Journal of Sound and Vibration (1999) **227**(1), 159–181
Article No. jsvi.1999.2338, available online at http://www.idealibrary.com on IDEAL®

A PRINCIPAL COMPONENT ALGORITHM FOR FEEDFORWARD ACTIVE NOISE AND VIBRATION CONTROL

R. H. CABELL

Structural Acoustics Branch, NASA Langley Research Center, Mail Stop 463, *Hampton, VA* 23681, *U.S.A.*

AND

C. R. FULLER

Department of Mechanical Engineering, Vibration and Acoustics Labs, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, *U.S.A.*

(*Received* 11 *December* 1998, *and in final form* 22 *April* 1999)

A principal component least-mean-square (PC-LMS) adaptive algorithm is described that is applicable to large control systems used for feedforward control of single-frequency disturbances. The algorithm is a transform domain version of the multiple-error LMS algorithm. A transformation given by the principal components (PCs) of the transfer function matrix between the sensors and actuators in a control system at a single frequency is used to rotate the control-filter coefficient axes to a more convenient co-ordinate system, where: (1) independent convergence factors can be used on each co-ordinate to accelerate convergence; (2) insignificant control co-ordinates can be eliminated from the controller; and (3) co-ordinates that require excessive control effort can be eliminated. The PC-LMS algorithm has lower computational requirements than the multiple-error LMS algorithm. The two algorithms were compared in an experiment involving active structural acoustic control of a tone inside a closed cylindrical shell using 12 control actuators and 48 error sensors. Both algorithms reduced the primary response by more than 20 dB, averaged across the 48 microphones. The control system was slightly ill-conditioned at the test frequency, but the PC-LMS algorithm demonstrated stable convergence to the optimal control solution.

1. INTRODUCTION

Active control of tonal and broadband disturbances has received considerable attention in the past decade, partly due to the advances in the computational power of microprocessors needed to implement the common control algorithms. The literature on feedforward control of total disturbances, the subject of the current paper, is rich and varied and includes early work on adaptive filtering [1–4], as well as applications with relevance to acoustics [5–8]. Of particular interest for the current paper is the active control of tonal disturbances in complex systems such as

0022-460X/99/410159 + 23 $30.00/0

the interior of a passenger aircraft, where acceptable performance can require a large number of control actuators and error sensors. While the theoretical foundations of feedforward control are well-developed and are described in detail elsewhere [9–12], there are practical issues that can limit the performance of a feedforward control system. These include: ill-conditioning in the control system due to the placement of the actuators and sensors on a continuous structure; excessive computational requirements; and the spillover of control energy into unobservable modes that may have implications unrelated to acoustics, such as fatigue life [13, 14]. These secondary issues become especially important for large control systems, and control systems that rely on force inputs to a structure in order to reduce the radiated sound [15], also known as active structural acoustic control (ASAC).

Various methods have been proposed to address the specific limitations of large feedforward control systems. These include decreasing the convergence sensitivity to ill-conditioning by transforming the broadband input signal to the frequency domain with a discrete Fourier or cosine transformation [16]. This approach was first described in the context of a single-input/single-output adaptive filter [17–19] to increase the convergence speed when the input signal was not spectrally white. A recursive least-squares method has also been used in the context of a multiple-error LMS controller [20], although the computational complexity of the approach limits its application to small control systems. Reducing the computational complexity of the multiple-error LMS algorithm has also been studied [21]. Work has also been done to reduce the complexity of a large feedforward control system by grouping the actuators together, thereby creating a group of actuators driven by a single-control channel [22, 23]. This reduces the computational complexity since fewer control outputs must be calculated, and can improve the conditioning of the control system by removing redundancy. An additional benefit is an increase in the spatial extent over which a control force is applied to the structure, which can sometimes reduce the spillover of control energy into higher order modes.

The current work describes a method to deal with ill-conditioning, high computational complexity, as well as issues of control effort and controller performance, for feedforward control of single-frequency disturbances with high channel count control systems. The proposed algorithm, called principal components LMS, or PC-LMS, is applicable to situations where the multiple error LMS algorithm would normally be used. The PC-LMS algorithm is based on a signal-dependent transformation of the controller co-ordinates. The transformation requires an estimate of the transfer function between control actuators and error sensors at the frequency of interest. This is the same transfer function matrix that is used to filter the reference signal in the multiple-error LMS algorithm.

The most important aspect of the principal component transformation is that it decouples the co-ordinates of the control system. Once decoupled, each co-ordinate can be assigned its own step size and control effort penalty. Thus, the slow convergence associated with ill-conditioning in the transfer function matrix can be improved, and large control efforts can be limited as well. This single-frequency

transformation is similar to a transformation of the controller co-ordinates to modal space, as suggested elsewhere [24, 25]. However, the principal component transformation only requires knowledge of the transfer function matrix between the actuators and sensors at the frequencies of interest, and does not require any knowledge about the modal charactreristics of the system being controlled. Although PC-LMS requires a transformation of co-ordinates, the decoupling produced by the PC transformation lowers the net computational burden of the algorithm relative to the multiple-error LMS algorithm. The computational savings are most dramatic when there are more error sensors than control actuators.

The PC-LMS algorithm is closely related to an algorithm described in a recent US Patent [26, 27]. The patent describes the implementation of feedforward control using the principal components of the control system, and further notes that only the well-conditioned PCs should be controlled. An equally valid criteria not discussed in that patent is to control only those PCs that show substantial correlation with the primary response [28], which is independent of the conditioning of the PC [29, 30].

We begin by introducing relevant terminology in the context of the multiple-error LMS algorithm. The PC-LMS algorithm, and its roots in the statistical technique known as Principal Component Analysis, are then described within the notational framework used for the multiple-error LMS algorithm. The paper ends with a description of an active noise control experiment on a closed cylindrical shell, using 12 force actuators and 48 microphones to reduce tonal noise inside the shell.

2. MULTIPLE-ERROR LMS ALGORITHM

A schematic of a single-frequency feedforward control system based on the multiple-error LMS algorithm is shown in Figure 1. The quantities in the figure are in the frequency domain, and for clarity matrices and vectors are represented by boldface letters. The $(m \times 1)$ vector of error sensor responses, $\mathbf{e}(n)$, is given by a summation of the primary and control fields. The control field consists of the $(r \times 1)$ vector of control outputs, $\mathbf{y}(n)$, filtered by the $(m \times r)$ matrix of error path

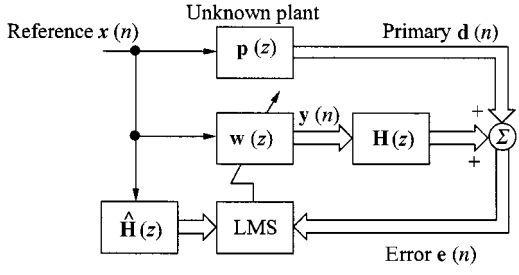

Figure 1. A multichannel LMS controller with transfer functions in the error path.

transfer functions $\mathbf{H}(z)$. The control outputs are the product of the control filter weights, $\mathbf{w}(z)$, and the reference signal. The matrix $\hat{\mathbf{H}}(z)$ denotes a model of the error path transfer functions [4].

The steady state sensor responses at a frequency ω are given by [9, 31]

$$\mathbf{e}(\omega) = \mathbf{H}(\omega)\mathbf{w}(\omega) + \mathbf{d}(\omega), \tag{1}$$

where the reference input is assumed to have a unit amplitude and zero phase, and the individual elements of each vector or matrix term are complex. The control filter weights can be computed to minimize a cost function such as

$$J = \mathbf{e}^H\mathbf{e} + \beta\mathbf{w}^H\mathbf{w}, \tag{2}$$

where $()^H$ denotes the complex conjugate transpose, β is a real scalar, and the frequency dependence is implicit. The weight vector that minimizes this cost function is [9]

$$\mathbf{w}_{\text{opt}} = -[\mathbf{H}^H\mathbf{H} + \beta\mathbf{I}]^{-1}\mathbf{H}^H\mathbf{d}. \tag{3}$$

A steepest descent recursion for $\mathbf{w}$ is written (for example, see references [9, 10])

$$\mathbf{w}(n+1) = (1 - \mu\beta)\mathbf{w}(n) - \mu\mathbf{H}^H\mathbf{e}(n), \tag{4}$$

where μ is the step-size parameter. The multiple-error LMS algorithm is a time-domain version of equation (4), using an instantaneous estimate of the gradient of the cost function [4].

