Institute of Sound and Vibration Research

Experimental Validation of Force Identification Techniques

A.N. Thite and D.J. Thompson

ISVR Technical Memorandum 862

May 2001

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

Experimental Validation of Force Identification Techniques

by

A.N.Thite and D.J. Thompson

ISVR Technical Memorandum No. 862

May 2001

EXPERIMENTAL VALIDATION OF FORCE IDENTIFICATION TECHNIQUES

A N Thite

D J Thompson

Technical Memorandum

CONTENTS

1. INTRODUCTION

In transfer path analysis (TPA) [1-8] forces are usually identified indirectly from an accelerance matrix and a set of operational accelerations. The measurements of accelerances (measured from source locations to the response locations) and operational responses (measured at a series of locations due to the operational source), however, involve errors that are magnified by the matrix inversion, particularly at frequencies where the condition number of the accelerance matrix is high. This ultimately leads to errors in estimates of the contributions from these forces via different paths to the response. Hence it is important that the forces are identified accurately for the TPA to produce reliable results.

The force identification errors can generally be reduced by over-determination, i.e. using a larger number of responses than the number of forces to be identified, and employing a least squared error solution. The least squared error solution can be accomplished by Moore-Penrose pseudo-inversion of the accelerance matrix [5]. In such a least squares solution, however, it is observed that the errors in force reconstruction continue to be large at frequencies where the accelerance matrix is ill-conditioned. This can, for example, be due to the fact that only a small number of modes contribute significantly to the operational responses close to resonances; this number can be smaller than the number of forces to be identified [9] (over-determination, by adding extra response points does not help under this situation).

The least squares solution can also be achieved through singular value decomposition of the accelerance matrix. It is possible in this case to discard insignificant singular values in order to improve the effective condition number of the accelerance matrix and hence improve the force identification. For this approach a threshold is required for identifying singular values to be discarded. The singular values can be discarded based on an estimate for the error in the accelerances [1,5] or an estimate for the error in the measurement of operational accelerations [6,8]. As seen in [5,6], however, singular value rejection does not overcome the large force reconstruction error that may be present in the vicinity of antiresonances and the individual contribution from each force may contain large errors when a smaller number of modes (less than the number of forces) are contributing to the responses, even though the overall response might be accurately reconstructed.

For a long time in the field of digital image processing and more recently in relation to Nearfield Acoustic Holography (NAH), other techniques have been employed to improve source reconstruction [10-13]. These include iterative techniques of inversion and Tikhonov regularization. In these techniques, instead of minimising a cost function based on ordinary least squared errors, a function that incorporates some bias is introduced. This new function is minimised to identify the source. In both techniques, the error in the force reconstruction is separated into the bias error and the magnified variance. In the ordinary least squares solution only the variance is magnified and no bias error is introduced. There can be a large magnification of the variance at frequencies where the accelerance matrix is ill-conditioned. By introducing a bias error that increases with the value of some regularization parameter, it is observed that there exists a point (value of regularization parameter or number of iterations) where the bias error and the random error cross over. For larger values the bias error dominates, for smaller values the random error dominates. At the cross over point the cost function is minimised. The introduction of a bias error limits the magnification of random errors in the accelerance and the operational responses. In simulations the force identification and individual force contributions estimation using regularization has been found to be superior to both singular value rejection and the resampling of the accelerance matrix [7]. Both Tikhonov regularization and the iterative inversion performed equally well in reducing the error propagation.

All the above conclusions are based on the assumption of a noise model used in the numerical simulations to represent the experiments [5,6]. The robustness of the above techniques can only be determined by experimental validation. In this report experiments on a flat plate performed for this purpose are presented. The experimental set-up, measurements, data processing and the predictions based on the above techniques are discussed. In the 2nd chapter a brief description of the experimental set-up and the typical measurements is given. The predictions by different methods are discussed in chapter 3. The conclusions are given in chapter 4.

2. EXPERIMENTS

2.1 Experimental set-up

The experimental set-up used is shown in figure 1. Basically it consists of a rectangular plate (700 x 500 x 1.5 mm) made of steel. The plate was hung from a frame using elastic threads. Two electrodynamic shakers were used for excitation which were fed with signals from the same signal generator and the amplitudes of forces were varied by the gain adjustment on the amplifier. The shakers were connected to the flat rectangular plate through force gauges. The force gauges were mounted on the plate in such way that the mass loading which results in frequency response peak shift is minimal. Additional damping was introduced by damping films glued to the back of the plate.

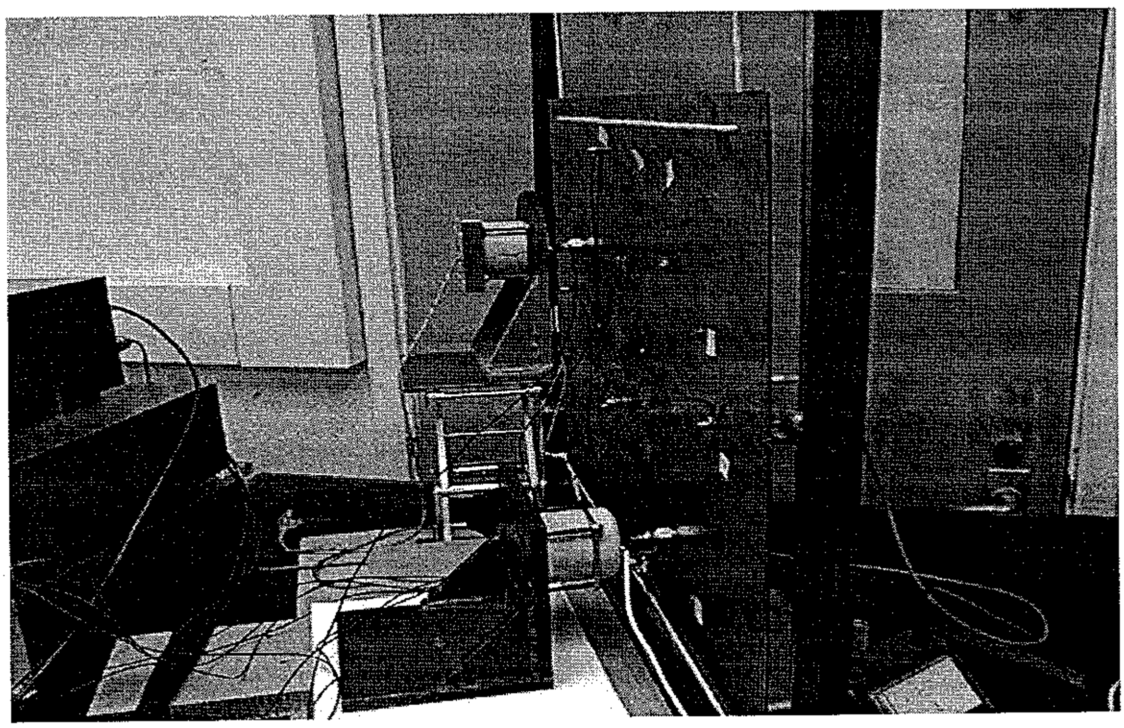

Figure 1 : Experimental set-up

Accelerometers and force gauges were mounted on cementing studs which were glued to the plate. This was done to ensure repeatability in the high frequency range (spatial deviations). The locations of force gauges and accelerometers are shown in figure 2. Three force positions are shown in the figure – F1 and F3 are driven by the two shakers while the force applied to F2 is zero. The extent to which the reconstruction is able to identify this is one of the aspects of the results that will be studied. Ten accelerometer positions are shown, though not all of them were measured together. Some are deliberately placed on lines of symmetry.

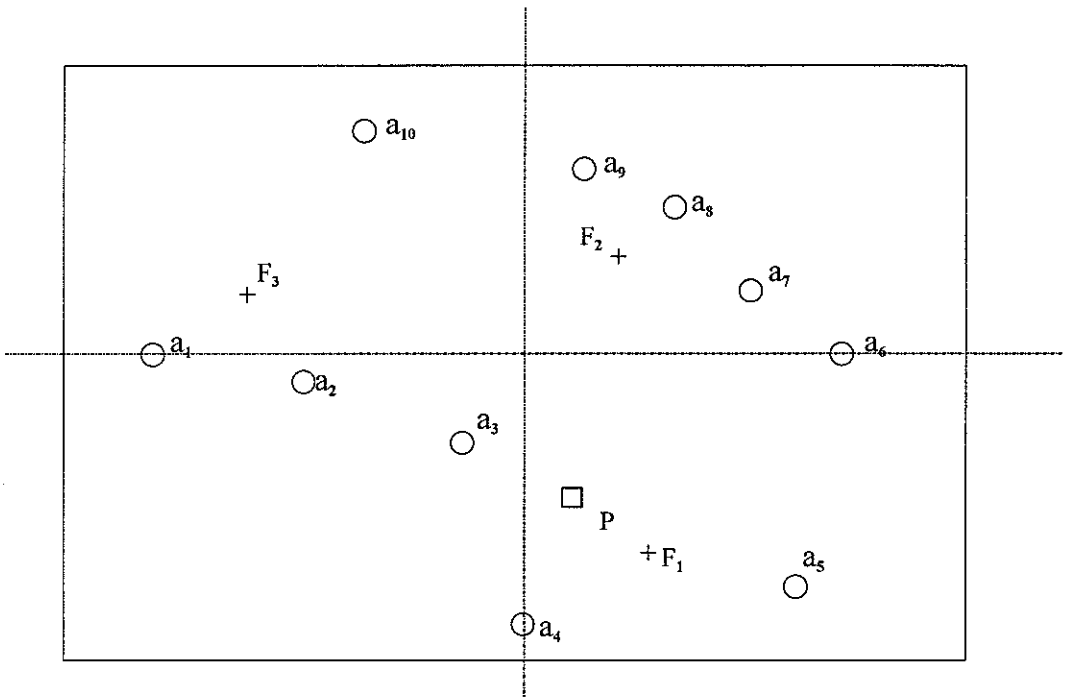

Figure 2. Locations of Accelerometers (O), Force gauges (+) and Microphone ( )

Details of equipment used:

Accelerometers	- Piezoelectric B&K type 4393 V
Microphone	- Condenser microphone B&K type 4191
Force gauges	- PCB force gauges type 218C
Charge amplifier	- FLYDE type FE-128-CA 4 channel

Measuring amplifier	- B&K type 2609
Preamplifiers	- B&K type 2669
Analyser	- HP 3566A 8-channel data acquisition system

Frequency range

A frequency range of 30 to 1600 Hz is used in the analysis. Below 30 Hz the coherence was found to be very low and errors were large on FRF estimates. The high frequency limit corresponds to the frequency at which the modal overlap is large enough not to notice the individual modes.

Frequency resolution

The frequency resolution is chosen as 1 Hz. Coherence improvement because of further refinement of the resolution is found to be small.

2.2 Measurements

The measurements for transfer path analysis can be categorised into two sections,

a. Operational response measurements

Operational responses were measured at locations 'a' on the plate when both shakers were in operation (this represents a situation where some machine is mounted on a plate). Due to the availability of only 8 channels, 6 accelerations were measured along with the two forces. To cover all the accelerometer locations, two sets of measurements were made. In a third set of measurements, one of the channels was used to measure the sound pressure near the rear surface of the plate (shown in figure 3). Excitation was applied at force locations F_1 and F_3 and the F_2 was taken as zero. The three sets of measurements contained the following responses (the order is based on the channel number used for the respective locations)

I. $\mathbf{a_8}$, $\mathbf{a_{10}}$, $\mathbf{a_7}$, $\mathbf{a_1}$, $\mathbf{a_3}$, $\mathbf{a_5}$ and $\mathbf{F_1}$, $\mathbf{F_3}$

II. $\mathbf{a_8}$, $\mathbf{a_{10}}$, $\mathbf{a_6}$, $\mathbf{a_1}$, $\mathbf{a_3}$, $\mathbf{a_4}$ and $\mathbf{F_1}$, $\mathbf{F_3}$

III. **p**, $\mathbf{a_{10}}$, $\mathbf{a_7}$, $\mathbf{a_1}$, $\mathbf{a_3}$, $\mathbf{a_5}$ and $\mathbf{F_1}$, $\mathbf{F_3}$ - where **p** represents sound pressure.

NB: Positions $\mathbf{a_2}$ and $\mathbf{a_9}$ were not used.

b. Transfer function measurements

Frequency response functions from each of the three forcing locations F to response locations a and p were measured, keeping one shaker operational and disconnecting the other shaker. All transducers were left screwed to the plate through cementing studs during these measurements.

Mass loading

Accelerometer - Stud (0.23 g)+ Accelerometer (2.4 g)= 2.63 g

The frequency above which the mass of the accelerometer plays a significant role is determined by equating the impedance of the mass with that of an infinite plate. The expression is derived as follows

$$m_{ad}\omega = 8\sqrt{\frac{E\rho h^4}{12(1-\nu^2)}}$$

$$m_{ad}2\pi f = 8h^2\sqrt{\frac{E\rho}{12(1-\nu^2)}}$$

For $\nu = 0.3$, this reduces to

$$f_{crossover} = \frac{0.39h^2\sqrt{E\rho}}{m_{ad}} \qquad (1)$$

where h is the thickness of the structure on which accelerometer is mounted, E is Young's modulus, ρ the density and ν the Poisson's ratio.

For the structure under consideration, a 1.5mm thick steel plate, this frequency is 13300 Hz.

Force gauge - Stud (1.48 g) + Force gauge (11.0 g) = 12.48 g

Using (1) again, the cut off frequency is calculated as 2790 Hz. Hence it is safe to consider that the frequency range 30 to 1600 Hz is not greatly influenced by mass loading.

For the second set of measurements, noise levels in all the channels were measured when both shakers were switched off before the actual measurements. Examples of frequency spectra for one force and one response channel along with noise levels are shown in figures 4 and 5.

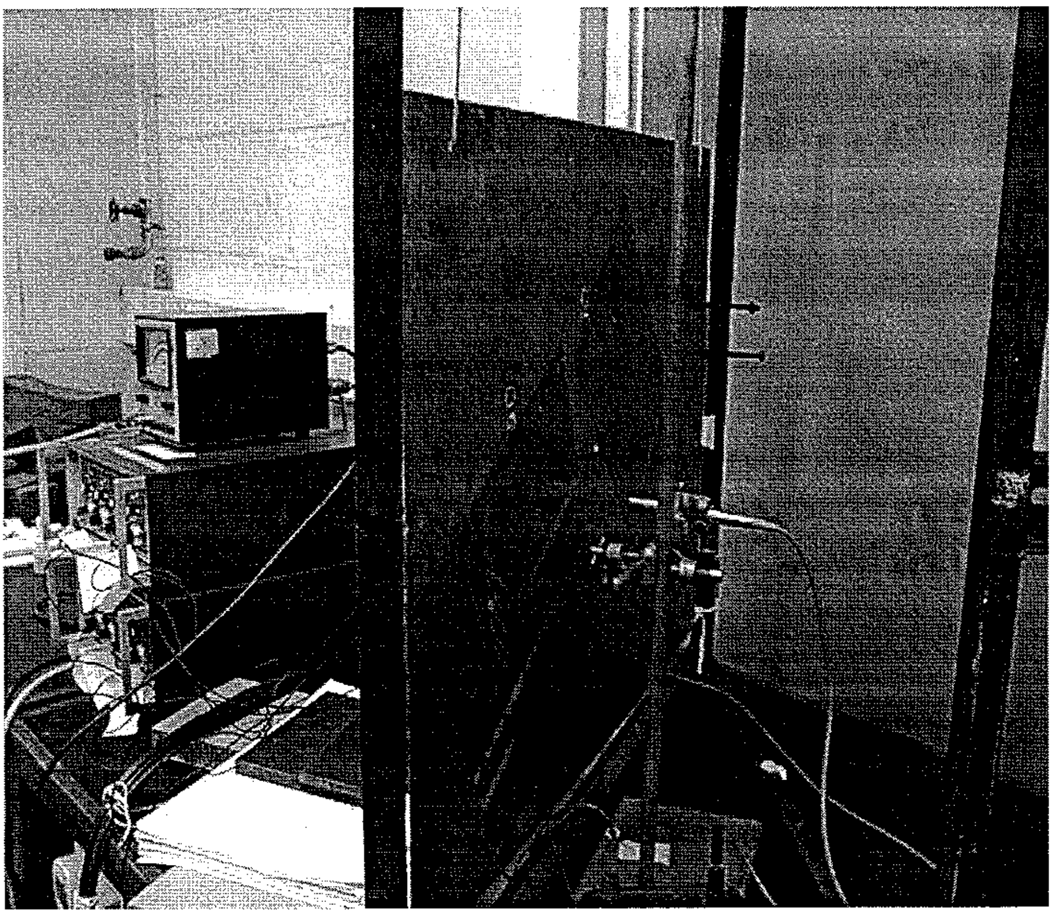

Figure 3. View showing microphone location

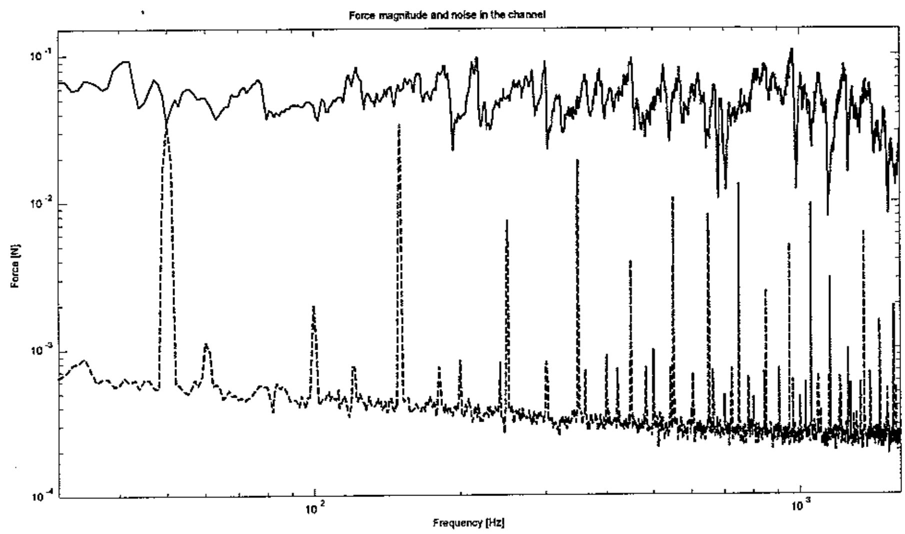

Figure 4. Force spectrum during operational measurements along with noise in channel (F_1)

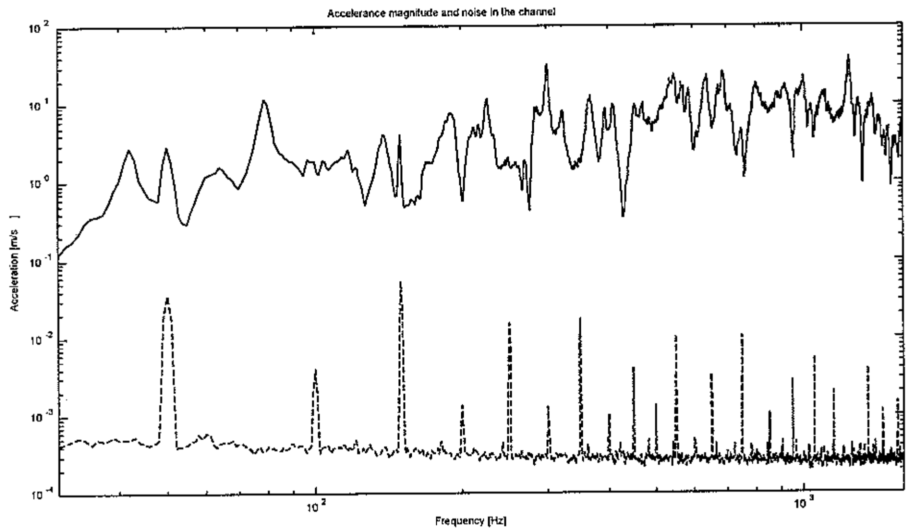

Figure 5. Example of acceleration spectrum during operational measurements along with noise in channel (a_1)

The force measurement is found to be influenced by a significant amount of measurement noise (Figure 4) particularly at 50, 150 and 350 Hz. This might create problems in comparing the measured forces and identified forces. However, the response signals are found to be well above the noise floor in most of the frequency range (Figure 5).

All data were captured initially as time histories. The operational responses are estimated (in the frequency domain) using a MATLAB program RESPONSE.M which is given in Appendix A. The amplitude of the response is taken as the square root of the auto spectrum and the phase is given by that of the cross spectrum between the response and a reference signal. One of the responses is taken as the reference signal (in this case response channel 6, which is position a_5 or a_4). The expression for the response can thus be written as

$$a_i = \sqrt{S_{i,i}}\, e^{i\angle S_{i,n}} \qquad (2)$$

where i is a response number and n is the reference response number. The magnitude of one of the responses calculated by this method is given in figure 6. All the responses for the 1st set of measurements are shown in figure 7 in one-third octave bands.

Based on the earlier study [6], the H_1 estimator is used in calculating the frequency response functions between different forcing points and response points. The MATLAB program used for this purpose is given in Appendix B. One of the frequency responses (accelerance) is given in figure 8. From the figure, the first few modes are observed to have very high damping. The coherence is observed to be poor at low frequencies and at 50, 150 and 250 Hz.

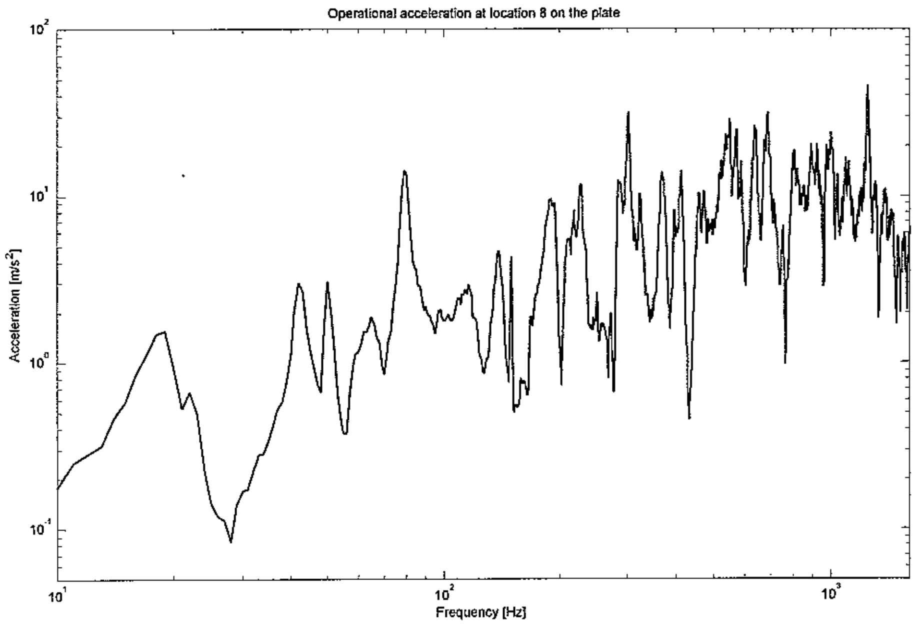

Figure 6. Operational acceleration spectrum at location a_8 for 1st set of measurements

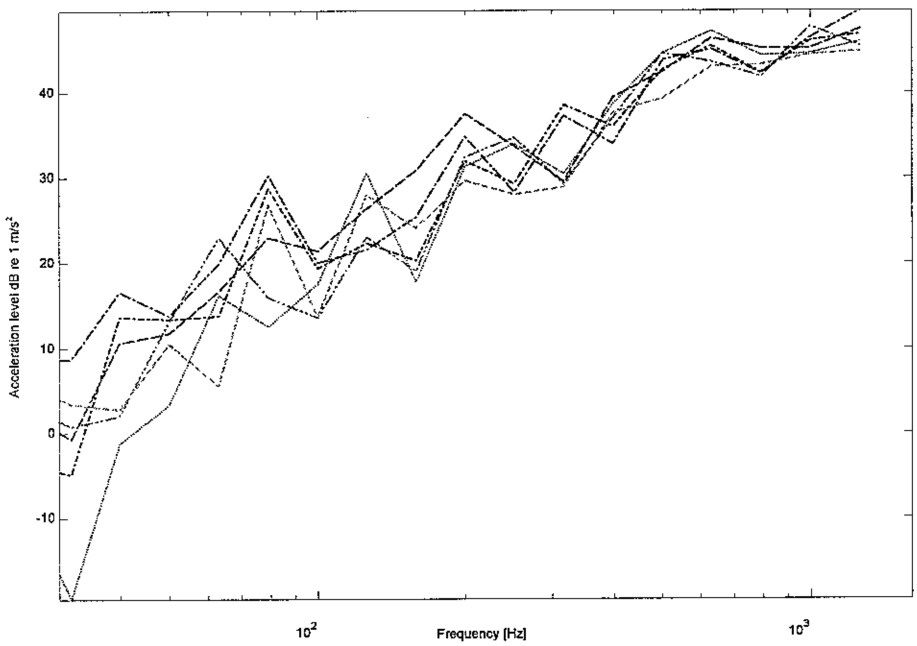

Figure 7. Measured acceleration responses in one-third octave bands - 1st set of measurements

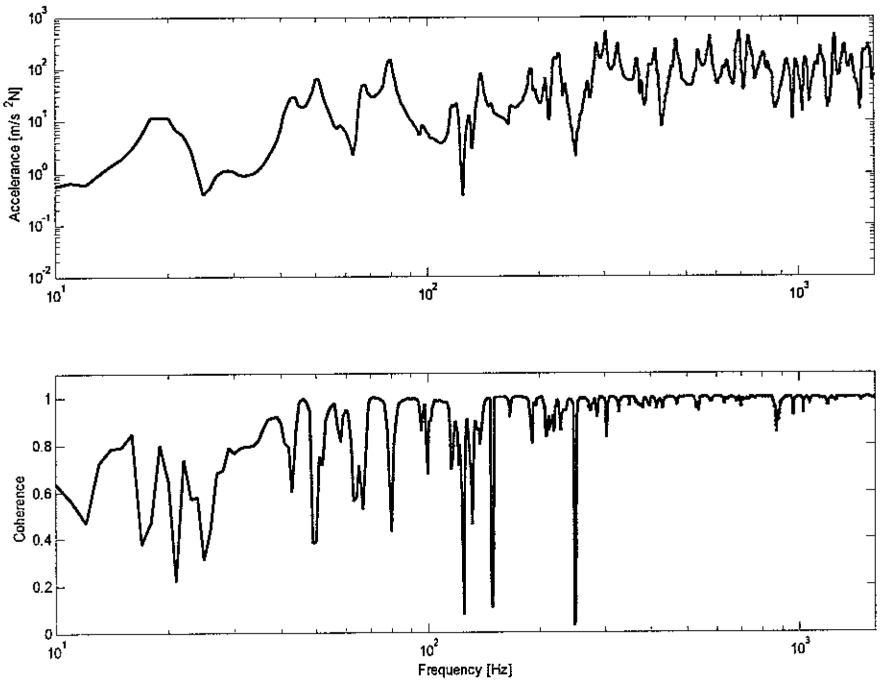

Figure 8. Magnitude of accelerance from force location 1 to response location 8 and associated coherence

Modal damping ratios have been calculated from the measured FRF's using the half power bandwidth for the first few modes and are shown in figure 9. The modal damping ratio is found to decrease from 0.08 to around 0.01 as the frequency increases. At higher frequencies it was not possible to extract damping values using this method.

2.3 Checking of data

The validity of measurements can be established by reconstructing the response in a direct way, which is possible here since the forces are also measured. Therefore the following expression is used in calculating the operational response, which is then compared with the measured response

$$a_i^r = \sum_{j=1}^{3} \hat{A}_{i,j} F_j \qquad (3)$$

where a'_i is a directly reconstructed i[th] response, $\hat{A}_{i,j}$ is the measured accelerance and F_j is the measured force.

The directly reconstructed response and the measured response for location a_8 in the 2nd set of measurements is given figure 10. The direct reconstruction of the response agrees very well with the measured one although some differences can be seen particularly in the high frequency region. The 1/3 octave band representation is shown in figure 11. The variations in direct reconstruction are found to average out in the 1/3 octave band representation. The root mean square difference between the two curves is 1.3 dB.

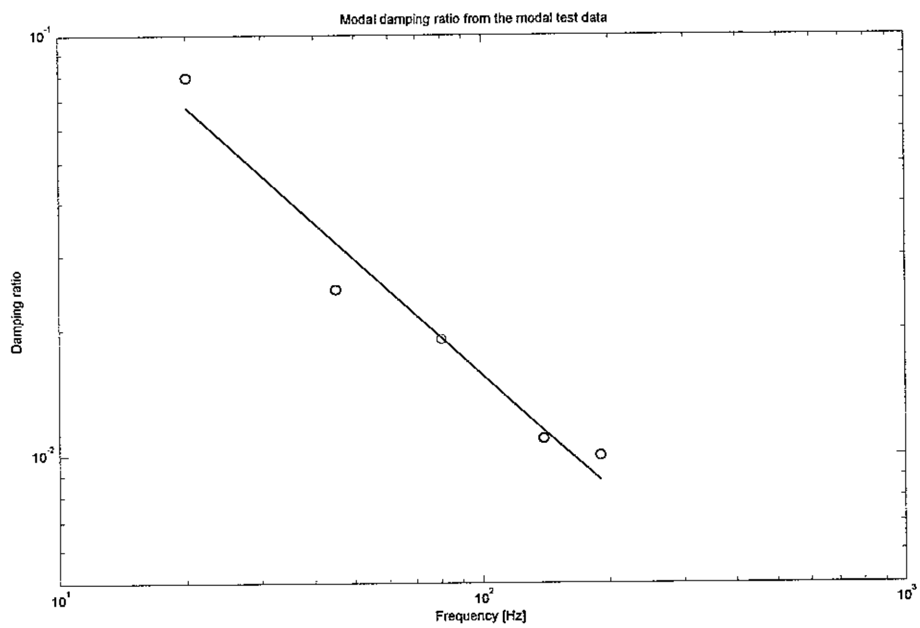

Figure 9. Measured modal damping ratio

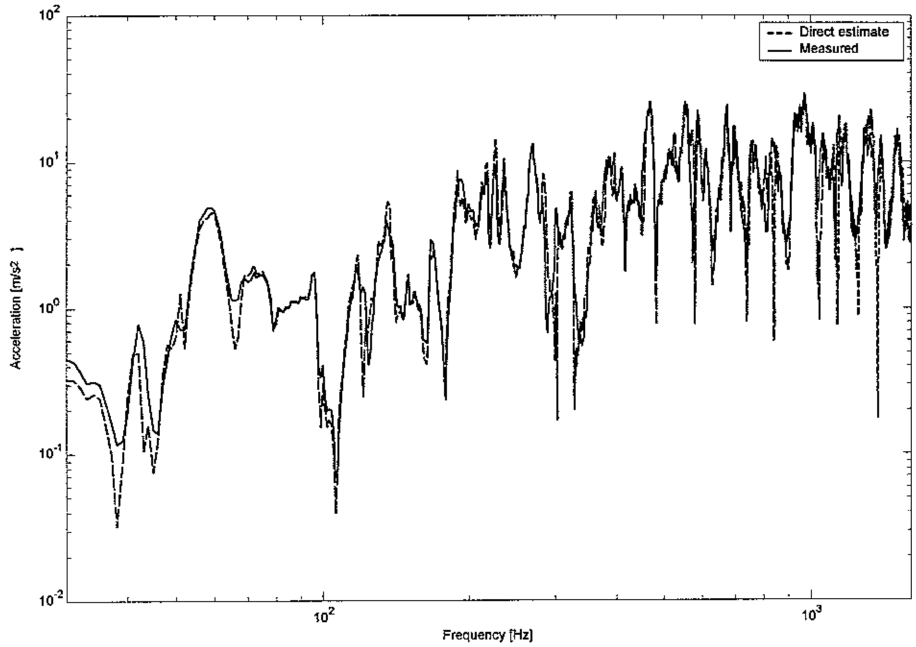

Figure 10. Acceleration response a_8 measured and directly reconstructed.

