isvr Institute of Sound and Vibration Research

Study of Indirect Force Determination and Transfer Path Analysis Using Numerical Simulations for a Flat Plate

A.N, Thite and D.J. Thompson

ISVR Technical Memorandum 851

May 2000

University of Southampton

SCIENTIFIC PUBLICATIONS BY THE ISVR

Technical Reports are published to promote timely dissemination of research results by ISVR personnel. This medium permits more detailed presentation than is usually acceptable for scientific journals. Responsibility for both the content and any opinions expressed rests entirely with the author(s).

Technical Memoranda are produced to enable the early or preliminary release of information by ISVR personnel where such release is deemed to the appropriate. Information contained in these memoranda may be incomplete, or form part of a continuing programme; this should be borne in mind when using or quoting from these documents.

Contract Reports are produced to record the results of scientific work carried out for sponsors, under contract. The ISVR treats these reports as confidential to sponsors and does not make them available for general circulation. Individual sponsors may, however, authorize subsequent release of the material.

UNIVERSITY OF SOUTHAMPTON

INSTITUTE OF SOUND AND VIBRATION RESEARCH

DYNAMICS GROUP

Study of Indirect Force Determination and Transfer Path Analysis Using Numerical Simulations for a Flat Plate

by

A.N. Thite and D.J. Thompson

ISVR Technical Memorandum No. 851

May 2000

ABSTRACT

Experiments representing a transfer path analysis are simulated numerically using a simply supported plate as a test structure. The point and transfer accelerance relations of a plate are derived using the modal superposition principle. To make the simulations more representative, random noise is introduced in the input and output signals. A study is carried out to assess the effect of an ill-conditioned accelerance matrix on the inverse estimation of several simultaneous coherent excitation forces and in turn the response at some arbitrary position due to these forces. The effect on these estimations of the number of responses and excitations used in the calculation is analysed. For all the estimations, strategies using (i) all singular values and (ii) elimination based on an error norm have been implemented and compared. A confidence interval for the predictions at the receiver location has also been derived at each frequency based on the covariance of the inputs and the covariance of the fit and compared with earlier calculations.

CONTENTS

APPENDICES

1. INTRODUCTION

Over the years the noise level targets for road vehicles have been substantially lowered and noise contributions by primary sources have been reduced significantly. For example, engines are much quieter than 10-20 years ago. Consequently, secondary sources have an increased importance. These secondary sources, as well as the primary sources, transmit noise to the interior of the vehicle through air-borne paths and/or structure-borne paths. Air-borne noise is transmitted into the vehicle due to fluid loading of panels. This in turn is due to the disturbance introduced by the source on the surrounding air. On the other hand, structure-borne sound reaches the receiver by transmission of vibrations from the source to the interior panels, in turn these panels radiate sound into the vehicle. In both of these cases the source could be at some distance from the receiver and the energy transfer could be through complicated paths. Solving problems of such a high complexity would require detailed knowledge of the paths and the relative contribution from each of these paths. Transfer path analysis is an experimental tool which helps in understanding these paths and their contribution, and which has found wide application in the automotive industry as well as other sectors.

1.1 Transfer path analysis

Transfer path analysis (TPA) [1,2] is a way of quantifying the contribution of vibrational energy through various paths from a source to the receiver location. These paths could be either air-borne or structure-borne. Basically, it involves the estimation of source strength and contributions due to these sources of different paths to the response. A source strength can be estimated using responses measured at a series of locations due to the operational source in combination with accelerances or mechano-acoustic transfer functions measured from source locations to these response locations. The response

measurement locations have to be chosen such that responses measured can capture contributions from all possible paths involved. The following expression can be used to estimate the strength of sources, here given as forces acting on the structure due to the source,

$$\{\hat{F}_\omega\} = \left[\hat{A}_\omega\right]^{-1} \{\hat{a}_\omega\} \qquad (1)$$

where $\hat{a}_\omega$ – is a vector containing operational responses at positions i

$\hat{A}_\omega$ – is the measured accelerance matrix between forces F_j and accelerations a_i

$\hat{F}_\omega$ – is a vector containing forces F_j that act to produce responses a_i

In each case $\hat{}$ is used to indicate measured or approximate quantities. Note that (1) applies for each frequency ω, and that force, accelerance and accelerations are complex quantities to include relative phase information. The matrix inversion in (1) can only be performed if $\hat{A}_\omega$ is a square full rank matrix. Otherwise a pseudo-inversion has to be performed to give a least-square solution.

Once the source strengths are known, the contribution from different paths at a receiver location can be estimated using

$$\{\hat{p}_\omega\} = \left[\hat{H}_\omega\right] \{\hat{F}_\omega\} \qquad (2)$$

where $\hat{H}_\omega$ are measured accelerances quantifying the path between the source locations and the receiver location.

Based on the contribution of different paths, they can be ranked in terms of their importance. These rankings can be used for modifying the structure so that the overall

noise is reduced. Subjective assessment can be combined with this analysis to achieve greater effectiveness.

In TPA, the accuracy of the predictions depends on

- Accuracy of the measurements of operational responses, $\hat{a}_i$
- Accuracy of measurements of transfer functions, $\hat{A}_\omega$ and $\hat{H}_\omega$
- Accuracy of source strength estimation i.e. the matrix inversion
- Appropriate choice of force and response locations.

1.2 Matrix condition

As the accelerance matrix may be ill-conditioned at many of the frequencies, peaky force predictions may occur at these frequencies due to the presence of small errors in the measurement of accelerances $\hat{A}$. This can lead to over- or under-estimation of the contributions from these paths. A means of improving the condition of the matrix is required without much compromise in the estimation of source strengths. One way of achieving this is by singular value decomposition (SVD) [1]. Any matrix can be decomposed using SVD as

$\hat{A} = USV^H$

where,

S = diag(σ_1, σ_2 ,...., σ_n) , where ($\sigma_1 \geq \sigma_2$,...., $\geq\sigma_n$) are singular values and n is the dimension of the accelerance matrix. The singular values are square roots of eigen values of $\hat{A}^H \hat{A}$ with H the Hermitian transpose. They are positive real values.

U = Eigen vectors of $\hat{A}^H \hat{A}$ and $U U^H = I$, i.e. U is a unitary matrix

and

V = Eigen vectors of $\hat{A}\hat{A}^H$ and $V V^H = I$, i.e. V is a unitary matrix

Pseudo-inversion of the accelerance matrix can be obtained by writing

$\hat{A}^+ = VS^{-1}U^H$ with $S^{-1}=\text{diag}(\sigma_1^{-1}, \sigma_2^{-1},\ldots,\sigma_n^{-1})$

If any of the singular values are 0, they are replaced by 0 not σ^{-1} in S^{-1}. Singular values which are small are influenced more by measurement noise. They can be eliminated using a threshold value without much effect on the estimated source strengths during the inversion process i.e. all values of $\sigma < \varepsilon$, the error threshold, are taken as zero. The problem here is how to estimate the threshold value for the rejection of singular values.

In this study the process of rejecting singular values is investigated [1] in order to understand the process of inversion for force reconstruction. Computer simulations are performed to represent experiments carried out on a simply supported plate. Chapter 2 introduces the formulation of experiments. The results are discussed in chapters 3-6 along with different strategies for averaging. Chapter 7 describes the confidence interval estimation which may be used in practice. The theoretical formulation for plate flexural vibration used in the simulations is given in Appendix A, based on [3,4].

2. FORMULATION OF EXPERIMENTS

2.1 Introduction

A series of experiments is simulated based on a simply supported rectangular plate, in order to investigate the process of matrix inversion in force reconstruction. The arrangement of the plate along with force and acceleration measurement locations are as shown in the Figure 1. For convenience only 3 force and 4 response locations are shown on the plate. Other details of the plate are given in section 2.4 along with details of the force and response locations.

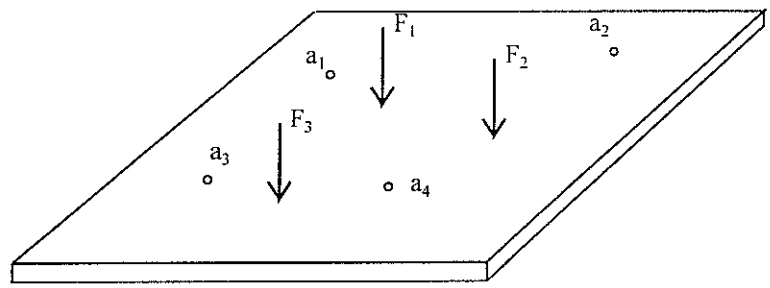


Figure 1. Simply supported plate showing force and response positions shematically.

The following 'measurements' are made with computer simulations representing the experiments,

I) Accelerations under operating conditions $\{\hat{a}\}$ due to the action of a number of coherent forces acting simultaneously

II) Accelerances $[\hat{A}]$ to represent the paths from each force position to each response location

III) Accelerances from force locations to a receiver location $[\hat{H}]$, in this case acceleration being chosen as the receiver quantity. This point is different from those used in II.

The measurements I and II are used to estimate the forces at the source locations. Once these reconstructed forces are known, the response at the receiver location is predicted using measurement III and the reconstructed forces.

2.2 Representation of acceleration measurement under operating condition

Accelerations at each response location are calculated as,

$$\{\hat{a}_i\} = \left[\sum_{m=1}^{n_e}\left(\{A_{i,m}\}.\{F_m\}\right)\right] + N_i \qquad i=1, 2, \ldots\ldots n_r \tag{3}$$

where,

A - Theoretical accelerances which are calculated based on equation (A15) in Appendix A

F - Forces acting on the structure

n_r - Number of locations for acceleration measurement

n_e - Number of sources

N – Random noise with a normal distribution and is a complex quantity.

2.3 Representation of accelerance measurement

Experiments to estimate the accelerance that would be measured in the presence of noise can be modelled as in Figure 2. X is the 'measured' force and Y is the 'measured' acceleration.

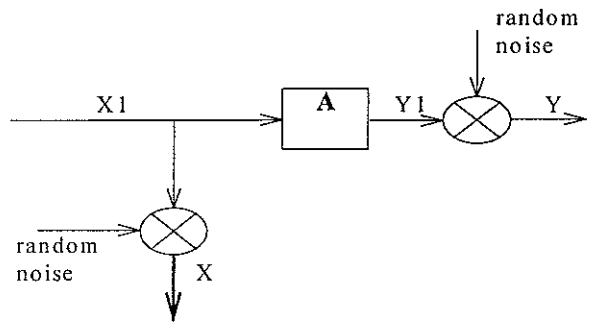


Figure 2. Model of experiment for the measurement of accelerance

$$\text{Measured accelerance} = \frac{S_{xy}}{S_{xx}}$$

where S_{xy} is the cross spectrum between X and Y

S_{xx} is the power spectrum of X

The Coherence γ is given by

$$\gamma^2 = \frac{|S_{xy}|^2}{S_{xx} . S_{yy}}$$

The coherence indicates the degree of dependence of the input and output signals X and Y. For a single calculation, the coherence is always 1 [5]. However, when an average is taken over a number of samples, the coherence becomes less than or equal to one. Based on the above theory, experiments can be simulated by the following relations,

$$\hat{A}(\omega) = \frac{\hat{S}_{xy}(\omega)}{\hat{S}_{xx}(\omega)} \tag{4}$$

where,

$$\hat{S}_{xy}(\omega) = \frac{1}{n_{av}} \sum_{1}^{n_{av}} X_n^*(\omega) Y_n(\omega) \tag{5a}$$

$$\hat{S}_{xx}(\omega) = \frac{1}{n_{av}} \sum_{1}^{n_{av}} X_n^*(\omega) X_n(\omega) \tag{5b}$$

n_{av} - number of samples for averaging

and

$$X = \{F\} + N_{in} \tag{6a}$$

$$Y = \{A\}.\{F\} + N_{out} \tag{6b}$$

where F - Force spectrum of unit amplitude

N_{in} - Random noise spectrum for the input

N_{out} - Random noise spectrum for the output

2.4 Force reconstruction

From the above 'measurements', acceleration and accelerance data are available which can be used in the reconstruction of the forces. For the reconstruction of forces, the accelerance and acceleration matrices have to be assembled as follows for each frequency,

$$\hat{A}_\omega = \begin{bmatrix} \hat{A}_{1,1} & \hat{A}_{1,2} & . & \hat{A}_{1,n_c} \\ \hat{A}_{2,1} & . & . & . \\ . & . & . & . \\ . & . & . & . \\ \hat{A}_{n_r,1} & . & . & \hat{A}_{n_r,n_c} \end{bmatrix} \quad \text{and} \quad \hat{a}_\omega = \begin{Bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ . \\ . \\ \hat{a}_{n_r} \end{Bmatrix}$$

The coherence matrix for each frequency can be generated as,

$$\gamma_\omega = \begin{bmatrix} \gamma_{1,1} & \gamma_{1,2} & . & \gamma_{1,n_c} \\ \gamma_{2,1} & . & . & . \\ . & . & . & . \\ . & . & . & . \\ \gamma_{n_r,1} & . & . & \gamma_{n_r,n_c} \end{bmatrix}$$

In the next step, the accelerance matrix is decomposed using SVD. If the condition of the matrix is poor at a particular frequency and at the same time the coherence is also low, then as suggested by reference [1], a threshold value can be established for rejecting singular values by the relation

$$E_{ij} = \alpha \left[\frac{1-\gamma^2_{\omega,ij}}{2n_{av}\gamma^2_{\omega,ij}} \right]^{1/2} \left| \hat{A}_{\omega,ij} \right| \qquad (7)$$

E_{ij} defines the limits within which the $\hat{A}_{\omega,ij}$ will fall [7]. If α is 3, then 99.7% of $\hat{A}_{\omega,ij}$ will fall within E_{ij}. The norm of the error matrix ||E|| can be used as a threshold to reject

the singular values. If all the singular values are less than the error norm then the largest singular value is replaced by error norm itself.

Once a pseudo-inverse of $\hat{A}_\omega$ has been obtained with the above considerations, forces can be reconstructed as

$$\{\hat{F}_\omega\} = \left[\hat{A}_\omega\right]^+ \{\hat{a}_\omega\}$$

These reconstructed forces in turn can be used to estimate the response at the receiver location as,

$$\{\hat{p}_\omega\} = \left[\hat{H}_\omega\right] \{\hat{F}_\omega\}$$

while the actual response is given by

$$\{p_\omega\} = [H_\omega] \{F_\omega\}$$

where H_ω is the exact transfer function matrix, F are the exact forces.

Based on the above explanation, a matlab program has been written to simulate the experiments. This is given in Appendix B.

2.5 Input data

The simulations carried out represent the measurements for the following situation,

Material

Material of the plate - Steel

Young's modulus $E = 2.07 \times 10^{11}$ N/m^2

Poisson's ratio $\nu = 0.3$

Mass density $\rho = 7850$ kg/m^3

Damping factor $\mu = 0.03$

Dimensional data

The rectangular plate has dimensions 0.6m x 0.5m x 0.0015m. Table 1 gives the position of responses and forces along with force magnitude, while Figure 3 gives a pictorial view of the positions.

Table 1. Non-dimensional force and response positions

Force positions				Response positions		
Position	x/a	y/b	Force [N]	Position	x/a	y/b
1	0.41	0.43	19	1	0.55	0.4
2	0.51	0.63	10	2	0.8	0.2
3	0.62	0.41	27	3	0.9	0.8
4	0.31	0.72	6	4	0.6	0.5
5	0.83	0.25	35	5	0.2	0.3
6	0.71	0.89	0.1	6	0.3	0.7
				7	0.5	0.9
				8	0.7	0.71
				9	0.62	0.31

The positions of excitation and response points are obtained by multiplying the above values with the dimensions of the plate in the x (0.6m) and y (0.5m) directions.

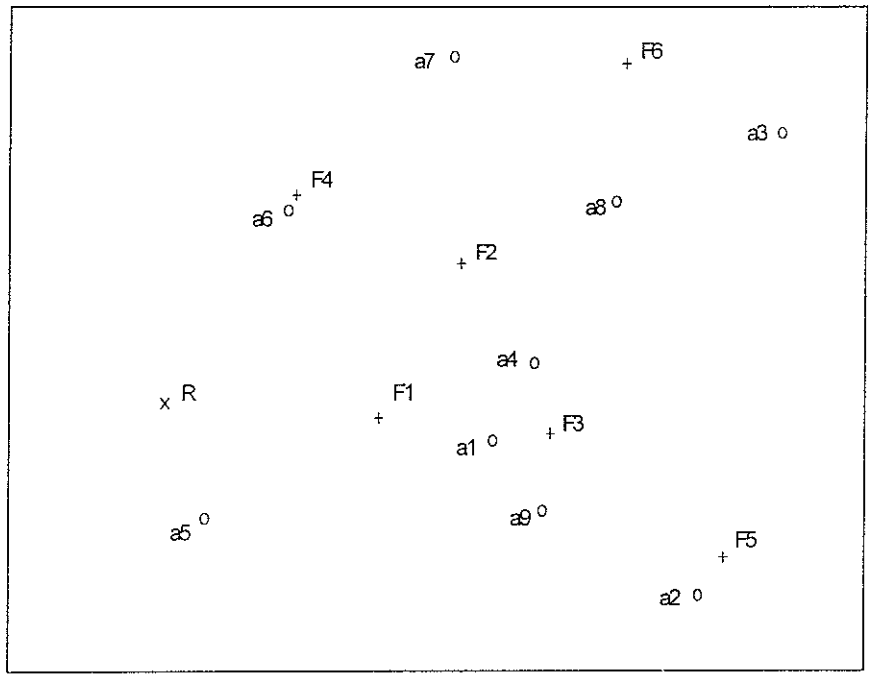


Figure 3. Response and force locations on the plate, R is the position of the receiver

Experiments are conducted with

$n_r = 4$ and $n_e = 4$,

$n_r = 6$ and $n_e = 4$,

$n_r = 6$ and $n_e = 6$,

$n_r = 9$ and $n_e = 6$.

This allows differences in matrix size to be studied as well as the effect of over-determination ($n_r > n_e$) on the force reconstruction and the response prediction. For the various experiments with different numbers of force and response points, the first n_r response points and n_e forces from Table 1 are used in each case. The force spectra are of constant amplitude as listed when converted to 1/3 octave bands. The phase difference between forces is taken to be zero, i.e. all forces are in phase.

The receiver location (see Figure 3)

(x,y)=(0.15a,0.45b)

Noise data

The noise added to the signals (see equations (3) and (6)) is Gaussian random.

Typical 1/3 octave band noise spectra, as added to the inputs and outputs of accelerance measurements, are shown in Figures 4a and 4b. The same noise model is used for the acceleration measurements used in obtaining both the accelerances and the operational responses. The level of noise in the force measurement at all frequencies is constant, whereas the noise in the velocity signals increases at 10 dB per decade (although the theory has been presented in terms of acceleration, results will be given in terms of velocity).

N.B. The force spectrum itself is unity for accelerance measurements, so the noise is 1% of the signal.

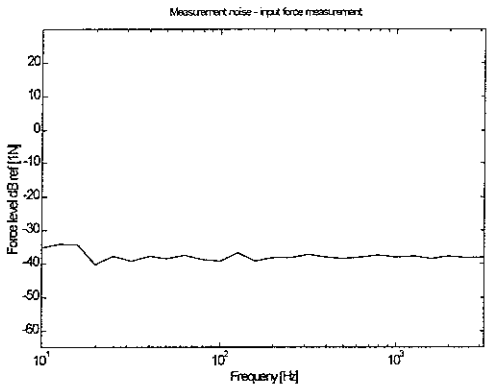


Figure - 4a. 1/3 octave band noise in the measurement of force during accelerance measurement.

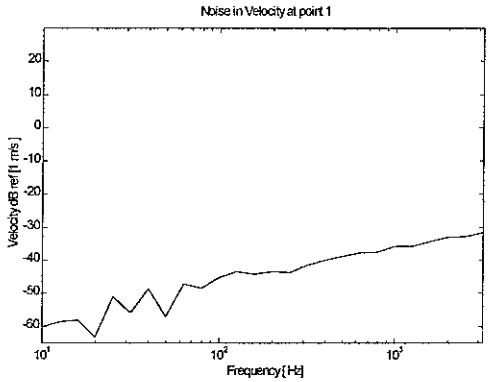


Figure - 4b. 1/3 octave band noise in the measurement of acceleration converted to velocity

Averaging

The number of averages used in estimating the accelerance and acceleration are as follows

$\hat{A}$ - 50 samples , $n_{av} = 50$

$\hat{H}$ - 200 samples

$\hat{a}_i$ - 25 samples, $n_s = 25$

The frequency range used for the study is from 10 Hz to 4 kHz. At low frequencies the response is dominated by individual modes, at high frequencies by multiple modes. The first resonance is at 25 Hz and there are 250 modes up to 4 kHz.

A typical operational velocity response for one of the response positions is as shown in Figure 5. This can be compared with the 'measurement' noise of Figure 4b. Figure 6 shows the transfer mobility magnitude, phase and coherence for the test structure for one of the response and excitation positions. The influence of measurement noise can be seen at high frequencies and is very clear from the fall in coherence. For the plate used in this experiment, the theoretical modal overlap is shown in Figure 7 for the damping factor of 0.03. It exceeds unity for frequencies above 500 Hz. This is obtained based on the expression given in reference [6].

In this study all the responses are presented in 1/3 octave bands. However, the force reconstruction and receiver position estimations are carried out in narrow bands and then converted to 1/3 octaves.

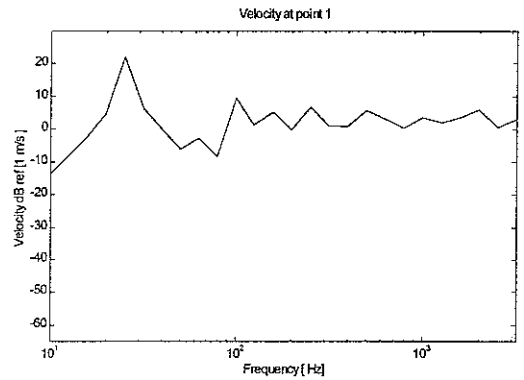


Figure 5. Velocity at position 1 for the combined action of first four forces given in table 1.

The experiments contain trials which can help investigate the following:

- Singular value rejection based on the error norm
- Using all singular values
- Effect of numbers of forces and responses
- In addition the effect of various averaging strategies are investigated in the following chapters

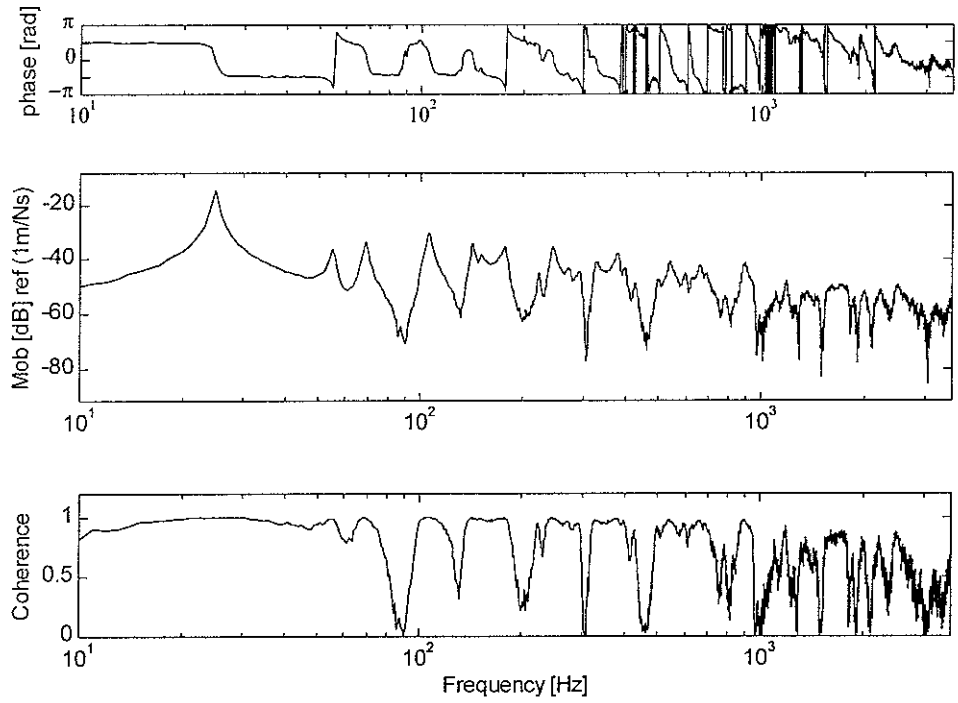


Figure 6. Transfer mobility from excitation point 1 to response point 1

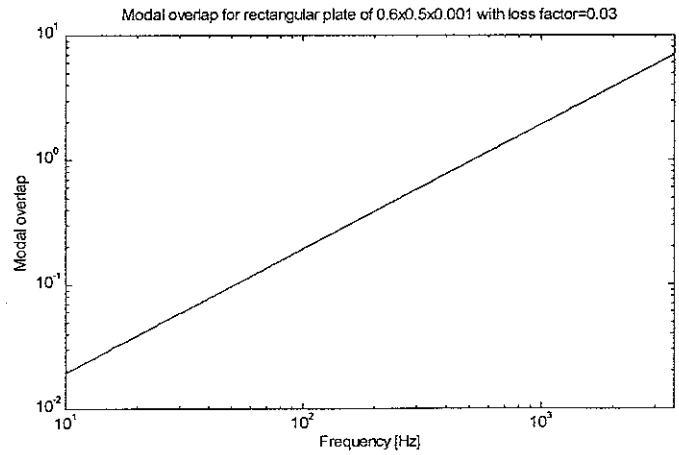
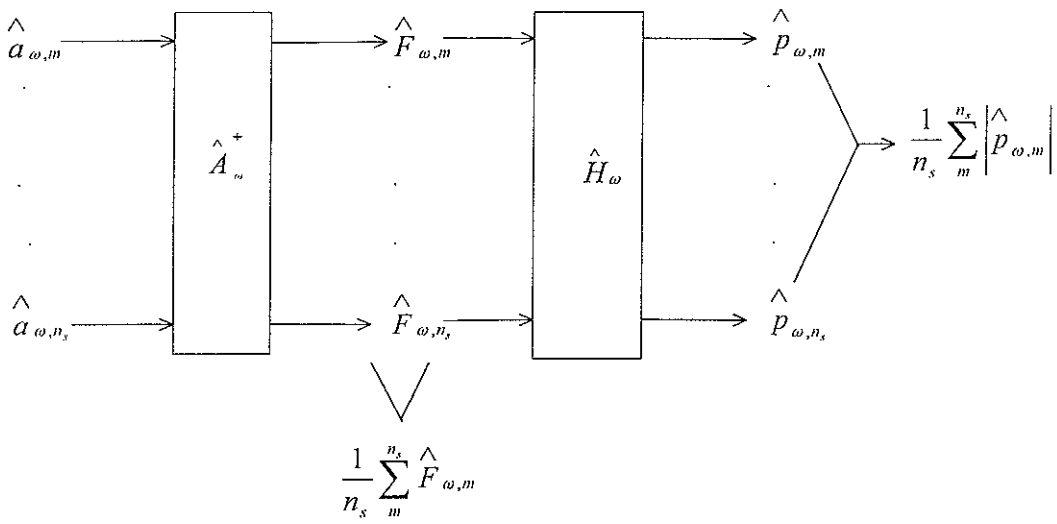


Figure 7. Modal overlap for hysteretic damping factor of 0.03.

3. EXPERIMENTS AND RESULTS USING SINGLE ACCELERANCE MATRIX

The process used for force reconstruction and the response prediction at the receiver location, in this chapter, is shown in the flow chart below. It is refered to as the 'single accelerance-based averaging' for reasons which will become clear in the next chapter.



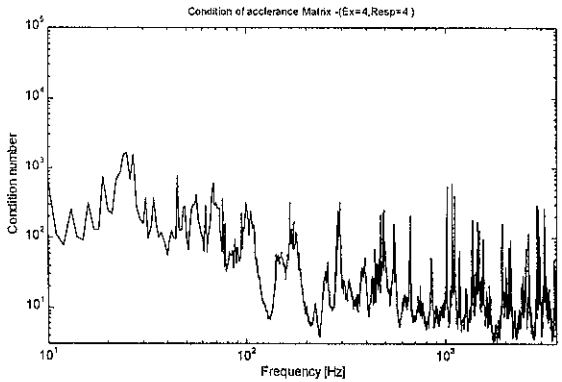
A series of experiments is performed to measure the operational accelerations $\hat{a}_m$ at n_r locations. The accelerances $\hat{A}_{i,j}$ from each forcing location j to each operational response location i are also measured. The accelerances are estimated by averaging over n_{av} samples (see equation (5)) and this same average accelerance is used in the reconstruction of forces throughout. The operational acceleration data $\hat{a}_m$ is rearranged for each frequency to form a vector of length n_r. Each of the n_s measurement samples of this operational acceleration $\hat{a}_{\omega,m}$ is used to calculate a force vector $\hat{F}_{\omega,m}$ and the receiver response $\hat{p}_{\omega,m}$ at each frequency. Averaging is performed over the magnitude of these estimations to form the mean and the 68% confidence intervals of the responses.

The process described here differs from the more usual methods of application of TPA in the way that all samples of operational responses are used in the estimation of a series of reconstructed forces, while the usual methods [1,2] consider only the mean operational response. A comparison with the latter is given in section 7 below.

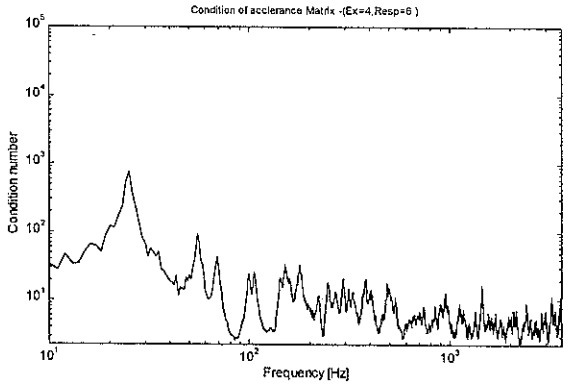
3.1 Condition number

Figures 8a-f show the condition number of the $\hat{A}$ matrix for six different experiments. These include the four combinations of n_r and n_e discussed above and the two cases with no 'measurement noise'. In each of the cases, the condition number decreases as the frequency increases. At low frequencies the mode shapes are very simple and the modal density very low so deflections are related at all the locations. The placement of forces at any location would excite the same modes and responses at any location would contain the same modes. This results in dependent columns or rows in the accelerance matrix. Hence the condition numbers are quite high at these frequencies. As the modal overlap increases, the chance of placing forces and response positions at closely related locations is smaller. This results in lower condition numbers at high frequencies in general. However, the condition numbers are high at some high frequencies, especially when less responses and excitations are used i.e. lower n_r and n_e.

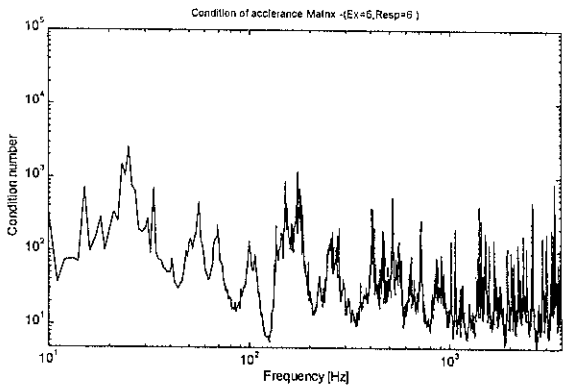
The effect of over-determination (i.e. $n_r > n_e$) on the condition number of the accelerance matrix at different frequencies is clear from the Figures (8b,d). With over-determination for example when n_r=6 and n_e=4, the condition numbers at higher frequencies improve significantly. Using more response positions without over-determination can lead to a worse condition number as seen from Figures 8a and 8c.



