

where $\boldsymbol{\phi}_i$ is introduced to simplify the notation. Assuming \mathbf{x}_i is an $(n \times 1)$ vector, its derivative will also be an $(n \times 1)$ vector, and therefore can be expressed as a linear combination of the n eigenvectors, \mathbf{x}_k . Writing this out yields

$$\frac{\partial \mathbf{x}_i}{\partial \omega} = \sum_{k=1}^n c_k \mathbf{x}_k = \mathbf{X} \mathbf{c} \quad (6.10)$$

where the columns of \mathbf{X} contain the eigenvectors of \mathbf{A} , and \mathbf{c} is an $(n \times 1)$ vector of weighting coefficients. Substituting this result into Eq. 6.9 yields

$$[\mathbf{A} - \lambda_i \mathbf{I}] \mathbf{X} \mathbf{c} = \boldsymbol{\phi}_i \quad (6.11)$$

Pre-multiplying by \mathbf{X}^H ,

$$\mathbf{X}^H [\mathbf{A} - \lambda_i \mathbf{I}] \mathbf{X} \mathbf{c} = \mathbf{X}^H \boldsymbol{\phi}_i \quad (6.12)$$

which simplifies to

$$[\boldsymbol{\Lambda} - \lambda_i \mathbf{I}] \mathbf{c} = \mathbf{X}^H \boldsymbol{\phi}_i \quad (6.13)$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix with the eigenvalues of \mathbf{A} on its diagonal. All but the i th element of the coefficient vector \mathbf{c} can be determined from Eq. 6.13 by solving the expression

$$c_k = \frac{\mathbf{x}_k^H \boldsymbol{\phi}_i}{\lambda_k - \lambda_i}, \quad i \neq k \quad (6.14)$$

To compute the i th coefficient, the normalization constraint $\mathbf{x}_i^H \mathbf{x}_i = 1$ is applied. Differentiating this constraint,

$$\frac{\partial (\mathbf{x}_i^H \mathbf{x}_i)}{\partial \omega} = \frac{\partial (1)}{\partial \omega} \quad (6.15)$$

$$\frac{\partial \mathbf{x}_i^H}{\partial \omega} \mathbf{x}_i + \mathbf{x}_i^H \frac{\partial \mathbf{x}_i}{\partial \omega} = 0 \quad (6.16)$$

$$2 * \text{Re} \left(\mathbf{x}_i^H \frac{\partial \mathbf{x}_i}{\partial \omega} \right) = 0 \quad (6.17)$$

and substituting Eq. 6.14 into this expression, the i coefficient c_i is found to be

$$c_i = -\text{Re} \left(\mathbf{x}_i^H \sum_{k=1, k \neq i}^n c_k \mathbf{x}_k \right) \quad (6.18)$$

where $\text{Re}()$ indicates the real part of the expression. Thus the $(n \times 1)$ derivative vector can be calculated by solving Eqs. 6.14 and 6.18 for the coefficient vector \mathbf{c} , which is then substituted into Eq. 6.10.

These expressions for the eigenvector derivatives can now be used to compute the derivatives of the singular vectors of the transfer function matrix with respect to frequency.

6.2 Calculation of PC derivatives

Transfer function matrices, denoted $\mathbf{H}(\omega)$, were computed for three control systems across a range of frequencies. In order to compute these matrices, a unit amplitude, single frequency input was applied

to an actuator in the particular control system being studied, and a numerical model of the system being controlled was used to compute the complex amplitude and phase response at each sensor. This was repeated for each actuator, and the resulting complex sensor responses were assembled into a transfer function matrix at the corresponding frequency. This process was repeated at a number of discrete frequencies within a range of frequencies that were deemed to be interesting for each control system.

The derivatives of both the left and right singular vectors were evaluated. To evaluate the derivatives of the right singular vectors in \mathbf{V} , the cross product ($\mathbf{H}^H(\omega)\mathbf{H}(\omega)$) was computed. To compute the derivatives of the left singular vectors in \mathbf{U} , the cross product ($\mathbf{H}(\omega)\mathbf{H}^H(\omega)$) was computed. The matrix $\mathbf{A}(\omega)$ will be used to denote either of these cross products. From the analytical expressions given earlier in this chapter, the derivatives of the eigenvectors of \mathbf{A} depend on $(\partial\mathbf{A}(\omega)/\partial\omega)$. This quantity was computed using the finite difference approximation

$$\frac{\partial\mathbf{A}(\omega)}{\partial\omega} \approx \frac{\mathbf{A}(\omega + \Delta\omega) - \mathbf{A}(\omega - \Delta\omega)}{2(\Delta\omega)} \quad (6.19)$$

where $\Delta\omega$ denotes the spacing between discrete frequencies. This is a reasonably accurate approximation except near the natural frequency of a lightly damped mode, so the frequency spacing $\Delta\omega$ was set sufficiently small for each of the three modeled control systems to reduce errors near natural frequencies. The eigenvector derivative is a vector of dimension $(m \times 1)$ for the eigenvectors in \mathbf{U} and of dimension $(r \times 1)$ for the eigenvectors in \mathbf{V} . Only the norm of the derivative in either case is plotted to simplify the presentation of the results. The frequency variation of the phase of individual terms of the eigenvectors is plotted in a few cases where it provides necessary insight.

More detailed analyses of the three control systems can be found in Appendix A.

6.2.1 Simply supported plate

The first of the three models studied here is a simply supported rectangular steel plate whose bending response was modeled using a summation of the first eighteen plate modes; the analytical model of the plate is discussed in greater detail in Appendix A. The material properties of the plate are listed in Table 6.1; the natural frequencies of the first nine modes of the plate are listed in Table 6.2. The dimensions of the plate, L_x and L_y , were defined in order to separate the natural frequencies of the first few modes of the plate. The transfer function matrices were computed from 60 Hz to 500 Hz in 0.22 Hz increments. The small frequency spacing was needed to reduce errors in the finite difference approximation in Eq. 6.19 at the natural frequencies of the plate.

The control system for the plate consisted of five accelerometers and three point-force actuators in the arrangement shown in Fig. 6.1. Note that these components were not used to actually control any disturbances on the plate, but were only used to designate the type and location of sensors and actuators for generating the transfer function matrices. The three actuators were evenly distributed along the x -dimension of the plate, at the midpoint of the plate in the y -direction. With this arrangement, the even modes in the y direction were not controllable because the actuators were on the nodes of those modes. This reduced the number of modes contributing to the response, thereby simplifying the response plots. The five sensors were evenly spaced along the diagonal of the plate to observe the two-dimensional nature of the plate modes. The size of the control system was chosen as a trade-off between having too few actuators and sensors, making the results difficult to generalize to large systems, and having too many actuators and sensors, which would needlessly complicate the results.

In relating the principal components to the dynamics of the structure being controlled, it is useful

Table 6.1: Plate properties

thickness, h	0.002 m
Young's modulus, E	2.e11
damping, η	0.03
Poisson's ratio, ν	0.3
x-length, L_x	0.38 m
y-length, L_y	0.27 m
density, ρ	7833 kg/m^3

Table 6.2: Natural frequencies

Frequency (Hz)	Mode (n_x, n_y)
99.0	(1,1)
199.0	(2,1)
296.9	(1,2)
365.3	(3,1)
396.7	(2,2)
563.1	(3,2)
598.1	(4,1)
626.4	(1,3)
726.2	(2,3)

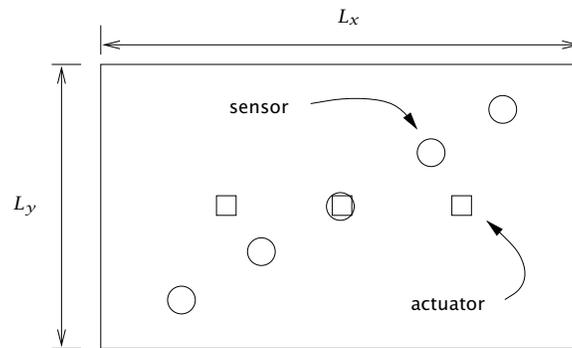


Figure 6.1: Actuator and sensor placement on simply supported plate

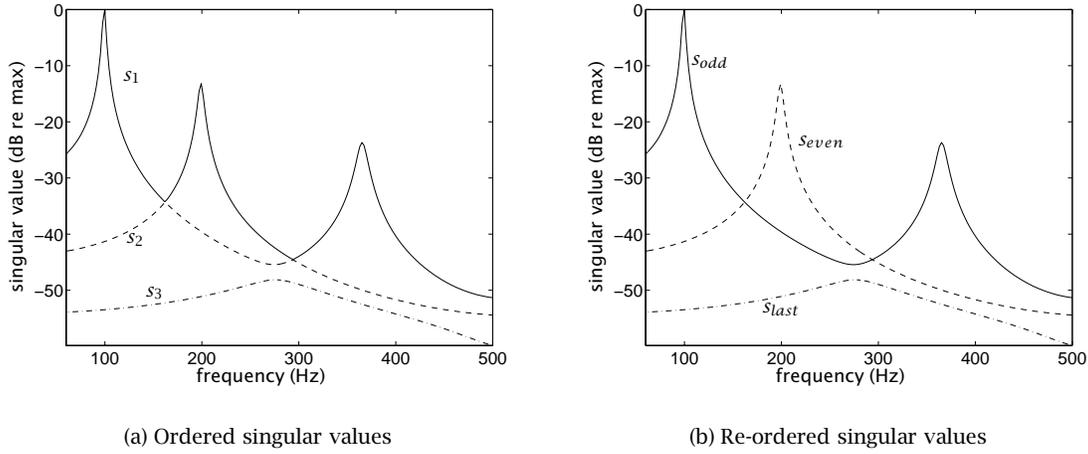


Figure 6.2: Singular value variation with frequency for control system on a simply supported plate

to recall that the singular vectors represent an orthogonal decoupling of the row and column spaces of the transfer function matrix at a single frequency. If there are a very large number of actuators and sensors spread uniformly over the structure being controlled, then it is possible that the principal components will correspond very closely to the actual mode shapes of the structure, at least in frequency ranges where the spacing of the sensors and actuators is small relative to the wavelength of the dominant modes. The shapes of the PCs in this case would be insensitive to frequency and would therefore constitute a useful decoupling transformation over a wide range of frequencies. As the spacing between sensors and actuators becomes large relative to the wavelength of a mode on the structure, they will no longer sample the dominant modes with sufficient precision to completely decouple those modes from one another. In this case, the PCs become mathematically convenient decoupling transformations that bear relatively little resemblance to the underlying modes of the structure.

The singular values of the transfer function matrix for the control system are plotted as a function of frequency in Fig. 6.2(a). The transfer function matrix at a single frequency had dimensions (5×3) , hence the SVD of \mathbf{H} produced three singular values at each frequency, labeled s_1 through s_3 in the plot. The values are plotted on a logarithmic scale, and were normalized relative to the maximum singular value near 100 Hz. Although the singular values are usually ordered from largest to smallest as in the plot, a more useful ordering that emphasizes their continuous variation with frequency is shown in Fig. 6.2(b). The curves are labeled according to the peaks in the plot; the curve for s_{odd} peaks at the natural frequencies of the odd-odd (1,1) and (3,1) modes. The curve for s_{even} has a peak at the natural frequency of the even-odd (2,1) mode, while s_{last} does not have a clear association with the natural frequency of a mode.

The curves for s_{odd} and s_{even} resemble frequency response functions from an individual actuator to an individual sensor for a one- or two-degree of freedom system. This behavior can be partially explained by considering the definition of the SVD of a transfer function matrix. By definition of the SVD, the matrix \mathbf{H} for this particular control system can be written

$$\mathbf{H} = \mathbf{u}_1 s_1 \mathbf{v}_1^H + \mathbf{u}_2 s_2 \mathbf{v}_2^H + \mathbf{u}_3 s_3 \mathbf{v}_3^H \quad (6.20)$$

where \mathbf{u}_i and \mathbf{v}_i are, respectively, the i th columns of the matrices \mathbf{U} and \mathbf{V} from the SVD of \mathbf{H} . These

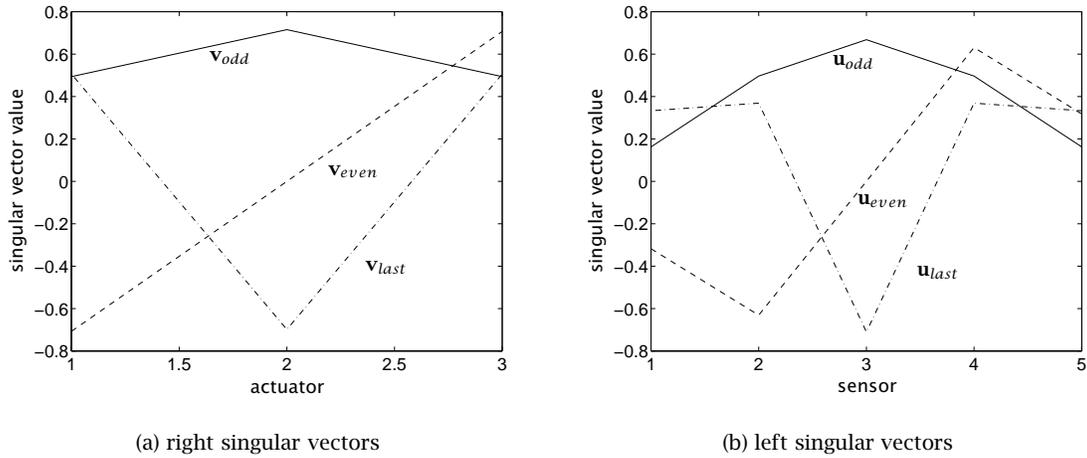


Figure 6.3: Singular vectors at 80 Hz

columns have unit length, so magnitude variation in \mathbf{H} from frequency to frequency is described by the singular values, s_1, s_2 , and s_3 . The response of the simply supported plate to a unit amplitude input to an actuator is higher at resonance than off resonance, assuming the actuator couples into the resonant mode and the sensors can observe the resulting response. This accounts for the peaks and valleys in the singular values in Fig. 6.2.

The singular vectors \mathbf{u}_i and \mathbf{v}_i quantify the transformation from sensor and actuator responses to the response of the i th principal component. For example, \mathbf{u}_1 is an $(m \times 1)$ vector of complex values that describes a weighting of the sensor inputs that gives the response of the first principal component of the control system. The values of the individual elements of \mathbf{u}_1 should therefore provide some indication of the association between the first PC and the dynamics of the plate. To investigate this relationship, the values in the singular vectors are plotted at three frequencies close to but below the resonances of the (1, 1), (2, 1), and (3, 1) modes of the plate. The frequencies were chosen to be away from resonance because the elements of the singular vectors are predominantly real off resonance, and are complex on resonance. This is due to damping in the plate model, which is significant only very close to the natural frequency of a mode. The fact that the singular vectors are primarily real off resonance is not a general property of the singular vectors of any transfer function matrix, but is true here due to the simplicity of the plate system and the arrangement of actuators and sensors on the plate.

The shapes of the singular vectors of \mathbf{H} at 80 Hz are shown in Fig. 6.3. The vectors are labeled to correspond to the singular values in Fig. 6.2(b). At this frequency, which is below the natural frequency of the (1, 1) mode, the shape of \mathbf{v}_{odd} and \mathbf{u}_{odd} corresponds to the shape of the (1, 1) mode at the actuators and sensors, respectively. Therefore at this frequency, the odd principal component is associated with the response of the (1, 1) mode of the plate. The shape of \mathbf{v}_{even} and \mathbf{u}_{even} corresponds to the shape of the (2, 1) mode, and thus the even principal component is associated with the response of the (2, 1) mode at this frequency. The shape of \mathbf{v}_{last} and \mathbf{u}_{last} resembles the shape of the (3, 1) mode so a similar association can be made for the last principal component.

The association between principal components and individual modal responses should be made with some care because in reality the three principal components describe contributions from all 18

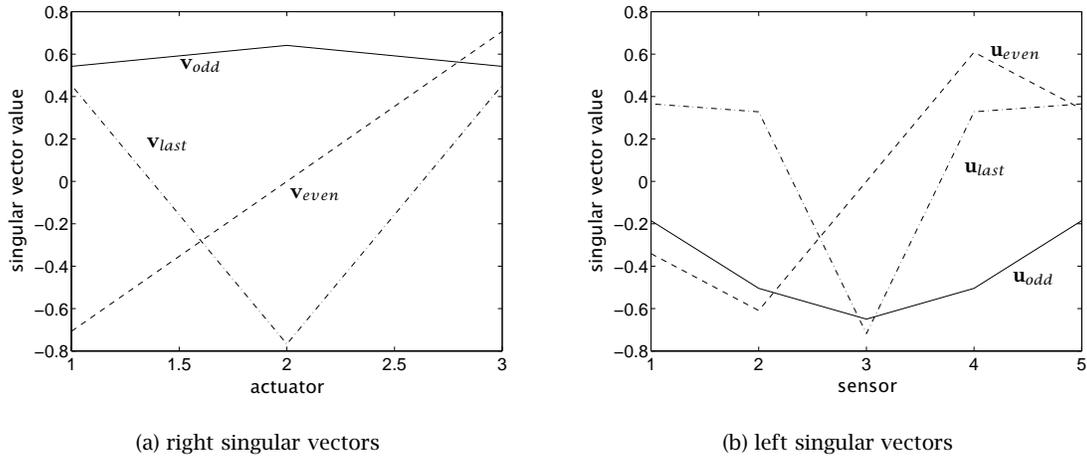


Figure 6.4: Singular vectors at 180 Hz

modes in the plate model. Most of these modes contribute very little to the plate response at 80 Hz, so the error in making the correspondence between individual PCs and modes is small. Nonetheless, in general it would be incorrect to assert a strict one-to-one relationship between a principal component and an individual mode.

Proceeding from 80 Hz to 180 Hz, the shapes of the singular vectors shown in Fig. 6.4 are almost unchanged. The even and last singular vectors appear to be nearly identical at 180 Hz and 80 Hz. The only difference between the odd singular vector at the two frequencies is that \mathbf{u}_{odd} appears to be flipped by 180° in Fig. 6.4(b) relative to Fig. 6.3(b). This is a result of passing through the natural frequency of the (1, 1) mode and the corresponding 180° degree phase shift between the input and output response of that mode.