The convergence of the weight vector to its optimal value is usually analyzed in terms of the principal components of the control system [9, 31]. Summarizing this analysis, a difference vector at iteration n is described as $\boldsymbol{\varepsilon}(n) = \mathbf{w}(n) - \mathbf{w}_{\text{opt}}$, and a recursion for $\boldsymbol{\varepsilon}(n)$ is written as

$$\boldsymbol{\varepsilon}(n+1) = (\mathbf{I} - \mu[\mathbf{H}^H\mathbf{H} + \beta\mathbf{I}])\boldsymbol{\varepsilon}(n). \tag{5}$$

This expression can be simplified using the similarity transform

$$[\mathbf{H}^H\mathbf{H} + \beta\mathbf{I}] = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^H, \tag{6}$$

where the columns of $\mathbf{V}$ contain the eigenvectors of the bracketed matrix and the diagonal of $\boldsymbol{\Lambda}$ contains the eigenvalues $(\lambda_1, \ldots, \lambda_r)$. Defining a transformed weight-error vector as $\boldsymbol{\xi}(n) = \mathbf{V}^H \in (n)$ yields the relation

$$\boldsymbol{\xi}(n+1) = (\mathbf{I} - \mu\boldsymbol{\Lambda})^{n+1}\boldsymbol{\xi}(0), \tag{7}$$

where $\boldsymbol{\xi}(0)$ is the transformed weight-error vector at time zero. Because $\boldsymbol{\Lambda}$ is diagonal, equation (7) demonstrates that the convergence of $\mathbf{w}$ can be described in terms of the convergence of a set of orthogonal co-ordinates, or principal co-ordinates, of the control system [2, 9, 32].

When the transfer function model, $\hat{\mathbf{H}}(z)$, contain errors relative to the true error path transfer functions, $\mathbf{H}(z)$, the weight vector does not converge to the optimum in equation (3), but instead converges to [33]

$$\mathbf{w}_{\infty} = -[\hat{\mathbf{H}}^H\mathbf{H} + \beta\mathbf{I}]^{-1}\hat{\mathbf{H}}^H\mathbf{d}. \tag{8}$$

In this case, the convergence behavior is given by [33]

$$\boldsymbol{\varepsilon}(n) = [\mathbf{I} - \mu(\hat{\mathbf{H}}^H\mathbf{H} + \beta\mathbf{I})]^n\hat{\boldsymbol{\varepsilon}}(0), \tag{9}$$

where $\hat{\boldsymbol{\varepsilon}}(n) = \mathbf{w}(n) - \mathbf{w}_\infty$. The difference converges to zero if the real parts of the eigenvalues of $[\mathbf{I} - \mu(\hat{\mathbf{H}}^H\mathbf{H} + \beta\mathbf{I})]$ are positive [3, 33].

For both the steepest descent and the multiple-error LMS algorithms, the convergence behavior of the weight vector depends on the eigenvalue spread of $[\hat{\mathbf{H}}^H\mathbf{H} + \beta\mathbf{I}]$ [2, 9, 32]. When this matrix is ill-conditioned, convergence can be slow and the optimal control solution may require high control forces. The effort penalty term in the cost function reduces this problem to some degree [32, 33] but will limit the reduction of the error norm. Other adaptive methods such as Newton's algorithm (for example, see references [9, 10]) or recursive least squares [20] can be used to reduce the convergence sensitivity to eigenvalue spread. However, these methods have very high computational requirements that make them impractical for large control systems.

3. PC-LMS ALGORITHM

The previous section illustrated how the principal components have been used to analyze the convergence behavior of an adaptive controller; PC-LMS takes this one step further by actually implementing the control algorithm in terms of the principal components. PC-LMS relies on a transformation of the controller co-ordinates to improve the convergence behavior and performance of the multiple-error LMS algorithm. This approach is similar to the co-ordinate transformations of the LMS algorithm that have been described elsewhere [17, 18, 34]. However, instead of a Fourier or cosine transformation, we use the principal components (PCs) of the transfer function matrix between the control actuators and error sensors at the frequency of interest. This decouples the control system co-ordinates, and thus error reduction and control effort can be determined independently for each co-ordinate. The statistical properties of the PCs also indicate the sensitivity of each control co-ordinate to the measurement noise in the error sensors. The decoupling transformation allows for the control of only a subset of the controller degrees of freedom, which gives greater flexibility for tailoring controller performance to meet the conflicting requirements of error reduction and control effort minimization.

The PCs are computed from the transfer function matrix at a single frequency, hence this formulation of the PC-LMS algorithm can only be used for single-frequency control problems. Control of multiple discrete frequencies could be achieved using several independent PC-LMS controllers, each targeting a different frequency. However, a causal broadband decoupling transformation is unlikely to exist for a general multichannel control system on a continuous structure, which will restrict the range of applications for this algorithm. Nonetheless, the benefits of the PC-LMS algorithm merit its consideration for controlling multiple discrete frequencies in complex systems, such as tones in an aircraft cabin.

The principal components may be familiar to engineers in the context of a spectral decomposition or similarity transformation, as in equation (6), but the statistical properties of the principal components are generally less well known. In statistics, a principal component analysis (PCA) is often used to examine data sets with many interrelated parameters [29, 30]. PCA is used to create new independent variables from linear combinations of the original variables that are more useful for describing the variation in the data set [30]. A graphical illustration of the PCs of a data set characterized by x and y values is shown in Figure 2. The principal co-ordinates of this data set are indicated in the figures by the dashed lines v_1 and v_2. These new co-ordinates are obtained by rotating the original x- and y-axis, such that the first PC, v_1, captures the greatest portion of the variation in the data set. The second PC, v_2, describes less variation. By definition the PCs are mutually orthogonal, and describe successively smaller amounts of variation in the original data set going from the first to the last PC. As a result, the first few PCs can often be used to efficiently represent the characteristics of a large data set.

In the context of a feedforward control problem, PCA is applied to the transfer function matrix between the actuators and error sensors at a particular frequency. It can be shown that each of the r eigenvectors of $(\mathbf{H}^H\mathbf{H})$ specifies a linear transformation from the actuator co-ordinates to the principal co-ordinates, and each of the m eigenvectors of $(\mathbf{H}\mathbf{H}^H)$ specifies a transformation from the sensor co-ordinates to principal co-ordinates [30, 35]. The singular-value decomposition provides a computationally convenient method to compute the eigenvectors, where the SVD of $\mathbf{H}$ is written (for example, see reference [36]) as

$$\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^H. \tag{10}$$

The $(m \times m)$ matrix $\mathbf{U}$ contains the eigenvectors of $\mathbf{H}\mathbf{H}^H$, and the $(r \times r)$ matrix $\mathbf{V}$ contains the eigenvectors of $\mathbf{H}^H\mathbf{H}$. The $(m \times r)$ matrix $\mathbf{S}$ contains the singular values, which are the square roots of the eigenvalues of $\mathbf{H}^H\mathbf{H}$. By definition, the singular values are decreasing, such that $s_1 > s_2 > \cdots > s_r$. Using the results of the

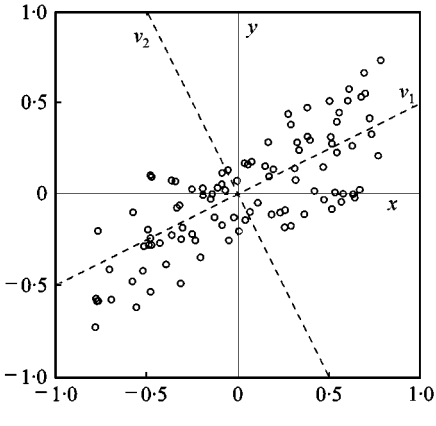

Figure 2. The principal components of a two-dimensional data set: —, original axes; - - -, principal axes.

SVD, a vector of actuator inputs can be mapped into principal co-ordinates as $\mathbf{z} = \mathbf{V}^H\mathbf{w}$, and conversely a vector of PC inputs can be mapped into actuator co-ordinates as $\mathbf{w} = \mathbf{V}\mathbf{z}$. Similarly, a vector of sensor responses can be mapped to PC co-ordinates as $\boldsymbol{\zeta} = \mathbf{U}^H\mathbf{e}$. The individual elements of U and $\mathbf{V}$ are complex, and therefore can be represented in the time domain by a two coefficient FIR filter at the frequency of interest.

