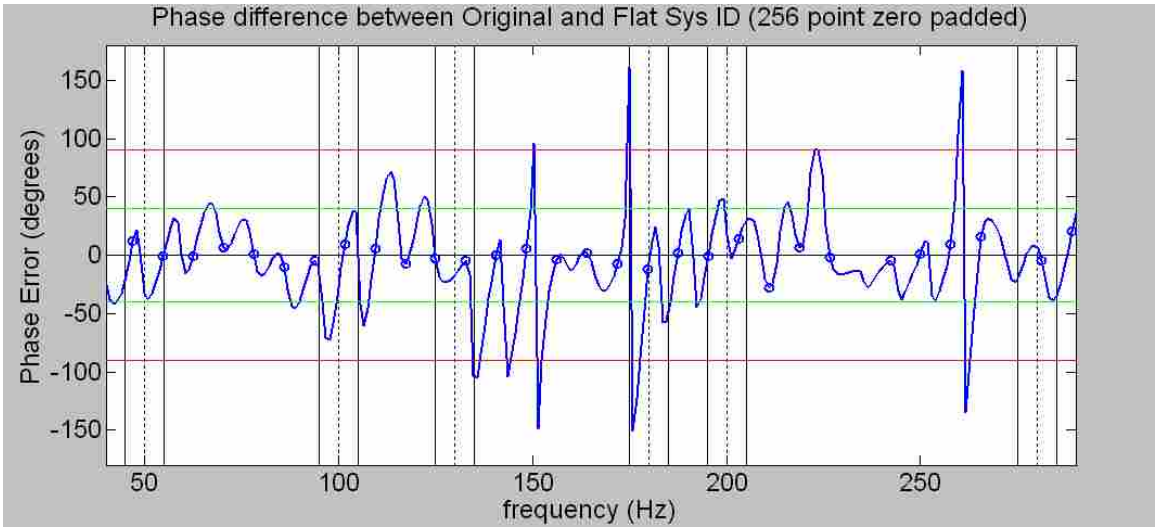

Figure 5.10 Phase errors (difference from original) for the 256-coefficient flat Sys ID model. Dashed vertical lines mark tonal frequencies and solid vertical lines mark +/- 5 Hz around those tonal frequencies.

The flat H/equal X combination shown in Figures 5.11-5.14 would be a good match giving minimum eigenvalue span as long as the variation between bins in the magnitude is not large and significant phase errors are not introduced. This does however require that the tones in the reference signal have equal amplitudes. The overall reduction and convergence times were 25.8 dB and 1.0 seconds for the 128-coefficient model and 13.4 dB and 5.8 seconds for the 256-coefficient model. Pairing the equal amplitude reference to the flat 128-coefficient model further reduces the eigenvalue span and gives slightly better convergence time with approximately the same overall attenuation. Comparison of Figures 5.5 and 5.11 shows that the equal reference gave fast convergence

all the way down to the steady state level. The weighted reference had fast initial convergence which slowed later in the learning curve for the combined signal. This is due to less uniform convergence of the individual tones when the weighted reference is used. Based on the eigenvalue pairs in Table 5.5 the 180 Hz tone should converge about as fast as the 200 Hz tone, but is slower probably due to the phase error at that frequency.

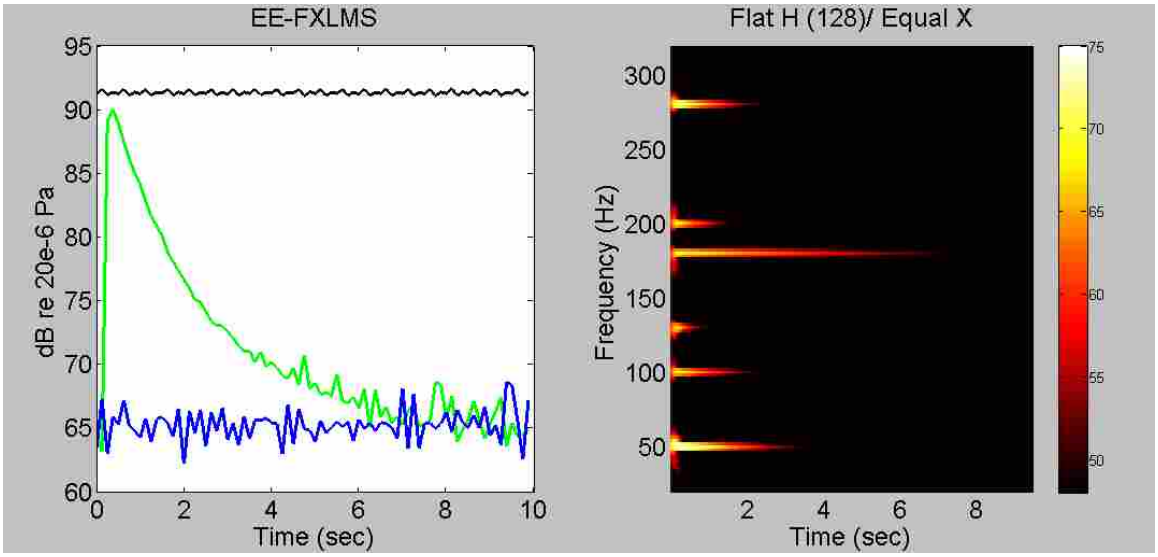

Figure 5.11 Plot of three time captures (left) and spectrogram of converging error signal from the time ANC was enabled (right) for ANC test with EE-FXLMS algorithm using 128-coefficient flat Sys ID model and equal reference.

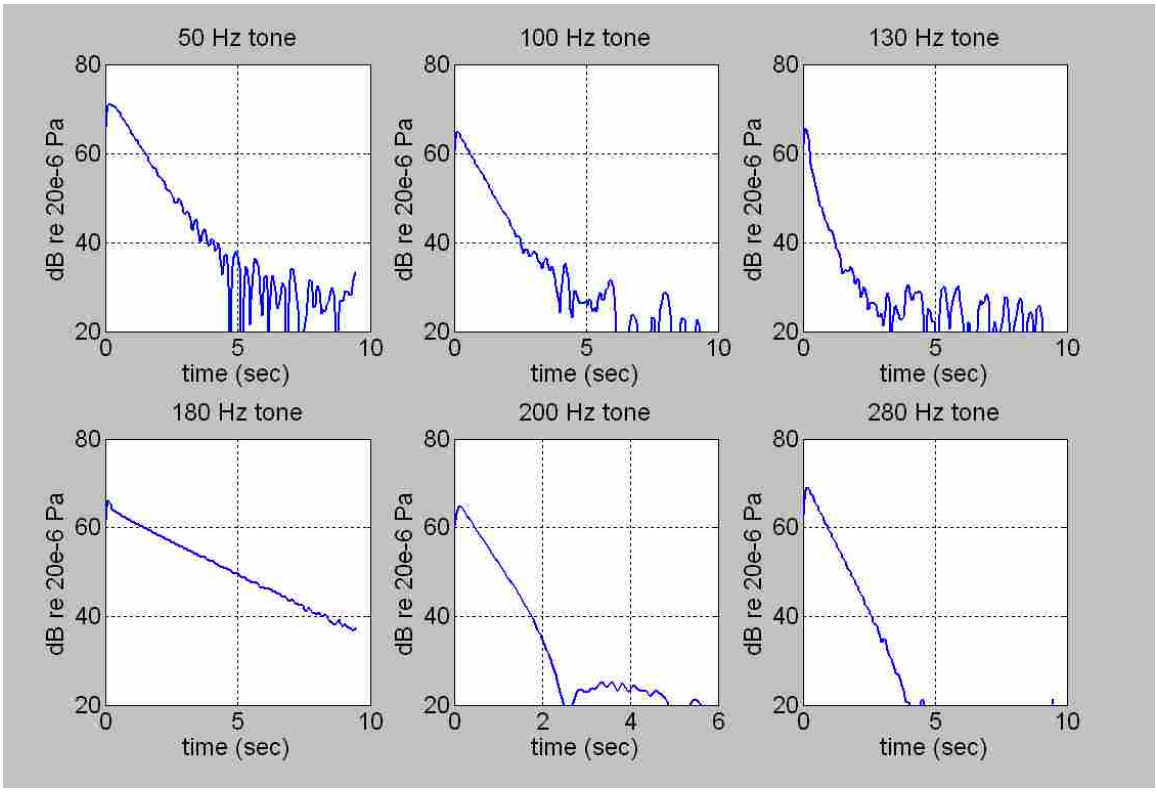

Figure 5.12 Individual learning curves for tones in noise signal for ANC with EE-FXLMS algorithm using 128-coefficient flat Sys ID model and equal reference.

Table 5.5 Eigenvalue pairs for each tone and eigenvalue span for all tones for 128-coefficient flat Sys ID model and equal reference. Color coded phase error at each tone given by: Green Φ<40º, Yellow 40º ≤ Φ < 90º, Red Φ ≥ 90º.

Tone (Hz)	50	100	130	180	200	280	Span
Eigenvalues	12.1 11.5	9.7 9.2	7.3 7.2	3.6 3.7	3.9 3.9	6.3 6.1	3.4
Phase Error							

The 256-coeficient model paired with an equal amplitude reference performed worse than when paired to the weighted reference and worse than the FXLMS algorithm. From the eigenvalue pairs in Table 5.6 it is expected that the 200 Hz tone would converge the slowest while Figure 5.14 shows that the 50 Hz tone is the slowest. This is an unexplained result.

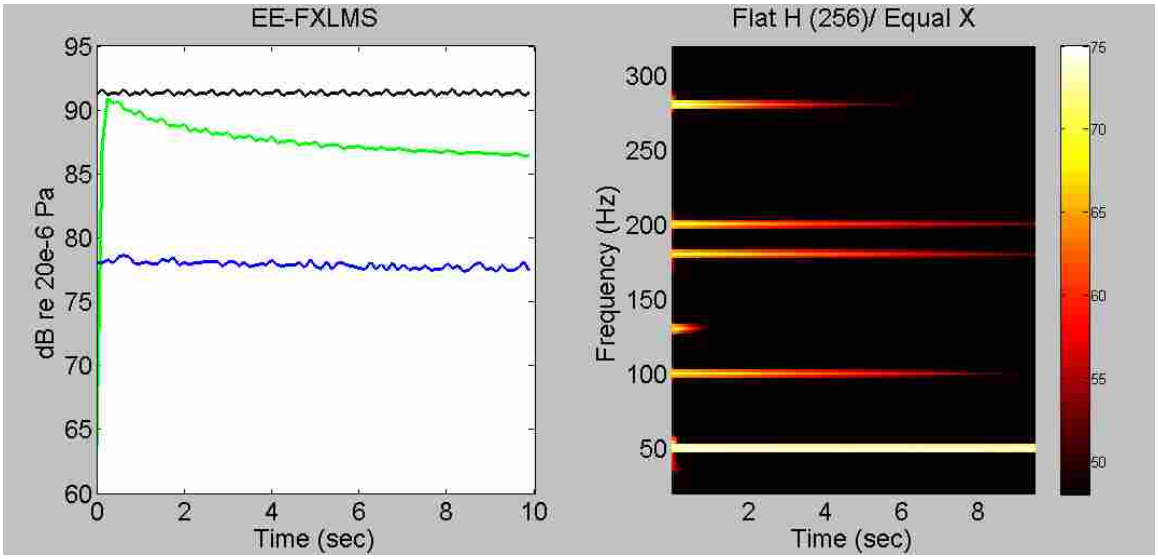

Figure 5.13 Plot of three time captures (left) and spectrogram of converging error signal from the time ANC was enabled (right) for ANC test with EE-FXLMS algorithm using 256-coefficient flat Sys ID model and equal reference.