Continuing further up to 323 Hz, which lies between the natural frequencies of the (2, 1) and (3, 1) modes, we see additional changes in the shapes of the vectors. The most notable change is that the shapes of \mathbf{v}_{odd} and \mathbf{v}_{last} appear to have switched somewhere between 180 and 323 Hz. At 180 Hz, \mathbf{v}_{odd} and \mathbf{v}_{last} resembled the shapes of the (1, 1) and (3, 1) modes, respectively. However, at 323 Hz the opposite is true; \mathbf{v}_{odd} resembles the (3, 1) mode while \mathbf{v}_{last} resembles the (1, 1) mode. In addition, the phase between \mathbf{v}_{even} and \mathbf{u}_{even} has flipped by 180° due to the natural frequency of the (2, 1) mode at 199 Hz, with which the even PC is associated. The shape of \mathbf{u}_{last} is no longer easily correlated with the shape of one of the first four modes of the plate. This could indicate the last principal component is partly associated with the decreasing response of the (1, 1) mode and the increasing response of a higher order mode, such as the (1, 3) mode whose natural frequency is at 626 Hz.

The progression of the eigenvector shapes from 80 Hz to 323 Hz offers some insight into the behavior of the eigenvector derivatives in this frequency range. The shapes remained relatively constant from 80 to 180 Hz, except for a 180° phase shift in the odd PC, so it is likely that the eigenvector derivatives are small in this frequency range. From 180 to 323 Hz, the shapes of the odd and last principal components swapped places, so this probably corresponds to a high derivative for these two eigenvectors somewhere between these two frequencies. The shape of the even principal component remained relatively constant across all three frequencies, so its derivative is likely to be small from 80 to 323 Hz.

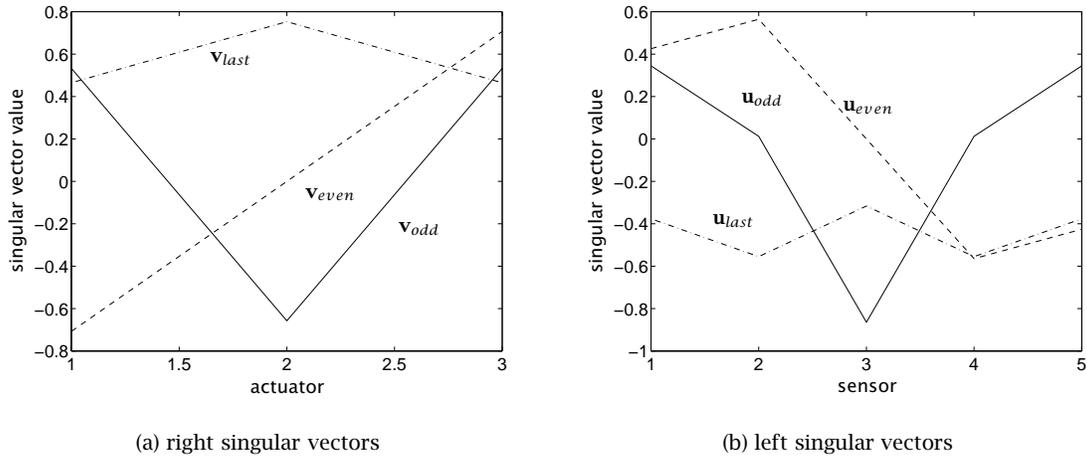


Figure 6.5: Principal component vectors at 323 Hz

The derivatives of the singular vectors in the columns of \mathbf{V} and \mathbf{U} are shown in Fig. 6.6(a) and 6.6(b), respectively. The norm of the derivatives are plotted, hence the values in the plot represent the rate of change of the shape of the vectors as a function of frequency. Because the singular vectors have unit length by definition, the derivatives can be interpreted as the rate of change as a percentage of the length of each vector. A single curve is visible in Fig. 6.6(a) because the derivatives of \mathbf{v}_{odd} and \mathbf{v}_{last} are equal throughout the frequency range of the plot, and the derivative of \mathbf{v}_{even} is zero. The derivatives of only the first three columns of \mathbf{U} are shown in Fig. 6.6(b) since only these PCs are controllable. Below about 225 Hz, the derivatives of the left and right singular vectors are below 0.005, which indicates the vectors were changing less 0.5% of their total length as the frequency changed. This very small rate of change verifies that the eigenvector shapes were nearly constant from 80 to 180 Hz. The natural frequencies of the plate modes at 99, 199, and 365 Hz do not appear to have caused an increase in the eigenvector derivatives.

The highest rate of change in either plot occurs just below 300 Hz for the odd and last principal components. The high derivative can be explained by noting that the odd and last PCs, at frequencies below 300 Hz and just above 300 Hz, are associated with the responses of two modes that are not orthogonal at both the sensors and the actuators. The eigenvector shapes shown in Figs 6.3 and 6.4 suggest that the odd PC was associated with the response of the (1, 1) mode and the last PC with the response of the (3, 1) mode, at 80 and 180 Hz. These two modes are not orthogonal at the sensors. As a result, when these two modes contribute nearly equally to the plate response, just below 300 Hz where s_{odd} and s_{last} are nearly equal, the singular vectors correspond to mathematically convenient decouplings that are not related to one mode or the other. As the frequency changes below 300 Hz and the relative responses of the (1, 1) and (3, 1) modes change, the decoupling changes as well, hence the high rate of change of the odd and last singular vectors in this frequency range.

This type of interaction is absent between the odd and even principal components near 175 Hz, where the responses of the (1, 1) and (2, 1) modes contribute equally to the plate response. These two modes are orthogonal at the sensors and actuators, so decoupling the contributions of the two modes when they contribute nearly equally does not depend on the relative levels of the two modes.

The derivatives of the left singular vectors in Fig. 6.6(b) are slightly higher and change more rapidly

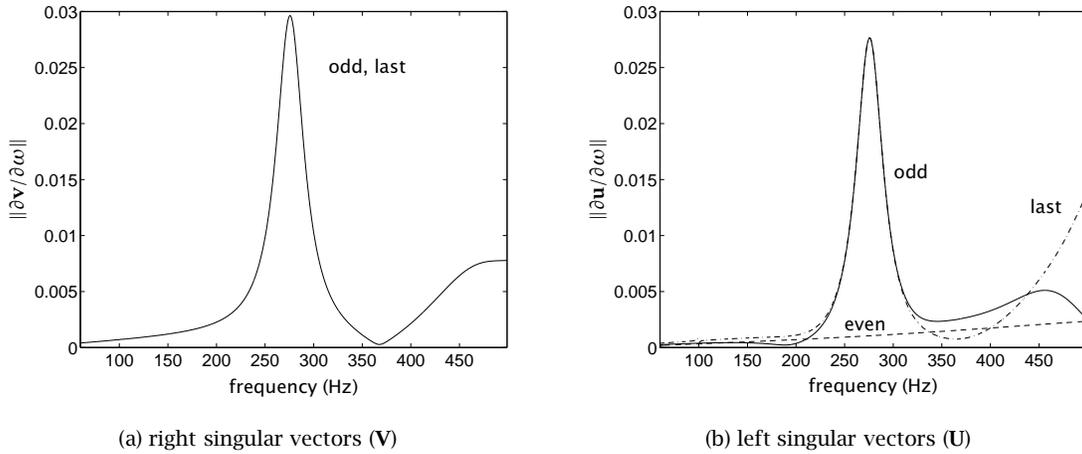


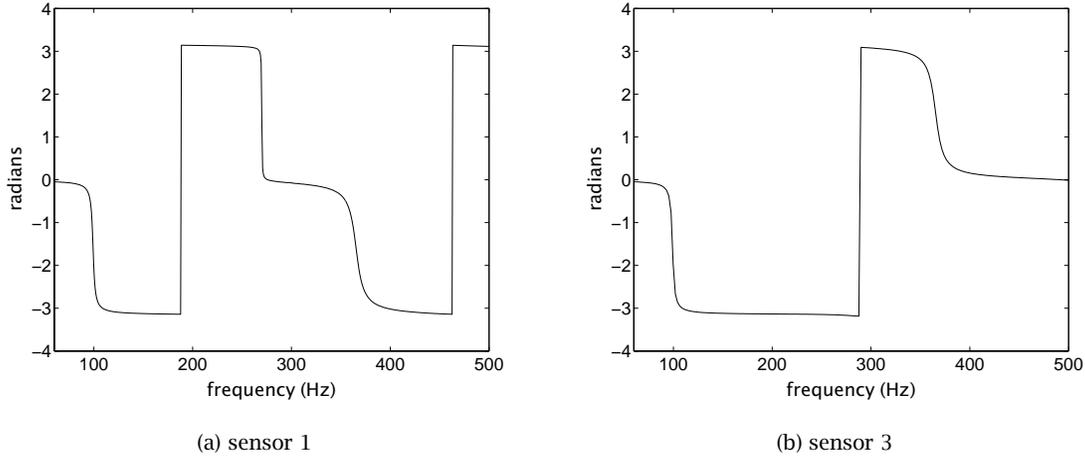
Figure 6.6: Derivatives of singular vectors for vibrating plate

at higher frequencies than the derivatives of the right singular vectors in Fig. 6.6(a). The sensors are located on the diagonal of the plate and can observe the 2-dimensional shape of the plate modes, whereas the actuators are at a constant y location and can only couple into variation in the x direction. For example, \mathbf{v}_{even} is zero throughout the frequency range in Fig. 6.6(a), but is greater than zero in Fig. 6.6(b). This is likely due to the fact that near 500 Hz, \mathbf{u}_{even} is partially associated with the increasing response of a higher order plate mode.

The plots of the eigenvector derivatives indicate the eigenvectors do not change shape at the natural frequencies of the first few plate modes, but nonetheless there is an important change in phase at these frequencies. The phase of the first and third elements of the $(m \times 1)$ singular vector \mathbf{u}_{odd} is shown in Fig. 6.7. The first and third elements of \mathbf{u}_{odd} correspond to weighting coefficients for the response of the first sensor, located at the corner of the plate, and the third sensor, located at the middle of the plate. The phase of both elements changes by 180° at the natural frequencies of the (1, 1) and (3, 1) modes at 99 and 365 Hz, respectively. This phase change occurs in every element of \mathbf{u}_{odd} , so the shape of the vector does not change at resonance, hence the derivative of \mathbf{u}_{odd} does not increase at resonance. Just below 300 Hz, however, the phase of the first element changes by 180° but the phase of the third element is unchanged. This does correspond to a change in the shape of the eigenvector and hence the derivative of \mathbf{u}_{odd} increases at this frequency.

The eigenvector derivatives in Fig. 6.6 and the phase plot in Fig. 6.7 illustrate two changes to the singular vectors that could affect the stability of convergence of the PC-LMS algorithm. The convergence could be affected if the transfer function matrix was measured at one frequency but the controller operated at a second frequency. For example, consider the case where the transfer function matrix was measured at 89 Hz but the controller was operating at 109 Hz. The natural frequency of the (1, 1) mode lies between these two frequencies and the 180° phase change in the odd PC could cause convergence to go unstable. A second scenario to consider is if the transfer function matrix were measured at 245 Hz and the controller operated at 305 Hz. The peak in the derivative of the odd and last PCs lies between these two frequencies, so the shape change that occurs at about 275 Hz could also cause convergence to go unstable.

The stability of convergence for these two hypothetical cases will be quantified using two criteria.

Figure 6.7: Phase of \mathbf{u}_{odd} at sensors 1 and 3

The first is approximate, suggested by Omoto and Elliott (1997), and was given in Eq. 4.49. The second criterion is exact and was discussed in Morgan (1980) and Boucher et al. (1991). It is not necessary to use both criteria to establish the stability of convergence, but doing so provides the opportunity to evaluate the accuracy of the approximate criterion. For this discussion we assume the transfer function matrix is measured accurately at a frequency ω_1 , but the controller operates at a different frequency, ω_2 . The measured transfer function matrix, $\mathbf{H}(\omega_1)$, thus contains errors relative to the true transfer function matrix, $\mathbf{H}(\omega_2)$. The SVD of $\mathbf{H}(\omega_2)$ can be written in terms of the singular values and vectors of $\mathbf{H}(\omega_1)$ as

$$\mathbf{H}(\omega_1) = \mathbf{U}_1 \mathbf{S} \mathbf{V}_1^H \quad (6.21)$$

$$\mathbf{H}(\omega_2) = \mathbf{U}_1 (\mathbf{S} + \Delta \mathbf{S}) \mathbf{V}_1^H \quad (6.22)$$

The approximate convergence criterion from Omoto and Elliott (1997) states that convergence should be stable at the second frequency if

$$s_i > -\text{Re}(\Delta s_{ii}) \quad (6.23)$$

where Δs_{ii} is the (i, i) th element of $\Delta \mathbf{S}$ and s_i is the i th singular value from Eq. 6.21. The exact convergence criterion specifies that convergence is stable if the real parts of the eigenvalues of the matrix $(\mathbf{H}^H(\omega_1) \mathbf{H}(\omega_2))$ are positive.

The first case considered is for two frequencies on either side of the resonance of the (1, 1) plate mode at 99 Hz. Let $\omega_1 = 89$ Hz and $\omega_2 = 109$ Hz. Computing $\mathbf{H}(\omega_1)$ at 89 Hz using the plate model described in Appendix A, its singular values are found to be

$$10^{-4} * [0.68 \quad 0.03 \quad 0.01] \quad (6.24)$$

The matrix $\mathbf{H}(\omega_2)$ was then computed at 109 Hz, and the matrix $\Delta \mathbf{S}$ was computed by rearranging

Eq. 6.22. The resulting matrix $\Delta\mathbf{S}$ is

$$\Delta\mathbf{S} = 10^{-3} * \begin{bmatrix} -0.13 - 0.02i & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. \end{bmatrix} \quad (6.25)$$

Only the element Δs_{11} is non-zero, which indicates the singular vectors computed from \mathbf{H} at 89 Hz still decouple the transfer function matrix at 109 Hz. However, the fact that Δs_{11} is negative and approximately two times s_1 listed in Eq. 6.24 indicates the phase between \mathbf{u}_1 and \mathbf{v}_1 changed by 180° between the two frequencies, as expected. Comparing Δs_{11} and s_1 reveals that the convergence criterion from Eq. 6.23 is not satisfied, so convergence would be unstable if the transfer function matrix from 89 Hz were used at 109 Hz. To verify this conclusion, the eigenvalues of $(\mathbf{H}^H(\omega_1)\mathbf{H}(\omega_2))$ were computed and found to be

$$10^{-8} * [-0.4330 + 0.13i \quad 0.0013 + 0i \quad 0.0001 + 0i] \quad (6.26)$$

The real part of the first eigenvalue is less than zero, so this confirms that convergence will not be stable. This example illustrates where assuming the stability of convergence is determined by the diagonal elements of $\Delta\mathbf{S}$ is clearly valid, since all other elements of $\Delta\mathbf{S}$ are zero.

Note that convergence could be stabilized in this example, although with a large performance penalty, by not controlling the first PC. This is the only PC that would go unstable if the frequency were to change from 89 to 109 Hz, so eliminating it from the controller would stabilize convergence of the PC-LMS algorithm.

The second case considered is for two frequencies on either side of the peak in the derivative of the odd and last singular vectors near 275 Hz. Let $\omega_1 = 245$ Hz and $\omega_2 = 305$ Hz. The singular values for $\mathbf{H}(\omega_1)$ at 245 Hz are

$$10^{-5} * [0.54 \quad 0.26 \quad 0.14] \quad (6.27)$$

The matrix $\Delta\mathbf{S}$ that gives $\mathbf{H}(\omega_2)$ as a function of the singular vectors of $\mathbf{H}(\omega_1)$ is

$$\Delta\mathbf{S} = 10^{-5} * \begin{bmatrix} -0.32 - 0.01i & 0. & 0. \\ 0. & -0.083 - 0.002i & -0.086 + 0.01i \\ 0. & -0.036 & 0.0986 - 0.01i \\ -0.013 & -0.001 & 0.006 \\ 0.004 & -0.004 & 0.0224 - 0.003i \end{bmatrix} \quad (6.28)$$

The presence of off-diagonal elements in this matrix indicates there was a change in the shape of the singular vectors between the two transfer function matrices. As a result, the singular vectors computed from \mathbf{H} at 245 Hz do not decouple the transfer function matrix at 305 Hz. Nonetheless, comparing the diagonal elements of $\Delta\mathbf{S}$ with the singular values in Eq. 6.27 shows that convergence should be stable according to the approximate criterion in Eq. 6.23. To confirm this result, the eigenvalues of $(\mathbf{H}^H(\omega_1)\mathbf{H}(\omega_2))$ were computed and were

$$10^{-10} * [0.116 + 0.004i \quad 0.0535 + 0.001i \quad 0.0287 + 0.006i] \quad (6.29)$$

The real parts of all of the eigenvalues are positive, which confirms that convergence would be stable if the transfer function matrix measured at 245 Hz were used to control disturbances at 305 Hz. Thus even when the shapes of the singular vectors change with frequency, and no longer decouple the control system, convergence of the PC-LMS algorithm can remain stable.

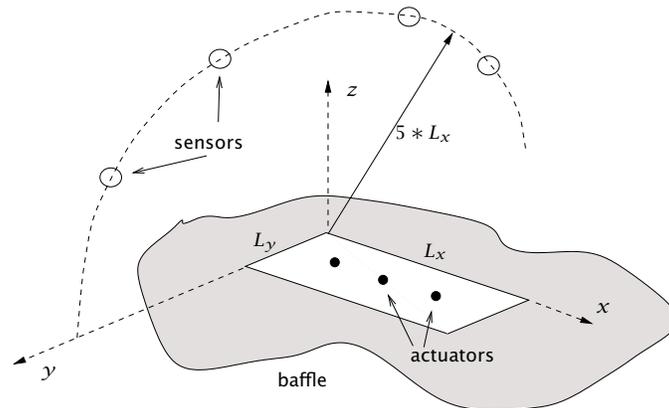


Figure 6.8: Actuator and sensor placement for radiating plate

6.2.2 Radiating plate

The second model consisted of the plate from Section 6.2.1 radiating sound into the half-space above the plate. The control system consists of point-force actuators on the plate and pressure transducers on a hemispherical surface in the far-field. The response of the pressure sensors due to an actuator input on the plate cannot be described by a summation of fixed mode shapes, as for the case where the sensors are accelerometers on the plate. Instead, the sensor responses depend on the sound radiated by the flexural modes of the plate. The nature of the sound radiation problem has a significant effect on the phase as a function of frequency.