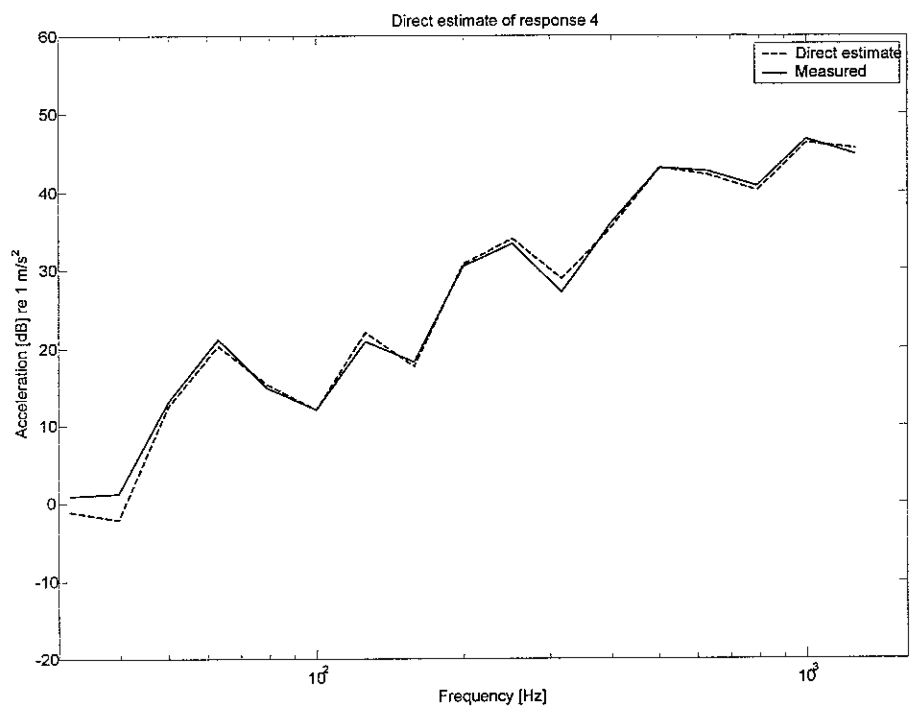

Figure 11. 1/3 octave acceleration response a_8 measured and directly reconstructed.

Measured forces in 1/3 octave band representation are shown in figure 12a. One of the forces is deliberately always larger than the other one by an average of 3.5 dB. The coherence between these two forces is shown in figure 12b. In most of the frequency region the forces are found to be well correlated.

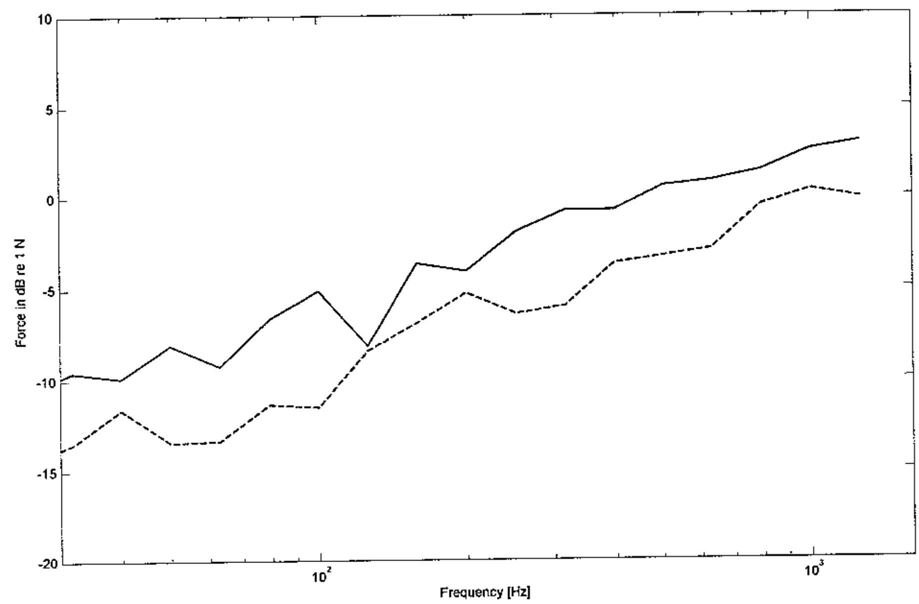

Figure 12a. 1/3 octave measured forces - Second set of measurements

__________ F_1, _ _ _ _ _ _ _ F_3

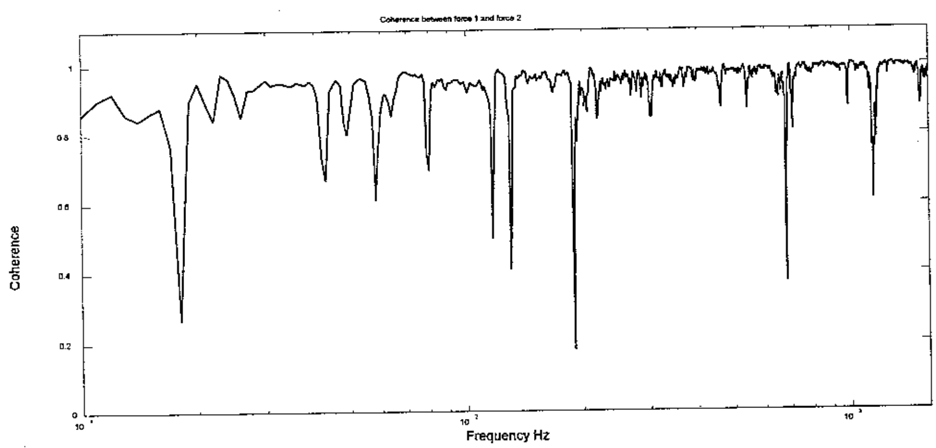

Figure 12b. Coherence between force F_1 and force F_3.

3. VALIDATION

3.1 Selection of data sets

In order to validate the theoretical concepts developed in [5-7], one of the response points is chosen as the 'receiver' location and others are used for force identification. Throughout this report 4 responses are used to identify 3 forces. In general it is found that the quality of the inversion on the measured data presented here is higher than in [7]. This is due to the assumed noise model in [7] combined with higher damping in the measurements. There are many possible combinations of response points that could be used. In order to discriminate between the various inversion methods, combinations that give a poor prediction using Moore-Penrose pseudo-inversion are selected. To select the combinations of response points, the reconstruction errors are evaluated using the Moore-Penrose pseudo inverse for all combinations.

In terms of singular value decomposition, the Moore-Penrose pseudo inversion is given by

$$\hat{F} = VS^{-1}U^{H}a \qquad (4)$$

This force vector is used in reconstructing the response. An example of the responses predicted by force identification from one of the combinations is given in figure 13. Response predictions are found to have large errors as seen from the 1/3 octave band representation. The root mean square error over all 1/3 octave bands is found to be 1.8 dB. This measure will be used to compare different strategies.

One interesting feature observed is the under-estimation of the response at most of the frequencies. This could be due to the principle used in response estimation, equation (2), where the phase difference is introduced from $S_{i,n}$. It is interesting to explore other concepts to include the phase information into the response. One of the concepts which is tried out in this study is based on the H_1 estimator. In this concept, the i[th] operational response is estimated as

$$a_i^h = \frac{S_{i,n}}{S_{n,n}} \qquad (5)$$

Using the response from equation (5), force identification is carried out on the earlier combination. The response predicted is shown in figure 14. The improvement observed is significant (rms error 0.7 dB). However, since this concept magnifies the response at antiresonances, which can create problems in assessing some of the techniques described in earlier reports, the concept based on (2) is used rather than (5).

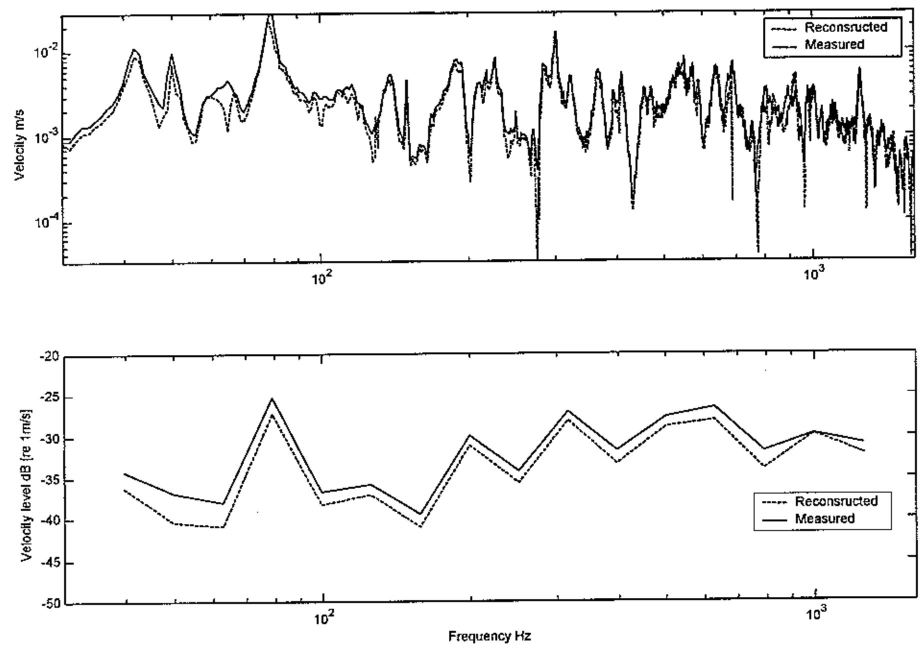

Figure 13. Receiver point response prediction based on Moore-Penrose pseudo inversion using responses from equation (2)

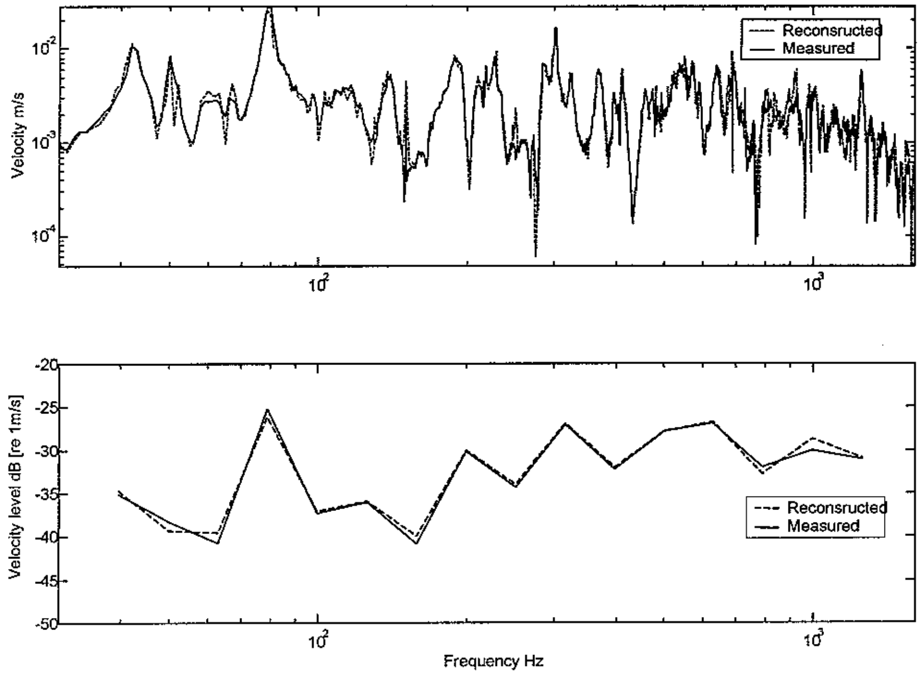

Figure 14. Receiver point response prediction based on Moore-Penrose pseudo inversion using responses from equation (5)

The root mean square errors over all one-third octave bands for the first set of measurements are shown in table 1 for 10 combinations of response points. The first column shows the 4 responses used for the force identification and the response that is predicted. For example, in the first case the positions 8, 10, 7 and 1 are used for force identification and the response at location 3 is predicted. The error in the predicted response and the errors in the contributions from each of the forces are shown in the table. The errors in force contributions 1 and 3 are obtained by comparing the results with those calculated using measured forces and the accelerance matrix as in section 2.3. The contribution from force 2 should be zero, so its predicted contribution is compared with the total response. This number should be as large as possible for good prediction.

It is found in this table that, due to the 1/3 octave band conversions, the variations in narrow band results are found to be averaged out (compare the narrow band and one-third octave band representations in figure 13). The variations between different combinations are consequently found to be minimal. To give more emphasis to the errors found in narrow bands, two other representations are also investigated: 1/12 octave band representation and constant bandwidth (15 Hz) representation. The results are shown in tables 2 and 3. Errors in forces are also included in these tables for more clarity. Since force 2 is zero, the identification error for force 2 is just the average magnitude of the dB level. This number should be as high as possible for a good prediction. The numbers indicated for the other forces are root mean square dB level errors and should be small for a good prediction.

Table 1. Average errors in contributions from each of the forces and the overall response in dB 1/3 octave bands (1st set of measurements)

Combination	**Error in force 1 contribution**	**Error in force 2 contribution**	**Error in force 3 contribution**	**Overall**
$a_8a_{10}a_7a_1$ - a_3	1.5	22.9	1.5	***1.6***
$a_8a_{10}a_7a_3$ - a_1	1.4	25.7	1.3	***1.9***
$a_8a_{10}a_1a_3$ - a_7	1.4	22.2	1.2	***1.9***
$a_8a_7a_1a_3$ - a_{10}	1.7	22.4	1.3	***1.6***
$a_{10}a_7a_1a_3$ - a_8	1.6	80.2	1.5	***1.5***
$a_{10}a_7a_1a_3$ - a_5	1.5	19.9	1.3	***1.2***
$a_{10}a_7a_1a_5$ - a_3	1.6	20.6	1.4	***1.4***
$a_{10}a_7a_3a_5$ - a_1	1.6	21.0	1.2	***1.7***
$a_{10}a_1a_3a_5$ - a_7	1.5	20.6	1.3	***1.9***
$a_7a_1a_3a_5$ - a_{10}	1.8	17.6	1.4	***1.5***

Table 2. Average errors in 1/12 octave representation in dB.

(1st set of measurements)

Combination	Force errors			Response errors (Contribution from forces)			
	F 1	F 2	F 3	***Overall***	error 1	error 2	error 3
$a_8a_{10}a_7a_1$ - a_3	2.0	34.6	1.5	***2.2***	2.1	24.9	1.8
$a_8a_{10}a_7a_3$ - a_1	2.2	34.9	1.6	***2.3***	2.2	26.9	1.5
$a_8a_{10}a_1a_3$ - a_7	2.1	33.9	1.5	***2.6***	2.2	23.6	1.6
$a_8a_7a_1a_3$ - a_{10}	2.1	33.9	1.6	***2.2***	2.5	25.2	1.6
$a_{10}a_7a_1a_3$ - a_8	2.1	33.0	1.5	***1.8***	2.3	83.5	1.7
$a_{10}a_7a_1a_3$ - a_5	2.1	33.0	1.5	***1.8***	2.0	25.8	1.7
$a_{10}a_7a_1a_5$ - a_3	1.9	33.2	1.5	***2.2***	2.2	23.2	1.8
$a_{10}a_7a_3a_5$ - a_1	1.9	33.2	1.5	***2.2***	2.0	24.7	1.4
$a_{10}a_1a_3a_5$ - a_7	1.8	32.0	1.5	***2.3***	2.1	21.2	1.6
$a_7a_1a_3a_5$ - a_{10}	1.9	32.3	1.4	***2.0***	2.1	23.4	1.6
$a_7a_1a_3a_5$ - a_8	1.9	32.3	1.4	***2.1***	2.1	82.3	1.5
$a_7a_1a_3a_8$ - a_5	2.1	33.9	1.6	***1.7***	2.0	26.5	1.7
$a_7a_1a_5a_8$ - a_3	2.3	33.9	1.6	***2.1***	2.4	24.5	1.8
$a_7a_3a_5a_8$ - a_1	2.4	34.0	1.5	***2.0***	2.3	26.9	1.6
$a_1a_3a_5a_8$ - a_7	2.3	31.7	1.7	***2.5***	2.4	21.5	1.8
$a_1a_3a_5a_8$ - a_{10}	2.3	31.7	1.7	***2.3***	2.5	23.4	1.8
$a_1a_3a_5a_{10}$ - a_8	1.8	32.0	1.5	***1.8***	2.2	82.2	1.6
$a_1a_3a_8a_{10}$ - a_5	2.1	33.9	1.5	***1.8***	2.3	26.3	1.7
$a_1a_5a_8a_{10}$ - a_3	2.1	32.6	1.6	***2.5***	2.3	23.8	1.9
$a_3a_5a_8a_{10}$ - a_1	2.1	33.3	1.5	***2.1***	2.1	25.6	1.4
$a_3a_5a_8a_{10}$ - a_7	2.1	33.3	1.5	***2.0***	2.2	23.5	1.6
$a_3a_5a_8a_7$ - a_{10}	2.4	34.0	1.5	***2.1***	2.6	26.1	1.7
$a_3a_5a_{10}a_7$ - a_8	1.9	33.2	1.5	***2.0***	2.1	83.4	1.6
$a_3a_8a_{10}a_7$ - a_5	2.2	34.9	1.6	***2.1***	2.3	27.8	1.7
$a_5a_8a_{10}a_7$ - a_3	2.1	35.4	1.6	***2.2***	2.2	25.9	1.9
$a_5a_8a_{10}a_7$ - a_1	2.1	35.4	1.6	***2.3***	2.2	27.7	1.6
$a_5a_8a_{10}a_1$ - a_7	2.1	32.6	1.6	***2.7***	2.3	22.9	1.8
$a_5a_8a_7a_1$ - a_{10}	2.3	33.9	1.6	***2.2***	2.5	25.6	1.8
$a_5a_{10}a_7a_1$ - a_8	1.9	33.2	1.5	***2.0***	2.2	83.3	1.7
$a_8a_{10}a_7a_1$ - a_5	2.0	34.6	1.5	***1.8***	2.1	27.1	1.7

Table 3. Average errors in constant 15 Hz band representation in dB.

(1st set of measurements)

Combination	Force errors			Response errors (Contribution from forces)			
	F 1	F 2	F 3	***Overall***	error 1	error 2	error 3
$a_8a_{10}a_7a_1$ - a_3	2.5	28.6	1.8	***1.9***	2.4	22.3	2.0
$a_8a_{10}a_7a_3$ - a_1	2.9	28.4	1.9	***2.8***	2.8	21.7	2.0
$a_8a_{10}a_1a_3$ - a_7	2.8	28.9	1.8	***3.0***	2.8	23.3	1.8
$a_8a_7a_1a_3$ - a_{10}	2.2	29.1	2.1	***2.3***	2.4	22.1	2.2
$a_{10}a_7a_1a_3$ - a_8	2.4	28.3	1.8	***2.0***	2.5	78.0	1.9
$a_{10}a_7a_1a_3$ - a_5	2.4	28.3	1.8	***2.4***	2.4	19.7	1.9
$a_{10}a_7a_1a_5$ - a_3	2.1	28.3	1.7	***2.3***	2.2	21.7	1.8
$a_{10}a_7a_3a_5$ - a_1	2.3	27.7	2.6	***2.6***	2.2	20.5	1.7
$a_{10}a_1a_3a_5$ - a_7	2.1	27.3	1.7	***2.7***	2.2	21.0	1.7
$a_7a_1a_3a_5$ - a_{10}	2.3	28.1	1.7	***2.2***	2.4	20.5	1.7
$a_7a_1a_3a_5$ - a_8	2.3	28.1	1.7	***2.2***	2.4	77.5	1.7
$a_7a_1a_3a_8$ - a_5	2.2	29.1	2.1	***2.3***	2.2	20.7	2.2
$a_7a_1a_5a_8$ - a_3	2.9	29.9	1.8	***2.5***	2.8	23.8	1.9
$a_7a_3a_5a_8$ - a_1	2.9	28.4	1.9	***3.0***	2.9	22.0	2.0
$a_1a_3a_5a_8$ - a_7	2.6	27.5	2.2	***3.5***	2.6	21.5	2.3
$a_1a_3a_5a_8$ - a_{10}	2.6	27.5	2.2	***2.4***	2.7	20.6	2.3
$a_1a_3a_5a_{10}$ - a_8	2.1	27.3	1.7	***2.1***	2.3	76.7	1.7
$a_1a_3a_8a_{10}$ - a_5	2.8	28.9	1.8	***2.8***	2.7	20.9	1.9
$a_1a_5a_8a_{10}$ - a_3	2.6	28.5	1.8	***2.5***	2.6	22.2	1.9
$a_3a_5a_8a_{10}$ - a_1	2.7	28.2	1.8	***2.6***	2.6	21.5	1.9
$a_3a_5a_8a_{10}$ - a_7	2.7	28.2	1.8	***2.6***	2.7	22.6	1.9
$a_3a_5a_8a_7$ - a_{10}	2.9	28.4	1.9	***2.9***	3.0	21.3	2.0
$a_3a_5a_{10}a_7$ - a_8	2.3	27.7	1.7	***2.2***	2.3	77.3	1.7
$a_3a_8a_{10}a_7$ - a_5	2.9	28.4	1.9	***2.5***	2.8	20.7	2.1
$a_5a_8a_{10}a_7$ - a_3	2.7	29.2	1.8	***2.0***	2.6	23.4	2.1
$a_5a_8a_{10}a_7$ - a_1	2.7	29.2	1.8	***2.6***	2.7	22.5	1.9
$a_5a_8a_{10}a_1$ - a_7	2.6	28.5	1.8	***3.0***	2.5	22.5	1.9
$a_5a_8a_7a_1$ - a_{10}	2.9	29.9	1.8	***2.2***	2.9	23.0	1.9
$a_5a_{10}a_7a_1$ - a_8	2.1	28.3	1.7	***2.3***	2.2	77.8	1.6
$a_8a_{10}a_7a_1$ - a_5	2.5	28.6	1.8	***2.6***	2.4	20.8	2.0

The representations in 1/12 octave bands and constant bandwidth bands are found to show the errors more distinctly than those in 1/3 octave bands (compare tables 1-3), although the results are still remarkably consistent. In the 1/12 octave band representation, the bandwidths are greater at higher frequencies. This means that the errors at each frequency are effectively weighted differently. Since this weighting matches the human perception of sound, when predicting the sound pressure at the receiver location, the 1/12 octave band representation is implemented. However, when predicting the vibration response or if only the force identification is of significance, then the errors are given equal weighting in each of the bands. This can be achieved by the constant bandwidth representation. Accordingly in this study these two representations are followed in estimating the errors. Out of all the combinations shown in table 3 (from constant bandwidth representation, as the vibration response is to be predicted), five are selected for investigating the techniques described in [5-7] from the first set of measurements. These five have equally poor force identification and the overall response prediction. These combinations are

i. $a_8a_{10}a_1a_3$ - a_7

ii. $a_{10}a_7a_3a_5$ - a_1

iii. $a_7a_3a_5a_8$ - a_1

iv. $a_1a_3a_8a_{10}$ - a_5

v. $a_3a_8a_{10}a_7$ - a_5

To investigate the techniques in relation to the sound pressure prediction, the third set of measurements is used. The errors in predictions for all five combinations are shown in table 4 (Response 1 is predicted in all the cases since it is the sound pressure response). From the table, two combinations are chosen for further investigation. As in the earlier case these combinations result in equally poor force identification and response prediction. These combinations are

i. $a_{10}a_7a_1a_3$ - a_p

ii. $a_1a_3a_5a_{10}$ - a_p

The second set measurements performed is not used in this study. However, these measurements will be used in future for studying the concept of selecting the best locations of accelerometer positions in identifying forces with minimum errors.

Table 4. Average errors in 1/12 octave band representation in dB.

(third set of measurements)

	Force errors			Response errors (Contribution from forces)			
Combination	F 1	F 2	F 3	***Overall***	error 1	error 2	error 3
$a_{10}a_7a_1a_3$ -a_p	2.9	31.0	2.4	***3.1***	2.9	21.6	2.4
$a_7a_1a_3a_5$ - a_p	2.3	30.1	2.1	***2.8***	2.3	20.6	2.1
$a_1a_3a_5a_{10}$ - a_p	2.3	29.1	2.0	***3.0***	2.3	19.6	2.0
$a_3a_5a_{10}a_7$ - a_p	2.3	30.3	2.2	***2.9***	2.3	20.6	2.1
$a_5a_{10}a_7a_1$ - a_p	2.4	31.4	2.3	***3.2***	2.6	21.3	2.4

3.2 Techniques used for inversion

From the combinations selected from the first and third sets of measurements, the forces are identified and the responses predicted using the following techniques, as described in [5-7]

- Moore-Penrose pseudo inversion
- Singular value rejection based on threshold selected by

  Error in accelerance matrix

  Error in acceleration measurement

- Perturbation of accelerance matrix
- Tikhonov regularization with ordinary cross validation
- Iterative inversion based on

  % of response used as standard deviation

  % of response used as standard deviation along with validation

  'Cross validation' approach

Even though two shakers are used for excitation, the third force with zero magnitude at all frequencies is added to make the force identification more realistic. By doing this the condition numbers of matrices can vary appreciably which is the case in practical problems. The problem to be solved is inversion of a 4 x 3 matrix. In Tikhonov regularization and iterative inversion with cross validation, however, a 3 x 3 inversion is performed with the fourth position used for cross validation.

3.3 Vibration response prediction

All 8 techniques are used in force identification and response predictions for the 5 combinations of response positions mentioned earlier. The results are tabulated in tables 5-7. Along with the tables, condition number plots are also given in Figures 15-19 for clarity. From the tables it is clear that the iterative inversion method combined with validation consistently results in better *response* predictions than all other techniques. Tikhonov regularization and the perturbation techniques result in predictions that are better than or the same as Moore-Penrose pseudo inversion. The technique based on singular value rejection is found to predict marginally better than Moore-Penrose pseudo inversion.

In *force identification*, Tikhonov regularization, Iterative inversion and the Perturbation technique are found to be consistently better than any other techniques.

To illustrate the results, the graphical results for combination $a_7a_3a_5a_8$ - a_1 are given in this report for all eight cases. These are discussed below.

