Eigenvalue equalization filtered-x algorithm for the multichannel active noise control of stationary and nonstationary signals

Jared K. Thomas
Department of Mechanical Engineering, Brigham Young University, 435 CTB, Provo, Utah 84602

Stephan P. Lovstedt
Department of Physics and Astronomy, Brigham Young University, N283 ESC, Provo, Utah 84602

Jonathan D. Blotter
Department of Mechanical Engineering, Brigham Young University, 435 CTB, Provo, Utah 84602

Scott D. Sommerfeldt
Department of Physics and Astronomy, Brigham Young University, N283 ESC, Provo, Utah 84602

(Received 9 August 2007; revised 8 February 2008; accepted 6 March 2008)

The FXLMS algorithm, which is extensively used in active noise control, exhibits frequency dependent convergence behavior. This leads to degraded performance for time-varying and multiple frequency signals. A new algorithm called the eigenvalue equalization filtered-*x* least mean squares (EE-FXLMS) has been developed to overcome this limitation without increasing the computational burden of the controller. The algorithm is easily implemented for either single or multichannel control. The magnitude coefficients of the secondary path transfer function estimate are altered while preserving the phase. For a reference signal that has the same magnitude at all frequencies, the secondary path estimate is given a flat response over frequency. For a reference signal that contains tonal components of unequal magnitudes, the magnitude coefficients of the secondary path are adjusted to be the inverse magnitude of the reference tones. Both modifications reduce the variation in the eigenvalues of the filtered-*x* autocorrelation matrix and lead to increased performance. Experimental results show that the EE-FXLMS algorithm provides 3.5–4.4 dB additional attenuation at the error sensor compared to normal FXLMS control. The EE-FXLMS algorithm's convergence rate at individual frequencies is faster and more uniform than the normal FXLMS algorithm with several second improvement being seen in some cases. © *2008 Acoustical Society of America.*
[DOI: 10.1121/1.2903857]

PACS number(s): 43.50.Ki, 43.40.Vn [KAC] Pages: 4238–4249

I. INTRODUCTION

An active noise control (ANC) system relies on the theory of superposition of sound waves—propagating waves can constructively and destructively interfere to either increase or decrease the sound, respectively. Applications of ANC are widespread but can, in general, be categorized into two types of controllable signals: signals which are stationary in time and signals which are nonstationary or time varying. Signals of both types may be single frequency, multiple frequency, broadband, or some combination of these three. The most common control approach for the ANC of these signals is based on some version of the filtered-*x* least mean squares (FXLMS) algorithm.[1,2] The FXLMS algorithm has proven successful for applications such as single frequency noise in a duct,[2] broadband noise in an enclosure,[3] multiple frequency noise in a helicopter,[4] and time-varying frequency noise in a tractor.[5]

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. The performance degradation is evident for the case of noise that is time varying, such as that of a tractor engine, where the frequency changes as the speed of the engine, in rpm, changes during operation. If the frequency associated with the engine speed changes faster than the algorithm can converge and attenuate that particular frequency, then performance of the ANC system will be degraded. The degradation is also evident for the case of stationary multiple frequency noise, such as that in a helicopter, where multiple harmonics of the engine, tail rotor, and main rotor can be controlled. Poor performance is expected at frequencies where the convergence of the algorithm is slow. The frequency dependent problem is not manifested for stationary single frequency noise, as optimal performance is still possible by the correct selection of the convergence parameter μ. Since μ also exhibits frequency dependence, optimal performance by selection of the correct μ is not guaranteed for the case of multiple stationary frequencies and time-varying frequencies.

Solutions to the frequency dependent problem for multiple stationary frequency noise have been proposed such as the higher harmonic filtered-*x* (HLMS) algorithm by Clark and Gibbs,[6] similar work by Lee *et al.*,[7] and the modified FXLMS algorithm.[8] The drawback of these approaches is that they add complexity and computational burden to the algorithm. The work of Kuo *et al.*[9,10] suggested a relatively simple solution for the case of internally generated sinusoids. More of their work will be discussed at a later point. For the

 0001-4966/2008/123(6)/4238/12/$23.00

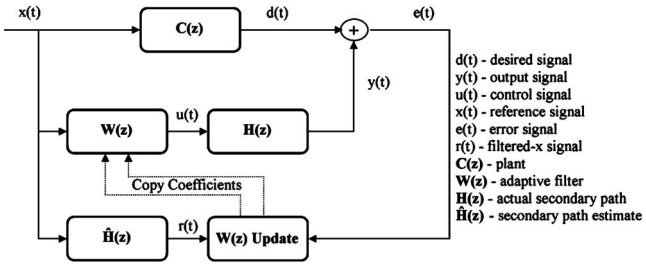

FIG. 1. Block diagram of the FXLMS algorithm.

case of time-varying frequencies, the filtered-x gradient adaptive lattice (FXGAL) algorithm by Vicente and Masgrau[11] improves the convergence behavior when an acoustic reference signal is used at the expense of computational complexity. For the case of a single time-varying frequency, the normalized FXLMS can be an effective solution.

This paper will discuss two simple approaches for dealing with noise characterized as multiple stationary or time-varying frequencies, which largely overcomes this frequency dependent performance and improves the overall performance of the ANC system. These approaches are appropriate for both single and multiple channel controls, are relatively simple to implement, and do not increase the computational burden of the algorithm. The effectiveness of these approaches will be experimentally demonstrated.

II. BACKGROUND

For this research, a feedforward multiple channel implementation of the FXLMS algorithm is used, which relies on a reference signal being "fed" forward to the control algorithm so that it can predict in advance the control signal needed to attenuate the unwanted noise. A feedforward implementation of the FXLMS algorithm involves adaptive signal processing to filter the reference signal in such a way that the measured residual noise is minimized. The measured residual is called the error signal and for this research it will be measured as an energy density (ED) quantity. The advantages of an ED based FXLMS algorithm[12] for noise in an enclosure[3,13] and for the application of tractor engine noise[5,14] are well documented. For simplicity in developing the control approaches, a brief derivation of the general FXLMS algorithm for a single channel is given. The extension of the approaches for multiple channel control[15] is straightforward. The use of an ED based FXLMS is also straightforward and well documented in Ref. 12.

A. Single channel FXLMS

The goal of the FXLMS algorithm is to reduce the mean-squared error of the error signal at a location where the sound is to be minimized. Boucher *et al.*[16] provided a good reference for the derivation of the single channel FXLMS algorithm, which is shown in block diagram form in Fig. 1. In the figure and in all equations presented, the variable t is used as a discrete time index and the variable z is used as a discrete frequency domain index. Signals in the time domain are represented as lower case letters, while capital letters are used in the frequency domain. Vectors in each domain are represented as bold letters.

The mean-squared error is a quadratic function (a "bowl") with a unique global minimum. For each iteration of the algorithm, $\mathbf{W}(z)$, which is represented as an adaptive finite impulse response (FIR) control filter, takes a step of size μ, the convergence coefficient, times the gradient in search of a single global minimum that represents the smallest attainable mean-squared error. The control filter update equation for $\mathbf{W}(z)$ can be expressed in vector notation as

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

where $e(t)$ is the error signal and $\mathbf{r}(t)$ and $\mathbf{w}(t)$ are defined as

$$\mathbf{r}^T(t) = [r(t), r(t-1), \ldots, r(t-I+1)], \tag{2}$$

$$\mathbf{w}^T(t) = [w_0, w_1, \ldots, w_{t-1}]. \tag{3}$$

The filtered-x signal $r(t)$ is the convolution of $\hat{\mathbf{h}}(t)$, which is the estimate of the impulse response of the secondary path transfer function, and $x(t)$, which is the reference signal. The secondary path transfer function includes the effects of digital-to-analog converters, reconstruction filters, audio power amplifiers, loudspeakers, the acoustical transmission path, error sensors, signal conditioning, antialias filters, and analog-to-digital converters. The reference signal contains information correlated with the unwanted noise that the ANC system will target when control is enabled.

B. Secondary path transfer function

One difficulty in implementing the FXLMS algorithm is that the secondary path, which is represented as $\mathbf{H}(z)$ in Fig. 1, is unknown. An estimate, $\hat{\mathbf{H}}(z)$, of the secondary path must be used. The estimate is obtained through a process called system identification (SysID).