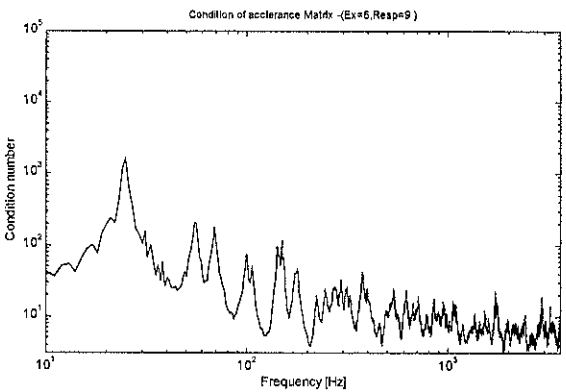
a. 4 sources and 4 responses



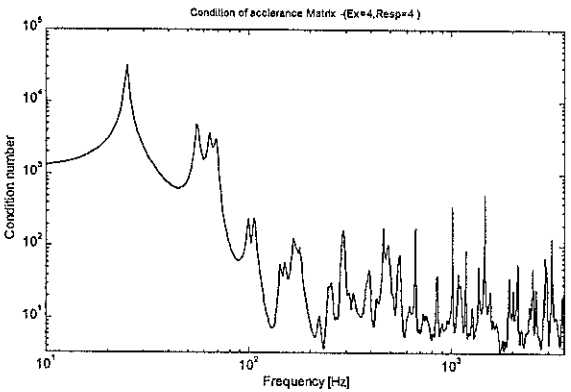
b. 4 sources and 6 responses



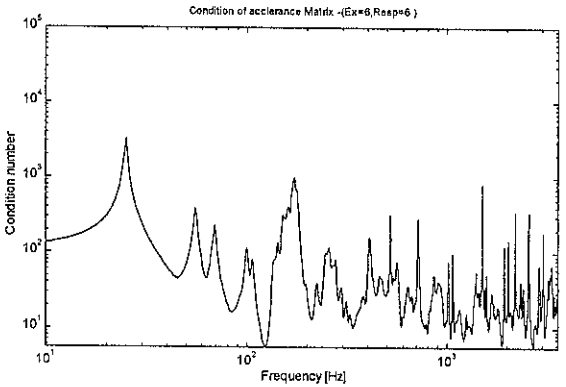
c. 6 sources and 6 responses



d. 6 sources and 9 responses



e. 4 sources and 4 responses without noise



f. 6 sources and 6 responses without noise

Figure 8. Condition number of Accelerance matrix for different set of experiments

The introduction of random noise increases the condition numbers at some frequencies, especially at higher frequencies, and also results in spiky behaviour with respect to frequency. On the other hand it has good effect in reducing the condition number at lower frequency. This is clear from comparison of figures 8a and 8e.

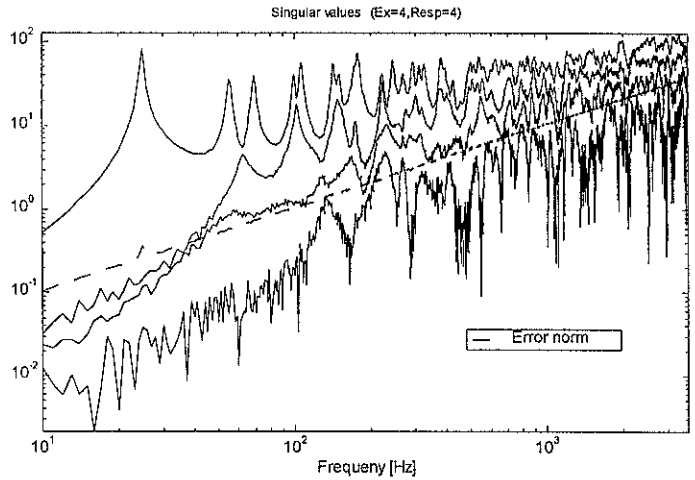
3.2 Error norm

Figures 9a-d show the singular values and the error norm of the $\hat{A}$ matrix for four different experiments. The singular values are well separated at lower frequencies, hence leading to higher condition numbers. When $n_r > n_e$, the lowest singular values increase at a higher rate with the frequency than for $n_r = n_e$. Hence for over-determination the condition numbers are improved at higher frequencies.

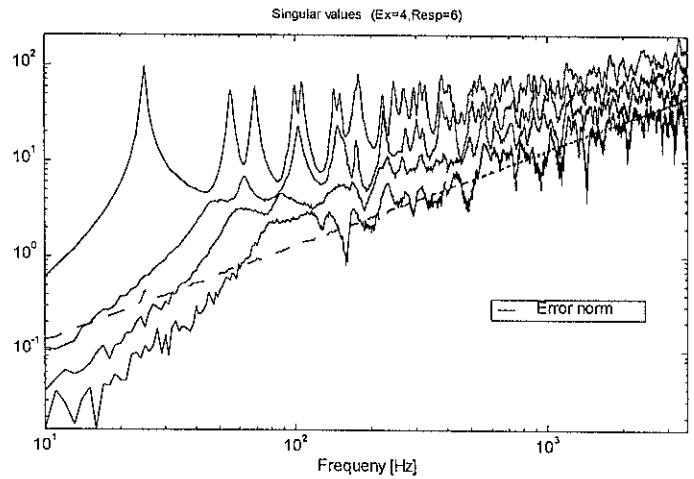
The error norm increases as the frequency increases. This is a feature of the assumed noise spectrum. The error norm becomes quite large compared to the singular values at higher frequencies, in fact about half of the singular values fall below it at certain frequencies.

3.3 Reconstructed forces

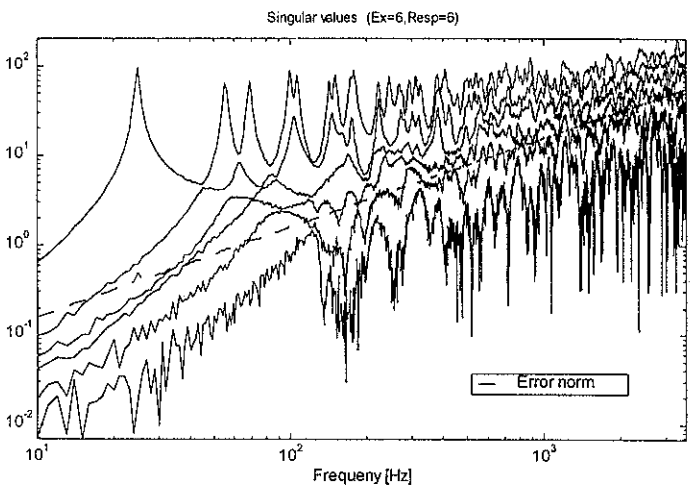
Figures 10-13 show the estimated forces along with the number of singular values used, for four experiments involving singular value rejection based on the error norm. The actual forces have constant levels at each frequency (Table 1) as shown at the right for comparison. The sum of the forces shown in the figures is the summation at each frequency including phase in narrow bands and then converted to 1/3 octave bands. The reconstructed forces are erroneous at lower frequencies in all the cases considered, although they improve with over-determination. When $n_r = n_e$ the results at higher frequencies are also rather erratic. This is due to the fact that about half of the singular



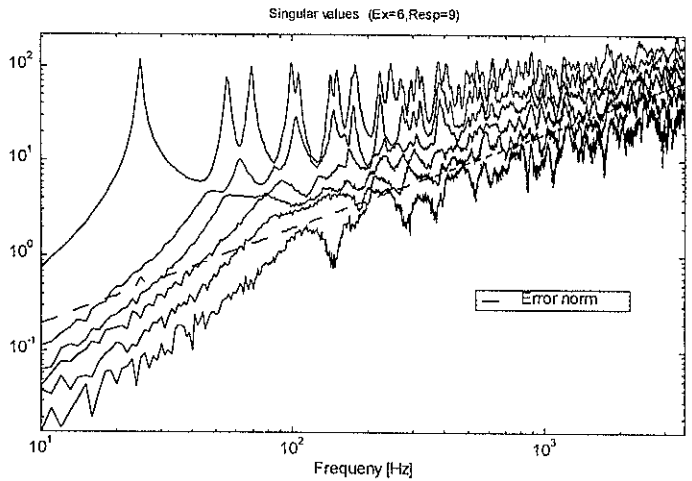
a. 4 sources and 4 responses



b. 4 sources and 6 responses



c. 6 sources and 6 responses



d. 6 sources and 9 responses

Figure 9. Singular values along with error norm of accelerance matrix

values fall below the error norm at higher frequencies and eventually get rejected. With over-determination, there is an improvement in the reconstruction of the forces, although force 6 which has a small amplitude is always considerably overpredicted.

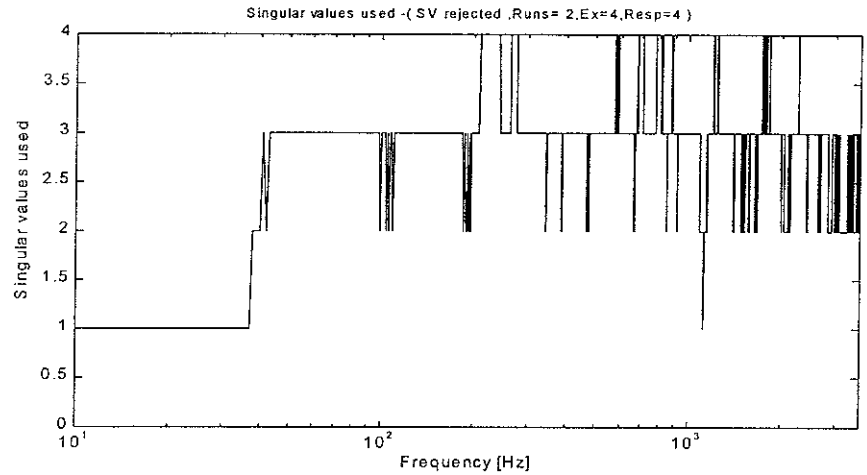


Figure 10a. Number of singular values used in force reconstruction for 4 sources and 4 responses

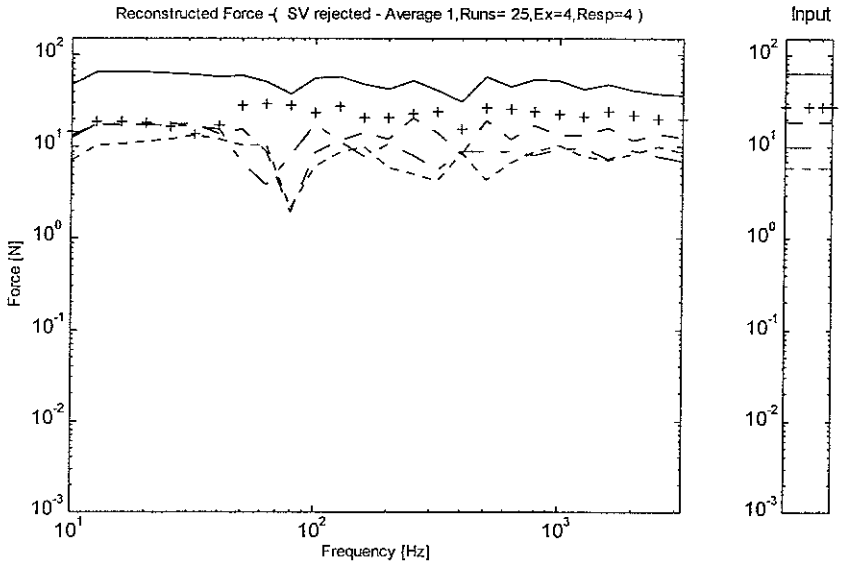


Figure 10b. Reconstructed forces for 4 sources and 4 responses- Singular values eliminated using error norm. Force1- - - -,Force2__ _ __,Force3 ++++,Force4 __ __ __, Sum of forces _____

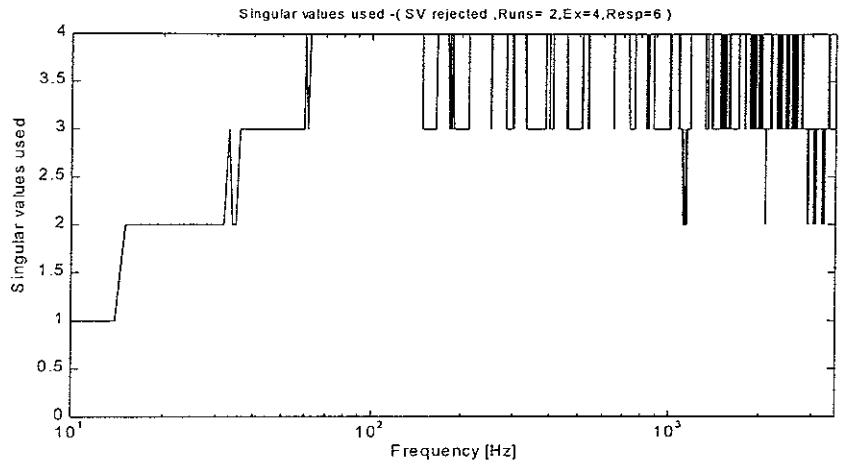


Figure 11a. Number of singular values used in force reconstruction for 4 sources and 6 responses

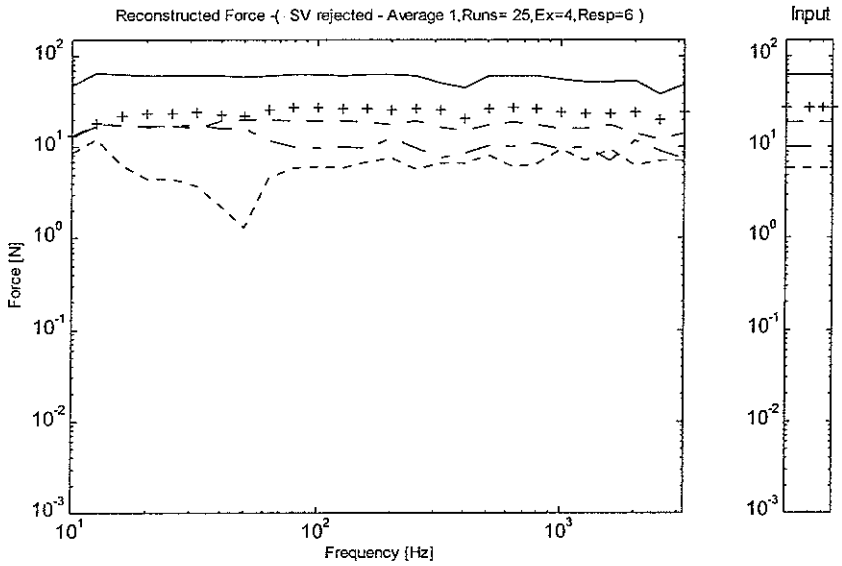


Figure 11b. Reconstructed forces for 4 sources and 6 responses- Singular values eliminated using error norm. Force1- - - -,Force2__ _ __,Force3 ++++,Force4 __ __ __, Sum of forces _____

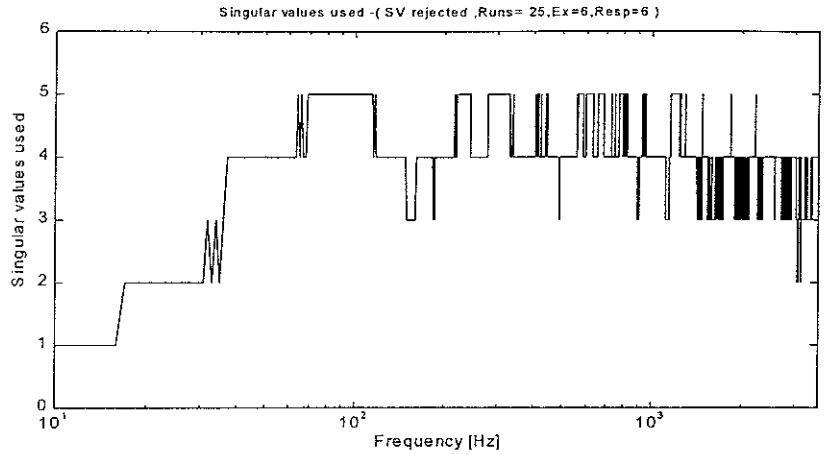


Figure 12a. Number of singular values used in force reconstruction for 6 sources and 6 responses

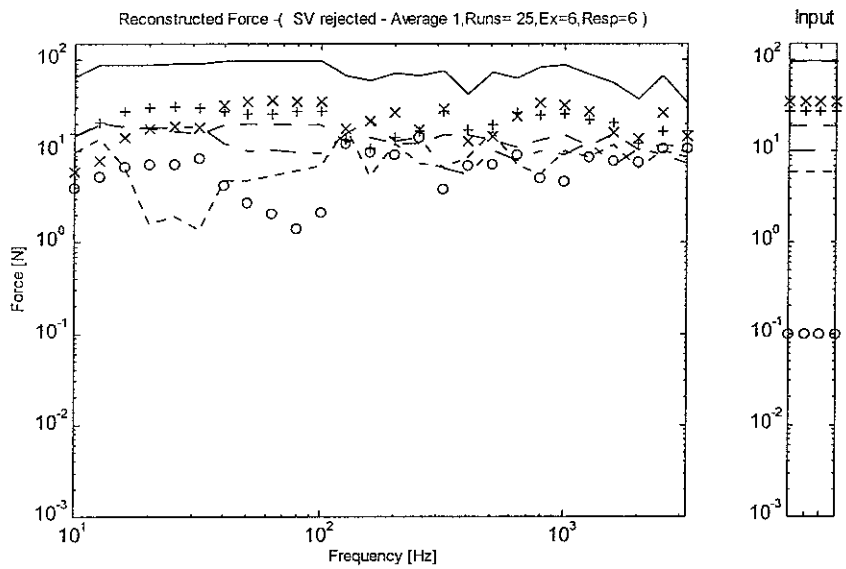


Figure 12b. Reconstructed forces for 6 sources and 6 responses- Singular values eliminated using error norm. Force1- - - -,Force2__ _ __,Force3 ++++,Force4 __ __ __,Force5 xxxxxx,Force6 oooo, Sum of forces _____

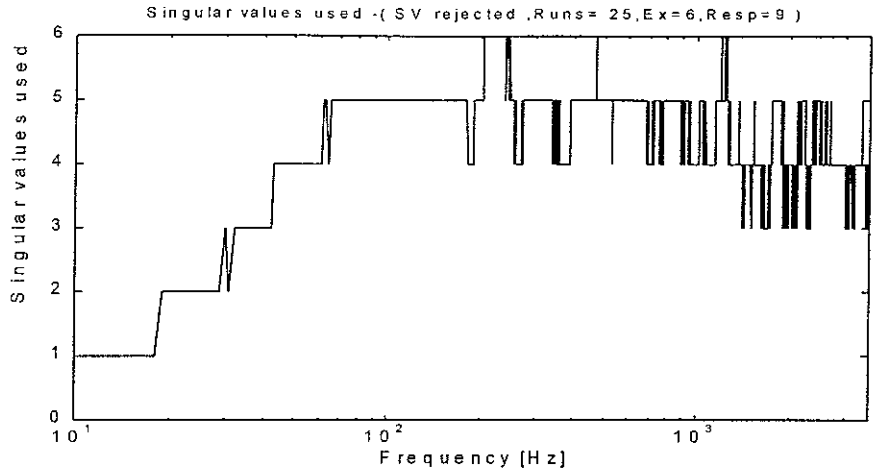


Figure 13a. Number of singular values used in force reconstruction for 6 sources and 9 responses

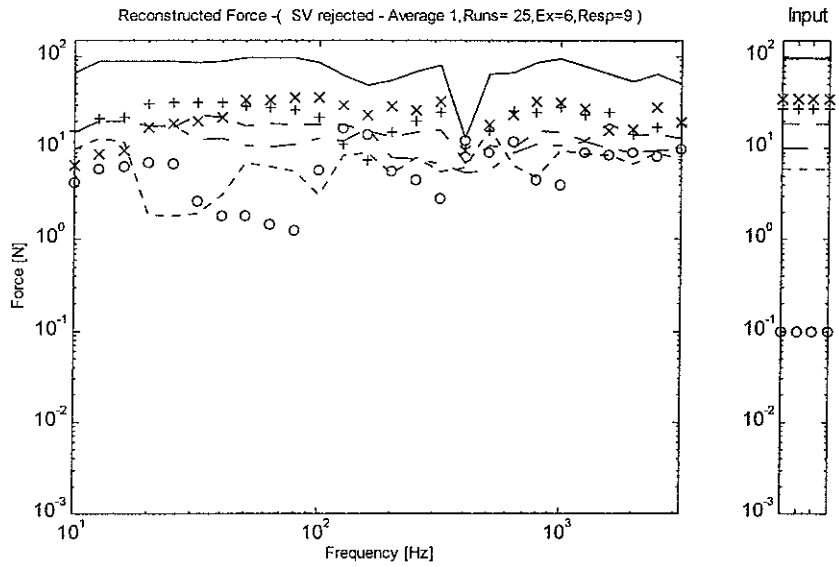


Figure 13b. Reconstructed forces for 6 sources and 9 responses- Singular values eliminated using error norm. Force1- - - -,Force2__ _ __,Force3 ++++,Force4 __ __ __,Force5 xxxxxx,Force6 oooo, Sum of forces _____

When all singular values are used, the forces are reconstructed more reliably, as shown in Figures 14-17. For $n_r = n_e$, the reconstruction error is found to be high at lower frequencies, below 1000 Hz. There is a considerable effect of the higher condition number (Figures 8a,c) on these estimations at lower and middle frequencies, but it improves considerably at higher frequencies. Neverthless the total force is estimated more reliably than individual forces. When the system is over-determined, the force reconstruction improves considerably, however the reconstruction remains poor at low frequencies.

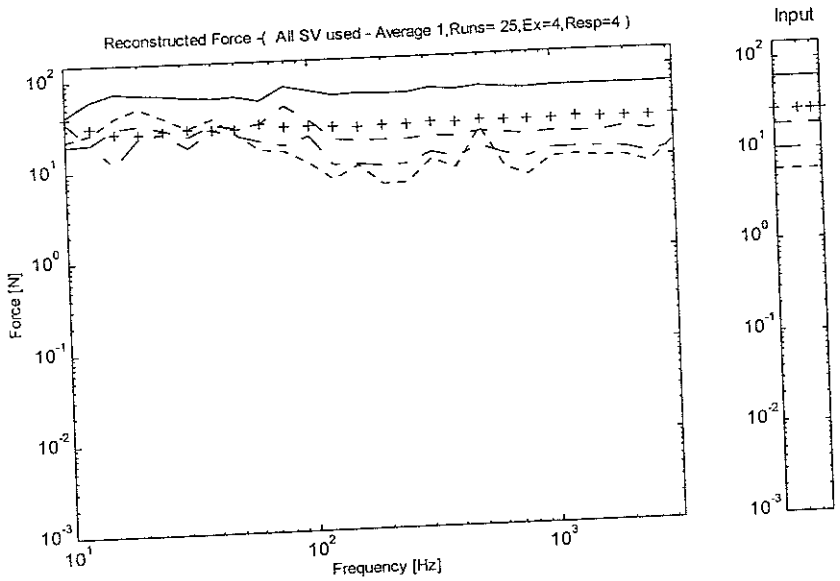


Figure 14. Reconstructed forces for 4 sources and 4 responses - All singular values used

Force1- - - -,Force2__ _ __,Force3 ++++,Force4 __ __ __,Sum of forces _____

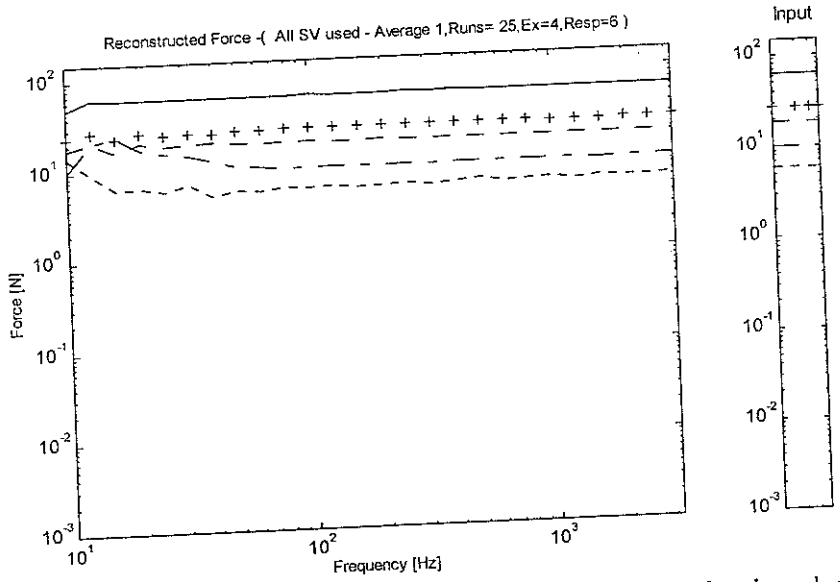


Figure 15. Reconstructed forces for 4 sources and 6 responses - All singular values used

Force1- - - -,Force2__ _ __,Force3 ++++,Force4 __ __ __, Sum of forces _____

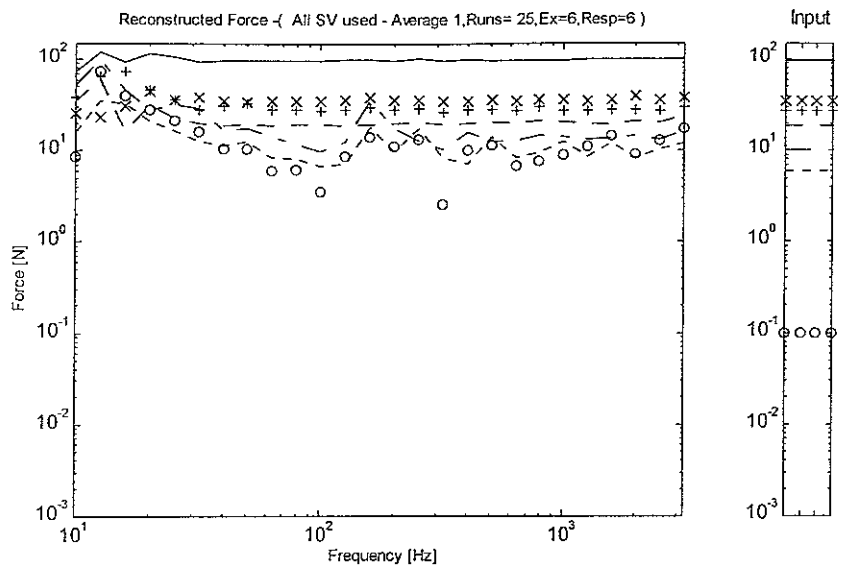


Figure 16. Reconstructed forces for 6 sources and 6 responses - All singular values used
Force1- - - -,Force2__ _ __,Force3 ++++,Force4 __ ___ __,Force5 xxxxxx,Force6 oooo,
Sum of forces _____

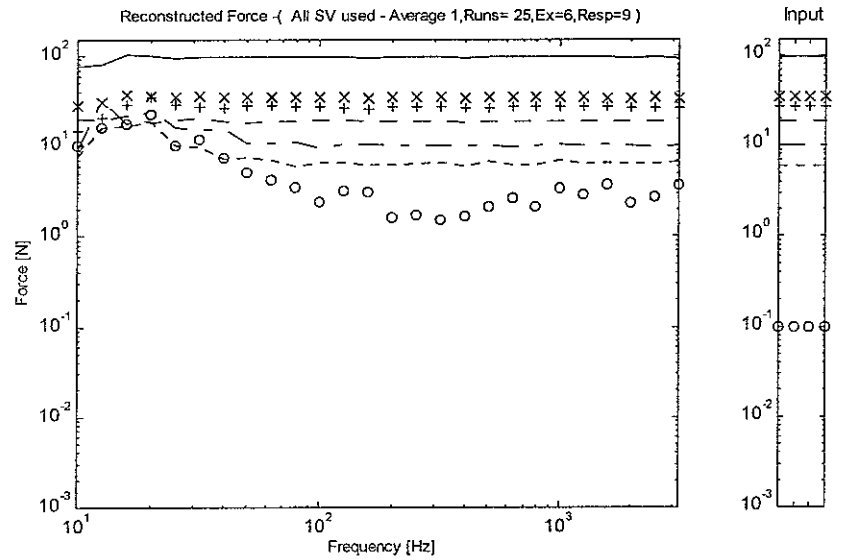


Figure 17. Reconstructed forces for 6 sources and 9 responses - All singular values used
Force1- - - -,Force2__ _ __,Force3 ++++,Force4 __ ___ __,Force5 xxxxxx,Force6 oooo,
Sum of forces _____

3.4 Velocity response at the receiver location

As the velocity response is flat in the frequency domain and the sound radiated can be directly related to it, it is used for the comparative study rather than acceleration. The predictions at the receiver location are as shown in Figures 18-21. These figures compare the actual response with the reconstructed response for various experiments, the upper figure showing the case with singular value rejection, the lower figure that for a full rank $\hat{A}$ matrix. Also shown is a range of ± standard deviation around the reconstructed spectrum. This range corresponds to the averaging performed over the operational response samples as shown in the flow chart. As the accuracy of force reconstruction affects the final prediction of response at the receiver location, the predictions are erratic for $n_r = n_e$ whether or not singular values are rejected. However the response improves at frequencies below 150 Hz as both n_r and n_e increase even if $n_r = n_e$, as shown in Figure 20. When $n_r > n_e$, the estimations improve. The improvement is considerable for the cases where all singular values are used.

Note that the response can be predicted more reliably than individual forces particularly for low frequencies (below 50 Hz) which is clear e.g. from comparing Figure 18 with Figures 10, 14. Good response predictions appear to be linked to reliable estimates of the sum of forces in this low frequency region.

The predictions at the receiver location are biased for $n_r = n_e$, that is the true result does not fall within the +/- standard deviation range. When $n_r > n_e$, this bias error is reduced, especially for the case with all singular values used. This reduction in error is significant for frequencies above 200 Hz compared to the singular value rejection method (compare Figures 18, 20 with 19, 21).