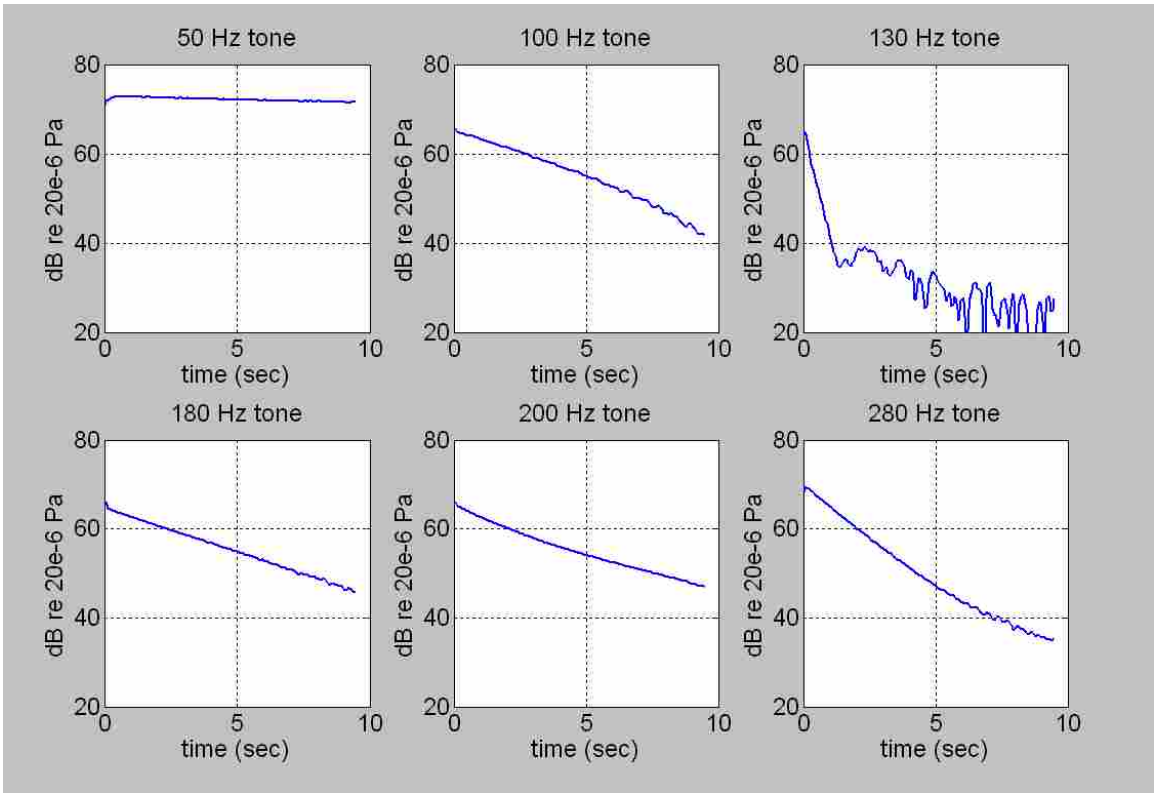

Figure 5.14 Individual learning curves for tones in noise signal for ANC with EE-FXLMS algorithm using 256-coefficient flat Sys ID model and equal reference.

Table 5.6 Eigenvalue pairs for each tone and eigenvalue span for all tones for 256-coefficient flat Sys ID model and equal reference. Color coded phase error at each tone given by: Green Φ<40º, Yellow 40º ≤ Φ < 90º, Red Φ ≥ 90º.

Tone (Hz)	50	100	130	180	200	280	Span
Eigenvalues	2.0 2.1	9.7 9.4	38.7 38.8	6.7 6.6	0.7 0.7	12.0 11.9	**55.7**
Phase Error							

5.3.2 X-Inverse Magnitude Sys ID Model

Figures 5.15-5.16 show ANC results for the 128-coefficient x-inverse Sys ID model paired with a weighted reference, and Figures 5.18-5.19 show the results for the 256-coeffficeient model. The overall reduction and convergence times were 4.6 dB and 10+ seconds for the 128-coefficient model and 2.6 dB and 10+ seconds for the 256-coefficient model.

The drawbacks of using an x-inverse Sys ID model were discussed in section 3.2.2. In Figures 5.17 and 5.20 the phase errors in the two x-inverse Sys ID models are shown. The phase error exceeds the 90 degree stability limit at 180 Hz for the 128-coefficient model and at 200 Hz for the 256-coefficient model. These two tones were shown to diverge and increase in level in the error signal in Figures 5.16 and 5.19. Phase error was in excess of 40º at other frequencies for both cases. Those phase errors, specifically those exceeding 90º, made ANC for these test cases perform poorly because of instability problems, even though the eigenvalue span was reduced from 217 for the original model to 16 for the 128-coefficient x-inverse case. Both x-inverse cases shown in Figures 5.15-5.20 performed worse than the FXLMS algorithm shown in Figures 5.1-5.4. It is difficult to make comparisons between the eigenvalue pairs from Tables 5.7 and 5.8

to the convergence of individual tones in Figures 5.16 and 5.19. The control output had to be limited to maintain some level of overall convergence due to the divergent tones.

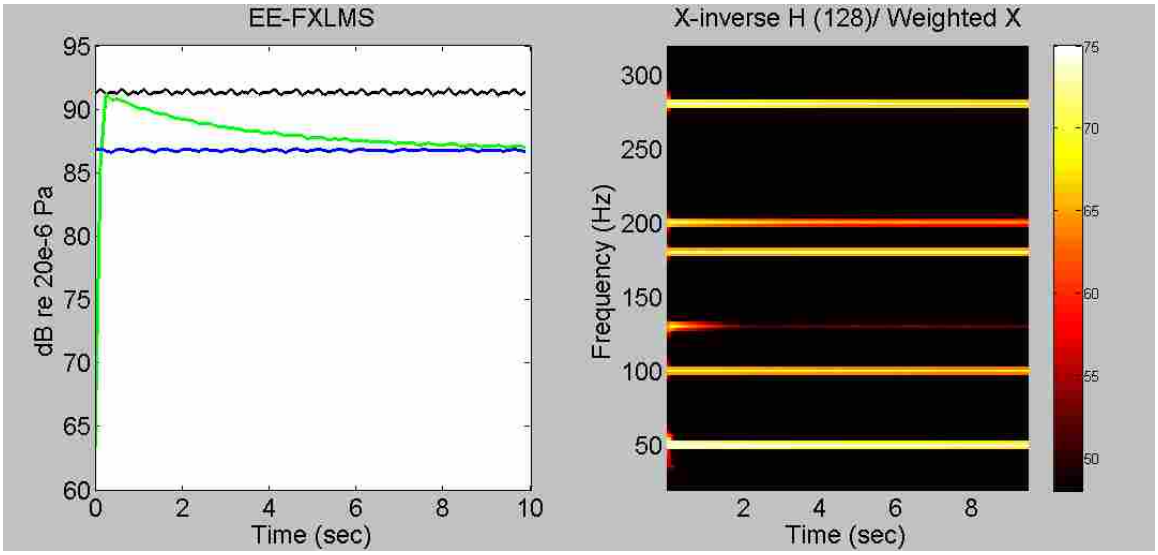

Figure 5.15 Plot of three time captures (left) and spectrogram of converging error signal from the time ANC was enabled (right) for ANC test with EE-FXLMS algorithm using 128-coefficient x-inverse Sys ID model and weighted reference.

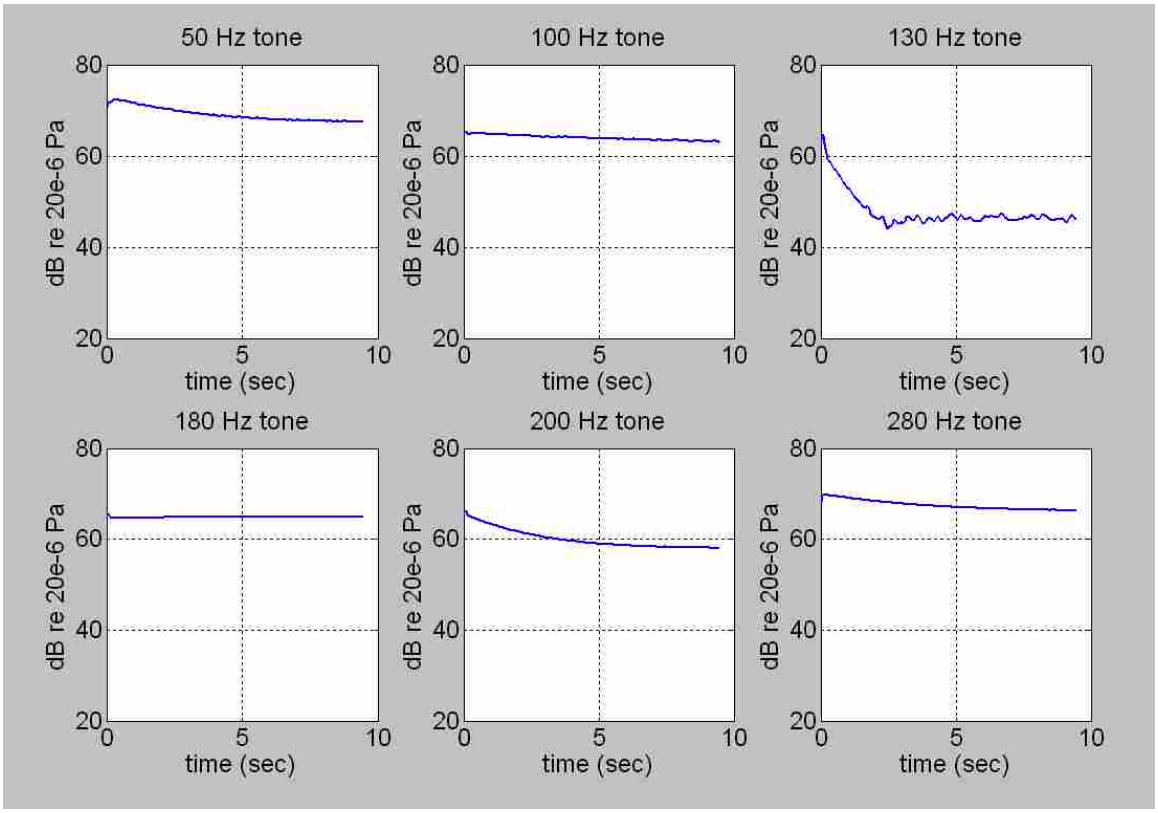

Figure 5.16 Individual learning curves for tones in noise signal for ANC with EE-FXLMS algorithm using 128-coefficient x-inverse Sys ID model and weighted reference.

Table 5.7 Eigenvalue pairs for each tone and eigenvalue span for all tones for 128-coefficient x-inverse Sys ID model and weighted reference. Color coded phase error at each tone given by: Green Φ<40º, Yellow 40º ≤ Φ < 90º, Red Φ ≥ 90º.