The SysID process is performed either online (while ANC is running) or offline (before ANC is started). For the fastest convergence of the algorithm, an offline approach is used. The offline SysID process is performed before ANC is started and consists of playing white noise through the control speaker(s) and measuring the output at the error sensor. The measured impulse response is obtained as a FIR filter $\hat{\mathbf{h}}(t)$ that represents $\hat{\mathbf{H}}(z)$. The coefficients of $\hat{\mathbf{h}}(t)$ are stored and used to run control. For multiple channel control, there is an $\hat{\mathbf{h}}(t)$ estimate for every error sensor and control speaker combination. Each is obtained in turn through the SysID process.

C. Reference signal

The reference signal may be an acoustic signal (e.g., from a microphone) or a nonacoustic signal (e.g., a tachometer signal from an engine) depending on the control application. Generally, the reference signal will be either stationary or time varying. Signals of either type may be single frequency, multiple frequency, broadband, or some combination of these three.

Significant signal conditioning may be required to get the reference signal in a form suitable for control. For ex-

ample, for control of engine noise, a tachometer signal related to the engine speed (in rpm) is typically used as the reference signal. The tachometer signal is usually a multiple or some fraction of the engine firing frequency and must be filtered and passed through a frequency multiplier to be directly used as the reference signal. If harmonics are also targeted for control, they too are usually generated either in hardware or software from the fundamental frequency. Where multiple noise sources are present, a reference signal may be obtained for each and combined into a single reference. The resulting signal will, in general, have varying magnitude at the various tonal components.

III. FREQUENCY DEPENDENT CONVERGENCE BEHAVIOR

The inclusion of $\hat{\mathbf{H}}(z)$, while necessary for algorithm stability, degrades performance by slowing the algorithm's convergence. One reason for the decreased performance is the delay associated with $\hat{\mathbf{H}}(z)$. For many ANC applications, such as enclosures of less than a few meters, the delay is on the order of 10 ms or less and convergence is still rapid.[17] A more significant problem is that the inclusion of $\hat{\mathbf{H}}(z)$ causes a frequency dependent convergence behavior. The frequency dependent behavior can be better understood by looking at the eigenvalues of the autocorrelation matrix of the filtered-x signal, which is a function of $\hat{\mathbf{H}}(z)$ and $\mathbf{X}(z)$.

A. Eigenvalues

The eigenvalues of the autocorrelation matrix of the filtered-x signal relate to the dynamics or time constants of the modes of the system. Typically, a large spread is observed in the eigenvalues of this matrix, which corresponds to fast and slow modes of convergence. The slowest modes limit the performance of the algorithm because they determine the overall convergence of the algorithm to the optimum. The fastest modes have the fastest convergence and the greatest reduction potential but limit how large of a convergence parameter μ can be used.[18] For stability, μ is set based on the slowest converging mode (the maximum eigenvalue), which leads to degraded performance. If μ is increased, the slower modes will converge faster, but the faster modes will drive the system unstable.

The autocorrelation matrix of the filtered-x signal is defined as

$$E[\mathbf{r}(t)^{*}\mathbf{r}^{T}(t)], \tag{4}$$

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 $\mathbf{r}^{T}(t)$. In general, it has been shown that the algorithm will converge and remain stable as long as the chosen μ satisfies the following equation:[16]

$$0 < \mu < \frac{2}{\lambda_{\max}}, \tag{5}$$

where $\lambda_{\max}$ is the maximum eigenvalue of the autocorrelation matrix in the range of frequencies targeted for control.

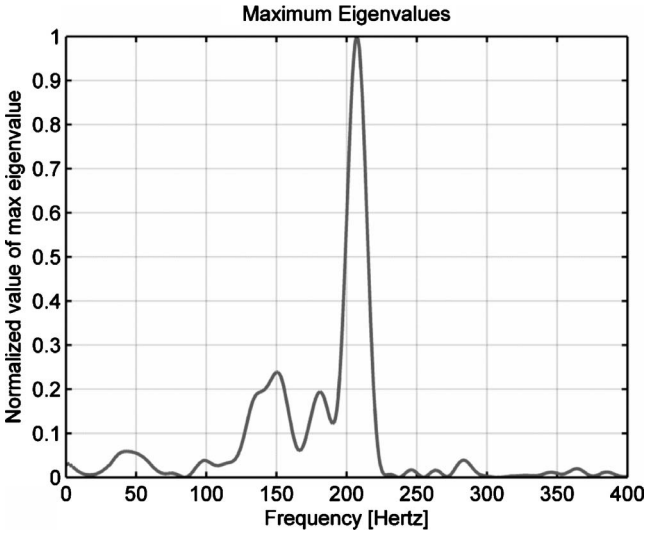

FIG. 2. Plot of normalized maximum eigenvalues over frequency—equally weighted reference signal.

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. An offline estimate of the autocorrelation matrix is made by taking an actual $\hat{\mathbf{H}}(z)$ model from a mock cabin and importing it into a numerical computer package. If a single frequency reference signal is used, $\lambda_{\max}$ can be computed for that frequency. If the simulation is repeated over a range of frequencies, $\lambda_{\max}$ for each frequency can be found. Figure 2 shows an offline simulation using an actual $\hat{\mathbf{H}}(z)$ from a mock cabin and equal amplitude tonal inputs from 0 to 400 Hz. The eigenvalues in the figure have been normalized to the largest eigenvalue in the range. In this eigenvalue simulation, and all others in this paper, the results are shown for only a single $\hat{\mathbf{H}}(z)$ for a single channel. The results for multiple channel and/or ED control [where the filtered-x signal is a combination of each $\hat{\mathbf{H}}(z)$ component of each channel] follow a similar trend as the single channel case and so, to facilitate explanation of the concepts presented in this paper, only a single $\hat{\mathbf{H}}(z)$ for a single channel is shown.

Figure 2 illustrates the frequency dependent behavior. The largest eigenvalue occurs at about 208 Hz. This location corresponds to the smallest stable μ in the frequency range from 0 to 400 Hz, as given by Eq. (5). Most other frequencies have a smaller 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 the figure 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 desirable as they lead to faster convergence and increased attenuation.

If the frequency range for control is 0–400 Hz, the μ associated with 208 Hz (the smallest in the range) must be used for stability. If, for example, 100 Hz was the only tone targeted for control, a μ larger than the 1 used at 208 Hz could be used and convergence would be faster. If both 100 and 208 Hz were targeted for control, the smaller μ associated with 208 Hz must be used for stability and degraded

performance at 100 Hz is expected. In summary, because the μ associated with the largest eigenvalue in the range of frequencies targeted for control must be used for stability, degraded performance is expected at the other frequencies in the range that would benefit from the use of a larger μ.

IV. EIGENVALUE EQUALIZATION

If the variance in the eigenvalues of the autocorrelation matrix was minimized, a single convergence parameter could then be chosen that would converge at nearly the same rate over all frequencies. As previously stated, the autocorrelation matrix is directly dependent on the filtered-x signal, which is computed by filtering the input reference signal $\mathbf{X}(z)$ with $\hat{\mathbf{H}}(z)$. Thus, to adjust the eigenvalues, changes can either be made to $\mathbf{X}(z)$ or to $\hat{\mathbf{H}}(z)$. For either $\mathbf{X}(z)$ or $\hat{\mathbf{H}}(z)$ changes must be carefully done so that control is not only still possible, but at worst, it is as good as if they were left unmodified.

The choice of whether to adjust $\mathbf{X}(z)$ or $\hat{\mathbf{H}}(z)$ largely depends on the control application being investigated. For simplicity, it can be said that there are two possible cases: (1) applications where changes can easily be made to $\mathbf{X}(z)$ [leave $\hat{\mathbf{H}}(z)$ unmodified] and (2) applications where changes cannot be easily made to $\mathbf{X}(z)$ [modify $\hat{\mathbf{H}}(z)$]. The adverb "easily" is included in the previous sentence to emphasize that in some control cases, it may be a simple procedure to make adjustments to $\mathbf{X}(z)$, and for other control cases, although it may be possible to make changes to $\mathbf{X}(z)$, it may be a difficult or undesirable procedure. An example of the first case would be if the fundamental and higher harmonics of the reference signal were computer generated. If such was the case, it would be a straightforward process to digitally adjust the weightings of each tone in the signal. An example of the latter case would be if an acoustic reference signal was used that included several tones, each with a different amplitude. Adjustments to the weightings of individual tones could require significant signal processing, which may add an undesired complexity to the system.