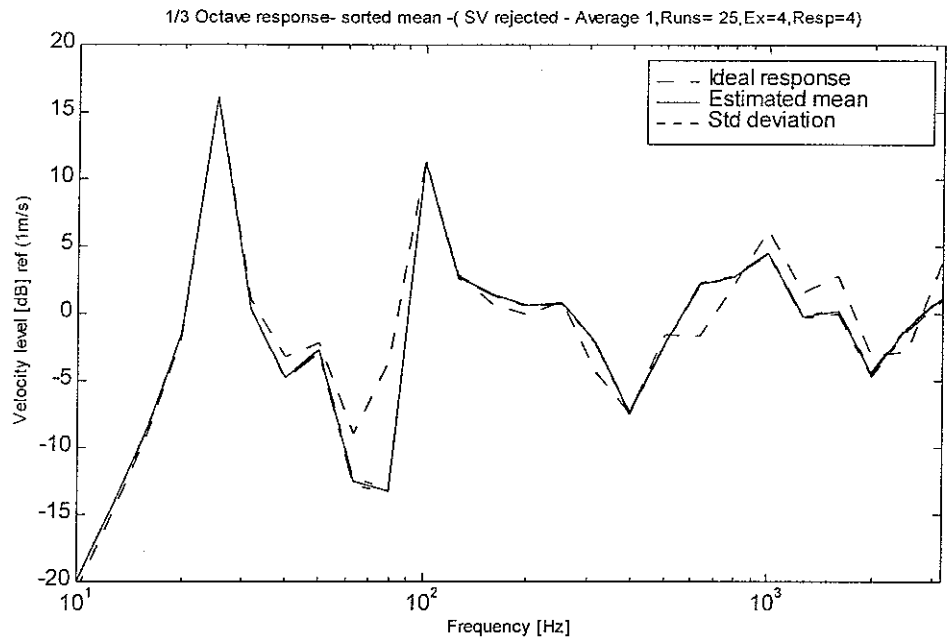


Figure 18a. 1/3 octave velocity response at receiver location due to 4 forces reconstructed from 4 responses - Singular values rejected using the error norm

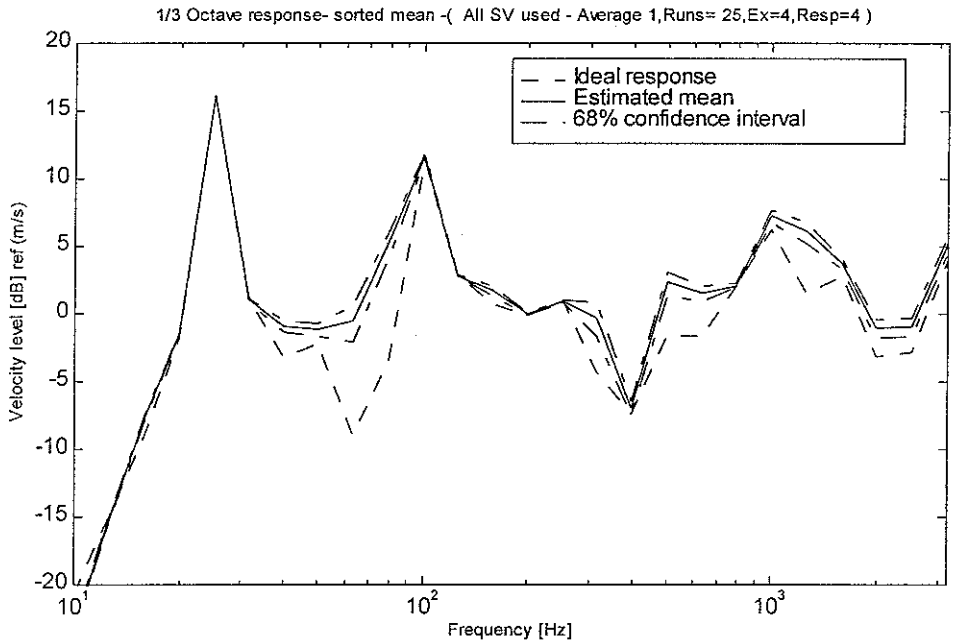


Figure 18b. 1/3 octave velocity response at receiver location due to 4 forces reconstructed from 4 responses - All singular values used

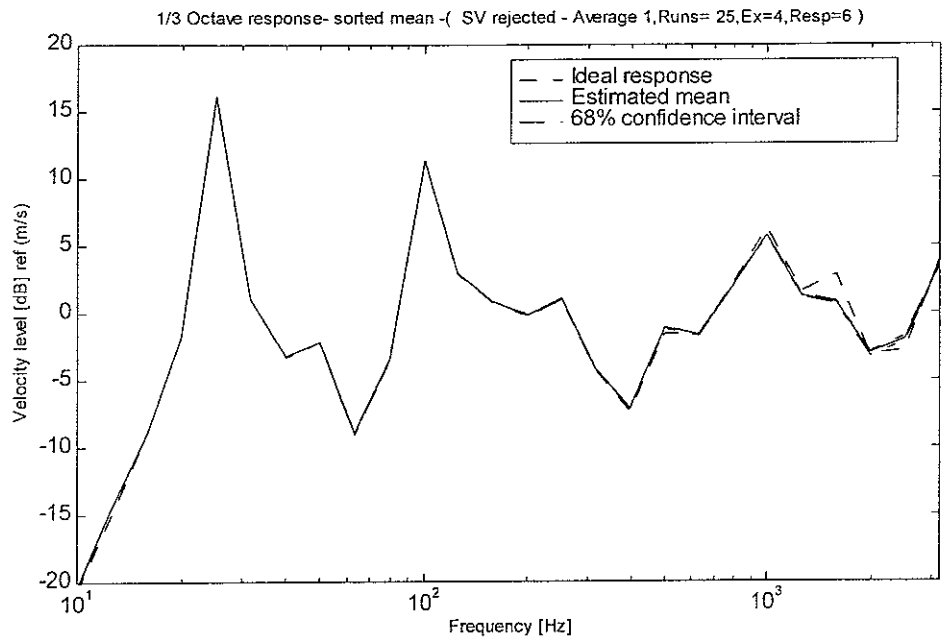


Figure 19a. 1/3 octave velocity response at receiver location due to 4 forces reconstructed from 6 responses - Singular values rejected using the error norm

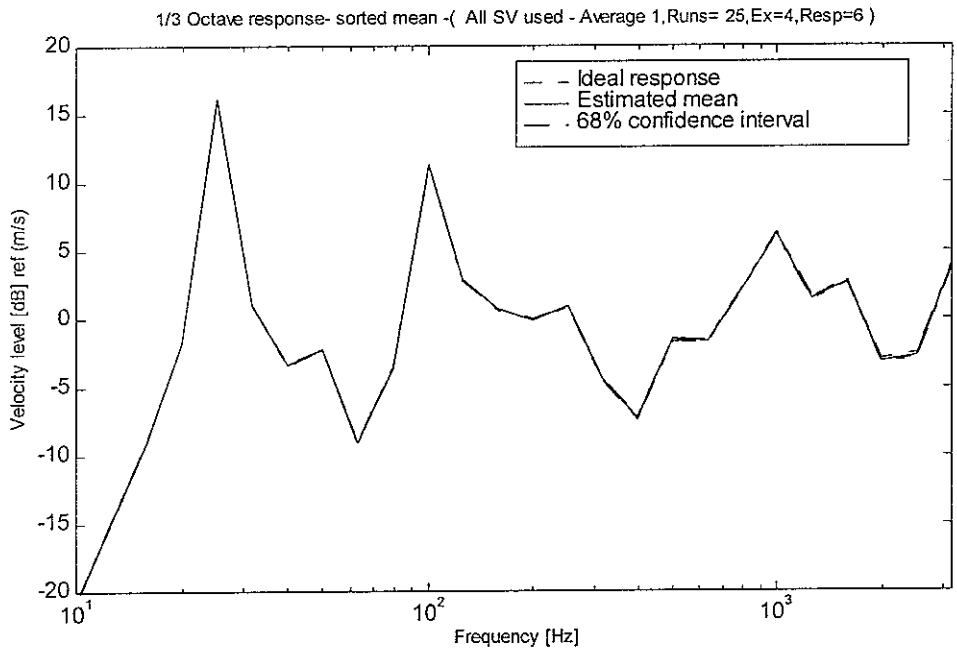


Figure 19b. 1/3 octave velocity response at receiver location due to 4 forces reconstructed from 6 responses - All singular values used

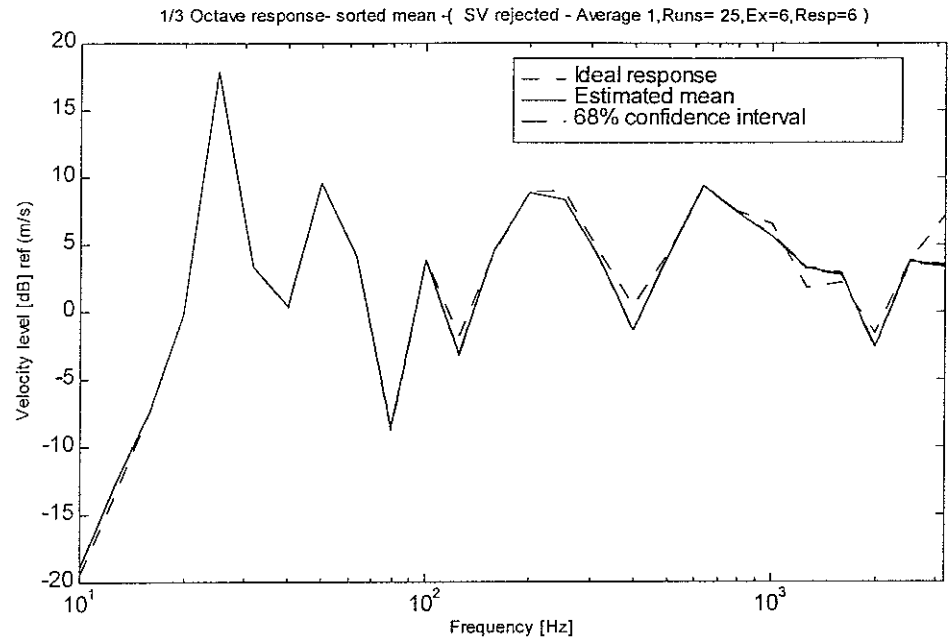


Figure 20a. 1/3 octave velocity response at receiver location due to 6 forces reconstructed from 6 responses - Singular values rejected using the error norm

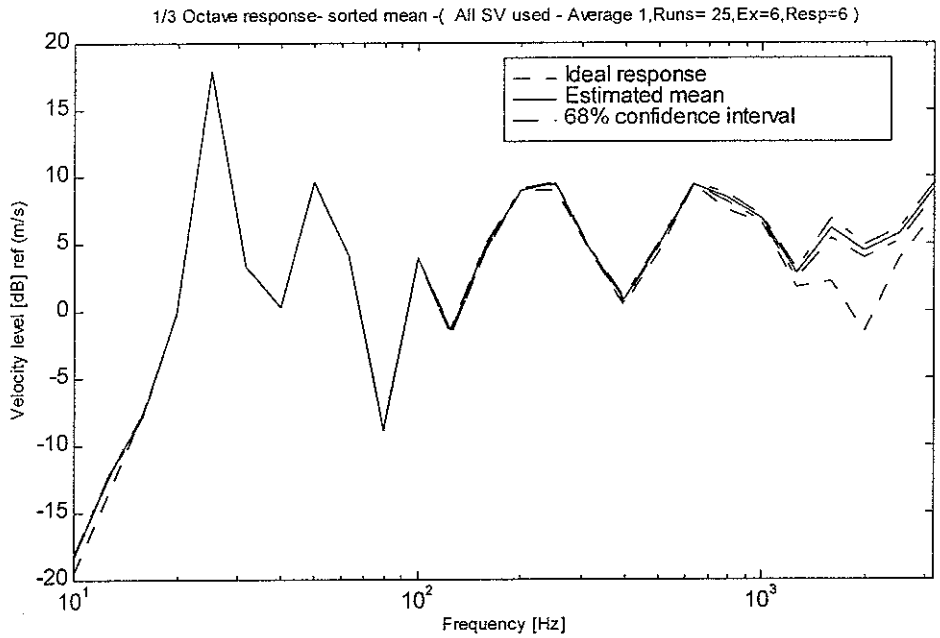


Figure 20b. 1/3 octave velocity response at receiver location due to 6 forces reconstructed from 6 responses - All singular values used

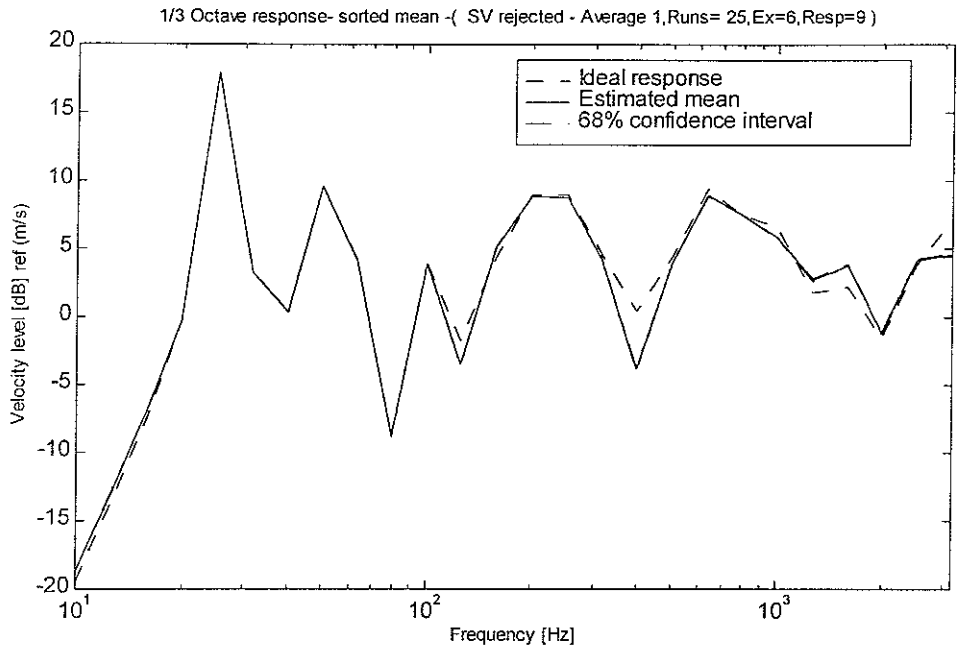


Figure 21a. 1/3 octave velocity response at receiver location due to 6 forces reconstructed from 9 responses - Singular values rejected using the error norm

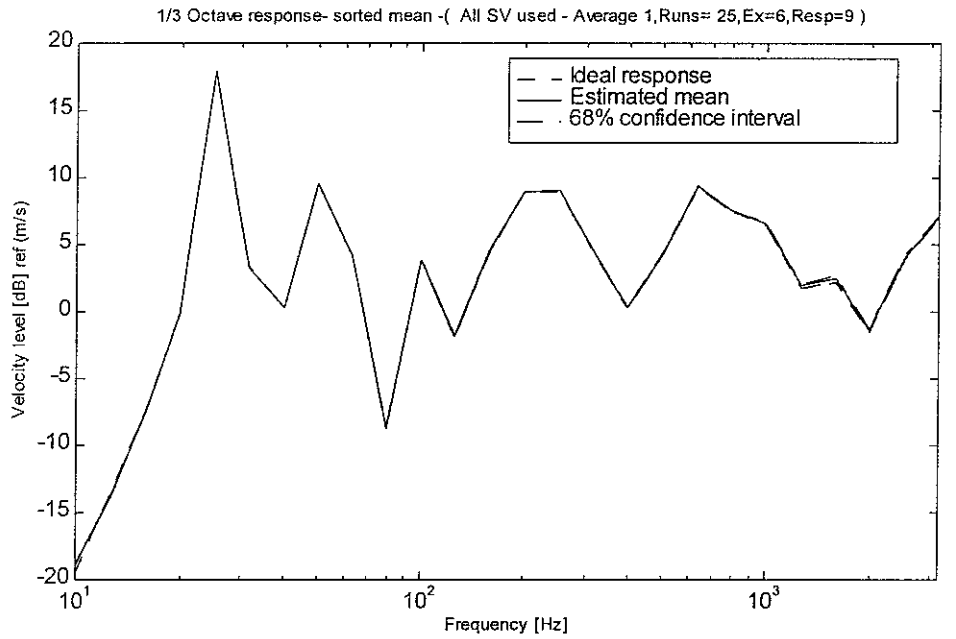


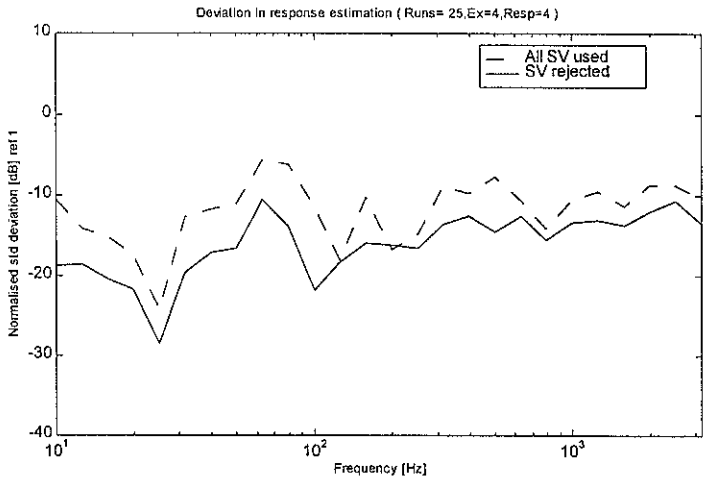
Figure 21b. 1/3 octave velocity response at receiver location due to 6 forces reconstructed from 9 responses - All singular values used

Figures 22a-d show the normalised standard deviation in the estimation of velocity response at the receiver location. This standard deviation corresponds to the averaging according to the flow chart given earlier. The standard deviation of predictions is low for both the cases. It is lower for the case where singular values are rejected based on the error norm.

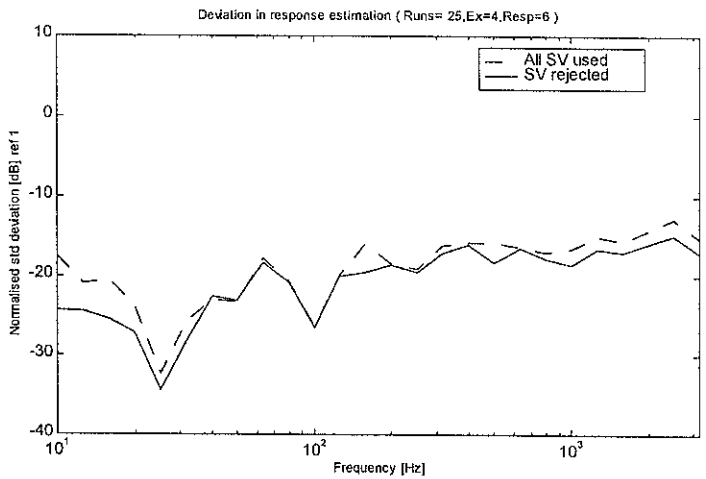
To find out the variation in the estimates for different confidence limits of accelerance measurement (Equation (7)) i.e. different values of α, a simulation is carried out for 68% confidence limit (i.e. α=1 instead of α=3). The prediction of response at the receiver for this case improves considerably at higher frequency as shown in Figure 23 compared to Figure 18a, as less singular values are rejected.

3.5 Summary

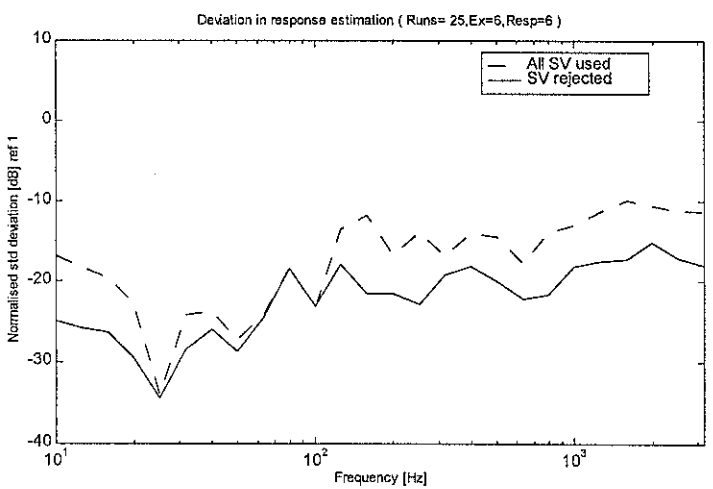
Based on the above results, it is observed that the estimations are reliable with over-determination when all singular values are used. When singular values are rejected based on the threshold from the error norm, the reconstructed forces differ from the actual ones at some frequencies and this leads to bias error in the response prediction. However these results are more robust with singular value rejection because the standard deviation is lower. The results improve when α=1 (i.e. one standard deviation in the measurement of accelerance) is used in singular value rejection, as less singular values are rejected.



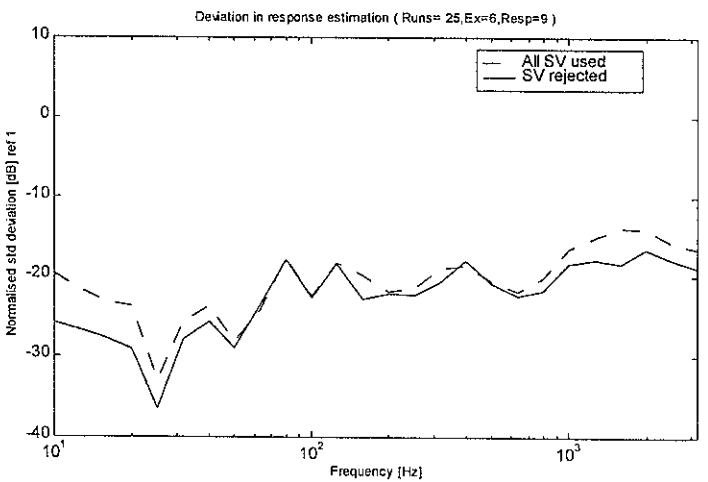
a. 4 sources and 4 responses



b. 4 sources and 6 responses



c. 6 sources and 6 responses



d. 6 sources and 9 responses

Figure 22. Comparison of normalised standard deviation for estimating velocity response at the receiver location

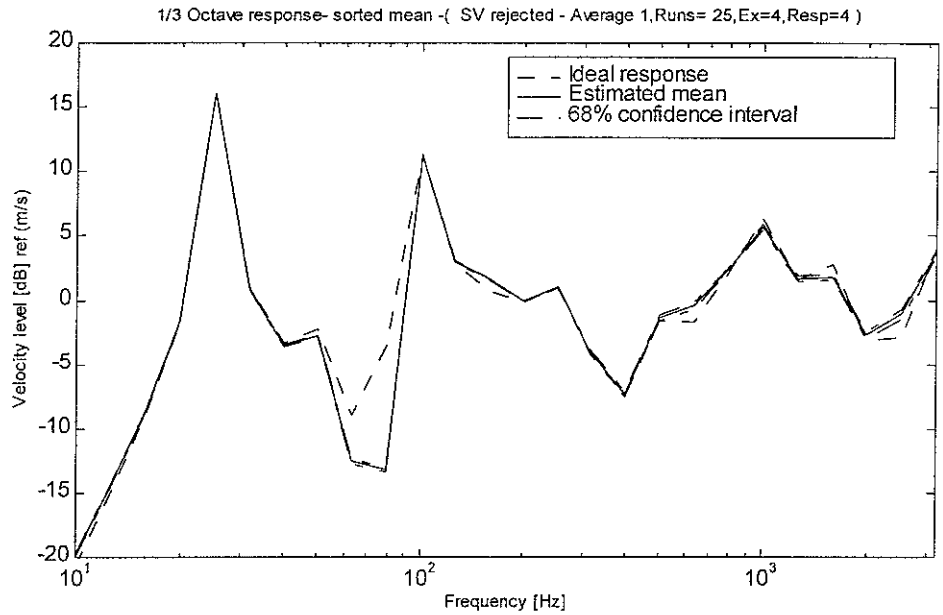
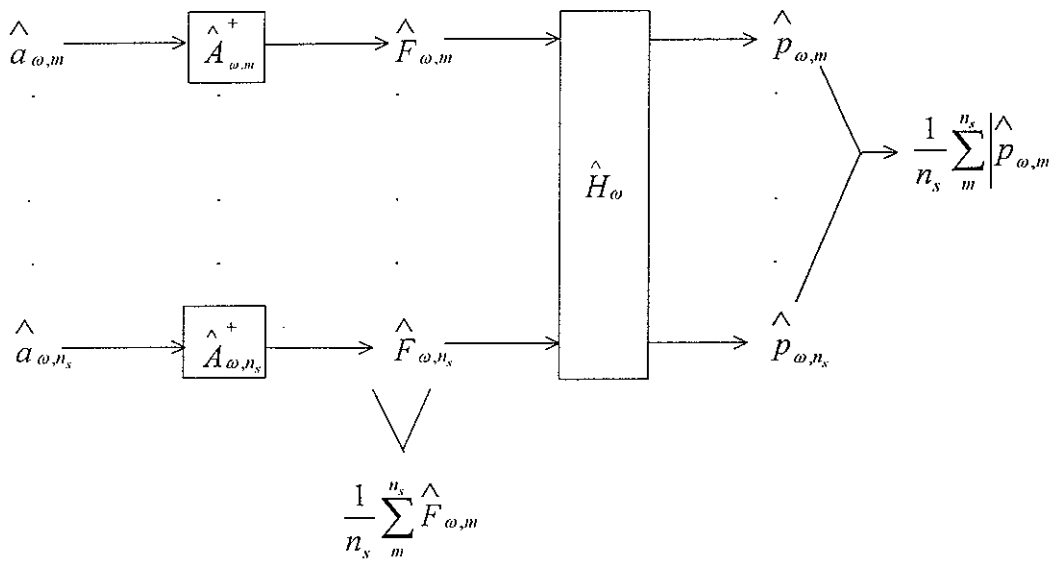


Figure 23. 1/3 octave velocity response at receiver location due to 4 forces reconstructed from 4 responses - Threshold decided based on +/- standard deviation. Singular values rejected using the error norm

4. EXPERIMENTS AND RESULTS USING RESAMPLED ACCELERANCE MATRIX

It has been observed in chapter 3 that the force estimation, obtained by inversion of the $\hat{A}$ matrix, is erroneous at frequencies with high condition numbers, due to the presence of small errors in the measurement of the accelerances. Hence the prediction of the response at the receiver location is also poor. However, if the small errors in the measurement are random and a new set of measurements is made for the accelerance every time the forces are estimated, the mean of the reconstructed forces could become more accurate, provided large numbers of samples are taken.

Then the flow chart for this process, which is modified from the earlier one, is shown below



The total number of samples required for estimating the accelerances for these calculations is ($n_{av}n_s$). This concept is investigated this chapter for a selection of the cases

considered above. Only cases with square $\hat{A}$ matrices are considered as these had the largest error in the previous section.

The trend for singular values and condition numbers remains the same for this group of experiments as in the previous chapter, and so these are not shown here.

4.1 Reconstructed forces

Figures 24-27 show the reconstructed forces for two experiments. Force reconstruction for the case with all singular values used, in general, is found to be improved considerably over the earlier cases in chapter 3 for both experiments (compare Figures 25 with 14 and 27 with 16). The effect is particularly significant at lower frequencies. This is not the case where singular values are rejected, it remains the same as in the earlier group of experiments (compare Figures 24 and 26 with 10 and 12).

4.2 Velocity response at the receiver location

The response predicted at the receiver location is also improved considerably when all singular values are used. It does not change much from the earlier group of experiments for the case where singular values are rejected. This is confirmed by Figures 28 and 29 which shows the results of two experiments. However, the band within which the mean can fall has widened for the case where all singular values are used, compared to that in Figures 18b and 20b.

Figures 30a,b show the standard deviation of the estimation of the velocity response at the receiver location. The standard deviation for the case where all singular values are used is very poor for $n_r = n_e$. This can be attributed to the high condition number and the presence of small random errors in the estimation of the accelerances.

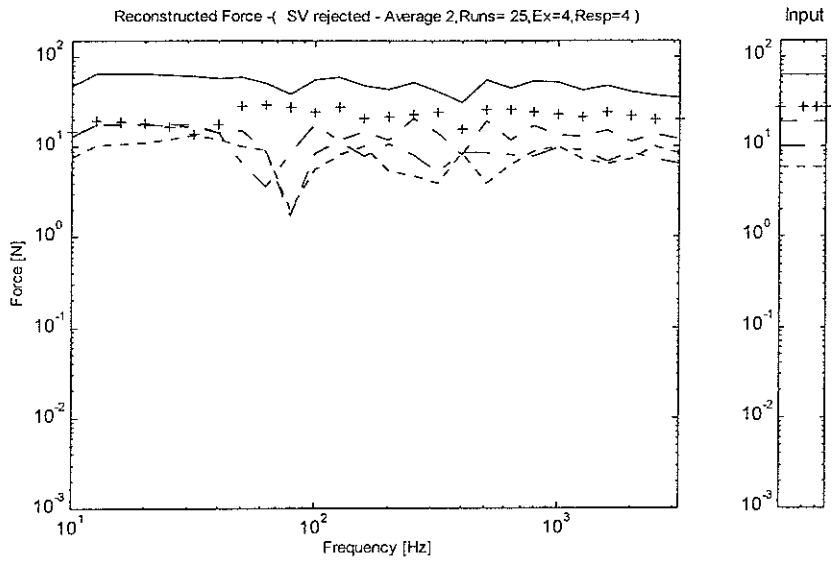


Figure 24. Reconstructed forces for 4 sources and 4 responses -Singular values eliminated using error norm. Force1- - - -,Force2__ _ __,Force3 ++++,Force4 __ __ __, sum of force _____

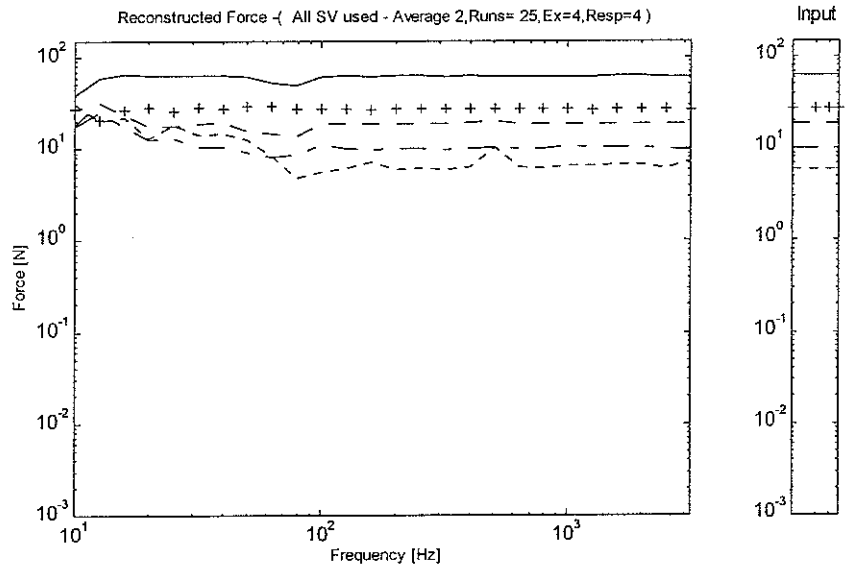


Figure 25. Reconstructed forces for 4 sources and 4 responses - All singular values used Force1- - - -,Force2__ _ __,Force3 ++++,Force4 __ __ __, sum of force _____

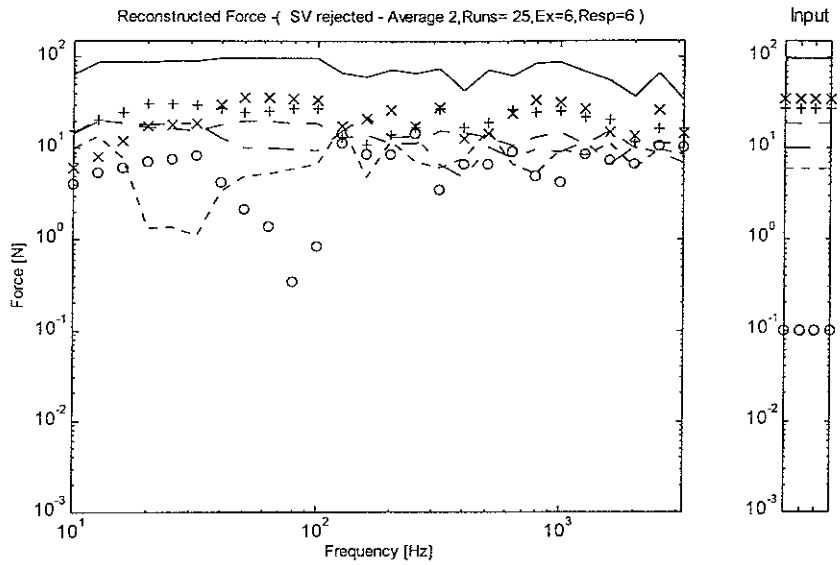


Figure 26. Reconstructed forces for 6 sources and 6 responses -Singular values eliminated using error norm. Force1- - - -,Force2__ _ __,Force3 ++++,Force4 __ __ __, Force5 xxxx,Force6 oooo, sum of force _____

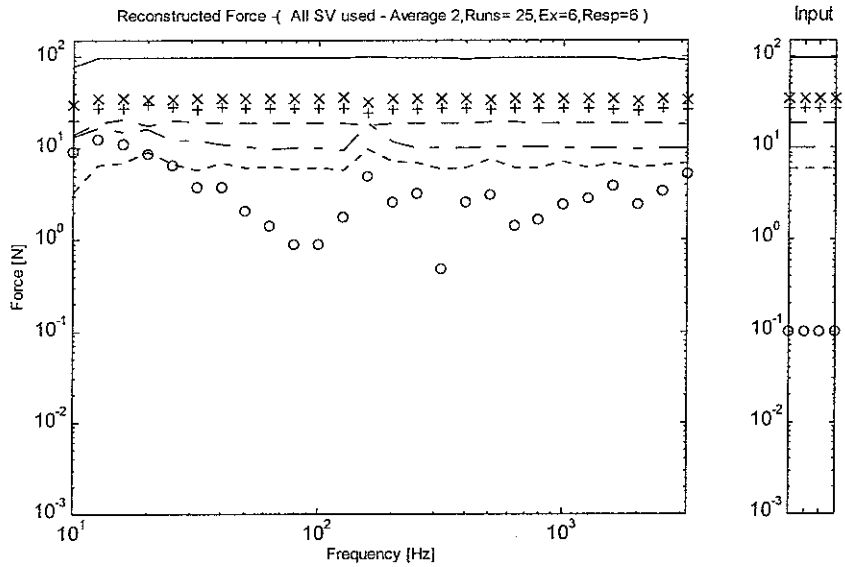


Figure 27. Reconstructed forces for 6 sources and 6 responses - All singular values used Force1- - - -,Force2__ _ __,Force3 ++++,Force4 __ __ __,Force5 xxxx,Force6 oooo, sum of force _____