Substituting the SVD of $\mathbf{H}$ into equation (1) yields

$$\mathbf{e} = \mathbf{U}\mathbf{S}\mathbf{V}^H\mathbf{w} + \mathbf{d}, \qquad \mathbf{U}^H\mathbf{e} = \mathbf{S}\mathbf{V}^H\mathbf{w} + \mathbf{U}^H\mathbf{d}, \qquad \boldsymbol{\zeta} = \mathbf{S}\mathbf{v} + \mathbf{p}, \qquad (11\text{–}13)$$

where $\boldsymbol{\zeta} = \mathbf{U}^H\mathbf{e}$ denotes the mapping of the sensor responses onto the PCs, $\mathbf{v} = \mathbf{V}^H\mathbf{w}$ gives the mapping of the actuator inputs onto the PCs, and $\mathbf{p} = \mathbf{U}^H\mathbf{d}$ is the mapping of the primary field onto the PCs. Expanding equation (13) term by term, and assuming more sensors than actuators ($m > r$), produces

$$\zeta_i = \begin{cases} s_i v_i + p_i & \text{for } i = 1, \ldots, r, \\ p_i & \text{for } i = r+1, \ldots, m. \end{cases} \qquad (14)$$

Each PC error terms ζ_i, depends on the corresponding PC control input, v_i, and the mapping of the primary response onto the ith PC. The last $(r + 1)$ through m PCs are observable but not controllable, and constitute the residual field after the control is applied.

A schematic diagram of the feedforward control problem reformulated in terms of the PCs of the control system is shown in Figure 3. Relative to the multiple-error LMS algorithm the error path model $\hat{\mathbf{H}}(z)$ is absent, having been replaced by the matrices $\mathbf{V}(z)$ and $\mathbf{U}^H(z)$. The purpose of $\hat{\mathbf{H}}(z)$ in the multiple-error LMS algorithm is to ensure correct phasing between the weight updates and the error signal. In PC-LMS, this is accomplished by the transformation to PC co-ordinates. Examination of equation (14) shows that the ith PC input and output are related by a real value, s_i. This means there is no phase difference between the input and output in PC co-ordinates and consequently no need to pre-filter the reference signal by the plant model. However, this does not suggest that the physical delays in the error path can be ignored. The transformation matrices $\mathbf{U}^H(z)$ and $\mathbf{V}(z)$ actually increase the error path delay, even though their net effect is to bring the PC inputs

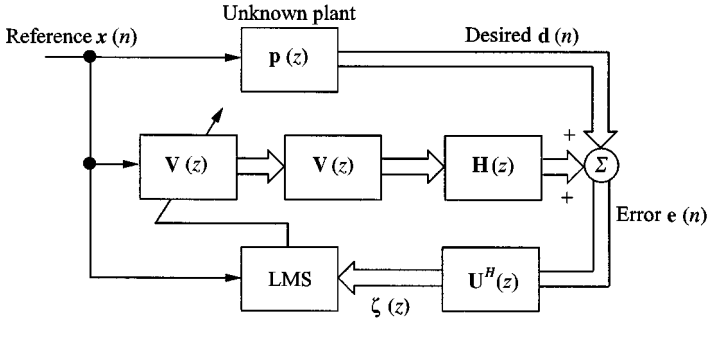

Figure 3. A principal component LMS controller with transfer functions in the error path.

TABLE 1

The number of multiply/accumulate operations (MACs) for multiple-error LMS and PC-LMS ($m = \#$ of sensors, $r = \#$ of actuators, $c = \#$ of PCs)

Operation	# of MACs Multiple-error LMS	LC-LMS
Compute PC output	N/A	$2c$
Compute actuator output	$2r$	$2cr$
Filter-x	$2rm$	N/A
Transform error inputs	N/A	$2cm$
Update control weights	$2rm$	$2c$
Total	$2r(2m + 1)$	$2c(m + r + 2)$

and outputs in phase with one another. This increased delay means the maximum allowable value of the step-size parameter is slightly lower for the PC-LMS algorithm than for the multiple-error LMS algorithm.

The elimination of the filtered-**x** computations and more importantly the simplification of the control problem from an $(m \times r)$ fully coupled system to a set of r independent co-ordinates reduces the computational requirements of the PC-LMS algorithm relative to the multiple-error LMS algorithm. Assuming a standard time domain implementation of a single reference, single-frequency multiple-error LMS controller such as that described in reference [4], the computational requirements of the PC-LMS and multiple-error LMS algorithms are compared in Table 1. The table lists the number of multiply-accumulate operations (MACs) required to update the control filter weights for a single iteration of the algorithm. The values in the table assume individual elements of the PC transformation matrices **U** and **V** are implemented using two coefficient FIR filters.

The computational savings of PC-LMS can be further increased if certain PCs are not controlled, thereby reducing the number of PC weights that have to be updated. The symbol c in Table 1 is used to denote the number of controlled PCs, where $c \leqslant r$. Assuming the full set of PCs is controlled ($c = r$), the PC-LMS algorithm requires $2r(m - r - 1)$ fewer computations per iteration than the multiple-error LMS algorithm. Therefore, when $m > (r + 1)$ the PC-LMS algorithm offers a computational savings relative to the multiple-error LMS.

The optimal input to the ith PC corresponding to the cost function

$$J = \mathbf{e}^H\mathbf{e} = \boldsymbol{\zeta}^H\mathbf{U}^H\mathbf{U}\boldsymbol{\zeta} = \boldsymbol{\zeta}^H\boldsymbol{\zeta} \tag{15}$$

is given by [32]

$$v_{i,\,opt} = -\frac{p_i}{s_i} \quad \text{for } i = 1, \ldots, r. \tag{16}$$

Therefore, in an ill-conditioned control system where the last few singular values are nearly zero, the corresponding control inputs can be very high for non-trivial

values of p_i. This result is the motivation for a control effort penalty where only those co-ordinates with s_i values greater than some threshold are controlled [26, 37]. When the ith PC is completely controlled, the reduction in the primary expressed as a percentage of the magnitude of the uncontrolled primary is

$$\% \text{ reduction} = 100 * \frac{p_i^H p_i}{\mathbf{d}^H \mathbf{d}}. \tag{17}$$

This reduction is independent of the singular value, s_i, and therefore is unrelated to the conditioning of the control system.

A recursive relation for updating the ith PC weight is obtained by substituting the SVD of $\mathbf{H}$ into equation (4) (and ignoring the effort penalty term), which yields

$$v_i(n+1) = v_i(n) - \mu s_i \zeta_i(n) \tag{18}$$

for the ith principal component. This equation represents the multiple-error LMS algorithm expressed in the principal co-ordinates of the control system. Combining the scalars μ and s_i yields the update relation

$$v_i(n+1) = v_i(n) - \alpha_i \zeta_i(n), \tag{19}$$

where α_i will be referred to as the PC step size. It is important to note the distinction between the PC step size α_i and the step-size parameter μ. Equation (18) indicates that when the weights are adapted according to the multiple-error LMS algorithm, each PC has a different PC step size given by $\alpha_i = \mu s_i$. If the control weights are updated in PC co-ordinates, each PC can be assigned a different PC step size, α_i, within stability bounds. This provides greater flexibility for tailoring the convergence behavior of the weight vector than is possible with multiple-error LMS algorithm. Any PC step size can be set to zero, in which case the PC is not controlled. This would be useful when the PC would require excessive control effort or would produce only a slight reduction in the error norm.

At the nth time step, the difference between the ith PC weight and its optimal value can be defined as $\varepsilon_i(n) = v_i(n) - v_{i,opt}$. Convergence of ε_i is then given by the difference equation

$$\varepsilon_i(n) = (1 - \alpha_i s_i)^n \varepsilon_i(0), \tag{20}$$

where $\varepsilon_i(0)$ is the initial value of the weight difference vector. Neglecting physical delays in the system, the PC step-size parameter must therefore be bounded as

$$0 < \alpha_i < \frac{2}{s_i}. \tag{21}$$

If the α_i's are defined to be proportional to the singular values, as in equation (18), the overall convergence rate of the weight vector will be equal to the convergence rate of the steepest descent algorithm. If the step sizes are defined to be inversely proportional to the singular values, the overall rate of convergence will be equal to that of Newton's algorithm [27, 35].

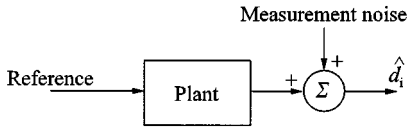

Figure 4. Sensor response with measurement noise.