A solution for case 1 was proposed by Kuo *et al.*[9,10] The solution is most applicable for stationary multiple frequency single channel control where the fundamental and harmonics are internally generated. A general solution for case 2 is presented in this paper as the eigenvalue equalization filtered-x least mean squares (EE-FXLMS) algorithm. The algorithm has been developed to handle both the case of multiple stationary frequency control and the case of time-varying frequency control for either single or multiple channel control. The EE-FXLMS algorithm has two unique implementations to handle the two possible conditions of the reference signal. These are the following: (1) the frequencies of interest in the reference signal are equally weighted or (2) the frequencies of interest in the reference signal are unequally weighted.

A. Case 1 solution: The method of Kuo *et al.*

Kuo *et al.* observed that for multiple frequency control, the convergence parameter μ must be chosen to ensure that the system is stable at the frequency where the magnitude response of $\hat{\mathbf{H}}(z)$ is largest and that this causes the convergence at frequencies where the magnitude response of $\hat{\mathbf{H}}(z)$ is small to be slow.[9] They showed that if the amplitude of each frequency in the reference signal is optimized as the inverse of the magnitude response of $\hat{\mathbf{H}}(z)$ at that frequency, then the performance of the algorithm is greatly improved: the biggest improvement being seen at the frequencies where the magnitude response of $\hat{\mathbf{H}}(z)$ is small and convergence was originally slow. In terms of the filtered-x autocorrelation matrix eigenvalues, they show that the eigenvalue spread is close to 1 for the frequencies of interest, which should result in better convergence properties. This method was developed for single channel control.

A strength of their method is that it can be performed offline (before control is enabled) and thus does not increase the computational burden on the algorithm. One drawback is that, as they suggest, it is applicable for cases where frequency information is first obtained through a source such as a tachometer or accelerometer and is then used to digitally synthesize a reference signal that contains the fundamental frequency and appropriate harmonics.[9] Because the reference signal is digitally synthesized, it is a simple process to adjust the amplitude of each frequency in the reference signal to its optimal value. In many control applications, however, it is desirable to directly use the reference signal from its source, which makes the adjustment of the individual frequency amplitudes a more difficult task requiring extensive filtering and signal conditioning. Such might be the case when an acoustic reference signal is used. Directly using the reference signal from its source is especially important when time-varying frequencies are involved. For example, when controlling engine noise, it is desirable to directly use the tachometer signal so that the engine firing frequency and harmonics can be tracked and controlled as the speed of the engine changes during operation.

B. Case 2 solution: EE-FXLMS

Often times, it is either difficult or undesirable to alter the reference signal. Assuming that the reference signal is left unchanged, changes to the autocorrelation matrix must stem from changes to $\hat{\mathbf{H}}(z)$ but must be done carefully as any errors in its estimation already contribute to lower convergence rates and instability. Estimation errors can be considered in two parts: errors in the amplitude estimation and errors in the phase estimation.[19] It has been shown that phase estimation errors greater than $\pm 90°$ cause algorithm instability,[16] but errors as high as 40° have little effect on the performance.[16] Magnitude estimation errors can be compensated for by the choice of μ (Ref. 20), and consequently do not affect stability. Ideally, changes would be made to the magnitude information of $\hat{\mathbf{H}}(z)$, while the phase information is preserved. The method to equalize the eigenvalues of the autocorrelation matrix by changing the magnitude information of $\hat{\mathbf{H}}(z)$ while preserving the phase information will be referred to as the EE-FXLMS algorithm.

Two implementations of the EE-FXLMS algorithm have been developed to handle the two possible conditions of the

reference signal. These are the following: (1) the frequencies of interest in the reference signal are equally weighted or (2) the frequencies of interest in the reference signal are unequally weighted. If the frequencies of interest in the reference signal are equally weighted, it is proposed that the magnitude coefficients of $\hat{\mathbf{H}}(z)$ be optimized to also have an equal (flat) weighting over frequency. One example of a signal that would have an equal weighting over frequency would be the use of a tachometer signal to control the engine firing frequency of an engine. As the rpm of the engine changes during operation, the engine firing frequency will change, but the voltage level of the tachometer signal will remain constant. In other words, the amplitude of the reference signal will not change as the frequency changes. If the frequencies of interest in the reference signal are unequally weighted, it is proposed that the magnitude coefficients of $\hat{\mathbf{H}}(z)$ be optimized to have the inverse of the magnitude response of $\mathbf{X}(z)$. An example would be multiple frequency noise where the signal is nominally stationary, such as helicopter noise. With helicopter noise, each of the major noise sources (engine, main rotor, and tail rotor) will require a different reference signal obtained through a tachometer, photocell, or other method. Each of these signals will contain harmonics, each with a unique amplitude (generally, each successive higher harmonic will have a lower amplitude), resulting in an unequally weighted multiple frequency reference signal.

1. EE-FXLMS *Flat* $\hat{\mathbf{H}}(z)$ *implementation*

If all frequencies in $\mathbf{X}(z)$ are equally weighted, then Kuo *et al.*[9] suggested that μ must be chosen based on the frequency where the magnitude of $\hat{\mathbf{H}}(z)$ is the largest. This slows down the convergence at frequencies where the magnitude of $\hat{\mathbf{H}}(z)$ is small. This agrees with the eigenvalue simulation shown in Fig. 2. Note that in that simulation, each frequency was given an equal weighting. As previously mentioned, μ must be chosen based on the maximum eigenvalue in the frequency range of interest, and performance is degraded at frequencies where the eigenvalues are small. For the equally weighted $\mathbf{X}(z)$, the eigenvalue spread is mostly a function of the magnitude response of $\hat{\mathbf{H}}(z)$. This can be seen in Fig. 3. In Fig. 3, the magnitude coefficients of $\hat{\mathbf{H}}(z)$ are overlaid on the same plot of the maximum eigenvalues shown in Fig. 2. It can be seen that the magnitude coefficients are highly correlated with the eigenvalues. The maximum eigenvalue occurs where the response of $\hat{\mathbf{H}}(z)$ is large.

The data in Fig. 3 suggest that manipulating the magnitude coefficients of $\hat{\mathbf{H}}(z)$ should modify the eigenvalue spread. If the magnitude coefficients were "flat" over frequency, the eigenvalue spread should also be more flat over frequency. A method for flattening the magnitude coefficients has been developed, which is simple to implement and does not increase the computational burden of the algorithm.

The basic procedure for implementing the EE-FXLMS is to adjust the coefficients of $\hat{\mathbf{h}}(t)$ before ANC control is started as follows.

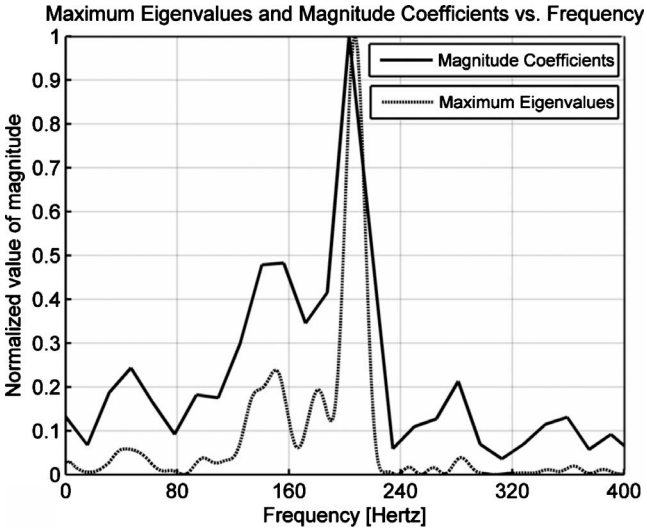

FIG. 3. Maximum eigenvalues and magnitude coefficients vs frequency for a mock cab—equally weighted reference signal.