A partial schematic of the control system is shown in Fig. 6.8, which consists of sensors and actuators mounted on and around a simply supported plate in a baffle. Three point force actuators were evenly spaced along the x dimension of the plate, at $L_y/2$. This arrangement simplified the transfer functions since none of the even plate modes in the y direction were excited. Eight pressure sensors were located in the far field on a hemispherical surface centered over one corner of the plate. The eight sensors were divided into two sets; four of the sensors were evenly distributed from 0° to 180° on a semicircular arc in the $y-z$ plane, as pictured in the figure. Four other sensors were distributed on a similar arc in the $x-z$ plane. The far field assumption for this case is satisfied if the distance from the microphones to the plate is much greater than the larger of the two plate dimensions (Fahy 1985), so the radius of each semicircular arc was set at $(5 * L_x)$.

The pressure at the sensors due to the flexural motion of the plate was computed using the Rayleigh integral (Wallace 1972), and the details of the model are discussed in Appendix A. The material properties of the plate are listed in Table 6.1. The fluid surrounding the plate was assumed to be air at standard conditions, with a density of 1.21 kg/m^3 and a sound speed of 343 m/s . Fluid loading of the plate was neglected. Transfer functions from actuators to sensors were computed at discrete frequencies from 60 to 500 Hz with a frequency spacing of 0.22 Hz.

The singular values of the transfer function matrix for the control system are shown in Fig. 6.9. The curves are plotted and labeled to emphasize the continuous nature of the singular values with frequency. The curves resemble those seen earlier for the vibrating plate, with peaks at the resonances of the (1,1), (2,1), and (3,1) modes at 99, 199, and 365 Hz, respectively. The singular values are labeled according to the peaks in the plot: the odd singular value peaks at the natural frequencies of the (1,1) and (3,1) modes; the even singular value peaks at the natural frequency of the (2,1) mode;

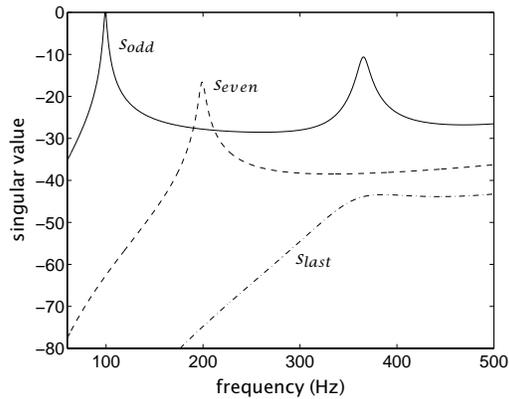


Figure 6.9: Singular values vs. frequency

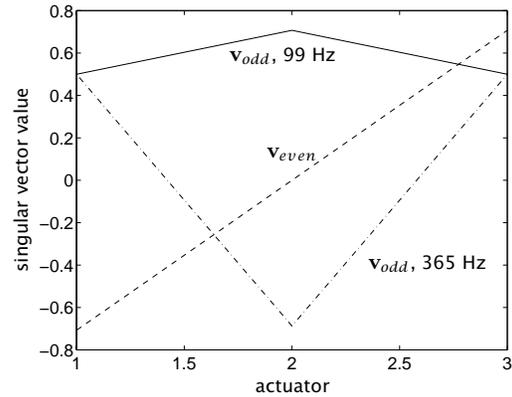


Figure 6.10: Shape of right singular vectors

and the last singular value is not clearly associated with the resonance of any mode.

In order to examine the association between the the singular values and the mode shapes on the plate, the shapes of the right singular vectors, \mathbf{v}_{odd} and \mathbf{v}_{even} , are shown in Fig. 6.10. The left singular vectors, \mathbf{u}_{odd} and \mathbf{u}_{even} , are not shown because it is difficult to plot their three-dimensional shape, and they do not provide additional insight. The shape of \mathbf{v}_{odd} is shown at 99 Hz and 365 Hz, and at each of these frequencies it resembles the corresponding resonant mode shape at the actuators. For example, at 99 Hz it resembles the shape of the (1, 1) mode, and at 365 Hz it resembles the shape of the (3, 1) mode. The shape of \mathbf{v}_{even} is constant throughout the frequency range of Fig. 6.9, and it resembles the shape of the (2, 1) mode at the actuators.

Differences between the frequency variation of the singular values in Fig. 6.9 and the variation of singular values for the vibrating plate are due to the effects of sound radiation in the transfer function. For example, the (1, 1) plate mode radiates like a monopole acoustic source, and therefore it radiates sound more efficiently than the (2, 1) mode, which radiates like a dipole source. The net effect is that the odd singular value, which is associated with the monopole radiator, is the dominant singular value for a broad frequency range. The even singular value is dominant only for a relatively small frequency range about the natural frequency of the (2, 1) mode. The radiation efficiency also accounts for the greater dynamic range between the maximum and minimum singular values at any given frequency in Fig. 6.9, relative to the dynamic range of the singular values for the vibrating plate in Fig. 6.2(b).

The derivatives of the left and right singular vectors as a function of frequency are shown in Fig. 6.11. As for the vibrating plate, only the norms of the derivative vectors are plotted, so the values in the plot indicate the rate of change of the shape of the singular vectors. The derivatives of \mathbf{v}_{odd} and \mathbf{v}_{last} are equal in Fig. 6.11(a), while the derivative of \mathbf{v}_{even} is zero. The large dynamic range of the singular values of the transfer function created numerical problems at the natural frequencies of the plate modes. These problems are evident in Fig. 6.11(b) near 199 Hz. The derivatives of the even and last singular vectors are not plotted below about 125 Hz due to very large numerical errors at the natural frequency of the (1, 1) mode produced by the finite difference approximation in Eq. 6.19.

For frequencies below 250 Hz, the derivatives of the singular vectors are very small. The values in both plots indicate the shapes of the singular vectors changed at a rate less than 0.5% of the total length of each singular vector with frequency. The maximum derivative occurs near the natural frequency of the (3, 1) mode at 365 Hz. This peak is probably due to an interaction between the (1, 1) mode and the (3, 1) mode, as seen with the vibrating plate. However, in this example the peak occurs

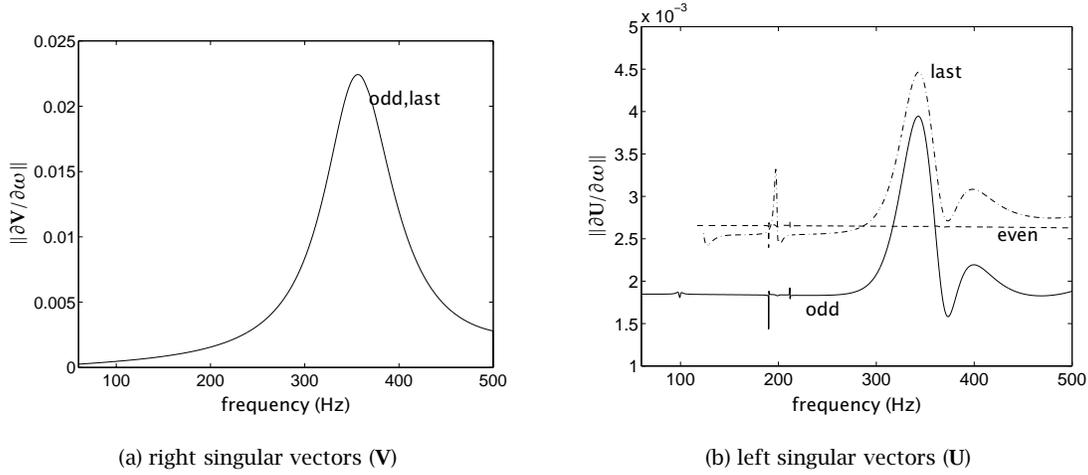


Figure 6.11: Derivatives of singular vectors for radiating plate

much closer to the natural frequency of the (3,1) mode, because the low radiation efficiency of that mode, relative to the (1,1) mode, means it does not become significant in the transfer function matrix until very close to its natural frequency.

Below about 250 Hz, the rate of change of \mathbf{u}_{odd} and \mathbf{u}_{even} can be related to directionality effects. Directionality refers to the variation in the far-field pressure amplitude as a function of the angle off the normal to the radiating surface. For example, below 250 Hz, \mathbf{u}_{odd} is associated with the radiation due to the (1,1) mode. The beam pattern of acoustic radiation from this mode varies with frequency, changing from a nearly uniform, semicircular pattern for low frequencies, to a more complicated, multi-lobed pattern for higher frequencies. The rate of change of the shape of \mathbf{u}_{odd} is due to this changing beam pattern. Similarly, the rate of change of \mathbf{u}_{even} corresponds to the changing radiation characteristic of the (2,1) mode.

The variation of the phase of individual elements of the singular vectors with frequency is very important for the radiating plate. The phase of the second element of \mathbf{u}_{odd} is shown in Fig. 6.12 as a function of frequency. The second element of \mathbf{u}_{odd} corresponds to an error sensor in the $x - z$ plane at 54° off the normal to the plate. The phase variation is nearly linear with frequency, with two exceptions. The first is a 180° change at 99 Hz, the natural frequency of the (1,1) mode, and the second is near the natural frequency of the (3,1) mode at 365 Hz. The linear phase variation is simply due to propagation effects. Specifically, the pressure at a point a distance r from a pulsating sphere is (Kinsler et al. 1982)

$$p(r, t) \propto \frac{1}{r} e^{j(\omega t - kr)} \quad (6.30)$$

where ω is the frequency of the vibrations, and k is the acoustic wavenumber, ω/c_f (c_f is the speed of sound). The phase of $p(r, t)$ varies with frequency, and this variation is given by

$$\phi = \tan^{-1} \left(\frac{\sin kr}{\cos kr} \right) \quad (6.31)$$

which is a linear function of frequency. If this variation is plotted, it exactly equals the rate of change of the phase in in Fig. 6.12, once the phase discontinuities due to the resonances are removed.

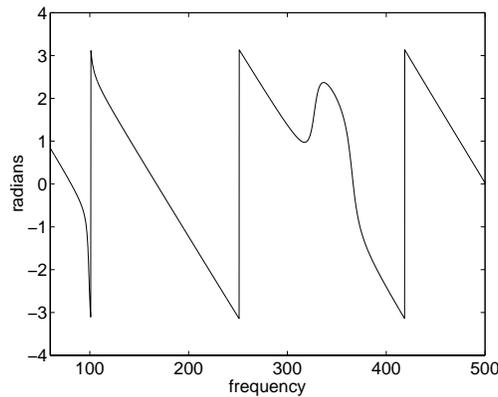


Figure 6.12: Phase of \mathbf{u}_{odd} at second sensor

The phase variation of the individual elements of the principal components is thus a very important part of the overall frequency sensitivity of the singular vectors for the radiating plate. If the PC-LMS controller were used to control radiation from a simply supported plate, the frequency would have to be very stable to keep the phase of the singular vectors from changing, causing the controller to go unstable. A possible solution to this frequency sensitivity would be to identify the transfer function matrix online, while control was being applied. As long as the online modeling took place faster than the frequency changed, the PC-LMS controller would remain stable.

The phase variation also illustrates why the controller shown in Fig. 6.8 may not be the best arrangement of sensors and actuators for controlling sound radiated from a vibrating plate. The transfer function matrix varies rapidly with frequency, hence any model of the dynamics of the controller will have to model this phase variation. In addition, it may not always be practical to mount sensors in the far-field in order to monitor the performance of the controller. For these reasons, an alternative control approach that relies on structural sensors instead of acoustic sensors to reduce radiated power from a plate may be more useful for this particular control problem (Baumann et al. 1991; Elliott and Johnson 1993; Burdisso and Fuller 1994; Cunefare 1991).

6.2.3 Cylindrical shell

The third model studied was a simply supported, thin-wall cylindrical shell with rigid end caps, forming a closed cylindrical cavity. The coupling between the cylinder response and the acoustic response of the cavity was modeled. The model can thus be used to evaluate control systems for reducing sound inside a structure that approximates an aircraft fuselage. The model was originally created to evaluate the coupling between the cylinder and the interior cavity for different configurations of piezoelectric (PZT) actuators mounted on the walls of the cylinder (Lester and Silcox 1992; Lester and Lefebvre 1993). The philosophy behind this control approach is to use force actuators mounted on the fuselage structure to reduce sound transmission through the structure (Fuller and Jones 1987).

The response of the cylinder and the enclosed cavity are described by different modal basis functions that are coupled to one another at the interior cylinder wall. This problem thus differs from the sound radiated from a vibrating plate discussed earlier, because the sound radiated into the infinite half-space above the plate could not be described by mode shapes. The transfer functions from points on the cylinder to pressures inside the cavity are influenced by structural resonances of the shell and

acoustic resonances of the cavity. The coupling between the structure and the acoustic response of the cavity is very selective, which simplifies the transfer functions to some degree. In fact, the acoustic resonances dominate the transfer function matrices for the current model in the frequency range of interest.

It should be noted that the coupling between the cylinder and the cavity is quite complicated, hence, the results presented here cannot be analyzed with the same detail used to analyze the results from either of the two plate models discussed earlier. For example, the curvature of the cylinder increases the phase velocity of certain structural modes such that these modes become efficient radiators of sound at much lower frequencies than would be expected due to flat plate theory alone (Fahy 1985). Thus in considering the coupling between the shell and the interior acoustic space at a given frequency, several structural modes with different modal orders could be responsible for the sound field. In spite of these complications, the model is useful for illustrating that the frequency variation of the PCs that was observed on the simpler plate models carries over to this more complicated dynamical system. These results provide a measure of confidence that the PC-LMS controller can be applied to a complex structure because the PCs will vary in a deterministic manner.

Transfer functions were computed from eight pairs of PZT patches mounted on the cylinder to the acoustic pressure at forty-eight points inside the cylinder. This control system is much larger than the controllers used on the plate models, but it was used because it resembles the controller used to experimentally verify the PC-LMS algorithm on a cylindrical structure (described in Chapter 8). A single actuator pair consisted of two PZT patches mounted on opposite sides of the cylinder wall; one on the outside surface and one on the inside surface. The two patches were driven in phase with one another to produce an in-plane stress distribution in the shell. This phasing of actuator inputs produces a favorable distribution of low order structural modal responses that is useful for interior noise control (Lester and Lefebvre 1993). The transfer functions between PZTs and interior pressures were computed over a frequency range of 75 to 250 Hz, in 0.5 Hz increments. These frequencies correspond to non-dimensional frequencies (with respect to the ring frequency of the cylinder) of 0.073 to 0.24. At these relatively low non-dimensional frequencies, the effect of curvature on the sound radiated from structural modes is significant (Fahy 1985).

The physical dimensions and properties of the shell are listed in Table 6.3. The shell structure was uniform in all directions, and did not contain a floor. The material properties of the shell are those of aluminum, and the shell dimensions correspond to the dimensions of the cylindrical shell used in the experiment to be described later. The complete analytical model of the shell and cavity is discussed in detail in Appendix A. The responses of the shell and the interior acoustic space were coupled by a momentum boundary condition at the interior shell wall. The enclosed fluid was assumed to be air at standard conditions. Because the enclosed fluid is light, the response of the shell can be represented by its *in vacuo* mode shapes, and the response of the interior acoustic space by its rigid-walled mode shapes (Fuller and Fahy 1982). Losses in the structure and the interior were modeled by introducing small imaginary components in the sound speeds for the two media. These loss factors, listed in Table 6.3, were 0.10 in the shell and 0.05 in the fluid.

A schematic of the control system is shown in Fig. 6.13. Eight PZT pairs were positioned in two axial planes on the cylinder, one at $L/3$ and the other at $2L/3$, where L is the overall length of the cylinder. At each axial location, the actuators were placed at $\pm 20^\circ$ and $\pm 60^\circ$, relative to the top centerline of the cylinder, as shown in the cross section on the right side of Fig. 6.13. The interior acoustic pressure was computed at forty-eight locations. These locations will be referred to as microphone locations for simplicity, even though the model only calculated the pressure at a point, and did not consider the dynamic response of a pressure transducer. The microphones were arranged in six axial planes, uniformly spaced down the length of the cylinder at a spacing of $(L/7)$. In each axial plane, eight microphones were evenly arranged along a semicircular arc at a radius of $0.9r$, where r is the radius

Table 6.3: Properties of cylindrical shell and enclosed fluid

property	value
radius	0.84 <i>m</i>
thickness	0.0017 <i>m</i>
length	3.66 <i>m</i>
shell density	2700 <i>kg/m</i> ³
structural loss factor	0.10
fluid loss factor	0.05
Young's modulus	7.1×10^{10}
Poisson's ratio	0.33
Fluid density	1.21 <i>kg/m</i> ³
Sound speed, fluid	343 <i>m/s</i>
# circumferential modes in model	15
# axial modes in model	15

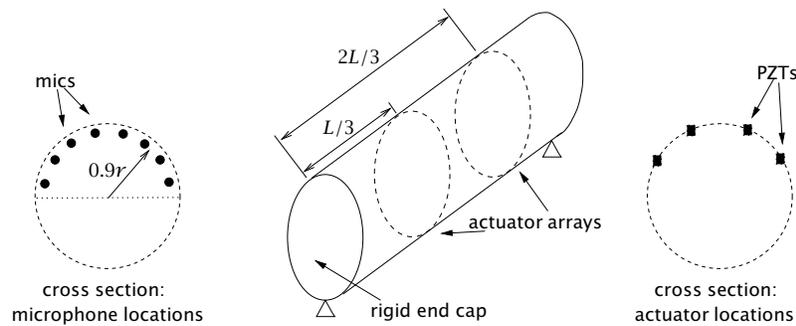


Figure 6.13: Control system on cylinder

of the shell, as shown in the cross section in the left side of Fig. 6.13.

The singular values of the transfer function matrix between PZTs and interior pressures are plotted as a function of frequency in Fig. 6.14. At any frequency, there are eight singular values, since the rank of the transfer function matrix at a single frequency was eight. The values are plotted in *dB*, relative to the maximum singular value near 150 Hz. The curves in the plot are considerably more complicated than the curves for the two plate models discussed earlier. This is due to the fact that the control system contains more sensors and actuators than for the plate models, and due to the complexity of the coupled shell-cavity system. Nonetheless, the singular values still appear to vary continuously with frequency, as was observed with the plate models. The singular values peak at four frequencies in the plot. By comparison with the natural frequencies of the interior cavity, which are listed in Table 6.4 (Blevins 1984), the resonances correspond to modes of increasing axial order as labeled in the plot. The specific modal indices for each of the resonances in the singular value plot are labeled according to the circumferential, axial and radial mode, such that the (0, 2, 0) mode is the second order axial mode, and zeroth order in the other directions.