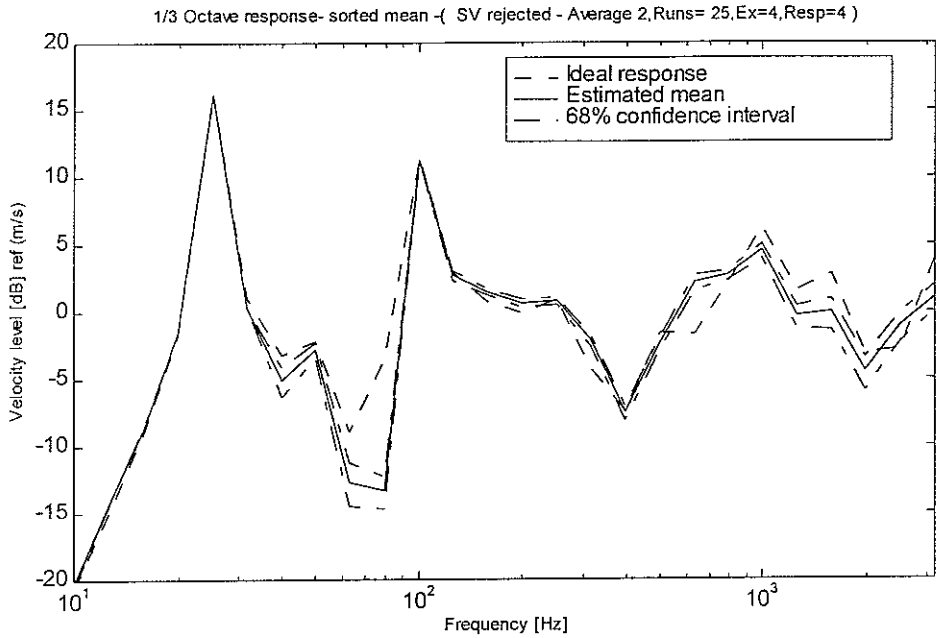


Figure 28a. 1/3 octave velocity response at receiver location due to 4 forces reconstructed from 4 responses - Singular values eliminated using error norm

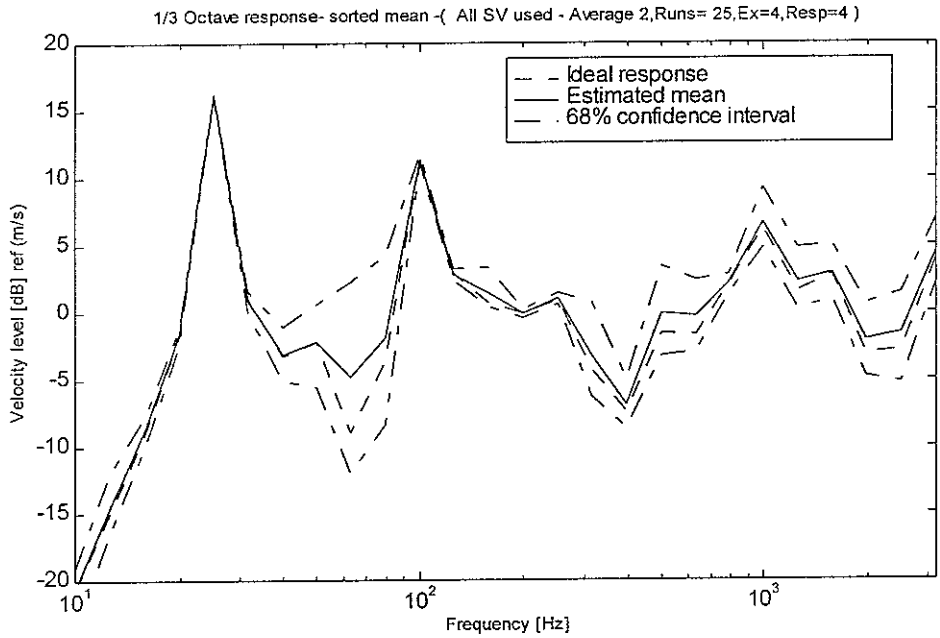


Figure 28b. 1/3 octave velocity response at receiver location due to 4 forces reconstructed from 4 responses - All singular values used

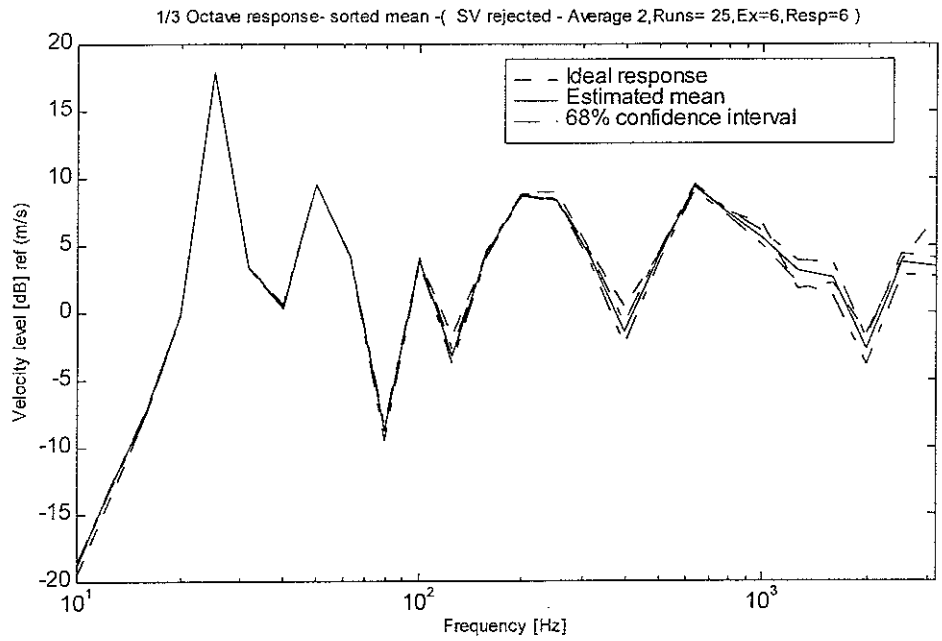


Figure 29a. 1/3 octave velocity response at receiver location due to 6 forces reconstructed from 6 responses - Singular values eliminated using error norm

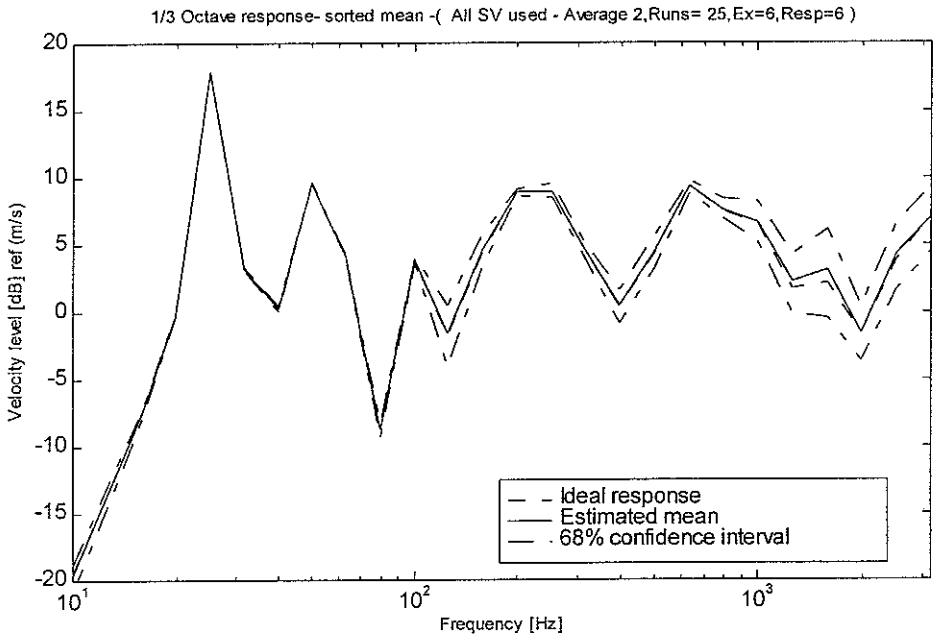
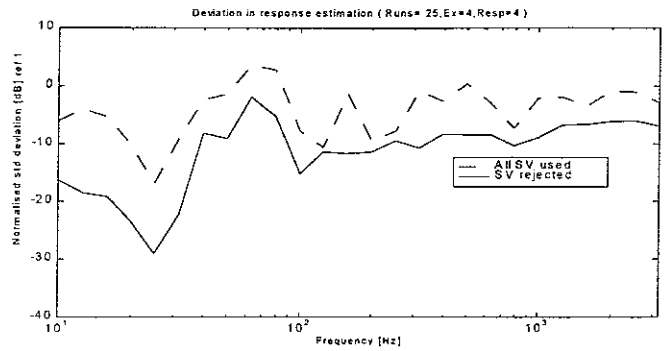
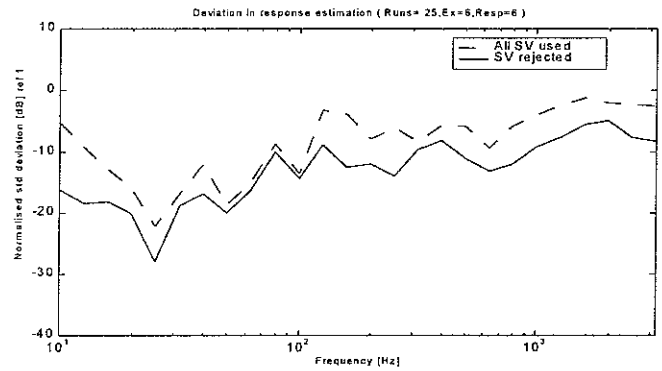


Figure 29b. 1/3 octave velocity response at receiver location due to 6 forces reconstructed from 6 responses - All singular values used



a. 4 forces, 4 response



b. 6 forces, 6 response

Figure 30. Comparison of normalised standard deviation for estimating velocity response at receiver location

4.3 Summary

The predictions based on the strategy used in this chapter have been shown to be reliable, even for $n_r=n_e$. The bias error is reduced considerably when all singular values are used. That is, the mean of the predictions is closer to the exact results. The effect of small random errors due to noise in the accelerance measurement is cancelled out in the averaging strategy proposed. However, a major disadvantage with this strategy is that the number of measurements required is enormous.

5. RESULTS USING PERTURBATION OF ACCELERANCE MATRIX

In the averaging strategy of chapter 4, where a resampled accelerance matrix is used, the results are encouraging. To bring it into practice could be difficult, however, since the number of measurements required is enormous. However, there could be a possibility of simulating the effect by perturbing the measured accelerance matrix $\hat{A}$ with random noise before pseudo-inversion. The magnitude of noise to be used for perturbation could be decided based on the coherence at the frequency of calculation, along with the magnitude of the accelerance at that frequency. However such an approach may be less reliable if $\hat{A}$ is a biased estimate of A.

The standard deviation of error in the estimation of accelerance is given by the relation in equation (7) with $\alpha = 1$

$$E_{ij} = \left[\frac{1-\gamma^2_{\omega,ij}}{2n_{av}\gamma^2_{\omega,ij}}\right]^{1/2} \left|\hat{A}_{\omega,ij}\right|$$

This error can be represented in a phasor diagram as shown in the Figure 31. The zone within the circle indicates schematically the possible values that the accelerance can take. These possible values can be used in the perturbation of the accelerance from the mean value.

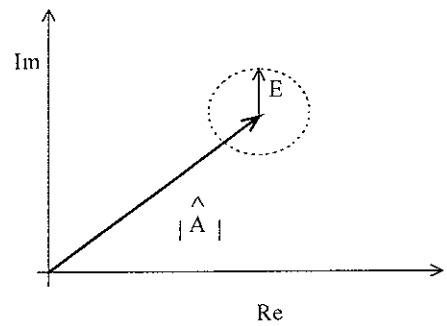


Figure 31. Phasor representation of accelerance along with the error

Then the modified accelerance matrix can be written as

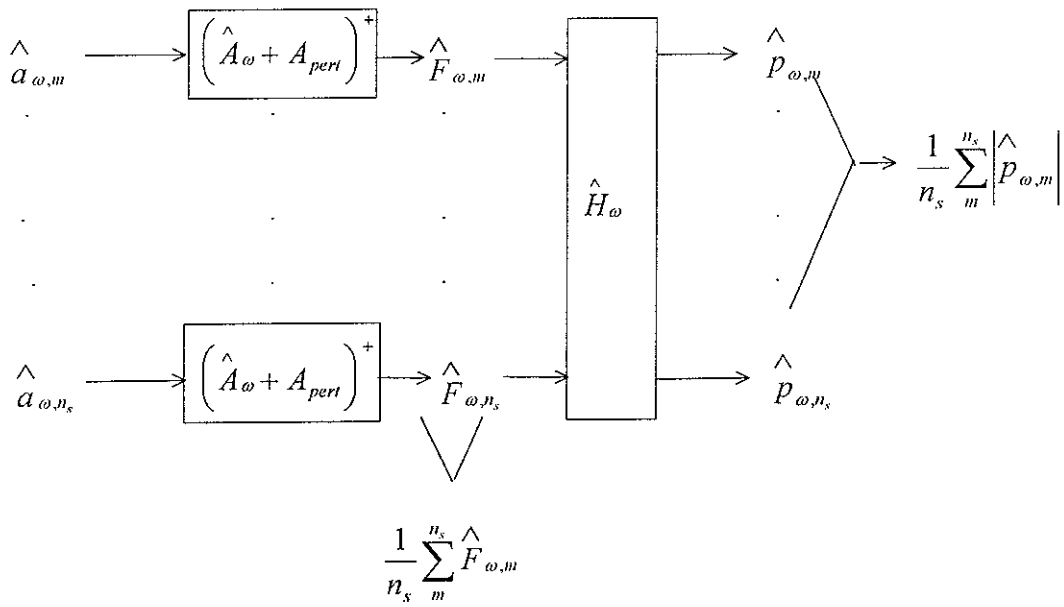
$$\hat{A}_{\omega k} = \hat{A}_{\omega} + A_{pert}$$

where $A_{pert} = N_k E(\cos(2\pi N) + j.\sin(2\pi N))$ is a perturbation matrix

N_k is a random number with variance 1 and zero mean (normally distributed)

N is a random number between 0 and 1 (uniformly distributed)

The flow chart for this strategy is shown below



The reconstructed forces and velocity response using this technique are shown in Figures 32 and 33 for the case where $n_r = n_e = 4$. The response estimated at the receiver location is far better than that estimated with a single accelerance matrix (compare Figure 33 with 18b). However, it does not replicate the strategy of a resampled accelerance as seen from Figure 28b. The bias error in the response is less compared to the single accelerance matrix strategy but more compared to the resampled accelerance strategy.

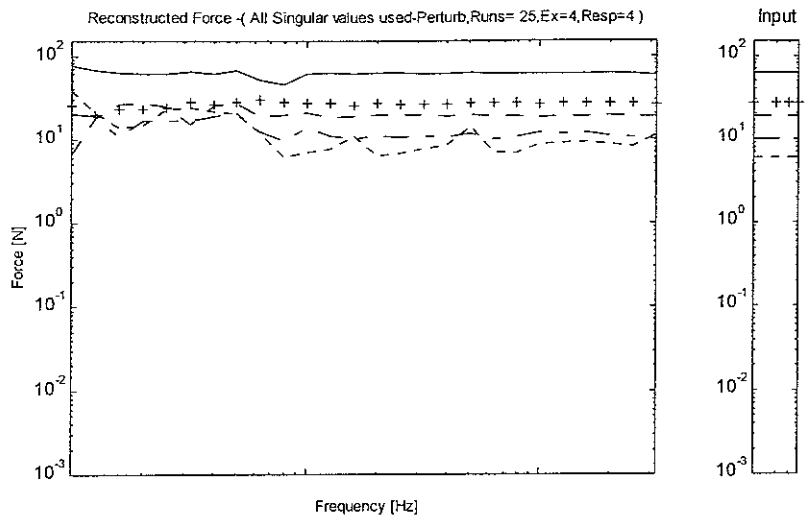


Figure 32. Reconstructed forces for 4 sources and 4 responses - All singular values used Force1- - - -,Force2__ _ __,Force3 ++++,Force4 __ __ __, Sum of forces_____ Perturbation technique.

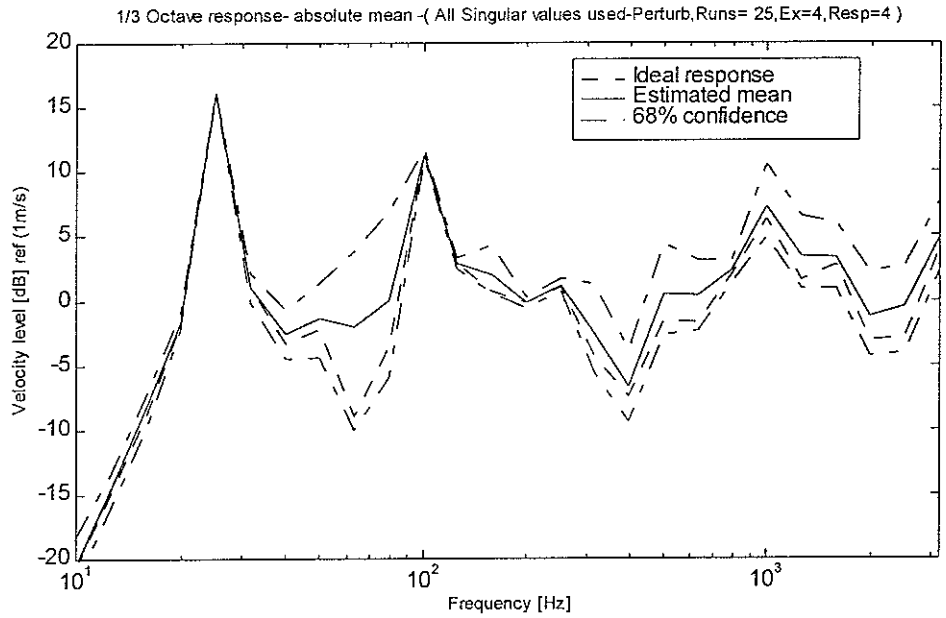


Figure 33. 1/3 octave velocity response at receiver location due to 4 forces reconstructed from 4 responses - All singular values used - Perturbation technique

The perturbation technique proposed in this chapter has been found to give results that are better than those with the single accelerance matrix strategy (chapter 3), but do not replicate the results of the strategy with the resampled accelerance matrix (chapter 4). Nonetheless this is a good compromise giving better results with no extra measurements.

6. RESULTS WITH AVERAGE FORMED BY CONSIDERING PHASE INFORMATION

In all the cases in earlier chapters, only the absolute value is considered while forming the average of the velocity response. The mean response formed with the absolute values could be larger than that formed with complex values since

$$\frac{1}{n_s}\sum_{m}^{n_s}\left|\hat{p}_{\omega,m}\right| \geq \frac{1}{n_s}\left|\sum_{m}^{n_s}\hat{p}_{\omega,m}\right|$$

Hence, to account for the effect of the phase information, in this chapter averaging is carried out including the phase information. Figures 34, 35 and 36 show the results for the case of $n_r=n_e=4$. When a single accelerance matrix is used (chapter 3), there is no significant improvement in the predictions (compare Figures 18 and 34). However, there is a remarkable improvement for the resampled accelerance case (chapter 4) with all singular values used (compare Figures 35b and 28b). The mean value based on averaging with phase gives a smaller estimation than that based on absolute values and is closer to the exact result. The difference between these two mean values is significant when the confidence interval is wide. Note that the confidence intervals are the same as in Figures 18 and 28, as phase can not be used in calculating the confidence interval. The mean is therefore not in the centre of the interval. A similar improvement is observed for the case where the accelerance matrix is perturbed (compare Figures 33 and 36).

These results indicate that it may be useful to form the average with the phase information included. However, this can only be used in practice if there is a phase reference, eg. a tacho signal, for the operational response.

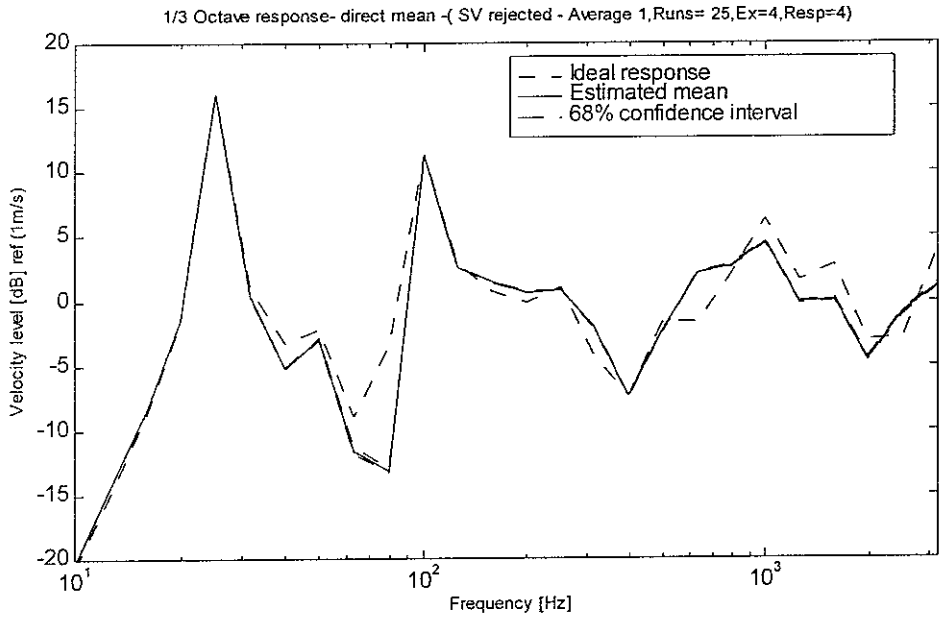


Figure 34a. 1/3 octave velocity response at receiver location due to 4 forces reconstructed from 4 responses - Averaging with phase information. Singular values rejected

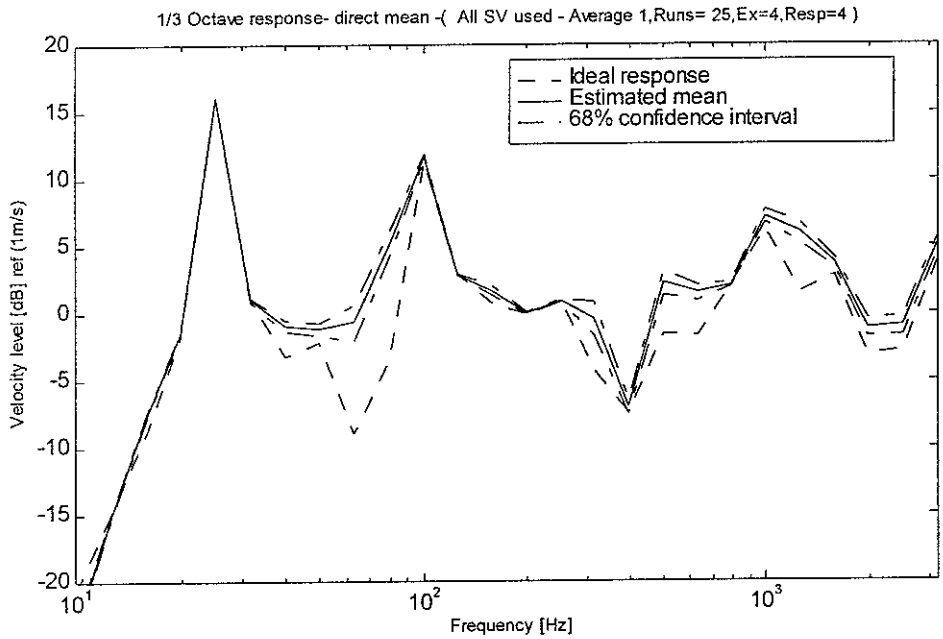


Figure 34b. 1/3 octave velocity response at receiver location due to 4 forces reconstructed from 4 responses - Averaging with phase information. All singular values used

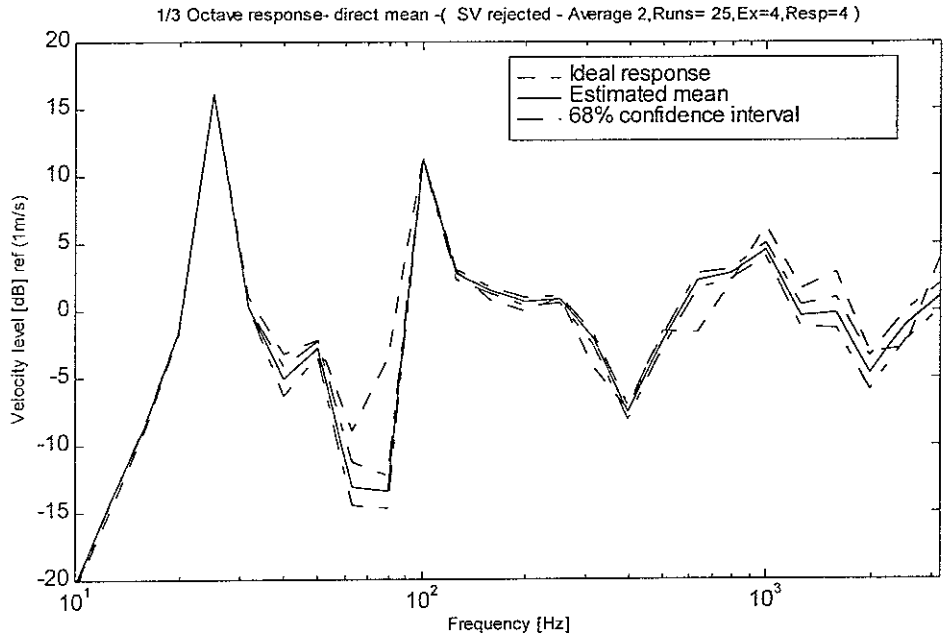


Figure 35a. 1/3 octave velocity response at receiver location due to 4 forces reconstructed from 4 responses - Averaging with phase information. Singular values rejected

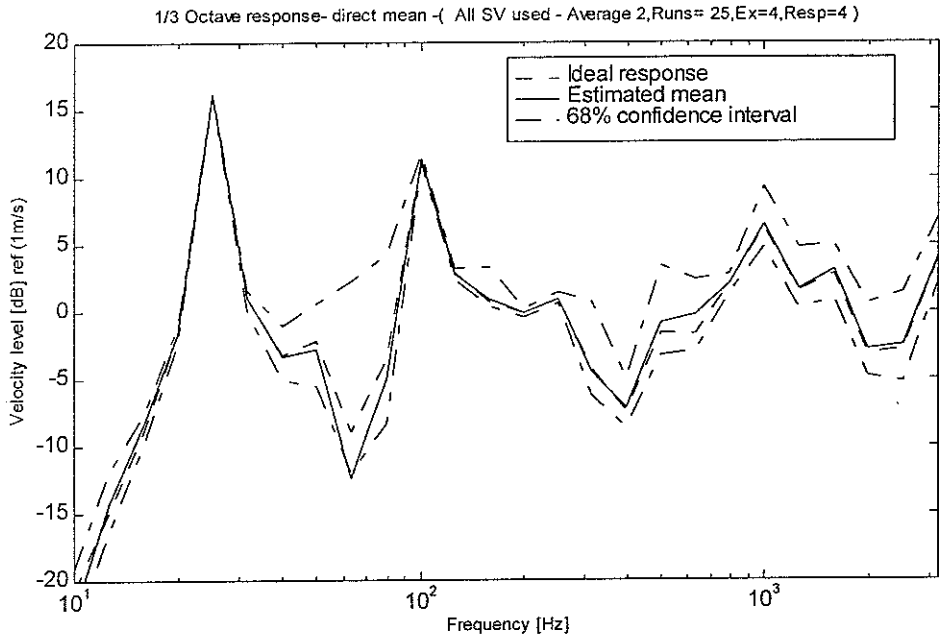


Figure 35b. 1/3 octave velocity response at receiver location due to 4 forces reconstructed from 4 responses - Averaging with phase information. All singular values used

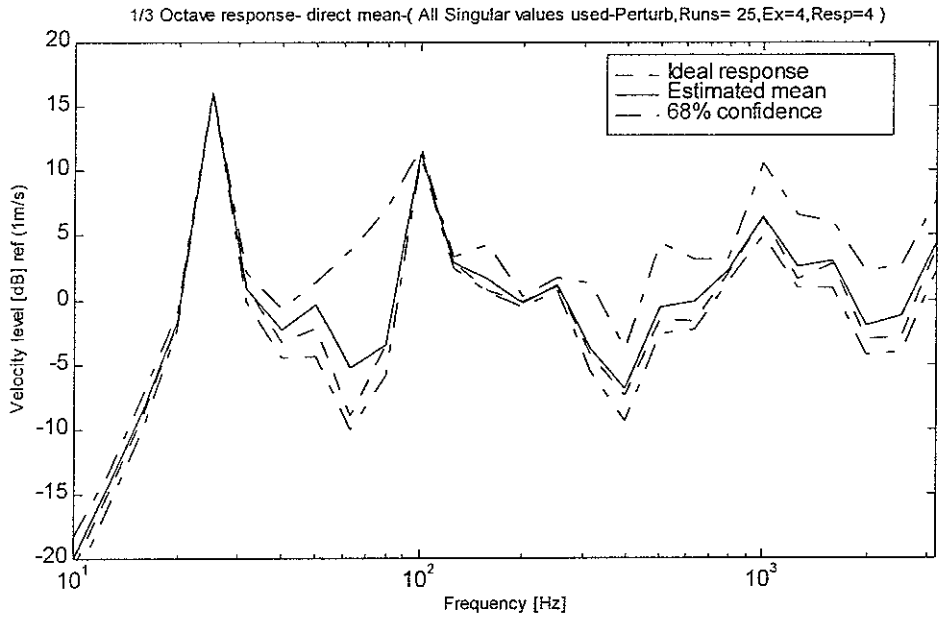
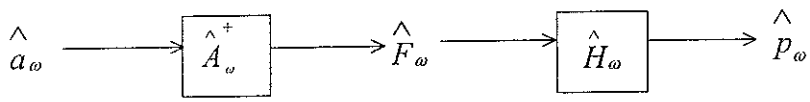


Figure 36. 1/3 octave velocity response at receiver location due to 4 forces reconstructed from 4 responses - Averaging with phase information. All singular values used - Perturbation technique

7 COMPARISON WITH ESTABLISHED TECHNIQUES

7.1 Introduction

In the previous sections, a number of averaging strategies have been considered for cases where all available samples of the operational response have been recorded. More usually [2] these are averaged at the time of the measurement and only the mean response is used to calculate the forces and receiver location response, the flow chart for the usual method is given below. In this section it is investigated whether such a procedure leads to a biased estimate of the response.



A refinement of the basic procedure, developed in reference [1], is to record a covariance matrix C_{aa} of the operational responses {a}. This requires less data to be stored than if all samples were recorded, whilst allowing some measure of the variability of the responses to be included in the reconstruction.

7.2 Confidence interval based on the covariance

The inverse force determination and response estimation are subject to several sources of error. Errors in the estimation of the accelerances and random errors in the measurement of operational accelerations are critical for response prediction. These errors are magnified by inversion if the accelerance matrix is ill-conditioned. Errors may also occur due to the omission of possible transfer paths. This usually leads to the over-estimation of forces related to the other paths which are considered in the analysis.

It is possible to estimate confidence intervals for the predicted response by using a covariance matrix to account for the errors [1]. The covariance matrix can be constructed

separately for the effects introduced in the operational acceleration measurement and the consequences of the pseudo-inversion in the response prediction at the receiver location.

The covariance matrix for a complex quantity can be expressed in terms of four components viz. ReRe, ReIm, ImIm and ImRe. For the measurement of operational accelerations $\hat{a}_i^j$, a series (j) of samples are taken at position i which are not all the same. Following [1], the covariance is evaluated as

$$C_{aa} = \sum_j \frac{\Delta s \Delta s^T}{n_s - 1}$$ and is a square matrix of size ($2n_r$ x $2n_r$)

where $\Delta s = \hat{a}_{ci}^j - \underline{\hat{a}_{ci}^j}$

$$\hat{a}_{ci}^j = \left[\mathrm{Re}(\hat{a}_i^j) \quad \mathrm{Im}(\hat{a}_i^j) \right]^T$$

$\underline{\hat{a}_{ci}^j}$ is a moving average of operational acceleration

The subscript c refers to the partitioning of complex vectors into real and imaginary parts. The covariance of estimated forces due to errors in the acceleration measurement can be estimated as

$$C_{ff} = \left[[\hat{\mathrm{A}}_c]^T [C_{aa}]^{-1} [\hat{\mathrm{A}}_c] \right]^{-1}$$ and is a square matrix of $2n_e$ x $2n_e$

where $$\hat{\mathrm{A}}_c = \begin{bmatrix} \mathrm{Re}(\hat{\mathrm{A}}) & \mathrm{Im}(\hat{\mathrm{A}}) \\ -\mathrm{Im}(\hat{\mathrm{A}}) & \mathrm{Re}(\hat{\mathrm{A}}) \end{bmatrix}$$ and is of size $2n_r$ x $2n_e$

while the covariance of the response prediction is given by

$$C_{path} = \left[[\hat{\mathrm{H}}][C_{ff}][\hat{\mathrm{H}}]^T \right]$$ and is of size 2 x 2 (for a single response location)

where $\hat{H}_c = \begin{bmatrix} Re(\hat{H}) & -Im(\hat{H}) \\ Im(\hat{H}) & Re(\hat{H}) \end{bmatrix}$ and is of size 2 x $2n_e$

Confidence intervals for the predicted response can be estimated using the Monte Carlo technique, which is implemented as follows

$$\{\hat{p}_{com}\} = \{\hat{p}_{oc}\} + [C_{path}]^{1/2}[N]$$

where N is a matrix of random numbers, of size 2 x 1 with zero mean and variance 1

$\{\hat{p}_{oc}\}$ is a vector of real and imaginary parts of the mean of the response derived by the relations given in chapter 2

$[C_{path}]^{1/2} = V D^{1/2} V^T$, V and D are eigen vector and eigen values of C_{path} respectively

Monte Carlo simulation is carried out over set of random number matrices. The mean of these estimations gives the expected response. The 68% confidence interval at each frequency is given by the values of 34% of the data above and below the median at each frequency.