Tone (Hz)	50	100	130	180	200	280	Span
Eigenvalues	5.0 4.6	6.5 6.2	74.0 73.7	10.1 10.1	12.0 12.0	34.2 34.0	16
Phase Error							

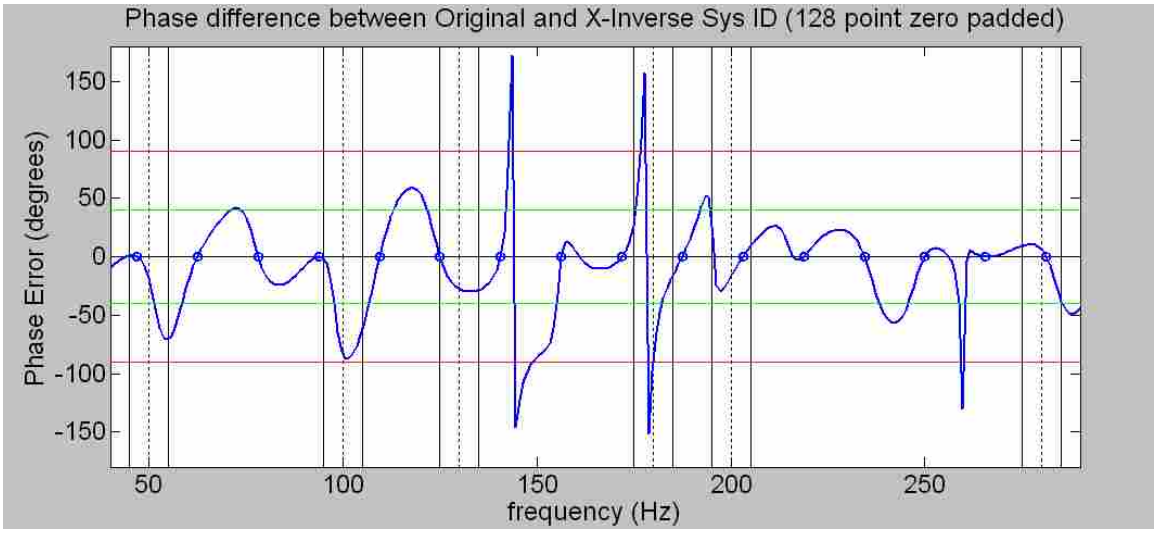

Figure 5.17 Phase Errors (difference from original) for the 128-coefficient x-inverse Sys ID model. Dashed vertical lines mark tonal frequencies and solid vertical lines mark +/-5 Hz around those tonal frequencies. Note that the phase error exceeds 40º (green horizontal lines) at 100 Hz and approaches the 90 degree stability limit (red line) at 180 Hz.

For the 256-coefficient model the eigenvalue span was calculated to be 991, which is an increase from the original eigenvalue span of 239. This is because of the deviation from the desired optimized trend of the magnitude between frequency bins. This increase may have also contributed to the poor ANC performance for that test, although the instability from phase error is the biggest limiting factor.

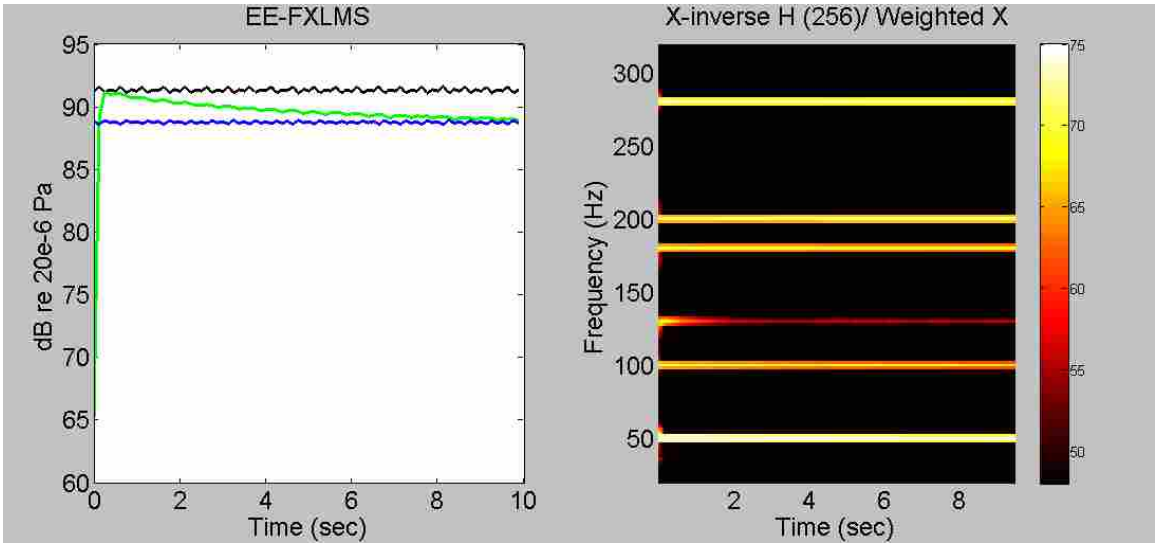

Figure 5.18 Plot of three time captures (left) and spectrogram of converging error signal from the time ANC was enabled (right) for ANC test with EE-FXLMS algorithm using 256-coefficient x-inverse Sys ID model and weighted reference.

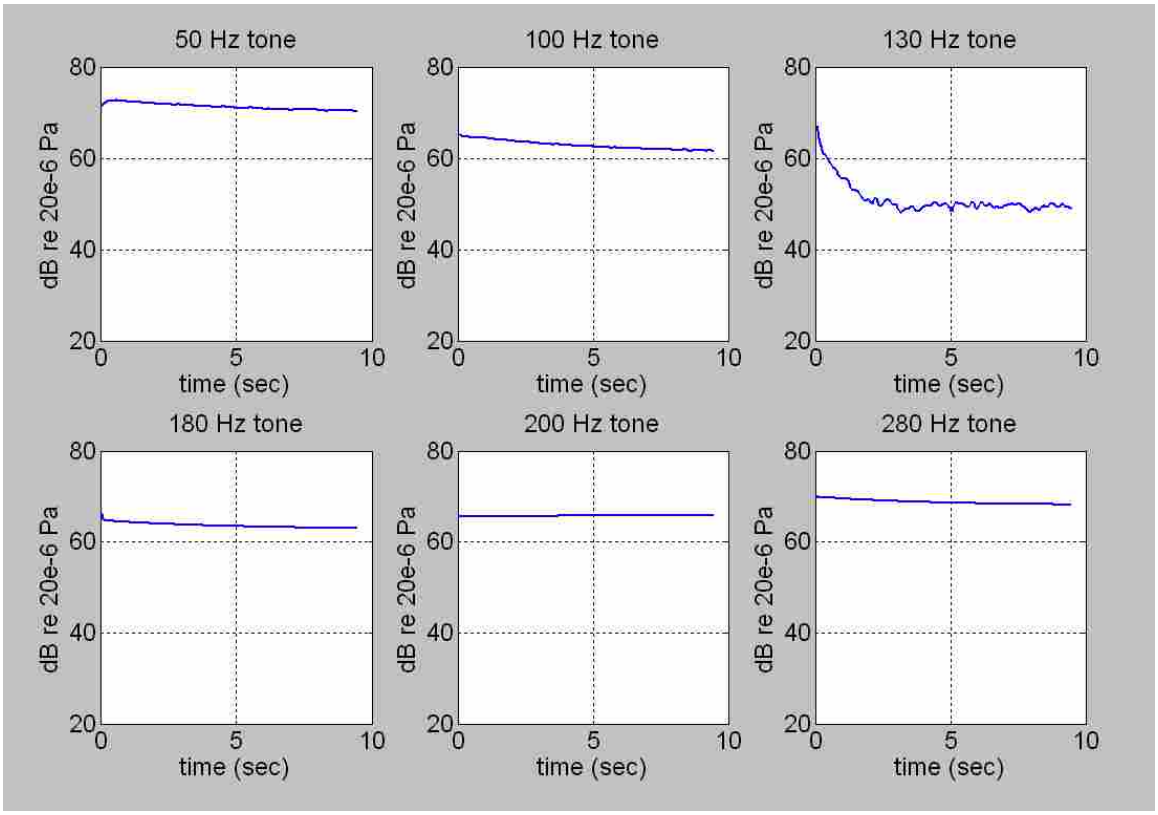

Figure 5.19 Individual learning curves for tones in noise signal for ANC with EE-FXLMS algorithm using 256-coefficient x-inverse Sys ID model and weighted reference.

Table 5.8 Eigenvalue pairs for each tone and eigenvalue span for all tones for 256-coefficient x-inverse Sys ID model and weighted reference. Color coded phase error at each tone given by: Green Φ<40º, Yellow 40º ≤ Φ < 90º, Red Φ ≥ 90º.

Tone (Hz)	50	100	130	180	200	280	Span
Eigenvalues	12.1 12.7	36.1 37.5	156.9 157.7	21.3 21.2	0.2 0.2	39.4 39.2	990.7
Phase Error							

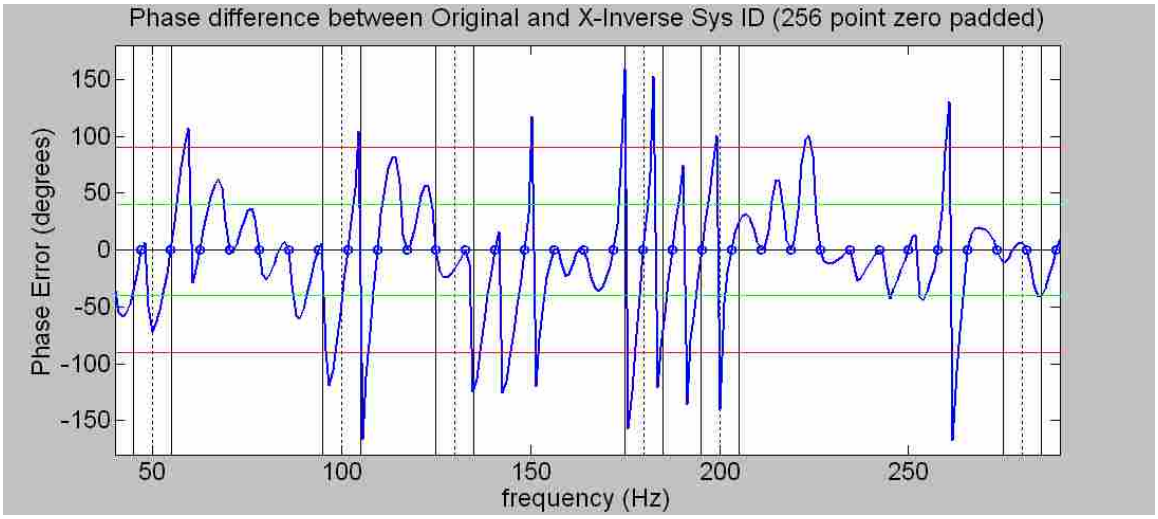

Figure 5.20 Phase Errors (difference from original) for the 256-coefficient x-inverse Sys ID model. Dashed vertical lines mark tonal frequencies and solid vertical lines mark +/-5 Hz around those tonal frequencies. Note that the phase error exceeds 40º (green horizontal lines) at the 50 and 100 Hz and approaches the 90 degree stability limit (red line) at 200 Hz.