The singular values in the plot appear to fall into one of two general categories: those associated with the even axial modes and those associated with the odd axial modes. For example, the solid curve that corresponds to the maximum singular value at 94 Hz does not increase at the resonance of the (0, 3, 0) mode at 141 Hz, but increases at the resonance of the (0, 4, 0) mode at 188 Hz. Likewise, the maximum singular value at 141 Hz does not appear to be associated with the resonance of the (0, 4, 0) mode at 188 Hz, but does increase at the resonance of the (0, 5, 0) mode at 234 Hz. Even

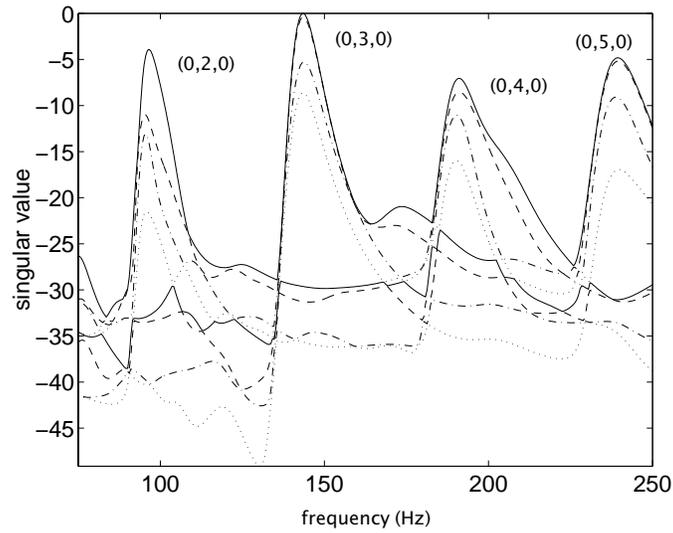


Figure 6.14: Singular values vs. frequency for control system on cylindrical shell

Table 6.4: Natural frequencies (Hz) of cavity; radial mode order = 0

Axial order	Circumferential order			
	0	1	2	3
0	0	120	199	274
1	47	129	204	278
2	94	152	220	289
3	141	185	244	308
4	188	223	273	332
5	234	263	307	360
6	281	306	345	392

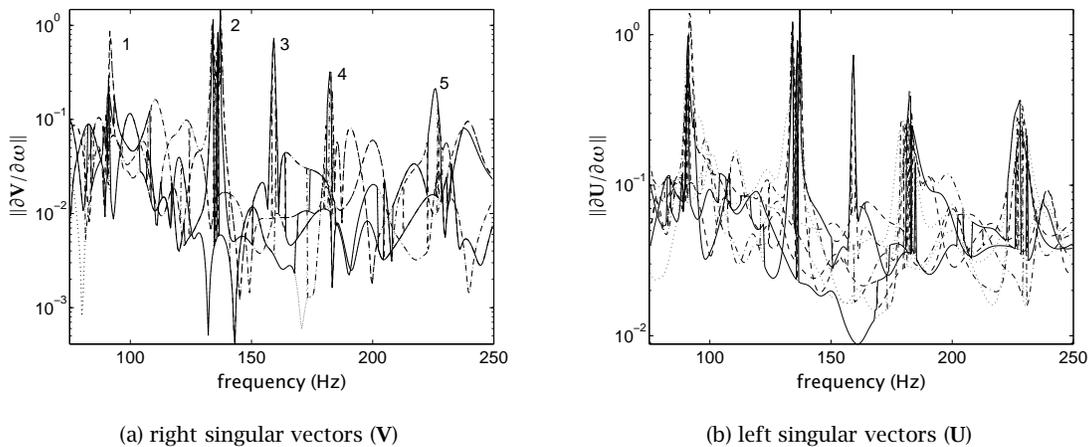


Figure 6.15: Derivatives of singular vectors for cylindrical shell

for this complex dynamic system of shell and cavity, the variation of certain singular values can be associated with responses of specific modes or families of modes in the system. The importance of this observation is that the SVD of a transfer function matrix computed at a single frequency can be related to the dynamics of the underlying system, and thus the singular values and singular vectors change in a deterministic manner as a function of frequency.

The derivatives of the left and right singular vectors for this example are shown in Fig. 6.15. The norms of the derivatives are shown. The curves in the plots are quite complicated, and we will make no attempt to analyze the behavior of any one curve, but instead will comment only on the five peaks that appear in both plots, and are labeled in Fig. 6.15(a). The peaks occur at approximately 91, 137, 159, 183, and 226 Hz. The frequencies of peaks 1, 2, 4, and 5 occur just below the natural frequencies of the axial modes that are labeled in the singular value plot. These natural frequencies are 94, 141, 187, and 234 Hz. This close correspondence between the derivative peaks and the natural frequencies of the acoustic modes suggests the high derivatives are due to a shape change in the singular vectors that occurs near the resonances. Similar derivative peaks were seen for the two plate examples whenever the responses of two modes were nearly equal, in which case the singular vectors changed shape rapidly as a function of frequency.

Peak number three from Fig. 6.15(a) does not appear to be associated with an acoustic response, since there are no acoustic resonances near 175 Hz. However, the natural frequency of the (3, 2) structural mode (third order in circumferential direction, second order axially) occurs at 173.4 Hz, and there is a slight increase in the singular values in Fig. 6.14 at this frequency. Thus peak 3 in the derivative plot in Fig. 6.15(a) is likely due to this structural mode. These peaks indicate that for a structure that is modally dense, the shapes of the singular vectors can change rapidly as a function of frequency.

While earlier results demonstrated that shape changes in singular vectors do not necessarily cause convergence to be unstable, the changes mean the control system is no longer decoupled. This loss of decoupling may be important for a large control system on a complex structure, where the decoupling is used to limit the total control effort due to ill-conditioned principal components of the control system. For the cylindrical shell model discussed here, small changes in frequency appear to produce large changes in shapes of the singular vectors. In this case, an online procedure to identify the

transfer function matrix and compute the singular vectors may be useful.

Chapter 7

Online System Identification

Previous chapters have demonstrated how a feedforward control system can be decoupled at a single frequency using the singular value decomposition (SVD) of the transfer function matrix between actuators and sensors at the frequency of interest. The robustness of the resulting controller (called PC-LMS) to errors in the transfer function matrix estimate was found to be identical to that of the filtered-x LMS algorithm. The analytical results in Chapter 6 demonstrated the significant but smooth variation of the singular vectors and values of the transfer function matrix with frequency for a typical control applications. The current chapter describes an online procedure to identify the transfer function matrix between sensors and actuators as the dynamics of the system being controlled change or the reference frequency changes. The online identification method described here does not represent an advancement in the state of the art of system identification, but is described and demonstrated for the sake of completeness of the thesis.

An online identification procedure should satisfy the following requirements to be useful for the PC-LMS control algorithm:

- the transfer function estimate should be updated fast enough to track slowly variations in the plant, including changes to the frequency of the reference signal. The speed of the identification procedure determines how much change in the transfer function matrix can be tolerated before the controller goes unstable
- the method should be applicable to a multichannel controller with a high number of actuators and error sensors
- the estimated transfer function matrix should be unique at each frequency, so variation from frequency to frequency is due to the dynamics of the system and not due to the initialization state or convergence parameters of the identification algorithm

Online identification of the transfer function matrix has been described previously in the context of a feedforward control system. Methods for online identification are generally categorized as being either intrusive, requiring the injection of a random signal to the actuator inputs, or non-intrusive, relying only on the reference signal to excite the system. The intrusive methods can identify the transfer functions over any desired frequency range, as long as the injected signal excites the system at these frequencies. However, the injected random signal creates a noise floor in the error sensor responses that limits the performance of the control system. The non-intrusive methods do not suffer

from this limitation, but because they rely on the reference input to perform the identification, the resulting transfer functions are accurate only at the frequencies in the reference signal.

One non-intrusive method is the overall modeling algorithm (Sommerfeldt and Tichy 1990; Kuo and Morgan 1996). This algorithm simultaneously models the transfer function between control actuators and error sensors (the secondary path), and the transfer function between the reference input and the primary excitation (the primary path). This has two drawbacks that make it unsuitable for the PC-LMS controller. The first is that the transfer function estimates must be updated at every sample instant, which can be computationally intensive if the controller contains a high number of actuators and sensors. The second drawback is that the secondary path identification is not unique, because the identified transfer function contains the effects of both the primary and secondary path (Kuo and Morgan 1996). This introduces an unneeded source of variability into the singular vectors of the transfer function matrix.

Thus it appears that Intrusive methods are better suited for online identification for PC-LMS control, although they require a balance between minimizing the additive signal level while maximizing the accuracy of the system identification (Kuo and Morgan 1996). An early reference to online identification in the context of feedforward controller is given by Eriksson and Allie (1989). The authors described a recursive least squares algorithm for identification of the transfer function between an actuator and sensor for a single input/single output system. Unfortunately, the computational requirements of a recursive least squares type of algorithm would be too large for a control system with many control channels and error sensors. The LMS algorithm has also been used to adapt the coefficients of an FIR model of the transfer functions in a multichannel controller (C. Bao 1993a; C. Bao 1993b). However, the presence of the primary noise field greatly limits the adaptation rate of the algorithm (Kuo and Morgan 1996; C. Bao 1993a), hence the method is best-suited for very slowly varying systems. Convergence of the algorithm can be accelerated by using an additional LMS algorithm to cancel the effects of the primary noise field from the inputs to the identification algorithm (C. Bao 1993a).

A different intrusive approach based on methods for modeling multiple-input/output systems Bendat and Piersol (1986) will be described here. This method assumes requires computation of the cross-spectral and auto-spectral densities of the input and output time series from each actuator to each error sensor. The associated computational burden is small, hence the approach is useful for large control systems. The theory is described here, followed by a computer simulation of the approach as part of a multichannel PC-LMS controller on a simply supported vibrating plate.

7.1 System Identification

A formal definition of system identification in the context of feedforward control is described for the single input/single output system depicted in Fig. 7.1. An input time series is represented by $x(t)$, an output time series by $y(t)$, and the transfer function between the two by $H(f)$, where f signifies the frequency dependence of the transfer function. For an active control system, $H(f)$ can represent a linear combination of transfer functions from actuator input to error sensor output, including: digital to analog converters, conditioning filters, amplifiers, the dynamics of the system being controlled, and analog to digital converters.

For the system in Fig. 7.1, the frequency response function is written (Bendat and Piersol 1986)

$$H(f) = \frac{G_{xy}(f)}{G_{xx}(f)} \quad (7.1)$$

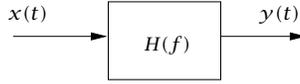


Figure 7.1: SISO system

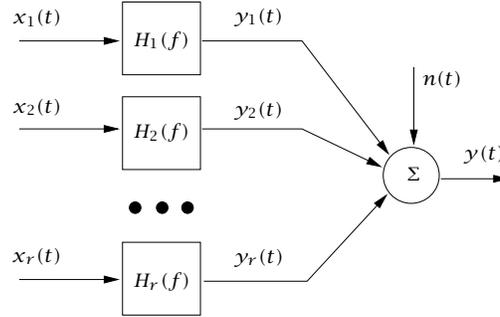


Figure 7.2: MISO system

where $G_{xx}(f)$ is the one-sided auto-spectral density of the input $x(t)$, and $G_{xy}(f)$ is the cross-spectral density between the input and the output, $y(t)$. The auto-spectral density is computed from N values of the input $x(t)$ as (Bendat and Piersol 1986)

$$\hat{G}_{xx}(f_k) = \frac{2}{N\Delta t} |X(f_k)|^2 \quad k = 0, 1, \dots, N/2 \quad (7.2)$$

where Δt is the sampling interval, and the hat symbol, $\hat{(\)}$, signifies that the computed value is only an estimate of the true spectral density of $x(t)$ that may contain errors due to random noise. The term $X(f_k)$ is the value of the finite Fourier transform of $x(t)$ at the frequency f_k , and is computed as

$$X(f_k) = \Delta t \sum_{n=0}^{N-1} x(n) e^{-j2\pi i f_k \Delta t} \quad (7.3)$$

This expression for the Fourier transform assumes a boxcar window is applied to the time series $x(t)$, which is suitable for time series containing tones, but other window functions (e.g., Hanning) may be more suitable other types of time series. The cross-spectral density, $\hat{G}_{xy}(f)$, is defined similarly as

$$\hat{G}_{xy}(f_k) = \frac{2}{N\Delta t} [X^*(f_k)Y(f_k)] \quad k = 0, 1, \dots, N/2 \quad (7.4)$$

where $X^*(f)$ is the complex conjugate of $X(f)$, and $Y(f)$ is the finite Fourier transform of $y(t)$. Assuming the time series $x(t)$ and $y(t)$ are stationary, and the transfer function $H(f)$ is constant with time, variability due to random noise can be reduced by averaging several spectral density estimates from Eqs. 7.2 and 7.4. If a set of frequencies f_1, \dots, f_k are known *a priori*, values of $e^{-j2\pi i f_k \Delta t}$ can be precomputed, and thus computation of each finite Fourier transform will require a single complex multiply and accumulate operation for each new sample of $x(t)$ and $y(t)$.

Computation of the transfer function can be generalized to the multiple input/single output case, pictured in Fig. 7.2 (Bendat and Piersol 1986). The symbols $(x_1(t), x_2(t), \dots, x_r(t))$ denote inputs to the r control actuators. Alternatively, if the reference signal is periodic and synchronously sampled, it could be considered as an input, and the remaining $(r - 1)$ inputs in the figure would then denote inputs to the control actuators. The output represents the response of a single error sensor, and the term $n(t)$ represents random noise present in the sensor output. Following the description in Bendat and Piersol (1986, Chapter 7), the sensor output is written

$$y(t) = \sum_{i=1}^r y_i(t) + n(t) \quad (7.5)$$

Computing the Fourier transform of each term produces

$$Y(f) = \sum_{i=1}^r Y_i(f) + N(f) \quad (7.6)$$

$$= \sum_{i=1}^r H_i(f)X_i(f) + N(f) \quad (7.7)$$

Multiplying both sides by $X_j^*(f)$, and taking the expected value yields

$$E[X_j^*(f)Y(f)] = E\left[\sum_{i=1}^r H_i(f)X_j^*(f)X_i(f)\right] + E[X_j^*(f)N(f)] \quad (7.8)$$

$$G_{jy}(f) = \sum_{i=1}^r H_i(f)G_{ji}(f) + G_{jn}(f) \quad (7.9)$$

substituting the definitions of the one-sided spectral densities. Assuming the noise $n(t)$ is random, the $G_{jn}(f)$ terms are zero, so the expression simplifies to

$$G_{jy}(f) = \sum_{i=1}^r H_i(f)G_{ji}(f) \quad (7.10)$$

This represents a set of r equations in r unknowns (the transfer functions, $H_i(f)$, are unknown). The G_{ji} terms represent cross spectral densities between pairs of input signals, $x_i(t)$ and $x_j(t)$, while G_{jy} represents the cross spectral density between $y(t)$ and the j th input, $x_j(t)$. This equation can be used to solve for the $H_i(f)$ terms as long as the input and output signals satisfy the following restrictions (Bendat and Piersol 1986):

- none of the ordinary coherence functions between any of the input records equals 1
- none of the ordinary coherence functions between any input and the output y equals 1
- the multiple coherence between any of the input records does not equal 1
- the multiple coherence between any input and the output should be sufficiently high (a value of 0.5 is recommended)

For a single frequency feedforward control system, these restrictions are satisfied if a random signal is added to the input to each actuator. If each input signal is uncorrelated with every other signal, the G_{ji} terms in Eq. 7.10 are zero for $i \neq j$.

7.2 Computer Simulation

A computer simulation was used to study the interaction between the online identification procedure and the PC-LMS control algorithm for a simple dynamic system. A time-domain simulation was created, and the stability of the PC-LMS algorithm evaluated both with and without online identification of the transfer functions from control actuators to error sensors. A single frequency reference signal provided the primary disturbance. The frequency of the disturbance was changed in discrete steps while the controller was operating in order to evaluate the accuracy of the online identification.

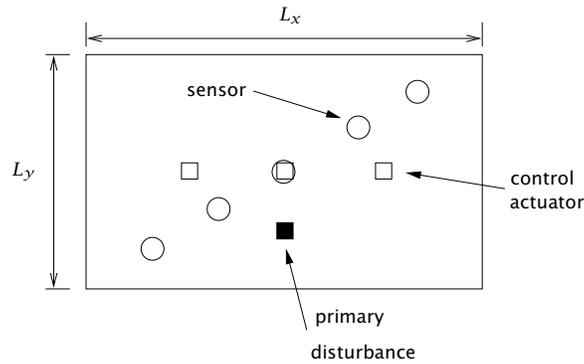


Figure 7.3: Actuator and sensor placement on simply supported plate

A feedforward control system for controlling vibrations of a simply supported plate was modeled. The plate was identical to the one discussed in Chapter 6, and its material properties are listed in Table 6.1. The control actuator and error sensor arrangement was also identical to that used in Chapter 6, as shown in the schematic in Fig. 7.3. Control was applied with three point-force actuators that were evenly distributed along the x -axis of the plate and at a constant y location of $L_y/2$. Five accelerometers were distributed along the diagonal of the plate, as indicated in the figure. An additional point-force input was located at $(L_x/2, L_y/4)$ and was used to produce the primary disturbance. The control actuators were used to minimize the mean square responses of the five sensors due to the input from the primary disturbance actuator.

The frequency of the primary excitation was initially set to 95 Hz, and was incremented in 2 Hz steps at regular intervals while the controller was running. From the list of the natural frequencies of the plate in Table 6.2, the natural frequency of the (1, 1) plate mode occurs at 99 Hz. The changes to the reference frequency thus tested the stability of the controller with online identification as the excitation frequency passed through a plate resonance. From the results in Chapter 6, the principal component corresponding to the resonant mode goes through a 180° phase shift at resonance, which will cause convergence of the controller to go unstable if unaccounted for. The reference signal was sampled at a fixed rate of 10 times per cycle for this simulation, therefore each change in the reference frequency corresponded to a change in the sample rate of the controller. This behavior can be achieved in an actual control application if the sampling rate is determined by a phase-lock loop fixed on the reference signal.

A realistic simulation of the dynamics of the controlled plate was achieved by using a state space model of the control system, including the primary excitation. A state space model can be used to represent the dynamic response of a system via a set of state vectors and state transition matrices (e.g., see Juang (1994), or Hale (1973)). A discrete-time state space model is written

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (7.11)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \quad (7.12)$$

where k represents discrete time values. The $(n \times 1)$ vector $\mathbf{x}(k+1)$ is the state vector at time $(k+1)$, where n denotes the number of states in the model; $\mathbf{u}(t)$ is an $(r \times 1)$ vector of control inputs; and $\mathbf{y}(t)$ is an $(m \times 1)$ vector of sensor responses. The matrices \mathbf{A} and \mathbf{B} relate the state at time $(t+1)$ to the previous state, $\mathbf{x}(t)$, and the current control inputs, $\mathbf{u}(t)$. The matrices \mathbf{C} and \mathbf{D} relate the sensor responses to the current state and control input.