(i) Obtain the time domain impulse response $\hat{\mathbf{h}}(t)$ for each $\hat{\mathbf{H}}(z)$ through an offline SysID process.
(ii) Take the fast fourier transform (FFT) to obtain $\hat{\mathbf{H}}(z)$.
(iii) Divide each value in the FFT by its magnitude and then multiply by the mean value of the FFT.
(iv) Compute the inverse FFT to obtain a new $\hat{\mathbf{h}}(t)$ and use the new modified $\hat{\mathbf{h}}(t)$ in the FXLMS algorithm as normal.

If using multiple channel and/or ED control, the process is repeated for each $\hat{\mathbf{H}}(z)$ estimate. This procedure flattens the magnitude coefficients of $\hat{\mathbf{H}}(z)$ while preserving the phase. It is an offline process directly done by following SysID and can be incorporated into any existing algorithm with only a few lines of code. As an offline process, it adds no computational burden to the algorithm while control is running. The results of the flattening process can be seen in Fig. 4. Figure 4 shows the original and modified $\hat{\mathbf{H}}(z)$ magnitude coefficients in the top plot and shows that the phase information of $\hat{\mathbf{H}}(z)$ has been preserved in the bottom plot. Note that the two lines representing the original and modified phase information of $\hat{\mathbf{H}}(z)$ are directly on top of each other in the bottom plot. The plots for the other $\hat{\mathbf{H}}(z)$ models for the other channels and ED components are similar.

An attempt to quantify any improvement in the eigenvalue spread has been made by using the following metrics:

(1) *Span*—λ_{max} divided by λ_{min}. Ideally 1.
(2) *rms value*—root mean square. Ideally 1.
(3) *Crest factor*—λ_{max} divided by rms value (how close the rms value is to the peak value) Ideally 1.

The effect of the flattening process on the eigenvalues can be seen in Fig. 5. The data for the figure were computed as before by an offline estimate of the autocorrelation matrix by using an actual $\hat{\mathbf{H}}(z)$ model from a mock cabin and finding the λ_{max} for each frequency from 0 to 400 Hz. The curve

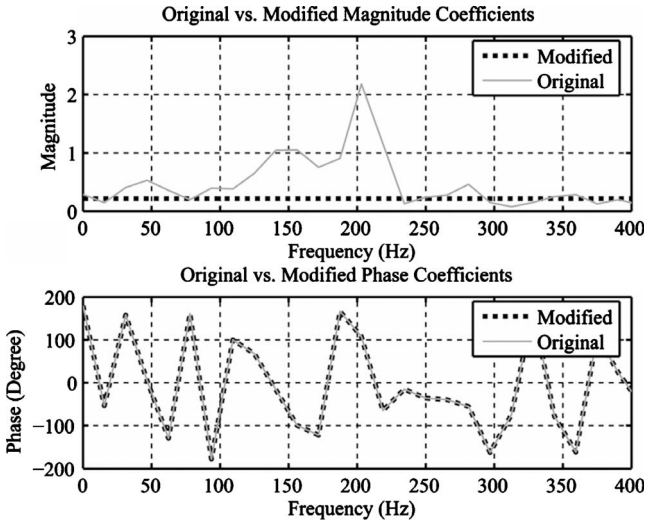

FIG. 4. Original and modified magnitude coefficients of $\hat{\mathbf{H}}(z)$–EE-FXLMS flat $\hat{\mathbf{H}}(z)$ implementation and the original and modified phase coefficients of $\hat{\mathbf{H}}(z)$–EE-FXLMS flat $\hat{\mathbf{H}}(z)$ implementation.

labeled "original" represents the same data shown in Fig. 2, and the curve labeled "modified" is an estimate of the eigenvalues by using the modified $\hat{\mathbf{H}}(z)$ model. In Fig. 5, the eigenvalues in both the original and modified cases have been normalized by the largest of the original eigenvalues. It is seen that the modified eigenvalues are more uniform ("equalized") over all frequencies. While the variation in the modified eigenvalues would ideally be zero, the decreased variation compared to the original eigenvalues should produce an observable performance advantage. The algorithm should converge at near the same rate over all frequencies and should not exhibit the frequency dependent behavior of the standard FXLMS.

Table I shows the improvement of the modified eigenvalues according to the defined metrics over the range from 0 to 400 Hz. The range from 0 to 400 Hz was selected because the experimental hardware has a cutoff frequency at 400 Hz.

TABLE I. Comparison of original and modified eigenvalues by using defined metrics—EE-FXLMS flat $\hat{\mathbf{H}}(z)$ implementation.

Metric	Original	Modified	% improvement
Span	2.37×10^5	2920	99
rms	0.19	0.3	58
Crest factor	5.283	3.413	35

In Table I it can be seen that the modified case has a lower span, a higher rms value, and a lower crest factor. In all three metrics, the values for the modified case are closer to the optimum values that would be present if the eigenvalues across all frequencies were exactly the same. These modifications to $\hat{\mathbf{H}}(z)$ should make a noticeable improvement in the performance of the algorithm.

2. EE-FXLMS $\hat{\mathbf{H}}(z)=1/|\mathbf{X}(z)|$ implementation

In this case, the reference signal is unequally weighted, and so the eigenvalue simulation shown in Fig. 2 must be redone by using an unequally weighted reference signal. For this simulation, the reference signal was chosen so that there was a linear descending trend in the amplitude of each successive frequency. Figure 6 shows the offline simulation by using an actual $\hat{\mathbf{H}}(z)$ from a mock cabin and tonal inputs from 0 to 400 Hz. The eigenvalues in the figure have been normalized to the largest eigenvalue in the range.

The same ideas behind flattening the magnitude coefficients of $\hat{\mathbf{H}}(z)$ when $\mathbf{X}(z)$ is equally weighted apply for the case when the frequencies in $\mathbf{X}(z)$ have unequal weighting. The magnitude coefficients of $\hat{\mathbf{H}}(z)$ at each frequency bin must be made to compensate for the unequal weighting of each tone in $\mathbf{X}(z)$ so that the eigenvalue spread becomes essentially flat over frequency. As with the flat $\hat{\mathbf{H}}(z)$ implementation, the phase information must be preserved.

When $\mathbf{X}(z)$ has unequally weighted frequencies, the magnitude coefficients of $\hat{\mathbf{H}}(z)$ are set to be $1/|\mathbf{X}(z)|$. This

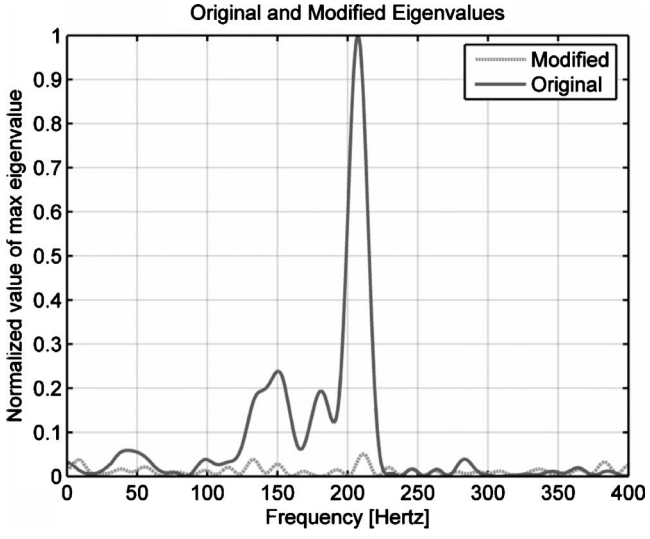

FIG. 5. Normalized original and modified eigenvalues—EE-FXLMS flat $\hat{\mathbf{H}}(z)$ implementation.