Table 5. Errors from all the techniques for predicting a_7 from $a_8\,a_{10}\,a_1$ and a_3

	Force errors			Response errors			
Method	F 1	F 2	F 3	***Overall***	error 1	error 2	error 3
Moore-Penrose	2.8	28.9	1.8	***3.0***	2.8	23.3	1.8
Perturbation	2.4	26.4	1.3	***3.0***	2.4	21.9	1.3
Sv rej (accelerance)	2.8	28.8	1.8	***2.9***	2.8	23.1	1.9
Sv rej (response)	4.2	26.9	3.1	***2.8***	4.3	21.0	3.2
Tikhonov -OCV	2.3	27.6	1.3	***3.0***	2.4	23.2	1.3
Iterative - % of resp as std. deviation	2.6	26.4	2.3	***2.9***	2.6	22.0	2.3
Iterative - ocv - Variable std deviation	2.7	26.0	1.4	***2.5***	2.8	21.0	1.5
Iterative - ocv - Common standard deviation	2.9	26.1	1.7	***2.5***	2.9	21.2	1.6

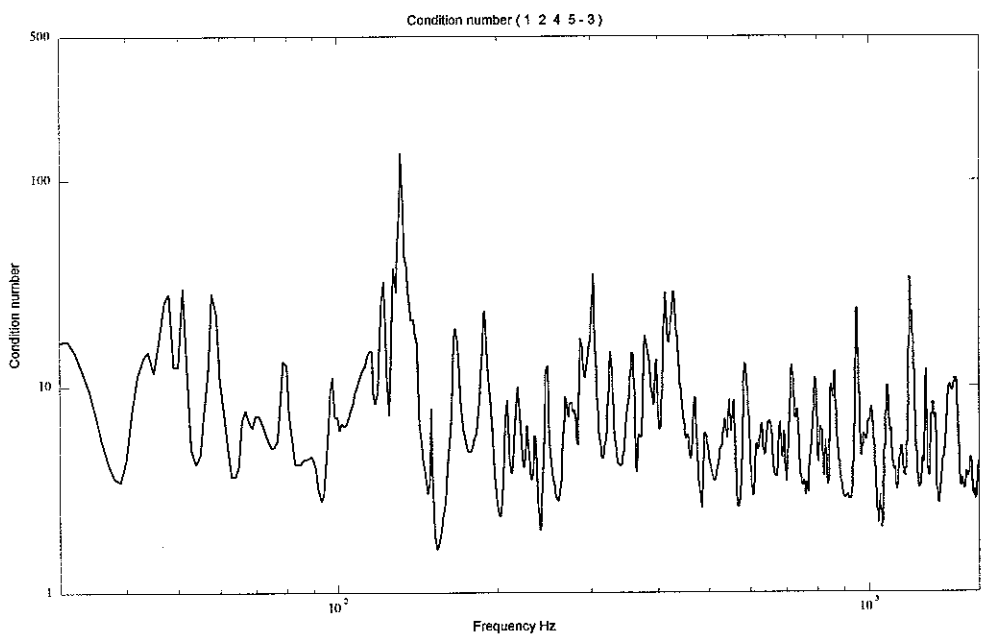

Figure 15. Condition numbers for the combination $a_8\,a_{10}\,a_1\,a_3$ - a_7

Table 6. Errors from all the techniques for predicting a_1 from $a_{10}\, a_7\, a_3$ and a_5

	Force errors			Response errors			
Method	F 1	F 2	F 3	***Overall***	error 1	error 2	error 3
Moore-Penrose	2.3	27.7	2.6	***2.6***	2.2	20.5	1.7
Perturbation	1.9	25.4	1.5	***2.3***	1.9	19.5	1.5
Sv rej (accelerance)	2.3	27.4	1.7	***2.6***	2.3	20.1	1.7
Sv rej (response)	3.5	24.2	4.1	***3.5***	3.8	16.7	4.2
Tikhonov -OCV	1.9	26.1	1.3	***2.3***	1.9	20.2	1.3
Iterative - % of resp as std. deviation	1.9	25.7	1.5	***2.3***	1.9	20.1	1.5
Iterative - ocv - Variable std deviation	1.9	25.5	1.3	***2.3***	1.9	19.7	1.3
Iterative - ocv - Common standard deviation	2.2	24.7	1.6	***2.3***	2.3	18.5	1.7

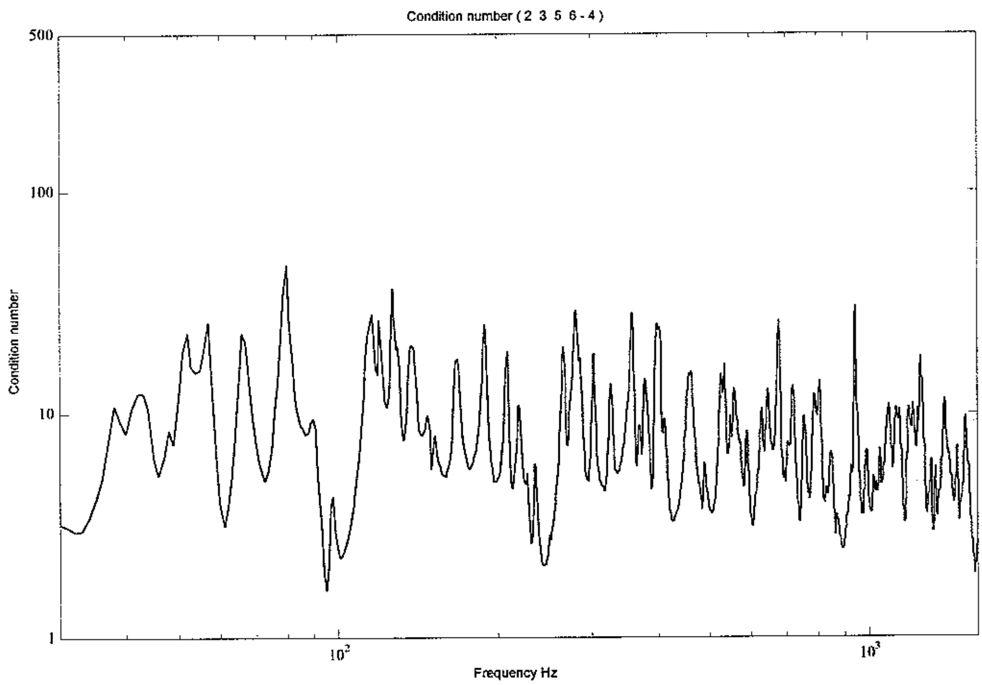

Figure 16. Condition numbers for the combination $a_{10}a_7a_3a_5$ - a_1

Table 7. Errors from all the techniques for predicting a_1 from $a_7\,a_3\,a_5$ and a_8

	Force errors			Response errors			
Method	F 1	F 2	F 3	***Overall***	error 1	error 2	error 3
Moore-Penrose	2.9	28.4	1.9	***3.0***	2.9	22.0	2.0
Perturbation	2.8	25.5	1.5	***2.8***	2.9	20.2	1.6
Sv rej (accelerance)	2.9	28.1	2.0	***3.0***	3.1	21.5	2.0
Sv rej (response)	3.0	24.5	3.7	***3.6***	3.3	17.3	3.8
Tikhonov -OCV	2.6	26.6	1.4	***2.6***	2.6	21.3	1.5
Iterative - % of resp as std. deviation	2.8	26.3	1.5	***2.9***	2.9	21.2	1.6
Iterative - ocv - Variable std deviation	2.3	24.9	1.7	***2.3***	2.4	18.9	1.8
Iterative - ocv - Common standard deviation	2.5	24.6	1.8	***2.3***	2.5	18.4	1.9

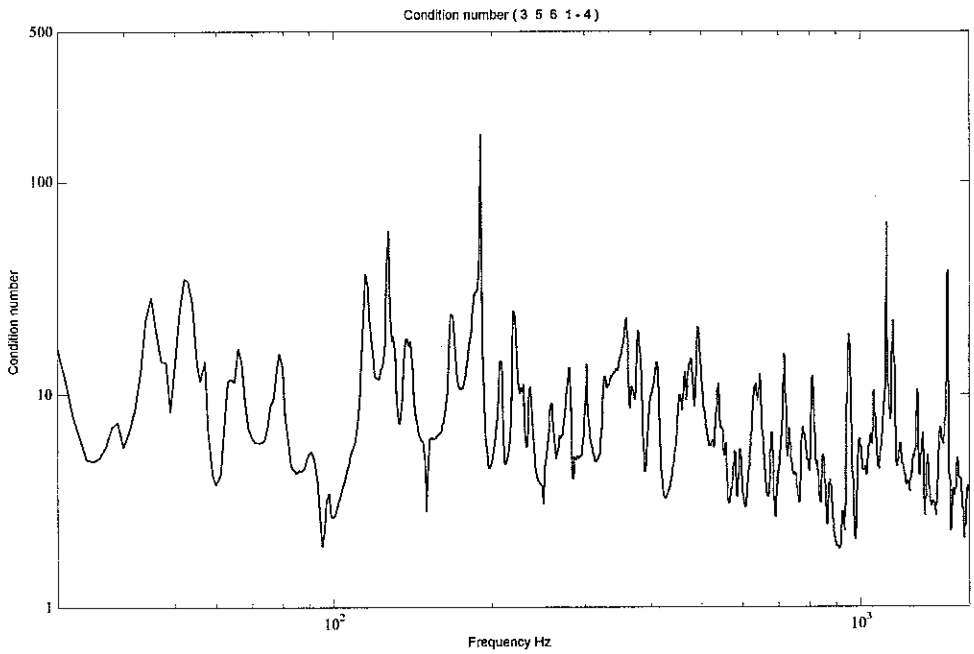

Figure 17. Condition numbers for the combination $a_7a_3a_5a_8$ - a_1

Table 8. Errors from all the techniques for predicting a_5 from a_1 a_3 a_8 and a_{10}

	Force errors			Response errors			
Method	F 1	F 2	F 3	***Overall***	error 1	error 2	error 3
Moore-Penrose	2.8	28.9	1.8	***2.8***	2.7	20.9	1.9
Perturbation	2.4	26.4	1.3	***2.5***	2.3	19.3	1.3
Sv rej (accelerance)	2.8	28.8	1.8	***2.8***	2.8	20.5	2.1
Sv rej (response)	4.2	26.9	3.1	***2.8***	4.0	18.0	3.6
Tikhonov -OCV	2.3	27.6	1.3	***2.5***	2.3	20.7	1.3
Iterative - % of resp as std. deviation	2.4	27.0	1.3	***2.6***	2.4	20.1	1.4
Iterative - ocv - Variable std deviation	2.7	26.2	1.4	***2.4***	2.7	18.6	1.6
Iterative - ocv - Common standard deviation	2.9	26.1	1.7	***2.3***	2.7	18.7	1.9

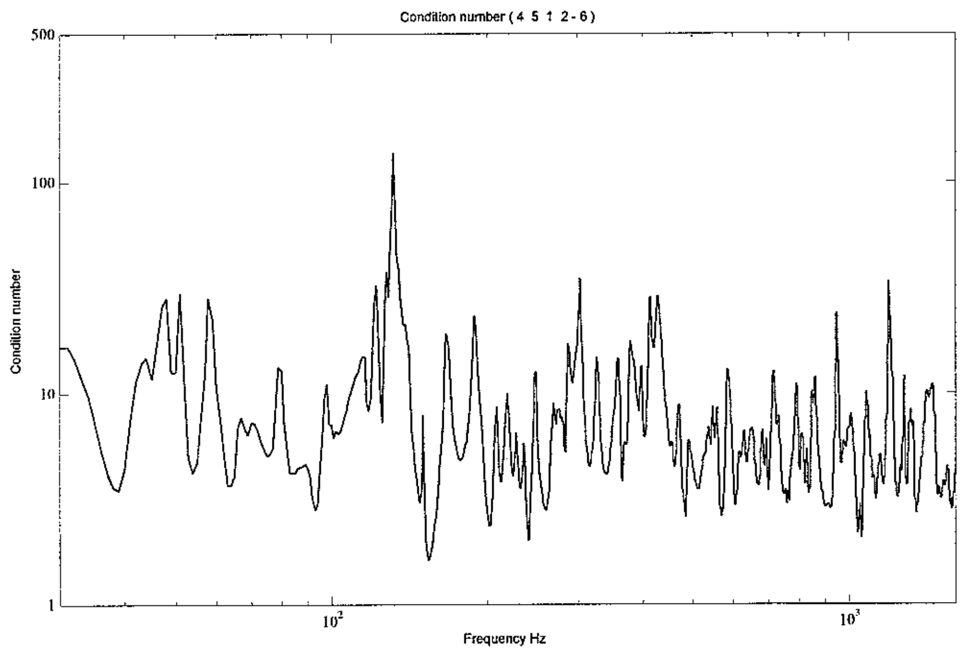

Figure 18. Condition numbers for the combination $a_1a_3a_8a_{10}$ - a_5

Table 9. Errors from all the techniques for predicting a_5 from a_3 a_8 a_{10} and a_7

Method	Force errors			Response errors			
	F 1	F 2	F 3	***Overall***	error 1	error 2	error 3
Moore-Penrose	2.9	28.4	1.9	***2.5***	2.8	20.7	2.1
Perturbation	2.5	25.4	1.4	***2.0***	2.5	18.7	1.5
Sv rej (accelerance)	2.9	28.3	2.0	***2.4***	2.9	20.3	2.2
Sv rej (response)	4.2	23.9	3.7	***2.6***	4.3	15.9	4.0
Tikhonov -OCV	2.5	26.9	1.4	***2.0***	2.5	20.1	1.5
Iterative - % of resp as std. deviation	2.6	26.6	1.4	***2.1***	2.6	19.9	1.6
Iterative - ocv - Variable std deviation	2.6	25.6	1.4	***2.0***	2.5	19.3	1.6
Iterative - ocv - Common standard deviation	2.6	25.2	1.6	***2.0***	2.6	18.2	1.8

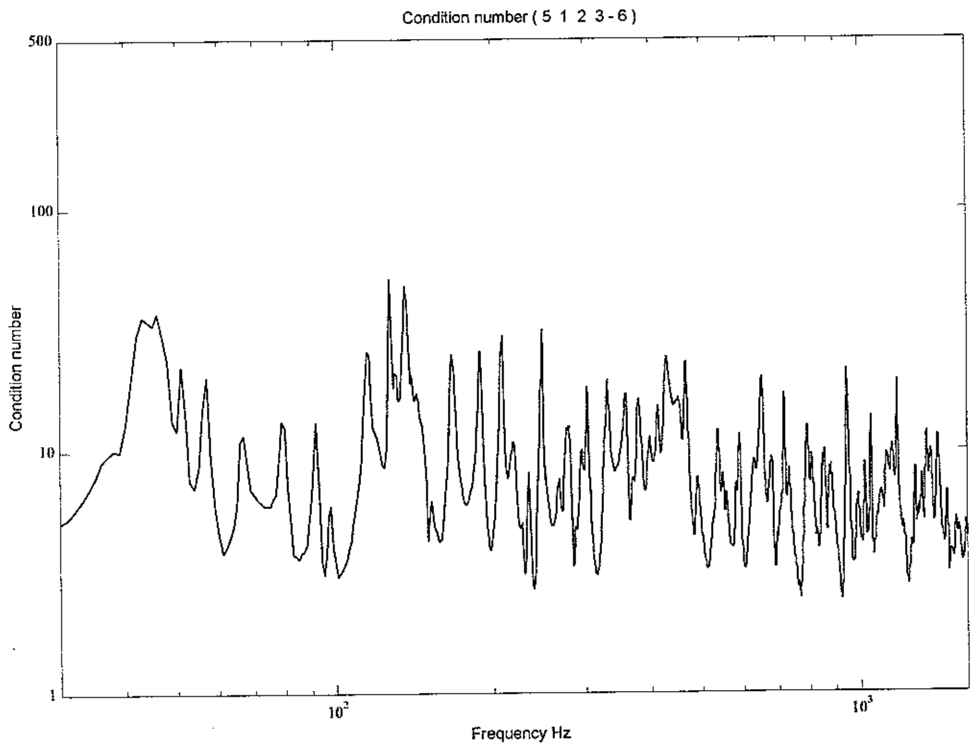

Figure 19. Condition numbers for the combination $a_3a_8a_{10}a_7$ - a_5

Figure 20 shows results for Moore-Penrose pseudo inversion. Forces 1 and 2, which are smaller than the force 3, are identified with comparatively large errors (top two plots in Figure 20a). Spiky behaviour is observed in the high frequency region. The error contribution of each of the forces to the overall responses follows the errors in forces (Figure 20b). The error in the contribution from force 3 is lowest as it is the largest force in most of the frequency range (bottom most plot in Figure 20b). The errors in overall response prediction are more concentrated in the high frequency region (Figure 20c). The benefit of using a constant bandwidth representation is clear from variations depicted in high frequencies which are very distinctive.

The perturbation technique results in much better identification of force 3, while the errors in forces 1 and 2 remain as before (Figure 21a). The spiky force identification at high frequencies is again found in this case. Similar error trends are observed in the contributions to the response (Figure 21b). Since the largest force is identified much better, the errors in overall response are found to be small compared to Moore-Penrose pseudo inversion (compare Figure 20c and Figure 21c).

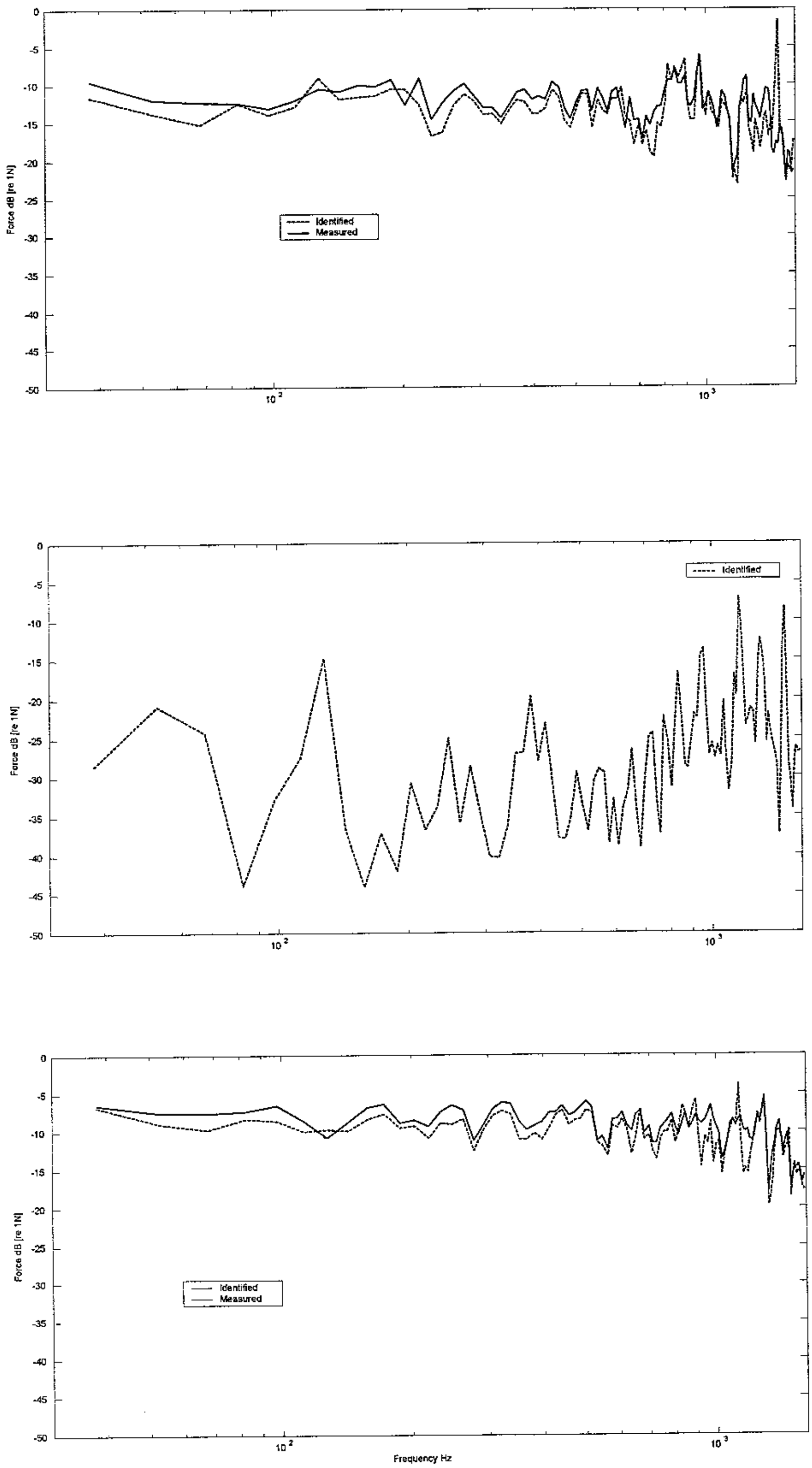

Figure 20a. Identified forces by Moore-Penrose pseudo inversion

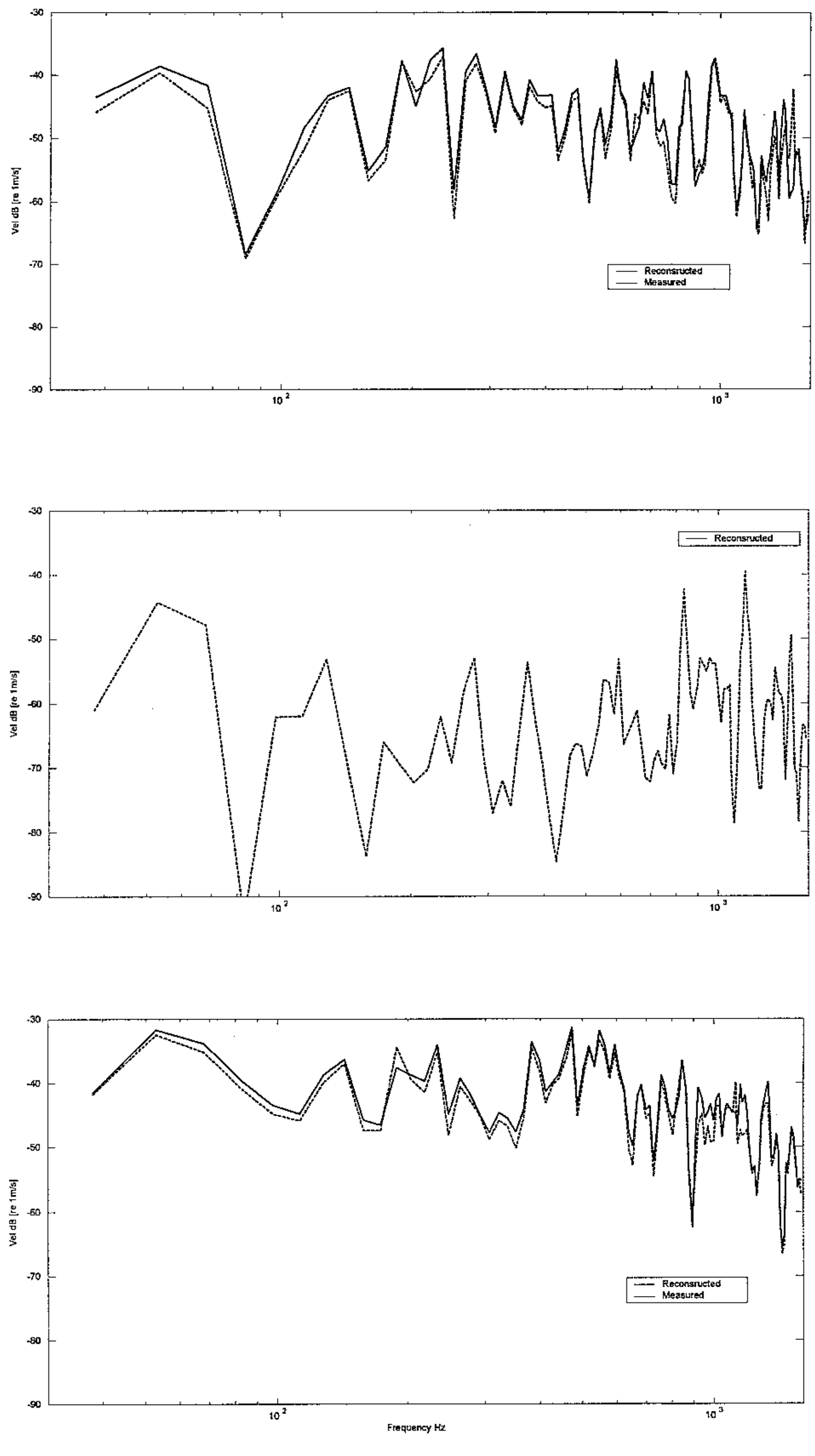

Figure 20b. Contribution from each force by Moore-Penrose pseudo inversion

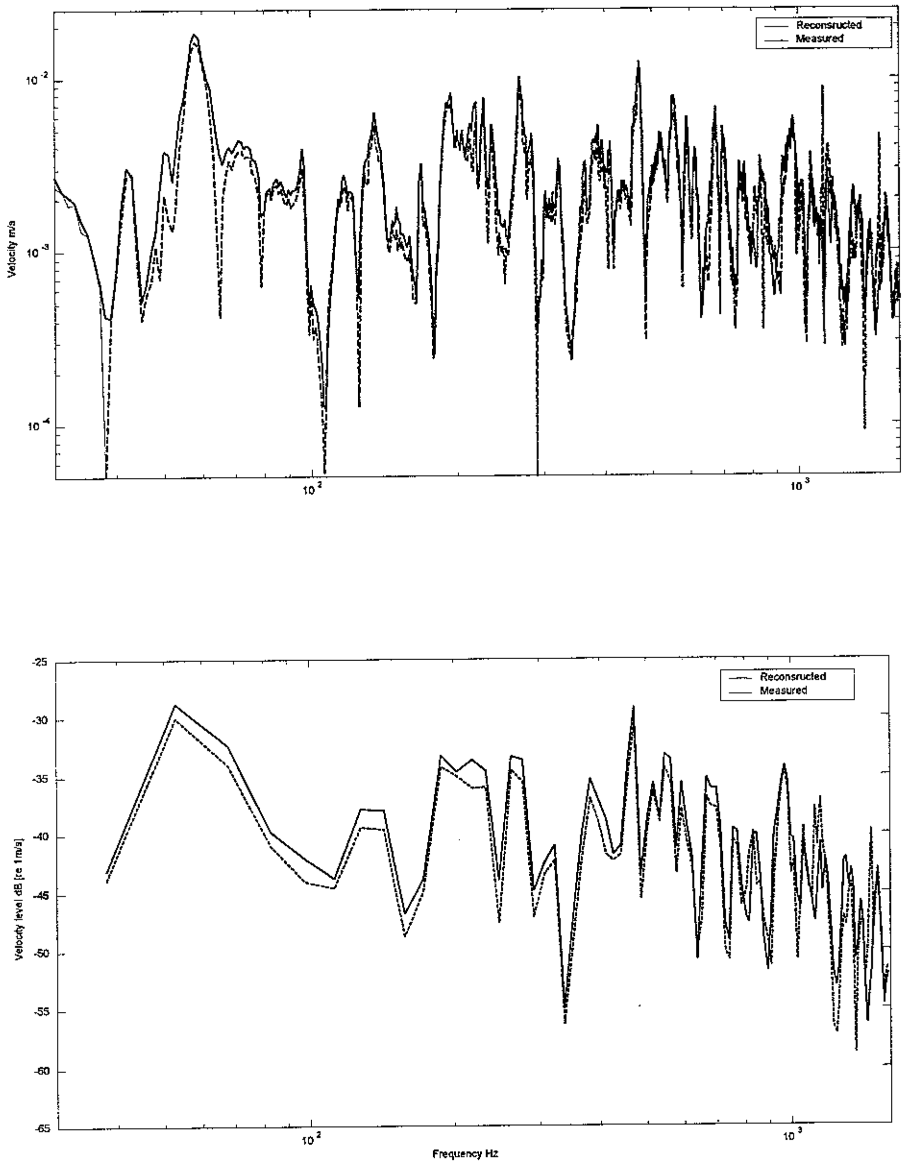

Figure 20c. Overall contribution by Moore-Penrose pseudo inversion in narrow band and 15 Hz bandwidth forms.