The uncertainties in accelerance measurement and the pseudo-inversion process can be assessed by estimating the variance of the fit from the reconstructed operational accelerations and the actual measured accelerations. However, this estimation may not include the uncertainty due to some of the possible transfer paths being omitted.

The matrix of the covariance of the fit is constructed according to [1] as

$$[C_{aa}]_{fit} = \left[\begin{array}{ccc|ccc} \ddots & & 0 & \ddots & & 0 \\ & Var_{\mathrm{Re,Re},fit} & & & Cov_{\mathrm{Re,Im},fit} & \\ 0 & & \ddots & 0 & & \ddots \\ \hline \ddots & & 0 & \ddots & & 0 \\ & Cov_{\mathrm{Im,Re},fit} & & & Var_{\mathrm{Im,Im},fit} & \\ 0 & & \ddots & 0 & & \ddots \end{array}\right]$$

where

$$Var_{\mathrm{Re,Re},fit} = \frac{\sum \{\mathrm{Re}(a_i - a_{i,fit})\}^2}{(n_r - r)}$$

$$Var_{\mathrm{Im,Im},fit} = \frac{\sum \{\mathrm{Im}(a_i - a_{i,fit})\}^2}{(n_r - r)}$$

$$Cov_{\mathrm{Re,Im},fit} = Cov_{\mathrm{Im,Re},fit} = \frac{\sum \{\mathrm{Re}(a_i - a_{i,fit})\,\mathrm{Im}(a_i - a_{i,fit})\}}{(n_r - r)}$$

$$a_{i,fit} = \left[\hat{\mathrm{A}}_\omega\right]\left\{\hat{\mathrm{F}}_\omega\right\}$$

and $n_r \geq n_e \geq r$ in which r is the rank of the matrix used for the pseudo-inversion (i.e the number of singular values retained).

The covariance of the fit and the covariance matrix for acceleration measurements can be combined to form a total covariance. This total covariance can be used in estimating the confidence intervals, as discussed above.

7.3 Results

Figure 37 shows the velocity response for the 4 x 6 case where singular values are rejected based on the error norm, while Figure 38 shows the corresponding reconstructed forces. These are calculated based on the mean 'measured' operational responses. The

velocity response predicted by this calculation contains a bias error, similar to that found in Figure 20a.

The results of the confidence interval calculation for the force and the response at the receiver location, for a case where only C_{aa} is considered, are shown in Figures 39 and 40. The 68% confidence limit, estimated by using C_{aa} alone, does not envelope the actual response at all frequencies, but it is comparable to what is obtained in chapter 3 when using the single accelerance matrix (compare Figures 40 and 20a).

Figures 41 and 42 show the case where $C_{aa,fit}$ alone is considered. When the confidence interval is estimated based only on the $C_{aa,fit}$ (Covariance of the fit) the interval includes the actual response at most of the frequencies (Figure 42). The error contribution in the response estimation at the receiver location from operational response measurements (C_{aa}) is less compared to pseudo-inversion ($C_{aa,fit}$) which can be seen from the width of the confidence intervals (compare Figures 40 and 42).

For the total covariance the results are shown in Figures 43 and 44. The confidence interval calculated based on both C_{aa} and $C_{aa,fit}$ should include all possible errors in the estimation of response at the receiver location. It is therefore encouraging that the exact response lies within this confidence interval, see Figure 44. The confidence interval calculated based on this approach is larger than that obtained by the resampled accelerance approach given in Figure 45 for comparison.

It should be noted that in the last two cases the confidence intervals for individual forces are wide at low frequencies, but the confidence interval for the response is considerably smaller.

The major advantage with this approach for calculating confidence intervals compared to that considered earlier is that the amount of data to be stored is much less and it is therefore very practical.

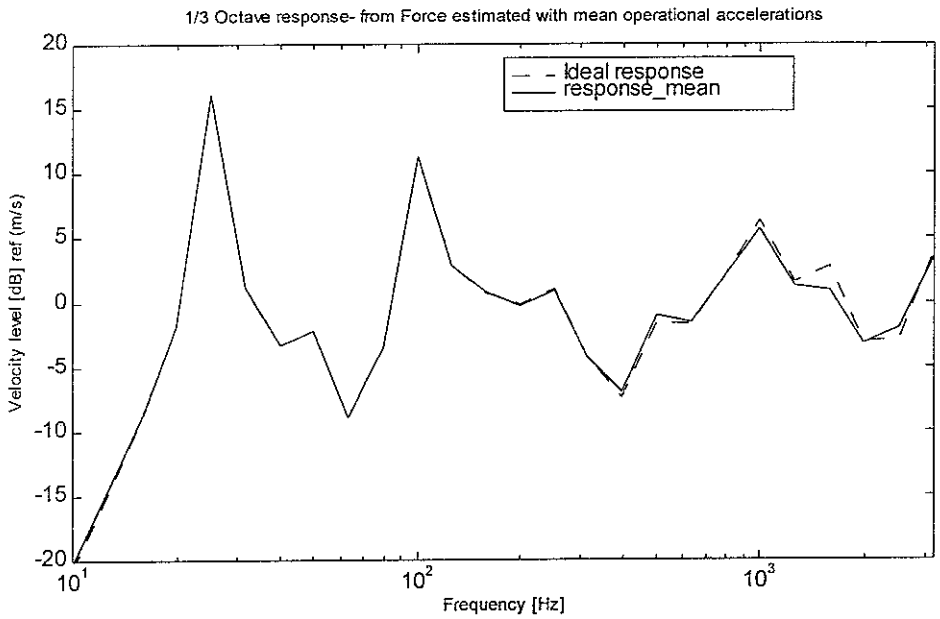


Figure 37. Velocity response calculated from reconstructed forces using mean operational accelerations - 4 forces and 6 responses

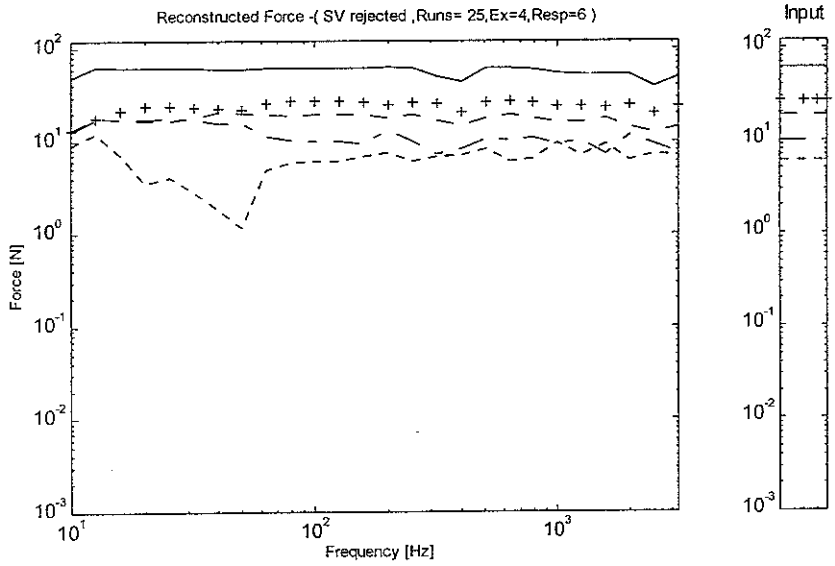
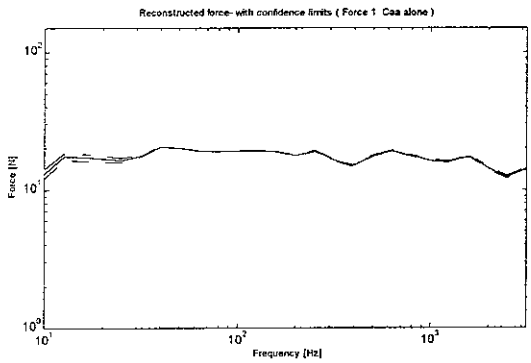
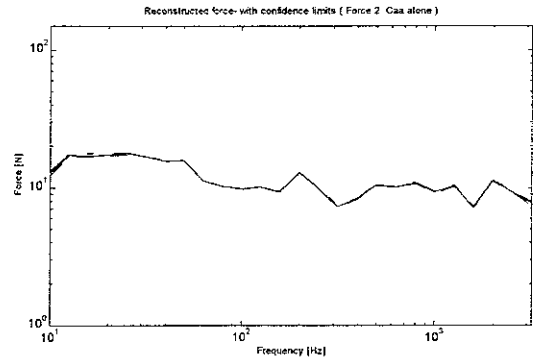


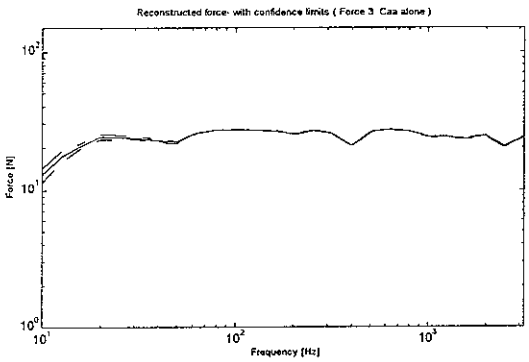
Figure 38. Reconstructed forces for 4 sources and 6 responses- Singular values eliminated using error norm. Force1- - - -,Force2__ _ __,Force3 ++++,Force4 __ __ __, sum of forces _____



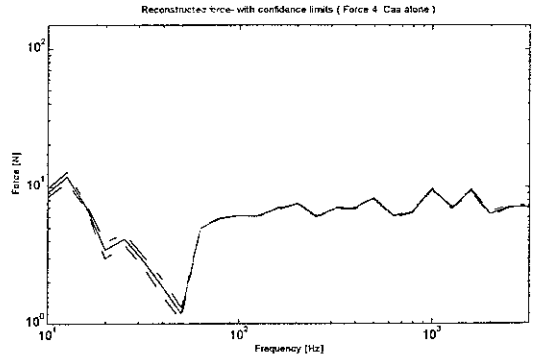
a. Force 1



b. Force 2



c. Force 3



d. Force 4

Figure 39. Reconstructed forces along with confidence limits based on C_{aa} alone

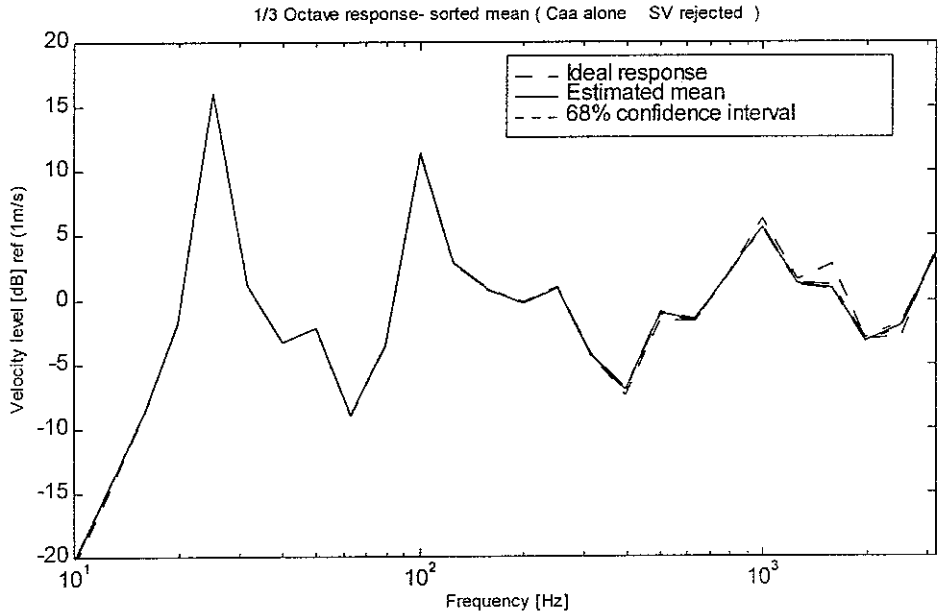
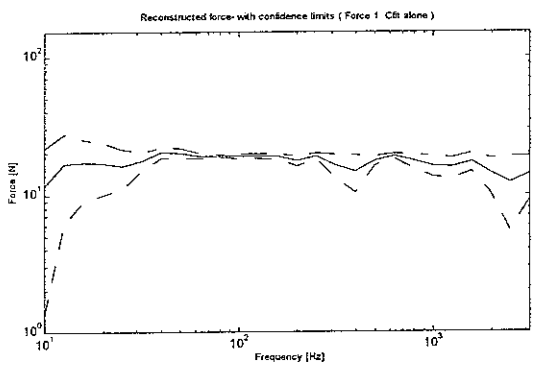
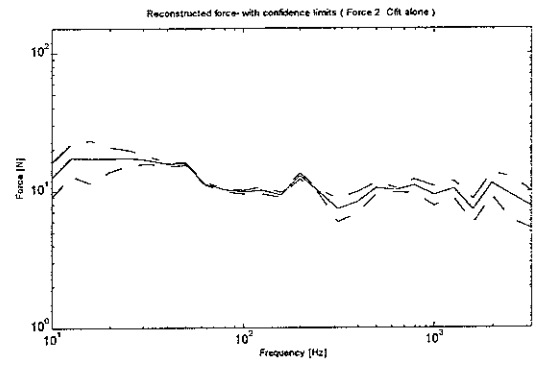


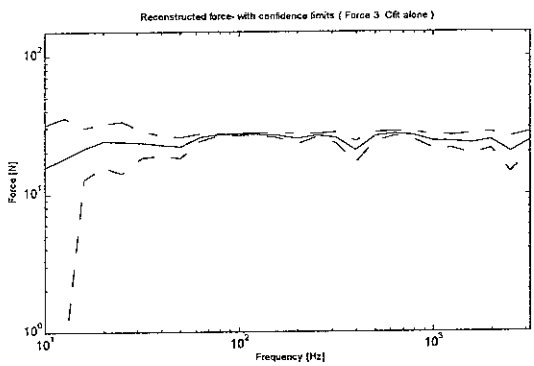
Figure 40. 1/3 octave response with confidence interval based on C_{aa} alone



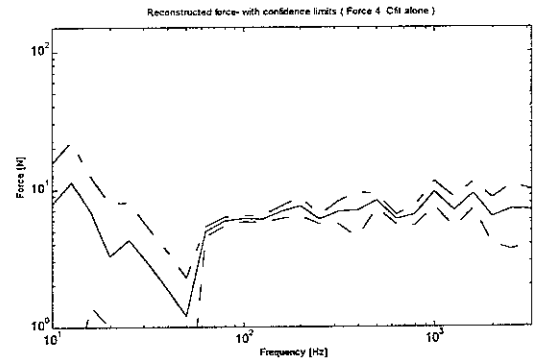
a. Force 1



b. Force 2



c. Force 3



d. Force 4

Figure 41. Reconstructed forces along with confidence limits based on C_{fit} alone

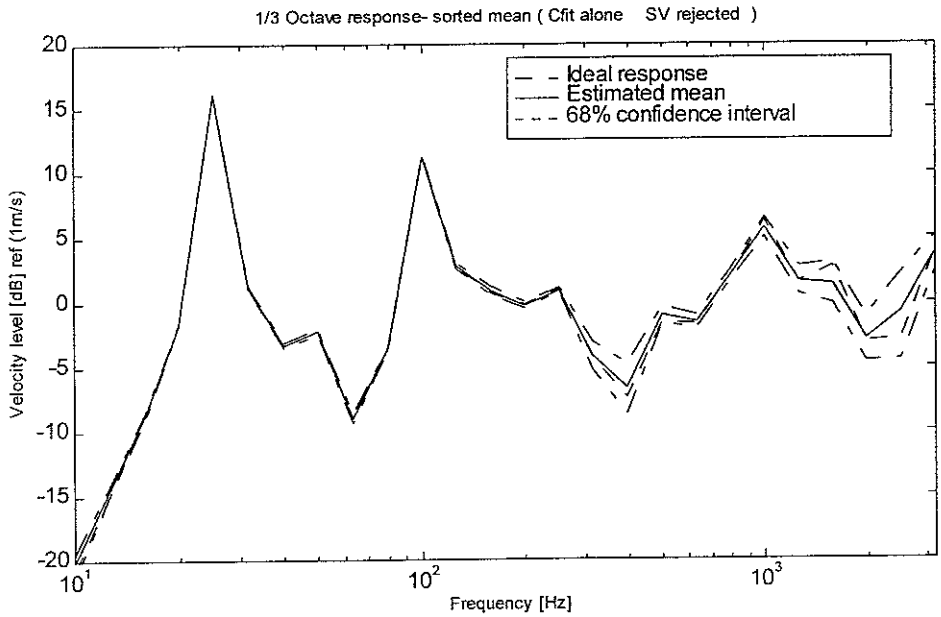
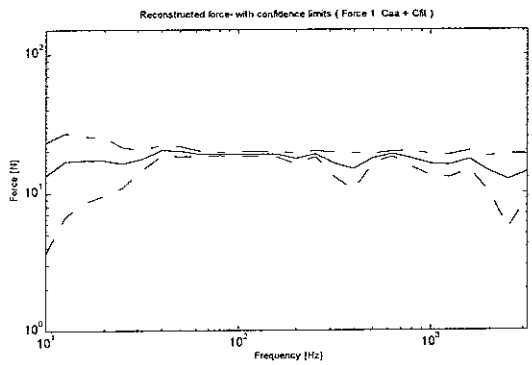
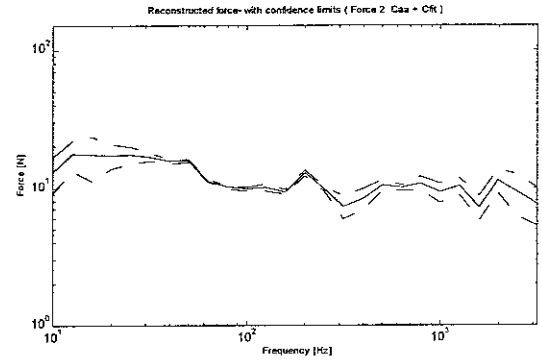


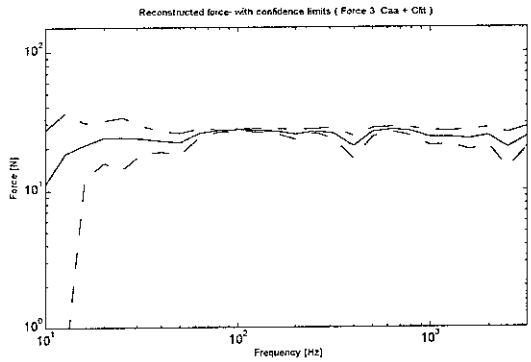
Figure 42. 1/3 octave response with confidence interval based on C_{fit} alone



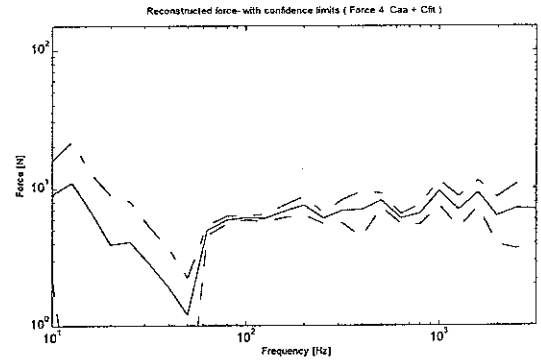
a. Force 1



b. Force 2



e. Force 3



d. Force 4

Figure 43. Reconstructed forces along with confidence limits based on C_{aa} & C_{fit} combined

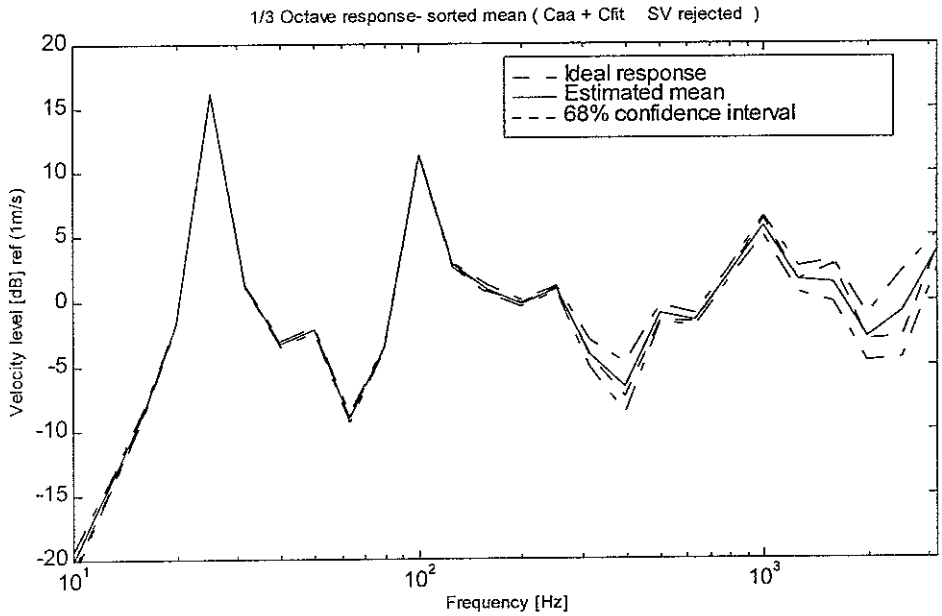


Figure 44. 1/3 octave response with confidence interval with C_{aa} & C_{fit} combined

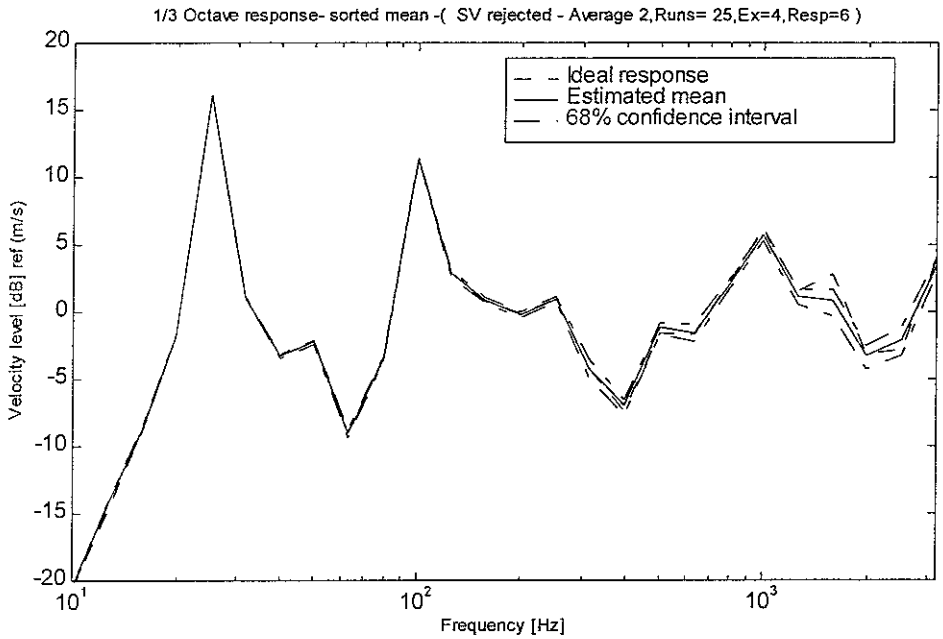


Figure 45. 1/3 octave response with confidence interval for 4 forces and 6 responses. Singular values are rejected based on the error norm - Resampled accelerance method

8. CONCLUSIONS

An example of indirect force determination and transfer path analysis has been carried out using numerical simulations. The effects of either using a full rank accelerance matrix or eliminating singular values based on the error norm of the accelerance matrix have been investigated. Different strategies for averaging are implemented to improve the predictions. Some of the findings are listed below.

1. At low frequencies near resonances the response prediction at the receiver location does not depend on the condition number or the errors in the measurement. Even though the individual force reconstruction is very poor, the response prediction is quite good, since the summation of the forces is estimated fairly reliably. The location of the force and response positions does not affect the prediction at these frequencies either. Even for a square accelerance matrix, as the number of forces increases the response is predicted correctly over a wider frequency band. In general it can be concluded that the response prediction at lower frequencies improves with an increase in the number of forces and responses even if the system is not over-determined.

2. When singular values are eliminated in the inversion process, the predictions are poor at higher frequencies. In this case, the threshold for rejection is based on the error norm calculated using ± three standard deviations of the estimated accelerance, according to reference [1]. However, the high frequency performance improves considerably when ± one standard deviation is taken for the error norm estimation. In any case, this does not affect the performance at low frequencies. This process of using the threshold based on the error norm may lead to under-

estimation of the response when the noise floor is high and consequently the number of singular values rejected increases.

3. By resampling the accelerance for each sample of response data, very good results are found even if the error in determining the accelerance is large and random. When using such a procedure, the system need not be over-determined for reliable results and no other variable is involved in the process. This process, however, would need an enormous amount of measured data, which makes it impractical.

4. If the average of estimated responses is calculated including phase information, the results become more reliable and repeatable. It is therefore worthwhile to include phase information in the averaging process, provided a suitable phase reference can be obtained in measurements, eg. a tacho signal. On the other hand, the confidence intervals have to be determined on the basis of the magnitudes only and the mean no longer lies in the centre of these intervals.

5. A perturbation technique for the accelerance matrix has been implemented to replicate the strategy of the resampled accelerance matrix with less experimental effort. This technique is found to give very good results in the simulations although it may not give such good estimates if the accelerance measurement is biased.

6. When only the mean operational response is used to calculate the forces and ultimately receiver location response, it has been shown that using a covariance matrix to obtain an estimate of uncertainties in the predictions at the receiver location, this is similar to that obtained by averaging over predictions for each sample of measured operational responses. This can be achieved with much less

measurement effort, but does not resolve possible bias errors in the mean estimates.

9. FURTHER WORK

9.1 Diagnostic transfer path analysis

The above conclusions rely on the validity of the noise models used in the simulations. A noise model has been used and this needs to be refined to make it more realistic. Means have to be found out to create a proper mathematical model of these noise sources. This has to be compared and correlated with experimental results for validation. It would be more realistic to implement different techniques of improving force estimation once a realistic model of noise is formulated.

To cover the whole domain of diagnosis the technique has to incorporate sound quality analysis as well. Since the narrow band predictions are not so reliable as the 1/3 octave band results given here it could be a challenge to develop reliable techniques for force reconstruction and eventually the response at the receiver location. Other regularization techniques exist as well as singular value rejection based on the error norm which can be further explored to determine and improve the precision of narrow band predictions.

9.2 Predictive transfer path analysis

In order to solve structure-borne noise problems in built-up structures, the ultimate aim is to develop reliable predictive models. These could be based on transfer path analysis. The importance of this is shown in simulations where if one path contribution is reduced the contribution from other paths may increase and hence the overall situation may not be improved. This leads to complications if a problem is tackled at an advanced stage of product development. To counter this problem effectively, a hybrid model combining FE/BE and experimental models [2] could be used in the prediction of contributions from different transfer paths before the prototype is built. Over and above the restriction on the number of elements, type of element and the modelling damping, there would be an

additional effect of numerical errors along with the already existing experimental errors. It would be interesting to explore whether modal and multi-modal approaches along with experimental models can be combined to represent a built-up structure so as to predict the contributions from different paths. The question would be whether similar techniques as in experimental analysis can be used to take care of these errors in estimation due to numerical analysis. The accuracy of predictions at the receiver location in a narrow band representation could decide whether sound quality analysis can be included in this approach. This is essential for optimal solution of the problem.

It is essential that the simulations to be carried out are on realistic structures (built-up structures) which have a combination of well defined and some not so well defined paths. The proposed work described above can be extended further to simulate conditions such that the following can be studied;

a. Path with no power flow

b. Effect of different damping models like proportional, linear etc.

c. Modal and multi-modal structural behaviour

Differences between the low frequency, mid-frequency and high frequency regions affect the transfer path analysis with the above areas, and this will have to be explored for better understanding and optimal implementation on real structures.

REFERENCES

1. M.H.A. Janssens, J.W. Verheij and D.J. Thompson 1999 Journal of sound and vibration 226, 305-328. The use of an equivalent forces method for the experimental quantification of structural sound transmission.

2. LMS application notes on transfer path analysis 1995. The qualification and quantification of vibro-acoustic transfer paths.

3. G.B. Warburton 1976 The dynamical behaviour of structures. Oxford : Pergamon, 2nd edition.

4. D.J. Gorman 1982 Free vibration analysis of rectangular plates. New York : Elsevier, 1st edition.

5. D.E. Newland 1984 An introduction to random vibrations and spectral analysis. 2nd edition.

6. L. Cremer, M. Heckl / translated and revised by E.E. Ungar 1988 Structure-borne sound : structural vibrations and sound radiation at audio frequencies. Berlin : Springer. 2nd edition.

7. J.L. Bendat and A.G. Piersol 1993 Engineering application of correlation and spectral analysis. New York : Wiley interscience. 2nd edition.

APPENDIX A THEORETICAL BACKGROUND FOR PLATE FLEXURAL VIBRATIONS

The following assumptions have been made in the derivation of equations for flexural vibrations of a plate,

1. The thickness of the plate is small compared to its lateral dimensions
2. For higher mode shapes (complicated) the plate thickness is small compared to the distances between nodal lines
3. The lateral displacements are small compared to the thickness
4. The effects of rotary inertia are negligible such that the plate never possesses significant energy due to rotation of the element
5. There are no significant net tensile or compressive forces in the plate
6. The plate is assumed to be flat and homogeneous

The derivation follows that of [3] and [4].