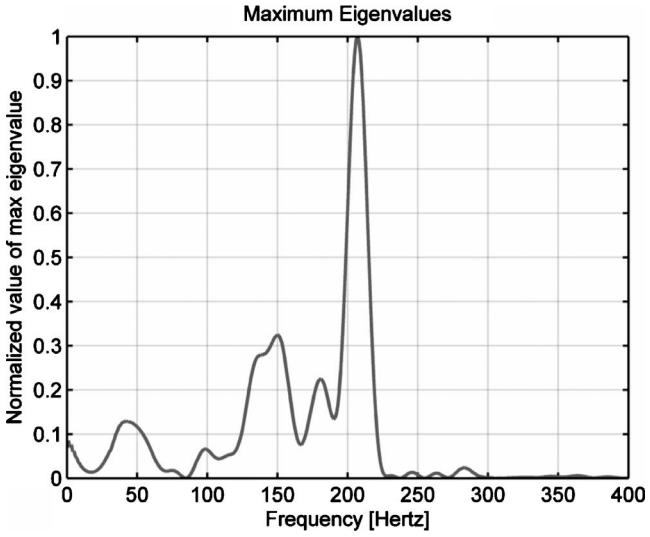

FIG. 6. Plot of normalized maximum eigenvalues over frequency—unequally weighted reference signal.

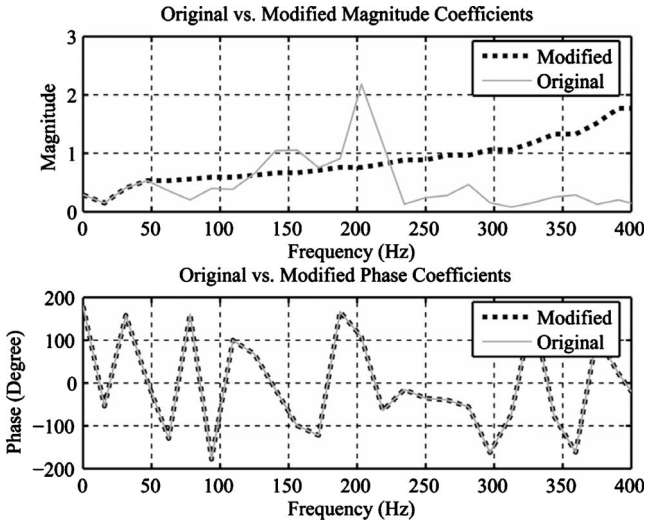

FIG. 7. Original and modified magnitude coefficients of $\hat{\mathbf{H}}(z)$–EE-FXLMS $\hat{\mathbf{H}}(z)=1/|\mathbf{X}(t)|$ implementation and the original and modified phase coefficients of $\hat{\mathbf{H}}(z)$–EE-FXLMS $\hat{\mathbf{H}}(z)=1/|\mathbf{X}(t)|$ implementation.

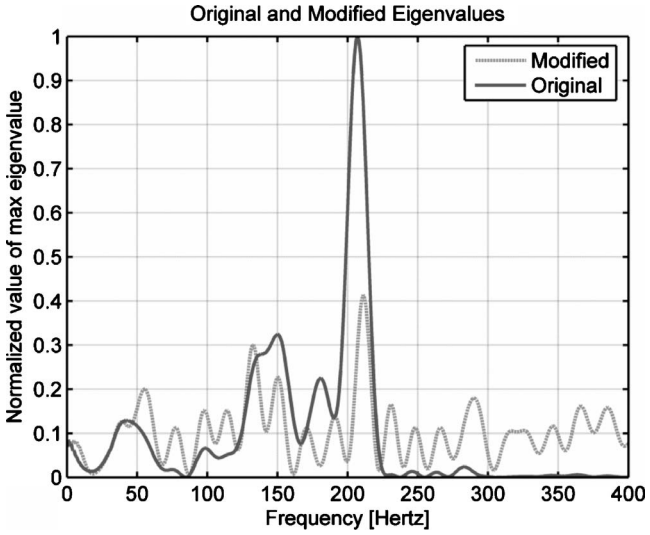

FIG. 8. Normalized original and modified eigenvalues—EE-FXLMS $\hat{\mathbf{H}}(z)=1/|\mathbf{X}(t)|$ implementation.

has the same effect as flattening the magnitude coefficients of $\hat{\mathbf{H}}(z)$ when $\mathbf{X}(z)$ has equally weighted frequencies. The method is simple to implement and does not increase the computational burden of the algorithm. The basic procedure is run before ANC control and is as follows:

(1) Obtain the time domain impulse response $\hat{\mathbf{h}}(t)$ for each $\hat{\mathbf{H}}(z)$ through an offline SysID process.
(2) Take the FFT to obtain $\hat{\mathbf{H}}(z)$.
(3) Compute the phase coefficients from the FFT of $\hat{\mathbf{H}}(z)$ and save them in a vector.
(4) Take a time sample of $\mathbf{x}(t)$ and compute its FFT, $\mathbf{X}(z)$.
(5) Find the magnitude coefficients of $\mathbf{X}(z)$ at the frequencies of interest.
(6) Create a vector of magnitude coefficients that is equal to $1/|\mathbf{X}(z)|$ from the coefficients computed in step (5).
(7) Take the vector of phase coefficients from step (3) and combine them with the vector of magnitude coefficients in step (6) to create a new single vector of complex coefficients having the original phase and the $1/|\mathbf{X}(z)|$ magnitudes.
(8) Take the inverse FFT of the new vector and use it as the new modified $\hat{\mathbf{h}}(t)$ in the FXLMS algorithm as normal.

If using multiple channel and/or ED control, the process is repeated for each $\hat{\mathbf{H}}(z)$ estimate. This procedure adjusts the magnitude coefficients of $\hat{\mathbf{H}}(z)$ while preserving the phase. As an offline process, it adds no computational burden to the algorithm when control is running. The results of the process can be seen in Fig. 7. The top plot in Fig. 7 shows the original and modified $\hat{\mathbf{H}}(z)$ magnitude coefficients. The bottom plot shows that the phase information of the same $\hat{\mathbf{H}}(z)$ has been preserved. Again, note that the two lines representing the original and modified phase information of $\hat{\mathbf{H}}(z)$ are directly on top of each other. The plots for the other $\hat{\mathbf{H}}(z)$ models for the other channels and ED components are similar.

The effect of the modification process on the eigenvalues can be seen in Fig. 8. The data for the figure were computed as before by an offline estimate of the autocorrelation matrix by using an actual $\hat{\mathbf{H}}(z)$ model from a mock cabin and finding the λ_{max} for each frequency from 0 to 400 Hz. The curve labeled "original" represents the same data shown in Fig. 7, and the curve labeled "modified" is an estimate of the eigenvalues using the modified $\hat{\mathbf{H}}(z)$ model. In Fig. 8, the eigenvalues in both the original and modified cases have been normalized by the largest of the original eigenvalues. It is seen that the modified eigenvalues are more uniform (equalized) over all frequencies, though not perfectly flat. While the variation in the modified eigenvalues would ideally be zero, the decreased variation compared to the original eigenvalues should produce an observable performance advantage.

Table II shows the improvement of the modified eigenvalues according to the same metrics defined for the flat $\hat{\mathbf{H}}(z)$ implementation over the range from 0 to 400 Hz. In Table II it can be seen that the modified case has a lower span a higher rms value, and a lower crest factor. In all three metrics, the values for the modified case are closer to the optimum values that would be present if the eigenvalues across all frequencies were exactly the same. It is also of note that setting the magnitude coefficients of $\hat{\mathbf{H}}(z)=1/|\mathbf{X}(z)|$ with an unequally weighted reference signal offered nearly the same percentage improvement as flattening the magnitude coefficients of $\hat{\mathbf{H}}(z)$ with an equally weighted reference signal (compare Tables I and II).

TABLE II. Comparison of original and modified eigenvalues using defined metrics—EE-FXLMS $\hat{\mathbf{H}}(z)=1/|\mathbf{X}(t)|$ implementation.