For the simulation described here, the matrices \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} , were generated from transfer

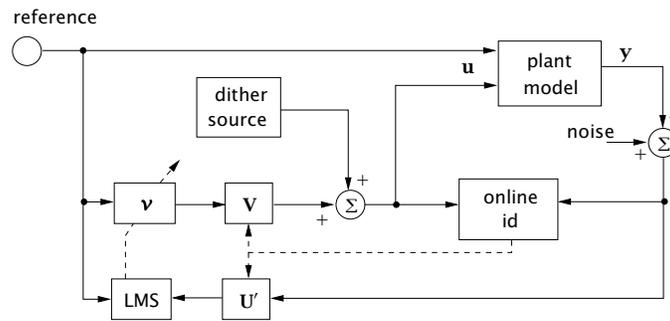


Figure 7.4: PC-LMS control with online system identification

function data between the four point force actuators and the five error sensors, computed at discrete frequencies. The transfer functions were created using the numerical plate model described in Appendix A, over a frequency range of 60 to 400 Hz, in 1 Hz increments. The conversion from frequency domain transfer functions to the discrete-time state space matrices was accomplished using a frequency domain to state space conversion tool (Horta et al. 1993). A 20 state model accurately modeled the dynamics of the plate from 60 to 400 Hz. Note that the control input vector $\mathbf{u}(t)$ contains inputs to all four actuators on the plate.

In order to simulate changes in the excitation frequency, the state space matrices, \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} , were generated for six different sample rates. As mentioned earlier, the reference input signal was sampled at a fixed rate of 10 samples per cycle, hence a change in the excitation frequency corresponded to a change in the sample rate of the control system. The state space matrices depend on the sample rate, so they were generated at the six sample rates that correspond to six excitation frequencies of 95, 97, 99, 101, 103, and 105 Hz. Therefore, changes in the excitation frequency were simulated by abruptly changing the state space matrices at fixed intervals during the simulation. It is unlikely that an actual system being controlled would undergo such abrupt changes, but nonetheless the technique was useful for evaluating the online identification procedure.

A block diagram of the simulated control system is shown in Fig. 7.4, including the PC-LMS algorithm, the online identification, and the state space model. The single frequency reference signal is indicated in the top left corner of the plot. This signal is one element of the control input vector $\mathbf{u}(t)$ in the state space model, as indicated in the figure. The reference signal is also an input to the PC control system. The control output, in PC coordinates, is computed by multiplying the reference input signal by the PC control filter weights, \mathbf{v} . These PC outputs are then transformed to actuator inputs by the matrix of right singular vectors, \mathbf{V} . A random dither signal is added to the actuator inputs, and these inputs make up the rest of the state space control input vector, $\mathbf{u}(t)$. The control inputs are then propagated through the state space plant model to determine the sensor responses, $\mathbf{y}(t) + \text{noise}$, and to compute the updated state vector, $\mathbf{x}(t + 1)$ (the state vector was initialized to the zero vector). The sensor responses thus contain the combined effect of the primary and control inputs to the plate. Random noise can be added to the sensor outputs, as indicated by the additive noise signal in the figure. Continuing along the outside of the schematic, the sensor responses are then transformed into PC coordinates by the matrix \mathbf{U}' , and used by the PC-LMS algorithm to update the control weights, \mathbf{v} .

A random signal, or dither signal, was added to the actuator inputs and used by the online identification procedure to estimate the transfer function matrix. This dither signal can have any spectral shape that satisfies the four requirements listed on page 90. The transfer function matrix was computed at the single frequency corresponding to the reference signal. The finite Fourier transforms from each actuator input to each sensor output were computed, from which the cross- and auto-spectral

Table 7.1: Simulation parameters

Parameter	Value
SNR	60 dB
adaptation rate	10
frequency resolution	2 Hz
# periods for initial id	750
# periods to change s.s. model	4650
leaky averaging factor	0.99
dither amplitude	0.001

densities were computed as in Eq. 7.10. The spectral densities and transfer function estimate were updated every N points, where N was the length of the Fourier transform in Eq. 7.3.

A leaky average of the spectral densities in Eq. 7.10 was used to average out random noise while retaining the ability to adapt to changes in the system. With each new Fourier transform estimate, the spectral densities were update according to

$$G_{ij}(f_k) = \kappa * G_{ij}(f_k) + (1 - \kappa) * (X^*(f_k) * X(f_k)) \quad (7.13)$$

where κ is close to but less than unity. This equation averages out random variations in the spectral density estimate. Each time a new transfer function estimate was computed, its SVD was also computed and the singular vectors, U and V , in the PC-LMS controller were updated.

7.3 Simulation Results

Simulation results of the dynamical system with and without online identification are compared here. The MATLAB code used for this simulation is listed in Appendix B. The PC-LMS algorithm was used to control only the first principal component of the transfer function matrix, since only this component was significantly excited in the frequency range that was studied. The singular values of this control system, shown in Fig. 6.2(a), indicated the transfer function matrix was dominated by the first singular value for a range of frequencies around 99 Hz, and this singular value corresponds closely to the response of the (1, 1) plate mode.

Simulation parameters are listed in Table 7.1. The SNR is the signal to noise ratio of the random noise in the sensor responses, relative to the sensor responses due to the primary excitation alone. Because the purpose of the present simulation was to establish the utility of online identification, the SNR was set to an extremely low value of 60 dB to remove the effect of random noise in the sensors. The adaptation rate was used in the weight update equations for the PC-LMS control filter weights. The frequency resolution refers to the Fourier transforms, $X_i(f_k)$ and $Y_j(f_k)$, computed using Eq. 7.3. The number of periods for the initial identification describes how long the spectral density functions were initially estimated before control was turned on. This delay insured the initial transfer function estimate was accurate before it was used by the controller. Once control was turned on, the sample rate was changed to a new value every 4650 periods of the reference signal. This interval was chosen because it provided enough time for the controller and online identification procedure to converge when the sample rate was changed. The leaky averaging factor corresponds to the value of κ in Eq. 7.13, and the dither amplitude gives the maximum amplitude of the random dither signal, relative to the unit amplitude reference signal.

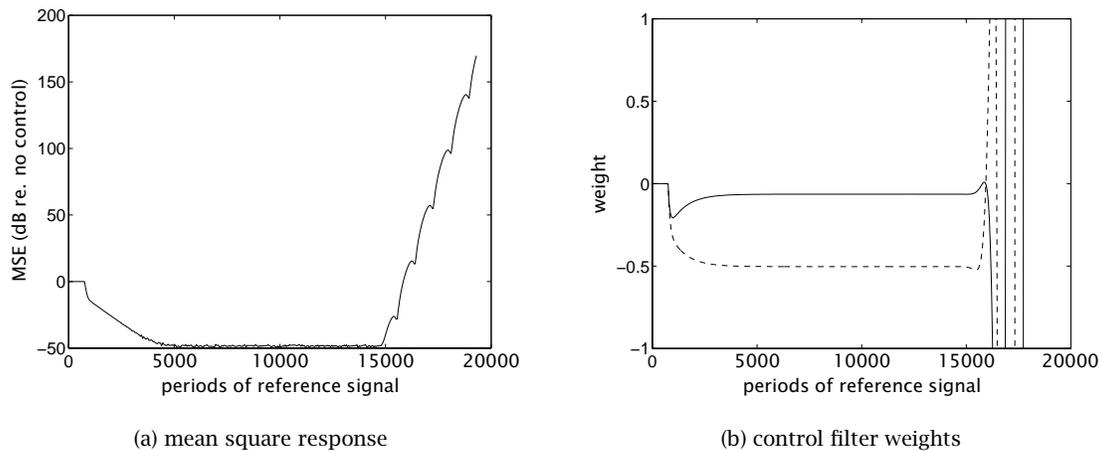


Figure 7.5: Simulation results without online identification

For each simulation, startup transients due to the dynamics of the state space model were allowed to decay until the system reached steady state conditions under primary excitation alone. The random dither signal was then turned on and the leaky averaging procedure started to accumulate averages of the spectral densities for the transfer function estimate. After the delay for the initial identification, the PC-LMS algorithm was used to adapt the control filter weights (for the first PC only, as discussed above). For the first simulation, identification of the transfer function stopped when control was turned on. For the second simulation, identification of the transfer function continued online.

7.3.1 Results without online id

The sum of the mean square responses of the five sensors as a function of periods of the reference input signal are shown in Fig. 7.5. The values are plotted in decibel units, relative to the total mean square responses due to the primary excitation alone without control. Following the 750-period startup delay, the controller quickly reduced the mean square sensor responses by nearly 50 dB. Changes in the sample rate at 5400 and 10500 periods, from 950 to 970 Hz, and then 970 to 990 Hz, had no effect on the controller. However, when the sample rate changed from 990 to 1010 Hz at 14700 periods, the controller went unstable. This sample rate change corresponded to a change in the primary excitation frequency from 99 to 101 Hz, thus crossing through the natural frequency of the (1, 1) plate mode at 99 Hz. At 101 Hz, the transfer function originally computed at 95 Hz contained errors due to the 180° phase change at the resonance of the first plate mode.

The two PC control filter weights, shown in Fig. 7.5(b), behaved in a similar fashion. The weights were stable after an initial period of convergence, and remained stable while the sample rate changed from 950 to 990 Hz. However, when the sample rate changed from 990 to 1010 Hz, the controller quickly went unstable and the weights began very large amplitude oscillations as a result.

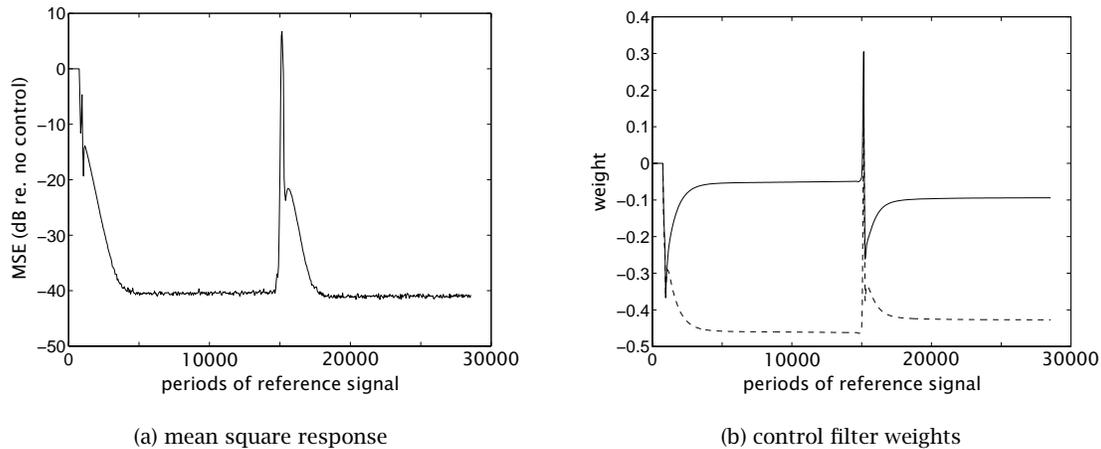


Figure 7.6: Simulation results with online identification

7.3.2 Results with online identification

For the second simulation the transfer function estimate was continually updated while the controller was running. The resulting error reduction and control filter weights are shown in Fig. 7.6. After the initial 750 period startup delay, the mean square responses of the error sensors were quickly reduced, as shown in Fig. 7.6(a). A careful examination of the curve in Fig. 7.6(a) reveals a slight discontinuity just after the start of convergence. During this initial convergence, the control inputs changed significantly over the time required to compute a single Fourier transform estimate, and as a result the estimate was inaccurate. This inaccuracy caused some instability in the adaptation of the control filter weights. The instability illustrates how the adaptive processes of control and system identification can interact with one another and possibly have a significant effect on the stability of the controller (Feintuch et al. 1993), especially when the controller is adapting quickly. A lower adaptation rate would reduce the inaccuracy in the Fourier transform, and hence reduce the likelihood of instability during convergence of the controller.

As in the previous simulation, the performance of the controller was not affected when the sample rate changed from 950 to 970 Hz, or from 970 to 990 Hz. When the sample rate changed from 990 to 1010 Hz at 14700 periods, the controller appeared to go unstable momentarily. However, the online identification procedure accurately updated the transfer function matrix before the controller went completely unstable, allowing the controller to reconverge and reduce the sensor responses once again.

The control filter weights for this case are shown in Fig. 7.6(b). The weights converged quickly and remained stable until the frequency changed to 101 Hz at 14700 cycles. The weights diverged slightly at this point, but reconverged quickly. A lower adaptation rate would be beneficial here because it would reduce the magnitude by which the weights diverge from their optimum values, while the online identification procedure is updating the transfer function matrix.

This simulation demonstrates the necessity of lowering the adaptation rate of the control algorithm when an online identification procedure is used. A lower adaptation rate will reduce errors in the finite Fourier transform of the control inputs when the control filter weights are convergence very quickly.

A smaller adaptation rate will also help to reduce the rate at which the control filter weights diverge from their optimum values, when the transfer function matrix is momentarily inaccurate.

A comparison of the simulation results for the cases with and without online identification demonstrates the benefits of identifying the transfer function matrix while the controller is running. Online identification allows the controller to adapt to changes in the dynamics of the system being controlled, which in this case were due to a change in the reference frequency. There are several important issues to consider before adding online identification to a feedforward controller, including the required level of dither signal to accurately identify the transfer function matrix in a reasonable amount of time and the interaction between the dynamics of the adaptive control algorithm and the online identification procedure.

Chapter 8

Experimental Results

The theory of the principal component LMS (PC-LMS) algorithm was described in previous chapters; this chapter describes an experimental investigation of the algorithm. The experiments were conducted on an elastic cylindrical shell with rigid end caps enclosing a cylindrical cavity. The PC-LMS and filtered-x algorithms were used to compute control inputs to reduce sound levels in the cavity produced by two loudspeakers outside the cylinder. This cylindrical shell has been used in the past to evaluate control strategies for aircraft interior noise reduction (Silcox et al. 1989; Palumbo et al. 1996). The control system for the experiment consisted of 12 force actuators mounted on the shell and 48 microphones inside the cavity, and the control algorithms were implemented in real time on a digital signal processor (DSP). The computational requirements, convergence behavior, and minimization of the mean square responses of the microphones for the filtered-x and PC-LMS control algorithms were compared at two frequencies.

The purpose of this experiment was not to demonstrate global reduction of sound inside the cylinder using structural actuators, since that has been demonstrated elsewhere (Fuller and Jones 1987; Silcox et al. 1987). Instead, the purpose was to study the PC-LMS algorithm and compare it to the filtered-x algorithm on a complex structure. The control system minimized the responses of 48 microphones, hence there is some assurance of global noise reduction in the cylinder. However, without a second set of sensors to measure the sound level away from the 48 microphones, global noise reduction could not be confirmed. For this reason, only the noise reduction measured at the 48 microphones, the convergence behavior of the controllers, and the magnitude of the control signals sent to the control actuators are discussed.

This chapter begins with a description of the cylinder test structure and the control system components, including actuators, microphones, and the control algorithms. The control system is then characterized by comparing the interior acoustic response due to loudspeaker excitation with the response produced by a single control actuator. The average magnitude of the transfer function from two actuators to the interior microphones is then related to the modal characteristics of the cylinder near the test frequencies. Noise reduction results at two test frequencies are then discussed. A control effort penalty was implemented with the filtered-x LMS and PC-LMS algorithms and the resulting noise reduction performance compared. The benefit of controlling a subset of the principal components of the controller is next demonstrated. A final test compared the noise reduction of the two control algorithms when the transfer function matrix was measured at one frequency and used to control a disturbance at several different frequencies.

The experimental results verified several observations from previous chapters concerning the PC-LMS algorithm. For example, the PC-LMS algorithm has lower computational requirements than the filtered-x LMS algorithm for controlling a single frequency disturbance. In addition, the PC-LMS algorithm provides a convenient technique to limit control effort by eliminating PCs from the controller. A simple robustness test demonstrated that the PC-LMS algorithm appeared to have the same robustness to modeling error as the filtered-x LMS algorithm.

8.1 Experimental Setup

The experiments were conducted in the test cylinder of NASA Langley's Acoustics and Dynamics Laboratory. This cylinder is a large cylindrical shell containing ribs, stringers, and an interior floor. The shell is closed at both ends, and thus approximately models an aircraft fuselage. The entire cylinder was located inside a room whose walls and floor were lined with sound absorbing foam wedges. Two loudspeakers were positioned on opposite sides of the cylinder exterior to simulate excitation of an aircraft fuselage by noise from wing-mounted propellers. The filtered-x LMS and PC-LMS control algorithms were implemented on a digital signal processor residing in a desktop computer. The mean square microphone responses and actuator outputs were periodically saved on the desktop computer while the controllers were running. The control system components and test facility are next described.

8.1.1 Acoustics and Dynamics Laboratory

A photo of the cylindrical shell is shown in Fig. 8.1. The shell was constructed from wound carbon fiber filaments embedded in an epoxy resin. The cylinder had an overall length of 3.66 m, a diameter of 1.68 m, and a wall thickness of 1.7 mm. The interior of the shell was strengthened with 10 J-section ring frames and 22 hat-section stringers, which were riveted and bonded to the shell. A 1.27 cm thick plywood floor was mounted 0.54 m above the bottom of the shell, and the joint between the floor and shell was sealed to acoustically isolate the spaces above and below the floor. The ends of the cylinder were closed with rigid baffles made from 3.18 cm thick particle board. The shell rested within grooves in the particle board to approximate simply supported end conditions. Access to the shell interior was provided by a hatch in one of the end caps.

The shell was supported on four legs attached to the rigid end caps, shown in Fig. 8.1, and the entire structure was located inside a large workshop area at NASA Langley. The shell was partitioned off from the rest of the workshop by four movable walls lined with sound absorbing foam wedges. The floor of the partitioned area was also lined with the foam wedges, but the ceiling was not partitioned off from the workshop area. The resulting facility resembled an anechoic chamber on its floor and walls but not on its ceiling. As a result, background noise levels were higher than would be present in an ideal anechoic environment.