5.3.3 Genetically Optimized Sys ID Model

The genetic algorithm optimized the magnitude response of the Sys ID model to minimize the eigenvalue span while assuring that the phase was not significantly in error at frequencies near the tones in the noise. This resulted in good ANC performance in the tests as shown in Figures 5.21-5.24. The eigenvalues shown in Table 5.9 predict that the 100 Hz or 200 Hz tones ought to converge the fastest for the 128-coefficeint genetic model since they have the largest eigenvalues of the set and the convergence parameter

used would be closest to the optimum for these frequencies. The 130 Hz tone converges fastest initially for this case. The phase error at 100 Hz is less than 40° but greater than the phase errors at 130 Hz or 200 Hz which were <10°. This may explain why the 100 Hz tone converged slower than the tone at 130 Hz. It is unclear why the 200 Hz tone did not converge faster than the 130 Hz tone. Convergence of all tones is more uniform than for any other case with the exception of the Flat Sys ID model paired with the equally weighted reference which had faster convergence for the 280 Hz tone.

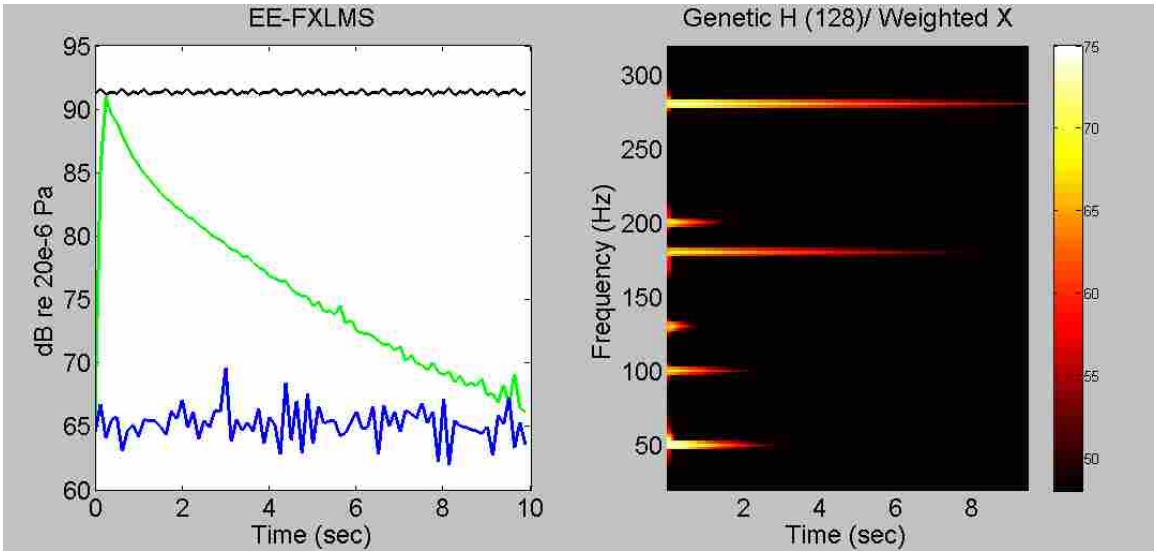

Figure 5.21 Plot of three time captures (left) and spectrogram of converging error signal from the time ANC was enabled (right) for ANC test with EE-FXLMS algorithm using 128-coefficient genetic Sys ID model and weighted reference.

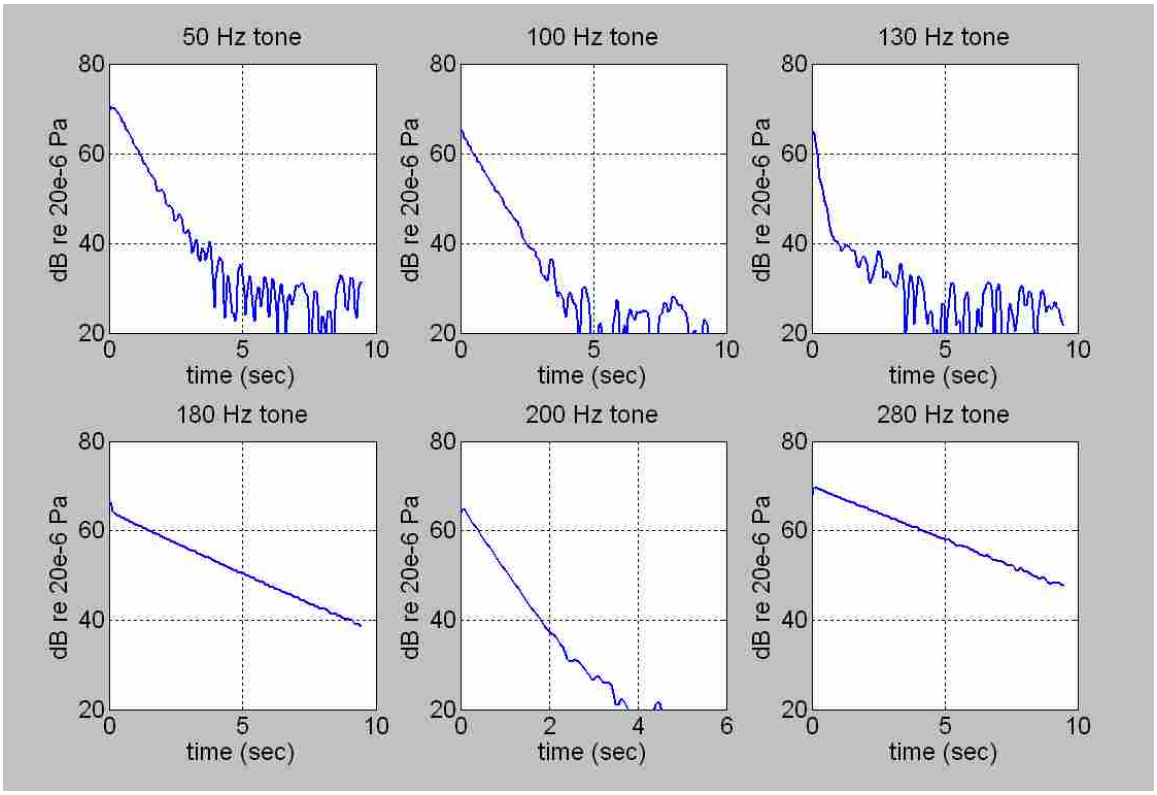

Figure 5.22 Individual learning curves for tones in noise signal for ANC with EE-FXLMS algorithm using 128-coefficient genetic Sys ID model and weighted reference.

Table 5.9 Eigenvalue pairs for each tone and eigenvalue span for all tones for 128-coefficient genetic Sys ID model and weighted reference. Color coded phase error at each tone given by: Green Φ<40º, Yellow 40º ≤ Φ < 90º, Red Φ ≥ 90º.

Tone (Hz)	50	100	130	180	200	280	Span
Eigenvalues	81.0 73.2	135.3 133.1	100.5 92.8	75.0 74.1	144.4 143.5	63.8 65.9	2.3
Phase Error							

The eigenvalue pairs for the 256-coefficent model shown in Table 5.10 do not perfectly explain the convergence trends in the learning curves for the individual tones in Figure 5.24. The overall reduction and convergence times were 25.9 dB and 1.0 seconds for the 128-coefficient model and 26.1 dB and 2.2 seconds for the 256-coefficient model. These results are the best case of those considered thus far.

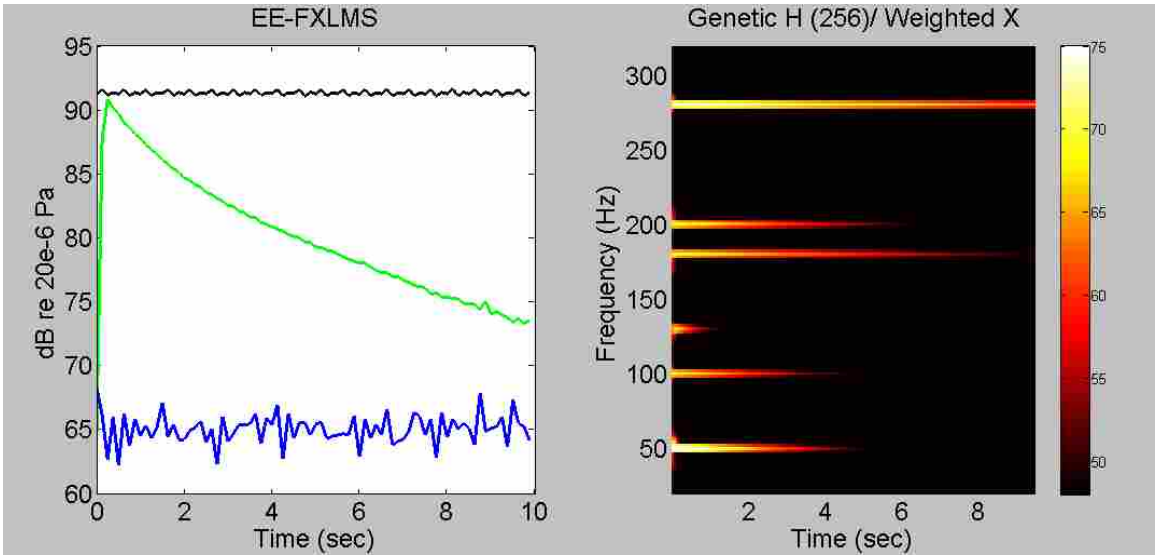

Figure 5.23 Plot of three time captures (left) and spectrogram of converging error signal from the time ANC was enabled (right) for ANC test with EE-FXLMS algorithm using 256-coefficient genetic Sys ID model and weighted reference.

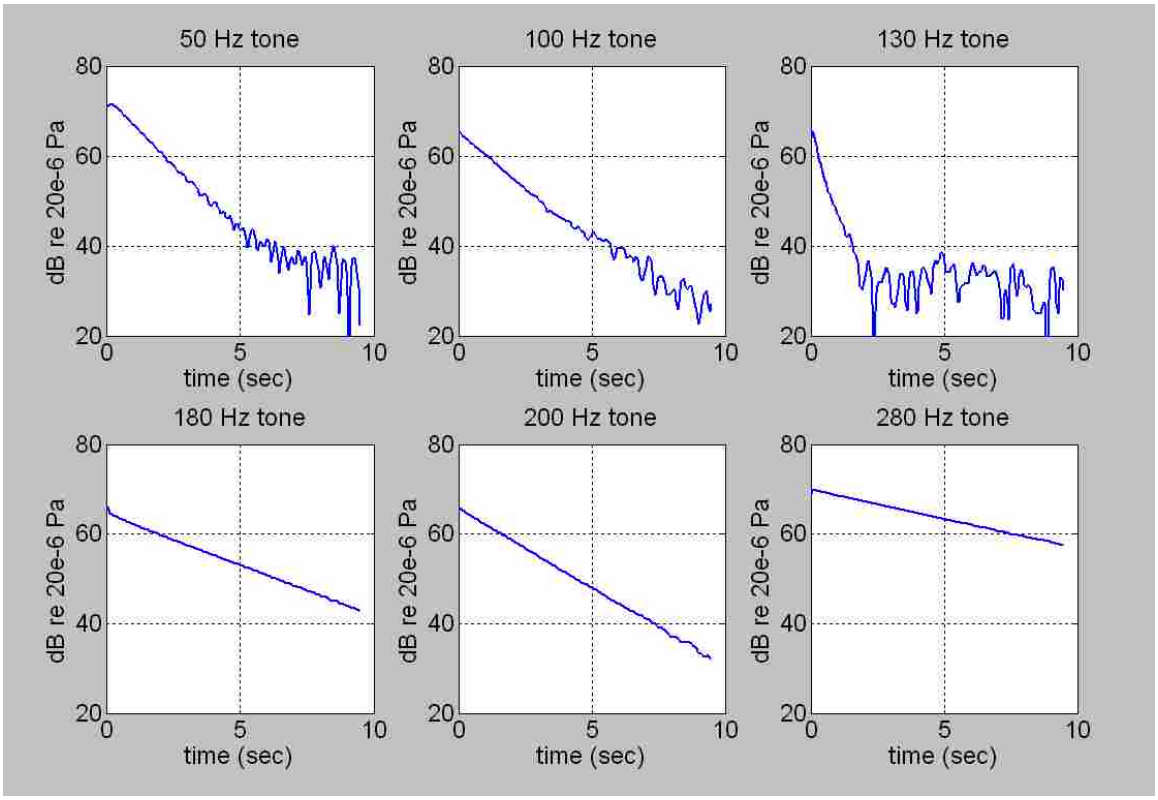

Figure 5.24 Individual learning curves for tones in noise signal for ANC with EE-FXLMS algorithm using 256-coefficient genetic Sys ID model and weighted reference.