Metric	Original	Modified	% improvement
Span	4.77×10^5	99.24	100
rms	0.2	0.3	50
Crest factor	4.983	3.33	33

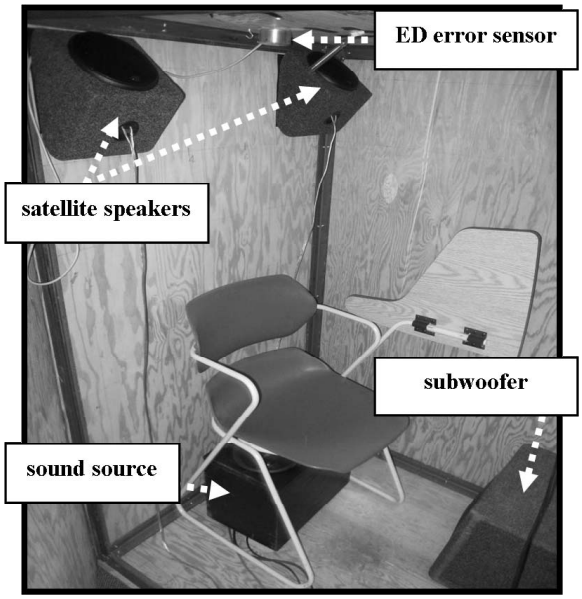

FIG. 9. Photo of inside of mock cab.

V. EXPERIMENTAL RESULTS

The performance advantages of the EE-FXLMS algorithm were verified for both single time-varying frequency and stationary multiple frequency test cases. First, the experimental setup will be explained, second the results for the EE-FXLMS flat $\hat{\mathbf{H}}(z)$ implementation will be shown, and, lastly, the results for the EE-FXLMS $\hat{\mathbf{H}}(z)=1/|\mathbf{X}(z)|$ implementation will be shown.

A. Experimental setup

The experiments were conducted inside a mock cabin enclosure with nominal dimensions of $1.0\times1.5\times1.1$ m^3. The cabin has a steel frame, 0.01-m-thick plywood sides, and a 0.003-m-thick Plexiglass® front panel. A speaker placed under a chair served as the sound source and three loudspeakers were set up in a two channel control configuration. The control signals were routed through a crossover circuit to route the low-frequency content (below 90 Hz) of both channels to a subwoofer on the floor of the cab and to route the high-frequency content (above 90 Hz) of each control channel to one of two smaller satellite speakers mounted in the top corners of the cab, near the back. An ED error sensor consisting of four equally spaced microphones around a small disk was placed on the ceiling near where an operator's head would be. The performance of the algorithms will be reported at the error sensor. A photo of the cab, error sensor, and speakers is seen in Fig. 9.

The control algorithms were implemented on a Texas Instruments TMS320C6713 DSP processor, capable of 1.350×10^6 floating point operations/s. Both adaptive control filters consisted of 32 taps for control of single tones and 100 taps for multiple tones, and all secondary path transfer functions were modeled with 128 taps. All input channels were simultaneously sampled at 2 kHz, and all input and output signals had 16 bits of resolution. Fourth-order Butterworth low pass filters (400 Hz cutoff) provided antialiasing and reconstruction of input and output signals, respectively.

TABLE III. Comparison of EE-FXLMS flat $\hat{\mathbf{H}}(z)$ implementation and normal FXLMS control for time-varying frequency experimentation. A positive number indicates that EE-FXLMS control performed better.

Sweep rate (Hz)	Control type	Average reduction at error mic (dB)	Difference[a] (dB)
2	Normal	6.5	3.5
	EE	10.0	
4	Normal	5.2	2.1
	EE	7.3	
8	Normal	4.3	1.4
	EE	5.7	
16	Normal	4.4	1.1
	EE	5.5	
32	Normal	4.0	0.3
	EE	4.3	
64	Normal	3.9	0.2
	EE	4.1	
128	Normal	3.9	0.0
	EE	3.9	
256	Normal	3.9	0.0
	EE	3.9	
	TOTAL AVERAGE		1.1

[a]Positive number indicates that EE-FXLMS performed better.

B. Results for EE-FXLMS Flat $\hat{\mathbf{H}}(z)$ implementation

The EE-FXLMS flat implementation was tested for a time-varying frequency. For the time-varying frequency, a swept sine signal with different sweep rates was used. The signal maintained the same amplitude at each frequency in the sweep.

1. Time-varying frequency results

Several swept sine test signals with different sweep rates were created. Each test signal consisted of a swept sine from 50 to 200 Hz and the rates ranged from 2 to 256 Hz/s. The time-averaged sound pressure level over the entire duration of the test signal was measured with and without control running. The convergence coefficient μ was experimentally determined by finding the largest stable value for the entire frequency range and then scaling it back by a factor of 10 to ensure stability. The μ for EE-FXLMS control was found to be 1×10^{-7} and the μ for standard FXLMS control was found to be 1×10^{-8}. Each measurement was repeated three times and the average and standard deviation were computed. The actual attenuation for both control types at the error sensor is shown in Table III. The difference in attenuation between EE-FXLMS and FXLMS controls is also shown in Table III. A positive number indicates the EE-FXLMS performed better. The standard deviation for each test case was small (usually less than 0.02 dB) and is not reported in the table.

The data show that when averaged over all of the data, EE-FXLMS performs 1.1 dB better than normal FXLMS at the error sensor. The data also show that the slower the sweep rate, the more advantage the EE-FXLMS control provides. For

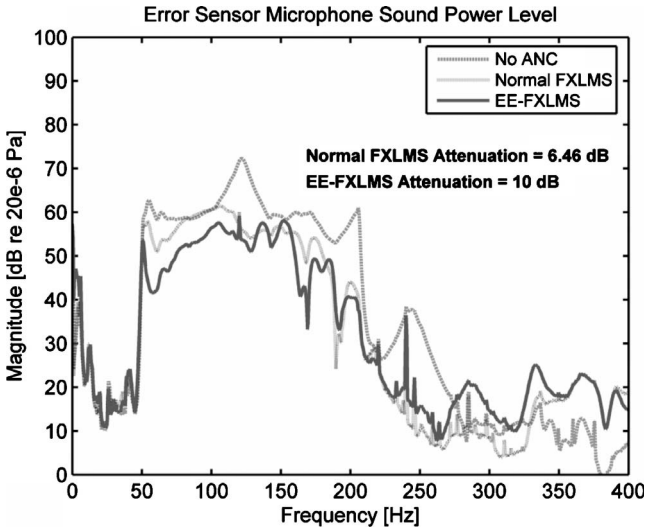

FIG. 10. SPL at the error sensor for normal FXLMS and EE-FXLMS control.

the 2 Hz sweep rate, EE-FXLMS control provides 3.5 dB more reduction at the error sensor. Figure 10 shows a plot of the control results for both normal FXLMS and EE-FXLMS for the 2 Hz sweep rate. For this run, the sound pressure level (SPL) at the error sensor before control was enabled was 87.9 dB (calculated over the entire frequency range). The SPL dropped to 81.5 dB for normal FXLMS control and 77.9 dB for EE-FXLMS control.

At the fastest sweep rates, the differences were almost negligible. An explanation of this can be found by looking at the fastest convergence times for the single frequency case. For this case, the fastest convergence times were seen to be on the order of 0.10 s. At the faster sweep rates, such as 128 Hz/s, the algorithm has 0.0078 s (1/128 Hz/s =0.0078 s/Hz) to converge at each frequency. At the slower sweep rates, such as 2 Hz/s, the algorithm has 0.5 s (1/2 Hz/s=0.5 s/Hz) to converge at each frequency. When the sweep rates are faster, the convergence times are several orders of magnitude larger than the time the algorithm has to converge on each frequency before it shifts, which leads to poor performance and little gain from the faster convergence times of the EE-FXLMS. When the sweep rates are slower, the convergence times are on the same order of time that the algorithm has to converge on each frequency before it shifts, which leads to better performance and noticeable gains from the faster convergence times of the EE-FXLMS.

C. Results for EE-FXLMS $\hat{\mathbf{H}}(z)=1/|\mathbf{X}(z)|$ implementation

The EE-FXLMS was compared again to the FXLMS algorithm for a multiple frequency test signal, this time with the magnitude of the tones in the reference signal decreasing with increasing frequency. Since the frequency resolution in $\hat{\mathbf{H}}(z)$ was not high, the magnitude coefficients in $\hat{\mathbf{H}}(z)$ bracketing the frequencies in the reference were adjusted to be the inverse magnitude of the tones in the reference. Five tone (50, 125, 200, 250, and 300 Hz) and 11 tone (50, 75, 100, 125, 150, 175, 200, 225, 250, 275, and 300 Hz) noise and reference signals were created for this comparison. Many of the tones in the signal were intentionally chosen to match frequencies where the magnitude response of $\hat{\mathbf{H}}(z)$ is large; frequencies where the advantages of the EE-FXLMS $\hat{\mathbf{H}}(z)=1/|\mathbf{X}(z)|$ implementation should be the most observable. Additionally, they were chosen far enough apart that the magnitude of $\hat{\mathbf{H}}(z)$ could be individually adjusted for each tone. Control was run with both the normal FXLMS algorithm and the EE-FXLMS algorithm with the $\hat{\mathbf{H}}(z)=1/|\mathbf{X}(z)|$ implementation. The number of control taps was increased to 100 for these test cases.