Two loudspeakers positioned on either side outside the cylinder were used to create an external pressure disturbance on the shell, which excited an acoustic response inside the shell. This interior acoustic response was the primary disturbance field to be actively controlled. The loudspeakers were two 100-Watt electrodynamic speakers, positioned halfway along the length of the cylinder, as indicated in Fig. 8.2. The speakers were located at $\pm 90^\circ$ relative to the top centerline of the cylinder, and were approximately 0.5 m from the exterior surface of the shell. Although the speakers simulated excitation of a fuselage by external propeller noise, the actual pressure excitation due to propellers is more



Figure 8.1: Exterior view of cylindrical shell

complicated than could be reproduced with two loudspeakers. A more realistic acoustic source would have to simulate the trace velocity of the external acoustic field created by the propeller sweeping by the fuselage (Bullmore et al. 1987). In addition, the source would also have to model the rotational directions of the propellers on either side of the aircraft. To a passenger seated in the aircraft, the propeller on one side excites a response that proceeds from ceiling to floor, while the other propeller excites a response from floor to ceiling. This means one side of the plane is usually noisier than the other side (Dorling et al. 1989; Elliott et al. 1990). These discrepancies between the simulated and true propeller noise source imply the shell response due to the speakers was probably more symmetric than the response induced in an actual fuselage (Bullmore et al. 1987). Nonetheless, the test facility is useful for understanding the mechanisms of active control of enclosed sound fields on a reasonably complicated structure.

8.1.2 Control System Components

The control system consisted of 48 microphones as error sensors and 12 inertial force shakers as control actuators. The microphones were mounted on six metallic hoops rigidly attached to the plywood floor in the cylinder. Five of the metallic hoops are visible in the photo shown in Fig. 8.3 as bright arcs. The darker arcs in the photo are the ring frames of the cylinder. Foam damping material was installed between the cylinder ring frames for the photo in Fig. 8.3, but this material was not present during the experiments described here. Microphone holders were attached to the hoops with plastic tie-wraps, which are visible in the photo as short white lines extending radially from the hoops. The hoops were evenly spaced 0.52 m apart along the length of the cylinder, and each hoop supported eight microphones as shown in Fig. 8.4. Although this microphone arrangement cannot measure pressure variation in the radial direction, the hoops are located near antinodes of the first several radial acoustic modes (e.g., see Blevins (1984, Chapter 13)). Thus the microphone arrangement is practical

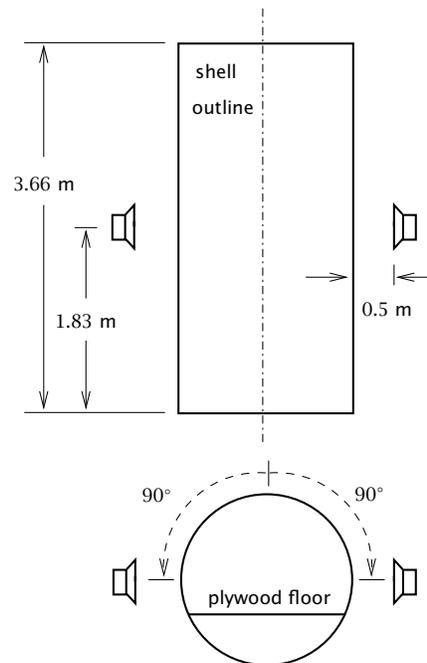


Figure 8.2: Speaker locations outside shell

for measuring interior sound levels at frequencies below the natural frequency of the first order radial cavity mode at 171 Hz (see Table 8.1, to be discussed shortly).

The error sensors were electret condenser microphones, manufactured by the Modal Shop, Inc. (model number 130A10), with attached pre-amplifiers (Modal Shop model number 130P10). The nominal sensitivity of each microphone was 20 mV/Pa, and the frequency sensitivity was ± 1 dB from 10 to 20000 Hz. During data collection and control, the output of each microphone was multiplied by an individually measured calibration factor to convert the response voltages to Pascals.

Twelve inertial force actuators provided control inputs to the cylindrical structure and were used to create an acoustic response inside the shell. This control approach is referred to as active structural acoustic control (Fuller and Jones 1987; Fuller 1987). The force actuators were manufactured by Motran Industries (model number IFX21C50); each had a nominal force constant of 22 N per Ampere and a natural frequency of approximately 60 Hz. The shakers were clamped on the ring frames of the cylinder, halfway between two stringers as pictured in Fig. 8.5.

The attachment locations for the control actuators were selected from a set of candidate locations in order to optimize noise reduction at one of the test frequencies. Because the actuators must excite structural motions that couple with the cavity response, the noise reduction performance depends on the actuator locations. With 10 ring frames and 13 mounting points per ring frame, there were 130 possible actuator locations. Transfer functions between the 130 candidate locations and the 48 microphones were collected at one of the test frequencies. A multi-variable optimization procedure described in Palumbo et al. (1996) was then used to select twelve locations that maximized noise reduction at the test frequency of 147 Hz. The selected locations are shown as solid black circles on an unwrapped view of the ring frames and stringers of the cylinder in Fig. 8.6. The approximate locations of the external loudspeakers are indicated by the X symbols in the figure. Most of the actuators were at approximately the same circumferential location as the external speakers.

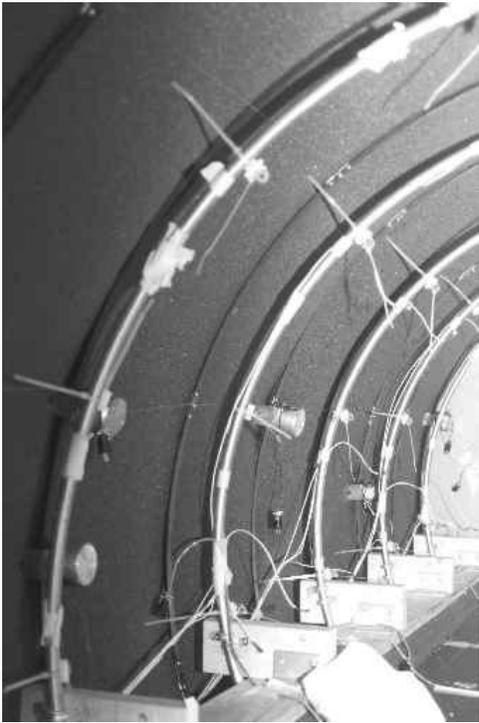


Figure 8.3: Photo of cylinder interior

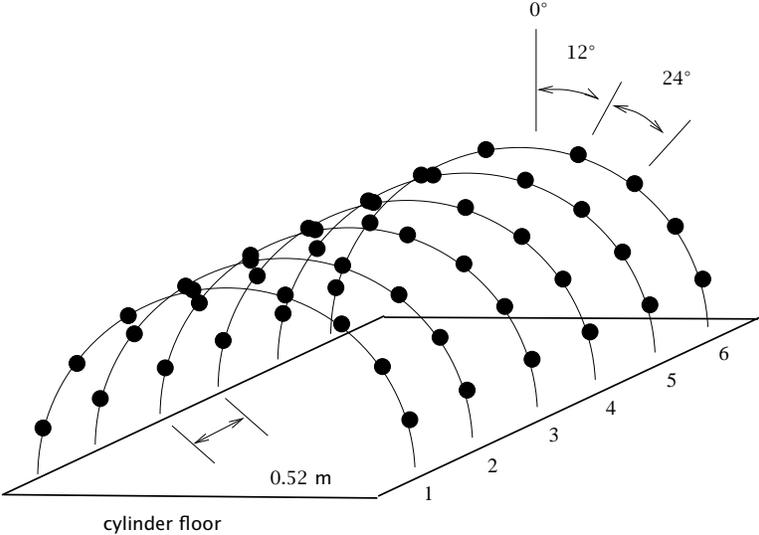


Figure 8.4: Microphone mounting locations

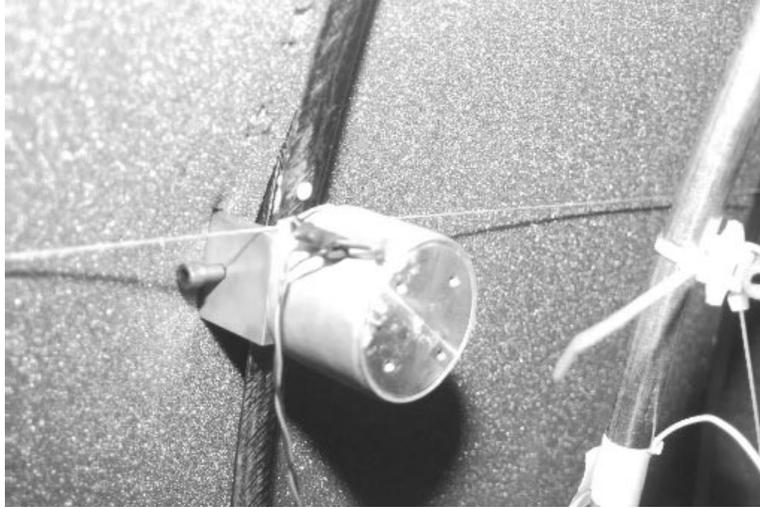


Figure 8.5: Actuator mounted on ring frame

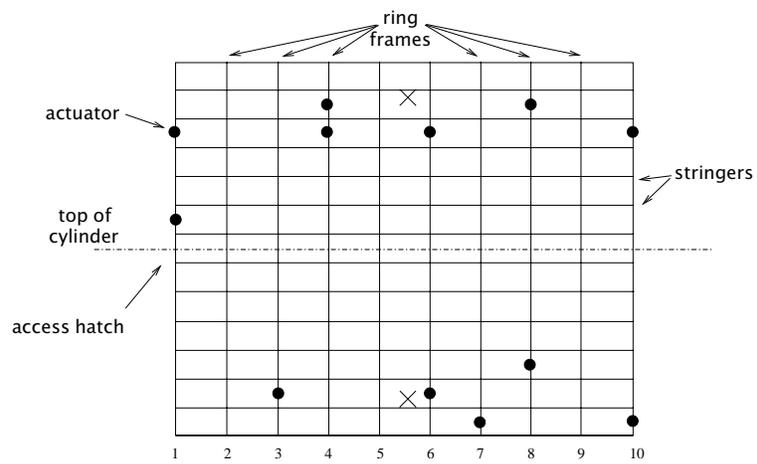


Figure 8.6: Actuator mounting locations

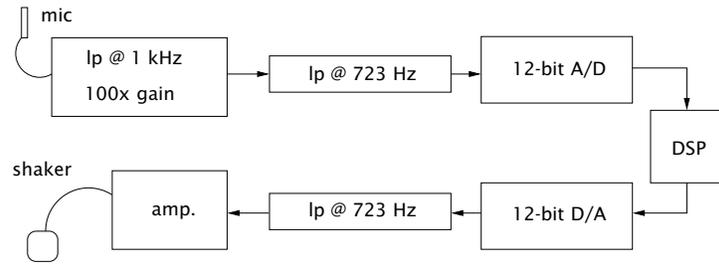


Figure 8.7: Signal conditioning

A schematic of the signal conditioning applied to the microphone responses and the control actuator inputs is shown in Fig. 8.7. The output of each microphone was low-pass filtered at 1000 Hz, a gain of 100 applied, and the signal was then low-pass filtered again at 723 Hz. The second low-pass filter was not needed but was present in the analog-to-digital converter. The microphone signal was then digitized by a 12-bit analog to digital (A/D) converter for processing by the control algorithm running on the DSP. The control inputs to the shakers were computed on the DSP and converted to analog signals via a 12-bit digital-to-analog (D/A) converter. The signal was low-pass filtered to remove stair-stepping due to the D/A conversion, and amplified before being sent to the shaker. The D/A converter had a range of ± 10 Volts. The control effort values discussed later in this chapter are expressed in terms of the voltages sent out the D/A converter.

8.1.3 Control Algorithms

The control algorithms and a system identification routine were written and executed on a Texas Instruments TMS320C40 digital signal processor (DSP) residing in a host personal computer. Details concerning the implementation of each of these algorithms are discussed here.

System Identification Algorithm

The system identification algorithm was used to compute an estimate of the transfer function matrix at a single frequency between the control actuators and the microphones. The definition of the transfer function from the j th control actuator to the i th microphone, assuming $x(t)$ denotes the input to the actuator and $y(t)$ denotes the corresponding microphone response, is written

$$H_{ij}(f) = \frac{G_{xy}(f)}{G_{xx}(f)} \quad (8.1)$$

The cross spectral density, $G_{xy}(f)$, and auto-spectral density, $G_{xx}(f)$, were measured at the frequency of interest, f , as defined in Eqs. 7.2 and 7.4. The transfer function values were measured one actuator at a time with the primary disturbance turned off, to insure good coherence between the actuator input and the microphone responses. A random signal was used as the actuator input, although a sinusoid at the frequency f would have worked since the estimate was computed at only one frequency. The spectral densities were computed with a 1 Hz spacing between adjacent frequency bins, and 15 estimates were averaged to reduce variation due to random sensor noise. The sample rate for the controller was set as an integer multiple of the reference excitation frequency. This simplified system identification by insuring the reference excitation frequency fell directly into a known frequency bin

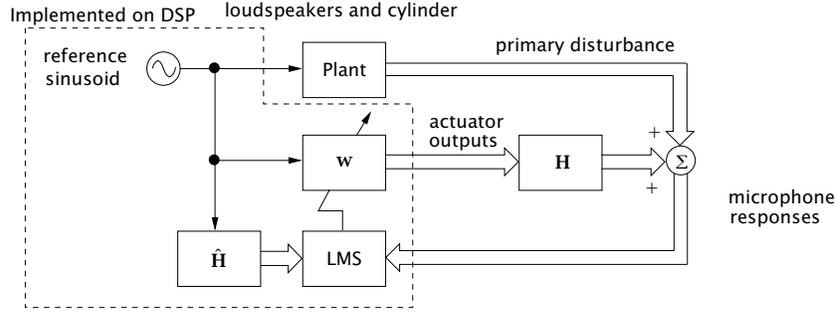


Figure 8.8: Filtered-x LMS Algorithm

of the Fourier transform. Although several different tests were conducted at each test frequency, the same transfer function estimate was used for each test. This reduced the possibility that one controller outperformed another one due to variations in the accuracy of the transfer function estimate.

Filtered-x LMS algorithm

A sampled data implementation of the filtered-x LMS algorithm for a single frequency excitation with a uniform control effort penalty was used for the tests. The algorithm was described previously in Section 3.3, and relevant equations are repeated here. A block diagram of the algorithm is shown in Fig. 8.8. The dotted line denotes the components of the algorithm that were implemented on the DSP. The reference signal was a single frequency sinusoid, hence only two control filter weights were needed for each actuator in order to produce an arbitrary amplitude and phase shift of the input sinusoid.

Using notation from Eqs. 3.25-3.32, the output of the i th control actuator was computed as

$$y_i(n) = w_{i0}(n)x(n) + w_{i1}(n)x(n-1) \quad (8.2)$$

where $x(n)$ is the reference input at time n . To update the control filter weights, (w_{i0}, w_{i1}) , for each actuator, the reference signal was pre-filtered by the transfer function estimate, Assuming the (i, j) th element of $\hat{\mathbf{H}}$ is represented by the FIR filter weights $(\hat{h}_{ij0}, \hat{h}_{ij1})$, the filtered reference was computed as

$$\hat{f}_{ij}(n) = \hat{h}_{ij0}x(n) + \hat{h}_{ij1}x(n-1) \quad (8.3)$$

If $e_i(n)$ denotes the response of the i th microphone at time n , the control filter weights corresponding to the j th actuator were updated according to:

$$w_{j0}(n+1) = (1 - \mu\beta)w_{j0}(n) - \mu \sum_{i=1}^m e_i(n) \hat{f}_{ij}(n) \quad (8.4)$$

$$w_{j1}(n+1) = (1 - \mu\beta)w_{j1}(n) - \mu \sum_{i=1}^m e_i(n) \hat{f}_{ij}(n-1) \quad (8.5)$$

assuming m microphones. Values of the step size parameter, μ , and the control effort penalty parameter, β , were varied depending on the particular control test. From Chapter 3, the maximum step size parameter for stable convergence is

$$\mu_{max} = \frac{2}{\lambda_{max}} \quad (8.6)$$

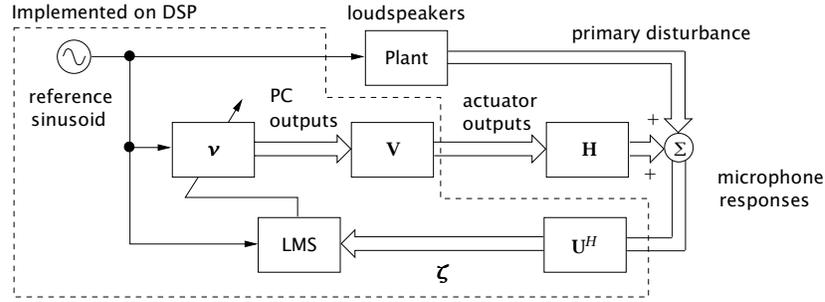


Figure 8.9: Principal Component LMS algorithm

where λ_{max} is the maximum eigenvalue of $\mathbf{H}^H\mathbf{H}$. The exact value of \mathbf{H} was unknown, so the step size was bounded using the eigenvalues of $(\hat{\mathbf{H}}^H\hat{\mathbf{H}})$ instead. Because there are usually estimation errors in $\hat{\mathbf{H}}$, a safe value for μ as suggested by Elliott et al. (1992) is one fourth of the maximum step size, i.e. ($\mu = \mu_{max}/4$).

Initial tests of the filtered-x LMS algorithm revealed that the DSP could not compute all of the values needed to update the control filter weights at the chosen sample rate. Therefore, to reduce the number of computations per sample iteration, the actuator weights were updated one actuator at a time, in a round-robin fashion. For example, the filtered reference values in Eq. 8.3 were computed for the first actuator, and were then used in Eqs. 8.4 and 8.5 to update the filter weights. The filtered reference values were then computed for the second actuator, and so on, for all twelve actuators. This slowed down the convergence of the algorithm since all of the control filter weights were not updated at every time step.

PC-LMS algorithm

A sampled data implementation of the PC-LMS algorithm, described in Section 4.2, was implemented on the DSP. The DSP was able to compute all values needed to update the weights for the PC-LMS algorithm within the specified sample interval, which verifies the conclusion from Chapter 4 that PC-LMS has a lower computation burden than filtered-x LMS. For the control system used here, the PC-LMS algorithm required only 2/3 as many computations as the filtered-x LMS algorithm.