3.1. SIGNIFICANCE OF COEFFICIENT VARIABLES

Feedforward control can be analyzed in the context of a linear regression problem [38, 39]. In this context, the variance–covariance matrix of the control filter weights [38, 40] has significance, and can be used to understand the effect of sensor noise. For this analysis we assume each sensor contains Gaussian measurement noise, as pictured in Figure 4. This noise is assumed to be independent from sensor to sensor with a common variance of σ^2.

The desired response vector now contains a stochastic component, and as a result the least-squares solution to the filtering problem will also vary stochastically. If $\hat{\mathbf{d}}$ denotes a single sampling of the response vector $\mathbf{d}$, the least-squares solution to the optimal filtering problem is given by

$$\mathbf{w}_{LS} = -(\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\hat{\mathbf{d}}, \tag{22}$$

where the Wiener solution is the expected values of $\mathbf{w}_{LS}$ [40]. From linear regression theory, the variance–covariance matrix of the magnitude of the elements of $\mathbf{w}_{LS}$ is [29, 38, 40]

$$V(|\mathbf{w}_{LS}|) = |(\mathbf{H}^H\mathbf{H})^{-1}|\sigma^2. \tag{23}$$

Only the variance of the magnitudes of the complex weights is considered here to simplify this discussion and analysis [38]. The variance–covariance matrix of the least-squares solution in PC co-ordinates is the diagonal matrix

$$V(|\mathbf{v}_{LS}|) = (\mathbf{S}'\mathbf{S})^{-1}\sigma^2 \tag{24}$$

or, for the ith PC,

$$V(|\mathbf{v}_{LS,i}|) = \frac{\sigma^2}{s_i^2}. \tag{25}$$

This expression indicates that the variability of the magnitude of a PC filter weight due to the stochastic nature of the desired response vector can exceed the variance of the random noise when s_i is less than one. This means measurement noise can produce large fluctuations in the values of certain v_i. Whether or not these fluctuations affect the performance of the adaptive solution depends on σ^2 and s_i, but in cases of excessive variability the PC step size may have to be lowered to bring the variations within acceptable limits.

It is worth noting that the variance of v_i given in equation (25) is independent of the reduction in the error norm obtained by controlling the ith PC. The variance is inversely proportional to s_i^2, whereas the error reduction depends only on the

mapping of the primary field onto the ith PC, p_i. This means PCs with low-variance control inputs may or may not be useful for reducing the error norm.

3.2. ROBUSTNESS TO MODELLING ERRORS

We next consider the robustness of the PC-LMS algorithm to errors in the model of the error path transfer functions, $\mathbf{H}$, based on the method used to study the convergence behavior of the steepest descent algorithm [9, 13]. The convergence of the weight-difference vector when the model, $\hat{\mathbf{H}}$, of the error path transfer function differs from the true path transfer functions, $\mathbf{H}$, is given by the recursion [9]

$$\mathbf{w}(n) - \mathbf{w}_\infty = (\mathbf{I} - \mu\hat{\mathbf{H}}^H\mathbf{H})(\mathbf{w}(n-1) - \mathbf{w}_\infty). \tag{26}$$

If $\hat{\mathbf{H}} = \mathbf{USV}^H$ denotes the SVD of $\hat{\mathbf{H}}$, the true transfer function matrix $\mathbf{H}$ can be written as $\mathbf{H} = \mathbf{UKV}^H$, where $\mathbf{K}$ is not necessarily a diagonal matrix. Substituting these expressions into equation (26) yields

$$\mathbf{v}(n) - \mathbf{v}_\infty = (\mathbf{I} - \mu\mathbf{S}'\mathbf{K})(\mathbf{v}(n-1) - v_\infty) \tag{27}$$

which describes the convergence behavior of the multiple-error LMS algorithm in terms of the principal co-ordinates. The eigenvalues of $(\mathbf{S}'\mathbf{K})$ are equal to the eigenvalues of $(\hat{\mathbf{H}}^H\mathbf{H})$, so the criteria for stable convergence discussed with respect to equation (9) have not changed. Equation (27) indicates that if the PC coefficients are adapted using the step sizes given in equation (18), the PC-LMS algorithm has the same robustness to modelling errors as the multiple-error LMS algorithm.

We next extend this result to the case where the PC step sizes are different from those dictated by equation (18), and where certain PCs are not adapted at all. Following the approach described in reference [41], the matrix $\mathbf{K}$ used in equation (27) is rewritten as a perturbation of the singular-value matrix, such that

$$\mathbf{K} = \mathbf{S} + \Delta\mathbf{S}, \tag{28}$$

where $\Delta\mathbf{S}$ is a general matrix. With this substitution, equation (27) becomes

$$\mathbf{v}(n) - \mathbf{v}_\infty = (\mathbf{I} - \mu(\mathbf{S}'\mathbf{S} + \mathbf{S}'\Delta\mathbf{S}))(\mathbf{v}(n-1) - \mathbf{v}_\infty). \tag{29}$$

An approximate condition for stable convergence in the presence of modelling errors that was described in reference [41] and experimentally verified for a limited number of cases in references [35], assumes that only the diagonal terms of equation (29) are relevant. With this assumption, convergence is stable if

$$\mathrm{Re}(s_i^2 + s_i\Delta s_{ii}) > 0, \qquad s_i > -\mathrm{Re}(\Delta s_{ii}), \tag{30, 31}$$

where Δs_{ii} is the ith diagonal element of $\Delta\mathbf{S}$. The validity of this assumption will naturally depend on the structure of $\Delta\mathbf{S}$, although it is difficult to offer general guidelines as to which kinds of uncertainty are tolerable. However, as long as the assumption leading to equation (31) is valid, the stability of convergence of one PC is independent of any other PC. It is therefore reasonable to use PC step sizes that are different from those specified in equation (18) and still expect stable convergence. If equation (31) is not satisfied for the ith PC, it should be possible to

set the corresponding PC step size to zero and allow the remaining PCs to converge to their respective optimal values.

Although the assumption leading to equation (31) is approximate, it does appear to be valid for small perturbation matrices, $\Delta\mathbf{S}$, such as those due to normal experimental error. Several successful experiments, including those discussed later in this paper, have been conducted in which the PC step sizes were different from those specified in equation (18), and in which only a subset of the PCs were controlled. In all cases the PC-LMS algorithm demonstrated the same robustness to modelling error as the multiple-error LMS algorithm.

Because of this ability to set the PC step sizes independently, the PC-LMS algorithm provides greater flexibility for stabilizing the convergence of a multichannel controller than the multiple-error LMS algorithm. In multiple-error LMS, a control effort penalty can be used to stabilize convergence by setting the penalty parameter β larger than the negative real parts of the eigenvalues of $(\hat{\mathbf{H}}^H\mathbf{H})$ [33, 34]. However, if the real part of one of the first few principal components were negative, the effort penalty would have to be set to a relatively large value to offset the negative eigenvalue. The large effort penalty would limit both the control effort and the reduction of the error norm. In contrast, if control is implemented using PC-LMS, the PC step-size parameter for the unstable principal component could be set to zero and the remaining principal components allowed to converge to their respective optimal values.

3.3. RESIDUAL ERROR

Errors in the estimated transfer function matrix affect the minimum value of the cost function. The residual error response in untransformed controller co-ordinates is written [9] as

$$\mathbf{e}_\infty = [\mathbf{I} - \mathbf{H}[\hat{\mathbf{H}}^H\mathbf{H}]^{-1}\hat{\mathbf{H}}^H]\mathbf{d}. \tag{32}$$

Substituting the SVD of $\hat{\mathbf{H}}$ and $\mathbf{H}$ into equation (32), the residual error in terms of PCs is given by

$$\boldsymbol{\zeta}_\infty = [\mathbf{I} - \mathbf{S}\mathbf{S}^+][\mathbf{I} + \Delta\mathbf{S}\mathbf{S}^+]^{-1}\mathbf{p}, \tag{33}$$

where $\mathbf{S}^+$ is the pseudo-inverse of $\mathbf{S}$. For small perturbation ΔS this can be further simplified as [41]

$$\boldsymbol{\zeta}_\infty = [\mathbf{I} - \mathbf{S}\mathbf{S}^+][\mathbf{I} - \Delta\mathbf{S}\mathbf{S}^+]\mathbf{p}. \tag{34}$$

The residual error response in PC co-ordinates is thus

$$\zeta_{i,\infty} = \begin{cases} 0, & i = [\text{indices of controlled PCs}], \\ p_i + \sum_{j=1}^{r} \dfrac{-\Delta s_{ij}}{s_j} p_j, & i = [\text{indices of uncontrolled PCs}], \end{cases} \tag{35}$$

where the list of uncontrolled PCs includes those that are not controllable, $i = r + 1, \ldots, m$. Therefore, when the transfer function matrix contains errors and the ith PC is is left uncontrolled, the residual error term will consist of the primary response component, p_i, plus contributions from all the controllable PCs as indicated by equation (33). When the model errors are small such that the Δs_{ij} are small, the additional residual errors will also be small. In this case the benefits of eliminating a PC, such as limiting control effort or stabilizing convergence, may be worth the slight increase in residual error.