1. Multiple frequency results

The convergence coefficient μ was determined as before. The scaled μ for EE-FXLMS and normal FXLMS controls for the noise signal containing five tones were found to be 8×10^{-9} and 1×10^{-9}, respectively. The scaled μ for EE-FXLMS and normal FXLMS controls for the noise signal containing 11 tones were found to be 2×10^{-8} and 4×10^{-9}, respectively. The measured performance for each configuration was the amount of attenuation in decibels and the convergence time in seconds at each frequency, as well as for the total error signal. The convergence time was defined as the time it takes the signal to converge to $1/e$ (natural log e) of its initial value. A convergence time of 9 s means that the signal did not converge at that frequency in the time period of the measurement, which was 9 s. Each measurement was performed three times for computation of an average and standard deviation.

The average for the three test runs and the difference between normal FXLMS and EE-FXLMS controls are shown for the 5 tone test case in Table IV and the 11 tone case in Table V. In both tables, a linear average of the reduction at each frequency is merely given to give a sense of the performance of the algorithms at the frequencies of interest. The actual overall reduction of the entire error signal is also given. Again, a positive number for the difference indicates that EE-FXLMS performed better.

In the tables, it can be seen that EE-FXLMS control performed about 4 dB better overall at the error sensor for the 5 tone case and about 2 dB better overall for the 11 tone case. Observing the convergence time and attenuation at each frequency shows the more uniform performance over frequency of the EE-FXLMS approach. At some frequencies, the EE-FXLMS algorithm provides as much as 16 dB additional attenuation and converged several seconds faster. At higher frequencies, where the weighting of the tones in the reference was smaller, the FXLMS algorithm had very long convergence times and poor attenuation. In many cases, those frequencies did not appreciably converge during the measurement. The EE-FXLMS algorithm outperformed the normal FXLMS algorithm in both attenuation and convergence speed at all frequencies except 150 Hz, which did not converge well for any test case. 150 Hz corresponds to a large resonance mode of the mock cabin. Further investigation found that the error sensor was at a nodal position for this frequency, which leads to reduced performance. The frequency spectrum for the error signal with no control and FXLMS and

TABLE IV. Comparison of EE-FXLMS $\hat{\mathbf{H}}(z)=1/|\mathbf{X}(t)|$ implementation and normal FXLMS control for multiple stationary frequency experimentation of five tones. A positive number indicates that EE-FXLMS control performed better. The overall attenuation and convergence times reported at the bottom of the table are for the entire error signal and not the average of all the values at each tone.

	Control of 5 tones					
	Normal FXLMS		EE-FXLMS		Difference[a]	
Frequency (Hz)	Average reduction at error mic (dB)	Convergence time (s)	Average reduction at error mic (dB)	Convergence time (s)	Average reduction at error mic (dB)	Convergence time (s)
50	25.5	1.84	30.2	0.58	4.7	1.26
125	25.9	0.44	28.3	0.33	2.4	0.11
200	30.7	0.89	30.1	0.44	−0.6	0.45
250	5.2	9	19.8	0.99	14.6	8.01
300	0.2	9	16.6	3.04	16.4	5.96
Linear Average of reduction at 5 tones	18.2	3.61	25.2	0.95	7.0	2.66
Overall reduction for entire error signal	21.6	0.5	26.0	0.34	4.4	0.16

[a]Positive number indicates that EE-FXLMS performed better.

TABLE V. Comparison of EE-FXLMS $\hat{\mathbf{H}}(z)=1/|\mathbf{X}(t)|$ implementation and normal FXLMS control for multiple stationary frequency experimentation of 11 tones. A positive number indicates that EE-FXLMS control performed better. The overall attenuation and convergence times reported at the bottom of the table are for the entire error signal and not the average of all the values at each tone.

	Control of 11 Tones					
	Normal FXLMS		EE-FXLMS		Difference[a]	
Frequency (Hz)	Average reduction at error mic (dB)	Convergence time (s)	Average reduction at error mic (dB)	Convergence time (s)	Average reduction at error mic (dB)	Convergence time (s)
50	29	1.4	31	0.763	2	0.637
75	12.3	3.95	21.1	1.05	8.8	2.9
100	7.2	7.27	11	2.15	3.8	5.12
125	25.6	0.43	28.5	0.39	2.9	0.04
150	1.7	9	2.2	9	0.5	0
175	8.5	7.42	13.3	0.82	4.8	6.6
200	27.7	0.68	28.9	0.52	1.2	0.16
225	5.2	9	14.7	3.4	9.5	5.6
250	4.9	9	21.2	1.34	16.3	7.66
275	1.5	9	9.8	4.18	8.3	4.82
300	0.1	9	12.5	3.54	12.4	5.46
Linear Average of reduction at 11 Tones	11.4	5.56	17.5	2.30	6.1	3.26
Overall reduction for entire error signal	12.8	0.61	15.2	0.49	2.4	0.12

[a]Positive number indicates that EE-FXLMS performed better.

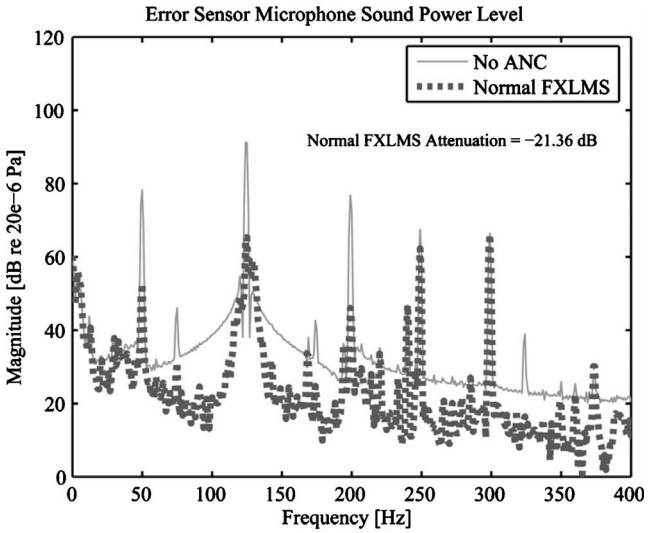

FIG. 11. SPL at the error sensor for a normal FXLMS.

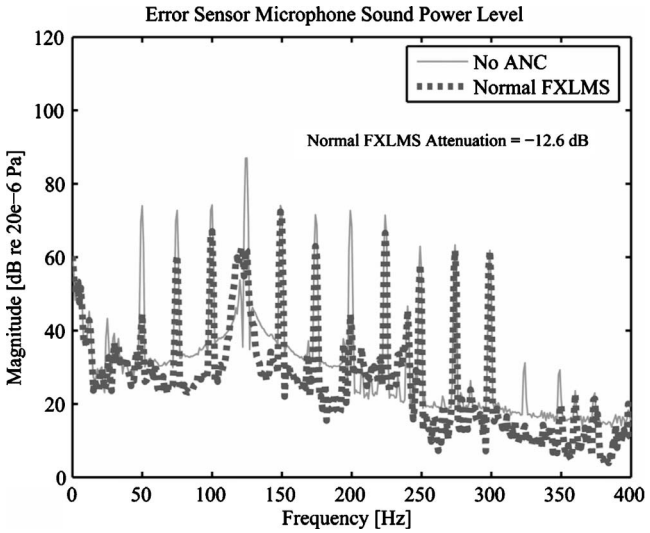

FIG. 13. SPL at the error sensor for a normal FXLMS.

EEFXLMS controls for the 5 tone case is shown in Figs. 11 and 12 and that for the 11 tone case is shown in Figs. 13 and 14.