Table 5.10 Eigenvalue pairs for each tone and eigenvalue span for all tones for 256-coefficient genetic Sys ID model and weighted reference. Color coded phase error at each tone given by: Green Φ<40º, Yellow 40º ≤ Φ < 90º, Red Φ ≥ 90º.

Tone (Hz)	50	100	130	180	200	280	Span
Eigenvalues	146.3 150.7	78.0 78.3	151.1 137.8	107.0 100.7	68.7 67.9	64.1 65.5	2.4
Phase Error							

5.4 Comparison of ANC Tests

The results from several ANC test cases from the previous sections are plotted together here for a more direct comparison. In Figures 5.25-5.28 the following test cases are plotted together to compare the various methods of modifying the magnitude coefficients in the EE-FXLMS algorithm:

(1) Flattened Sys ID with weighted reference. This combination may give lower eigenvalue span when used with weighted reference, but will not be optimal. There is no assurance that there is no significant phase error between frequency bins.

(2) X-inverse Sys ID with weighted reference. This combination may give lower eigenvalue span when used with weighted reference, but will not be optimal unless tones fall exactly on frequency bin values. There is no assurance that there is no significant phase error between frequency bins.

(3) Genetic Sys ID with a weighted reference signal. This is the closest to an ideal case that has been considered here. This minimizes eigenvalue span with an assurance that significant phase errors will not be introduced, keeping the new Sys ID model stable.

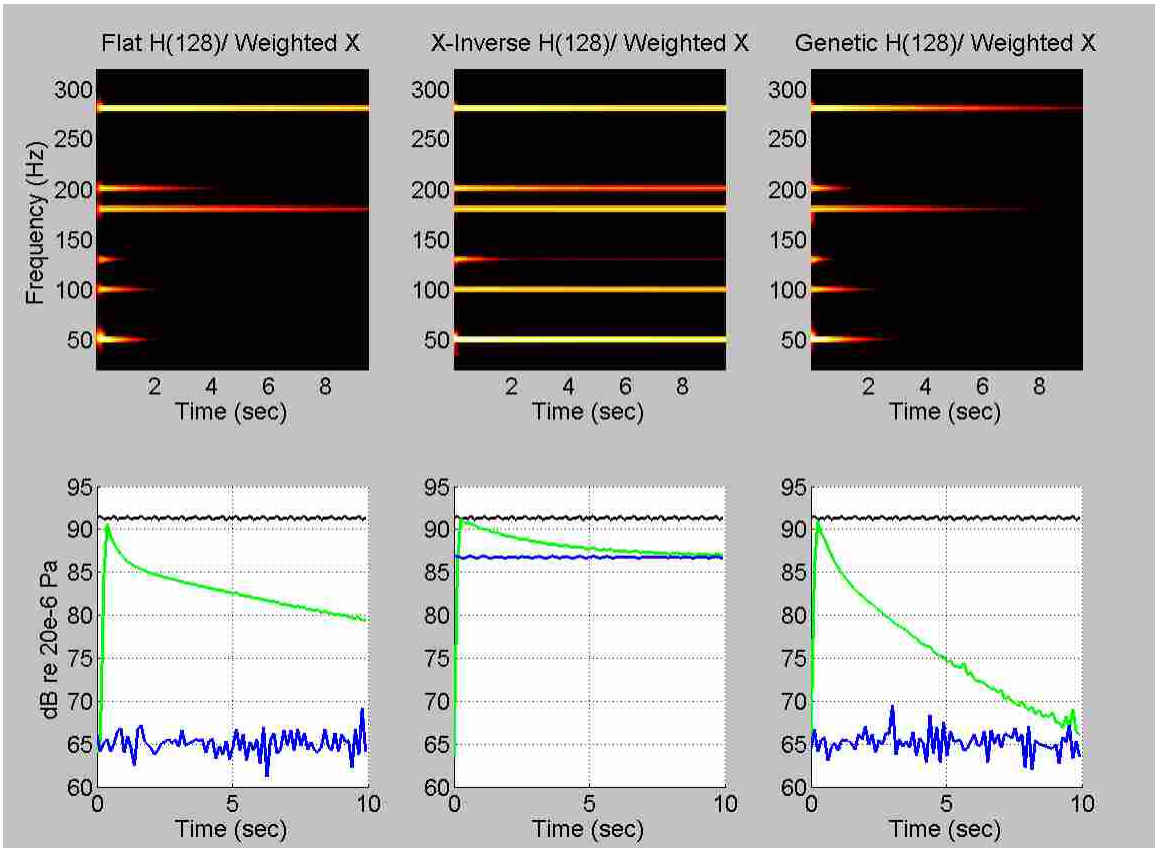

Figure 5.25 Spectrogram and plots of three time captures (paired vertically) for the 128-coefficient Sys ID model ANC tests for EE-FXLMS algorithm with flat, x-inverse and genetic Sys ID models and weighted reference.

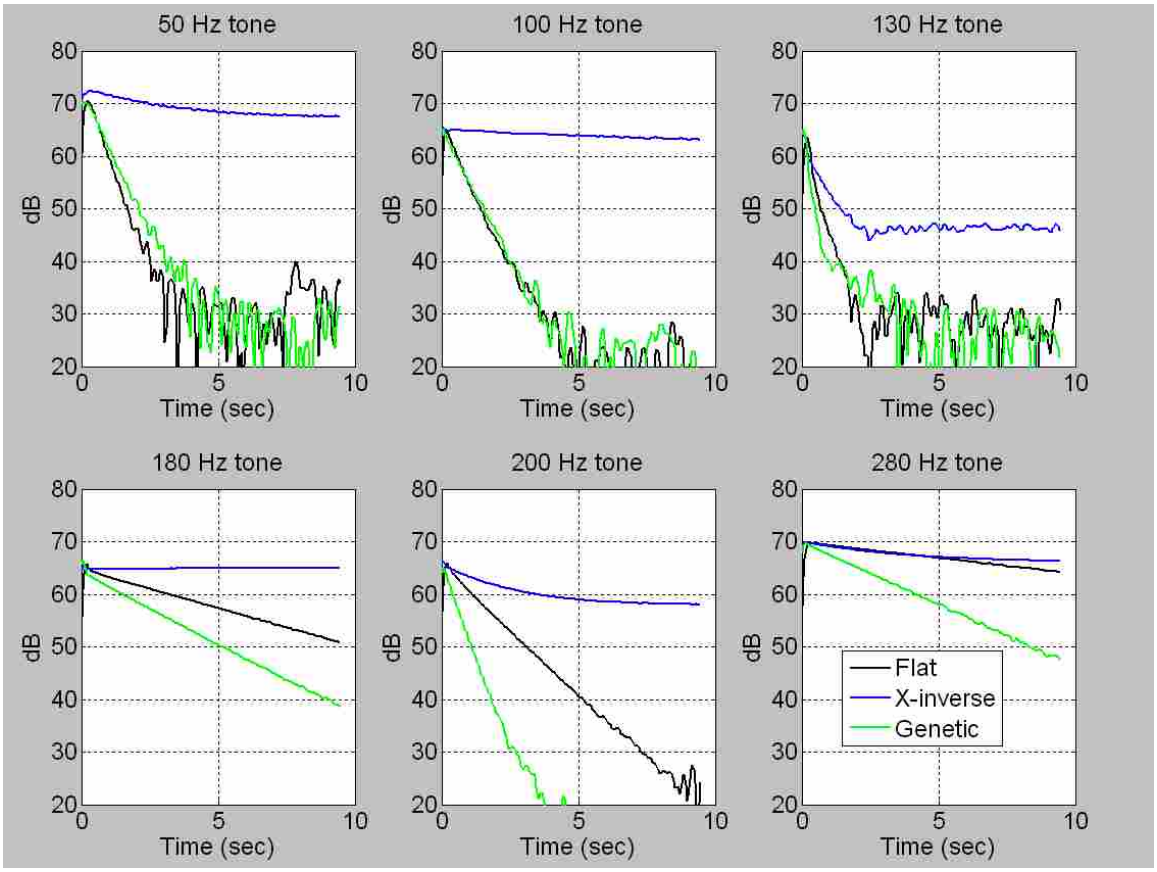

Figure 5.26 Learning curves for individual tones for EE-FXLMS algorithm with 128-coefficient flat, x-inverse and genetic Sys ID models and weighted reference.

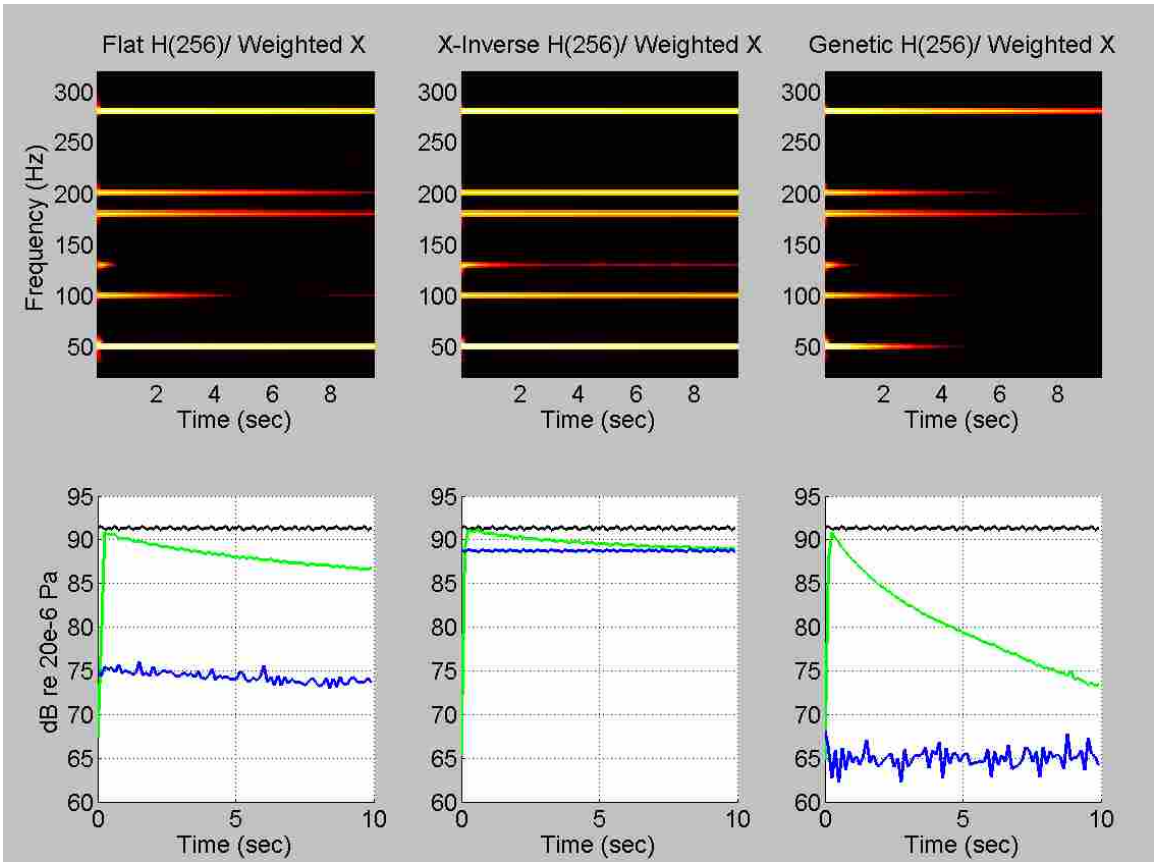

Figure 5.27 Spectrogram and plots of three time captures (paired vertically) for the 256-coefficient Sys ID model ANC tests for EE-FXLMS algorithm with flat, x-inverse and genetic Sys ID models and weighted reference.