4. EXPERIMENTAL RESULTS

The performance of the PC-LMS algorithm was compared with the multiple-error LMS algorithm in a single-frequency noise control experiment. The algorithms were used to actively control a tone inside a closed cylindrical shell created by loudspeakers outside the shell. Control inputs were provided by inertial force shakers mounted on the ring frames of the shell, and microphones inside the shell were used as error sensors. The overall noise reduction, control input magnitudes, and incorporation of control effort penalties were studied using both the control architectures at a single test frequency. A more thorough discussion of the experiment can be found in reference [35].

4.1. APPARATUS

The tests were conducted on a closed cylindrical shell in NASA Langley's Acoustic and Dynamics Laboratory. This facility has been used to evaluate control strategies for aircraft interior noise reduction [42, 43]. The shell is pictured in Figure 5, and has an overall length of 3·66 m, a diameter of 1·68 m, and a wall thickness of 1·7 mm. The structure is constructed from wound carbon fiber filaments in an epoxy resin, and contains 10 J-section ring frames and 22 hat-section stringers. A plywood floor is located 0·54 m above the bottom of the shell and the shell-floor joint is sealed to acoustically isolate the spaces above and below the floor. The ends of the cylinder are closed with rigid baffles made from 3·18 m thick particle board with an access hatch cut into one baffle. The entire structure is located in a semi-anechoic environment to reduce reverberation and minimize the background noise.

Two 100 W electrodynamic loudspeakers were positioned on either side of the cylinder exterior and used to create an external pressure disturbance on the shell. This disturbance created an internal acoustic response which constituted the primary field to be actively controlled. The loudspeakers were positioned at the midpoint of the cylinder in both the vertical and horizontal directions, and were located 0·5 m away from the exterior surface of the cylinder.

Forty-eight electret condenser microphones were uniformly arranged in the interior of the cylinder to provide error feedback to the control system. The microphones were mounted on six metal hoops that were attached to the plywood floor, with eight microphones per hoop. A schematic of the microphone locations is

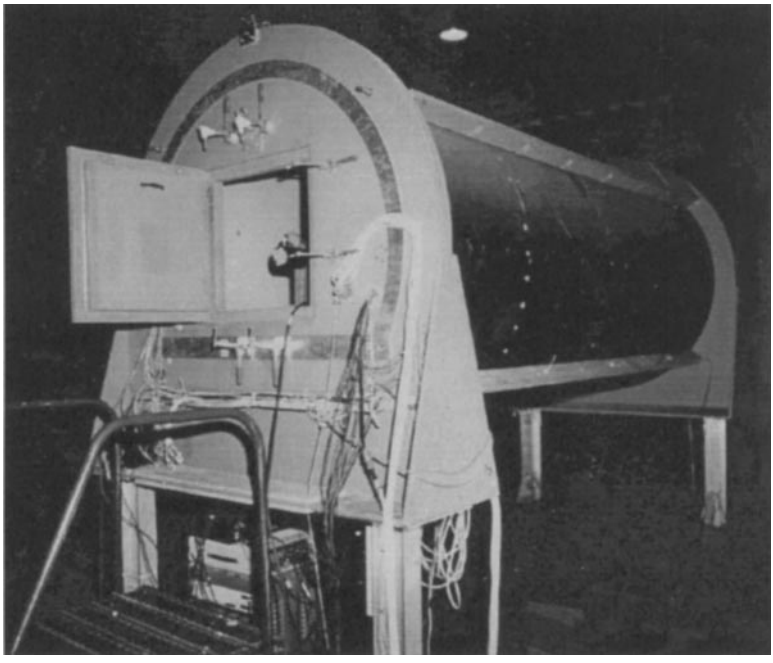

Figure 5. Exterior view of cylindrical shell.

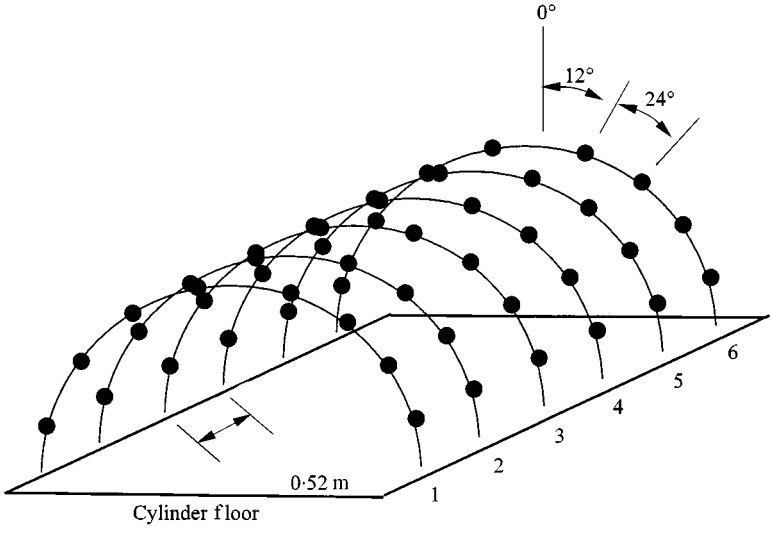

Figure 6. Microphone mounting locations: ●, microphone.

shown in Figure 6. All 48 microphones were used as error sensors in the control system, and there were no secondary microphones used to measure the noise reduction at points away from the error sensors.

Control inputs were provided by 12 inertial force actuators clamped onto the ring frames of the cylinder. The actuators produced radial force inputs that induced

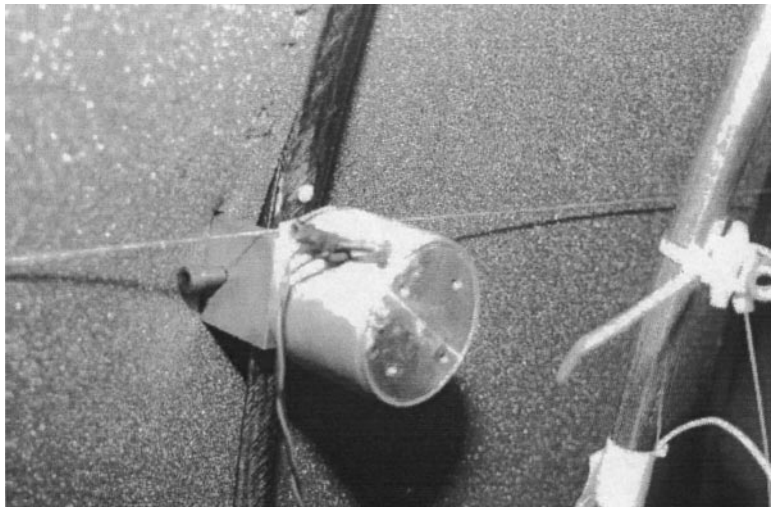

Figure 7. Actuator mounted on ring frame.

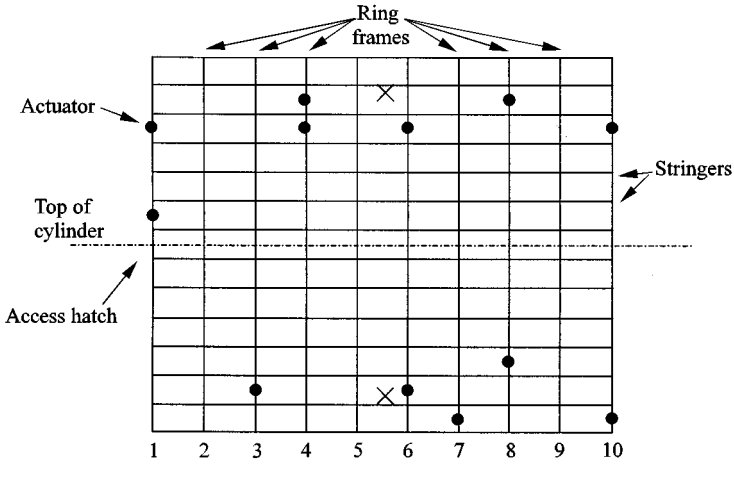

Figure 8. Actuator mounting locations: ●, control actuator; ×, location of external loudspeaker.

an interior acoustic response, similar to other implementations of active structural acoustic control [7]. The shakers had a nominal force constant of 22 N per Ampere and a natural frequency of ≈ 60 Hz. Each one was clamped onto a ring frame midway between two stringers, as shown in Figure 7. The foam damping material between the ring frames in the photo was not present during the experiments discussed here. The actuator locations were selected from a group of candidate locations using a combinatorial search routine [43] to maximize noise reduction at the test frequency. A schematic diagram of the actuator locations is shown in Figure 8, which is an unwrapped view of the portion of the cylinder above the plywood floor. Stringers are denoted by horizontal lines and ring frames by vertical

lines. The two X's in the figure designate the approximate locations of the external loudspeakers. The selected actuator locations are indicated by black circles. Most of the actuators were placed at approximately the same circumferential location as the external speakers.