VI. CONCLUSIONS

A new eigenvalue equalization approach (EE-FXLMS) has been demonstrated for time-varying and multiple frequency noise. It has been shown that adjustments to the magnitude coefficients of $\hat{\mathbf{H}}(z)$ while preserving the phase leads to a smaller eigenvalue spread, faster convergence times, and increased attenuation. Two offline methods for adjusting the magnitude coefficients of $\hat{\mathbf{H}}(z)$ to complement the magnitude of the reference signal have been demonstrated.

The EE-FXLMS implementation to flatten the magnitude coefficients of $\hat{\mathbf{H}}(z)$, when the magnitude of the reference signal is the same for all frequencies, led to as much as 3.5 dB additional attenuation at the error sensor for the slower sweep rates. An additional attenuation of 1.0 dB at the error sensor was seen at sweep rates of up to 16 Hz/s, with a slight increase still being seen at rates as high as 64 Hz/s. When averaged over all of the sweep rates tested, EE-FXLMS provided 1.1 dB additional attenuation at the error sensor.

The EE-FXLMS implementation of adjusting the magnitude coefficients of $\hat{\mathbf{H}}(z)$ to be the inverse magnitude of the reference signal, when the magnitude of the reference signal is different for all frequencies, led to as much as 4.4 dB additional overall attenuation at the error sensor and as much as 16 dB additional attenuation at an individual tone. The EE-FXLMS algorithm's convergence rate at individual frequencies was faster and more uniform than the normal FXLMS with several second improvement being seen in some cases.

The performance advantages of the EE-FXLMS become more meaningful when considering the simplicity of its implementation. It can be incorporated into any FXLMS algorithm with only a few lines of code. As an offline process, it does not increase the computational burden of the algorithm.

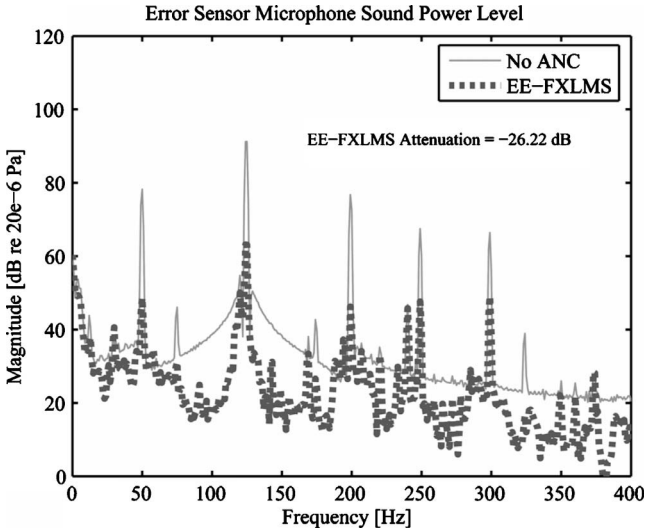

FIG. 12. EE-FXLMS control for reference signal with five tones.

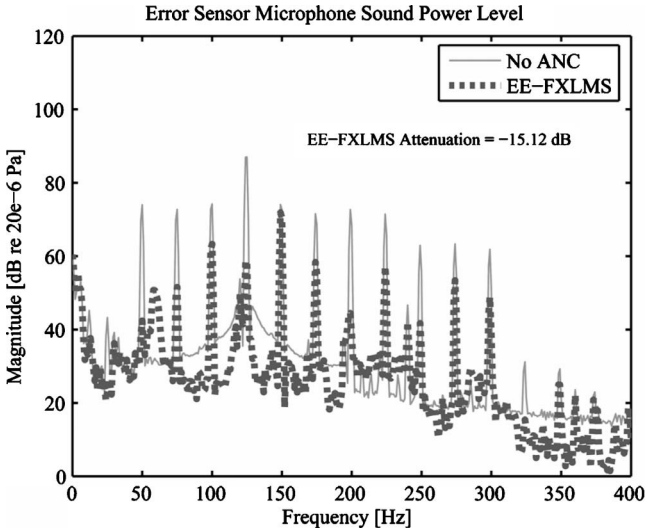

FIG. 14. EE-FXLMS control for reference signal with 11 tones.

Additionally, it does not require that the reference be internally generated or extensively modified.

These two methods of adjusting the magnitude coefficients of $\hat{\mathbf{H}}(z)$ provide a way to reduce the frequency dependent convergence of the FXLMS algorithm. As noted, the eigenvalue span resulting from these modifications is still not perfectly flat. Other alteration schemes may be developed that can further reduce this variation.

[1]D. R. Morgan, "An analysis of multiple correlation cancellation loops with a filter in the auxiliary path," IEEE Trans. Acoust., Speech, Signal Process. **28**, 454–467 (1980).

[2]J. C. Burgess, "Active adaptive sound control in a duct: A computer simulation," J. Acoust. Soc. Am. **70**, 715–726 (1981).

[3]Y. C. Park and S. D. Sommerfeldt, "Global attenuation of broadband noise fields using energy density control," J. Acoust. Soc. Am. **101**, 350–359 (1997).

[4]C. C. Boucher, S. J. Elliot, and K.-H. Baek, "Active control of helicopter rotor tones," *Proceedings of the INTER-NOISE 96*, pp. 1179–1182 (1996).

[5]B. Faber and S. D. Sommerfeldt, "Global control in a mock tractor cabin using energy density," in *Proceedings of the ACTIVE '04 (1994)*, edited by R. H. Cabell and George C. Maling, Jr..

[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, D. H. Youn, and I. W. Cha, "An active noise control algorithm for controlling multiple sinusoids," J. Acoust. Soc. Am. **104**, 248–254 (1998).

[8]M. Rupp and A. H. Sayed, "Modified FXLMS algorithms with improved convergence performance," *IEEE Proceedings of the ASILOMAR-29* (1995).

[9]S. M. Kuo, X. Kong, S. Chen, and W. Hao, "Analysis and design of narrowband active noise control systems," IEEE Trans. Acoust., Speech, Signal Process. **6**, 3557–3560 (1998).

[10]S. M. Kuo, M. Tahernezhadi, and W. Hao, "Convergence analysis of narrow-band active noise control systems," IEEE Trans. Circuits Syst., II: Analog Digital Signal Process. **46**, 220–223 (1999).

[11]L. Vicente and E. Masgrau, "Performance comparison of two fast algorithms for active control," *Proceedings of the ACTIVE 99*, pp. 1089–1100 (1999).

[12]S. D. Sommerfeldt and P. J. Nashif, "An adaptive filtered-x algorithm for energy-based active control," J. Acoust. Soc. Am. **96**, 300–306 (1994).

[13]J. W. Parkins, S. D. Sommerfeldt, and J. Tichy, "Narrowband and broadband active noise control in an enclosure using acoustic energy density," J. Acoust. Soc. Am. **108**, 192–203 (2000).

[14]D. C. Copley, B. Faber, and S. D. Sommerfeldt, "Energy density active noise control in an earthmoving machine cab," in *Proceedings of the NOISE-CON '05*, edited by J. Stuart Bolton, P. Davies, and G. C. Maling, Jr.. (1995)

[15]S. M. Kuo and D. R. Morgan, in *Active Noise Control Systems: Algorithms and DSP Implementations*, edited by J. G. Proakis (Wiley, New York, 1996), Chap. 5, pp. 147–186.

[16]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 Proc. F, Commun. Radar Signal Process. **138**, 313–319 (1991).

[17]S. J. Elliott and P. A. Nelson, "Active Noise Control," IEEE Signal Process. Mag. **10**, 12–35 (1993).

[18]L. A. Sievers and A. H. von Flotow, "Comparison and extensions of control methods for narrow-band disturbance rejection," IEEE Trans. Signal Process. **40**, 459–461 (1992).

[19]S. D. Snyder and C. H. Hansen, "The effect of transfer function estimation errors of the filtered-x LMS algorithm," IEEE Trans. Signal Process. **42**, 950–953 (1994).

[20]S. D. Snyder and C. H. Hansen, "The influence of transducer transfer functions and acoustic time delays on the implementations of the LMS algorithm in active noise control systems," J. Sound Vib. **141**, 409–424 (1990).