The PC-LMS algorithm is shown in the schematic in Fig. 8.9 and is summarized in the following equations. Any subset of the r PCs (where r is the number of actuators) can be included in the PC-LMS algorithm, hence the symbol c denotes the number of PCs included in the controller, where $c \leq r$. The singular value decomposition of $\hat{\mathbf{H}}$ is denoted $(\mathbf{U}\mathbf{S}\mathbf{V}^H)$. Assuming the reference signal at time n is $x(n)$, the output of the i th PC is

$$y_i^{PC}(n) = v_{i0}(n)x(n) + v_{i1}(n)x(n-1) \quad (8.7)$$

where (v_{i0}, v_{i1}) are the control filter weights for the i th PC. The outputs of the c PCs are then transformed into outputs for the r control actuators by the transformation matrix \mathbf{V} . As described in Eq. 4.35, the transformation is computed as

$$y_i(n) = \sum_{j=1}^c (v_{ij0} y_j^{PC}(n) + v_{ij1} y_j^{PC}(n-1)) \quad (8.8)$$

If $e_i(n)$ denotes the sampled output of the i th microphone, the response of the j th PC is

$$\zeta_j(n) = \sum_{i=1}^m (u_{ij0}e_i(n) + u_{ij1}e_i(n-1)) \quad (8.9)$$

In Eqs. 8.8 and 8.9, the coefficients (v_{ij0}, v_{ij1}) , and (u_{ij0}, u_{ij1}) , denote time domain representations of the complex elements of the singular vectors of $\hat{\mathbf{H}}$, as described in Section 4.2. The PC responses, $\zeta_i(n)$, are used to update the PC control filter weights as

$$v_{i0}(n+1) = v_{i0}(n) - \alpha_i \zeta_i(n) x(n) \quad (8.10)$$

$$v_{i1}(n+1) = v_{i1}(n) - \alpha_i \zeta_i(n) x(n-1) \quad (8.11)$$

where α_i is the step size parameter for the i th PC.

For simplicity, the step size parameters were all assigned the same value in the experiment. Initially, this value was equal to the step size used in the filtered-x LMS algorithm, however the PC-LMS algorithm was unstable with this step size. This was probably due to the additional delay between the PC control filter output and PC error input, as discussed in Section 4.2. After some experimentation, a value four times smaller than the step size for the filtered-x algorithm was chosen, or $\alpha_i = \mu_{max}/16$. Note that in setting the step size parameters for all PCs to the same value, the overall convergence rate of the algorithm should be faster than steepest descent, but slower than Newton's algorithm.

Data storage during controller operation

The performance of the control algorithms was quantified in terms of noise reduction and control effort. The sample rate for the controllers was too fast to stream the time domain microphone responses and actuator inputs to the desktop computer, and the DSP did not have enough memory to save this data during a typical test. As a compromise, the mean square values of the microphone responses and actuator outputs were computed on the DSP using a recursive low-pass filter. The filter response was then periodically sampled by the host computer and saved. The filter response corresponding to the mean square value of an input signal, $x_i(n)$, was

$$y_i(n+1) = \kappa y_i(n) + (1 - \kappa)(x_i(n) * x_i(n)) \quad (8.12)$$

where the scalar κ was 0.995. This averaging filter was used to compute the mean square of the control outputs, $y_i(n)$, and the microphone responses, $e_i(n)$. In addition, for tests of the PC-LMS algorithm the mean square values of the PC control outputs, $y_i^{PC}(n)$ (from Eq. 8.7), and PC error responses, $\zeta_i(n)$ (from Eq. 8.9), were saved.

8.1.4 Parameters for Analyzing the Control System

One benefit of principal component control is the ability to express the control problem in terms of independent coordinates whose properties can be analyzed separately. Various parameters to quantify the utility of each principal component have been discussed in previous chapters and are summarized here. Computation of these parameters requires an estimate of the transfer function matrix between actuators and sensors, $\hat{\mathbf{H}}$. Although $\hat{\mathbf{H}}$ is measured from time domain data, it is simple to transform its elements to the frequency domain. The tools developed in previous chapters for analyzing a multichannel control system in the frequency domain can then be used on experimentally measured data.

The basic frequency domain model of the feedforward controller at a single frequency is written

$$\mathbf{e} = \mathbf{H}\mathbf{w} + \mathbf{d} \quad (8.13)$$

Using the notation from Chapter 4, this equation is expressed in terms of the principal components of the control system as

$$\boldsymbol{\zeta} = \mathbf{S}\mathbf{v} + \mathbf{p} \quad (8.14)$$

The transfer function matrix, \mathbf{H} , and primary response, \mathbf{d} , are measured during the system identification and can be used to predict the noise reduction and control effort at the test frequency. However, the primary response was measured as a transfer function between the reference signal generated on the DSP and the microphone responses, so the magnitude of \mathbf{d} must be scaled by the actual primary response level measured during the experiment in order to accurately estimate noise reduction and control effort. The scaling factor is given by

$$\text{scale} = \frac{|\hat{\mathbf{d}}|}{|\mathbf{d}|} \quad (8.15)$$

where $\hat{\mathbf{d}}$ is the vector of primary microphone responses expressed in the frequency domain. Applying this scaling factor to the vector \mathbf{d} yields the scaled quantity

$$\tilde{\mathbf{d}} = \text{scale} * \mathbf{d} \quad (8.16)$$

The optimum control weights can thus be computed offline as

$$\mathbf{w}_{opt} = -[\hat{\mathbf{H}}^H \hat{\mathbf{H}}]^{-1} \hat{\mathbf{H}}^H \tilde{\mathbf{d}} \quad (8.17)$$

These weights were only used to predict the performance of the control algorithms; during the experiments, the weights were initialized to zero and allowed to converge to their steady-state values. The microphone responses, \mathbf{e}_{opt} , corresponding to the optimum control weights are found by substituting \mathbf{w}_{opt} and $\tilde{\mathbf{d}}$ into Eq. 8.13. The predicted noise reduction for the controller is given by

$$\text{Noise Reduction} = 10 * \log_{10} \left(\frac{\mathbf{e}_{opt}^H \mathbf{e}_{opt}}{\tilde{\mathbf{d}}^H \tilde{\mathbf{d}}} \right) \quad (8.18)$$

The principal components of the control system can be analyzed by computing the singular value decomposition of $\hat{\mathbf{H}}$. The condition number of the control system is the ratio of the maximum singular value to the minimum singular value, or (s_1/s_{12}). There is no distinct condition number above which a control system is considered to be ill-conditioned, but based on recommendations from Ruckman and Fuller (1993) and Rossetti et al. (1996), a condition number greater than 50 is an indication the controller may converge slowly.

The estimated noise reduction for each PC in the controller was given in Eq. 4.14 and is repeated below:

$$\Delta_{dB}^i = 10 * \log_{10} \left(1 - \frac{|p_i|^2}{|\mathbf{p}|^2} \right) \quad (8.19)$$

where $p_i = \mathbf{u}_i^H \tilde{\mathbf{d}}$ is the correlation between the primary response and the i th PC. As noted in Chapter 4, the decibel units in Eq. 8.19 can obscure the importance of the PCs, so a more useful quantity is the percent each PC reduces the primary,

$$\text{Percent Reduction} = 100 * \frac{|p_i|^2}{|\mathbf{p}|^2} \quad (8.20)$$

The control effort required to cancel each PC is also related to p_i , and from Eq. 4.11, the effort is

$$|v_{i,opt}|^2 = \frac{|p_i|^2}{s_i^2} \quad (8.21)$$

where $v_{i,opt}$ is the optimal control filter weight for the i th PC. Principal components with small singular values will thus require a large control effort.

Chapter 5 described the statistical significance of a PC as an additional metric for analyzing the PCs of a control system. The significance quantifies the importance of a principal component, relative to noise reduction performance, when the microphone outputs contain random noise. A selection metric based on the significance was given earlier in Eq. 5.18, but a more useful metric can be obtained by rearranging that equation. A PC should be eliminated from the controller if

$$|p_i| \leq t_{(\alpha/2, m-r-1)} \sigma \quad (8.22)$$

$$\frac{|p_i|}{t_{(\alpha/2, m-r-1)} \sigma} \leq 1.0 \quad (8.23)$$

where σ is the standard deviation, in Volts, of the random background noise in the sensors. The quantity $t_{(\alpha/2, m-r-1)}$ represents the value of the Student's t -distribution with $(m - r - 1)$ degrees of freedom at a $100(1 - \alpha)\%$ confidence level. At a given confidence level, (where α should not be confused with the step size parameter for the PC-LMS algorithm), the metric in Eq. 8.23 quantifies whether p_i , the correlation between the primary disturbance and the i th PC, is statistically significant or could have been caused by random effects. A confidence level of 95% was used for the experiment, and the resulting t statistic value, with 36 degrees of freedom, was 2.021. Thus, from Eq. 8.22, $|p_i|$ will have to be at least twice as large as σ to be significant.

8.2 Results

The results of noise reduction experiments at 138 Hz and 147 Hz are discussed here. The purpose of these experiments was to compare the performance of the filtered-x LMS and PC-LMS feedforward control algorithms. As a result, the selection of the test frequencies was motivated partially by the dynamics of the cylinder-cavity system, and partially by differences in the conditioning of the control system at the two frequencies. The operation of the control system is discussed here first.

8.2.1 Control System Characterization

The baseline performance of the noise control system was established prior to the noise reduction tests. The interior acoustic response excited by the external loudspeakers was compared with the response excited by the control actuators. All of the data presented here were acquired through the DSP at a sample rate of 2500 Hz, and include the effects of conditioning filters and amplifiers.

Response to broadband excitation

The response of a single microphone inside the cylinder to a broadband input to the loudspeakers was compared to the response of the microphone to a broadband input to a single control actuator. These

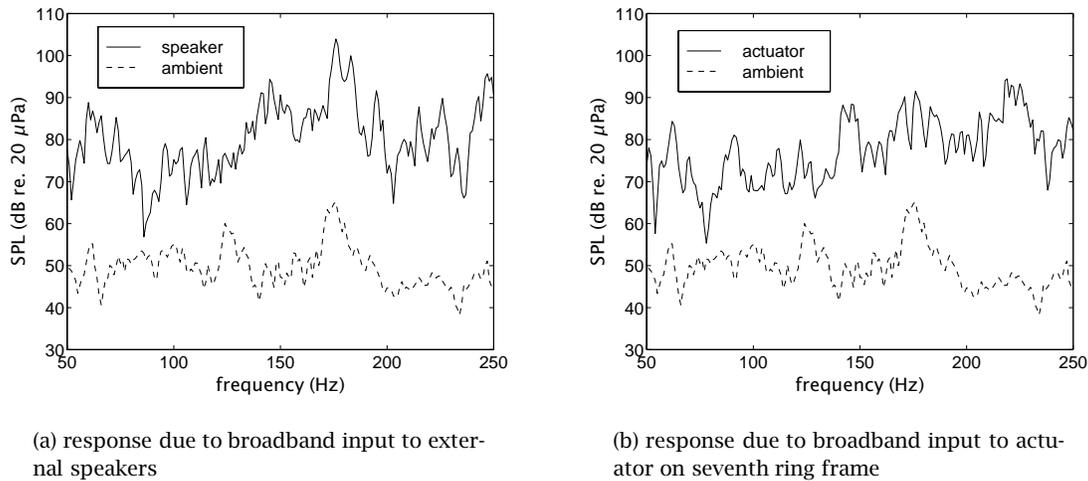


Figure 8.10: Response of microphone 37

responses quantify the forced response level relative to the background noise in the cylinder, and also quantify the authority of a control actuator relative to the loudspeakers. The response of microphone number 37, which was chosen arbitrarily for illustrative purposes, is shown here. This microphone was located on the fifth microphone array, as labeled in Fig. 8.4, 12° from the top of the cylinder.

The power spectral density (PSD) of the microphone response to a broadband signal sent to both loudspeakers is shown in Fig. 8.10(a). The response is plotted in calibrated sound pressure level (SPL), relative to a reference pressure of $20 \mu\text{Pa}$. The PSD is shown for a wide frequency range, approximately centered about the test frequencies. The dotted line in the plot is the PSD of the microphone response to background noise in the cylinder. Near the test frequencies of 138 and 147 Hz, the microphone response was 25 to 30 dB above the background noise level. The response of the microphone to a broadband input to the control actuator mounted on the seventh ring frame (see Fig. 8.6) is shown in Fig. 8.10(b). The response due to the actuator was similar to the response due to the loudspeakers, and was 20 to 25 dB above the background noise level. Thus the actuator excited an interior acoustic response with approximately the same level as the external loudspeakers.

Example of single transfer function

The control system was characterized in the vicinity of the test frequencies by examining the transfer function from actuators to microphones. The magnitude of the transfer function, averaged across all 48 microphones, quantifies the average cavity response due to the actuators as a function of frequency. Peaks in the transfer function can be correlated with data from a modal analysis of the cylinder. The average magnitude of the transfer functions from two actuators, averaged across all 48 microphones, is shown in Fig. 8.11. The solid curve corresponds to the actuator on the seventh ring frame, and the dotted line corresponds to the actuator on the tenth ring frame, in the upper half of Fig. 8.6. The data are plotted in dB, relative to the maximum transfer function value near 120 Hz.

The peaks in Fig. 8.11 correlate well with natural frequencies of the cavity and shell which were determined in a previous experimental modal analysis of the cylinder (Grosveld and Beyer 1986).

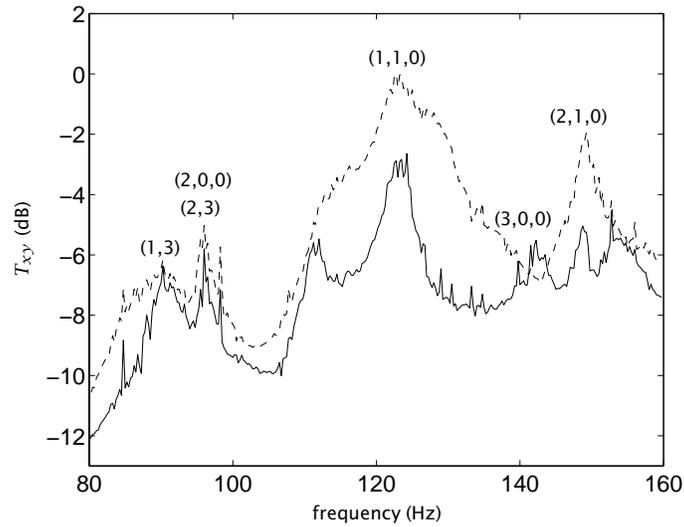


Figure 8.11: Average transfer function magnitude (— actuator on seventh ring frame; - - - actuator on tenth ring frame)

Table 8.1: Natural frequencies of shell and cavity

modal indices (m,n)	frequency (Hz)	modal indices (m,l,k)	frequency (Hz)
(1,2)	64	(1,0,0)	47
(1,3)	89	(2,0,0)	94
(2,3)	96	(1,1,0)	121
(1,4)	136	(3,0,0)	143
(2,4)	189	(2,1,0)	150
		(0,0,1)	171

(a) Shell modes

(b) Cavity modes

The natural frequencies of the first few shell and cavity modes are listed in Tables 8.2(a) and 8.2(b), respectively. Peaks in Fig. 8.11 that lie close to a natural frequency in either table are labeled with the corresponding modal indices. Because the correlation between microphone responses and mode shapes was not verified, the labels in Fig. 8.11 are only conjecture. In general, the contributions of individual modes to the response of the cylinder-cavity system are apparent in the vicinity of the test frequencies. The chosen test frequency of 138 Hz is close to the natural frequency of the (1,4) shell mode at 136 Hz, although the contribution of this particular mode to the transfer functions is not apparent in Fig. 8.11. The test frequency of 147 Hz is close to a natural frequency of the (2,1,0) cavity mode at 150 Hz, and this mode is evident in the transfer function plot. These test frequencies were chosen for their proximity to natural frequencies of structural and acoustic modes of the test structure.

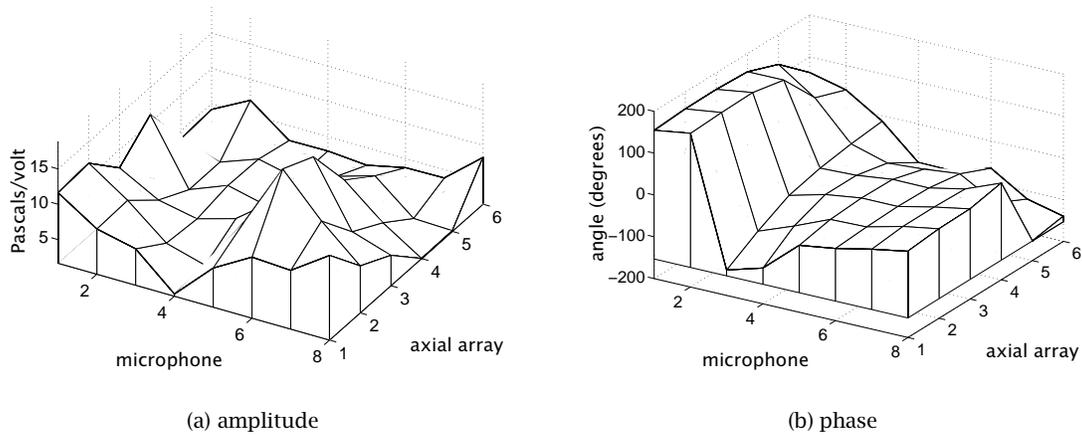


Figure 8.12: Primary response at 138 Hz

8.2.2 Results: 138 Hz

The noise control experiments consisted of the following steps:

1. measure the mean of the squared microphone responses due to background noise in the cylinder. The square root of this value was used as σ for computing the statistical significance in Eq. 8.23.
2. measure the microphone responses due to the primary disturbance, which was a single tone at either test frequency. The responses were computed as single frequency transfer functions, represented by two-coefficient FIR filters, from the reference signal generated on the DSP to each microphone.
3. with the primary disturbance turned off, compute $\hat{\mathbf{H}}$, the estimate of the transfer function matrix between sensors and actuators at the test frequency.
4. for the filtered-x LMS algorithm, $\hat{\mathbf{H}}$ was downloaded to the DSP for prefiltering the reference signal. For the PC-LMS algorithm, the SVD of $\hat{\mathbf{H}}$ was computed and the resulting singular vectors in the columns of \mathbf{U} and \mathbf{V} were downloaded to the DSP.
5. the primary disturbance was turned on and various control configurations were used to actively reduce the microphone responses.