The control algorithms were executed on a floating point digital signal processor residing in a desktop computer. The control system was originally designed for controlling a tone and its first five harmonics, hence the controller operated at a high sample rate relative to the fundamental frequency being controlled. Early testing indicated that the single DSP could not compute the weight updates fast enough when running the multiple-error LMS algorithm, although it was fast enough for PC-LMS due to its lower computational requirements. To avoid the complexity of adding a second DSP, the multiple-error LMS algorithm was modified slightly such that the weight updates were computed one actuator at a time in a round-robin fashion. This decreased the computational burden at each sample instant but increased the overall convergence time of the multiple-error LMS algorithm.

The step-size parameter μ for the multiple-error LMS algorithm was set to one-fourth of its maximum theoretical value, or $1/2\lambda_{max}$, where λ_{max} is the maximum eigenvalue of $(\mathbf{H}^H\mathbf{H})$. This value turned out to be too high for the PC-LMS algorithm, probably because of the increased error path delay due to the PC transformation matrices. The convergence of PC-LMS was stable when all of the PC step sizes set to $1/8\lambda_{max}$. From the discussion of equation (18), by setting all of the PC step sizes to $1/8\lambda_{max}$ the overall convergence rate of the weights was faster than if the multiple-error LMS algorithm had been used with a step size of $1/8\lambda_{max}$.

The signal conditioning applied to the microphone responses and actuator inputs is shown in Figure 9. Anti-aliasing and reconstruction filters were set at 723 Hz, and an additional low-pass filter with a cutoff frequency of 1000 Hz was applied to the microphone inputs. The digital to analog (D/A) converters had an output range of ± 10 V, and the control results given later are expressed in terms of the D/A output voltage. The reference signal, which is not shown in Figure 9, was generated on the DSP and sent to the loudspeakers. This ensured very high coherence between the primary disturbance and the reference signal.

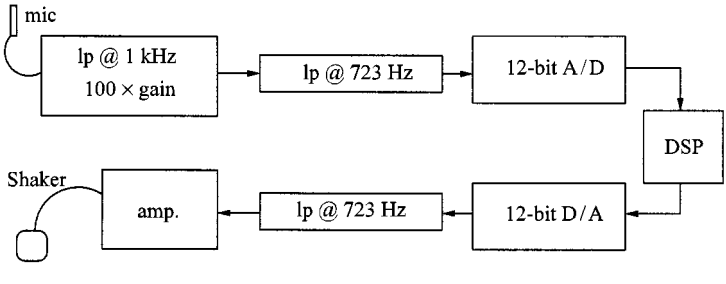

Figure 9. Signal conditioning.

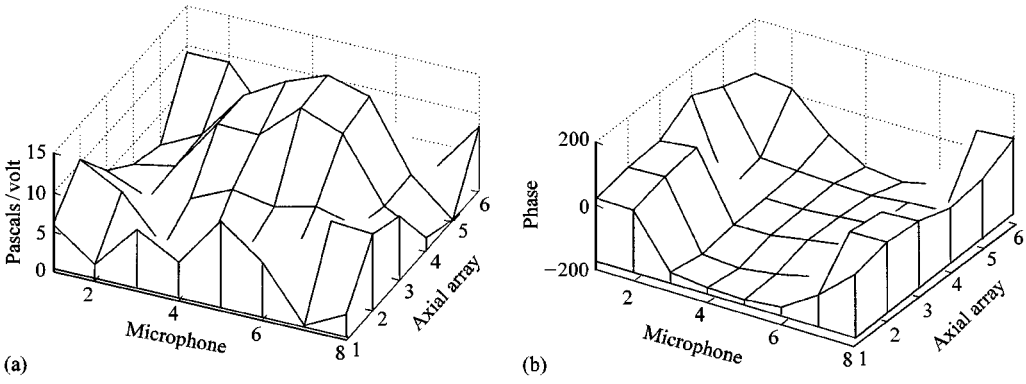

Figure 10. Primary response at 147 Hz: (a) amplitude; (b) phase.

4.2. MEASURED DATA

The noise-control experiments were conducted at 147 Hz, which was close to the natural frequency of the (2, 1, 0) cavity mode at 150 Hz [44]. The sampling rate for the control system was set to 2500 Hz, which was much higher than the ideal sampling rate of 4 times 147 Hz, or $\approx$590 Hz. As mentioned previously, the high sample rate was used so the control system could control the higher harmonics of the 147 Hz tone, although those frequencies were not studied during this test.

The microphone responses due to excitation by the external loudspeakers at 147 Hz are shown in Figure 10. The data were measured as complex valued transfer functions from the internal reference signal on the DSP, which was used to drive the loudspeakers, to each microphone response. The amplitude and phase of the transfer functions are plotted on an unwrapped view of the microphone grid, where the axial array number corresponds to the longitudinal cylinder dimension and the microphone number corresponds to the circumferential dimension. The amplitude and phase of the response correspond roughly to the second order variation along the circumference of the cylinder and first order variation along the length of the shell.

The transfer functions between the actuators and microphones at the test frequency were then measured and used to compute the optimal control solution. The singular values of the transfer function matrix and the properties of the optimal control solution are listed in Table 2, in terms of the principal components of the control system. Note that these values were computed offline from the transfer function and primary response measurements. The ratio of the largest to smallest singular value is 60, which means the control system was slightly ill-conditioned at the test frequency. The noise control corresponding to the optimal solution is given for each PC as a percent reduction of the primary response. The values indicate that only the first three PCs produce appreciable noise reduction, and of these the third produces the largest reduction of 58·6%. The optimal control inputs in Volts output of the D/A converters illustrate how the last few PCs can require extraordinarily high efforts but produce very little reduction in the primary, as noted elsewhere [32].

TABLE 2

Performance of optimal controller at 147 Hz

PC #	Singular value	Reduction of primary (%)	Control effort (V^2)
1	5·35	21·0	6·3
2	4·64	16·1	60·5
3	3·62	58·6	38·8
4	1·88	0·4	1·1
5	1·47	1·0	13·0
6	0·80	1·0	13·0
7	0·65	0·0	0·5
8	0·42	0·6	30·1
9	0·32	0·3	26·5
10	0·24	0·0	0·7
11	0·18	0·5	126·0
12	0·09	0·0	14·2

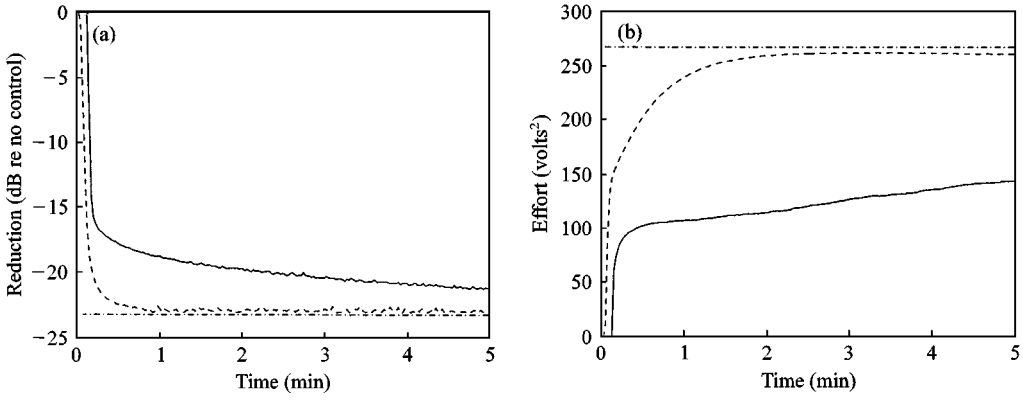

Figure 11. Control results at 147 Hz: (a) reduction of microphone responses; (b) control effort; —, multiple error LMS; - - -, PC-LMS, - · - · -, optimal controller.