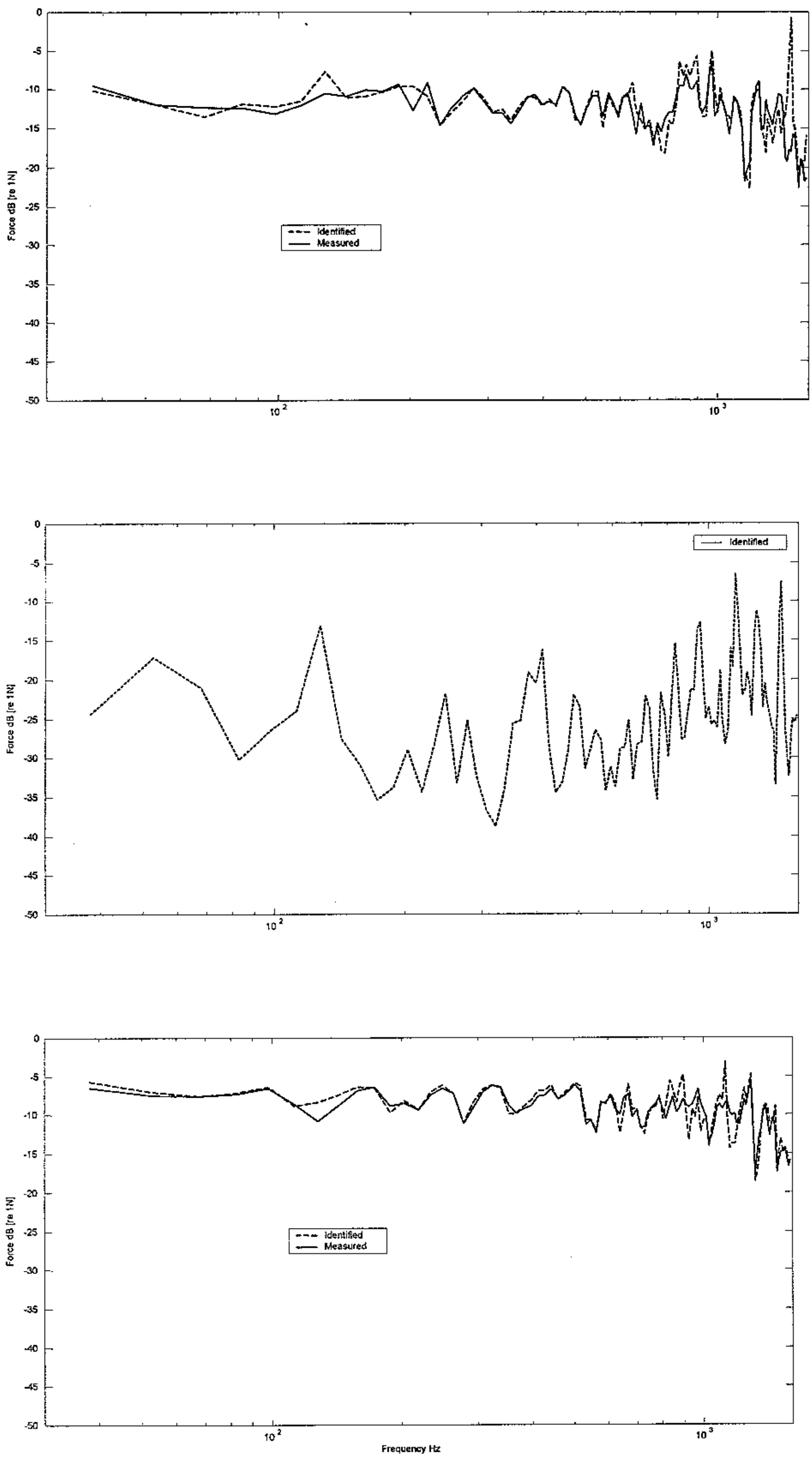

Figure 21a. Identified forces by Perturbation technique

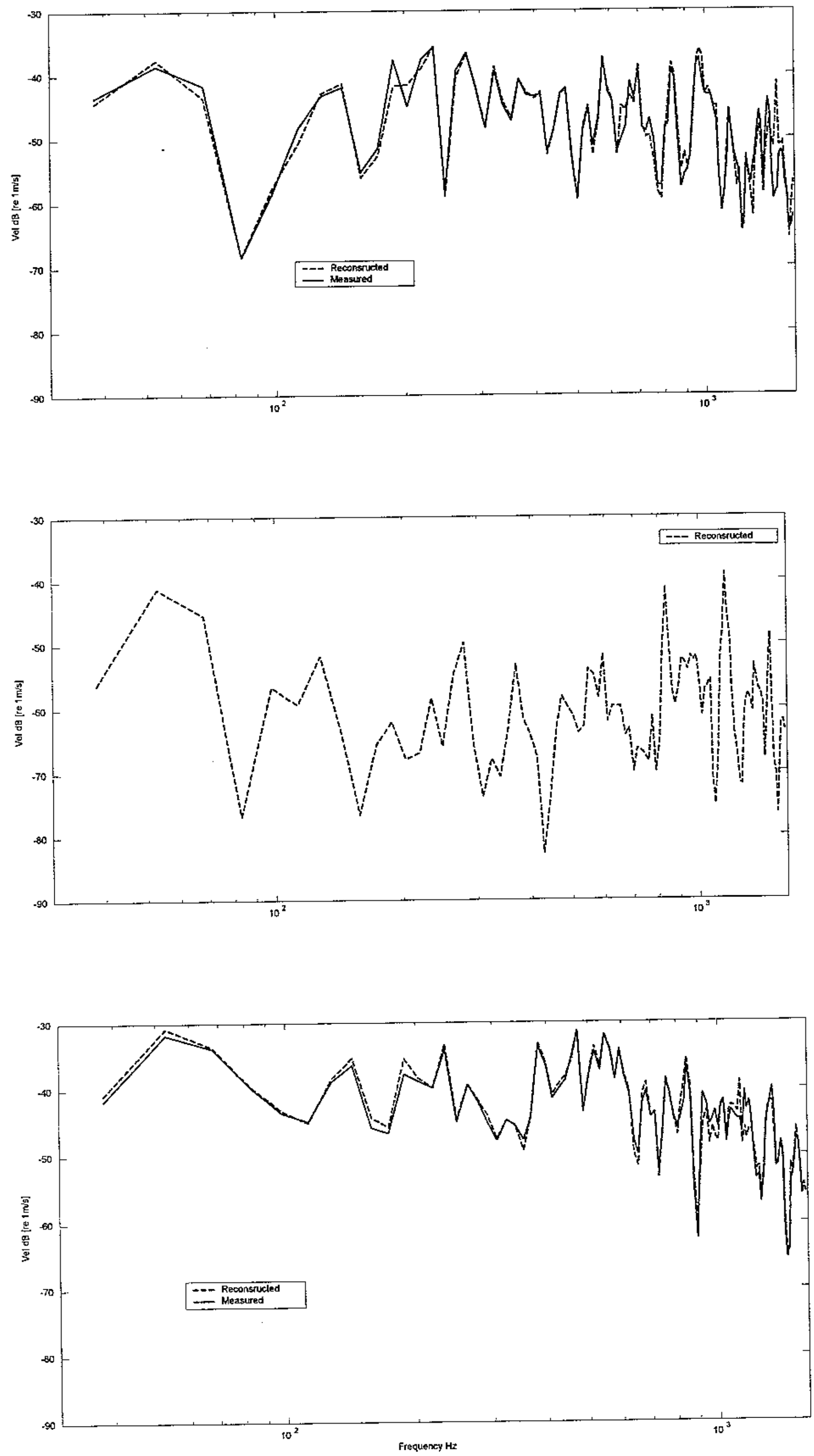

Figure 21b. Contribution from each force by Perturbation technique

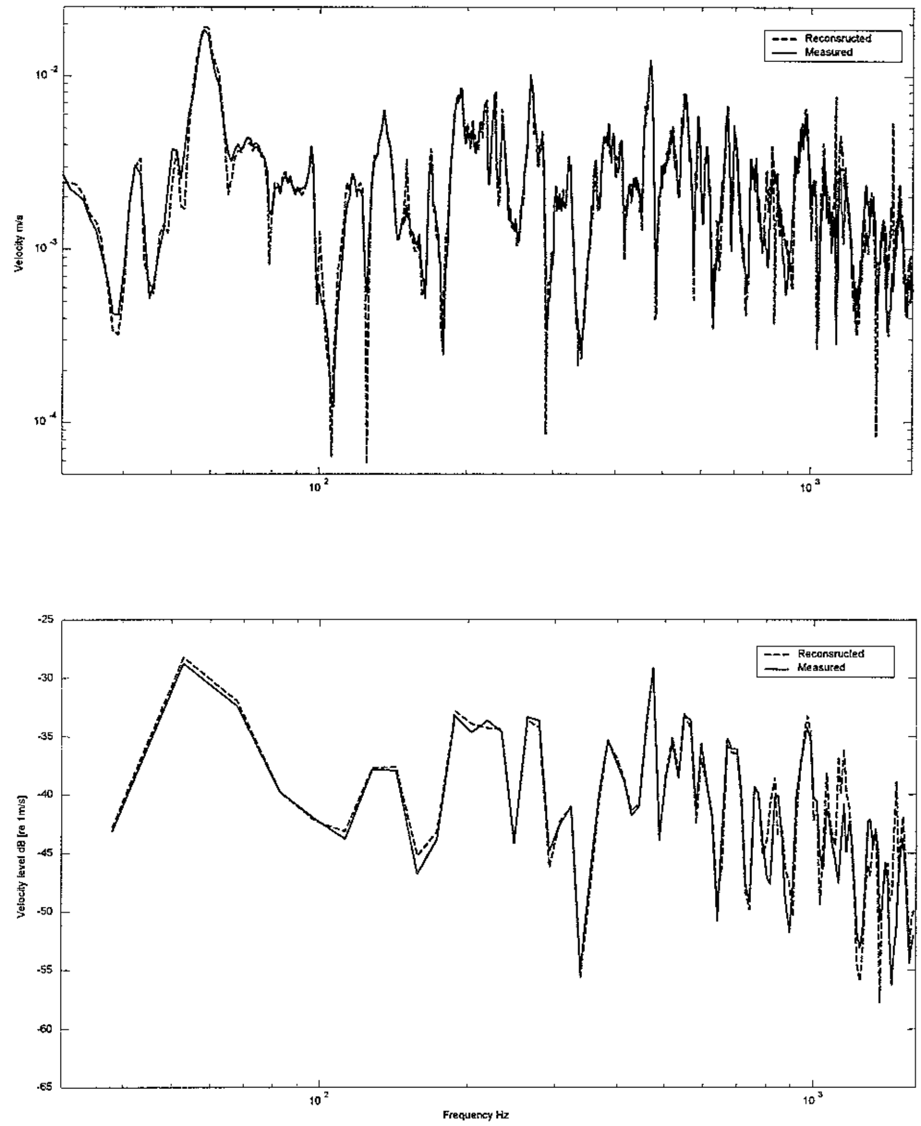

Figure 21c. Overall contribution by Perturbation technique in narrow band and 15 Hz bandwidth forms.

The force identification by singular value rejection based on the norm of the error in accelerance matrix is not improved compared to the Moore-Penrose pseudo inverse (Compare Figure 20a and 22a, and table 7). At high frequencies the error in FRF estimation is very small (since the coherence is large, Figure 8), which does not result in the rejection of any singular values at most of the frequencies. Hence the force identification from Moore-Penrose and singular value rejection by this way do not differ much here. Similar trends are found for the contributions from different forces (Figure 22b) and the overall response prediction (Figure 22c).

The singular value rejection method based on the response error norm is found to under-estimate the forces at most of the frequencies (Figure 23a). But this method eliminates the spiky behaviour of identified forces in the high frequency region seen in figure 20a. However, due to the under-estimation at many frequencies, large root mean square errors are found (table 7). The force contributions to the response also follow the same trend i.e. under-estimated (Figure 23b). The spikes observed earlier for the response prediction at some frequencies disappear in this case (See narrow band representation in Figure 23c). From the 15 Hz band representation, the response is found to be under predicted at most of the frequencies and spikes are eliminated. The under-estimation is very low at peaks of the response. Due to this feature one might think of using this method for response prediction since response at anti-resonance may not matter so much.

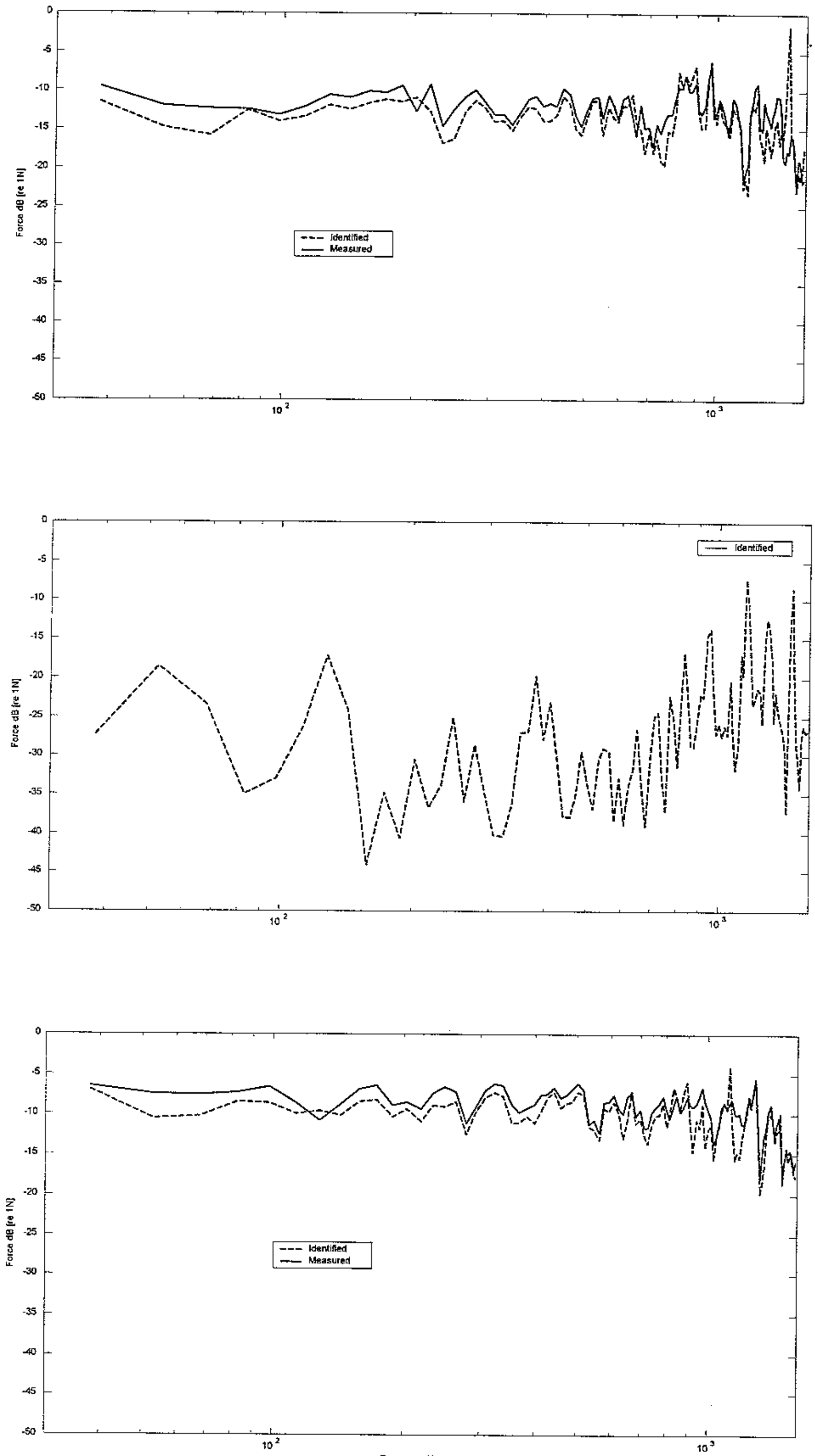

Figure 22a. Identified forces by Singular value rejection based on accelerance error

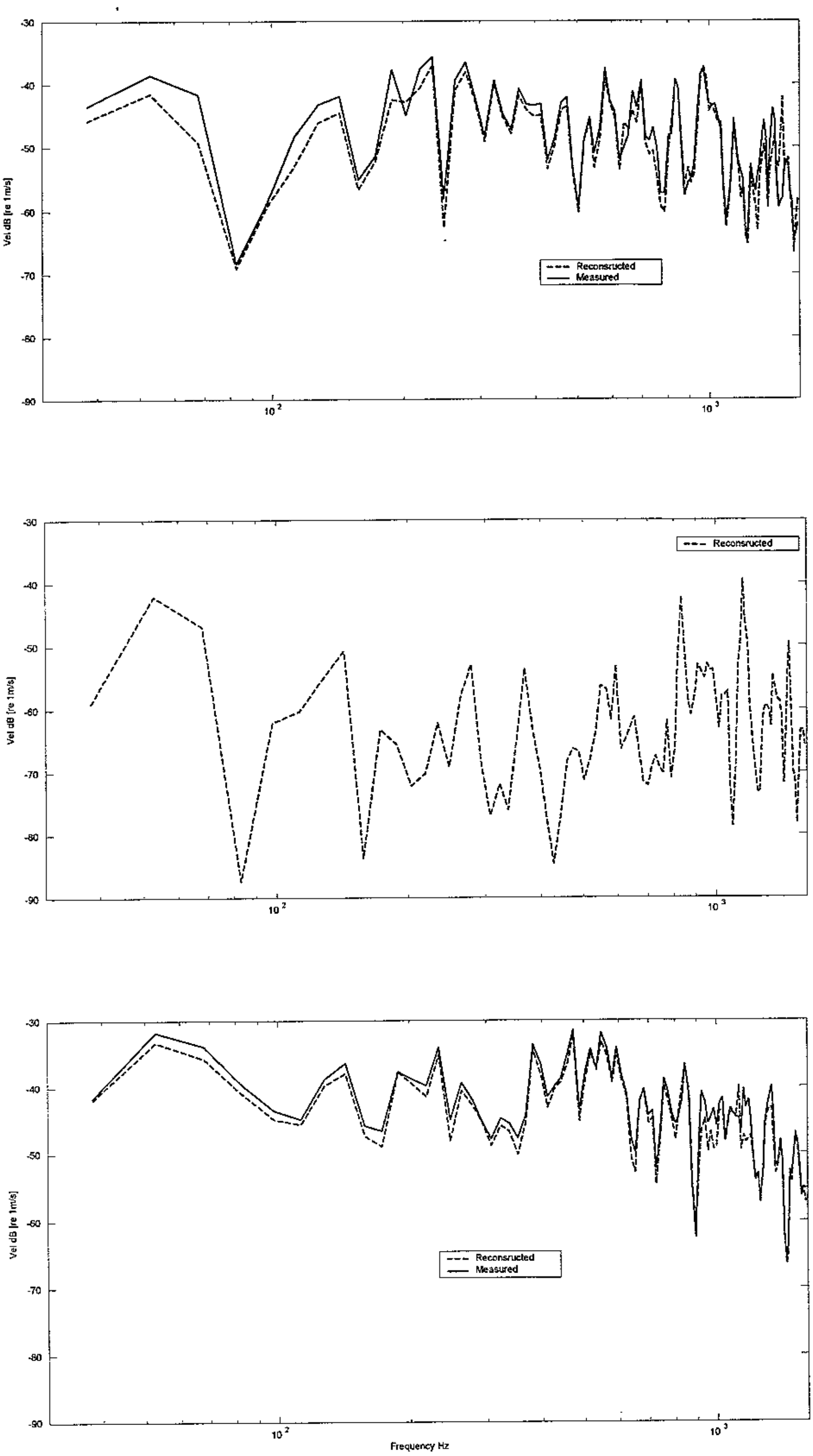

Figure 22b. Contribution from each force by Singular value rejection based on accelerance error

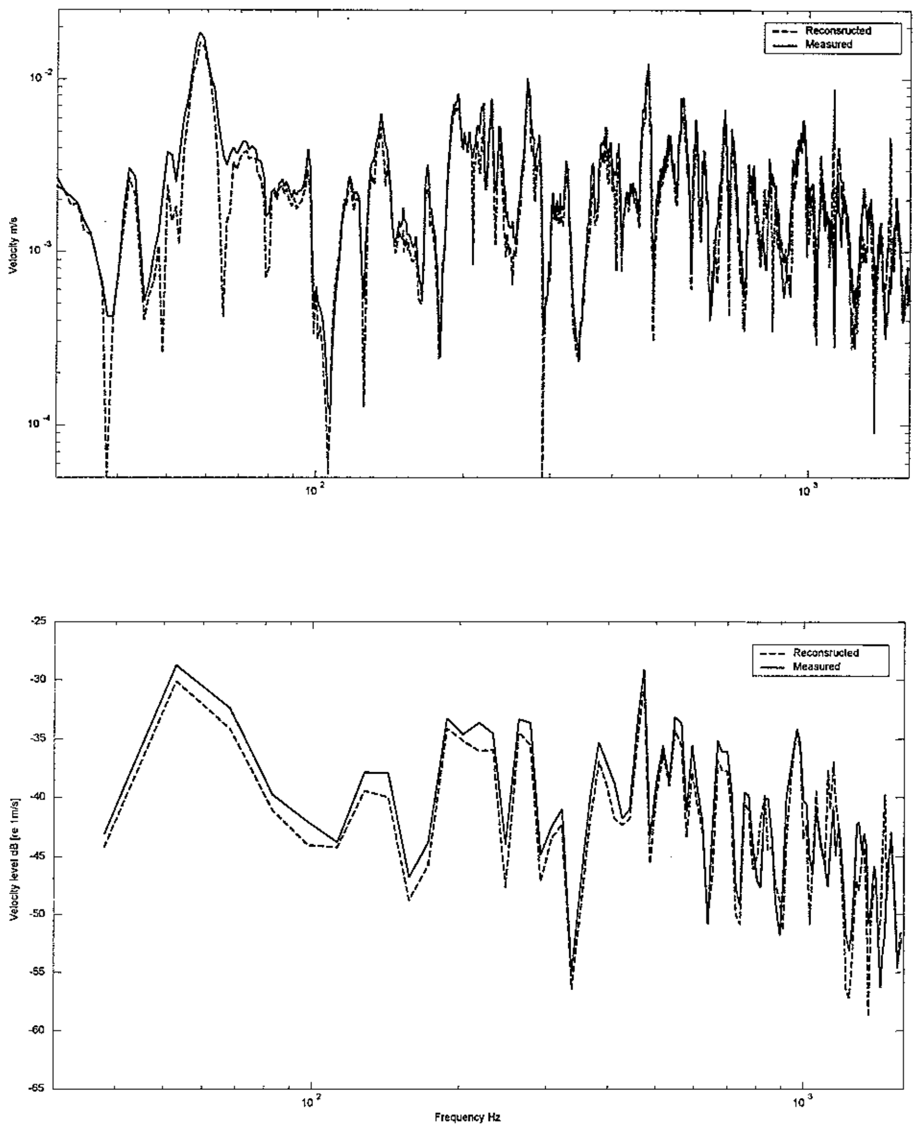

Figure 22c. Overall contribution by Singular value rejection based on accelerance error in narrow band and 15 Hz bandwidth forms.

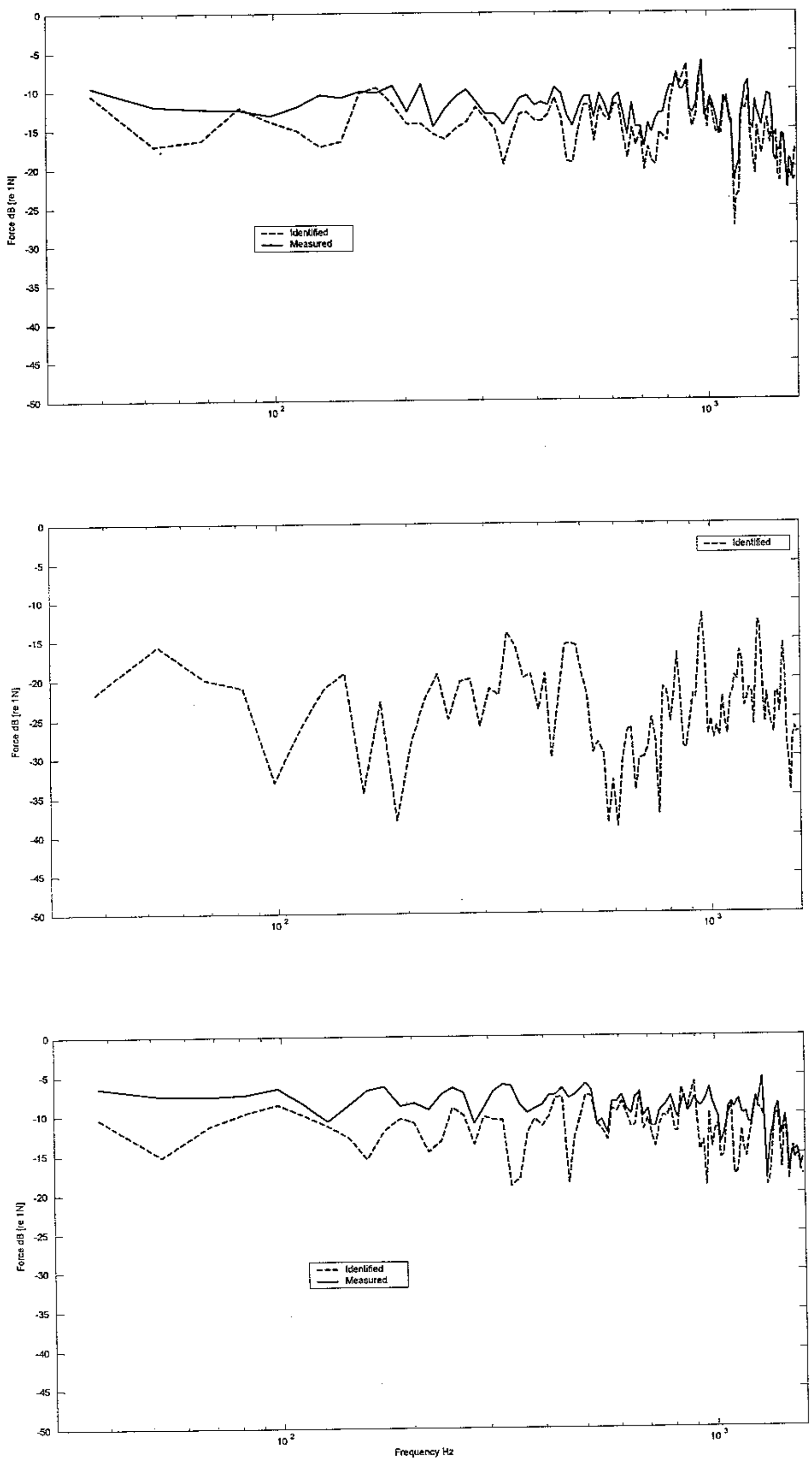

Figure 23a. Identified forces by Singular value rejection based on response error

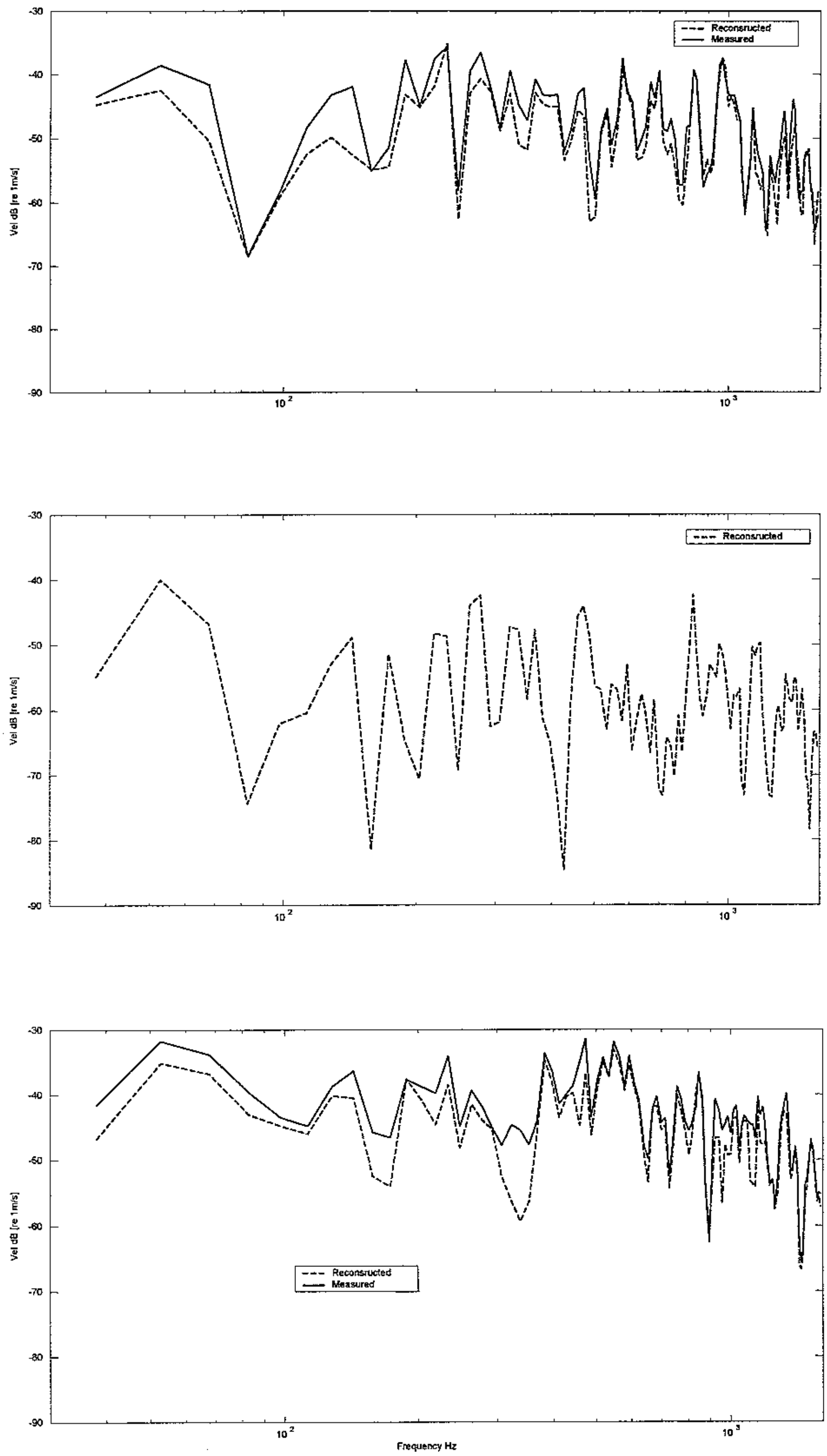

Figure 23b. Contribution from each force by Singular value rejection based on response error

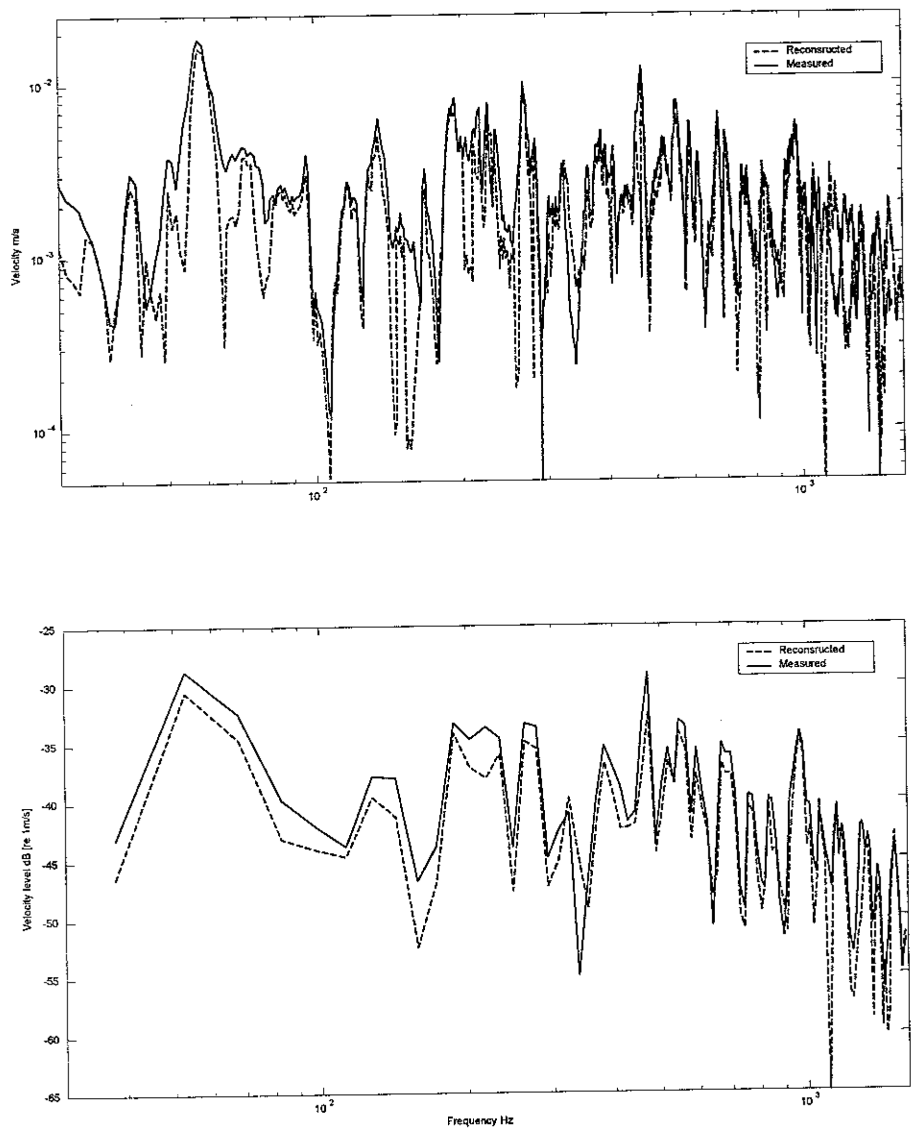

Figure 23c. Overall contribution by Singular value rejection based on response error in narrow band and 15 Hz bandwidth forms.

The Tikhonov regularization along with ordinary cross validation for the selection of regularization parameter results in better force identification at low frequencies (Figure 24a) than the Moore-Penrose pseudo inverse. Force 3 is found to be best predicted by this method (table 7). The magnitudes of spikes at high frequencies have also been reduced (compare Figure 20a and 24a). The contribution to the overall response from force 3 is very well predicted, since force 3 is identified accurately (Figure 24b). Improvement is also observed in the contribution from force 1. Due to these improvements in force contributions, the overall response is predicted with less error in this case compared to the earlier methods (Figure 24c).

The iterative inversion considering a fixed percentage of response as standard deviation (in this case it is taken to be the random error in the estimation of response which is inversely proportional to the square root of the number of averages), reduces the errors in force identification compared to Moore-Penrose pseudo inverse (Figure 25a). However, it is inferior to Tikhonov regularization. A similar trend is found for the force contribution errors and overall response errors (Figure 25b and 25c).

The iterative inversion with validation, considering a variable percentage of the response as the standard deviation (non-common standard deviation) results in much superior force identification at high frequencies (Figure 26a). The magnitudes of spikes are reduced. This, however, results in under-estimation of some forces and hence the force errors are increased compared to Tikhonov regularization. This is also clear from the force contribution errors (Figure 26b). Since the singular values are modified based on the error in the response and predictions are validated using one of the responses, the overall response is predicted much better than all other methods

(Figure 26c). For calculations the standard deviation is allowed to vary between 5% and 75% of the response in steps of 5%.

The above method with a common standard deviation also results in similar errors (Figure 27a-c).