At 138 Hz, the mean of the squared responses of the microphones due to background noise measured for 60 seconds was 0.37 Volts². The primary response was next measured, and the two-coefficient FIR filters converted to the frequency domain complex vector \mathbf{d} in Eq. 8.13. The amplitude and phase of the primary response are shown in Fig. 8.12 (the primary response was generated on the DSP, hence the data are plotted as a transfer function). The surface represents an unwrapped view of the microphone array, oriented such that the reader is looking into the access hatch of the cylinder. The circumferential pressure variation, along the axis labeled “microphone” in Fig. 8.12, corresponds roughly to that of a third order mode, with peak values at microphones six and one, which are 120° apart. The pressure varies relatively uniformly in the axial direction.

Table 8.2: Properties of principal components at 138 Hz

PC #	singular value	noise reduction (%)	control effort (Volts ²)	significance
1	7.91	75.3	12.5	22.7
2	4.21	18.3	10.7	11.2
3	2.40	1.7	3.0	3.4
4	1.77	3.0	10.1	4.6
5	0.92	0.4	5.0	1.7
6	0.60	0.2	6.8	1.3
7	0.42	0.2	9.3	1.0
8	0.33	0.1	7.6	0.7
9	0.30	0.2	17.6	1.0
10	0.19	0.0	11.3	0.5
11	0.16	0.1	21.9	0.6
12	0.10	0.0	2.1	0.1

The transfer function between actuators and sensors was measured next, and the predicted noise reduction from Eq. 8.18 was estimated to be 22.5 dB. Several other properties of the principal components, computed from the experimentally measured transfer function and primary response data, are listed in Table 8.2. The singular values are listed in the second column of Table 8.2, and indicate the condition number of $\hat{\mathbf{H}}$, given by (s_1/s_{12}) , was 79.1. As mentioned earlier, there is no strict threshold for the condition number that distinguishes a well-conditioned control system from an ill-conditioned one. Nonetheless, this condition number is high, and indicates the adaptive algorithms could converge slowly and the actuators could require high control inputs.

The predicted noise reduction was computed for each PC as a percent of the primary response using Eq. 8.20. The values listed in the third column of Table 8.2 indicate that only the first four PCs will reduce the primary response by more than 1%. The predicted control effort for each PC is listed in the next column, as computed from Eq. 8.21 and expressed in terms of Volts² of output of the D/A converter. The control effort is inversely proportional to the square of the singular value, hence the efforts do not decrease appreciably for PCs 9-11, even though these PCs produce negligible noise reduction. The last column in the table lists the statistical significance of each PC, computed using Eq. 8.23. According to the criteria given in Equation 8.23, PCs seven through twelve are not significant for reducing the primary response at this frequency.

Convergence behavior

The first noise reduction test involved a comparison of the performance of the filtered-x LMS algorithm with no effort penalty (i.e. $\beta = 0$) to the PC-LMS algorithm controlling all twelve PCs of the control system. Because the maximum step size parameter at this frequency was $\mu_{max} = (2/s_1^2) = 0.032$, the step size for the filtered-x LMS algorithm was initially set to one fourth of this value, or 0.008, and the step sizes for each PC in the PC-LMS algorithm were set to one sixteenth of this value, or 0.002. However, the filtered-x LMS algorithm was unstable with this step size, which may have been a result of the high condition number of $\hat{\mathbf{H}}$. After some experimentation, a μ of 0.0001 was found to be suitable for maintaining the stability of the algorithm for long periods of time.

Time histories of the squared microphone responses, $(\sum_i^m e_i^2(n))$, during five minutes of convergence of each control algorithm are shown in Fig. 8.13. The values are plotted on a logarithmic scale,

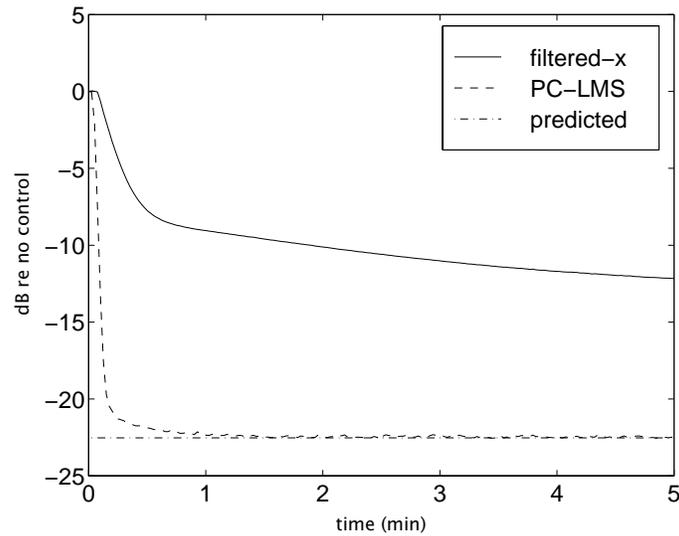


Figure 8.13: Convergence of sum of squared microphone responses at 138 Hz

relative to the total squared response with no control. The predicted noise reduction of 22.5 dB is plotted in the figure as a dotted line, and the PC-LMS algorithm quickly converged to this value. In contrast, the filtered-x LMS algorithm converged very slowly, and after five minutes reduced the primary response by only 12.2 dB. The slow convergence of this algorithm could have been due to two factors: the small step size parameter which was needed to maintain stable convergence, and the manner in which the weight updates were computed only one actuator at a time.

The time histories of the total control efforts, in Volts², used by each controller to obtain the noise reduction in Fig. 8.13, are shown in Fig. 8.14. The control effort for both controllers was still increasing after five minutes of convergence. The control effort for the filtered-x LMS algorithm appears to be increasing rapidly, which indicates the controller may have been unstable. This indicates the small step size used for the filtered-x LMS was small enough to maintain stable convergence for five minutes, but was not sufficient to completely stabilize the algorithm. The total predicted control effort at this frequency was 117 Volts², and the PC-LMS controller appears to have been converging to this value.

The squared control signals sent to the actuators, at the five minute point in Fig. 8.14, are shown in Fig. 8.15. In the terminology of the control cost function, these squared voltages correspond to the control effort for each actuator. The predicted control effort for each actuator is also shown in the plot. In general the control efforts computed by the PC-LMS controller agree well with the predicted values, while the filtered-x algorithm was driving actuators 6, 9, and 10 with much higher voltages than predicted. This large discrepancy, coupled with the excessive total control effort used by the filtered-x controller in Fig. 8.14, indicate the filtered-x LMS algorithm may not have been converging to the optimal control solution. The reason is not known, but it may have been a result of the relatively high condition number of the control system, or the manner in which the weights were updated one actuator at a time.

Control inputs to individual principal components are shown in Fig. 8.16. The values computed experimentally by the PC-LMS algorithm show generally good agreement with the predicted values, although the inputs to PCs 9 and 11 were higher than predicted. Time histories showing the convergence of the inputs to PCs 1, 11 and 12 are shown in Fig. 8.17. While the first PC converged almost

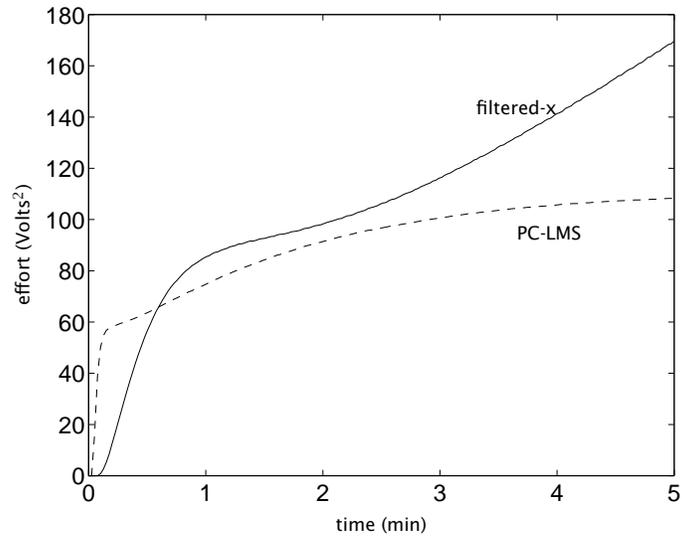


Figure 8.14: Control effort during convergence for 138 Hz

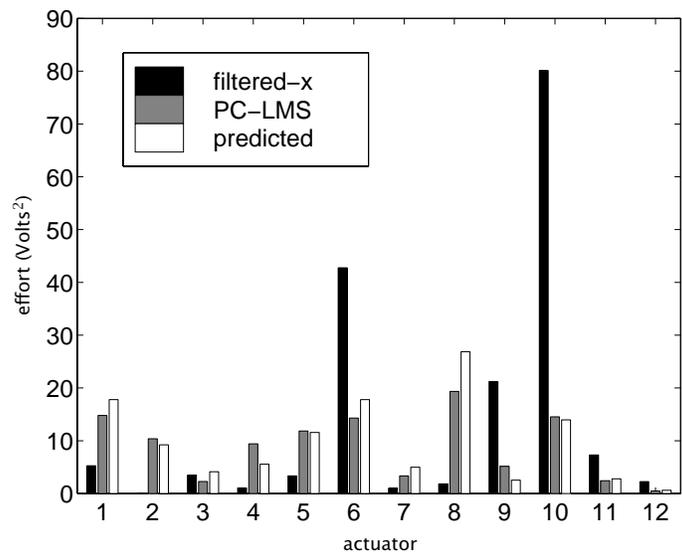


Figure 8.15: Control efforts for individual actuators

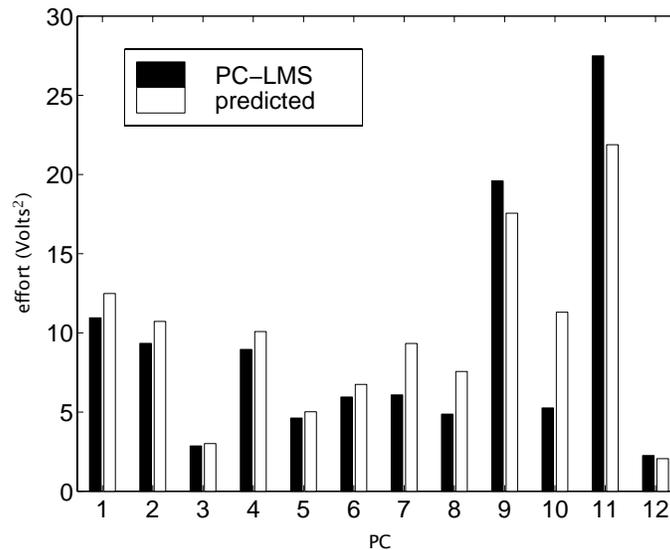


Figure 8.16: Control efforts for individual principal components

immediately, neither PC 11 nor 12 appears to have completely converged after five minutes. This slow convergence demonstrates the large time constant associated with the convergence of slow modes of the control system.

Control Effort Tests

The next series of tests concerned the implementation of a control effort penalty with the PC-LMS algorithm. As mentioned previously, high control efforts tend to be associated with the last few PCs. For this set of tests, the first one through nine PCs were used in the controller, adding one additional PC at a time. The step size parameter for all controlled PCs was set to $(\mu_{max}/16) = 0.002$, and the controller was allowed to converge for 60 seconds. The noise reduction and corresponding total control effort are shown in Fig. 8.18 as a function of the number of PCs in the controller. The noise reduction and control effort are plotted along the same axis for convenience. Both noise reduction and control effort increased as more PCs were added to the controller, although the control effort kept increasing after the noise reduction had leveled out after six PCs were controlled. This demonstrates how the last few PCs of a control system often produce very little noise reduction but can require large control efforts.

The implementation of a control effort penalty with the filtered-x LMS algorithm was then compared with the results obtained with the PC-LMS algorithm in Fig. 8.18. For these tests, the effort penalty parameter β in the filtered-x LMS algorithm was set to four different values of 10, 1, 0.1, and 0.01, to test a wide range of effort penalties. The results with the largest penalty, $\beta = 10$, and the smallest penalty, $\beta = 0.01$, are compared here with the PC-LMS algorithm controlling the first PC and the first four PCs, respectively. As described in Section 3.4, the effect of a uniform effort penalty in the filtered-x LMS algorithm can be described in terms of the principal components of the control system. Specifically, Eq. 3.40 quantifies how the effort penalty prevents the controller from canceling a PC when β is large relative to the singular value squared of the PC. By comparison with the singular values listed in Table 8.2, a value of $\beta = 10$ will affect PCs three through twelve, and $\beta = 0.01$ will affect PCs ten

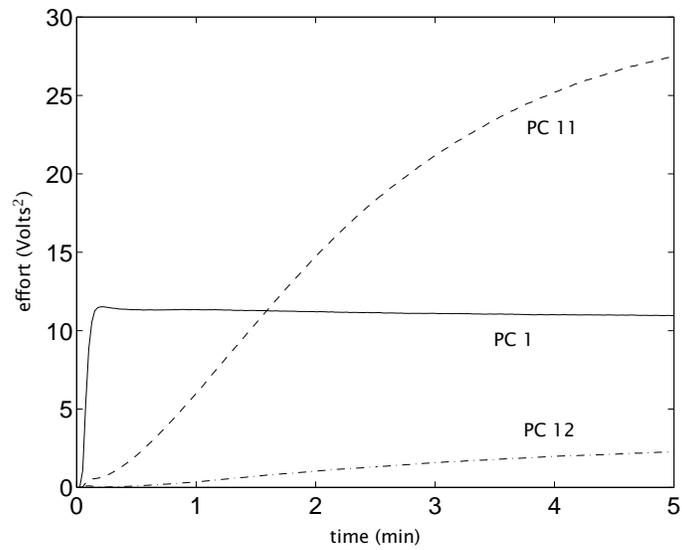


Figure 8.17: Control efforts during convergence for PCs 1, 11, 12

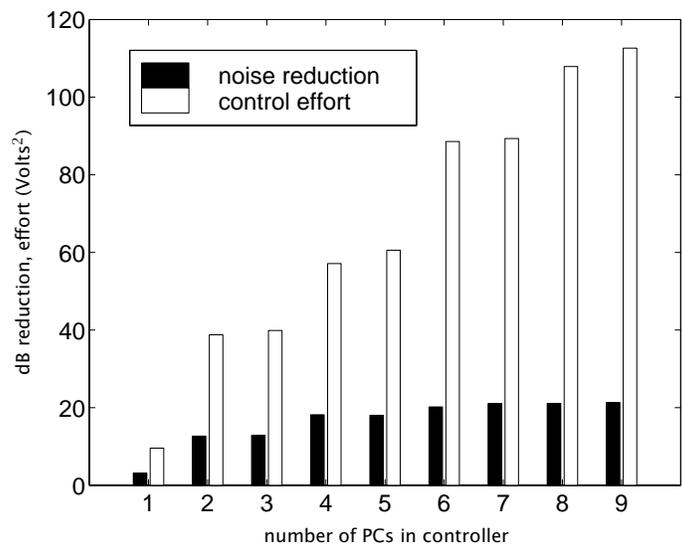


Figure 8.18: Noise reduction and control effort versus # of PCs in controller

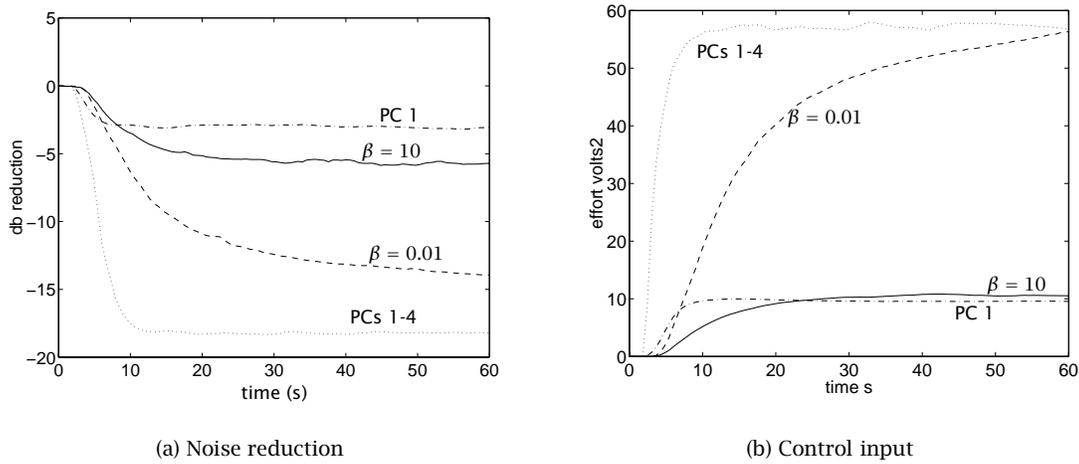


Figure 8.19: Comparison of PC-LMS and filtered-x effort penalties at 138 Hz

through twelve.

Time histories of the sum of squared microphone responses for the two cases are shown in Fig. 8.19(a), and the corresponding control efforts are shown in Fig. 8.19(b). For the large effort penalty of $\beta = 10$, the filtered-x LMS algorithm reduced the primary response by 5.5 dB, with a control effort of 10.5 Volts². By comparison, the PC-LMS algorithm, controlling only the first principal component, used a similar amount of control effort but reduced the primary by only 3.1 dB. For the smaller effort penalty of $\beta = 0.01$, the filtered-x LMS algorithm reduced the primary response by 14 dB, with a control effort of 57 Volts². By comparison, the PC-LMS algorithm, controlling the first four PCs, reduced the primary response by 18 dB using 57 Volts² of control effort. The PC-LMS algorithm therefore appears to be more useful for implementing a small effort penalty; more noise reduction is obtained with less control effort than the filtered-x LMS algorithm.

The differences between the two algorithms are due to the continuous nature of the effort penalty in the filtered-x LMS algorithm, and its discrete nature in the PC-LMS algorithm. Equation 3.40 from Chapter 3 quantified the minimum cost when the filtered-x LMS algorithm included a uniform effort penalty as

$$J_{min} = \sum_{i=1}^r \frac{\beta}{s_i^2 + \beta} |p_i|^2 + \sum_{i=r+1}^m |p_i|^2 \quad (8.24)$$

where p_i is the correlation between the i th PC and the primary. The effort penalty affects all r PCs of the control system, although its most significant effect is restricted to those PCs where $s_i^2 \leq \beta$. Even small values of β will thus have some effect on the first few PCs of the control system, which can limit the noise reduction achieved by the controller. In contrast, the PC-LMS approach to implementing an effort penalty involves completely canceling a subset of PCs, while ignoring the remaining PCs. When the cancelled subset contains a sufficient number of PCs, such as PCs 1-4 in Fig. 8.19, significant noise reduction can be obtained without spending any control effort on ill-conditioned PCs that produce little or no noise reduction.