Experimentally measured noise reduction for the controllers are shown in Figure 11(a) as time histories of reduction in the primary response averaged across all 48 microphones. The values in the plot were obtained by applying a low-pass filter to instantaneous measurements during convergence. The optimal solution of 23·2 dB is indicated in the figure by the dashed line. This dramatic reduction across the error microphones is possible, in part, because of the high coherence between the reference and the primary response. Both controllers showed stable convergence towards the optimal solution. The PC-LMS controller converged very quickly, while the multiple-error LMS controller had not fully converged after 5 min. The different step sizes and the slower weight adaptation for the multiple-error LMS algorithm make it difficult to draw any conclusions from the different convergence speeds.

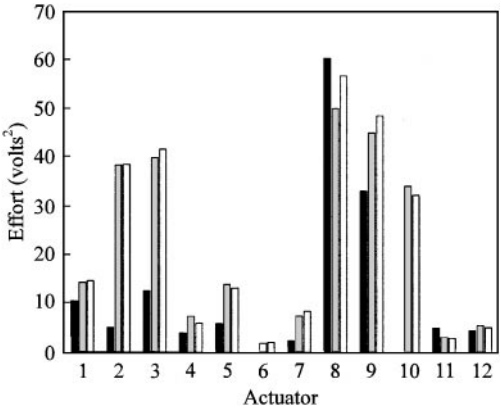

Figure 12. Control efforts for individual actuators after five minutes of convergence. ■, multiple error LMS; ■, PC-LMS; □, optimal controller.

The corresponding total control effort for the two controllers is plotted in Volts^2 (of output of the D/A converters) versus time in Figure 11(b). The PC-LMS controller converged to the expected total effort of 268 V^2 after 2 min, whereas the multiple-error LMS controller was well below the predicted value but still increasing after 5 min. The control efforts for the two controllers after 5 min of convergence are compared with the optimal solution, in terms of the inputs to individual actuators, in Figure 12. The PC-LMS algorithm, which appeared to have fully converged after 5 min, showed good agreement with the optimal input voltage for all 12 actuators, although the inputs were slightly low for actuators eight and nine. In contrast, the multiple-error LMS algorithm, which had not fully converged after 5 min, showed large discrepancies with the optimal input on several actuators.

These results demonstrate stable, predictable performance from the PC-LMS algorithm even when the PC step sizes were different from those given by equation (18).

An additional test was conducted to compare the implementation of control effort penalties with the two control algorithms. For the multiple-error LMS algorithm the effort penalty was of the form given in equation 2, with $\beta = 0{\cdot}1$. For PC-LMS, an effort penalty was implemented by controlling only the first five out of the 12 PCs. Noise reduction results and the associated control effort for the two controllers are shown in Figure 13. The multiple-error LMS algorithm reduced the primary by 15·4 dB, whereas the PC-LMS algorithm reduced it by 15·1 dB. The associated control efforts reveal that the PC-LMS algorithm was more efficient, requiring only 55 V^2 of effort compared to 72 V^2 for multiple-error LMS.

Previous work [32] has illustrated how a uniform effort penalty, such as that used here with the multiple-error LMS algorithm, constrains all PCs of the control system. The degree of constraint depends on the ratio of the penalty parameter β to the eigenvalue of each PC. Likewise, the uniform effort penalty allows the weights to converge along every principal co-ordinate of control system, although only

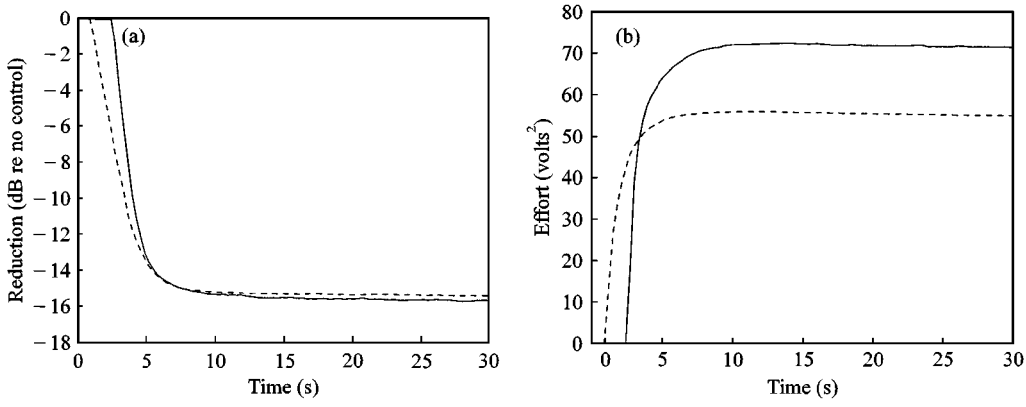

Figure 13. Comparison of control effort penalties: (a) reduction of microphone responses; (b) control effort; —, multiple error LMS; - - - , PC-LMS

very slightly in the direction of ill-conditioned co-ordinates. These two considerations taken together provide an explanation for the relative inefficiency of the multiple-error LMS algorithm in Figure 13. Well-conditioned principal co-ordinates that were capable of producing significant noise reduction were not completely controlled due to the slight constraint of the effort penalty. Similarly, a small amount of control effort was spent on the ill-conditioned co-ordinates that produced very little noise reduction.

In contrast, the effort penalty implemented in the PC-LMS algorithm controlled only the first few well-conditioned PCs and did not spend any control energy on the higher PCs which were slightly ill-conditioned. Because the first few PCs produced substantial noise reduction at the test frequency, the controller produced good noise reduction with an efficient use of control effort. The degree to which these conclusions hold in general will depend on the conditioning of the control system, and the noise-reduction potential of the first few PCs of the control system.

5. CONCLUSIONS

The principal component LMS algorithm was described and experimentally shown to be a useful alternative to the multiple-error LMS algorithm for the active control of tones using large control systems. The PC-LMS algorithm is a transform domain version of multiple-error LMS, where the transform is based on the principal components of the control system at a specific frequency, thereby decoupling the control system at that frequency. This decoupling reduces the computational burden of adaptively updating the control filter weights: if r denotes the number of control actuators and m the number of error sensors, PC-LMS requries $2r(m-r-1)$ fewer computations per sample iteration than the multiple-error LMS algorithm.

Problems associated with ill-conditioning in the control system, such as slow convergence and excessive control effort, can be addressed easily when control is implemented using PC-LMS. The decoupled nature of the PC-LMS means each

control co-ordinate can be converged independently of every other co-ordinate. Thus the convergence of slow co-ordinates can be accelerated by increasing the corresponding PC step sizes in the adaptive weight update equation. Likewise, the control effort spent on the co-ordinate can be constrained independently of every other co-ordinate. Ill-conditioned principal components that produce little reduction in the primary disturbance can thus be fully constrained without affecting the well-conditioned co-ordinates that reduce the primary response.

An experimental study of the PC-LMS algorithm demonstrated the stable, predictable convergence behavior of the algorithm in the presence of slight ill-conditioning in the control system. The PC-LMS and multiple-error LMS algorithms were studied in an active structural acoustic control experiment on an aircraft fuselage model. Twelve inertial force actuators were mounted on the ring frames of a closed cylindrical shell, and 48 microphones were used as error sensors inside the shell. Both controllers reduced the mean-square microphone response due to the excitation by external loudspeakers by over 20 dB. The PC-LMS controller converged to the optimal control solution, in terms of both the error reduction and the inputs to the control actuators.

The ease of limiting the control effort with the PC-LMS algorithm relative to the multiple-error LMS algorithm also was demonstrated experimentally. The control effort was constrained in the multiple-error LMS algorithm using a uniform effort penalty in the controller cost function. For PC-LMS, the effort was constrained by controlling only the first five PCs of the controller. Noise reductions of 15·4 and 15·1 dB were obtained using multiple-error and PC-LMS, respectively. However, the PC-LMS algorithm appeared to be more efficient in obtaining that noise reduction, requiring only 55 V^2 of the effort compared to 72 V^2 for multiple-error LMS.

ACKNOWLEDGMENTS

This work was supported by a grant from the Structural Acoustics Branch at NASA Langley Research Center.