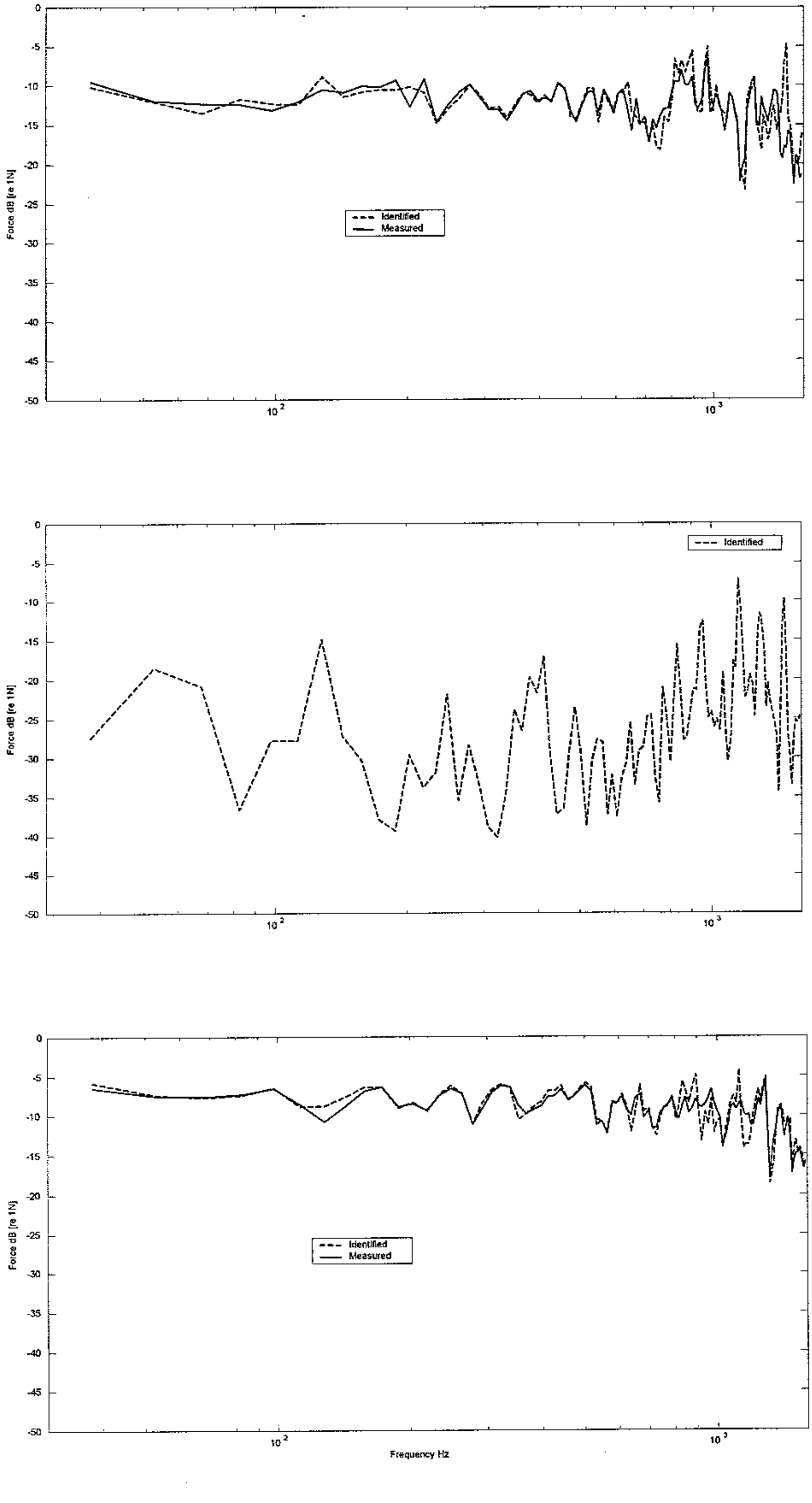

Figure 24a. Identified forces by Tikhonov regularization

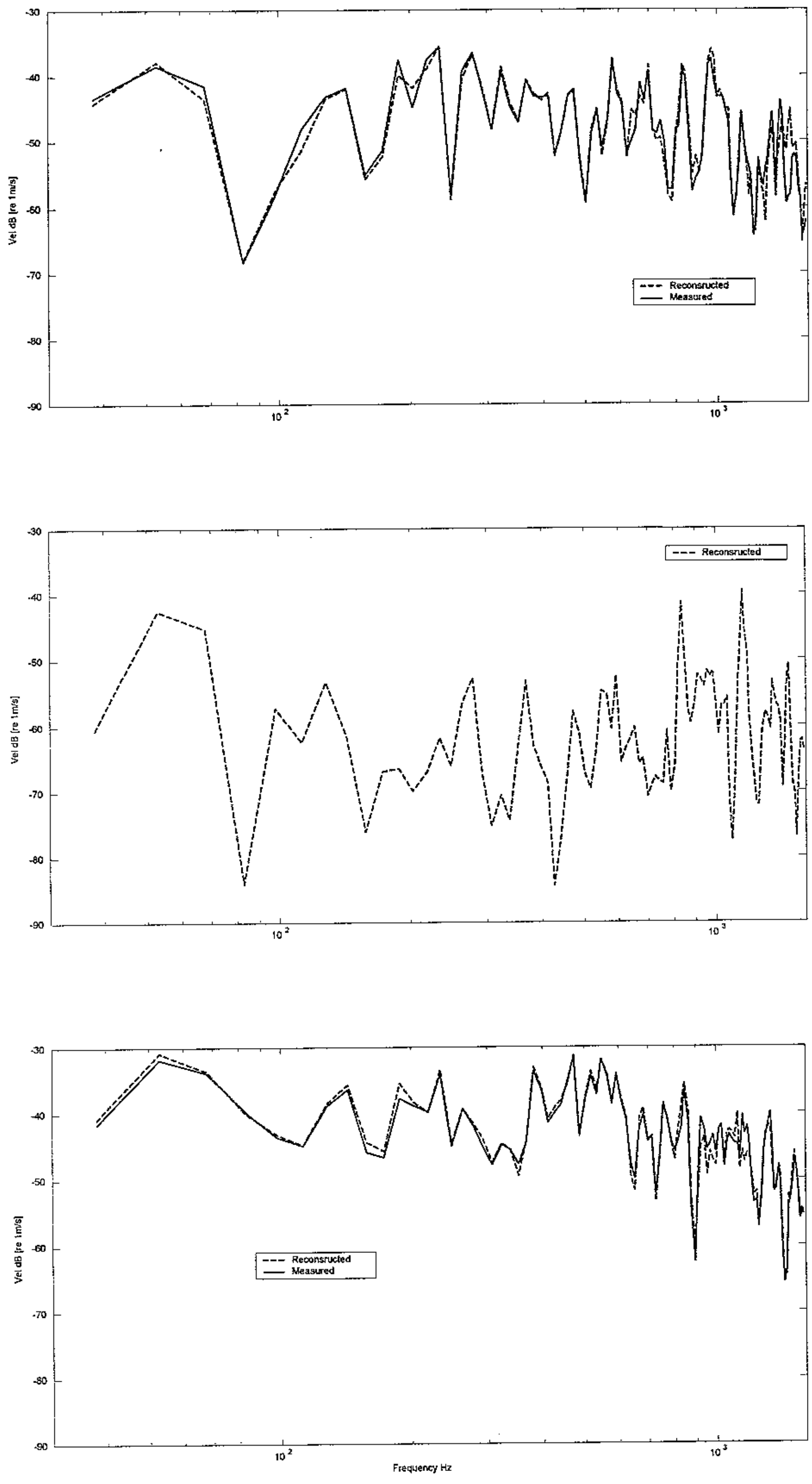

Figure 24b. Contribution from each force by Tikhonov regularization

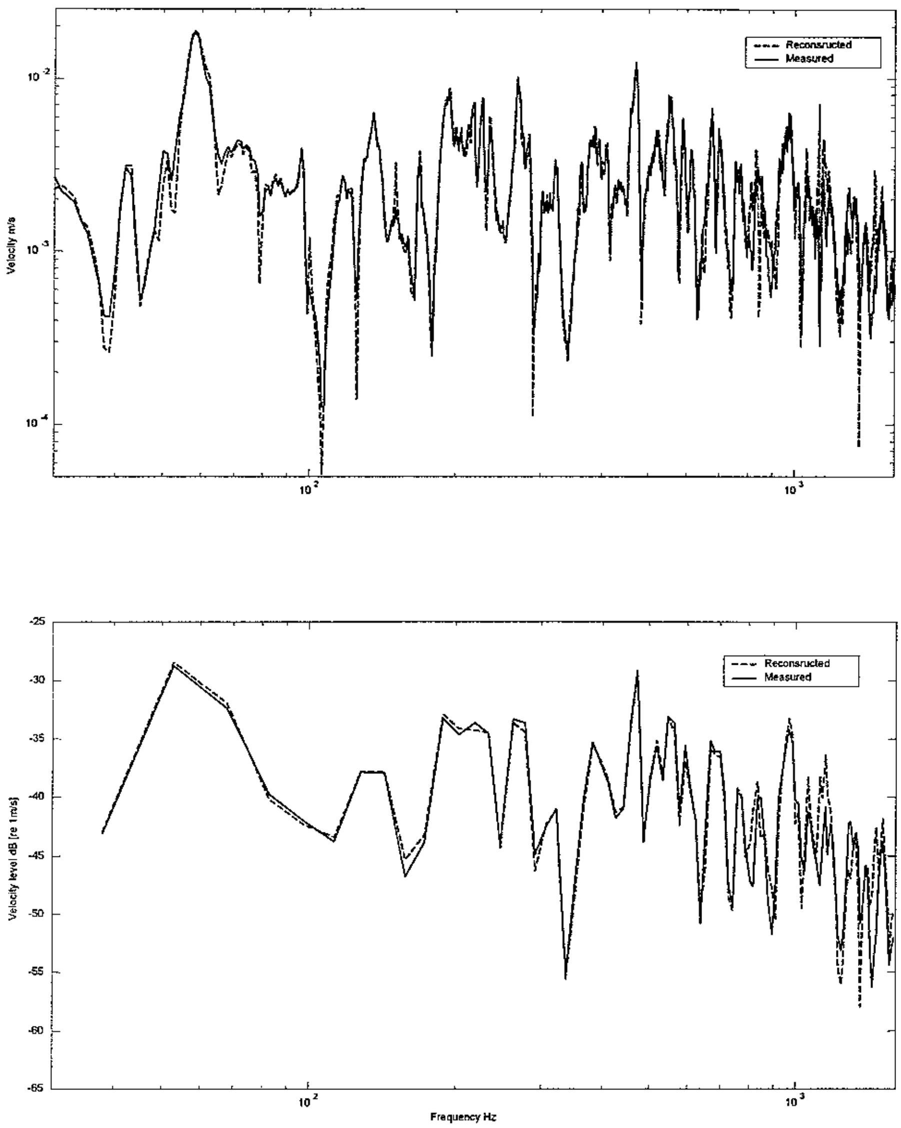

Figure 24c. Overall contribution by Tikhonov regularization in narrow band and 15 Hz bandwidth forms.

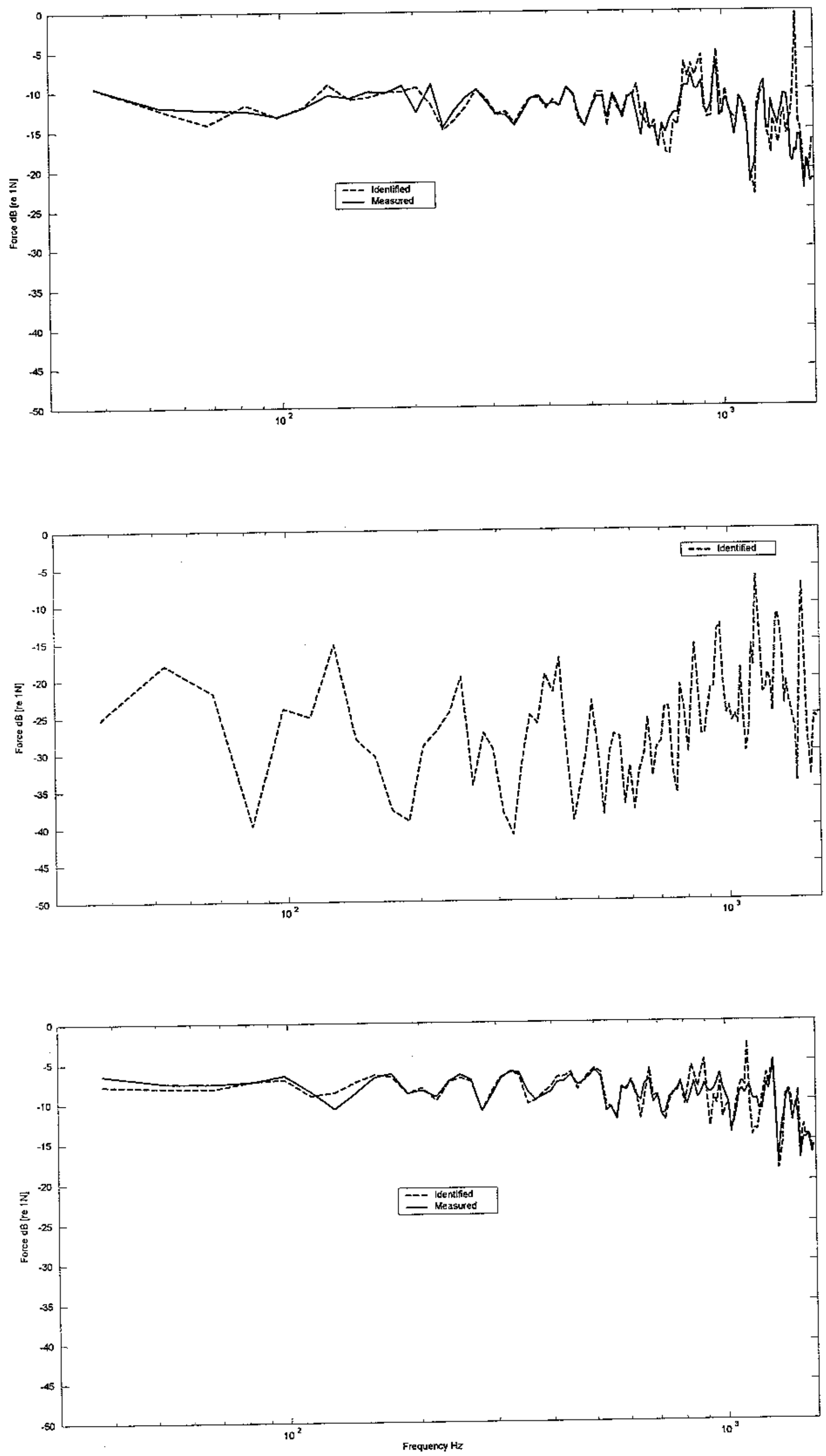

Figure 25a. Identified forces by Iterative inversion- Standard deviation as fixed % of response

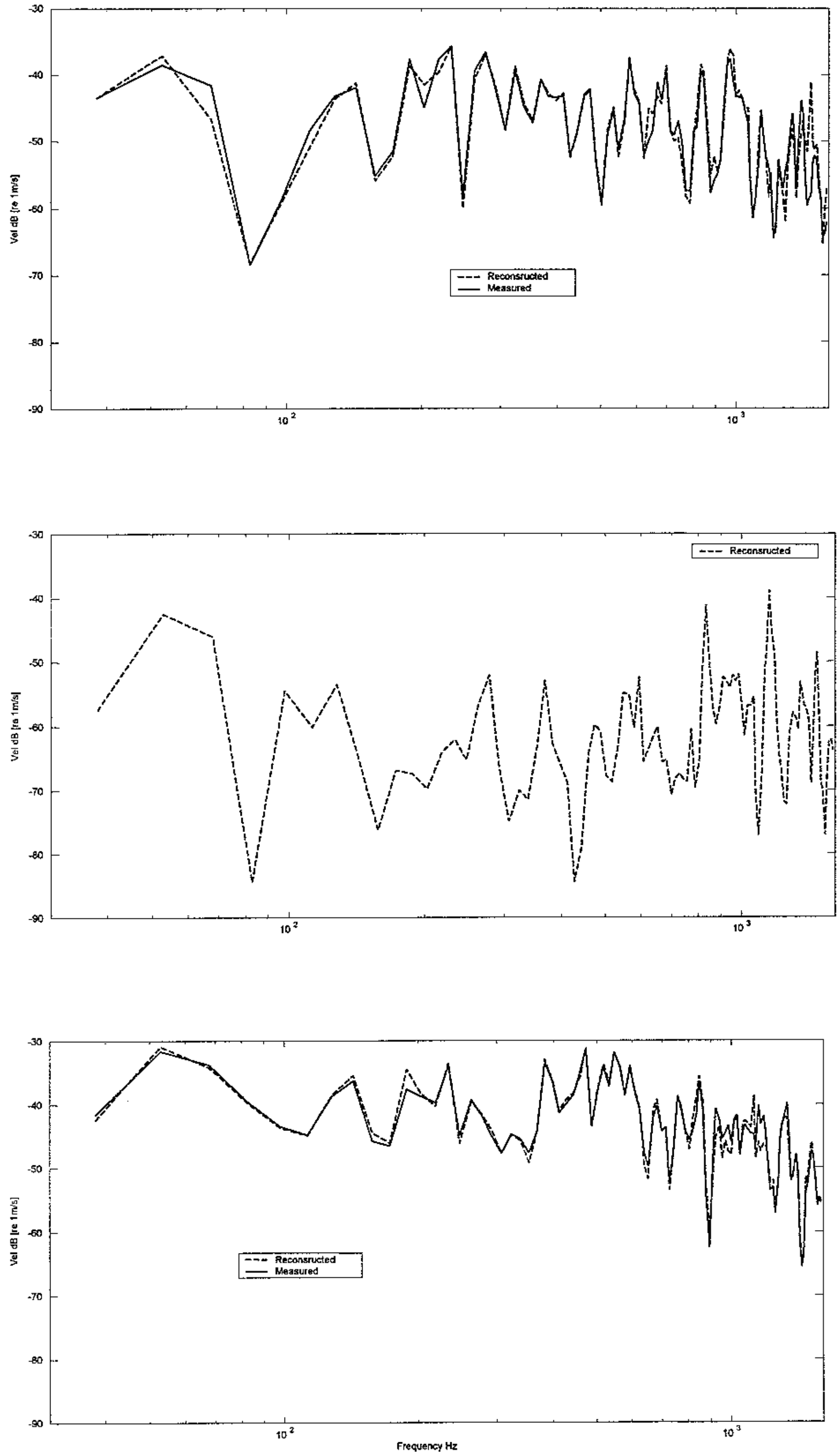

Figure 25b. Contribution from each force by Iterative inversion- Standard deviation as fixed % of response

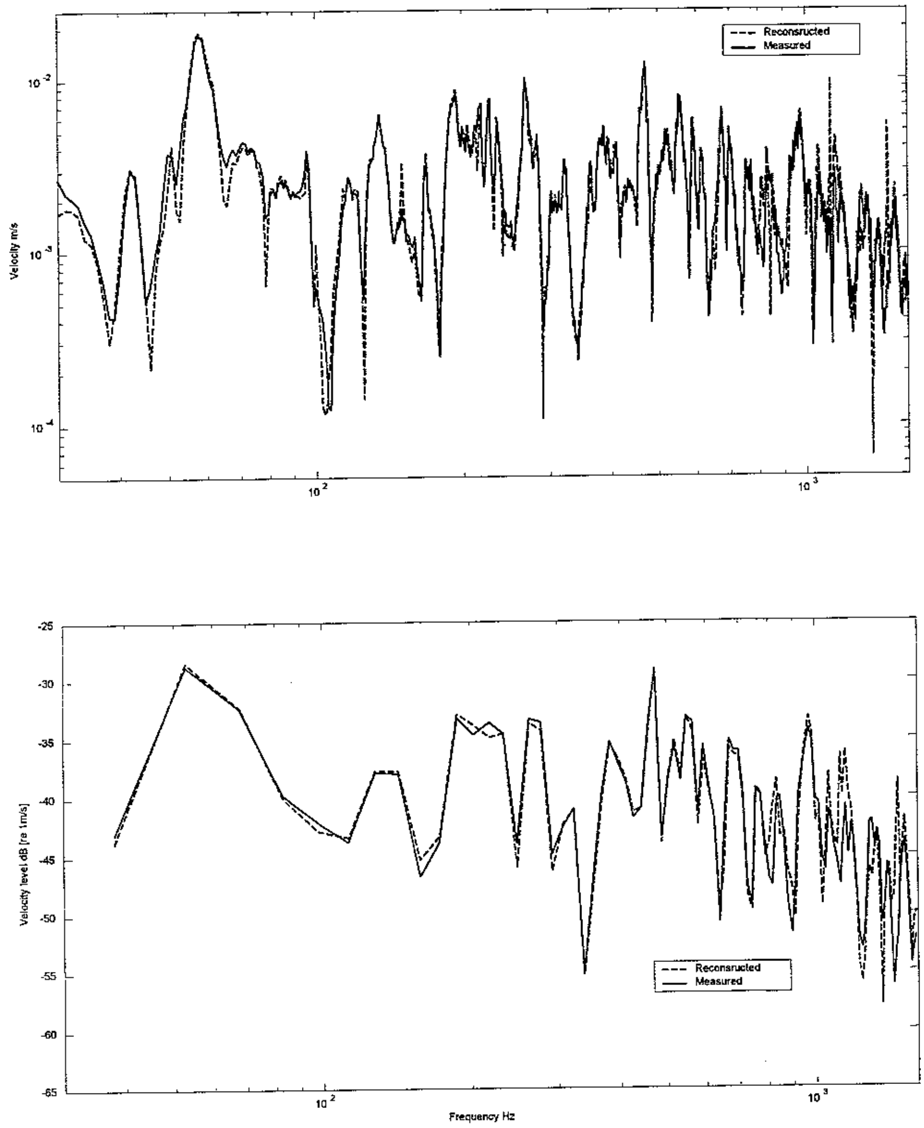

Figure 25c. Overall contribution by Iterative inversion in narrow band and 15 Hz bandwidth forms - Standard deviation as fixed % of response.

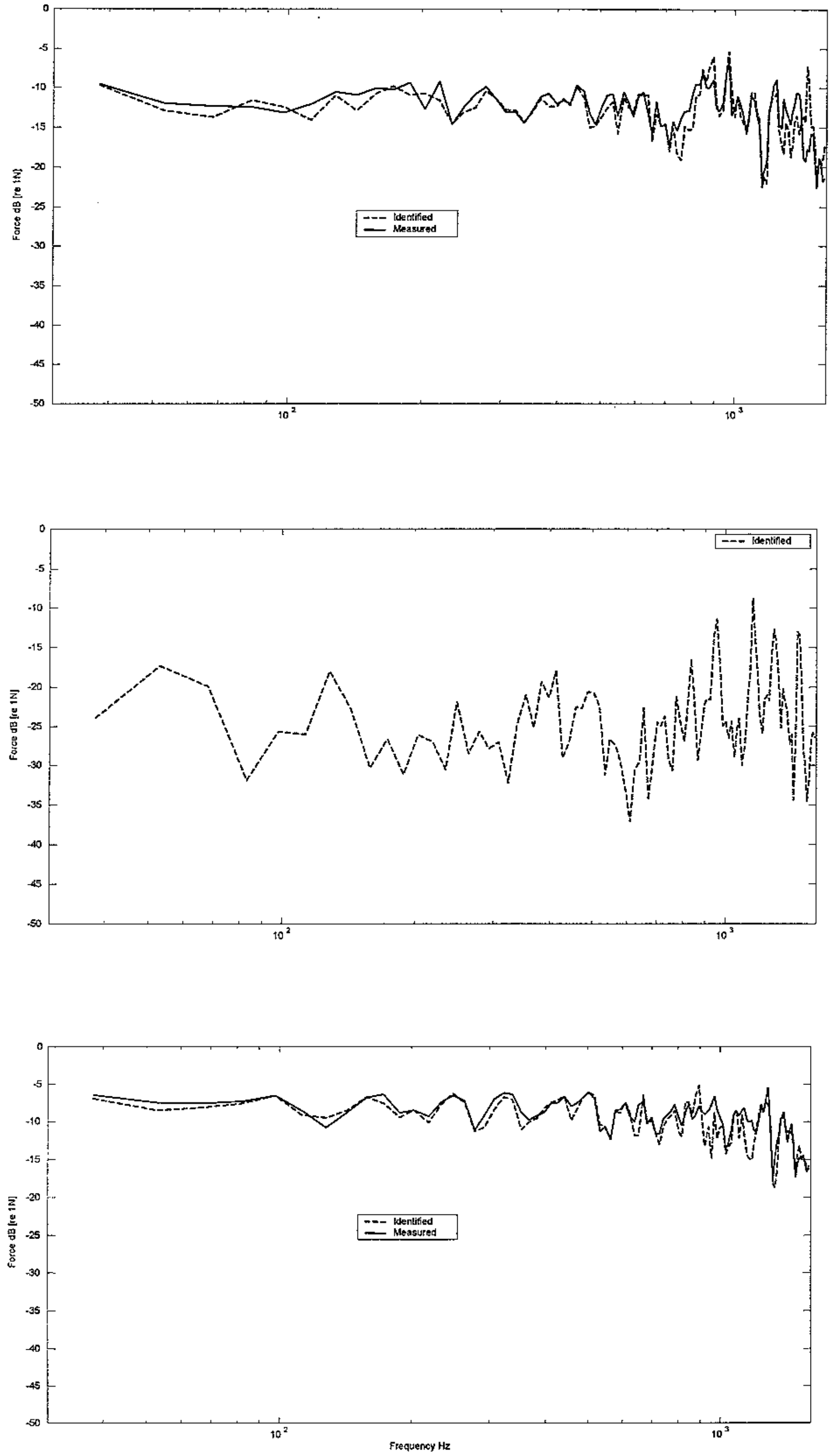

Figure 26a. Identified forces by Iterative inversion with validation - Variable standard deviation

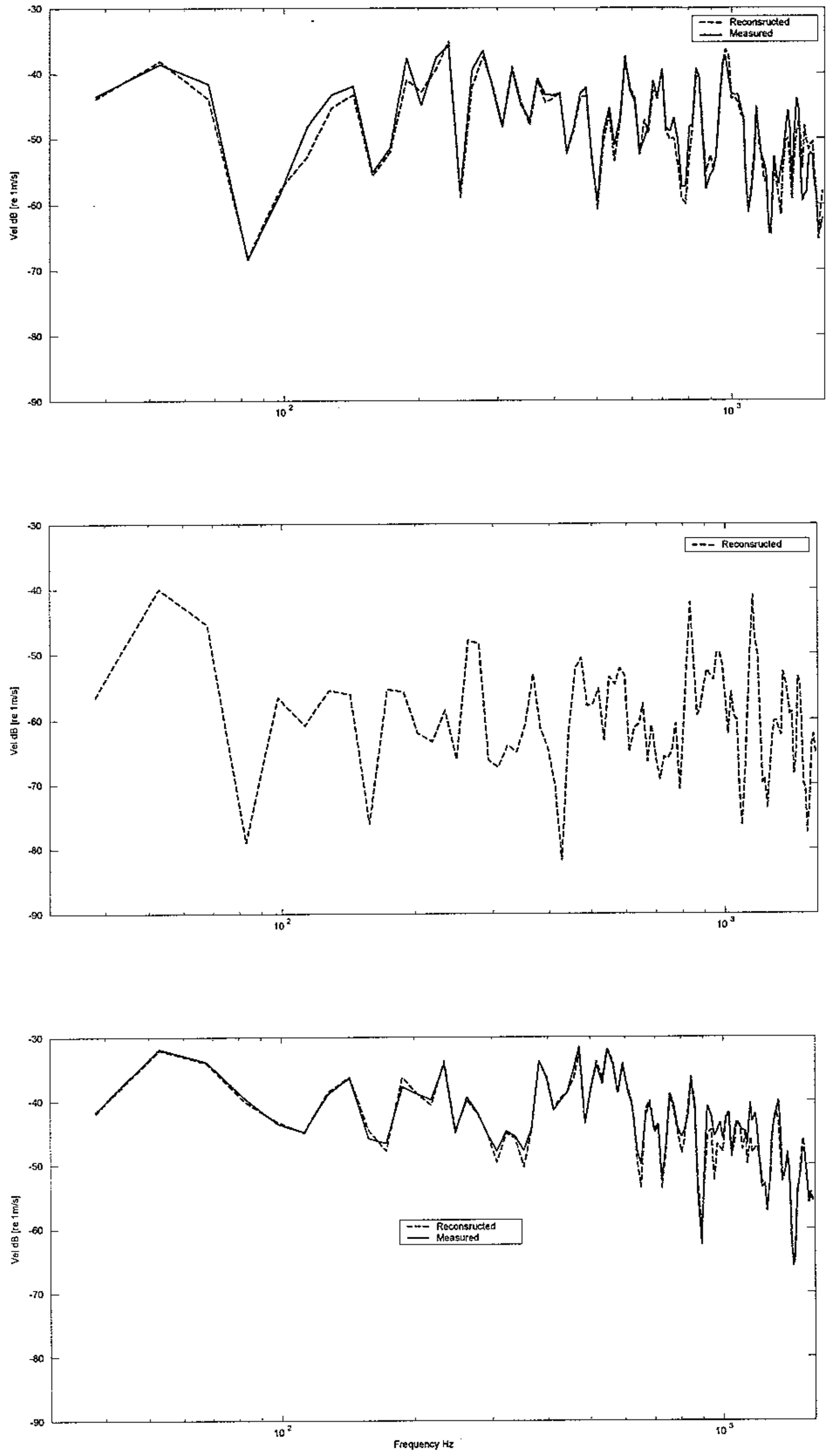

Figure 26b. Contribution from each force by Iterative inversion with validation - Variable standard deviation

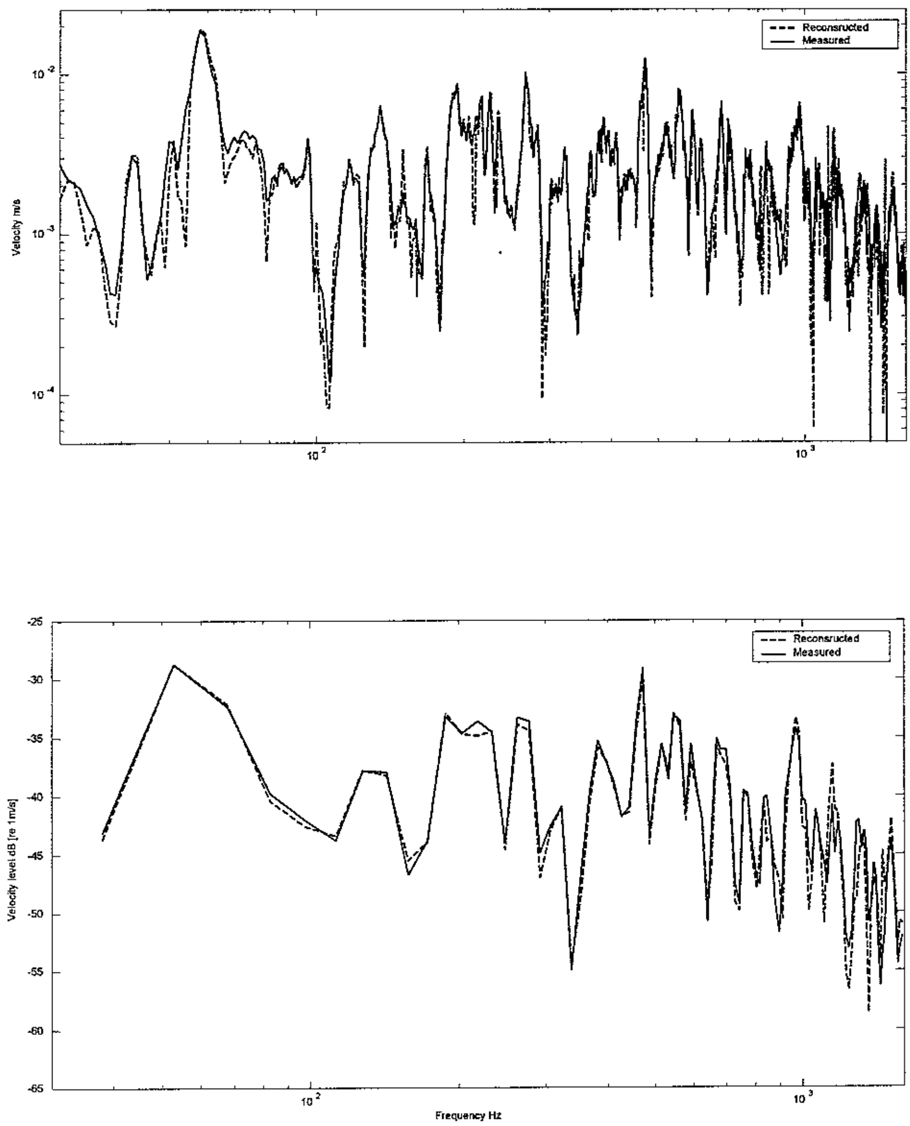

Figure 26c. Overall contribution by Iterative inversion with validation in narrow band and 15 Hz bandwidth forms - Variable standard deviation