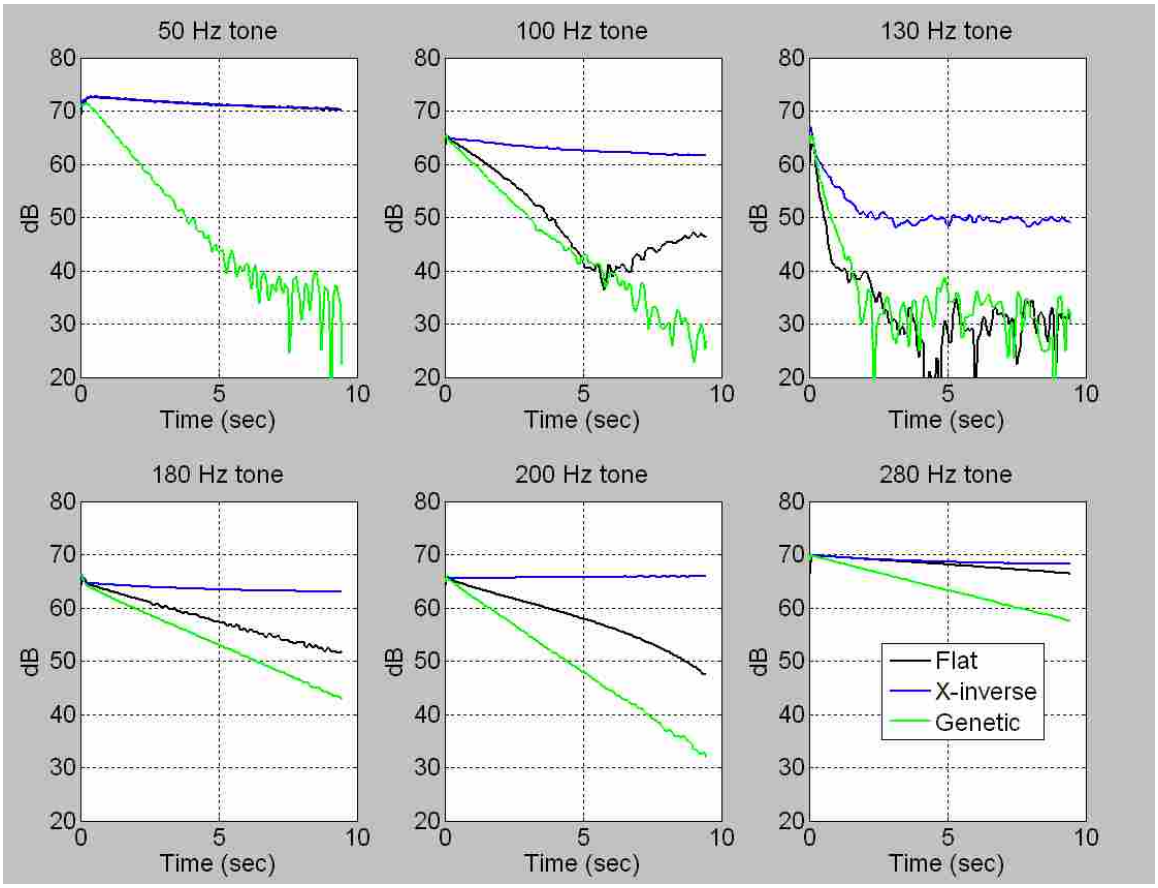

Figure 5.28 Learning curves for individual tones for EE-FXLMS algorithm with 256-coefficient flat, x-inverse and genetic Sys ID models and weighted reference.

It is evident that for both 128 and 256-coefficient Sys ID models, the genetically optimized magnitude coefficients give the best ANC performance in terms of rate of convergence for the combined error signal and convergence of individual tones, as well as lowest overall attenuation in the steady state solution.

In Figures 5.29-5.32 the following test cases are plotted together:

(1) FXLMS algorithm using original Sys ID with weighted reference. This is the baseline measurement showing the performance one would expect from an ANC system running the original FXLMS algorithm and an unconditioned reference.

(2) Flat Sys ID with equal amplitude tones in reference. This combination is expected to give low eigenvalue span and better ANC performance inasmuch as the magnitude does not deviate significantly from flat between bins, but it does require that the reference signal be conditioned. There is no assurance that there is no significant phase error between frequency bins.

(3) Genetic Sys ID with a weighted reference signal. This is the closest to an ideal case that has been considered here. This minimizes eigenvalue span with an assurance that significant phase errors will not be introduced, keeping the new Sys ID model stable. No conditioning of the reference is required.

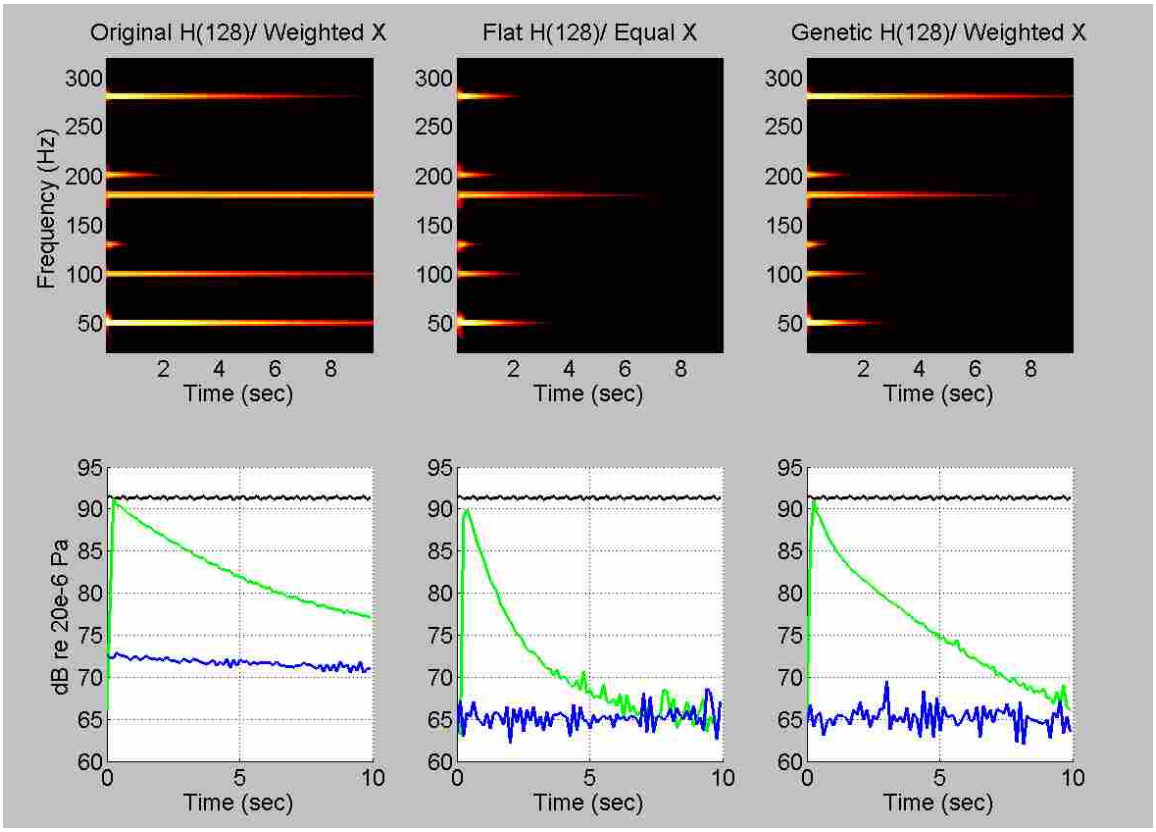

Figure 5.29 Spectrogram and plots of three time captures (paired vertically) for the 128-coefficient Sys ID model ANC tests for FXLMS algorithm using original Sys ID model and weighted reference, EE-FXLMS algorithm with flat Sys ID model and reference with equal amplitude tones, and EE-FXLMS using genetic Sys ID model and weighted reference.

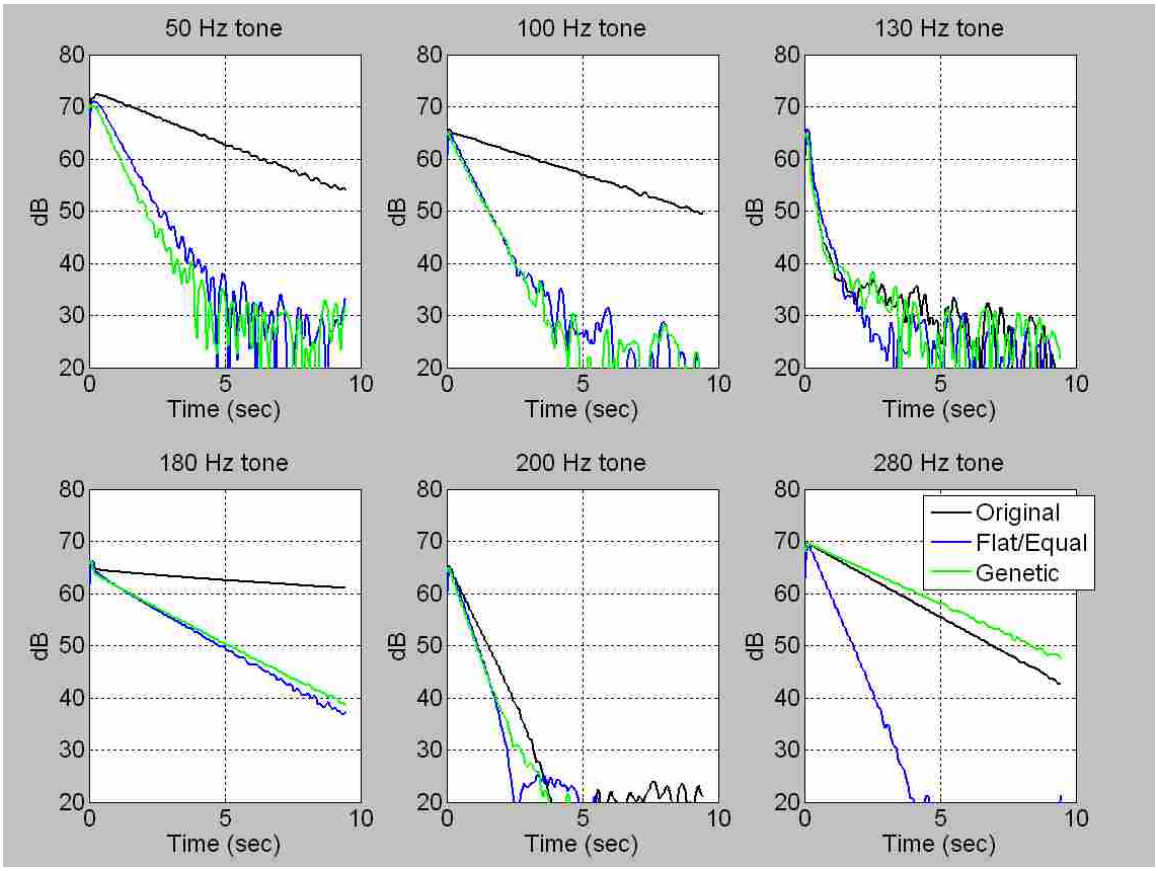

Figure 5.30 Learning curves for individual tones for EE-FXLMS algorithm with 128-coefficient Sys ID model ANC tests for FXLMS algorithm using original Sys ID model and weighted reference, EE-FXLMS algorithm with flat Sys ID model and reference with equal amplitude tones, and EE-FXLMS using genetic Sys ID model and weighted reference.