REFERENCES

1. B. Widrow and M. E. Hoff Jr 1960 *Proceedings IRE WESCON Convention Record*, 96–104. Adaptive switching circuits.
2. B. Widrow, J. R. Glover Jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hern, J. R. Zeidler, E. Dong Jr. and R. C. Goodlin 1975 *Proceedings of the IEEE* **63**, 1692–1716. Adaptive noise cancelling: Principles and applications.
3. D. R. Morgan 1980 *IEEE Transactions on Acoustics, Speech, and Signal Processing* **ASSP-28**, 454–467. An analysis of multiple correlation cancellation loops with a filter in the auxiliary path.
4. S. J. Elliott, J. M. Stothers and P. A. Nelson 1987 *IEEE Transactions on Acoustics, Speech, and Signal Processing* **ASSP-35**, 1423–1434. A multiple error LMS algorithm and its application to the active control of sound and vibration.
5. J. C. Burgess 1981 *Journal of the Acoustical Society of America* **70**, 715–726. Active adaptive sound control in a duct: a computer simulation.

6. A. J. Bullmore, P. A. Nelson and S. J. Elliott 1986 *AIAA* 10*th Aeroacoustics Conference*, *Seattle*, *WA*, *July* 9–11, 1986, *AIAA* 86-1958. Active minimisation of acoustic potential energy in harmonically excited cylindrical enclosed sound fields.
7. C. R. Fuller and J. D. Jones 1987 *Journal of Sound and Vibration* **112**, 389–395. Experiments on reduction of propeller induced interior noise by active control of cylinder vibration.
8. H. C. Lester and C. R. Fuller 1986 *AIAA* 10*th Aeroacoustics Conference*, *Seattle*, *WA*, *July* 9–11, 19896, *AIAA 86-1957*. Active control of propeller induced noise fields inside flexible cylinder.
9. P. A. Nelson and S. J. Elliott 1992 *Active Control of Sound*. New York: Academic Press.
10. C. R. Fuller, S. J. Elliott and P. A. Nelson 1996 *Active Control of Vibration*. New York: Academic Press.
11. A. J. Bullmore, P. A. Nelson and S. J. Elliott 1990 *Journal of Sound and Vibration* **140**, 191–217. Theoretical studies of the active control of propeller-induced cabin noise.
12. S. J. Elliottt, P. A. Nelson, I. M. Stothers and C. C. Boucher 1990 *Journal of Sound and Vibration* **128**, 355–357. Preliminary results of in-flight experiments on the active control of propeller-induced cabin noise.
13. D. R. Thomas, P. A. Nelson and S. J. Elliott 1993 *Journal of Sound and Vibration* **167**, 113–128. Active control of the transmission of sound through a thin cylindrical shell, part II: The minimization of acoustic potential energy.
14. R. J. Silcox, C. R. Fuller and H. C. Lester 1987 *AIAA Aeroacoustics Conference*, *Sunnyvale*, *CA*, *October* 19–21, 1987, *AIAA* 87-2703. Mechanisms of active control in cylindrical fuselage structures.
15. C. R. Fuller *U.S. Patent No.* 4715599 Apparatus and method for global noise reduction 1987.
16. M. Bouchard and B. Paillard 1996 *Journal of the Acoustical Society of America* **100**, 3203–3214. A transform domain optimization to increase the convergence speed of the multichannel filtered-x-least-mean-square algorithm.
17. G. Panda, B. Mulgrew, C. F. N. Cowan and P. M. Grant 1986 *IEEE Transactions on Acoustics*, *Speech*, *and Signal Processing* **ASSP-34**, 1573–1582. A self-orthogonalizing efficient block adaptive filter.
18. F. Beaufays 1995 *IEEE Transactions on Signal Processing* **43**, 422–431. Transform-domain adaptive filters: an analytical approach.
19. S. Haykin 1996 *Adaptive Filter Theory* Englewood Cliff, NJ: Prentice-Hall, third edition.
20. M. J. Bronzel and C. R. Fuller 1995 *Active* 95, *Newport Beach*, *CA*, *July* 6–9, 1995, 359–368. Implementation of fast recursive estimation techniques for active control of structural sound radiation.
21. S. C. Douglas 1997 *Proceedings of Noise-Con* 97, *University Park*, *PA*. Reducing the computational and memory requirements of the multichannel filtered-x lms adaptive controller.
22. R. H. Cabell, H. C. Lester, G. P. Mathur and B. N. Tran 1993 *AIAA* 15*th Aeroacoustics Conference*, *Long Beach CA*, *October* 25–27, 1993, *AIAA* 93-4447. Optimization of actuator arrays for aircraft interior noise control.
23. C. R. Fuller and J. P. Carneal 1993 *Journal of the Acoustical Society of America* **93**, 3511–3513. A biologically inspired control approach for distributed elastic systems.
24. D. R. Morgan 1991 *Journal of the Acoustical Society of America* **89**, 248–256. An adaptive modal-based active control system.
25. R. L. Clark 1995 *Journal of the Acoustical Society of America* **98**, 2639–2650. Adaptive feedforward modal space control.
26. S. R. Popovich 1997 U. S. Patent No. 5633795. Adaptive tonal control system with constrained output and adaptation 1997.
27. S. R. Popovich 1996 *Proceedings of Inter-Noise* 96, *Liverpool*, *UK*, 2825–2828. An efficient adaptation structure for high speed tracking in tonal cancellation systems.

28. R. H. Cabell and C. R. Fuller 1995 *Program of the* 129*th Meeting of the Acoustical Society of America* volume 97, 3254–3255. Principal component analysis applied to the active control of turbofan inlet noise.
29. N. R. Draper and H. Smith 1981 *Applied Regression Analysis*. New York: Wiley, second edition.
30. I. L. Jolliffe 1986 *Principal Component Analysis*. New York: Springer.
31. B. Widrow and S. D. Stearns 1985 *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall.
32. S. J. Elliott, C. C. Boucher and P. A. Nelson 1992 *IEEE Transactions on Signal Processing* **40**, 1041–1052. The behavior of a multiple channel active control system.
33. C. C. Boucher, S. J. Elliott and P. A. Nelson 1991 *IEE Proceedings-F* **138**, 313–319. Effect of errors in the plant model on the performance of algorithms for adaptive feedforward control.
34. J. C. Lee and C. K. Un 1986 *IEEE Transactions on Acoustics, Speech, and Signal Processing* **ASSP-34**, 499–510. Performance of transform-domain LMS adaptive digital filters.
35. R. H. Cabell 1998 *Ph.D. thesis, Virginia Tech*. A principal component algorithm for feedforward active noise and vibration control.
36. S. Gilbert 1988 *Linear Algebra and its Applications*. Harcourt Brace Jovanovich.
37. D. J. Rossetti, M. R. Jolly and S. C. Southward 1996 *Journal of the Acoustical Society of America* **99**, 2955–2964. Control effort weighting in feedforward adaptive control systems.
38. C. E. Ruckman and C. R. Fuller 1995 *Journal of the Acoustical Society of America* **97**, 2906–2918. A regression approach for simulating feedforward active noise control.
39. S. D. Snyder and C. H. Hansen 1991 *Journal of Sound and Vibration* **148**, 537–542. Using multiple regression to optimize active noise control system design.
40. B. Widrow and E. Walach 1984 *IEEE Transactions on Information Theory* **IT-30**, 211–221. On the statistical efficiency of the lms algorithm with nonstationary inputs.
41. A. Omoto and S. J. Elliott 1997 *Active* 97, *Budapest Hungary, August* 21–23, 1997, 825–836. The effect of structured plant uncertainty on the stability and performance of multichannel feedforward controllers.
42. R. J. Silcox, H. C. Lester and S. B. Abler 1989 *Journal of Vibration, Acoustics, Stress, and Reliability in Design* **111**, 337–342. An evaluation of active noise control in a cylindrical shell.
43. D. L. Palumbo, S. L. Padula, K. H. Lyle, J. H. Cline and R. H. Cabell 1996 2*nd AIAA/CEAS Aeroacoustics Conference* (17*th Aeroacoustics Conference*), *State College, PA, May* 6–8, 1996, *AIAA* 96-1724. Performance of optimized actuator and sensor arrays in an active noise control system.
44. F. W. Grosveld and T. B. Beyer 1986 *AIAA* 10*th Aeroacoustics Conference, Seattle, WA, July* 9–11, 1986, *AIAA* 86-1908. Modal characteristics of a stiffened composite cylinder with open and closed end conditions.