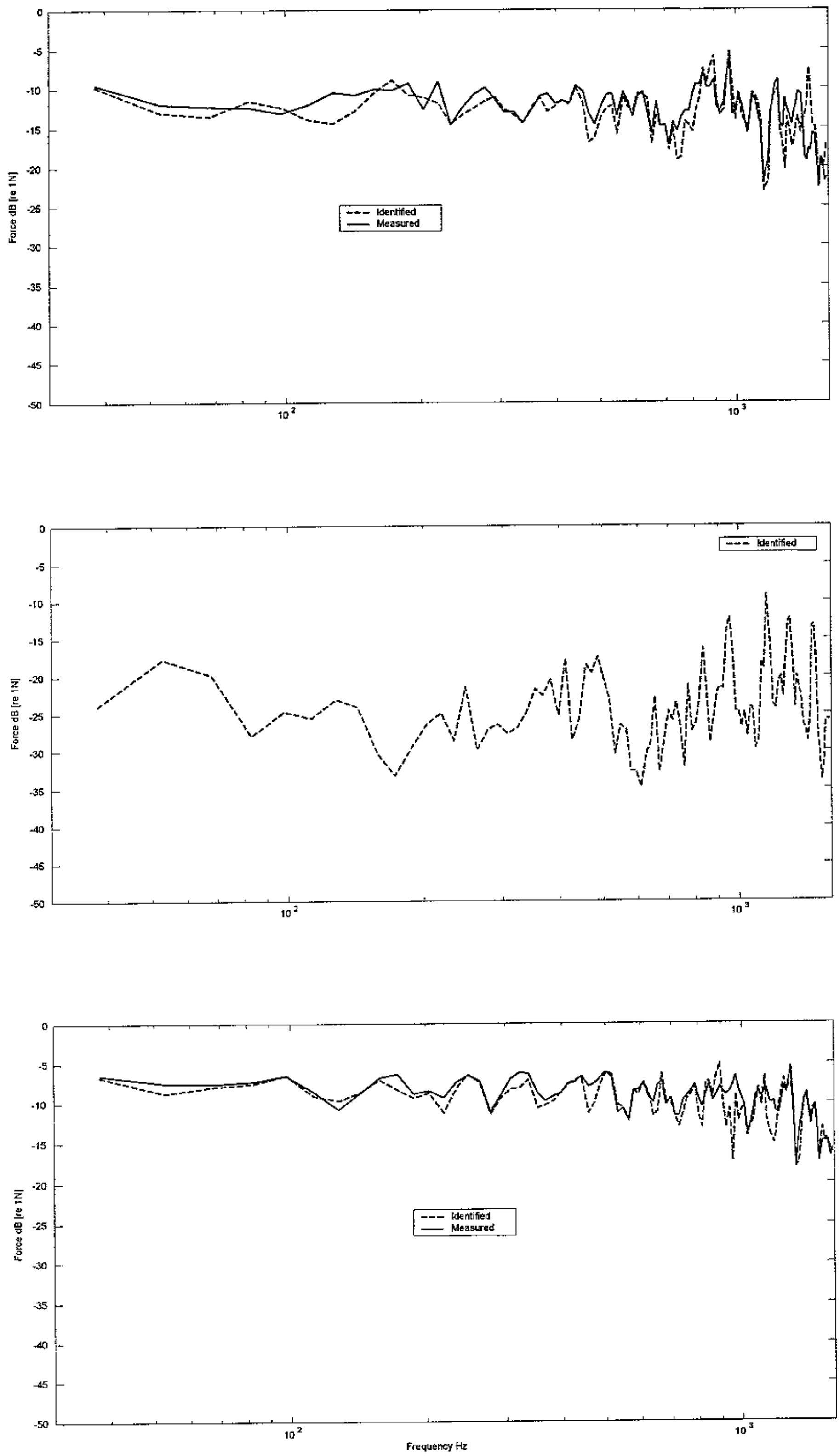

Figure 27a. Identified forces by Iterative inversion with validation -common standard deviation

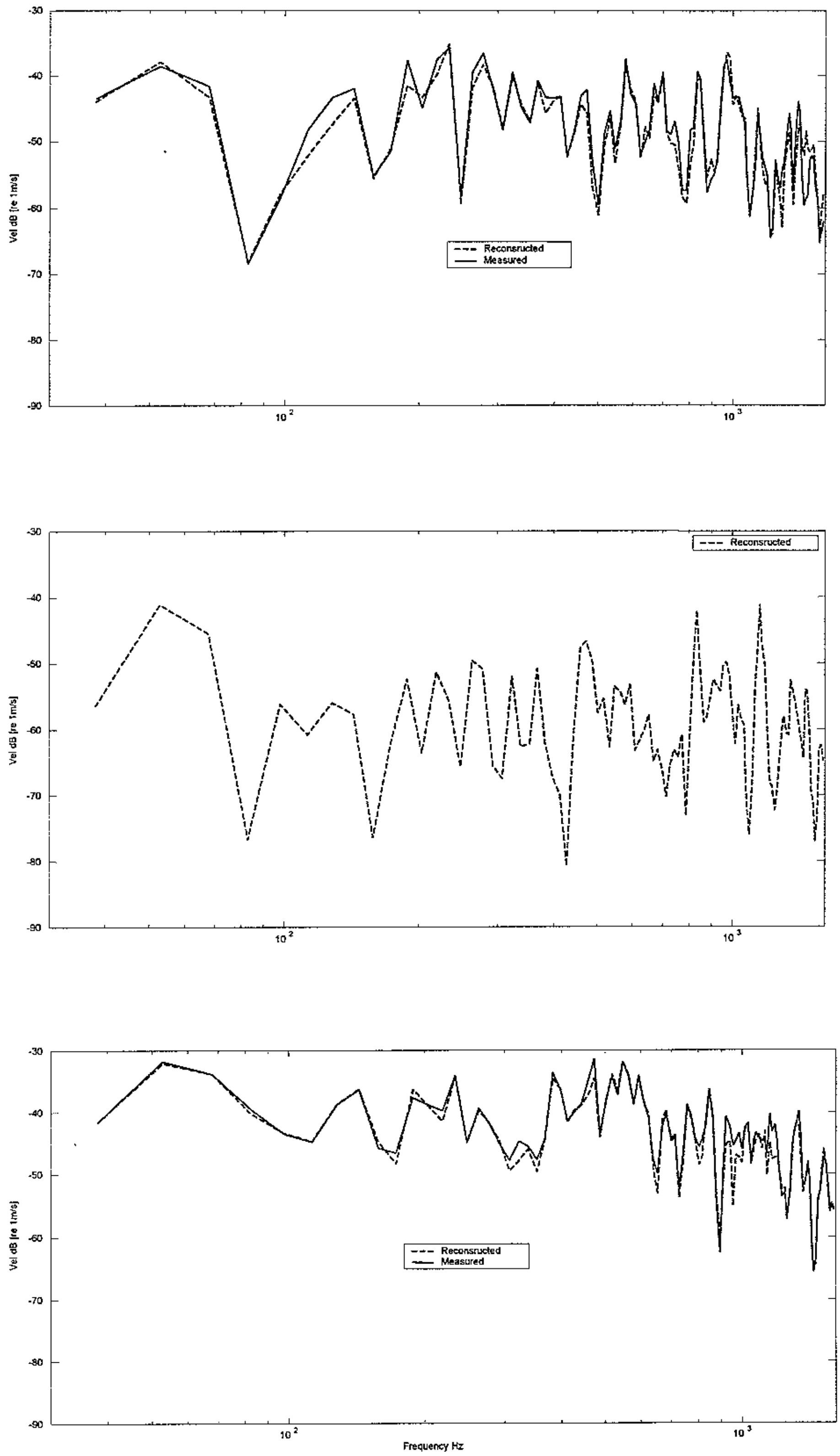

Figure 27b. Contribution from each force by Iterative inversion with validation - common standard deviation

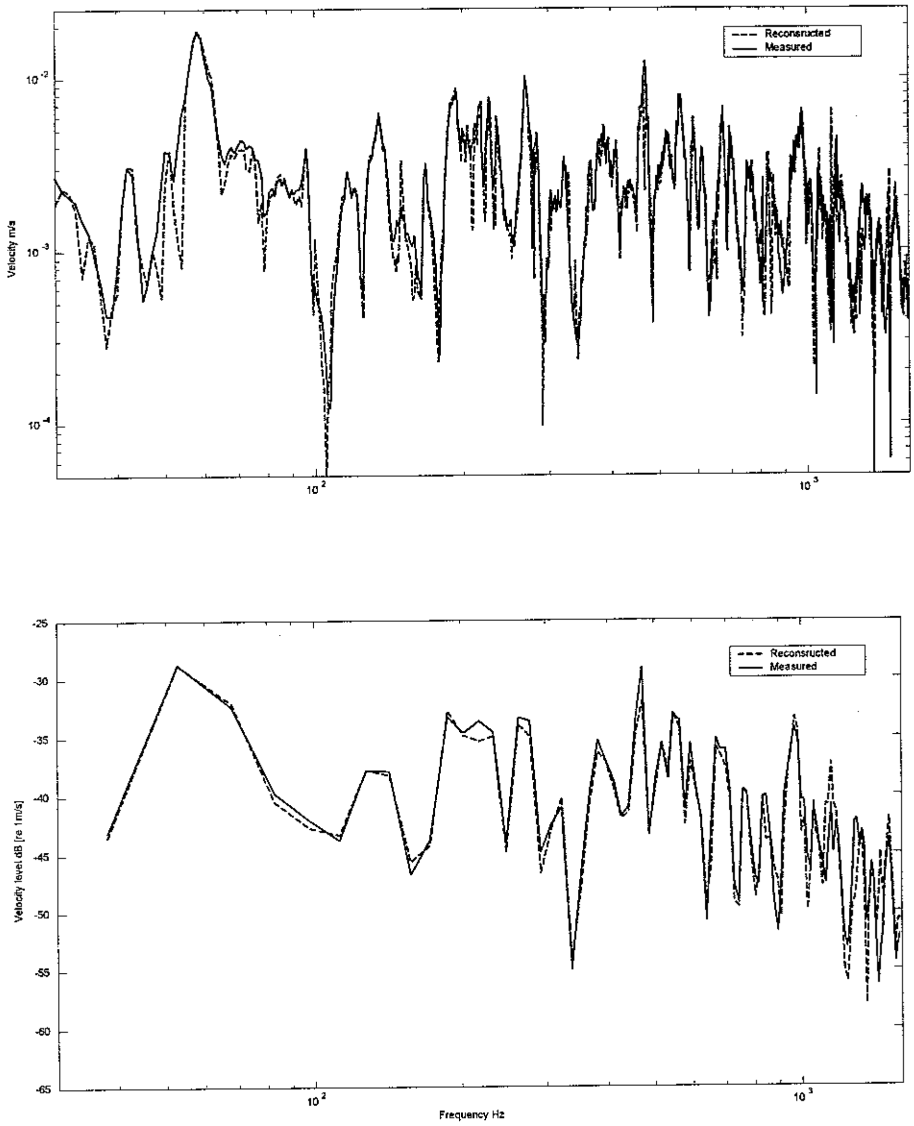

Figure 27c. Overall contribution by Iterative inversion with validation in narrow band and 15 Hz bandwidth forms -common standard deviation

3.4 Sound pressure prediction

The results of errors for the two combinations of response points with all the techniques used are shown in tables 10 and 11. As in the case of vibration response prediction, iterative inversion is found to be most consistent in sound pressure response predictions. This is closely followed by Tikhonov regularization. However, the forces themselves are better predicted by Tikhonov regularization (table 10 and 11).

Graphical results (1/12 octave band sound pressure response) for the combination $a_1a_3a_5a_{10}$ - a_p are given to demonstrate the effects of different methods (Figures 30-37). In all the cases at 150 Hz there is a sharp rise in the deviation. This is because the coherence is very low at this frequency (due to electrical noise) and the response is affected by noise as well (See figures 5 and 8). Singular value rejection methods give the best result at this frequency, next best is iterative inversion (Figure 32-33 and 37). Overall sound pressure prediction errors are smallest using Tikhonov regularization and iterative inversion with common standard deviation.

Table 10. Errors from all the techniques for predicting a_p from a_{10} a_7 a_1 and a_3

	Force errors			Response errors			
Method	F 1	F 2	F 3	***Overall***	error 1	error 2	error 3
Moore-Penrose	2.9	31.0	2.4	***3.1***	2.9	21.6	2.4
Perturbation	2.3	28.1	1.5	***2.5***	2.3	20.3	1.5
Sv rej (accelerance)	3.0	30.6	3.3	***3.1***	3.2	20.9	3.7
Sv rej (response)	6.5	28.1	7.2	***3.8***	6.9	18.1	7.4
Tikhonov -OCV	2.5	28.7	1.9	***2.4***	2.5	20.8	2.0
Iterative - % of resp as std. deviation	2.8	28.6	2.5	***2.9***	2.8	20.4	2.7
Iterative - ocv - Variable std deviation	3.5	26.9	2.5	***2.6***	3.5	18.3	2.8
Iterative - ocv - Common standard deviation	3.3	27.4	2.3	***2.5***	3.3	19.1	3.0

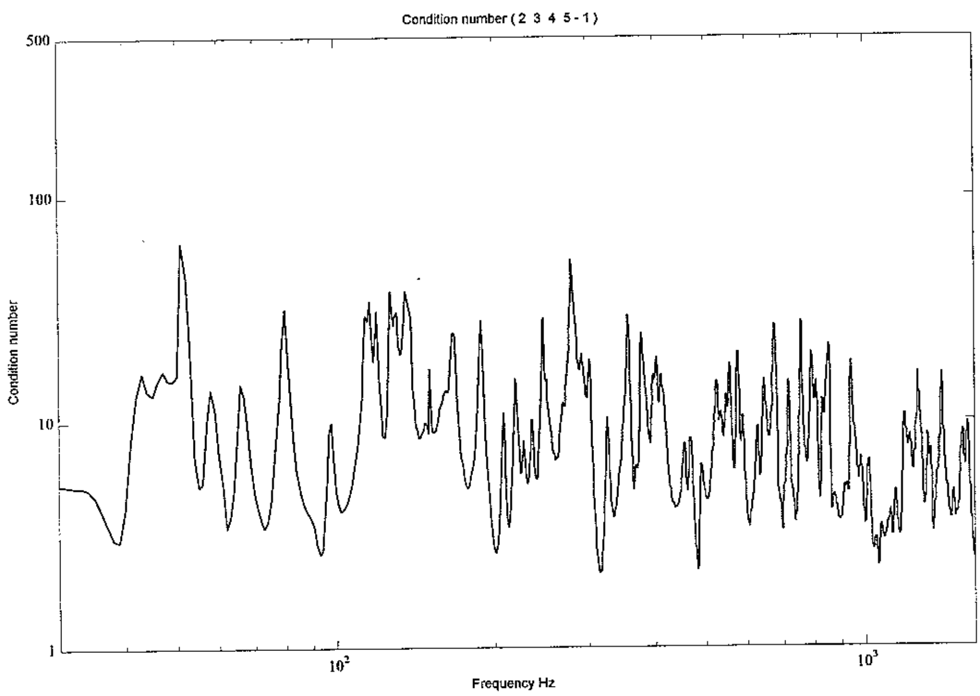

Figure 28. Condition numbers for the combination $a_{10}a_7a_1a_3$ - a_p

Table 11. Errors from all the techniques for predicting a_p from a_1 a_3 a_5 and a_{10}

	Force errors			Response errors			
Method	F 1	F 2	F 3	***Overall***	error 1	error 2	error 3
Moore-Penrose	2.3	29.1	2.0	***3.0***	2.3	19.6	2.0
Perturbation	1.6	27.0	1.2	***2.3***	1.7	19.1	1.3
Sv rej (accelerance)	2.8	28.9	2.3	***3.1***	2.9	19.3	2.7
Sv rej (response)	4.5	27.2	4.7	***3.8***	4.9	17.1	5.4
Tikhonov -OCV	1.7	28.3	1.0	***2.4***	1.8	20.1	1.2
Iterative - % of resp as std. deviation	1.8	28.0	2.0	***2.5***	1.9	19.9	2.2
Iterative - ocv - Variable std deviation	1.7	28.1	1.6	***2.3***	1.8	20.0	1.8
Iterative - ocv - Common standard deviation	1.8	27.7	1.8	***2.2***	1.9	19.6	2.0

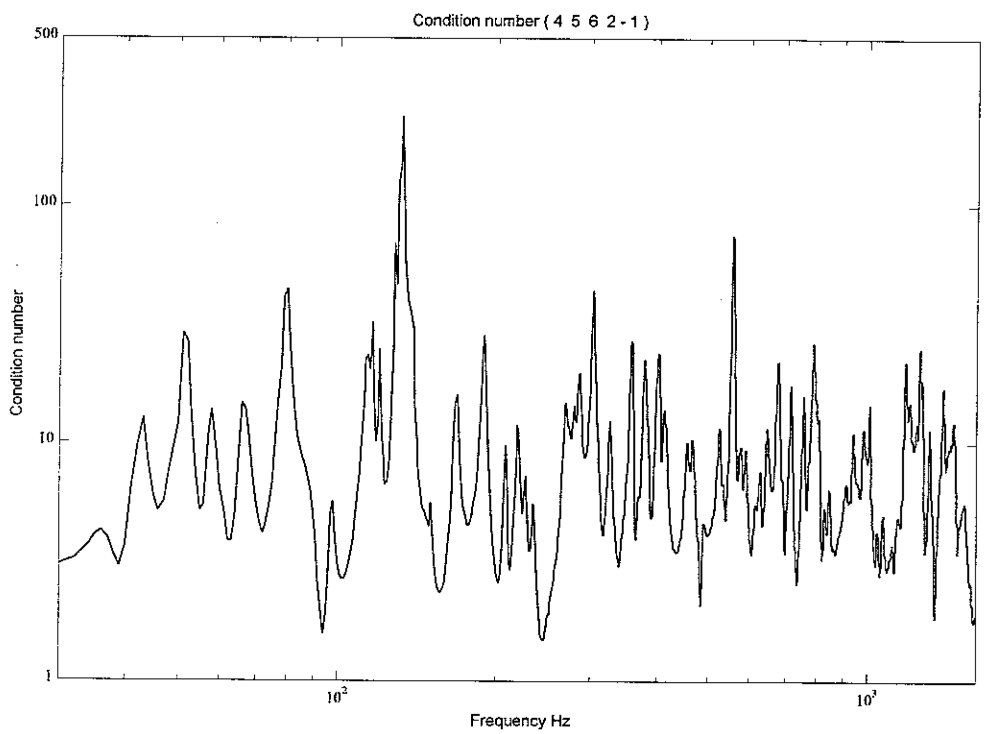

Figure 29. Condition numbers for the combination $a_1a_3a_5a_{10}$ - a_p

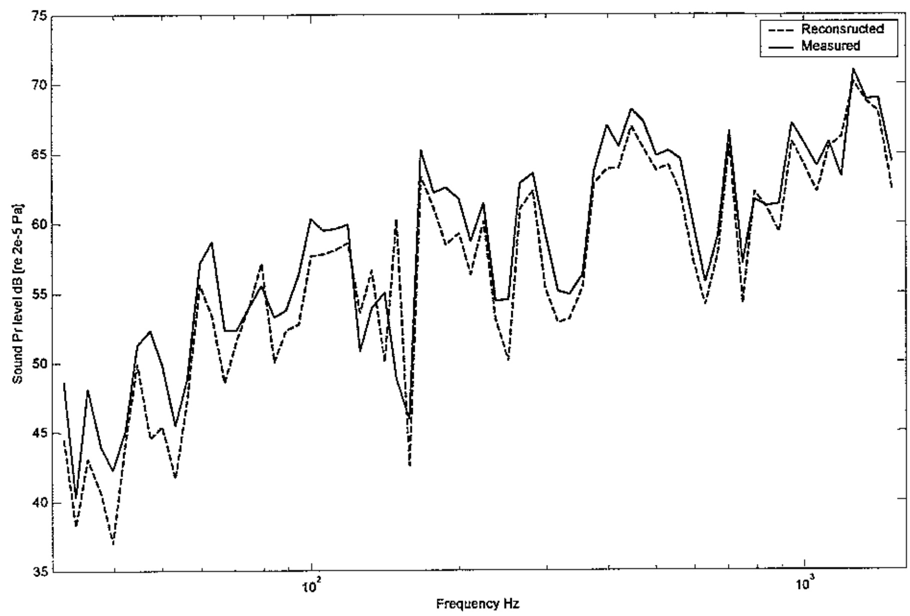

Figure 30. 1/12 octave overall sound pressure response by Moore-Penrose pseudo inversion

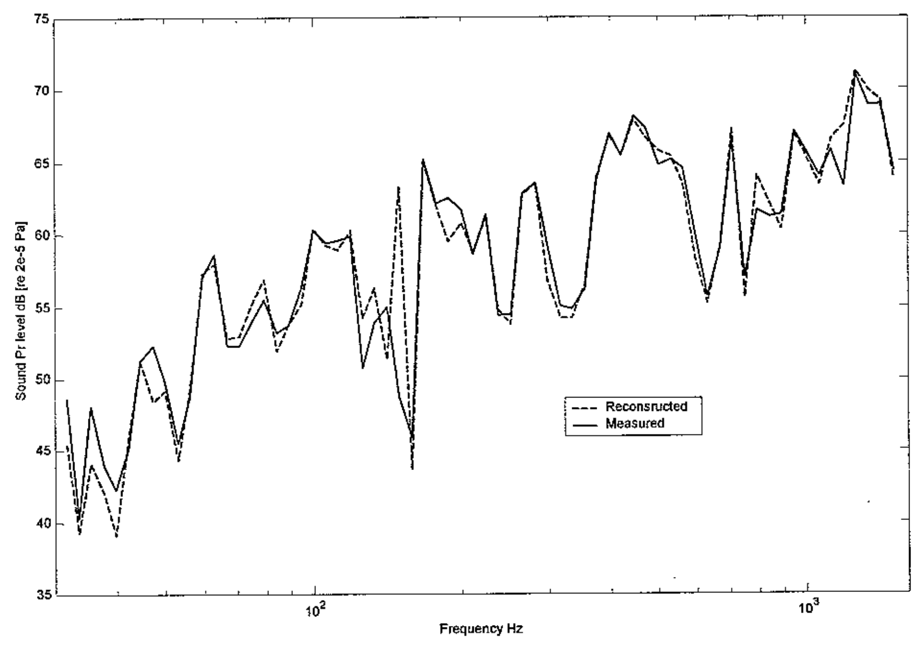

Figure 31. 1/12 octave overall sound pressure response by Perturbation technique

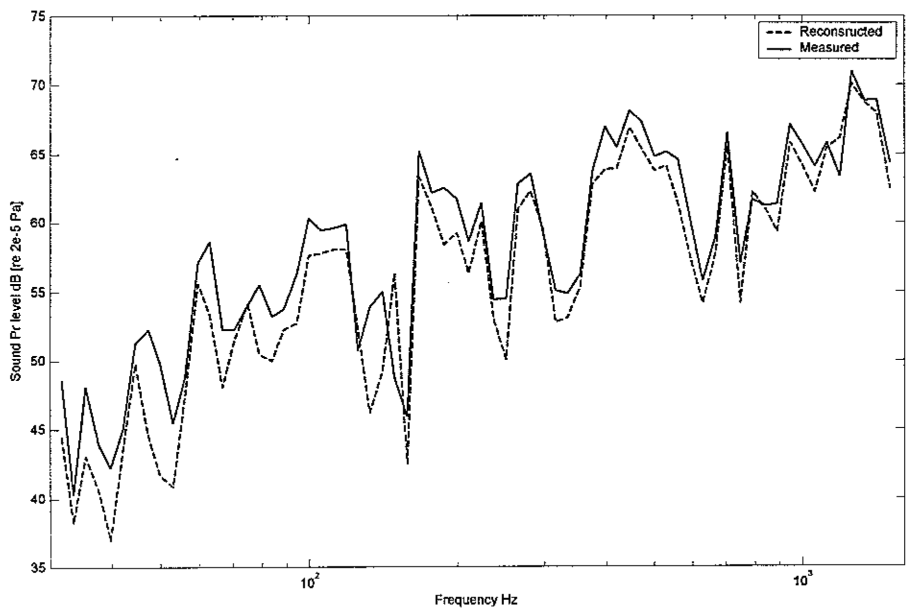

Figure 32. 1/12 octave overall sound pressure response by Singular value rejection based on error in accelerance

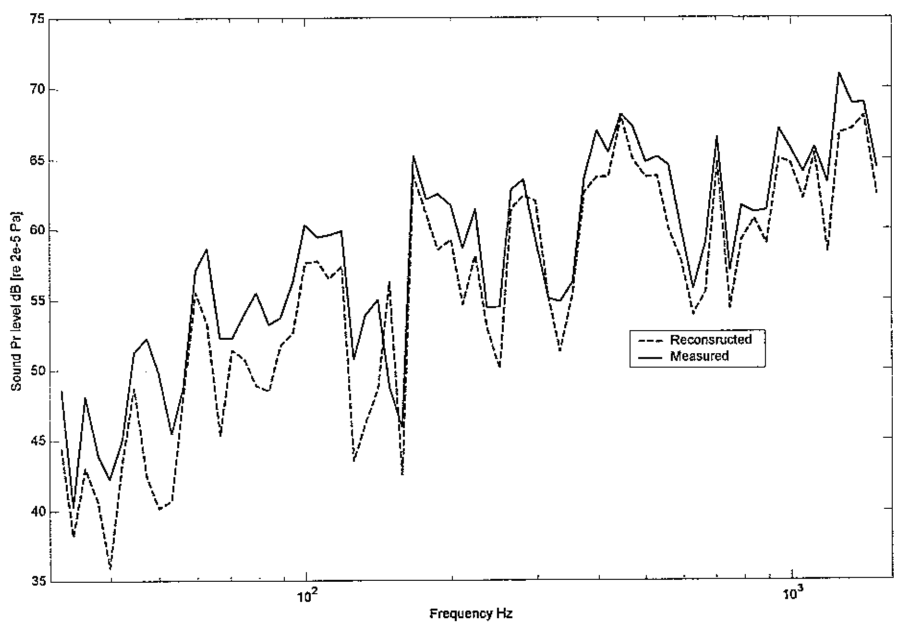

Figure 33. 1/12 octave overall sound pressure response by Singular value rejection based on error in responses

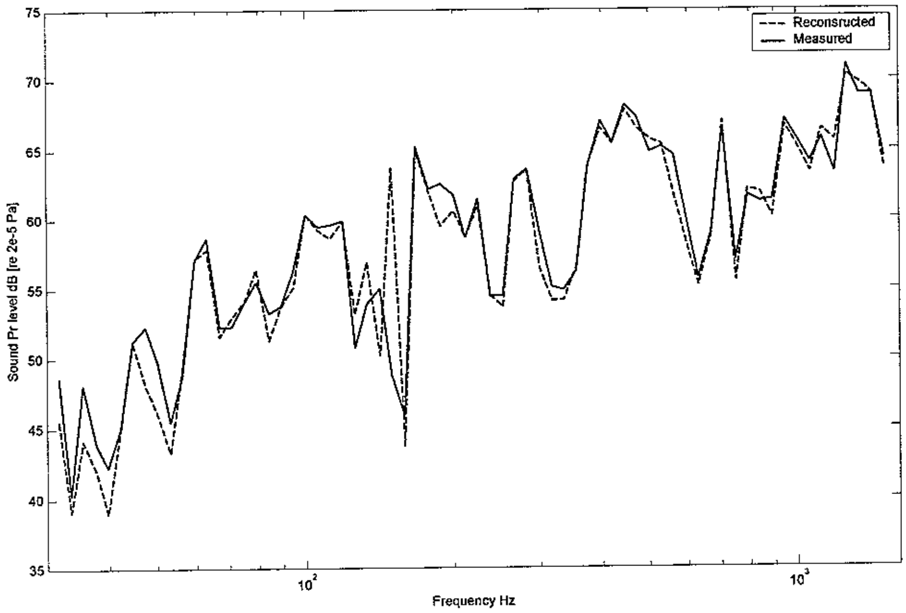

Figure 34. 1/12 octave overall sound pressure response by Tikhonov regularization

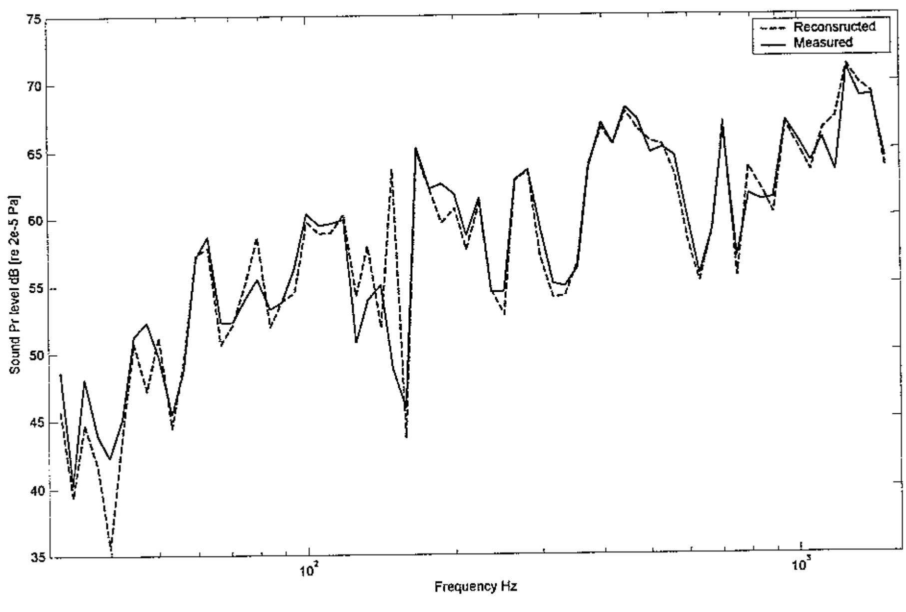

Figure 35. 1/12 octave overall sound pressure response by Iterative inversion with % of response as standard deviation

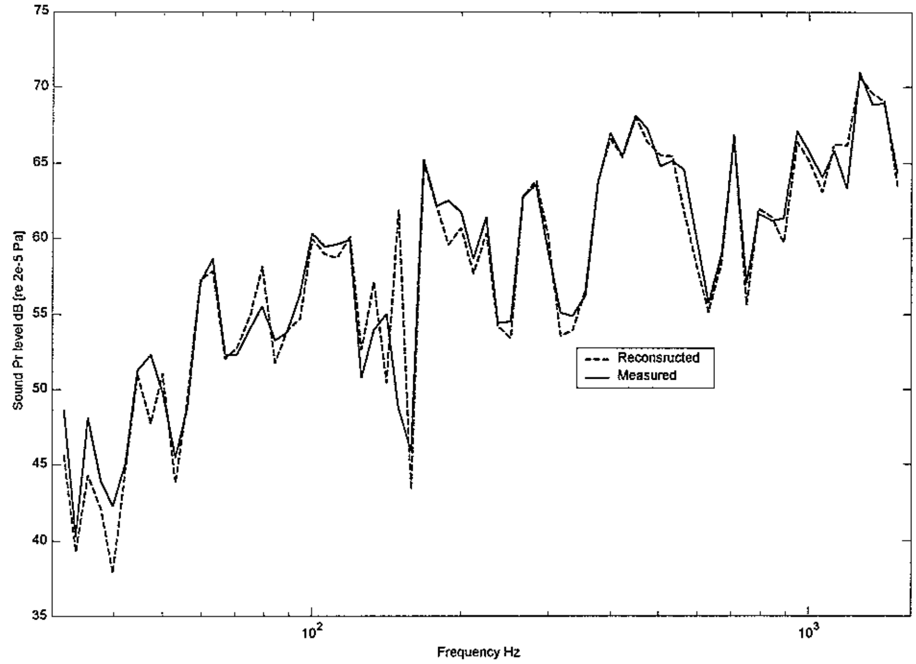

Figure 36. 1/12 octave overall sound pressure response by Iterative inversion with validation -variable standard deviation

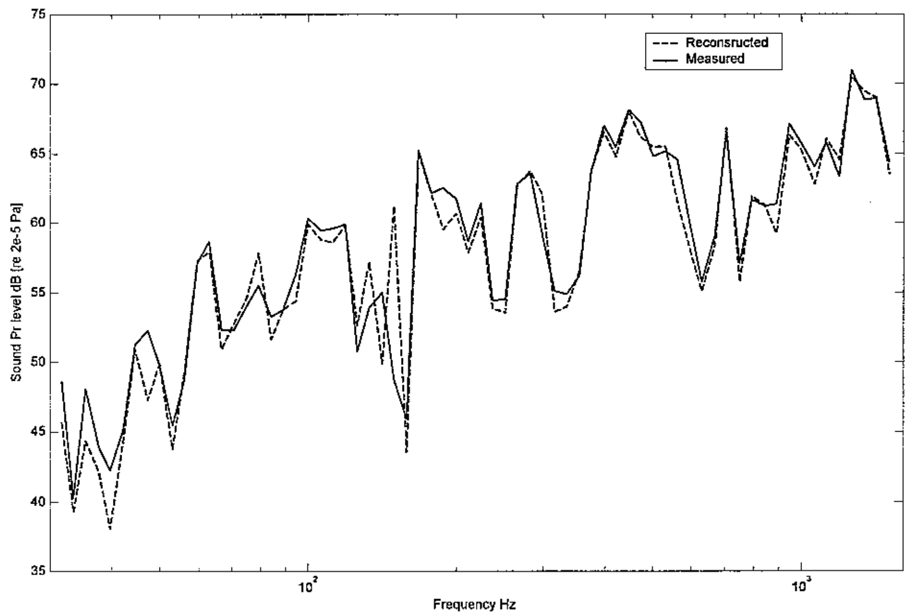

Figure 37. 1/12 octave overall sound pressure response by Iterative inversion with validation -common standard deviation

3.5 Discussion

Even though the concepts of Tikhonov regularization and iterative inversion are similar, the predictions differ since the regularization parameter for Tikhonov regularization varies from 0 to 50% of the maximum singular value. In many cases where there are high condition numbers, this results in selecting no regularization. This is clear from the variation of regularization parameter with frequency (Figure 38). On the other hand, for iterative inversion the minimum standard deviation is taken as 5% of the response. This is found to result in better response prediction.