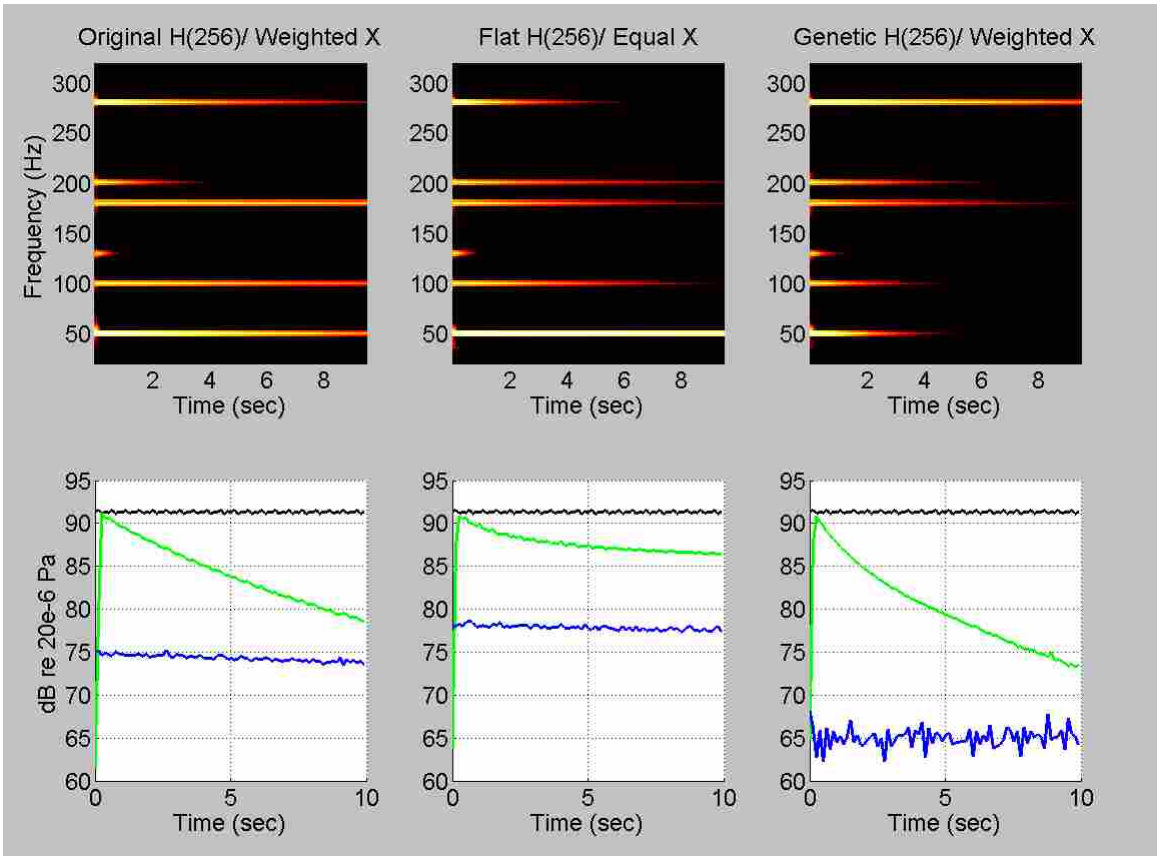

Figure 5.31 Spectrogram and plots of three time captures (paired vertically) for the 256-coefficient Sys ID model ANC tests for FXLMS algorithm using original Sys ID model and weighted reference, EE-FXLMS algorithm with flat Sys ID model and reference with equal amplitude tones, and EE-FXLMS using genetic Sys ID model and weighted reference.

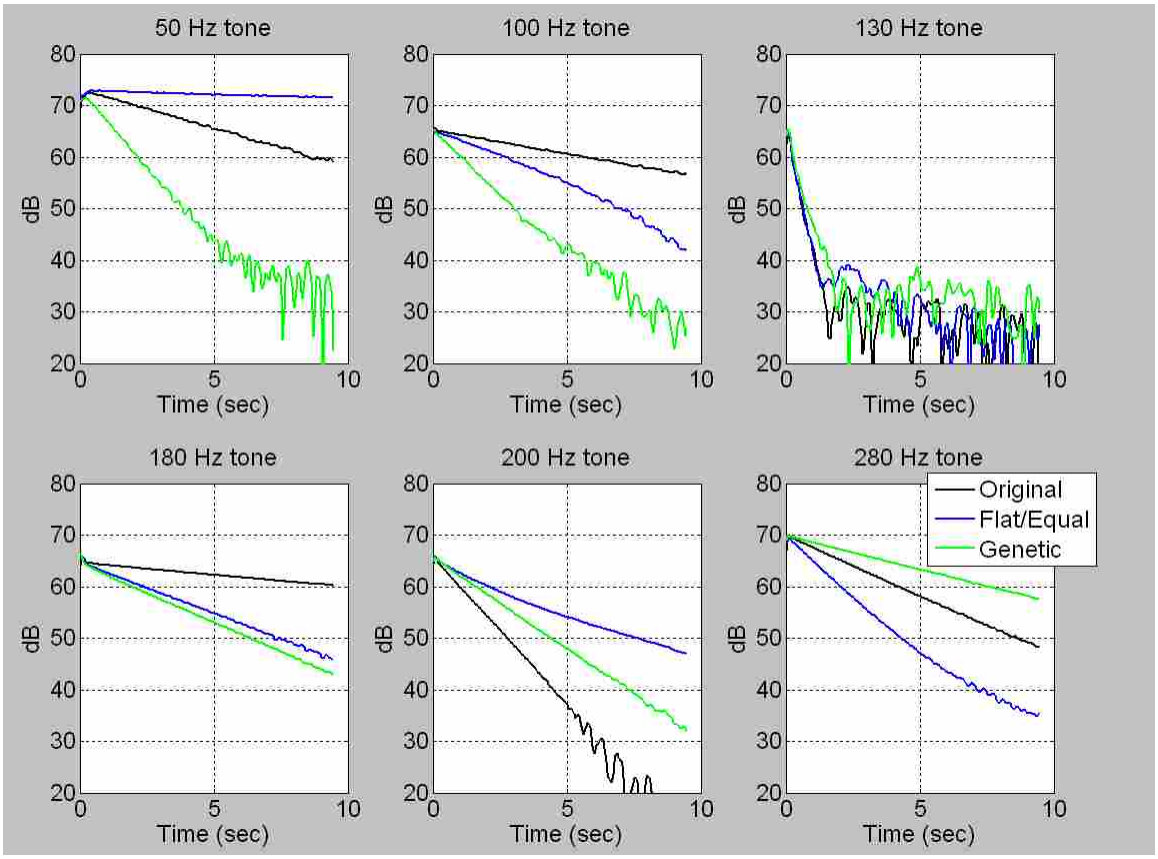

Figure 5.32 Learning curves for individual tones for EE-FXLMS algorithm with 256-coefficient Sys ID model ANC tests for FXLMS algorithm using original Sys ID model and weighted reference, EE-FXLMS algorithm with flat Sys ID model and reference with equal amplitude tones, and EE-FXLMS using genetic Sys ID model and weighted reference.

For the 128- coefficient test cases, the flat Sys ID model paired with a reference with equal amplitude tones converged fastest. The genetic test case was the next best in terms of convergence and performed about the same as the flat test case in terms of the overall steady state attenuation. The FXLMS had the poorest performance.

For the 256-coefficient tests, the EE-FXLMS algorithm with the genetically optimized Sys ID model performed the best in terms of convergence and overall attenuation (compare blue, steady state ANC time captures on Figure 5.31) . For this test, the flat/equal test performed worse than the FXLMS algorithm.

The EE-FXLMS algorithm using a Sys ID model that has been optimized using a genetic algorithm provides 6.4 dB additional overall attenuation for the 128-coefficient case and 9.2 dB for the 256-coefficient case compared to standard implementation of the FXLMS algorithm. The convergence time is reduced from 3.9 seconds to 1.0 seconds and from 6.3 seconds to 2.2 seconds for the 128 and 256-coefficient test cases respectively. The measures of performance for all results in Sections 5.2-5.4 are summarized in Table 5.11.

Table 5.11 Summary of ANC test results for all configurations.

Sys ID Model	Reference (off bins)	H taps	μ value (1/10 max stable)	Eigenvalue Span	Convergence Time (1/e) (sec)	Overall Reduction (dB)
original	weighted	128	5.E-11	217.0	3.9	19.5
original	weighted	256	3.E-11	239.0	6.3	16.9
flat	weighted	128	4.E-10	53.7	1.3	26.0
flat	equal	128	6.E-10	3.4	1.0	25.8
flat	weighted	256	2.E-10	134.2	9.0	16.9
flat	equal	256	2.E-10	55.7	5.8	13.4
x-inverse	weighted	128	5.E-11	16.0	10+	4.6
x-inverse	weighted	256	2.E-11	991.0	10+	2.6
genetic	weighted	128	9.E-11	2.3	1.0	25.9
genetic	weighted	256	5.E-11	2.4	2.2	26.1

CHAPTER 6- CONCLUSIONS AND RECOMMENDATIONS

6.1 Summary of Conclusions

A genetic algorithm has been used to find optimum values for the magnitude coefficients of the secondary path estimate as part of the Eigenvalue Equalization Filtered-x Least Mean Square algorithm applied to noise with multiple stationary tones. It has been shown to reduce the variation in the eigenvalues of the filtered-x autocorrelation matrix while preserving the phase of the original secondary path estimate. It has been shown to be superior to other methods of equalization applied to the EE-FXLMS algorithm, in that the optimization can overcome difficulties arising from the finite precision of the sampled ANC system and account for changes to the magnitude and phase that occur in between frequency bin values. Previously the optimization methods did not account for changes to the phase in these regions where it cannot be specifically defined by the user.