A.1 Governing equation for flexural vibration of plate

Consider [3] a small element of plate with dimension dx , dy and h as shown in Figure A1

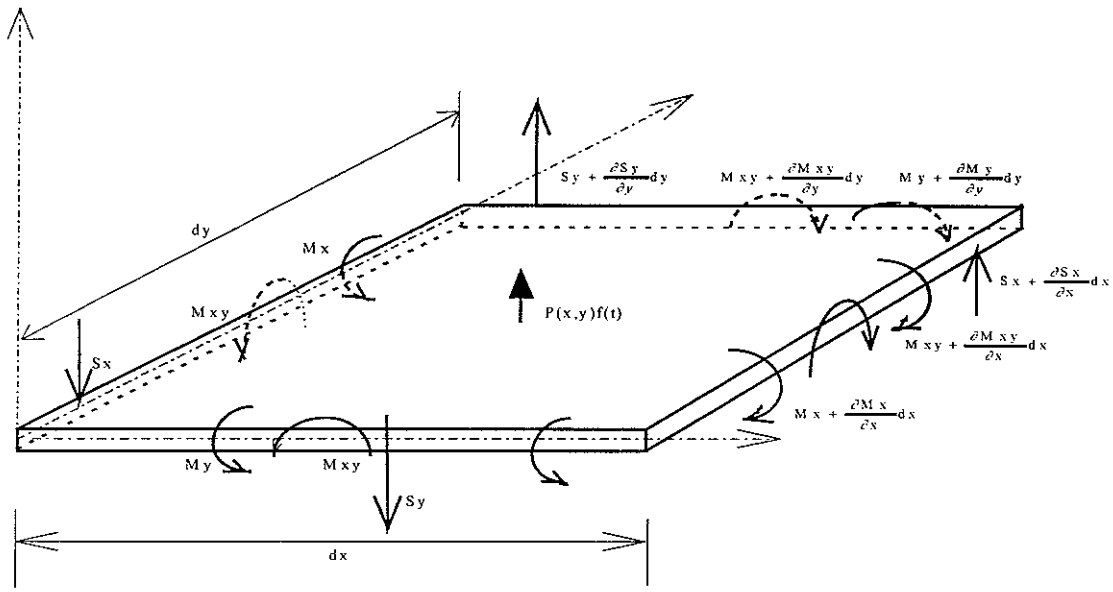


Figure A 1. Small element of a plate

The displacement along the z axis w(x,y,t) of the plate is due to the following forces and moments acting on it

- S_x, S_y -shear forces per unit length
- M_x, M_y and M_{xy} - bending moments per unit length
- P(x,y)f(t) - applied force per unit area
- $\rho h \frac{\partial^2 w}{\partial t^2}$ - inertia force per unit area

For equilibrium of this element

1. $\sum \mathbf{F} = \mathbf{0}$, which leads to

$$(S_y + \frac{\partial S_y}{\partial y} dy).dx - S_y dx + P(x,y).f(t)dxdy + (S_x + \frac{\partial S_x}{\partial x} dx)dy - S_x dy = \rho h \frac{\partial^2 w}{\partial t^2} dxdy$$

$$\frac{\partial S_y}{\partial y} dydx + P(x,y).f(t)dxdy + \frac{\partial S_x}{\partial x} dxdy = \rho h \frac{\partial^2 w}{\partial t^2} dxdy$$

$$\frac{\partial S_y}{\partial y} + P(x,y).f(t) + \frac{\partial S_x}{\partial x} = \rho h \frac{\partial^2 w}{\partial t^2} \qquad \text{(A1)}$$

and

2. $\sum \mathbf{M}_y = \mathbf{0}$, (taking moments about 'y' at centre line (dx/2)) , which leads to

$$(S_x \frac{dx}{2})dy + (S_x + \frac{\partial S_x}{\partial x} dx)\frac{dx}{2} dy + M_x dy - (M_x + \frac{\partial M_x}{\partial x} dx)dy + M_{xy} dx - (M_{xy} + \frac{\partial M_{xy}}{\partial y} dy)dx = 0$$

Neglecting products of smaller terms and simplifying

$$S_x \frac{dx}{2} dy + S_x \frac{dx}{2} dy - \frac{\partial M_x}{\partial x} dxdy - \frac{\partial M_{xy}}{\partial y} dydx = 0$$

$$S_x dxdy - \frac{\partial M_x}{\partial x} dxdy - \frac{\partial M_{xy}}{\partial y} dydx = 0$$

$$S_x = \frac{\partial M_x}{\partial x} + \frac{\partial M_{xy}}{\partial y} \qquad \text{(A2)}$$

similarly,

3. $\sum \mathbf{M_x} = \mathbf{0}$, (taking moments about 'x' at centre line (dy/2))

$$(S_y \frac{dy}{2})dx + (S_y + \frac{\partial S_y}{\partial y} dy)\frac{dy}{2} dx + M_y dx - (M_y + \frac{\partial M_y}{\partial y} dy)dx + M_{xy} dy - (M_{xy} + \frac{\partial M_{xy}}{\partial x} dx)dy = 0$$

Neglecting products of smaller terms and simplifying

$$(S_y \frac{dy}{2})dx + (S_y + \frac{\partial S_y}{\partial y} dy)\frac{dy}{2} dx + M_y dx - (M_y + \frac{\partial M_y}{\partial y} dy)dx + M_{xy} dy - (M_{xy} + \frac{\partial M_{xy}}{\partial x} dx)dy = 0$$

$$S_y dydx - \frac{\partial M_y}{\partial y} dydx - \frac{\partial M_{xy}}{\partial x} dxdy = 0$$

$$S_y = \frac{\partial M_y}{\partial y} + \frac{\partial M_{xy}}{\partial x} \qquad \text{(A3)}$$

Using expressions (A2) and (A3) for S_x and S_y , eqn (A1) can be written as

$$\frac{\partial^2 M_x}{\partial x^2} + 2\frac{\partial^2 M_{xy}}{\partial x dy} + \frac{\partial^2 M_y}{\partial y^2} + P(x,y)f(t) = \rho h \frac{\partial^2 w}{\partial t^2} \qquad \text{(A4)}$$

I. M_x and M_y are due to direct stresses σ_x and σ_y respectively

II. M_{xy} is due to shear stress σ_{xy}

From strength of materials

$$M_x = -D(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2})$$

$$M_y = -D(\frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2})$$

$$M_{xy} = -D(1-\nu^2)(\nu\frac{\partial^2 w}{\partial x \partial y})$$

where $D = \frac{h^3 E}{12(1-\nu^2)}$ = flexural rigidity of the plate

Using these relations in equation (A4) gives

$$-D(\frac{\partial^4 w}{\partial x^4} + \nu\frac{\partial^4 w}{\partial x^2 \partial y^2}) - 2D(1-\nu)\frac{\partial^4 w}{\partial x^2 \partial y^2} - D(\frac{\partial^4 w}{\partial y^4} + \nu\frac{\partial^4 w}{\partial x^2 \partial y^2}) + P(x,y)f(t) = \rho h\frac{\partial^2 w}{\partial t^2}$$

this can be simplified to give the general governing equation for flexural vibrations of plate

$$\boxed{\rho h\frac{\partial^2 w}{\partial t^2} + D(\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4}) = P(x,y)f(t)} \qquad \text{(A5)}$$

A.2 Free flexural vibration of simply supported plate

Using equation (A5) and letting P(x,y) f(t)=0, yields the governing equation for free vibration

$$\frac{\partial^4 w(x,y,t)}{\partial x^4} + 2\frac{\partial^4 w(x,y,t)}{\partial x^2 \partial y^2} + \frac{\partial^4 w(x,y,t)}{\partial y^4} + \frac{\rho h}{D}\frac{\partial^2 w(x,y,t)}{\partial t^2} = 0 \qquad \text{(A6)}$$

Consider a rectangular plate of dimension a and b. For convenience [4] the above equation can be made dimensionless using $\xi = x/a$ and $\eta = y/b$,

$$\frac{a\partial^4 w(x,y,t)/a}{a^4 \partial(x/a)^4} + 2a\frac{\partial^4 w(x,y,t)/a}{(ab)^2 \partial(x/a)^2 \partial(y/b)^2} + \frac{a\partial^4 w(x,y,t)/a}{b^4 \partial(y/b)^4} + \frac{\rho h}{D}\frac{a\partial^2 w(x,y,t)/a}{\partial t^2} = 0$$

Multiplying by a^3 gives

$$\frac{\partial^4 w(x,y,t)/a}{\partial(x/a)^4} + 2\frac{\partial^4 w(x,y,t)/a}{(b/a)^2 \partial(x/a)^2 \partial(y/b)^2} + \frac{\partial^4 w(x,y,t)/a}{(b/a)^4 \partial(y/b)^4} + \frac{\rho h}{D}\frac{a^4\partial^2 w(x,y,t)/a}{\partial t^2} = 0$$

and letting $\phi^4 = (b/a)^4$ gives

$$\frac{\partial^4 w(\xi,\eta,t)}{\partial\xi^4} + 2\frac{\partial^4 w(\xi,\eta,t)}{\phi^2\partial\xi^2\partial\eta^2} + \frac{\partial^4 w(\xi,\eta,t)}{\phi^4\partial\eta^4} + \frac{\rho h a^4}{D}\frac{\partial^2 w(\xi,\eta,t)}{\partial t^2} = 0$$

It is possible to separate the spatial and temporal part of $w(\xi,\eta,t)$ as

$$w(\xi,\eta,t) = W(\xi,\eta)\ W_t$$

Consider the case where W_t is a harmonic function of time, $e^{j\omega t}$

i.e

$$w(\xi,\eta,t) = W(\xi,\eta)\ e^{j\omega t}$$

Substituting this solution in the governing equation gives

$$\left\{ \frac{\partial^4 W(\xi,\eta)}{\partial\xi^4} + 2\frac{\partial^4 W(\xi,\eta)}{\phi^2\partial\xi^2\partial\eta^2} + \frac{\partial^4 W(\xi,\eta)}{\phi^4\partial\eta^4} - \lambda^4 W(\xi,\eta) \right\} e^{j\omega t} = 0 \qquad \text{(A7)}$$

where $\lambda^4 = \dfrac{\omega^2 \rho h a^4}{D}$,

Use is made of the Levy solution [4] which is valid for a plate with simple supports on two opposite edges and arbitrary constraints on the other two edges. Letting the two opposite edges at ξ=0 and ξ=1 simply supported, then the solution can be written as

$$W(x,y) = \sum_{m=1}^{\infty} Y_m(\eta)\sin(m\pi\xi)$$

where Y_m is function only of η, Substituting the spatial solution in equation (A7) ,

$$\sum_{m=1}^{\infty}\left(\frac{1}{\phi^4}\frac{d^4 Y_m(\eta)}{d\eta^4} - \frac{2(m\pi)^2}{\phi^2}\frac{d^2 Y_m(\eta)}{d\eta^2} + \left((m\pi)^4 - \lambda^4\right)Y_m(\eta) \right)\sin(m\pi\xi) = 0$$

which can also be written as

$$\sum_{m=1}^{\infty}\left(\frac{d^4 Y_m(\eta)}{d\eta^4} - 2\phi^2(m\pi)^2\frac{d^2 Y_m(\eta)}{d\eta^2} + \phi^4\left((m\pi)^4 - \lambda^4\right)Y_m(\eta) \right)\sin(m\pi\xi) = 0$$

From the above equation it is seen that for the equation to be valid for all values of ξ , it must satisfy the following condition

$$\frac{d^4 Y_m(\eta)}{d\eta^4} - 2\phi^2 (m\pi)^2 \frac{d^2 Y_m(\eta)}{d\eta^2} + \phi^4 \left((m\pi)^4 - \lambda^4\right) Y_m(\eta) = 0 \qquad \text{(A8)}$$

Equation (A8) is an ordinary differential equation of fourth order. A solution of this is of exponential form given by

$$Y_m(\eta) = A_m e^{\beta\eta}$$

Substitution of this solution into (A8) yields

$$\left(\beta^4 - 2\phi^2 (m\pi)^2 \beta^2 + \phi^4 \left((m\pi)^4 - \lambda^4\right)\right) e^{\beta\eta} = 0$$

This characteristic equation has four roots,

$$\beta = \pm \sqrt{\frac{2\phi^2 (m\pi)^2 \pm \sqrt{\left(2\phi^2 (m\pi)^2\right)^2 - 4\phi^4 \left((m\pi)^4 - \lambda^4\right)}}{2}}$$

$$= \pm \sqrt{\frac{2\phi^2 (m\pi)^2 \pm 2\phi^2 \lambda^2}{2}}$$

$$= \pm \phi \sqrt{(m\pi)^2 \pm \lambda^2}$$

The general solution will lead to oscillations only if $\lambda^2 > (m\pi)^2$ and then 2 roots will be real and 2 imaginary, the roots being

$$\beta_{1,2} = \pm \phi \sqrt{(m\pi)^2 + \lambda^2} \quad \text{and} \quad \beta_{3,4} = \pm j\phi \sqrt{\lambda^2 - (m\pi)^2}$$

The general solution then becomes

$$Y(\eta) = A_{m1} e^{(\phi\sqrt{(m\pi)^2+\lambda^2})\eta} + A_{m2} e^{(-\phi\sqrt{(m\pi)^2+\lambda^2})\eta} + A_{m3} e^{j(\phi\sqrt{\lambda^2-(m\pi)^2})\eta} + A_{m4} e^{-j(\phi\sqrt{\lambda^2-(m\pi)^2})\eta}$$

This can also be written as

$$Y(\eta) = A_1 \cosh(\eta\phi\sqrt{(m\pi)^2 + \lambda^2}) + A_2 \sinh(\eta\phi\sqrt{(m\pi)^2 + \lambda^2}) + A_3 \cos(\eta\phi\sqrt{\lambda^2 - (m\pi)^2})$$

$$+ A_4 \sin(\eta\phi\sqrt{\lambda^2 - (m\pi)^2})$$

or by letting

$\Theta_1 = \phi\sqrt{(m\pi)^2 + \lambda^2}$ and $\Theta_2 = \phi\sqrt{\lambda^2 - (m\pi)^2}$

$Y(\eta) = A_1 \cosh(\eta\Theta_1) + A_2 \sinh(\eta\Theta_1) + A_3 \cos(\eta\Theta_2) + A_4 \sin(\eta\Theta_2)$

The constants A_i in the above equation are required for complete understanding, which can be determined from specified boundary conditions of the plate. If the other two edges are also simply supported at η=0 and η=1, these are

1. Y(0)=0 and 2. Y(1)=0 } Displacement constraint

and

3. $\frac{\partial^2 Y(0)}{\partial\eta^2} = 0$ and 4. $\frac{\partial^2 Y(1)}{\partial\eta^2} = 0$ } Moment constraint

Applying the first condition gives

$0 = A_1 + A_3$ ---------- I

while applying 3rd condition

$0 = A_1\Theta_1^2 \cosh(0) + A_2\Theta_1^2 \sinh(0) - A_3\Theta_2^2 \cos(0) - A_4\Theta_2^2 \sin(0)$

$= A_1\Theta_1^2 - A_3\Theta_2^2$ -------II

Equations I and II imply that both A_1 and A_3 must be zero as Θ cant be zero.

From condition 2

$0 = A_2 \sinh(\Theta_1) + A_4 \sin(\Theta_2)$

or

$A_2 \sinh(\Theta_1) = -A_4 \sin(\Theta_2)$ ------III

and from condition 4

$0 = A_2\Theta_1^2 \sinh(\Theta_1) - A_4\Theta_2^2 \sin(\Theta_2)$ --- IV

Substituting III into equation IV gives

$$0 = A_2 \sinh(\Theta_1)(\Theta_1^2 + \Theta_2^2)$$

this term can only be equal to zero if A_2=0 . Then for vibration to occur at least A_4 must be non-zero . This leads to

$\sin(\Theta_2) = 0$ - this can be true for Θ_2= nπ where n= 1,2,3 .....

$$\text{i.e.} \quad \Theta_2 = \phi\sqrt{\lambda^2 - (m\pi)^2} = n\pi$$

$$\text{or} \quad \lambda^2 = (m\pi)^2 + \frac{(n\pi)^2}{\phi^2}$$

$$\lambda^2\phi^2 = \phi^2(m\pi)^2 + (n\pi)^2$$

From earlier relations ϕ and λ,

$$\lambda^2 = \omega a^2\sqrt{\frac{\rho h}{D}}, \text{ and } \phi^2 = (b/a)^2, \text{ hence}$$

$$\omega b^2\sqrt{\frac{\rho h}{D}} = (m\pi)^2\frac{b^2}{a^2} + (n\pi)^2$$

Rearranging gives the relation for natural frequencies of a simply supported plate as

$$\boxed{\omega = \left[\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2\right]\sqrt{\frac{D}{\rho h}}} \qquad \text{(A9)}$$

while the mode shape is given by

$$\boxed{\psi(x,y) = \sin\left(\frac{m\pi x}{a}\right)\sin\left(\frac{n\pi y}{b}\right)}$$

A.3 Forced Response of simply supported plate by modal superposition

Returning to equation (A5), the governing equation for forced response is given by

$$\rho h\frac{\partial^2 w}{\partial t^2} + D\left(\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2\partial y^2} + \frac{\partial^4 w}{\partial y^4}\right) = P(x,y)f(t)$$

From modal superposition [3], the response can be written as

$$w(x,y,t) = \sum_m \sum_n \psi_{mn}(x,y) q_{mn}(t)$$

where $\psi_{mn}(x,y)$ --- Mode shape function of mode (m,n)

$q_{mn}(t)$ --- Response of mode (m,n) in generalised co-ordinate system which is a function of time alone

Substituting this solution in equation (A5)

$$\rho h \sum\sum \psi_{mn} \ddot{q}_{mn} + \sum\sum D\left(\frac{\partial^4 \psi_{mn}}{\partial x^4} + 2\frac{\partial^4 \psi_{mn}}{\partial x^2 \partial y^2} + \frac{\partial^4 \psi_{mn}}{\partial y^4}\right) q_{mn} = P(x,y) f(t)$$

Use can be made of the orthogonality principle to simplify the above equation. To do that multiply above equation by mode shape ψ_{kl} and integrate it over the area of the plate

$$\int_0^a \int_0^b \left(\psi_{kl} \rho h \sum\sum \psi_{mn} \ddot{q}_{mn} dxdy\right) + \int_0^a \int_0^b \sum\sum D\left(\psi_{kl}\left(\frac{\partial^4 \psi_{mn}}{\partial x^4} + 2\frac{\partial^4 \psi_{mn}}{\partial x^2 \partial y^2} + \frac{\partial^4 \psi_{mn}}{\partial y^4}\right) q_{mn} dxdy\right)$$

$$= \int_0^a \int_0^b \psi_{kl} P(x,y) f(t) dxdy \qquad \text{(A10)}$$

For free vibration

$$\rho h \frac{\partial^2 w}{\partial t^2} + D\left(\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4}\right) = 0 \text{ ---------- I}$$

and the solution for the mn^{th} mode is

$$w = \psi_{mn} e^{j\omega_{mn} t}$$

Substituting this in equation (I) gives

$$\rho h \psi_{mn} \omega_{mn}^2 = D\left(\frac{\partial^4 \psi_{mn}}{\partial x^4} + 2\frac{\partial^4 \psi_{mn}}{\partial x^2 \partial y^2} + \frac{\partial^4 \psi_{mn}}{\partial y^4}\right)$$

Substitute this into equation (A10) , and reverse the order of Σ and $\int$ operators

$$\Sigma\Sigma\int_0^a\int_0^b\left(\psi_{kl}\rho h\psi_{mn}\ddot{q}_{mn}dxdy\right)+\Sigma\Sigma\int_0^a\int_0^b\left(\psi_{kl}\rho h\psi_{mn}\omega_{mn}^2 q_{mn}dxdy\right)=\int_0^a\int_0^b\psi_{kl}P(x,y)f(t)dxdy$$

$$\Sigma\Sigma\rho h\left(\ddot{q}_{mn}+\omega_{mn}^2 q_{mn}\right)\int_0^a\int_0^b\psi_{kl}\psi_{mn}dxdy=\int_0^a\int_0^b\psi_{kl}P(x,y)f(t)dxdy$$

From the orthogonality principle, if $\psi_{kl}\neq\psi_{mn}$, then

$$\int_0^a\int_0^b\psi_{mn}\psi_{kl}dxdy=0$$

Hence the summation disappears, the only non-zero terms being $k,l=m,n$

Hence

$$\ddot{q}_{mn}+\omega_{mn}^2 q_{mn}=\frac{\int_0^a\int_0^b\psi_{mn}P(x,y)f(t)dxdy}{\int_0^a\int_0^b\rho h\psi_{mn}^2 dxdy} \qquad \text{(A11)}$$

Write $\int_0^a\int_0^b\rho h\psi_{mn}^2 dxdy=M_{mn}$ where M_{mn} is the modal mass of the plate in mode m,n, then

the equation (A11) can be rewritten as

$$\ddot{q}_{mn}+\omega_{mn}^2 q_{mn}=\frac{\int_0^a\int_0^b\psi_{mn}P(x,y)f(t)dxdy}{M_{mn}}=F_{mn} \qquad \text{(A12)}$$

where F_{mn} is the generalized force in the mn^{th} mode

The modal mass for a rectangular plate can be estimated as

$$M_{mn}=\int_0^a\int_0^b\rho h\sin^2\left(\frac{m\pi x}{a}\right)\sin^2\left(\frac{n\pi y}{b}\right)dxdy$$

noting that $\sin^2(\theta)=(1-\cos(2\theta))/2$

$\therefore M_{mn}=0.25\,\rho\,a\,b\,h=0.25$ mass of the plate

If the force is a point source at (x_1,y_1) rather than distributed over the plate, then the integral in (A12) will have a non-zero value only at (x_1,y_1) and can be written as (using Kronecker's delta function to define a point force)

$P(x,y)= P_O\delta(x-x_1)\ \delta(y-y_1)$

Noting that ψ is function of (x,y),

$$\frac{f(t)\int\int P_o\psi_{mn}(x,y)\delta(x-x_1)\delta(y-y_1)dxdy}{M_{mn}} = \frac{f(t)\psi_{mn}(x_1,y_1)P_o}{M_{mn}}$$

$$\therefore \ddot{q}_{mn}+\omega_{mn}^2 q_{mn} = \frac{f(t)\psi_{mn}(x_1,y_1)P_o}{M_{mn}}$$

For harmonic excitation, f(t) = $e^{j\omega t}$ and $\ddot{q}_{mn} = -\omega^2 q_{mn}$. Substituting this in the above equation gives

$$-\omega^2 q_{mn}+\omega_{mn}^2 q_{mn} = \frac{\psi_{mn}(x,y)P_o e^{j\omega t}}{M_{mn}}$$

In order to introduce hysteretic damping, Young's modulus E becomes complex in nature and is E(1+jμ), where μ is the hysteretic damping factor. Due to the change in E , D also gets multiplied by (1+jμ). The above equation can be written as

$$-\omega^2 q_{mn}+(1+j\mu)\omega_{mn}^2 q_{mn} = \frac{\psi_{mn}(x,y)P_o e^{j\omega t}}{M_{mn}}$$

$$q_{mn} = \frac{\psi_{mn}(x_1,y_1)P_o e^{j\omega t}}{M_{mn}((\omega_{mn}^2-\omega^2)+j\mu\omega_{mn}^2)} \qquad \text{(A13)}$$

Hence, total solution is given by

$$w(x,y,t) = \sum_m\sum_n \psi_{mn}(x,y)q_{mn}(t)$$

$$= \sum_m\sum_n \frac{\psi_{mn}(x,y)\psi_{mn}(x_1,y_1)P_o e^{j\omega t}}{M_{mn}((\omega_{mn}^2-\omega^2)+j\mu\omega_{mn}^2)} \qquad \text{(A14)}$$

This equation can also be written in the form of a receptance as

$$\text{Receptance} = \alpha((x,y),(x_1,y_1)) = \frac{w(x,y,t)}{P_o e^{j\omega t}} = \sum_m \sum_n \frac{\psi_{mn}(x,y)\psi_{mn}(x_1,y_1)}{M_{mn}((\omega^2_{mn} - \omega^2) + j\mu\omega^2_{mn})} \tag{A15}$$

The magnitude of the receptance can be written as

$$\left|\alpha((x,y),(x_1,y_1))\right| = \sum_m \sum_n \frac{\psi_{mn}(x,y)\psi_{mn}(x1,y1)}{M_{mn}\left(\sqrt{(\omega^2_{mn} - \omega^2)^2 + (\mu\omega^2_{mn})^2}\right)} \tag{A16}$$

The mobility and accelerance can be obtained by multiplying equation (A15) with jω or $-\omega^2$.