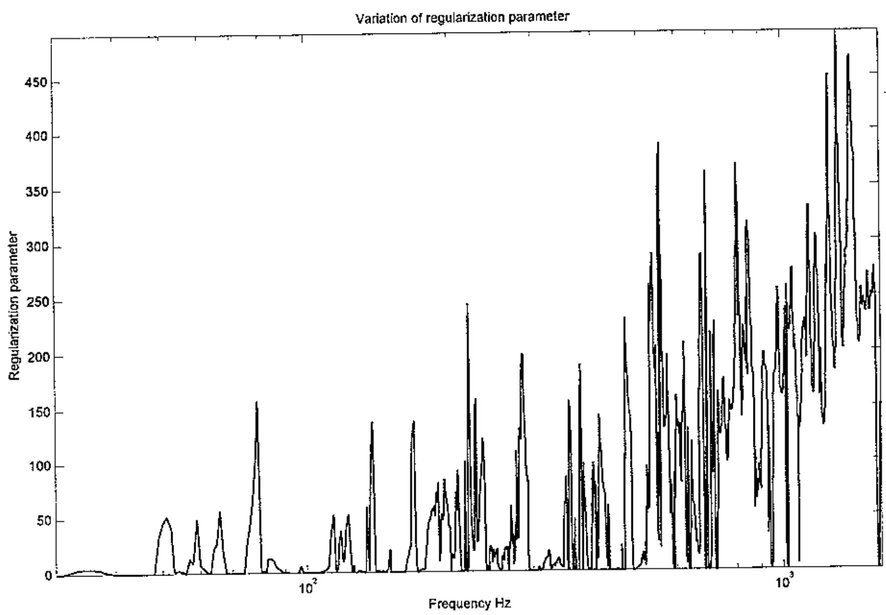

Figure 38. Variation of regularization parameter in Tikhonov regularization

Examples of the filter characteristics applied to the singular values by both of these methods are shown in Figure 39 for frequencies of 167-169 Hz. The regularization at these frequencies is more effective in iterative inversion, which results in reduction of the peaks found in response prediction.

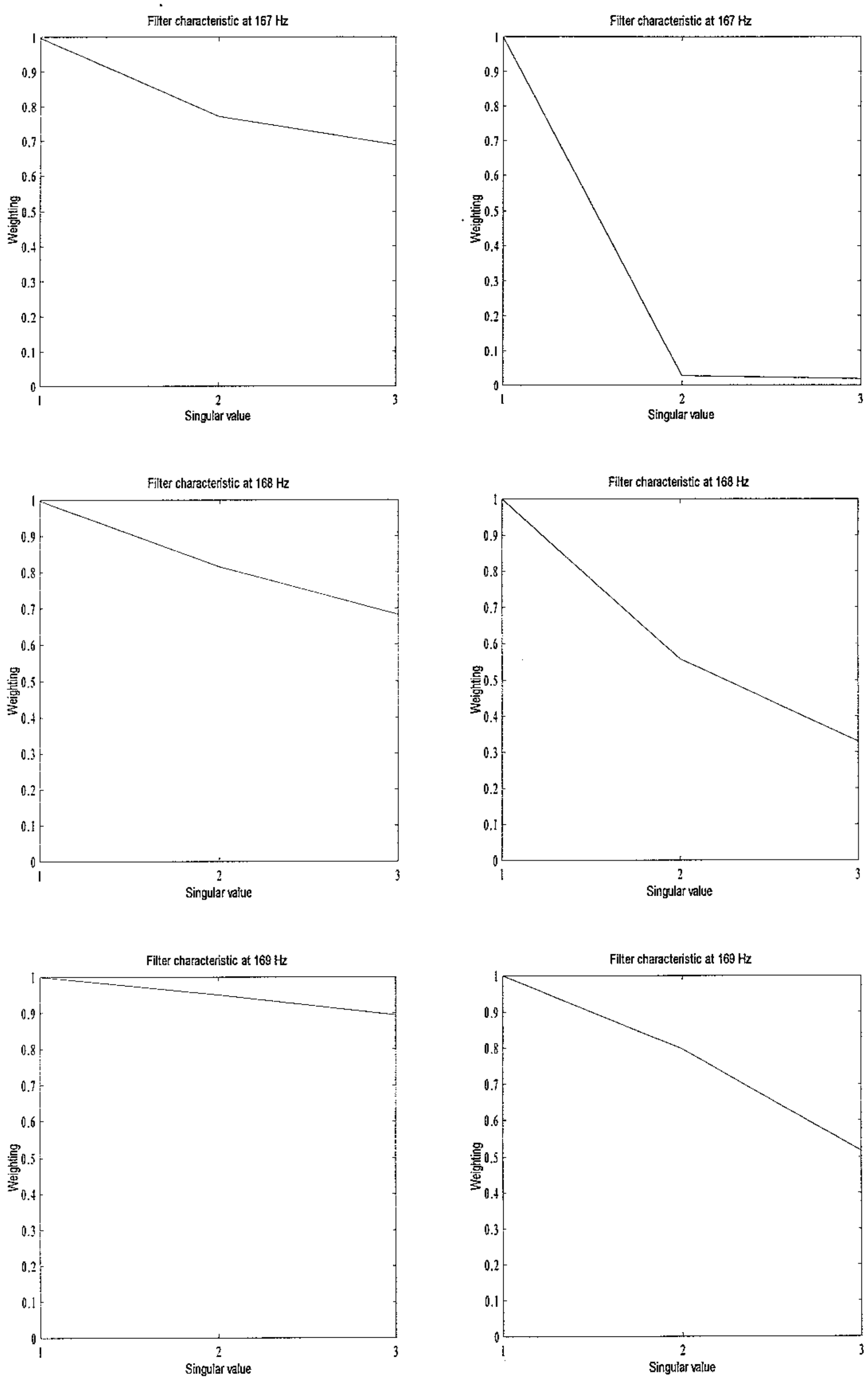

Figure 39. Filter characteristic at 167 - 169Hz. Left side figures Tikhonov regularization, Right side Iterative inversion

4. CONCLUSIONS

Experiments have been conducted on a hanging rectangular flat plate. Operational responses and FRF's were measured for three force positions, ten accelerometer positions and a microphone position. Operational forces were also measured to compare with the identified forces. The validity of measurements has been checked by directly reconstructing the responses from the measured forces and accelerances and comparing with the measured responses.

From the methods described in [5-7] to perform the reconstruction, 8 of them are investigated for the measured data. Based on the force identification and response predictions by all these methods, the following conclusions can be drawn.

1. Use of the H_1 transfer function estimator to determine the operational responses results in better force identification than taking auto spectra with phase relations from cross spectra (equation (5) and (2)). Nevertheless the latter was used in further investigation since the H_1 concept magnifies the response at antiresonances.
2. The singular value rejection method based on the threshold established by the norm of the response error vector results in better prediction of peaks in the responses (narrow band representation) than any other technique considered. However, at antiresonances the responses are under-estimated to a large extent. It might be necessary to evolve a different analytical approach in establishing the threshold.
3. The singular value rejection based on the norm of the accelerance error matrix does not improve the predictions compared to the Moore-Penrose pseudo inverse. This is due to good coherence in most of the frequency range, which means that the error matrix has small values. This leads to a conclusion that for an effective

singular value rejection strategy, a threshold for singular value rejection should be based on both response and FRF errors.

4. The perturbation technique developed in [5] was observed to result in much superior force prediction compared to singular value rejection. However, this method is not robust since perturbation takes care of errors in FRF's alone leaving errors in responses to be magnified. This was also the case in [6].
5. The Tikhonov regularization with OCV for regularization parameter selection is found to identify the forces much better than all the methods investigated. However, this method does not reduce the error magnification in certain cases because of the way the regularization parameter is varied (0 to 50% of maximum singular value). It seems advisable to set a non-zero minimum value for the regularization parameter based on either the error in the response or FRF.
6. The iterative inversion combined with validation is found to be the most robust of all the methods investigated. In arriving at the best fit response prediction, this method results in under-estimation of some forces at many frequencies. Hence the individual contributions might contain large errors. It also takes about 7 times longer to calculate the inverse than by Tikhonov regularization.

5. FURTHER WORK

Further improvements to Tikhonov regularization and singular value rejection methods can be carried out to improve the predictions. These improvements could be in the following directions

1. Reviewing the assumptions made in arriving at the threshold for rejecting singular values in the case of the response error vector. The measured data can be studied for this.
2. Combining threshold levels for singular values based on accelerance error and response error so as to include every possible error present in the measurements.
3. Deriving an expression for a minimum value of the regularization parameter in Tikhonov regularization that could be based on the response error.

6. REFERENCES

1. M.H.A. Janssens, J.W. Verheij and DJ 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. JW Verheij, Inverse and reciprocity methods for machinery noise source charectrization and sound path quantification, Part 1 : Sources. International Journal of Acoustics and Vibration, Vol 2, No1, 11-20 1997.
4. JW Verheij, Inverse and reciprocity methods for machinery noise source charectrization and sound path quantification, Part 2 : Transmission paths. International Journal of Acoustics and Vibration, Vol 2, No3, 103-112 1997.
5. AN Thite and DJ Thompson, Study of indirect force determination and transfer path analysis using numerical simulations for a flat plate. ISVR Technical Memorandum no. 851, May 2000.
6. AN Thite and DJ Thompson, Further study of indirect force determination and transfer path analysis using numerical simulations for a flat plate. ISVR Technical Memorandum no. 854, October 2000.
7. AN Thite and DJ Thompson, Study of regularized solutions to improve the indirect force determination and transfer path analysis using numerical simulations for a flat plate. ISVR Technical Memorandum no. 855, October 2000.
8. MHA Janssens, JW Verheij and T Loyau, Experimental example of the pseudo-forces method used in characterisation of a structure-borne sound source. Paper submitted to Applied Acoustics.

9. JA Fabunmi and FA Tasker, Advanced Techniques for measuring structural mobilities. ASME Journal of Vibration, Acoustics, Stress and Reliability in Design, 345-349, Vol. 110, July 1988.

10. J Biemond, RL Lagendijk and RM Mersereau, Iterative Methods for Image processing. Proceedings of the IEEE, Vol. 78, No. 5, May 1990.

11. PA Nelson, Some inverse problems in acoustics. Sixth international congress on sound and vibration, 5-8 July 1999, 7-32.

12. PA Nelson and SH Yoon, Estimation of acoustic source strength by inverse methods: Part I, conditioning of the inverse problem. Journal of sound and vibration (2000) 233(4), 643-668.

13. SH Yoon and PA Nelson, Estimation of acoustic source strength by inverse methods: Part II, Experimental investigation of methods for choosing regularization parameters. Journal of sound and vibration (2000) 233(4), 669-705.

```
clear
n_ex=3;                    % Number of forcing points
n_resp=6;                  % Number of responses
load c:\thite\ant\tpa_meas1\onstud\series_exp\a8106134\oper123
for ff=1:n_ex+n_resp
   delta_f=1;
   Fs=1/time(2);
   N=Fs/delta_f;
   f=[0:delta_f:1600]';
   f_min=find(f==0);
   f_max=find(f==1600);
   w=hanning(N);
   w=w(:);
   wp=sum((w').^2)./N;
   f_sxy=[0:N/2]'*Fs/N;
         x=(operate_resp(:,ff)-mean(operate_resp(:,ff)));
         y=(operate_resp(:,1)-mean(operate_resp(:,1)));
         blocks=floor(length(x)/N);
         sumSxy=zeros(size(f_sxy));
         sumSxx=zeros(size(f_sxy));
         sumSyy=zeros(size(f_sxy));
         for ii=1:blocks
              x_fft=fft(w.*x((ii-1)*N+1:N*ii));
               y_fft=fft(w.*y((ii-1)*N+1:N*ii));
               sumSxx=2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N)+sumSxx;
               sumSxy=2*((x_fft(1:N/2+1)).*conj(y_fft(1:N/2+1)))./
                     (Fs*N)+sumSxy;
               sumSyy=2*((abs(y_fft(1:N/2+1))).^2)./(Fs*N)+sumSyy;
         end
         Sxy=(sumSxy)./(blocks*wp);
         Syy=(sumSyy)./(blocks*wp);
         Sxx=(sumSxx)./(blocks*wp);
         Hxy=Sxy./Syy;
         resp_temp=sqrt(Sxx).*exp(i*angle(Sxy));
         operate_resp_freq(:,ff)=resp_temp(f_min:f_max);
         clear x y resp_temp
end
freq=f;

force_measure_freq=operate_resp_freq(:,1:n_ex);
operate_acc_freq=operate_resp_freq(:,n_ex+1:n_ex+n_resp);
```

```
clear
n_ex=3;                  % Number of forcing points
n_resp=6;                % Number of responses
source_file='trof';
source_path='c:\thite\ant\tpa_meas1\onstud\series_exp\ap107135'
% Frf esimation
for ff=1:n_ex
   eval(['load ' source_path '\FRF' source_file num2str(ff+1)])
   delta_f=1;
   Fs=1/time(2);
   N=Fs/delta_f;
   f=[0:delta_f:1600]';
   f_min=find(f==0);
   f_max=find(f==1600);
   w=hanning(N);
   w=w(:);
   wp=sum((w').^2)./N;
   f_sxy=[0:N/2]'*Fs/N;
   for kk=1:n_resp
      x=(acceleration(:,kk)-mean(acceleration(:,kk)));
      y=(force-mean(force));
      blocks=floor(length(x)/N);
      sumSxy=zeros(size(f_sxy));
      sumSxx=zeros(size(f_sxy));
      sumSyy=zeros(size(f_sxy));
      for ii=1:blocks
         x_fft=fft(w.*x((ii-1)*N+1:N*ii));
         y_fft=fft(w.*y((ii-1)*N+1:N*ii));
         sumSxx=2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N)+sumSxx;
         sumSxy=2*(conj(x_fft(1:N/2+1)).*(y_fft(1:N/2+1)))./
                (Fs*N)+sumSxy;
         sumSyy=2*((abs(y_fft(1:N/2+1))).^2)./(Fs*N)+sumSyy;
      end
      Sxy=(sumSxy)./(blocks*wp);
      Syy=(sumSyy)./(blocks*wp);
      Sxx=(sumSxx)./(blocks*wp);
      frf_temp=Sxy./Syy;
      coh_temp=((abs(Sxy)).^2)./(Sxx.*Syy);
      frf_measure(:,kk+(ff-1)*n_resp)=frf_temp(f_min:f_max);
      coh_measure(:,kk+(ff-1)*n_resp)=coh_temp(f_min:f_max);
      clear x y
    end
    clear force acceleration
```

```
end
freq=f;

eval(['save ' source_path '\FRF_' source_file ' frf_measure
coh_measure freq'])
```