The added constraint that the zero-padded phase of the modified secondary path estimate not exceed 40º error from the original zero padded phase near tonal frequencies limits somewhat the ability of the optimization to reduce the eigenvalue disparity. Relaxing this constraint gives a lower eigenvalue span, but does not result in a better design since when used in ANC the phase errors will limit control performance and/or the design may be too sensitive to small shifts in tonal frequencies. Because both the flat and the x-inverse method of equalizing the eigenvalues cannot guarantee that the optimized design will remain stable and can actually increase the eigenvalue disparity, the genetic optimization is always preferred. Additionally, the genetic optimization is able to reduce

eigenvalue span and increase ANC performance without the need for extensive conditioning of the reference signal to make the tones in the reference for multiple tone noise have a user defined amplitude.

ANC in a mock cab using EE-FXLMS control with the genetic $\hat{\mathbf{h}}(t)$ model provided as much as 6-9 dB additional attenuation over FXLMS control with the original $\hat{\mathbf{h}}(t)$ model, with faster convergence times of 1.0-2.2 seconds compared to 3.9-6.3 seconds. Convergence of individual tones is more uniform giving less variable convergence in the combined noise signal. Stability of the algorithm is ensured in the optimization.

The faster convergence of the EE-FXLMS is an expected result of the eigenvalue equalization. Equalizing the eigenvalues should give faster or more uniform convergence for all tones. The additional attenuation reported for the EE-FXLMS algorithm with genetic Sys ID model is not an expected result but can be explained in context of the experiment. It is expected that the FXLMS using an unmodified Sys ID model would eventually reach the same steady state level as the EE-FXLMS with a genetic Sys ID model, just later according to the slower convergence rate of some of the tones. As was explained in Section 5.1 the FXLMS never reached a steady state solution during the ANC tests in the mock cabin since convergence of some tones was too slow to reach a steady state level before the wav file repeated. Were this not the case, it is believed that the FXLMS would have eventually converged to a steady state solution giving more similar mean square error levels in the steady state to the EE-FXLMS. However, in practice the EE-FXLMS algorithm may indeed provide more attenuation since no real ANC system would be perfectly stable and the discontinuity in the wav file that kept the

FXLMS from fully converging to a steady state solution may be typical of other such anomalies in a real physical system.

Use of a genetic algorithm as an optimization method in implementing the EE-FXLMS algorithm extends its utility and increases the potential benefit of its use over the FXLMS algorithm. The complexity of the EE-FXLMS algorithm that runs in real-time is identical to the FXLMS algorithm. Some additional offline computation is required to characterize the reference signal and perform the optimization of the secondary path used in control, as discussed in Section 3.2.2.

6.2 Recommendations for Future Work

This work has demonstrated that significant improvements to performance of ANC systems can be achieved while maintaining relatively simple implementation for noise with multiple quasi-stationary tones. This result is limited to ANC applications where the frequencies of tones in the noise do not vary significantly in time. Many important applications are of this type. The genetic optimization was done to guarantee that the modified genetic Sys ID model would remain stable for shifts in frequency of +/- 5 Hz from the normal operating frequencies for all tones. Shifts of +/- 5 Hz in the fundamental frequency of a set of harmonically related tones in the noise would mean much larger shifts in frequency for higher harmonics. Knowledge of the amount of frequency shift expected from a particular noise problem could be better incorporated into the genetic optimization. It is expected that the larger the frequency range over which the genetic algorithm optimization is constrained to keep the phase errors small, the more it will be limited in its ability reduce the eigenvalue span.

This thesis did not consider multiple tone noise where the tones do vary significantly in frequency. There are many applications where this would be the case, such as controlling the fundamental and harmonics of an engine that change with the engine rpm and do not have equal amplitudes in the reference signal. The EE-FXLMS algorithm has not been applied to this kind of noise problem.

The optimization performed on the secondary path estimate in the EE-FXLMS algorithm in this thesis is also limited to applications where the secondary path model, (at least the phase response) is relatively stable since the secondary path is only characterized and the optimization performed as part of the set up of an ANC system. Further work could be done to implement the EE-FXLMS algorithm with genetic optimization for a changing secondary path. This may be accomplished in several ways. The secondary path transfer function may be characterized for several normal operating conditions of a given ANC system. The eigenvalue equalization using the genetic algorithm can be performed for each of these expected operating conditions as part of the set up. These multiple optimized secondary path estimates can be stored in a "look up table". The system could be made to sense which previously optimized secondary path best fits the current operating conditions and use the best fit from the designs stored in the look up table.

Another way to extend the eigenvalue equalization idea for a system with changing secondary path would be to incorporate the optimization in an online Sys ID routine[35]. The secondary path can be characterized online periodically and the eigenvalue equalization performed in the background while control is running. Every time a newly optimized secondary path model becomes available it can be updated and used to run

control. The time it would take to get a new optimized model for the secondary path estimate would be set by the time it takes for the genetic algorithm to execute.

REFERENCES

1 Scott D. Snyder, Active Noise Control Primer, (Springer-Verlag New York, Inc., New York, NY 2000)

2 D.R. Morgan, "An analysis of multiple correlation cancellation loops with a filter in the auxiliary path," IEEE Trans. Acoust. Speech, Sig. Process., **ASSP-28**, 454-467 (1980)

3 B. Widrow and S.D. Stearns, *Adaptive Signal Processing* (Prentice-Hall Inc., Englewood Cliffs, NJ, 1985), Chapter 3 pp. 33-43.

4 G.Chen, M. Abe, and T. Sone, "Evaluation of the convergence characteristics of the filtered-x LMS algorithm in the frequency domain," J.Acoust. Soc. Jpn. (E) **16**, 6 (1995)

5 S. M. Kuo and D. R. Morgan, Active Noise Control Systems: *Algorithms and DSP Implementations*, edited by John G. Proakis (Wiley., New York, 1996), Chap. 2, pp. 33-35

6 R. L. Clark and G. P. Gibbs, "A novel approach to feedforward higher-harmonic control", J. Acoust. Soc. Am. **96**, 926-936 (1994)

7 S. M. Lee H. J. Lee C. H. Yoo Dae Hee Youn and I. W. Cha, "An Active Noise Control Algorithm for Controlling Multiple Sinusoids," J. Acoust. Soc. Am., vol. 104 no. 1 pp. 248-254, Jul. 1998.

8 S. M. Kuo, X. Kong, S. Chen, and W. Hao, "Analysis and design of narrowband active noise control systems," IEEE, 3557-3560 (1998)

9 S. M. Kuo, M. Tahernezhadi, and W. Hao, "Convergence analysis of narrow-band active noise control system", IEEE Trans. on circuits and systems—II: Analog and digital signal processing, **46**, (1999)

10 L. Vicente and E. Masgrau, "Fast convergence algorithms for active noise control in vehicles," Proc. of Joint ASA/EAA/DEGA Meeting, Berlin (1999)

11 L. Vicente and E. Masgrau, "Performance comparison of two fast algorithms for active control," Proc. of ACTIVE 99, pp. 1089-1100 (1999).

12 M. Rupp and A. H. Sayed, "Modified FXLMS algorithms with improved convergence performance," IEEE Proc. of ASILOMAR-29 (1995)

13 S. C. Douglas, "An efficient implementation of the modified filtered-x LMS algorithm," IEEE Signal Processing Letters **4**(10), (1997).

14 P. A. C. Lopes, M. S. Piedade, "Effects of secondary path modeling errors on the modified FX-LMS algorithm for active noise control," IEEE Proc. ICASSP '01, (2001).

15 Thomas, J.K., *Secondary Path Transfer Function Improvements for Enhanced Tracking Capabilities and Convergence of Filtered-x LMS Based Active Control Algorithms*, MS Thesis in *Mechanical Engineering*. 2007, Brigham, Young University: Provo

16 S. J. Elliott and P. A. Nelson, "Active Noise Control," IEEE Signal Processing Mag. **10**, pp.12-35 (1993)

17 B. Faber and S. D. Sommerfeldt, "Global control in a mock tractor cabin using energy density," Proc. ACTIVE '04 (Sept), edited by Randolph H. Cabell, and George C. Maling, Jr. (1994), a04_039(10)

18 C. C. Boucher., S. J. Elliott, and K. H. Baek, "Active control of helicopter rotor tones", Proc. of Inter-noise Liverpool '96, 1179-1182, (1996)

19 C. D. Kestell, *Active control of sound in a small single engine aircraft cabin with virtual error sensors*, (Ph. D thesis, Dept. of Mechanical Engineering, Adelaide University, South Australia 5005, Australia, 2000)

20 S. M. Kuo and D. R. Morgan, Active Noise Control Systems: *Algorithms and DSP Implementations*, edited by John G. Proakis (Wiley., New York, 1996), Chap. 9, pp. 275-310.

21 J. Gorman, R. Hinchliffe, and I. Stothers, "Active Sound Control on the Flight Deck of a C130 Hercules", Proc. ACTIVE '04 Williamsburg, VA, (2004)

22 T. A. Millot, and W. A. Welsh, "Helicopter Active Noise and Vibration Reduction", Sikorsky Aircraft Corporation, Twenty-fifth Rotorcraft Forum, Paper no. G23, Sept. 14-16 1999.

23 T.L. Lagö, "Frequency Analysis of Helicopter Sound in the AS332 Super Puma", Report 96-8, ISSN 1103-1581, ISRN HKR-RES--96/8—SE, (1996)

24 B. Widrow and S.D. Stearns, *Adaptive Signal Processing* (Prentice-Hall Inc., Englewood Cliffs, NJ, 1985), Chapters 3-5

25 C. C. Boucher, S. J. Elliot, and P. A. Nelson, "Effects of errors in the plant model on the performance of algorithms for adaptive feedforward control," IEE Proceedings-F 138(4), 313-319 (1991).

26 B. Widrow and S.D. Stearns, *Adaptive Signal Processing* (Prentice-Hall Inc., Englewood Cliffs, NJ, 1985), Chapter 3 pp. 40

27 B. Widrow and S.D. Stearns, *Adaptive Signal Processing* (Prentice-Hall Inc., Englewood Cliffs, NJ, 1985), Chapter 5 pp. 88

28 Muhammad Tahir Akhtar, Masahide Abe, Masayuki Kawamata, On Active Noise Control Systems With Online Acoustic Feedback Path Modeling, IEEE Transactions on Audio, Speech, and Language Processing, 1558-7916 / 2006

29 C. H. Hansen, Understanding Active Noise Cancellation (Spon Press, 2001) pp. 83-85.

30 S. M. Kuo and D. R. Morgan, Active Noise Control Systems: *Algorithms and DSP Implementations*, edited by John G. Proakis (Wiley., New York, 1996), Chap. 4, pp. 130-141.

31 B. Widrow and S.D. Stearns, *Adaptive Signal Processing* (Prentice-Hall Inc., Englewood Cliffs, NJ, 1985), Chapters 5, pp. 89

32 L. Davis, Handbook of Genetic Algorithms (Van Nostrand Reinhold, 1991).

33 D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning (Addison Wesley, 1989).31

34 Parkinson, A., *Optimization Methods in Engineering Design*. ME 575 Course Notes, 2007. Brigham Young University.

35 S. M. Kuo and D. R. Morgan, Active Noise Control Systems: *Algorithms and DSP Implementations*, edited by John G. Proakis (Wiley., New York, 1996), Chap. 72, pp. 213-240