APPENDIX B1 PROGRAM FOR TPA SIMULATIONS BASED ON SINGLE ACCELERANCE AND RESAMPLED ACCELERANCE

```
%
%       This prgram calls a function tpa_new
%
%       tpa_new is for estimating
%                       1. Accelerances
%                       2. SVD and Matrix inversion
%                       3. Reconstruction of force
%                       4. Response at receiver location
%
%       Implements single accelerance and resampled accelerance based
%       averaging for mean response estimation
%
%        Anand Thite
%        David Thompson
%        20.03.2000

clear

for kk=1:13
TIC
av=1;
for global_av=1:av

        min_f=10;               % minimum frequency
        max_f=3600;             % maximum frequency

        n_s=25;
        if kk==1
                H_logic=0;
                correction=0;
                n_ex=4;
                n_resp=4;
        elseif kk==3
                H_logic=0;
                correction=0;
                n_ex=4;
                n_resp=6;
        elseif kk==5
                H_logic=0;
                correction=0;
                n_ex=6;
                n_resp=6;
        elseif kk==7
                H_logic=0;
```

```
        correction=0;
        n_ex=6;
        n_resp=9;
elseif kk==2
        H_logic=0;
        correction=1;
        n_ex=4;
        n_resp=4;
elseif kk==4
        H_logic=0;
        correction=1;
        n_ex=4;
        n_resp=6;
elseif kk==6
        H_logic=0;
        correction=1;
        n_ex=6;
        n_resp=6;
elseif kk==8
        H_logic=0;
        correction=1;
        n_ex=6;
        n_resp=9;
elseif kk==9
        H_logic=1;
        correction=0;
        n_ex=4;
        n_resp=4;
elseif kk==10
        H_logic=1;
        correction=1;
        n_ex=4;
        n_resp=4;
elseif kk==11
        H_logic=1;
        correction=0;
        n_ex=6;
        n_resp=6;
elseif kk==12
        H_logic=1;
        correction=1;
        n_ex=6;
        n_resp=6;
elseif kk==13
        H_logic=0;
        correction=1;
        n_ex=4;
        n_resp=4;
```

```
        end
        acc_logic=kk;
        n_av=50;
        % File name for the output
        load c:\ant\result\lastfile
        if global_av==1
                Pattern=' Pattern of the file name'
                '- r-year (single digit) -date (2 digit) - month - (2 digit) -file number (2
digit)'
                last_file_saved=filenm
                if kk==13
                        mm=16;
                else
                        mm==kk;
                end

                str_fil=(['r02704'  num2str(mm) ]);
                'New file name='
                %filenm=str_fil
                filenm=input('results file name:','s');

                save c:\ant\result\lastfile.mat filenm
        end

        % Comments on the file name
if H_logic==1
        if correction==1
                case=' SV rejected - Average 2'
        else
                case=' All SV used - Average 2'
        end
else
        if correction==1
                case=' SV rejected - Average 1'
        else
                case=' All SV used - Average 1'
        end
end


[est_direct_mean est_mean est_max est_min F_est_mean F_est_max.....
         F_est_min ideal f e sv c_numb sv_used omega force]=tpa_new.....
        (correction,max_f,min_f,H_logic,n_s,n_ex,n_resp,n_av,acc_logic);
                F=F_est_mean;
                Force=force;
                X=est_mean;
                V=est_mean./(j*omega);
                acc_stddvn=est_max-est_mean;
```

```
                vel_stddvn=(est_max./(j*omega))-V;
                V_ideal=ideal./(j*omega);
                X_ideal=ideal;
                vel_glob(:,global_av)=V;
                vel_resp(:,1)=est_mean./(j*omega);
                vel_resp(:,2)=est_max./(j*omega);
                vel_resp(:,3)=est_min./(j*omega);
                vel_resp(:,4)=est_direct_mean./(j*omega);
end
S=case;
Sav=num2str(n_s);
Se=num2str(n_ex);
Sr=num2str(n_resp);

case

figure
loglog(f,abs(V_ideal),'m--',f,abs(vel_resp(:,1)),'r-')
title(['Linear response ( ',S,' )',',Runs= ',Sav,',Ex=',Se,',Resp=',Sr])
axis([min(f) max(f) 0 max(abs(vel_resp(:,1)))])
h=gca;
ht=get(h,'title');
set(ht,'Fontsize',10)
legend('m--','Ideal response','r-','Estimated mean')

figure
loglog(f,e,'r--',f,sv,'w')
title([' Singular values  (','Ex=',Se,',Resp=',Sr,')'])
axis([min(f) max(f) 0.001 300])
h=gca;
ht=get(h,'title');
set(ht,'Fontsize',10)
legend('r--','Error norm')
xlabel('Frequeny [Hz]')

 [acc_resp_oct c_freq]=oct3rd((X),f,10);
 [vel_resp_oct c_freq]=oct3rd((V),f,10);

[acc_std_oct c_freq]=oct3rd((acc_stddvn),f,10);
[vel_std_oct c_freq]=oct3rd((vel_stddvn),f,10);

[acc_std_amp c_freq]=oct3rdam((acc_stddvn),f,10);
[vel_std_amp c_freq]=oct3rdam((vel_stddvn),f,10);


temp=vel_resp_oct-vel_std_oct;
non_zero=find(temp<0);
for ii=1:length(non_zero)
```

```
        temp(non_zero(ii))=1;
end
vel_resp_min=10*log10(temp);
vel_resp_max=10*log10(vel_resp_oct+vel_std_oct);
vel_respm=10*log10(vel_resp_oct);

temp=acc_resp_oct-acc_std_oct;
non_zero=find(temp<0);
for ii=1:length(non_zero)
        temp(non_zero(ii))=1;
end
acc_resp_min=10*log10(temp);
acc_resp_max=10*log10(acc_resp_oct+acc_std_oct);
acc_respm=10*log10(acc_resp_oct);
clear temp

figure
semilogx(c_freq,vel_resp_min,'g:',c_freq,vel_respm,'r',c_freq,.....
        vel_resp_max,'g:',c_freq,oct3rdlg(V_ideal,f,10),'m--')
axis([min(c_freq) max(c_freq) -20 40]);

legend('m--','Ideal response','r-','Estimated mean','g:','Std deviation')
title(['1/3 Octave response- Power -( ',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,' )'])
h=gca;
ylabel(' Velocity level [dB] ref (1m/s)')
xlabel('Frequency [Hz]');
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');

set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)


figure

semilogx(c_freq,acc_resp_min,'g:',c_freq,acc_respm,'r',c_freq,.....
        acc_resp_max,'g:',c_freq,oct3rdlg(X_ideal,f,10),'m--')
axis([min(c_freq) max(c_freq) 10 95]);
legend('m--','Ideal response','r-','Estimated mean','g:','Std deviation')
title(['1/3 Octave response- Power -( ',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,' )'])
h=gca;
ylabel(' Acceleration level [dB] ref (1m/s2)')
xlabel('Frequency [Hz]');
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');
```

```
set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)


norm_std=10*log10((oct3rdav((vel_stddvn),f,10))./(oct3rdav((V),f,10)));

figure
semilogx(c_freq,norm_std)
axis([min(c_freq) max(c_freq) -40 10]);
title(['Deviation in response estimation ( ',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,' )']);

ylabel(' Normalised std deviation [dB] ref 1')
xlabel('Frequency [Hz]');
h=gca;
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');

set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)

shift=100*(oct3rdav((abs(V_ideal)-abs(V))./abs(V_ideal),f,10));
if correction==0
        save devn0.mat norm_std shift
else
        save devn1.mat norm_std shift
end

figure
semilogx(c_freq,shift)
axis([min(c_freq) max(c_freq) 0 1.3*max(shift)]);
title(['Error in estimation of response ( ',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,' )']);
ylabel(' % shift in mean value')
xlabel('Frequency [Hz]');
h=gca;
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');

set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)

[F_oct c_freq]=oct3rdav(F,f,10);
[F_octsum c_freq]=oct3rdav((sum(F'))',f,10);
```

```
figure
loglog(f,c_numb,'r')
title(['Condition of acclerance Matrix -(','Ex=',Se,',Resp=',Sr,' )'])
ylabel(' Condition number')
axis([min(f) max(f) 0 100000 ])
h=gca;
ht=get(h,'title');
set(ht,'Fontsize',10)
xlabel('Frequency [Hz]')

figure
semilogx(f,sv_used,'r')
title(['Singular values used -( ',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,' )'])
xlabel(' Frequency [Hz]')
ylabel(' Singular values used')
axis([min(f) max(f) 0 n_ex ])
h=gca;
ht=get(h,'title');
set(ht,'Fontsize',10)
xlabel('Frequency [Hz]')

for ii=1:n_ex
        F_ideal(ii,:)=Force;
end
cntf=(1:n_ex);
F_ideal_sum=sum(F_ideal(:,1:n_ex)')';
S1=[ 'g--' 'm-.' 'y+ ' 'r: ' 'c. ' 'bo ' 'w- '];
if min(min(F_oct))>min(min(F_ideal))
        Fmin=min(min(F_ideal))
else
        Fmin=min(min(F_oct))
end
figure
subplot(1,2,2)
if n_ex==2
loglog(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m-.',cntf,F_idea_sum,'w-')
elseif n_ex==3
loglog(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m-.',cntf,F_ideal(:,3),'y+',.....
        cntf,F_ideal_sum,'w-')
elseif n_ex==4
loglog(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m-.',cntf,F_ideal(:,3),'y+',cntf,....
        F_ideal(:,4),'r:',cntf,F_ideal_sum,'w-')
elseif n_ex==5
loglog(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m-.',cntf,F_ideal(:,3),'y+',cntf,....
        F_ideal(:,4),'r:',cntf,F_ideal(:,5),'cx',cntf,F_ideal_sum,'w-')
elseif n_ex==6
loglog(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m-.',cntf,F_ideal(:,3),'y+',cntf,....
```

```
        F_ideal(:,4),'r:',cntf,F_ideal(:,5),'cx',cntf,F_ideal(:,6),'wo',cntf,F_ideal_sum,'w-')
end
axis([1 4 0.001 150]);
title(['Input'])
pos=get(gca,'posi');
pos(1)=pos(1)+0.85*pos(3);
pos(3)=pos(3)*0.15;
set(gca,'posi',pos)
set(gca,'xticklabels',[' '])

subplot(1,2,1)
if n_ex==2
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m-.',c_freq,F_octsum,'w-')
elseif n_ex==3
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m-.',c_freq,F_oct(:,3),'y+',c_freq,.....
        F_octsum,'w-')
elseif n_ex==4
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m-.',c_freq,F_oct(:,3),'y+',c_freq,.....
        F_oct(:,4),'r:',c_freq,F_octsum,'w-')
elseif n_ex==5
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m-.',c_freq,F_oct(:,3),'y+',c_freq,.....
        F_oct(:,4),'r:',c_freq,F_oct(:,5),'cx',c_freq,F_octsum,'w-')
elseif n_ex==6
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m-.',c_freq,F_oct(:,3),'y+',c_freq,.....
        F_oct(:,4),'r:',c_freq,F_oct(:,5),'cx',c_freq,F_oct(:,6),'wo',c_freq,F_octsum,'w-')
end
pos=get(gca,'posi');
pos(3)=pos(3)*1.95;
set(gca,'posi',pos)


axis([min(c_freq) max(c_freq) 0.001 150]);
legend('w-','Sum of forces')
title(['Reconstructed Force -( ',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,' )'])
xlabel(' Frequency [Hz]')
h=gca;
ylabel(' Force [N]')
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');
set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)

%
%

                vel_resp(:,1)=est_mean./(j*omega);
```

```
                vel_resp(:,2)=est_max./(j*omega);
                vel_resp(:,3)=est_min./(j*omega);

        [vel_mean_oct c_freq]=oct3rdlg(vel_resp(:,1),f,10);
        [vel_max_oct c_freq]=oct3rdlg(vel_resp(:,2),f,10);
        [vel_min_oct c_freq]=oct3rdlg(vel_resp(:,3),f,10);

figure
semilogx(c_freq,vel_min_oct,'g-.',c_freq,vel_mean_oct,'r',c_freq,vel_max_oct,.....
        'g-.',c_freq,oct3rdlg(V_ideal,f,10),'m--')
axis([min(c_freq) max(c_freq) -20 20]);
legend('m--','Ideal response','r-','Estimated mean','g-.','68% confidence interval')
title(['1/3 Octave response- sorted mean -(',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,')'])
h=gca;
ylabel(' Velocity level [dB] ref (1m/s)')
xlabel('Frequency [Hz]');
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');

set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)

%Trial plot
        [vel_mean_oct c_freq]=oct3rd(vel_resp(:,4),f,10);
        [vel_max_oct c_freq]=oct3rd(vel_resp(:,2),f,10);
        [vel_min_oct c_freq]=oct3rd(vel_resp(:,3),f,10);
        vel_max=vel_max_oct;
        vel_min=vel_min_oct;
        vel_max_oct=(vel_mean_oct+0.5*(vel_max-vel_min));
        vel_min_oct=(vel_mean_oct-0.5*(vel_max-vel_min));

        non_zero=find(vel_min_oct<0);
        for ii=1:length(non_zero)
                vel_min_oct(non_zero(ii))=1;
        end

        vel_mean_oct=10*log10(vel_mean_oct);
        vel_max_oct=10*log10(vel_max_oct);
        vel_min_oct=10*log10(vel_min_oct);
figure
semilogx(c_freq,vel_min_oct,'g--',c_freq,vel_mean_oct,'r',c_freq,vel_max_oct,....
        'g--',c_freq,oct3rdlg(V_ideal,f,10),'m--')
axis([min(c_freq) max(c_freq) -20 20]);
legend('m--','Ideal response','r-','Estimated mean','g-.','68% confidence interval')
title(['1/3 Octave response- direct mean -(',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,')'])
h=gca;
```

```
ylabel(' Velocity level [dB] ref (1m/s)')
xlabel('Frequency [Hz]');
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');

set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)

%
[vel_glob_oct c_freq]=oct3rdlg(vel_glob,f,10);
if av>1
        figure
        semilogx(c_freq,oct3rdlg(V_ideal,f,10),'m--',c_freq,vel_glob_oct,....
        c_freq,vel_min_oct,'g-.',c_freq,vel_max_oct,'g-.')
        axis([min(c_freq) max(c_freq) -20 20]);
        legend('m--','Ideal response','r-','Estimated mean','g:','Std deviation')
        title(['1/3 Octave response- Power -(',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,')'])
        h=gca;
        ylabel(' Velocity level [dB] ref (1m/s)')
        xlabel('Frequency [Hz]');
        ht=get(h,'title');
        hxl=get(h,'xlabel');
        hyl=get(h,'ylabel');

        set(ht,'Fontsize',10)
        set(hxl,'Fontsize',10)
        set(hyl,'Fontsize',10)
end
eval(['save c:\ant\result\final\' filenm ' f V_ideal  vel_resp F e sv c_numb sv_used case
n_ex n_resp n_s n_av Force norm_std ']);
clear  f V_ideal  vel_resp F e sv c_numb sv_used case n_ex n_resp n_s  n_av  Force
F_ideal
TOC
end
```

```
function [est_direct_mean,est_mean,est_max,est_min,F_est_mean, F_est_max,.....
        F_est_min,ideal, f, e, sv, c_numb ,sv_used,omega,force]=tpa_new.....
        (correction,max_f,min_f,H_logic,n_s,n_ex,n_resp,n_av,acc_logic);

TIC     % stop watch
%
%       Function for estimating
%                       1. Accelerances
%                       2. SVD and Matrix inversion
%                       3. Reconstruction of force
%                       4. Response at receiver location
%
%       This function calls functions for calculation of
%                       1. theoretical accelerance      plataccl.mat
%                       2. Measured accelerance         Hmeasur.mat
%
%       Program is for simply supported rectangular plate
%       Response is estimated using modal superposition principle
%
%
%
%       input data
%

E=2.07e11;              % Young's modulus
nu=0.3;                 % Poisson ratio
rho=7850;               % density
h=0.0015;               % thickness
a=0.6;                  % length
b=0.5;                  % width

force=[  19  10 27  6 35 0.1  ];                % Input force

% Non-dimensional  excitation point (x=xp*a)   (y=yp*b)
xp=[0.41   0.51  0.62  0.31  0.83  0.71  ];
yp=[ 0.43  0.63   0.41 0.72  0.25  0.89  ];

%Non-dimensional   response points
resp_x=[ 0.55 0.8  0.9   0.6 0.2  0.3  0.5 0.7  0.61 0.23 ];
resp_y=[ 0.4  0.2  0.8  0.5  0.3  0.7  0.9 0.71 0.31 0.89];

micropho=[0.15.*a  0.45.*b];            % Microphone position

% noise foor
nois_in=0.01;                   %       Noise in force measurement
nois_out=0.001;                 %       Noise in accelerance- acceleration
nois_op_acc=0.001;              %       Noise in operative-acceleration
```

```
eta=0.03;                   %       damping loss factor
mmax=20;                    %       modal orders to be considered
nmax=18;                    %       modal orders to be considered
f=[min_f:max_f]';           %       frequency range

%
% plot range
pltxmin=10;
pltxmax=length(f);
%

j=sqrt(-1);

force=force(1:n_ex);        %
xp=xp(1:n_ex);              % Deciding number of excitation positions
yp=yp(1:n_ex);              %

resp_x=resp_x(1:n_resp);    % Deciding number of response positions
resp_y=resp_y(1:n_resp);    %

x_r=a.*resp_x;              % Dimensionalisation
y_r=b.*resp_y;
x_e=a.*xp;
y_e=b.*yp;

        omega=2*pi.*f;
        acc_nois=omega;                 % Type of noise


m=[1:mmax]';        % modal orders to be considered
n=[1:nmax];

m_a=m*ones(size(n))./a;     % form (mmax,nmax) arrays of m/a and n/b
n_b=ones(size(m))*n./b;

k2=(pi.*m_a).^2+(pi.*n_b).^2;           % square of plate wavenumber in (m,n)th mode

% natural frequency of (m,n)th mode
omega_mn=h.*sqrt(E./(12.*rho.*(1-nu.^2))).*k2;

 % modal mass = integral of rho*h*(sin(kx)*sin(ky)).^2.dx.dy
mass_mn=0.25.*rho.*a.*b.*h;

%
%       Accelerance calculation - Theoretical
%
%       For different transfer paths
%
```

```
inc_jj=0;
if (acc_logic/2-(fix(acc_logic/2))) > 0
        loop_operated='true'   % Screen output for program test
        for kk=1:n_ex
                for jj=1:length(resp_x)
                        [acclrnce(:,jj+inc_jj) a1 a2]=plataccl(m_a,n_b,omega_mn,.....
                        x_r(jj),y_r(jj),x_e(kk),y_e(kk),f,eta,mass_mn);
                        if kk==1
                                pi_resp(jj,1)=a2;
                        end
                end
                pi_ex(kk,1)=a1;
                 inc_jj=length(resp_x)*kk;
        end

        save acclrnce.mat acclrnce f

%
%       Accelerance calculation - Theoretical
%
%       For receiver paths

        for kk=1:n_ex
                [acc_mic(:,kk) a1 a2]=plataccl(m_a,n_b,omega_mn,micropho(1),.....
                micropho(2),x_e(kk),y_e(kk),f,eta,mass_mn);
                pi_respm=a2;
        end
        save acc_mic.mat acc_mic

%
%       Estimation of measured FRF's for the receiver location
%
        [H_mic Coh_2]=Hmeasur(acc_mic,200,nois_in,nois_out,acc_nois*.....
                        ones(1,n_ex));
        save H_mic.mat H_mic Coh_2
%
%
%
else
        load acclrnce
        load acc_mic
        load H_mic
end

mic_forc(length(f),n_ex)=0;
        for jj=1:n_ex
                mic_forc(:,jj)=force(jj).*ones(size(f));
        end
```

```
%
% ideal response
%
ideal=sum((acc_mic.*mic_forc).').';
clear mic_forc


for sample=1:n_s        % Loop for sampling

sample_no=sample
%
% Estimation of measured FRF's for different paths contributing to
% operational acceleration
%
        load  acclrnce
if (H_logic==1 | sample==1)
        [H Coh_1]=Hmeasur(acclrnce,n_av,nois_in,nois_out,acc_nois*.....
                ones(1,n_ex*n_resp));
        save H_estimat.mat H Coh_1 f
else
        load H_estimat
end
%
%       Opeartional acceleration calculations
%
accln(length(f),n_resp)=0;
a=zeros(length(f),n_ex);
for ii=1:n_resp
        for jj=1:n_ex
                accln(:,jj)=force(jj).*acclrnce(:,ii+(jj-1)*n_resp);
        end
        a(:,ii)=(sum(accln.')).'+nois_op_acc.*acc_nois.*(randn(size(f))+j*randn(size(f)));
end
%
%       Matrix inversion  & Force reconstruction

% Initialization

F((length(f)),n_ex)=0;
H_acc(n_resp,n_ex)=0;
coh_acc(n_resp,n_ex)=0;
sv(length(f),n_ex)=0;
sv_used(length(f),1)=0;

for ii=1:length(f)

    % Generating accelerance and coherence matrix for each frequency
    %
```

```
    for jj=1:n_ex
        H_acc(:,jj)=H(ii,(jj-1)*n_resp+1:jj*n_resp).';
        coh_acc(:,jj)=Coh_1(ii,(jj-1)*n_resp+1:jj*n_resp)';
    end
    [U,S,V]=svd(H_acc);
    E=3*sqrt((1-coh_acc)./(2*n_av.*coh_acc)).*abs(H_acc);        %error matrix
      e(ii)=norm(E);
      singular=diag(S);
      c_numb(ii,:)=[max(singular)/min(singular) ];
      sv(ii,:)=singular';
      rec_sing=1./(singular);
      inv_S=zeros(size(S));
      sv_used(ii)=n_ex;
      mm=1;
      for jj=1:n_ex
          inv_S(jj,jj)=rec_sing(jj);
          if correction==1
              singul=singular(jj);            %threshold
              e_logic=1;
          else
              singul=max(singular)+1000;  %threshold
          end

          if  singul<e(ii)                          %e(ii)>threshold
                                                        % if jj >1
              if  jj==1
                  inv_S(jj,jj)=1/e(ii);    % If all singular values
                  S(jj,jj)=0;                  % are less than e
              else
                  inv_S(jj,jj)=0;
                  S(jj,jj)=0;
                  sv_used(ii)=n_ex-mm;
                  mm=mm+1;
              end

          end
      end
      inv_Hacc=V*inv_S'*U';
      F(ii,:)=(inv_Hacc*a(ii,:).').';

end

for force_count=1:n_ex
      eval(['f' int2str(sample+(force_count-1)*n_s) '=' 'F(:,' int2str(force_count) ');']);
      eval(['save ' 'f' int2str(sample+(force_count-1)*n_s) ' f' int2str(sample+.....
      (force_count-1)*n_s)])
end
```

```
end             %End of sampling
%
%
est_resp1(length(f),n_s)=0;
for force_count=1:n_ex

        for kk=1:n_s
                eval(['load ' ' f' int2str(kk+(force_count-1)*n_s)])
                eval(['F_sample(:,' int2str(kk) ')=' 'f' int2str(kk+(force_count-1)....
                                        *n_s) ';'])
                est_resp(:,kk)=F_sample(:,kk).*H_mic(:,force_count);
        end
%
        est_resp1=est_resp+est_resp1;                   % Adding of all the responses

        F_est_mean(:,force_count)=(mean(F_sample.')).';
        F_est_max(:,force_count)=(std(F_sample'))'+abs(F_est_mean(:,force_count));
        F_est_min(:,force_count)=-(std(F_sample'))'+abs(F_est_mean(:,force_count));

end

        est_sort=sort(est_resp1.');
        est_direct_mean=(mean(est_resp1.')).';
        size_of_est=size(est_direct_mean)

        est_sort_mean_abs=(est_sort(0.5*n_s,:))';
        est_sort_min=(est_sort(0.16*n_s,:))';
        est_sort_max=(est_sort(0.84*n_s,:))';
%
%
est_mean=est_sort_mean_abs;
est_min=est_sort_min;
est_max=est_sort_max;
clear H_mic Coh_2
TOC   % stop watch
```

APPENDIX B2 PROGRAM FOR IMPLEMENTING PERTURBATION TECHNIQUE

```
%
%       Practical implementation of resampled accelerance based strategy
%
%       tpa_prac is for estimating
%                       1. Accelerances
%                       2. SVD and Matrix inversion
%                       3. Reconstruction of force
%                       4. Response at receiver location
%
%

clear
TIC

        n_ex=4;                 % number of forcing points - MAX= 6    MIN= 2
        n_resp=4;               % Number of response points MAX=10   MIN= 2 and
                                % n_resp>= n_ex  always
        min_freq=10;
        max_freq=3600;
        n_av=50;                % number of averages for H estimation
        acc_av=25;              %Averages for acceleration
        perturb_av=25;

        case=' Simulations based on perturbation technique'
        [est_mean est_mean_abs est_max est_min ideal f  F .....
        omega nois_plt av_loop coh_nois Force force_phase]=tpaprac(max_freq,.....
        min_freq,n_ex,n_resp,n_av,perturb_av,acc_av);

    % 0 or 1 as argument fo function
    % 0 for all singular values to be used
    % 1 for correction applied

        acc_resp(:,1)=est_mean_abs;
        acc_resp(:,2)=est_max;
        acc_resp(:,3)=est_min;
        acc_resp(:,4)=est_mean;

         X_ideal=ideal;
        for kk=1:4
                vel_resp(:,kk)=acc_resp(:,kk)./(j*omega);
        end
        V_ideal=ideal./(j*omega);
        S='All Singular values used-Perturb'
```

```
Sav=num2str(perturb_av);
Se=num2str(n_ex);
Sr=num2str(n_resp);

X=abs(mean((acc_resp)'))';        % Mean and Standard deviation

V=abs(mean((vel_resp)'))';

[acc_respm c_freq]=oct3rdlg((X),f,10);
[vel_respm c_freq]=oct3rdlg((V),f,10);


save c:\ant\result\final\temp.mat c_freq X_ideal V_ideal vel_resp  F case Force f n_ex
n_resp perturb_av force_phase

vel_stddvn=std(abs((vel_resp)'))';

norm_std=10*log10((oct3rdav((vel_stddvn),f,10))./(oct3rdav((X),f,10)));
figure
semilogx(c_freq,norm_std)
axis([min(c_freq) max(c_freq) 1.1*min(norm_std) 15]);
title(['Deviation in response estimation ( ',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,' )']);

ylabel(' Normalised std deviation [dB] ref 1')
xlabel('Frequency [Hz]');
h=gca;
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');

set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)

shift=100*(oct3rdav((abs(X_ideal)-X)./abs(X_ideal),f,10));
figure
semilogx(c_freq,shift)
axis([min(c_freq) max(c_freq) 0 1.3*max(shift)]);
title(['Error in estimation of response ( ',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,' )']);
ylabel(' % shift in mean value')
xlabel('Frequency [Hz]');
h=gca;
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');

set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
```

```
set(hyl,'Fontsize',10)

[F_oct c_freq]=oct3rdav(F,f,10);
[F_octsum c_freq]=oct3rdav((sum(F'))',f,10);

for ii=1:n_ex
        F_ideal(ii,:)=Force;
end
cntf=(1:n_ex);
F_ideal_sum=sum(F_ideal(:,1:n_ex)')';

if min(min(F_oct))>min(min(F_ideal))
        Fmin=min(min(F_ideal))
else
        Fmin=min(min(F_oct))
end
figure
subplot(1,2,2)
if n_ex==2
loglog(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m-.',cntf,F_idea_sum,'w-')
elseif n_ex==3
loglog(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m-.',cntf,F_ideal(:,3),'y+',cntf,.....
        F_ideal_sum,'w-')
elseif n_ex==4
loglog(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m-.',cntf,F_ideal(:,3),'y+',cntf,.....
        F_ideal(:,4),'r:',cntf,F_ideal_sum,'w-')
elseif n_ex==5
loglog(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m-.',cntf,F_ideal(:,3),'y+',cntf,.....
        F_ideal(:,4),'r:',cntf,F_ideal(:,5),'cx',cntf,F_ideal_sum,'w-')
elseif n_ex==6
loglog(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m-.',cntf,F_ideal(:,3),'y+',cntf,.....
        F_ideal(:,4),'r:',cntf,F_ideal(:,5),'cx',cntf,F_ideal(:,6),'wo',cntf,F_ideal_sum,'w-')
end
axis([1 4 0.001 150]);
title(['Input'])
pos=get(gca,'posi');
pos(1)=pos(1)+0.85*pos(3);
pos(3)=pos(3)*0.15;
set(gca,'posi',pos)
set(gca,'xticklabels',[' '])

subplot(1,2,1)
if n_ex==2
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m-.',c_freq,F_octsum,'w-')
elseif n_ex==3
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m-.',c_freq,F_oct(:,3),'y+',c_freq,.....
        F_octsum,'w-')
elseif n_ex==4
```

```
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m-.',c_freq,F_oct(:,3),'y+',c_freq,.....
        F_oct(:,4),'r:',c_freq,F_octsum,'w-')
elseif n_ex==5
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m-.',c_freq,F_oct(:,3),'y+',c_freq,.....
        F_oct(:,4),'r:',c_freq,F_oct(:,5),'cx',c_freq,F_octsum,'w-')
elseif n_ex==6
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m-.',c_freq,F_oct(:,3),'y+',c_freq,.....
        F_oct(:,4),'r:',c_freq,F_oct(:,5),'cx',c_freq,F_oct(:,6),'wo',c_freq,F_octsum,'w-')
end
pos=get(gca,'posi');
pos(3)=pos(3)*1.95;
set(gca,'posi',pos)
set(gca,'xticklabels',[' '])

axis([min(c_freq) max(c_freq) 0.001 150]);
legend('w-','Sum of forces')
title(['Reconstructed Force -( ',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,' )'])
xlabel(' Frequency [Hz]')
h=gca;
ylabel(' Force [N]')
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');
set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)


[F_ideal_sum_oct c_freq]=oct3rdav((sum(force_phase'))',f,10);
[F_oct c_freq]=oct3rdav(F,f,10);
[F_octsum c_freq]=oct3rdav((sum(F'))',f,10);


for ii=1:n_ex
        F_ideal(ii,:)=Force;
end
cntf=(1:n_ex);
F_ideal_sum=sum(F_ideal(:,1:n_ex)')';
S1=[ 'g--' 'm-.' 'y+ ' 'r: ' 'c. ' 'bo ' 'w- '];
if min(min(F_oct))>min(min(F_ideal))
        Fmin=min(min(F_ideal))
else
        Fmin=min(min(F_oct))
end
figure
subplot(1,2,2)
if n_ex==2
loglog(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m*')
```

```
elseif n_ex==3
loglog(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m*',cntf,F_ideal(:,3),'y+')
elseif n_ex==4
loglog(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m*',cntf,F_ideal(:,3),'y+',cntf,....
        F_ideal(:,4),'r:')
elseif n_ex==5
loglog(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m*',cntf,F_ideal(:,3),'y+',cntf,....
        F_ideal(:,4),'r:',cntf,F_ideal(:,5),'cx')
elseif n_ex==6
loglog(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m*',cntf,F_ideal(:,3),'y+',cntf,....
        F_ideal(:,4),'r:',cntf,F_ideal(:,5),'cx',cntf,F_ideal(:,6),'wo')
end
axis([1 4 0.001 150]);
title(['Input'])
pos=get(gca,'posi');
pos(1)=pos(1)+0.85*pos(3);
pos(3)=pos(3)*0.15;
set(gca,'posi',pos)
set(gca,'xticklabels',[' '])

subplot(1,2,1)
if n_ex==2
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m*',c_freq,F_octsum,'w-
',c_freq,F_ideal_sum_oct,'m-.')
elseif n_ex==3
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m*',c_freq,F_oct(:,3),'y+',c_freq,.....
        F_octsum,'w-',c_freq,F_ideal_sum_oct,'m-.')
elseif n_ex==4
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m*',c_freq,F_oct(:,3),'y+',c_freq,.....
        F_oct(:,4),'r:',c_freq,F_octsum,'w-',c_freq,F_ideal_sum_oct,'m-.')
elseif n_ex==5
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m*',c_freq,F_oct(:,3),'y+',c_freq,.....
        F_oct(:,4),'r:',c_freq,F_oct(:,5),'cx',c_freq,F_octsum,'w-
',c_freq,F_ideal_sum_oct,'m-.')
elseif n_ex==6
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m*',c_freq,F_oct(:,3),'y+',c_freq,.....
        F_oct(:,4),'r:',c_freq,F_oct(:,5),'cx',c_freq,F_oct(:,6),'wo',c_freq,F_octsum,'w-
',c_freq,F_ideal_sum_oct,'m-.')
end
pos=get(gca,'posi');
pos(3)=pos(3)*1.95;
set(gca,'posi',pos)


axis([min(c_freq) max(c_freq) 0.001 150]);
legend('w-','Actual sum of forces','m-.','Ideal sum of forces')
title(['Reconstructed Force -( ',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,')'])
xlabel(' Frequency [Hz]')
```

```
h=gca;
ylabel(' Force [N]')
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');
set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)


figure
semilogx(f,abs(F))
axis([min(f) max(f) min(min(abs(F))) 1.3*(max(max(abs(F))))]);
legend('w-','Summed-up force')
title(['Reconstructed Force - Linear-( ',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,' )'])
xlabel(' Frequency [Hz]')
h=gca;
ylabel(' Force [N]')
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');
set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)

        [vel_mean_oct c_freq]=oct3rdlg(vel_resp(:,1),f,10);
        [vel_max_oct c_freq]=oct3rdlg(vel_resp(:,2),f,10);
        [vel_min_oct c_freq]=oct3rdlg(vel_resp(:,3),f,10);


figure
semilogx(c_freq,vel_min_oct,'g-.',c_freq,vel_mean_oct,'r',c_freq,vel_max_oct,.....
        'g-.',c_freq,oct3rdlg(V_ideal,f,10),'m--')
axis([min(c_freq) max(c_freq) -40 20]);
legend('m--','Ideal response','r-','Estimated mean','g-.','68% confidence')
title(['1/3 Octave response- absolute mean -( ',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,' )'])
h=gca;
ylabel(' Velocity level [dB] ref (1m/s)')
xlabel('Frequency [Hz]');
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');

set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)

 [vel_mean_oct c_freq]=oct3rdlg(vel_resp(:,4),f,10);
```

```
figure
semilogx(c_freq,vel_min_oct,'g-.',c_freq,vel_mean_oct,'r',c_freq,vel_max_oct,.....
        'g-.',c_freq,oct3rdlg(V_ideal,f,10),'m--')
axis([min(c_freq) max(c_freq) -40 20]);
legend('m--','Ideal response','r-','Estimated mean','g-.','68% confidence')
title(['1/3 Octave response- direct mean-( ',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,' )'])
h=gca;
ylabel(' Velocity level [dB] ref (1m/s)')
xlabel('Frequency [Hz]');
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');

set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)


TOC
```

```
function [est_mean,est_mean_abs,est_max,est_min, ideal, f,.....
        F,omega,av_loop,coh_nois,nois_plt,force,force_phase]=tpaprac(max_f,.....
        min_f,n_ex,n_resp,n_av,perturb_av,acc_av)

TIC     % stop watch
%       Practical implementation of averaging based on resampled accelerance
%       Perturbation technique
%       Function for estimating
%                       1. Accelerances
%                       2. SVD and Matrix inversion
%                       3. Reconstruction of force
%                       4. Response at receiver location
%
%
%       input data

E=2.07e11;      % Young's modulus
nu=0.3;         % Poisson ratio
rho=7850;       % density
h=0.0015;       % thickness
a=0.6;          % length
b=0.5;          % width

% Non-dimensional  excitation point (x=xp*a)
%  excitation point (y=yp*b)

xp=[0.41   0.51  0.62  0.31  0.83  0.71  ];
yp=[ 0.43  0.63   0.41 0.72  0.25  0.89  ];
% Strength of excitations -

force=[ 19  10 27  6 35 0.1  ];

 %Non-dimensional  response points

resp_x=[ 0.55 0.8  0.9   0.6 0.2  0.3  0.5 0.7  0.61 0.23 ];
resp_y=[ 0.4  0.2  0.8  0.5  0.3  0.7  0.9 0.71 0.31 0.89];

micropho=[0.15.*a  0.45.*b]; % Microphone position
%Noise floor
nois_in=0.01;
nois_out=0.001;
nois_acc=0.001;

eta=0.03;                       % damping loss factor
mmax=20;                        % modal orders to be considered
nmax=18;                        % modal orders to be considered
f=[min_f:max_f]';               % frequency range
%
```

```
% plot range
pltxmin=10;
pltxmax=length(f);
%
%
j=sqrt(-1);
%
xp=xp(1:n_ex);              % Deciding number of excitation positions
yp=yp(1:n_ex);              %
force=force(1:n_ex);        %

resp_x=resp_x(1:n_resp);    % Deciding number of response positions
resp_y=resp_y(1:n_resp);    %


x_r=a.*resp_x;              % Dimensionalisation
y_r=b.*resp_y;
x_e=a.*xp;
y_e=b.*yp;


        omega=2*pi.*f;
        acc_nois=omega;             % Type of noise - Incresing with the frequency


%
m=[1:mmax]'; % modal orders to be considered
n=[1:nmax];
%
% form (mmax,nmax) arrays of m/a and n/b
%
m_a=m*ones(size(n))./a;
n_b=ones(size(m))*n./b;
%
% square of plate wavenumber in (m,n)th mode
%
k2=(pi.*m_a).^2+(pi.*n_b).^2;
%
% natural frequency of (m,n)th mode
%
omega_mn=h.*sqrt(E./(12.*rho.*(1-nu.^2))).*k2;
%
%
% modal mass = integral of rho*h*(sin(kx)*sin(ky)).^2.dx.dy
%
mass_mn=0.25.*rho.*a.*b.*h;
%
%
```

```
%       loop over frequencies - Loop executed only ones in the program
%
        inc_jj=0;
        for kk=1:n_ex
                for jj=1:length(resp_x)
                        [acclrnce(:,jj+inc_jj) a1 a2]=plataccl(m_a,n_b,.....
                        omega_mn,x_r(jj),y_r(jj),x_e(kk),y_e(kk),f,eta,.....
                        mass_mn);
                        if kk==1
                                pi_resp(jj,1)=a2;
                        end
                end
                pi_ex(kk,1)=a1;
         inc_jj=length(resp_x)*kk;
        end
        save acclrnce.mat acclrnce

%
%
%       Direct force method

        for kk=1:n_ex
                [acc_mic(:,kk) a1 a2]=plataccl(m_a,n_b,omega_mn,.....
                micropho(1),micropho(2),x_e(kk),y_e(kk),f,eta,mass_mn);

        end
        save acc_mic.mat acc_mic

%
%
% Accelerance from noise modified signals
%
%
        [H Coh_1]=Hmeasur(acclrnce,n_av,nois_in,nois_out,acc_nois*.....
                ones(1,n_ex*n_resp));
        save H_estimat.mat H Coh_1

%
%       Estimation of measured accelerance
%


        [H_mic Coh_2]=Hmeasur(acc_mic,200,nois_in,nois_out,acc_nois*....
                        ones(1,n_ex));
        save H_mic.mat H_mic Coh_2

%
TOC
```

```
TIC
%
%
%       Data for random phase information in force
%
        for jj=1:n_ex
                rand_phas=rand(size(f));
                rand_phas=zeros(size(rand_phas));
                force_phase(:,jj)=force(jj)*exp(j*rand_phas*2*pi);
        end

%
%
%       Matrix inversion  & Force reconstruction
%
% Initialization

F((length(f)),n_ex)=0;
H_acc(n_resp,n_ex)=0;

coh_acc(n_resp,n_ex)=0;
sv(length(f),n_ex)=0;
sv_used(length(f),1)=0;

for ii=1:length(f)

        % Generating accelerance and coherence matrix for each frequency
        %
        for jj=1:n_ex
                H_acc(:,jj)=H(ii,(jj-1)*n_resp+1:jj*n_resp).';
               coh_acc(:,jj)=Coh_1(ii,(jj-1)*n_resp+1:jj*n_resp)';
              A_acc(:,jj)=acclrnce(ii,(jj-1)*n_resp+1:jj*n_resp).';
        end

       % Noise magnitude for accelerance matrix perturbation
       %
        coh_nois=1;        % Band of noise for perturbation like 1 sigma, 2 sigma etc.
       nois_mag=coh_nois*sqrt((1-coh_acc)./(2*n_av*coh_acc)).*abs(H_acc);

       nois_size(perturb_av,n_ex*n_resp)=0;
       rand_nois=randn(size(nois_size));
       max_nois=max(rand_nois);


        for av_loop=1:perturb_av      % Number of averages for force estimation
               rand_nois_phas=rand(size(H_acc));
```

```
        a=A_acc*force_phase(ii,:)'+nois_acc.*acc_nois(ii).*(randn(size(force'))+j*randn(
size(force')));;
                for kk=1:n_resp
                        %rand_nois_mag(kk,1:n_ex)=(rand_nois(av_loop,1+.....
                        %n_ex*(kk-1):n_ex*kk))./max_nois(1,1+n_ex*(kk-1).....
                        %:n_ex*kk);
                        rand_nois_mag(kk,1:n_ex)=(rand_nois(av_loop,1+.....
                        n_ex*(kk-1):n_ex*kk));
                end
                nois_perturb=rand_nois_mag.*(nois_mag).*(cos(2*pi*.....
                        rand_nois_phas)+j*sin(2*pi*rand_nois_phas));

                H_acc1=H_acc+nois_perturb;

                [U,S,V]=svd(H_acc1);

                rec_sing=1./(diag(S));
                inv_S=zeros(size(S));           % Initialization
                for jj=1:n_ex
                        inv_S(jj,jj)=rec_sing(jj);
                end

                inv_Hacc=V*inv_S'*U';
                F_av(av_loop,:)=(inv_Hacc*a).';


        end     % End of the perturbation

        est_resp=F_av*H_mic(ii,:).';
        est_sort=sort(est_resp);
        est_sort_mean(ii,:)=mean(est_resp);
        est_sort_mean_abs(ii,:)=est_sort(0.5*perturb_av);
        est_sort_min(ii,:)=est_sort(0.16*perturb_av);
        est_sort_max(ii,:)=est_sort(0.84*perturb_av);

        F_sort=sort(F_av);
        F_sort_max(ii,:)=F_sort(0.84*perturb_av,:);
        F_sort_min(ii,:)=F_sort(0.16*perturb_av,:);
        F_sort_mean(ii,:)=mean(F_av);

        if (fix((ii+min(f))/50)-(ii+min(f))/50)==0
                frequency=ii+min(f)
        end

    end         %End of frequency loop over

    F=F_sort_mean;
```

```
    save force.mat F
                        % End of the Inversion process
%
%
clear H          % Clearing un-wanted variables
clear Coh_1

TOC
TIC
%       Acceleration prediction at receiver position

mic_forc(length(f),n_ex)=0;
for jj=1:n_ex
        mic_forc(:,jj)=force_phase(:,jj);
end
%
% ideal response
%
ideal=sum((acc_mic.*mic_forc).').';
clear mic_forc
%
%       Prediction by reconstructed force
est_mean=est_sort_mean;
est_min=est_sort_min;
est_max=est_sort_max;
est_mean_abs=est_sort_mean_abs;


TOC    % stop watch
```

APPENDIX B3 PROGRAM FOR CONFIDENCE INTERVAL ESTIMATION

```
%
% It evokes a function called  tpa_cov
%
%       Response calculated based on the usual method
%       Only mean operational response is used in this estimation
%
%       Confidence intervals are estimated using covariance matrix
%

clear
Tic

        n_s=25;
        n_monte=100;

        n_ex=4;                 % number of forcing points - MAX= 6    MIN= 2
        n_resp=4;               % Number of response points MAX=10   MIN= 2 and
                        % n_resp>= n_ex  always
        min_freq=10;
        max_freq=3600;

        n_av=50;                % number of averages for H estimation
        H_logic=0;      % If combined averaging 1 or else 0

        correction=1;   % 0 or 1 as argument fo function
                        % 0 for all singular values to be used
                        % 1 for correction applied

        mv_avg=0;       %1 and  has to be combined with H_logic=1

        % Comment with the saved file
        case='Singular values eliminated  - trial for Average -1 - Covariance trials Caa &
Cfit combined '
          % Cfit alone Caa & Cfit combined

        [est_mean est_min est_max F_est_mean F_est_max F_est_min ideal f .....
        e sv c_numb sv_used omega Force F_usual]=tpa_cov(correction,max_freq,....
                                                min_freq,H_logic,n_s,.....
                                                n_ex,n_resp,n_av,mv_avg);

        load H_mic

        acc_resp(:,1)=sum((H_mic.*F_est_mean)')';
        acc_resp(:,2)=sum((H_mic.*F_est_max)')';
        acc_resp(:,3)=sum((H_mic.*F_est_min)')';
        for kk=1:3
```

```
            vel_resp(:,kk)=acc_resp(:,kk)./(j*0.1*omega);
        end

        acc_usual=sum((H_mic.*F_usual)')'; % Usual way of calculations
        vel_usual=acc_usual./(j*0.1*omega);

        X_ideal=ideal;
        V_ideal=ideal./(j*0.1*omega);


      F=F_est_mean;  % Averaged force
      Sav=num2str(n_s);
      Se=num2str(n_ex);
      Sr=num2str(n_resp);

        if correction==1
            S='SV rejected ';
        else
            S='All SV used ';
        end

%
%       Plot - singular values

figure
loglog(f,e,'r--',f,sv)
title([' Singular values  (','Ex=',Se,',Resp=',Sr,')'])
axis([min(f) max(f) 0 max(max(sv))])
h=gca;
ht=get(h,'title');
set(ht,'Fontsize',10)
legend('r--','Error norm')
xlabel('Frequeny [Hz]')

%
%       Plot - Condition number

figure
loglog(f,c_numb)
title(['Condition of acclerance Matrix -(','Ex=',Se,',Resp=',Sr,' )'])
ylabel(' Condition number')
axis([min(f) max(f) 0 100000 ])
h=gca;
ht=get(h,'title');
set(ht,'Fontsize',10)
xlabel('Frequency [Hz]')

%
```

```
%       Plot - Singular values used

figure
semilogx(f,sv_used)
title(['Singular values used -( ',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,' )'])
xlabel(' Frequency [Hz]')
ylabel(' Singular values used')
axis([min(f) max(f) 0 n_ex ])
h=gca;
ht=get(h,'title');
set(ht,'Fontsize',10)
xlabel('Frequency [Hz]')

%
% Plot - ideal force

[F_oct c_freq]=oct3rdav(F,f,10);
[F_octsum c_freq]=oct3rdav((sum(F'))',f,10);

for kk=1:4
        F_ideal(kk,:)=Force;
end
cntf=(1:4);
F_ideal_sum=sum(F_ideal(:,1:n_ex)')';
S1=[ 'g--' 'm-.' 'y+ ' 'r: ' 'c. ' 'bo ' 'w- '];
figure
subplot(1,2,2)
if n_ex==2
semilogy(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m-.',cntf,F_idea_sum,'w-')
elseif n_ex==3
semilogy(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m-.',cntf,F_ideal(:,3),'y+',....
        cntf,F_ideal_sum,'w-')
elseif n_ex==4
loglog(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m-.',cntf,F_ideal(:,3),'y+',.....
        cntf,F_ideal(:,4),'r:',cntf,F_ideal_sum,'w-')
elseif n_ex==5
semilogy(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m-.',cntf,F_ideal(:,3),'y+',....
        cntf,F_ideal(:,4),'r:',cntf,F_ideal(:,5),'cx',cntf,F_ideal_sum,'w-')
elseif n_ex==6
semilogy(cntf,F_ideal(:,1),'g--',cntf,F_ideal(:,2),'m-.',cntf,F_ideal(:,3),'y+',....
        cntf,F_ideal(:,4),'r:',cntf,F_ideal(:,5),'cx',cntf,F_ideal(:,6),'bo',.....
        cntf,F_ideal_sum,'w-')
end
axis([1 4 0.001 120]);
title(['Input'])
pos=get(gca,'posi');
pos(1)=pos(1)+0.85*pos(3);
pos(3)=pos(3)*0.15;
```

```
set(gca,'posi',pos)
set(gca,'xticklabels',[' '])

%
%       Plot - reconstructed forces

subplot(1,2,1)
if n_ex==2
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m-.',c_freq,F_octsum,'w-')
elseif n_ex==3
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m-.',c_freq,F_oct(:,3),'y+',....
                c_freq,F_octsum,'w-')
elseif n_ex==4
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m-.',c_freq,F_oct(:,3),'y+',....
        c_freq,F_oct(:,4),'r:',c_freq,F_octsum,'w-')
elseif n_ex==5
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m-.',c_freq,F_oct(:,3),'y+',....
        c_freq,F_oct(:,4),'r:',c_freq,F_oct(:,5),'cx',c_freq,F_octsum,'w-')
elseif n_ex==6
loglog(c_freq,F_oct(:,1),'g--',c_freq,F_oct(:,2),'m-.',c_freq,F_oct(:,3),'y+',.....
        c_freq,F_oct(:,4),'r:',c_freq,F_oct(:,5),'cx',c_freq,F_oct(:,6),'bo',...
        c_freq,F_octsum,'w-')
end
pos=get(gca,'posi');
pos(3)=pos(3)*1.95;
set(gca,'posi',pos)

axis([min(c_freq) max(c_freq) 0.001 120]);
legend('w-','Sum of forces')
title(['Reconstructed Force -( ',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,' )'])
xlabel(' Frequency [Hz]')
h=gca;
ylabel(' Force [N]')
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');
set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)

%
%       Plot - velocity response along with standard deviation

vel_resp(:,1)=est_mean./(j*0.1*omega);
vel_resp(:,2)=est_max./(j*0.1*omega);
vel_resp(:,3)=est_min./(j*0.1*omega);

        [vel_mean_oct c_freq]=oct3rdlg(vel_resp(:,1),f,10);
```

```
        [vel_max_oct c_freq]=oct3rdlg(vel_resp(:,2),f,10);
        [vel_min_oct c_freq]=oct3rdlg(vel_resp(:,3),f,10);

figure
semilogx(c_freq,vel_min_oct,'g-.',c_freq,vel_mean_oct,'r',c_freq,vel_max_oct,....
        'g-.',c_freq,oct3rdlg(V_ideal,f,10),'m--')
axis([min(c_freq) max(c_freq) 0 40]);
legend('m--','Ideal response','r-','Estimated mean','g:','Std deviation')
title(['1/3 Octave response- Power -( ',S,',Runs= ',Sav,',Ex=',Se,',Resp=',Sr,' )'])
h=gca;
ylabel(' Velocity level [dB] ref (0.1m/s)')
xlabel('Frequency [Hz]');
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');

set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)


%
%       Plot- Velocity response calculated the usual way
%       Velocity estimation from the force calculated from the mean alone
%
%
        [vel_usual_oct c_freq]=oct3rdlg(vel_usual,f,10);


figure
semilogx(c_freq,vel_usual_oct,'r',c_freq,oct3rdlg(V_ideal,f,10),'m--')
axis([min(c_freq) max(c_freq) 0 40]);
legend('m--','Ideal response','r-','response_mean')
title(['1/3 Octave response- from Force estimated with mean operational accelerations'])
h=gca;
ylabel(' Velocity level [dB] ref (0.1m/s)')
xlabel('Frequency [Hz]');
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');

set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)



for cc=1:3
```

```
%
%       Confidence limits based on Covariance using Monte-Carlo simulations
%
clear c_ff c_pp
if cc==1
        load c:\ant\result\caa          % Covaraince in measurement of
        confid='Caa alone'              % operational acceleration
elseif cc==2
        load c:\ant\result\cfit         % Covaraince of the fit
        confid='Cfit alone'
else
        load c:\ant\result\caafit       % Combined covariance
        confid='Caa + Cfit'
end

%
%       Force  &     Response variations
%
for ii=1:length(f)

        for kk=1:n_ex
                cov_ff(kk,:)=c_ff(ii,1+(kk-1)*n_ex:kk*n_ex);
        end

        cov_pp(1,:)=c_pp(ii,1:2);
        cov_pp(2,:)=c_pp(ii,3:4);
        acc_mean=[real(acc_resp(ii,1)) imag(acc_resp(ii,1))]';
        [v d]=eig(cov_pp);
        F_est=abs(F_est_mean(ii,:)');
        [V D]=eig(cov_ff);
        for sample=1:n_monte        % Monte - Carlo simulation

                F_monte(sample,:)=(F_est+(V)*(D.^(1/2))*(V')*.....
                                randn(size(F_est)))';
                acc_monte(sample,:)=(acc_mean+(v)*(d.^(1/2))*.....
                                (v')*randn(size(acc_mean)))';
        end
        a_monte_mean(ii,:)=mean(acc_monte);             %Phase information included

        a_receiver(ii,:)=a_monte_mean(ii,1)+j*a_monte_mean(ii,2);

        vel_monte=acc_monte./(j*0.1*omega(ii));
        v_monte_mean(ii,:)=mean(vel_monte);
        v_receiver(ii,:)=v_monte_mean(ii,1)+j*v_monte_mean(ii,2);

        v_std(ii,:)=std(sqrt((vel_monte(:,1).^2)+(vel_monte(:,2).^2)));     % Standard
                                                                            deviation
                                                % for the optional plot
```

```
        F_monte_mean(ii,:)=mean(F_monte);
        F_std(ii,:)=std(F_monte);
        %
        %
        est_sort=sort(vel_monte(:,1)+j*vel_monte(:,2));          % Confidence limit
        est_sort_mean(ii,:)=est_sort(0.5*n_monte);               % from actual
        est_sort_min(ii,:)=est_sort(0.16*n_monte);               % distribution
        est_sort_max(ii,:)=est_sort(0.84*n_monte);
        %
        %
end
%
%
%       Plotting Confidence limits based on Covariance
%
[F_oct_mean c_freq]=oct3rdav(F_monte_mean,f,10);
[F_oct_std c_freq]=oct3rdav(F_std,f,10);
F_oct_max=F_oct_mean+F_oct_std;
F_oct_min=F_oct_mean-F_oct_std;

temp=F_oct_min;
non_zero=find(temp<0);
for ii=1:length(non_zero)
        temp(non_zero(ii))=1;
end
F_oct_min=(temp);
for ii=1:n_ex
        figure
        loglog(c_freq,F_oct_mean(:,ii),'w-',c_freq,F_oct_max(:,ii),'m-.',.....
                c_freq,F_oct_min(:,ii),'g--')
        axis([min(c_freq) max(c_freq) 1 1000]);
        xlabel(' Frequency [Hz]')
        ylabel(' Force [N]')
        title(['Reconstructed force- with confidence limits ( Force ',int2str(ii),'  ',confid '
)'])
        h=gca;
        ht=get(h,'title');
        hxl=get(h,'xlabel');
        hyl=get(h,'ylabel');

        set(ht,'Fontsize',10)
        set(hxl,'Fontsize',10)
        set(hyl,'Fontsize',10)
end

% Plot - Response along with standard deviation
%
[v_mean_oct c_freq]=oct3rd(v_receiver,f,10);
```

```
[v_std_oct c_freq]=oct3rd(v_std,f,10);
v_max_oct=v_mean_oct+v_std_oct;
v_min_oct=v_mean_oct-v_std_oct;

temp=v_min_oct;
non_zero=find(temp<0);
for ii=1:length(non_zero)
        temp(non_zero(ii))=1;
end
v_min_oct=10*log10(temp);
v_max_oct=10*log10(v_max_oct);
v_mean_oct=10*log10(v_mean_oct);
figure
semilogx(c_freq,v_min_oct,'g-.',c_freq,v_mean_oct,'r',c_freq,v_max_oct,.....
        'g-.',c_freq,oct3rdlg(V_ideal,f,10),'m--')
axis([min(c_freq) max(c_freq) 0 40]);
legend('m--','Ideal response','r-','Estimated mean','g:','standard deviation')
title(['1/3 Octave response- mean with phase informn ( ',confid,'  ',S,' )'])
h=gca;
ylabel(' Velocity level [dB] ref (0.1m/s)')
xlabel('Frequency [Hz]');
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');

set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)

%
%  Plot - Response along with confidence limits
%
        [vel_mean_sort_oct c_freq]=oct3rdlg(est_sort_mean,f,10);
        [vel_max_sort_oct c_freq]=oct3rdlg(est_sort_max,f,10);
        [vel_min_sort_oct c_freq]=oct3rdlg(est_sort_min,f,10);

figure
semilogx(c_freq,vel_min_sort_oct,'g-.',c_freq,vel_mean_sort_oct,'r',.....
        c_freq,vel_max_sort_oct,'g-.',c_freq,oct3rdlg(V_ideal,f,10),'m--')
axis([min(c_freq) max(c_freq) 0 40]);

legend('m--','Ideal response','r-','Estimated mean','g:','Confidence limit')
title(['1/3 Octave response- sorted mean ( ',confid,'  ', S,' )'])
h=gca;
ylabel(' Velocity level [dB] ref (0.1m/s)')
xlabel('Frequency [Hz]');
ht=get(h,'title');
hxl=get(h,'xlabel');
```

```
hyl=get(h,'ylabel');

set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)

%
% Plot - Response mean ( from phase information as well) and confidence limit
%       combination plot

figure
semilogx(c_freq,vel_min_sort_oct,'g-.',c_freq,v_mean_oct,'r',.....
        c_freq,vel_max_sort_oct,'g-.',c_freq,oct3rdlg(V_ideal,f,10),'m--')
axis([min(c_freq) max(c_freq) 0 40]);
legend('m--','Ideal response','r-','Estimated mean','g:','Confidence limit')
title(['1/3 Octave response- combination ( ',confid,'  ',S,' )'])
h=gca;
ylabel(' Velocity level [dB] ref (0.1m/s)')
xlabel('Frequency [Hz]');
ht=get(h,'title');
hxl=get(h,'xlabel');
hyl=get(h,'ylabel');

set(ht,'Fontsize',10)
set(hxl,'Fontsize',10)
set(hyl,'Fontsize',10)
%
%
Toc

        if cc==1
                save  c:\ant\result\covar1.mat  f V_ideal  vel_resp F e sv c_numb sv_used
                case .....
                n_ex n_resp v_receiver v_std F_monte_mean F_std est_sort_mean.....
                est_sort_min est_sort_max Force c_ff c_pp
        elseif cc==2
                save  c:\ant\result\covar2.mat  f V_ideal  vel_resp F e sv c_numb sv_used
                case .....
                n_ex n_resp v_receiver v_std F_monte_mean F_std est_sort_mean.....
                est_sort_min est_sort_max Force c_ff c_pp

        else
                save  c:\ant\result\covar3.mat  f V_ideal  vel_resp F e sv c_numb sv_used
                case .....
                n_ex n_resp v_receiver v_std F_monte_mean F_std est_sort_mean.....
                est_sort_min est_sort_max Force c_ff c_pp
        end
end     %end of cc
```

```
function [est_mean,est_min,est_max,F_est_mean, F_est_max, F_est_min,ideal, f, e,.....
 sv, c_numb ,sv_used,omega,force,F_usual]=tpa_cov(correction,max_f,min_f,.....
                                                  H_logic,n_s,n_ex,n_resp,n_av,mv_avg)

TIC    % stop watch
%
%
%       Function for estimating
%                       1. Accelerances
%                       2. SVD and Matrix inversion
%                       3. Reconstruction of force
%                       4. Response at receiver location
%                       5. Covariance matrix for force and response
%
%
%
%       input data

E=2.07e11;    % Young's modulus
nu=0.3;               % Poisson ratio
rho=7850;             % density
h=0.0015;             % thickness
a=0.6;          % length
b=0.5;          % width

% Non-dimensional  excitation point (x=xp*a)
%  excitation point (y=yp*b)

xp=[0.41   0.51  0.62  0.31  0.83  0.71  ];
yp=[ 0.43  0.63   0.41 0.72  0.25  0.89  ];

% Strength of excitations -

force=[ 19  10 27  6 35 0.1 ];

%Non-dimensional  response points
resp_x=[ 0.55 0.8  0.9   0.6 0.2  0.3  0.5 0.7  0.61 0.23 ];
resp_y=[ 0.4  0.2  0.8  0.5  0.3  0.7  0.9 0.71 0.31 0.89];

micropho=[0.15.*a  0.45.*b]; % Microphone position
nois_in=0.01;         % noise foor   0.01;   0.001;
nois_out=0.001;

eta=0.03;                       % damping loss factor
mmax=20;                        % modal orders to be considered
nmax=18;                        % modal orders to be considered
f=[min_f:max_f]';               % frequency range
%
```

```
% plot range
pltxmin=10;
pltxmax=length(f);
%
%
j=sqrt(-1);
%                       % n_resp=length(x_r);
                        % n_ex=length(x_e);
xp=xp(1:n_ex);                  % Deciding number of excitation positions
yp=yp(1:n_ex);                  %

force=force(1:n_ex);            %
resp_x=resp_x(1:n_resp);        % Deciding number of response positions
resp_y=resp_y(1:n_resp);        %


x_r=a.*resp_x;                  % Dimensionalisation
y_r=b.*resp_y;
x_e=a.*xp;
y_e=b.*yp;


        omega=2*pi.*f;
        acc_nois=omega;         % Type of noise


%
m=[1:mmax]'; % modal orders to be considered
n=[1:nmax];
%
% form (mmax,nmax) arrays of m/a and n/b
%
m_a=m*ones(size(n))./a;
n_b=ones(size(m))*n./b;
%
% square of plate wavenumber in (m,n)th mode
%
k2=(pi.*m_a).^2+(pi.*n_b).^2;
%
% natural frequency of (m,n)th mode
%
omega_mn=h.*sqrt(E./(12.*rho.*(1-nu.^2))).*k2;
%
%
% modal mass = integral of rho*h*(sin(kx)*sin(ky)).^2.dx.dy
%
mass_mn=0.25.*rho.*a.*b.*h;
%
```

```
%
%       loop over frequencies
%
inc_jj=0;
loop_operated='true'            % Screen output for program test
for kk=1:n_ex
        for jj=1:length(resp_x)
                [acclrnce(:,jj+inc_jj) a1 a2]=plataccl(m_a,n_b,.....
                omega_mn,x_r(jj),y_r(jj),x_e(kk),y_e(kk),f,eta,mass_mn);
                if kk==1
                        pi_resp(jj,1)=a2;
                end
        end
        pi_ex(kk,1)=a1;
        inc_jj=length(resp_x)*kk;
end

save acclrnce.mat acclrnce

%
%
%       Acceleration prediction at receiver position
%
%       Direct force method

for kk=1:n_ex
        [acc_mic(:,kk) a1 a2]=plataccl(m_a,n_b,omega_mn,.....
                micropho(1),micropho(2),x_e(kk),y_e(kk),f,eta,mass_mn);
        pi_respm=a2;
end

%
%
mic_forc(length(f),n_ex)=0;
for jj=1:n_ex
        mic_forc(:,jj)=force(jj).*ones(size(f));
end
%
% ideal response
%
ideal=sum((acc_mic.*mic_forc)')';
clear mic_forc
%
%
%       Estimation of measured FRF's
%
        [H_mic Coh_2]=Hmeasur(acc_mic,200,nois_in,nois_out,.....
                        acc_nois*ones(1,n_ex));
```

```
        save H_mic.mat H_mic Coh_2
%
% Accelerance from noise modified signals
%
%
        [H Coh_1]=Hmeasur(acclrnce,n_av,nois_in,nois_out,.....
                        acc_nois*ones(1,n_ex*n_resp));
        save H_estimat.mat H Coh_1
%
%

for sample=1:n_s        % Loop for sampling
        sample_number=sample
%
%
%       Acceleration calculations
%
%

accln(length(f),n_resp)=0;
a=zeros(length(f),n_ex);
for ii=1:n_resp
        for jj=1:n_ex
                accln(:,jj)=force(jj).*acclrnce(:,ii+(jj-1)*n_resp);
        end
        a(:,ii)=(sum(accln'))'+nois_out.*acc_nois.*(randn(.....
                size(f))+j*randn(size(f)));
end
if sample==1
        a_average(length(f),n_resp)=0;
end
a_average=a_average+a;

if sample==n_s
        a_average=a_average./n_s;
end

%
%       Covariance estimation
%
a_cov(:,1:n_resp)=real(a);              % Partitioning
a_cov(:,(n_resp+1):2*n_resp)=imag(a);

for kk=1:length(f)

        if sample==1
                a_bar=a_cov(kk,:);
                a_mov_av(kk,:)=a_cov(kk,:);
```

```
        else
                a_mov_av(kk,:)=a_mov_av(kk,:)+a_cov(kk,:);
                a_bar=a_mov_av(kk,:)./sample;                    % Moving average
        end
        delta_s=a_cov(kk,:)-a_bar;
        C_aa=(delta_s')*delta_s;
        for nn=1:2*n_resp
                Cov_aa(kk,(1+(nn-1)*(2*n_resp)):nn*2*n_resp)=C_aa(nn,:);
        end
end

%
%

if sample==1
        save c:\ant\result\cov_aa.mat Cov_aa
else
        cov_aa1=Cov_aa;
        load c:\ant\result\cov_aa
        Cov_aa=Cov_aa+cov_aa1;
        save c:\ant\result\cov_aa.mat Cov_aa
end

clear  a_cov a_bar C_aa delta_s Cov_aa       % Clearing all the variables

end     % end of sampling
%
%       Matrix inversion  & Force reconstruction
%       Initialization

F((length(f)),n_ex)=0;
H_acc(n_resp,n_ex)=0;
coh_acc(n_resp,n_ex)=0;
sv(length(f),n_ex)=0;
sv_used(length(f),1)=0;

%
% Calculations based on the mean only
%
%
load H_estimat

for ii=1:length(f)

    % Generating accelerance and coherence matrix for each frequency
    %
     for jj=1:n_ex
            H_acc(:,jj)=H(ii,(jj-1)*n_resp+1:jj*n_resp).';
```

```
        coh_acc(:,jj)=Coh_1(ii,(jj-1)*n_resp+1:jj*n_resp)';
    end
    [U,S,V]=svd(H_acc);
    E=1*sqrt((1-coh_acc)./(2*n_av.*coh_acc)).*abs(H_acc);        %error matrix
      e(ii)=norm(E);
      singular=diag(S);
      c_numb(ii,:)=[max(singular)/min(singular) ];
      sv(ii,:)=singular';
      rec_sing=1./(singular);
      inv_S=zeros(size(S));
      sv_used(ii)=n_ex;
      mm=1;
      for jj=1:n_ex
            inv_S(jj,jj)=rec_sing(jj);
            if correction==1
                  singul=singular(jj);            %threshold
                  e_logic=1;
            else
                  singul=max(singular)+1000;  %threshold
            end

             if  singul<e(ii)                          %e(ii)>threshold
                                                       % if jj >1

                  if  jj==1
                        inv_S(jj,jj)=1/e(ii);    % If all singular values
                        S(jj,jj)=0;                 % are less than e
                  else
                        inv_S(jj,jj)=0;
                         S(jj,jj)=0;
                        sv_used(ii)=n_ex-mm;
                        mm=mm+1;
                  end

             end
      end
      inv_Hacc=V*inv_S'*U';
      F_usual(ii,:)=(inv_Hacc*a_average(ii,:).')';

end

%
est_mean=est_sort_mean;
est_min=est_sort_min;
est_max=est_sort_max;

%
%       Covariance of the fit estimation
```

```
%

accln(length(f),n_resp)=0;
a=zeros(length(f),n_ex);
for ii=1:n_resp
        for jj=1:n_ex
                accln(:,jj)=F_usual(:,jj).*H(:,ii+(jj-1)*n_resp);
        end
        a_reconst(:,ii)=(sum(accln'))';
end
%
%
if n_ex==n_resp
        diff=1;
else
        diff=n_resp-n_ex;
end
for kk=1:length(f)
        for nn=1:n_resp
                cov_re_re(nn)=((real(a_average(kk,nn)-a_reconst(kk,nn))).^2)./(diff);
                cov_im_im(nn)=((imag(a_average(kk,nn)-a_reconst(kk,nn))).^2)./(diff);
                cov_re_im(nn)=(real(a_average(kk,nn)-a_reconst(kk,nn)).*imag.....
                        (a_average(kk,nn)-a_reconst(kk,nn)))./(diff);
                cov_im_re(nn)=(real(a_average(kk,nn)-a_reconst(kk,nn)).*imag.....
                        (a_average(kk,nn)-a_reconst(kk,nn)))./(diff);
        end
        for nn=1:n_resp
                cov_fit(nn,nn)=sum(cov_re_re);
                cov_fit(n_resp+nn,n_resp+nn)=sum(cov_im_im);
                cov_fit(nn,nn+n_resp)=sum(cov_re_im);
                cov_fit(n_resp+nn,nn)=sum(cov_im_re);
        end

        for nn=1:2*n_resp
                Cov_aa_fit(kk,(1+(nn-1)*(2*n_resp)):nn*2*n_resp)=cov_fit(nn,:);
        end
end
cov_fit
save c:\ant\result\covaafit.mat Cov_aa_fit

clear cov_fit Cov_aa_fit        % Clearing all the variables
%

clear  Coh_2 Coh_1
%
%       COMBINED COVARIANCE ESTIMATION
%
for cc=1:3
```

```
%
for ii=1:length(f)
        for jj=1:n_ex
                        A_acc(:,jj)=H(ii,(jj-1)*n_resp+1:jj*n_resp).';
        end

        A_cov=[real(A_acc) imag(A_acc);-imag(A_acc) real(A_acc)];

        load  c:\ant\result\covaafit
        for nn=1:2*n_resp
                cov_fit(nn,:)=Cov_aa_fit(ii,(1+2*n_resp*(nn-1)):2*n_resp*nn);
        end
        clear Cov_aa_fit
        load  c:\ant\result\cov_aa
        for nn=1:2*n_resp
                c_aa(nn,:)=(Cov_aa(ii,(1+2*n_resp*(nn-1)):.....
                        2*n_resp*nn))./(n_s-1);
        end
        clear Cov_aa
        [V D]=eig(cov_fit);
        [v d]=eig(c_aa);
        if cc==1
                Cov_combined=c_aa;
        elseif cc==2
                Cov_combined=cov_fit;
        else
                Cov_combined=c_aa+cov_fit;
        end
        cov_ff=inv((A_cov')*(inv(Cov_combined))*(A_cov));

        for kk=1:2*n_ex
                c_ff(ii,1+(kk-1)*2*n_ex:kk*2*n_ex)=cov_ff(kk,:);
        end

        H_cov=[real(H_mic(ii,:)) imag(H_mic(ii,:)); -imag(H_mic(ii,:))
                real(H_mic(ii,:))];

        cov_path=(H_cov)*(cov_ff)*(H_cov');
        c_pp(ii,1:2)=cov_path(1,:);
        c_pp(ii,3:4)=cov_path(2,:);

  end  % end of covariance estimation
        if cc==1
                file1=1
                save c:\ant\result\caa.mat c_pp c_ff
        elseif cc==2
                file1=2
                save c:\ant\result\cfit.mat c_pp c_ff
```

```
        else
                file1=3
                save c:\ant\result\caafit.mat c_pp c_ff
        end
end     % end of 3 cycles

TOC     % stop watch
```

APPENDIX B4 PROGRAM FOR THEORETICAL PLATE ACCELERANCE

```
function    [acclrnce, a1, a2]=plataccl(m_a,n_b,omega_mn,x_resp,y_resp,.....
            x_excit,y_excit,f,eta,mass_mn)

% Theoretical accelerance of  a rectangular simply supported plate
omega=2*pi.*f;
  psi_mn=sin(pi.*x_excit.*m_a).*sin(pi.*y_excit.*n_b);      % mode shape at the
                                                            excitation point
  a1=sum(psi_mn(:));
  psi_xy=sin(pi.*x_resp.*m_a).*sin(pi.*y_resp.*n_b);        % mode shape at the
                                                            response point
  a2=sum(psi_xy(:));
  for ii=1:length(f)
            rec_mn=psi_mn.*psi_xy./(omega_mn.^2-
                                    omega(ii).^2+j.*eta.*omega_mn.^2)./mass_mn;
            acclrnce(ii,1)=(-omega(ii).^2).*sum(rec_mn(:));
  end
```

APPENDIX B5 PROGRAM FOR MEASURED ACCELERANCE

```
function [H,COH]=Hmeasur(acclrnce,n_av,nois_in,nois_out,acc_nois)
%
% Function for estimating acclerance matrix
%
%
Sum_Saa=zeros(size(acclrnce));
Sum_Sbb=zeros(size(acclrnce));
Sum_Sab=zeros(size(acclrnce));
imp_modi=zeros(size(acclrnce));
resp=zeros(size(acclrnce));

for ii=1:n_av

imp_modi=ones(size(acclrnce))+nois_in*(randn(size(acclrnce))+j*randn(size(acclrnce)));
   resp=acclrnce+nois_out.*acc_nois.*(randn(size(acclrnce))+j.*randn(size(acclrnce)));

   Saa=imp_modi.*conj(imp_modi);
   Sab=resp.*conj(imp_modi);
   Sbb=resp.*conj(resp);
   Sum_Saa=Sum_Saa+Saa;
   Sum_Sbb=Sum_Sbb+Sbb;
   Sum_Sab=Sum_Sab+Sab;
end

H=Sum_Sab./Sum_Saa;

COH=(abs(Sum_Sab)).^2./(Sum_Saa.*Sum_Sbb);
```

APPENDIX B6 PROGRAM FOR 1/3 OCTAVE BAND CONVERSION

```
function [oct,cent_frq] =oct3rdlg(z,f,nn)
% 1/3rd octave band estimation for given spectra
%  x is a vector for which 1/3rd octave estimation required
%  f - frequency vector
%  nn - is a starting band number
%
[row col]=size(z);
n=nn;  looprun=1; k=1;
if min(f) > 11
        n=0.5+10*log10(min(f));
        if (str2num(int2str(n)))~=n
                n=(str2num(int2str(n+0.5)));
        end
else
        n=10;
end
while looprun==1
        centf=10.^(n/10);
        minb=10.^((n-0.5)/10);
        maxb=10.^((n+0.5)/10);
        if maxb > max(f)
                break
        end
        min_band(k,1)=str2num(int2str(minb));
        cent_f(k,1)=str2num(int2str(centf));
        max_band(k,1)=str2num(int2str(maxb));
        cent_frq(k,1)=centf;
        n=n+1;
        k=k+1;
end

for ii=1:col
        x1(:,1)=z(:,ii);
        x(min(f):max(f))=x1;
        n=nn; m=1;  looprun=1; k=1; mm=1; jj=1;
    for k=1:length(cent_f)
                if (k+1) < length(cent_f)
                        x(max_band(k))=0.5*x(max_band(k));
                end
                y(k,1)=sum(
x(min_band(k):max_band(k)).*conj(x(min_band(k):max_band(k))) );
                k=k+1;

    end
oct(:,ii)=10*log10((y));
end
```