Appendix C Program for Force Identification by Pseudo-inverse and Singular value rejection

```
resp_number=[3 4 5 6];  % numbers upto 10 in set of 4 or 5 or ... 9

receiver_number=[2];    % any number from 1 to 10 other than above

pressure=0;             % Sound pressure or Vibration prediction

validation_number=[1];  % Cross validation response

break_program=0;
for ii=1:length(resp_number)
   if resp_number(ii)==validation_number;
      break_program=1;
   end
end
if break_program==1
   'Warning !!!!!! Same location used for validation and inversion'
   break
end

ref_signal=9;

concept=3;              % is 1 for H1, 2 for H2 and 3 for phase from
                        % cross spectra

correction=0;           % Whether singular values rejected
resp_accelerance=0;     % 1 - For responce based rejection, 2 for
                        % accelerance based rejection
coh_noise=1;            % Standard deviation for accelerance error

min_freq=30;
max_freq=1600;


load c:\thite\ant\tpa_meas1\onstud\series_exp\a8107135\frf_trof

clear frf_measure coh_measure       % Only retain frequency data

f=[min_freq:freq(2):max_freq]';
f_min=find(freq==min_freq);
f_max=find(freq==max_freq);

load c:\thite\ant\tpa_meas1\onstud\series_exp\a8107135\oper123

force_measure=operate_resp(:,1:3);

Fs=1/time(2);
N=Fs/freq(2);

n_resp=length(resp_number);
n_ex=length(force_measure(1,:));


% FFT of force signal
%
for kk=1:n_ex
   x=(force_measure(:,kk)-mean(force_measure(:,kk)));
   y=(operate_resp(:,ref_signal)-mean(operate_resp(:,ref_signal)));
```

```
   w=hanning(N);
   w=w(:);

   f_sxy=[0:N/2]'*Fs/N;

   sumSxy=zeros(size(f_sxy));
   sumSxx=zeros(size(f_sxy));
   sumSyy=zeros(size(f_sxy));

   blocks=floor(length(x)/N);

   for ii=1:blocks
      x_fft=fft(w.*x((ii-1)*N+1:N*ii));
      y_fft=fft(w.*y((ii-1)*N+1:N*ii));
    sumSxy=2*(conj(x_fft(1:N/2+1)).*(y_fft(1:N/2+1)))./(Fs*N)+sumSxy;
      sumSyy=2*((abs(y_fft(1:N/2+1))).^2)./(Fs*N)+sumSyy;
      sumSxx=2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N)+sumSxx;
   end
   wp=sum((w').^2)./N;
   Sxy=(sumSxy)./(blocks*wp);
   Syy=(sumSyy)./(blocks*wp);
   Sxx=(sumSxx)./(blocks*wp);
   if kk==2
      force_temp=zeros(size(Sxx));
   else
      if concept==1
         force_temp=(Sxy)./(Syy);              % H1 concept
      elseif concept==2
         force_temp=(Sxx)./(Sxy);              % H2 concept
      elseif concept==3
         force_temp=sqrt(Sxx).*exp(i*angle(Sxy));
      end
   end
   force_measure_freq(:,kk)=force_temp(f_min:f_max);
end

n_av=blocks;
clear signal_gen_force force_temp time_force x y


% Reading in FRF data

load c:\thite\ant\tpa_meas1\onstud\series_exp\a8107135\frf_trof
   %a8107135\frf_trof   a8106134 ap107135

frf_spacing=length(frf_measure(1,:))/n_ex;

for ii=1:n_ex
   for kk=1:n_resp
      temp_H_hat(:,kk+n_resp*(ii-1))=frf_measure. . .
      (:,resp_number(kk)+(ii-1)*frf_spacing);
      temp_coh_hat(:,kk+n_resp*(ii-1))=coh_measure. . .
      (:,resp_number(kk)+(ii-1)*frf_spacing);
   end
   temp_transfer_path(:,ii)=frf_measure. . .
   (:,receiver_number+(ii-1)*frf_spacing);
   temp_validation_path(:,ii)=frf_measure. . .
   (:,validation_number+(ii-1)*frf_spacing);
end

H_hat=temp_H_hat(f_min:f_max,:);
coh_hat=temp_coh_hat(f_min:f_max,:);
transfer_path=temp_transfer_path(f_min:f_max,:);
validation_path=temp_validation_path(f_min:f_max,:);
```

```
clear frf_measure coh_measure temp_H_hat temp_coh_hat
temp_transfer_path temp_validation_path


operate_acc=operate_resp(:,4:9);

for kk=1:n_resp
   temp_acc(:,kk)=(operate_acc(:,resp_number(kk)). . .
   -mean(operate_acc(:,resp_number(kk))));
end
   temp_acc(:,kk+1)=(operate_acc(:,receiver_number). . .
   -mean(operate_acc(:,receiver_number)));
   temp_acc(:,kk+2)=(operate_acc(:,validation_number). . .
   -mean(operate_acc(:,validation_number)));


% Receiver and validation measured response

for kk=n_resp+1:n_resp+2
   x=temp_acc(:,kk);
   y=(operate_resp(:,ref_signal)-mean(operate_resp(:,ref_signal)));
   w=hanning(N);
   w=w(:);

   f_sxy=[0:N/2]'*Fs/N;

   sumSxy=zeros(size(f_sxy));
   sumSxx=zeros(size(f_sxy));

   sumSyy=zeros(size(f_sxy));

   blocks=floor(length(x)./N);
   wp=sum((w').^2)./N;

   for ii=1:blocks
      x_fft=fft(w.*x((ii-1)*N+1:N*ii));
      y_fft=fft(w.*y((ii-1)*N+1:N*ii));
      sumSxy=2*(conj(x_fft(1:N/2+1)).*(y_fft(1:N/2+1). . .
      )). /(Fs*N)+sumSxy;
      sumSyy=2*((abs(y_fft(1:N/2+1))).^2)./(Fs*N)+sumSyy;
      sumSxx=2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N)+sumSxx;
   end
   Sxy=(sumSxy)./(blocks*wp);
   Syy=(sumSyy)./(blocks*wp);
   Sxx=(sumSxx)./(blocks*wp);

   ref_signal_mag=sqrt(Syy(f_min:f_max));      % Reference signal for
                                               % multiplication
   if concept==1
        acc_temp=(Sxy)./(Syy);                 % H1 concept
   elseif concept==2
        acc_temp=(Sxx)./(Sxy);                 % H2 concept
   elseif concept==3
        acc_temp=sqrt(Sxx).*exp(i*angle(Sxy));
   end
   if kk==n_resp+1
      rec_acc_freq=acc_temp(f_min:f_max);
   else
      vali_acc_freq=acc_temp(f_min:f_max);
   end
end

clear force_measure

w=hanning(N);
w=w(:);
```

```
wp=sum((w').^2)./N;

sum_a_hat=zeros(length(f),n_resp);

blocks=floor(length(x)/N);

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:blocks
   temp_acc_block=temp_acc((ii-1)*N+1:N*ii,:);
   temp_signal_gen_block=y((ii-1)*N+1:N*ii,:);

   for kk=1:n_resp
      Sxy=zeros(size(f_sxy));
      Syy=zeros(size(f_sxy));
      Sxx=zeros(size(f_sxy));

         x=temp_acc_block(:,kk);
         yy=temp_signal_gen_block;
      x_fft=fft(w.*x);
      y_fft=fft(w.*yy);
      Sxy=2*(conj(x_fft(1:N/2+1)).*(y_fft(1:N/2+1)))./(Fs*N*wp);
      Syy=2*((abs(y_fft(1:N/2+1))).^2)./(Fs*N*wp);
      Sxx=2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N*wp);
      if concept==1
         acc_temp=(Sxy)./(Syy);                % H1 concept
      elseif concept==2
         acc_temp=(Sxx)./(Sxy);                % H2 concept
      elseif concept==3
         acc_temp=sqrt(Sxx).*exp(i*angle(Sxy));
      end
      a_hat(:,kk)=(acc_temp(f_min:f_max));

      % Routine for standard deviation estimation

      if ii==1
         inst_resp(:,ii+(kk-1)*blocks)=(acc_temp(f_min:f_max));
         save inst_resp inst_resp
         clear inst_resp
      else
         load inst_resp
         inst_resp(:,ii+(kk-1)*blocks)=(acc_temp(f_min:f_max));
         save inst_resp inst_resp
         clear inst_resp
      end
   end
   sum_a_hat=sum_a_hat+a_hat;
   if ii==blocks
      a_hat_avg=sum_a_hat./blocks;
      load inst_resp
      for jj=1:n_resp
         measur_std(:,jj)=(std((inst_resp(:,1+(jj-1). . .
         *blocks:jj*blocks)).')).';
      end
      clear inst_resp
   end

   clear operate_acc time operate_resp

   for mm=1:length(f)
      for jj=1:n_ex
                H_acc(:,jj)=H_hat(mm,(jj-1)*n_resp+1:jj*n_resp).';
```

```
            coh_acc(:,jj)=coh_hat(mm,(jj-1). . .
                         *n_resp+1:jj*n_resp)';
 end
[U,S,V]=svd(H_acc);
 E=coh_noise*sqrt((1-coh_acc)./(2*n_av.*coh_acc)).*abs(H_acc);
                                                       %error matrix
 e(mm)=norm(E);
 c_numb(mm,:)=cond(H_acc);
 sv(mm,:)=diag(S)';
 inv_S=zeros(size(S));

 sv_used(mm)=n_ex;
 ll=1;
 corrected=zeros(n_ex,1);
    for jj=1:n_ex
    inv_S(jj,jj)=1/S(jj,jj);
    if correction==1
       singul=S(jj,jj);
       e_logic=1;
    else
       singul=max(diag(S))+10000000; %threshold
    end

    if  singul<e(mm)
       if  jj==1
          inv_S(jj,jj)=1/e(mm);
       else
          inv_S(jj,jj)=0;
          sv_used(mm)=n_ex-ll;
          ll=ll+1;
          corrected(jj)=1;
       end
    end
 end
 %corrected=zeros(n_ex,1);
 if ii==blocks
    if resp_accelerance==1
       ll=1;
       resp_error=abs(a_hat_avg(mm,:));
       resp_error_norm(:,mm)=norm(1/sqrt(blocks));
       sv_freq=sv(mm,:);
          for jj=1:n_ex

             Frob=sqrt(sum(sv_freq(jj:n_ex).^2));
             if Frob < sqrt(sum(sv_freq(1:n_ex).^2)). . .
                *resp_error_norm(:,mm)
                if corrected(jj)==0
                   if jj==1
                      inv_S(jj,jj)=1/S(jj,jj);
                   else
                      inv_S(jj,jj)=0;
                      sv_used(mm)=n_ex-ll;
                      ll=ll+1;
                   end
                else
                      sv_used(mm)=n_ex-ll;
                      ll=ll+1;
                end
             end
          end
    end
 end

 inv_Hacc=V*inv_S'*U';
 F(mm,:)=(inv_Hacc*a_hat(mm,:).').';
 if ii==blocks
```

```
        F_avg(mm,:)=(inv_Hacc*a_hat_avg(mm,:).').';
      end

    end % frequency

    for force_count=1:n_ex
      eval(['f' int2str(ii+(force_count-1)*blocks) '=' 'F(:,'. . .
            int2str(force_count) ');']);
      eval(['save c:\thite\ant\temp\' 'f' int2str(ii+. . .
            (force_count-1)*blocks) ' f' int2str(ii+. . .
            (force_count-1)*blocks)])
    end

end % blocks

operate_acc_freq=sum_a_hat/blocks;

clear temp_acc temp_signal_gen x y acc_temp Fs
clear N Sxy Syy sumSxy f_sxy sumSyy w freq wp x_fft y_fft

%Response reconstruction

est_resp1(length(f),blocks)=0;

for force_count=1:n_ex

    for kk=1:blocks
      eval(['load c:\thite\ant\temp\' 'f' int2str(kk+(force_count. .
            -1)*blocks)])
      eval(['F_ii(:,' int2str(kk) ')=' 'f' int2str(kk+(. . .
            force_count-1)*blocks) ';'])
      est_resp(:,kk)=F_ii(:,kk).*transfer_path(:,force_count);
    end
%
    est_resp1=est_resp+est_resp1;     % Adding of all the responses

    F_est_mean(:,force_count)=(mean(F_ii.')).';
    F_sort=(sort(F_ii.'));
    size(F_sort)
    temp=(F_sort(fix(0.16*blocks),:))';
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(fix(0.5*blocks),:))';

    % As practical case
    est_avg=(sum((transfer_path.*F_avg).')).';

    % Individual contributions
    est_ind_resp_cont=transfer_path.*F_avg;
if concept==3
    F_r=F_avg;
    est_mean=est_avg;
else
    for ii=1:n_ex
      F_r(:,ii)=F_avg(:,ii).*ref_signal_mag;

    force_measure_freq(:,ii)=force_measure_freq(:,ii).*ref_signal_mag;
    end
    rec_acc_freq=rec_acc_freq.*ref_signal_mag;
    est_mean=est_avg.*ref_signal_mag;
end
```

```
resp_number=[3 4 5 6];  % numbers upto 10 in set of 4 or 5 or ... 9

receiver_number=[2];    % any number from 1 to 10 other than above

pressure=0;             % Sound pressure or Vibration prediction

validation_number=[1];  % Cross validation response

break_program=0;
for ii=1:length(resp_number)
   if resp_number(ii)==validation_number;
      break_program=1;
   end
end
if break_program==1
   'Warning !!!!!! Same location used for validation and inversion'
   break
end

ref_signal=9;

concept=3;              % is 1 for H1, 2 for H2 and 3 for phase from
                        % cross spectra

correction=0;           % Whether singular values rejected
resp_accelerance=0;     % 1 - For responce based rejection, 2 for
                        % accelerance based rejection
coh_noise=1;            % Standard deviation for accelerance error

min_freq=30;
max_freq=1600;


load c:\thite\ant\tpa_meas1\onstud\series_exp\a8107135\frf_trof

clear frf_measure coh_measure      % Only retain frequency data

f=[min_freq:freq(2):max_freq]';
f_min=find(freq==min_freq);
f_max=find(freq==max_freq);

load c:\thite\ant\tpa_meas1\onstud\series_exp\a8107135\oper123

force_measure=operate_resp(:,1:3);

Fs=1/time(2);
N=Fs/freq(2);

n_resp=length(resp_number);
n_ex=length(force_measure(1,:));


% FFT of force signal
%
for kk=1:n_ex
   x=(force_measure(:,kk)-mean(force_measure(:,kk)));
   y=(operate_resp(:,ref_signal)-mean(operate_resp(:,ref_signal)));

   w=hanning(N);
   w=w(:);

   f_sxy=[0:N/2]'*Fs/N;
```

```
   sumSxy=zeros(size(f_sxy));
   sumSxx=zeros(size(f_sxy));
   sumSyy=zeros(size(f_sxy));

   blocks=floor(length(x)/N);

   for ii=1:blocks
      x_fft=fft(w.*x((ii-1)*N+1:N*ii));
      y_fft=fft(w.*y((ii-1)*N+1:N*ii));
    sumSxy=2*(conj(x_fft(1:N/2+1)).*(y_fft(1:N/2+1)))./(Fs*N)+sumSxy;
      sumSyy=2*((abs(y_fft(1:N/2+1))).^2)./(Fs*N)+sumSyy;
      sumSxx=2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N)+sumSxx;
   end
   wp=sum((w').^2)./N;
   Sxy=(sumSxy)./(blocks*wp);
   Syy=(sumSyy)./(blocks*wp);
   Sxx=(sumSxx)./(blocks*wp);
   if kk==2
      force_temp=zeros(size(Sxx));
   else
      if concept==1
         force_temp=(Sxy)./(Syy);                % H1 concept
      elseif concept==2
         force_temp=(Sxx)./(Sxy);                % H2 concept
      elseif concept==3
         force_temp=sqrt(Sxx).*exp(i*angle(Sxy));
      end
   end
   force_measure_freq(:,kk)=force_temp(f_min:f_max);
end

n_av=blocks;
clear signal_gen_force force_temp time_force x y


% Reading in FRF data

load c:\thite\ant\tpa_meas1\onstud\series_exp\a8107135\frf_trof
   %a8107135\frf_trof   a8106134 ap107135

frf_spacing=length(frf_measure(1,:))/n_ex;

for ii=1:n_ex
   for kk=1:n_resp
      temp_H_hat(:,kk+n_resp*(ii-1))=frf_measure. . .
      (:,resp_number(kk)+(ii-1)*frf_spacing);
      temp_coh_hat(:,kk+n_resp*(ii-1))=coh_measure. . .
      (:,resp_number(kk)+(ii-1)*frf_spacing);
   end
   temp_transfer_path(:,ii)=frf_measure. . .
   (:,receiver_number+(ii-1)*frf_spacing);
   temp_validation_path(:,ii)=frf_measure. . .
   (:,validation_number+(ii-1)*frf_spacing);
end

H_hat=temp_H_hat(f_min:f_max,:);
coh_hat=temp_coh_hat(f_min:f_max,:);
transfer_path=temp_transfer_path(f_min:f_max,:);
validation_path=temp_validation_path(f_min:f_max,:);

clear frf_measure coh_measure temp_H_hat temp_coh_hat
temp_transfer_path temp_validation_path


operate_acc=operate_resp(:,4:9);
```

```
for kk=1:n_resp
   temp_acc(:,kk)=(operate_acc(:,resp_number(kk)). . .
   -mean(operate_acc(:,resp_number(kk))));
end
   temp_acc(:,kk+1)=(operate_acc(:,receiver_number). . .
   -mean(operate_acc(:,receiver_number)));
   temp_acc(:,kk+2)=(operate_acc(:,validation_number). . .
   -mean(operate_acc(:,validation_number)));


% Receiver and validation measured response

for kk=n_resp+1:n_resp+2
   x=temp_acc(:,kk);
   y=(operate_resp(:,ref_signal)-mean(operate_resp(:,ref_signal)));
   w=hanning(N);
   w=w(:);

   f_sxy=[0:N/2]'*Fs/N;

   sumSxy=zeros(size(f_sxy));
   sumSxx=zeros(size(f_sxy));

   sumSyy=zeros(size(f_sxy));

   blocks=floor(length(x)./N);
   wp=sum((w').^2)./N;

   for ii=1:blocks
      x_fft=fft(w.*x((ii-1)*N+1:N*ii));
      y_fft=fft(w.*y((ii-1)*N+1:N*ii));
      sumSxy=2*(conj(x_fft(1:N/2+1)).*(y_fft(1:N/2+1). . .
      ))./(Fs*N)+sumSxy;
      sumSyy=2*((abs(y_fft(1:N/2+1))).^2)./(Fs*N)+sumSyy;
      sumSxx=2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N)+sumSxx;
   end
   Sxy=(sumSxy)./(blocks*wp);
   Syy=(sumSyy)./(blocks*wp);
   Sxx=(sumSxx)./(blocks*wp);

   ref_signal_mag=sqrt(Syy(f_min:f_max));     % Reference signal for
                                              % multiplication

   if concept==1
        acc_temp=(Sxy)./(Syy);                % H1 concept
   elseif concept==2
        acc_temp=(Sxx)./(Sxy);                % H2 concept
   elseif concept==3
        acc_temp=sqrt(Sxx).*exp(i*angle(Sxy));
   end
   if kk==n_resp+1
      rec_acc_freq=acc_temp(f_min:f_max);
   else
      vali_acc_freq=acc_temp(f_min:f_max);
   end
end

clear force_measure

sum_a_hat=zeros(length(f),n_resp);

for kk=1:n_resp
   x=temp_acc(:,kk);

   w=hanning(N);
   w=w(:);
```

```
   f_sxy=[0:N/2]'*Fs/N;

   sumSxy=zeros(size(f_sxy));
   sumSyy=zeros(size(f_sxy));
   sumSxx=zeros(size(f_sxy));

   blocks=floor(length(x)./N);

   wp=sum((w').^2)./N;

for ii=1:blocks
      x_fft=fft(w.*x((ii-1)*N+1:N*ii));
      y_fft=fft(w.*y((ii-1)*N+1:N*ii));

      sumSxy=2*(conj(x_fft(1:N/2+1)).*(y_fft(1:N/2+1)))./(. . .
            Fs*N)+sumSxy;
      sumSyy=2*((abs(y_fft(1:N/2+1))).^2)./(Fs*N)+sumSyy;
      sumSxx=2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N)+sumSxx;

      inst_Sxx=sqrt(2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N*wp));
      inst_acc(:,ii)=inst_Sxx;
   end
   Sxy=(sumSxy)./(blocks*wp);
   Syy=(sumSyy)./(blocks*wp);
   Sxx=(sumSxx)./(blocks*wp);
   if concept==1
         acc_temp=(Sxy)./(Syy);                % H1 concept
   elseif concept==2
         acc_temp=(Sxx)./(Sxy);                % H2 concept
   elseif concept==3
         acc_temp=sqrt(Sxx).*exp(i*angle(Sxy));
   end

   a_hat(:,kk)=acc_temp(f_min:f_max);
   measur_std(:,kk)=(std((inst_acc).'))';
end

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;


clear temp_acc  temp_acc_block temp_signal_gen temp_signal_gen_block
clear force_ref10 force_ref1_3 force_ref4_6 force_ref7_9 time


n_av=blocks;
averages=37;

for ii=1:averages
    averages_completed=ii
    for mm=1:length(f)
      for jj=1:n_ex
                 H_acc(:,jj)=H_hat(mm,(jj-1)*n_resp+1:jj*n_resp).';
                 coh_acc(:,jj)=coh_hat(mm,(jj-1)*n_resp. . .
                                 +1:jj*n_resp)';
      end
      E=coh_noise*sqrt((1-coh_acc)./(2*n_av.*coh_acc)).*abs(H_acc);
                                                      %error matrix
      nois_perturb=randn(size(H_acc)).*E.*exp(j*2*pi*rand. . .
                   (size(H_acc)));

      H_acc1=H_acc+nois_perturb;
```

```
    [U,S,V]=svd(H_acc1);
    e(mm)=norm(E);
    c_numb(mm,:)=cond(H_acc);
    sv(mm,:)=(diag(S)).';
    rec_sing=1./(diag(S));
    inv_S=zeros(size(S));        % Initialization
    for jj=1:n_ex
       inv_S(jj,jj)=rec_sing(jj);
    end

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Combining with singular value rejection
    if correction==1
          ll=1;
          resp_error=abs(a_hat(mm,:));
          resp_error_norm(mm)=norm(1/sqrt(blocks));
          sv_freq=sv(mm,:);
             for jj=1:n_ex
                inv_S(jj,jj)=1/S(jj,jj);
                Frob=sqrt(sum(sv_freq(jj:n_ex).^2));
                if Frob < S(1,1)*resp_error_norm(mm)
                      if jj==1
                         inv_S(jj,jj)=1/S(jj,jj);
                      else
                         inv_S(jj,jj)=0;
                         sv_used(mm)=n_ex-ll;
                         ll=ll+1;
                      end
                end
            end
       end

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


    a_force=a_hat(mm,:).'+randn(size(a_hat(mm,:).')).*abs. . .
       (a_hat(mm,:).').*exp(j*2*pi*rand(size(a_hat(mm,:).'). . .
       )).*/sqrt(blocks);
    inv_Hacc=V*inv_S'*U';
    F(mm,:)=(inv_Hacc*a_force).';
  end % frequency

  for force_count=1:n_ex
     eval(['f' int2str(ii+(force_count-1)*averages) '=' 'F. . .
        (:,' int2str(force_count) ');'']);
     eval(['save c:\thite\ant\temp\' 'f' int2str(ii+(. . .
        force_count-1)*averages) ' f' int2str(ii+. . .
        (force_count-1)*averages)])
     eval(['clear ' 'f' int2str(ii+(force_count-1)*averages) ])
  end
end

clear temp_acc temp_signal_gen x y acc_temp Fs
clear N Sxy Syy sumSxy f_sxy sumSyy w freq wp x_fft y_fft

%Response reconstruction

est_resp1(length(f),averages)=0;

for force_count=1:n_ex

   for kk=1:averages
      eval(['load c:\thite\ant\temp\' 'f' int2str(kk+. . .
      (force_count-1)*averages)])
```

```
      eval(['Force_ii=' 'f' int2str(kk+(force_count-1). . .
              *averages) ';'])
      est_resp=Force_ii.*transfer_path(:,force_count);
      eval(['clear ' 'f' int2str(kk+(force_count-1)*averages) ])
      est_resp1(:,kk)=est_resp+est_resp1(:,kk);     % Adding of all
                                                    % the responses
   end
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(fix(0.5*averages),:))';
   est_sort_min=(est_sort(fix(0.84*averages),:))';
   est_sort_max=(est_sort(fix(0.16*averages),:))';

   clear est_resp1 est_resp

for force_count=1:n_ex
   for kk=1:averages
      eval(['load c:\thite\ant\temp\' 'f' int2str(kk+(. . .
         force_count-1)*averages)])
      eval(['F_ii(:,' int2str(kk) ')=' 'f' int2str(kk+(. . .
         force_count-1)*averages) ';'])
      eval(['clear ' 'f' int2str(kk+(force_count-1)*averages) ])
    end
   F_est_mean(:,force_count)=(mean(F_ii.')).';
   F_sort=(sort(F_ii.'));
   temp=(F_sort(fix(0.16*averages),:))';
   F_est_max(:,force_count)=(F_sort(fix(0.84*averages),:))';
   F_est_min(:,force_count)=(F_sort(fix(0.16*averages),:))';
   clear F_ii
end

%Individual contributions
est_ind_resp_cont=transfer_path.*F_est_mean;

est_mean=est_direct_mean;
F_r=F_est_mean;
```

```
    f_sxy=[0:N/2]'*Fs/N;

    sumSxy=zeros(size(f_sxy));
    sumSxx=zeros(size(f_sxy));
    sumSyy=zeros(size(f_sxy));

    blocks=floor(length(x)/N);

    for ii=1:blocks
       x_fft=fft(w.*x((ii-1)*N+1:N*ii));
       y_fft=fft(w.*y((ii-1)*N+1:N*ii));
     sumSxy=2*(conj(x_fft(1:N/2+1)).*(y_fft(1:N/2+1)))./(Fs*N)+sumSxy;
       sumSyy=2*((abs(y_fft(1:N/2+1))).^2)./(Fs*N)+sumSyy;
       sumSxx=2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N)+sumSxx;
    end
    wp=sum((w').^2)./N;
    Sxy=(sumSxy)./(blocks*wp);
    Syy=(sumSyy)./(blocks*wp);
    Sxx=(sumSxx)./(blocks*wp);
    if kk==2
       force_temp=zeros(size(Sxx));
    else
       if concept==1
          force_temp=(Sxy)./(Syy);               % H1 concept
       elseif concept==2
          force_temp=(Sxx)./(Sxy);               % H2 concept
       elseif concept==3
          force_temp=sqrt(Sxx).*exp(i*angle(Sxy));
       end
    end
    force_measure_freq(:,kk)=force_temp(f_min:f_max);
end

n_av=blocks;
clear signal_gen_force force_temp time_force x y


% Reading in FRF data

load c:\thite\ant\tpa_meas1\onstud\series_exp\a8107135\frf_trof
   %a8107135\frf_trof   a8106134 ap107135

frf_spacing=length(frf_measure(1,:))/n_ex;

for ii=1:n_ex
   for kk=1:n_resp
      temp_H_hat(:,kk+n_resp*(ii-1))=frf_measure. . .
      (:,resp_number(kk)+(ii-1)*frf_spacing);
      temp_coh_hat(:,kk+n_resp*(ii-1))=coh_measure. . .
      (:,resp_number(kk)+(ii-1)*frf_spacing);
   end
   temp_transfer_path(:,ii)=frf_measure. . .
   (:,receiver_number+(ii-1)*frf_spacing);
   temp_validation_path(:,ii)=frf_measure. . .
   (:,validation_number+(ii-1)*frf_spacing);
end

H_hat=temp_H_hat(f_min:f_max,:);
coh_hat=temp_coh_hat(f_min:f_max,:);
transfer_path=temp_transfer_path(f_min:f_max,:);
validation_path=temp_validation_path(f_min:f_max,:);

clear frf_measure coh_measure temp_H_hat temp_coh_hat
temp_transfer_path temp_validation_path
```

Appendix E Program for Force Identification by Tikhonov regularization with OCV for regularization parameter selection

```
resp_number=[3 4 5 6];  % numbers upto 10 in set of 4 or 5 or ... 9

receiver_number=[2];    % any number from 1 to 10 other than above

pressure=0;             % Sound pressure or Vibration prediction

validation_number=[1];  % Cross validation response

break_program=0;
for ii=1:length(resp_number)
   if resp_number(ii)==validation_number;
      break_program=1;
   end
end
if break_program==1
   'Warning !!!!!! Same location used for validation and inversion'
   break
end

ref_signal=9;

concept=3;              % is 1 for H1, 2 for H2 and 3 for phase from
                        % cross spectra

correction=0;           % Whether singular values rejected
resp_accelerance=0;     % 1 - For responce based rejection, 2 for
                        % accelerance based rejection
coh_noise=1;            % Standard deviation for accelerance error

min_freq=30;
max_freq=1600;


load c:\thite\ant\tpa_meas1\onstud\series_exp\a8107135\frf_trof

clear frf_measure coh_measure       % Only retain frequency data

f=[min_freq:freq(2):max_freq]';
f_min=find(freq==min_freq);
f_max=find(freq==max_freq);

load c:\thite\ant\tpa_meas1\onstud\series_exp\a8107135\oper123

force_measure=operate_resp(:,1:3);

Fs=1/time(2);
N=Fs/freq(2);

n_resp=length(resp_number);
n_ex=length(force_measure(1,:));


% FFT of force signal
%
for kk=1:n_ex
   x=(force_measure(:,kk)-mean(force_measure(:,kk)));
   y=(operate_resp(:,ref_signal)-mean(operate_resp(:,ref_signal)));

   w=hanning(N);
   w=w(:);
```

```
operate_acc=operate_resp(:,4:9);

for kk=1:n_resp
   temp_acc(:,kk)=(operate_acc(:,resp_number(kk)). . .
   -mean(operate_acc(:,resp_number(kk))));
end
   temp_acc(:,kk+1)=(operate_acc(:,receiver_number). . .
   -mean(operate_acc(:,receiver_number)));
   temp_acc(:,kk+2)=(operate_acc(:,validation_number). . .
   -mean(operate_acc(:,validation_number)));


% Receiver and validation measured response

for kk=n_resp+1:n_resp+2
   x=temp_acc(:,kk);
   y=(operate_resp(:,ref_signal)-mean(operate_resp(:,ref_signal)));
   w=hanning(N);
   w=w(:);

   f_sxy=[0:N/2]'*Fs/N;

   sumSxy=zeros(size(f_sxy));
   sumSxx=zeros(size(f_sxy));

   sumSyy=zeros(size(f_sxy));

   blocks=floor(length(x)./N);
   wp=sum((w').^2)./N;

   for ii=1:blocks
      x_fft=fft(w.*x((ii-1)*N+1:N*ii));
      y_fft=fft(w.*y((ii-1)*N+1:N*ii));
      sumSxy=2*(conj(x_fft(1:N/2+1)).*(y_fft(1:N/2+1). . .
      )) ./(Fs*N)+sumSxy;
      sumSyy=2*((abs(y_fft(1:N/2+1))).^2)./(Fs*N)+sumSyy;
      sumSxx=2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N)+sumSxx;
   end
   Sxy=(sumSxy)./(blocks*wp);
   Syy=(sumSyy)./(blocks*wp);
   Sxx=(sumSxx)./(blocks*wp);

   ref_signal_mag=sqrt(Syy(f_min:f_max));      % Reference signal for
                                               % multiplication
   if concept==1
         acc_temp=(Sxy)./(Syy);                % H1 concept
   elseif concept==2
         acc_temp=(Sxx)./(Sxy);                % H2 concept
   elseif concept==3
         acc_temp=sqrt(Sxx).*exp(i*angle(Sxy));
   end
   if kk==n_resp+1
      rec_acc_freq=acc_temp(f_min:f_max);
   else
      vali_acc_freq=acc_temp(f_min:f_max);
   end
end

clear force_measure

sum_a_hat=zeros(length(f),n_resp);

for kk=1:n_resp
   x=temp_acc(:,kk);
```

```
   w=hanning(N);
   w=w(:);

   f_sxy=[0:N/2]'*Fs/N;

   sumSxy=zeros(size(f_sxy));
   sumSyy=zeros(size(f_sxy));
   sumSxx=zeros(size(f_sxy));

   blocks=floor(length(x)./N);

   wp=sum((w').^2)./N;

for ii=1:blocks
      x_fft=fft(w.*x((ii-1)*N+1:N*ii));
      y_fft=fft(w.*y((ii-1)*N+1:N*ii));

      sumSxy=2*(conj(x_fft(1:N/2+1)).*(y_fft(1:N/2+1)))./(. . .
            Fs*N)+sumSxy;
      sumSyy=2*((abs(y_fft(1:N/2+1))).^2)./(Fs*N)+sumSyy;
      sumSxx=2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N)+sumSxx;

      inst_Sxx=sqrt(2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N*wp));
      inst_acc(:,ii)=inst_Sxx;
   end
   Sxy=(sumSxy)./(blocks*wp);
   Syy=(sumSyy)./(blocks*wp);
   Sxx=(sumSxx)./(blocks*wp);
   if concept==1
         acc_temp=(Sxy)./(Syy);                % H1 concept
   elseif concept==2
         acc_temp=(Sxx)./(Sxy);                % H2 concept
   elseif concept==3
         acc_temp=sqrt(Sxx).*exp(i*angle(Sxy));
   end

   a_hat(:,kk)=acc_temp(f_min:f_max);
   measur_std(:,kk)=(std((inst_acc).'))';
end


for mm=1:length(f)
   if (mm/10)==fix(mm/10)
      frequency=f(mm)
   end
      for jj=1:n_ex
                  H_acc(:,jj)=H_hat(mm,(jj-1)*n_resp+1:jj*n_resp).';
                  coh_acc(:,jj)=coh_hat(mm,(jj-1)*n_resp. . .
                           +1:jj*n_resp)';
      end

      [U,S,V]=svd(H_acc);
      c_numb(mm,:)=cond(H_acc);
      percent_sing=1*S(1,1);
      min_percent=0;
      percent_sing=0.5;
      beta=[min_percent*S(n_ex,n_ex):(percent_sing*S(1,1). . .
         -min_percent*S(n_ex,n_ex))/199:percent_sing*S(1,1)].';
      Identity=diag(ones(size(1:n_ex)));  % Check this for diff
                                          % n_resp -overdetermination
      set1=[1:n_resp].';
      a_beta=a_hat(mm,:).';
      for kk=1:200
         ssd=0;
         for jj=1:n_resp
            set2=find(set1~=jj);
```

```
            for nn=1:n_resp-1
               H_reg(nn,:)=H_acc(set2(nn),:);
            end
            a_reg=a_beta(set2);
            if concept==3

            F_beta=(inv(H_reg'*H_reg+beta(kk)*Identity)*. . .
                  H_reg'*a_reg);
            ssd=(a_beta(jj)-H_acc(jj,:)*F_beta)*conj(a_beta. . .
                  (jj)-H_acc(jj,:)*F_beta)+ssd;
            else

            F_beta=(inv(H_reg'*H_reg+beta(kk)*Identity)*H_reg'. . .
                  *a_reg)*ref_signal_mag(mm);
            ssd=(a_beta(jj)*ref_signal_mag(mm)-H_acc(jj,:). . .
                 *F_beta)*conj(a_beta(jj)*ref_signal_mag(mm). . .
                  -H_acc(jj,:)*F_beta)+ssd;
            end

         end   %end of leave one out
         press(kk,:)=ssd/n_resp;
      end      %end beta
      [min_j index_min]=min(press);
      e(mm,:)=min_j;
      beta_freq(mm,:)=beta(index_min);
      F(mm,:)=(inv(H_acc'*H_acc+beta(index_min)*Identity). . .
         *H_acc'*a_beta).';

      mic_resp(mm,:)=transfer_path(mm,:)*(F(mm,:)).';
      if (f(mm)>166 & f(mm)<175)
         for rr=1:n_ex
          filter_char(rr,:)=S(rr,rr)^2/(S(rr,rr)^2+beta(index_min));
          end
          singular_vales_number=[1:n_ex].';
         figure
         plot(singular_vales_number,filter_char,'r')
         title(['Filter characteristic at ' num2str(f(mm)) ' Hz'])
         ylabel(' Weighting ')
         xlabel('Singular value')
         set(gca,'xtick',[1 2 3]');
         set(gca,'xticklabel',['1';'2';'3']);
         set(gca,'fontname','symbol');
         axis([1 n_ex 0 1])
      end

   end % frequency

%Response reconstruction

if concept==3
   F_r=F_r;
   est_mean=mic_resp;
else
   for ii=1:n_ex
      F_r(:,ii)=F_r(:,ii).*ref_signal_mag;

   force_measure_freq(:,ii)=force_measure_freq(:,ii).*ref_signal_mag;
   end
   rec_acc_freq=rec_acc_freq.*ref_signal_mag;
   est_mean=mic_resp.*ref_signal_mag;
end
```

Appendix F **Program for Force Identification by Iterative inversion with cross validation**

```
resp_number=[3 4 5 6];  % numbers upto 10 in set of 4 or 5 or ... 9

receiver_number=[2];    % any number from 1 to 10 other than above

pressure=0;             % Sound pressure or Vibration prediction

validation_number=[1];  % Cross validation response

break_program=0;
for ii=1:length(resp_number)
   if resp_number(ii)==validation_number;
      break_program=1;
   end
end
if break_program==1
   'Warning !!!!!! Same location used for validation and inversion'
   break
end

ref_signal=9;

concept=3;              % is 1 for H1, 2 for H2 and 3 for phase from
                        % cross spectra

correction=0;           % Whether singular values rejected
resp_accelerance=0;     % 1 - For responce based rejection, 2 for
                        % accelerance based rejection
coh_noise=1;            % Standard deviation for accelerance error

min_freq=30;
max_freq=1600;


load c:\thite\ant\tpa_meas1\onstud\series_exp\a8107135\frf_trof

clear frf_measure coh_measure       % Only retain frequency data

f=[min_freq:freq(2):max_freq]';
f_min=find(freq==min_freq);
f_max=find(freq==max_freq);

load c:\thite\ant\tpa_meas1\onstud\series_exp\a8107135\oper123

force_measure=operate_resp(:,1:3);

Fs=1/time(2);
N=Fs/freq(2);

n_resp=length(resp_number);
n_ex=length(force_measure(1,:));


% FFT of force signal
%
for kk=1:n_ex
   x=(force_measure(:,kk)-mean(force_measure(:,kk)));
   y=(operate_resp(:,ref_signal)-mean(operate_resp(:,ref_signal)));

   w=hanning(N);
   w=w(:);
```

```
   f_sxy=[0:N/2]'*Fs/N;

   sumSxy=zeros(size(f_sxy));
   sumSxx=zeros(size(f_sxy));
   sumSyy=zeros(size(f_sxy));

   blocks=floor(length(x)/N);

   for ii=1:blocks
      x_fft=fft(w.*x((ii-1)*N+1:N*ii));
      y_fft=fft(w.*y((ii-1)*N+1:N*ii));
    sumSxy=2*(conj(x_fft(1:N/2+1)).*(y_fft(1:N/2+1)))./(Fs*N)+sumSxy;
      sumSyy=2*((abs(y_fft(1:N/2+1))).^2)./(Fs*N)+sumSyy;
      sumSxx=2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N)+sumSxx;
   end
   wp=sum((w').^2)./N;
   Sxy=(sumSxy)./(blocks*wp);
   Syy=(sumSyy)./(blocks*wp);
   Sxx=(sumSxx)./(blocks*wp);
   if kk==2
      force_temp=zeros(size(Sxx));
   else
      if concept==1
         force_temp=(Sxy)./(Syy);                % H1 concept
      elseif concept==2
         force_temp=(Sxx)./(Sxy);                % H2 concept
      elseif concept==3
         force_temp=sqrt(Sxx).*exp(i*angle(Sxy));
      end
   end
   force_measure_freq(:,kk)=force_temp(f_min:f_max);
end

n_av=blocks;
clear signal_gen_force force_temp time_force x y


% Reading in FRF data

load c:\thite\ant\tpa_meas1\onstud\series_exp\a8107135\frf_trof
   %a8107135\frf_trof   a8106134 ap107135

frf_spacing=length(frf_measure(1,:))/n_ex;

for ii=1:n_ex
   for kk=1:n_resp
      temp_H_hat(:,kk+n_resp*(ii-1))=frf_measure. . .
      (:,resp_number(kk)+(ii-1)*frf_spacing);
      temp_coh_hat(:,kk+n_resp*(ii-1))=coh_measure. . .
      (:,resp_number(kk)+(ii-1)*frf_spacing);
   end
   temp_transfer_path(:,ii)=frf_measure. . .
   (:,receiver_number+(ii-1)*frf_spacing);
   temp_validation_path(:,ii)=frf_measure. . .
   (:,validation_number+(ii-1)*frf_spacing);
end

H_hat=temp_H_hat(f_min:f_max,:);
coh_hat=temp_coh_hat(f_min:f_max,:);
transfer_path=temp_transfer_path(f_min:f_max,:);
validation_path=temp_validation_path(f_min:f_max,:);

clear frf_measure coh_measure temp_H_hat temp_coh_hat
temp_transfer_path temp_validation_path
```

```
operate_acc=operate_resp(:,4:9);

for kk=1:n_resp
   temp_acc(:,kk)=(operate_acc(:,resp_number(kk)). . .
   -mean(operate_acc(:,resp_number(kk))));
end
   temp_acc(:,kk+1)=(operate_acc(:,receiver_number). . .
   -mean(operate_acc(:,receiver_number)));
   temp_acc(:,kk+2)=(operate_acc(:,validation_number). . .
   -mean(operate_acc(:,validation_number)));


% Receiver and validation measured response

for kk=n_resp+1:n_resp+2
   x=temp_acc(:,kk);
   y=(operate_resp(:,ref_signal)-mean(operate_resp(:,ref_signal)));
   w=hanning(N);
   w=w(:);

   f_sxy=[0:N/2]'*Fs/N;

   sumSxy=zeros(size(f_sxy));
   sumSxx=zeros(size(f_sxy));

   sumSyy=zeros(size(f_sxy));

   blocks=floor(length(x)./N);
   wp=sum((w').^2)./N;

   for ii=1:blocks
      x_fft=fft(w.*x((ii-1)*N+1:N*ii));
      y_fft=fft(w.*y((ii-1)*N+1:N*ii));
      sumSxy=2*(conj(x_fft(1:N/2+1)).*(y_fft(1:N/2+1). . .
      ))./(Fs*N)+sumSxy;
      sumSyy=2*((abs(y_fft(1:N/2+1))).^2)./(Fs*N)+sumSyy;
      sumSxx=2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N)+sumSxx;
   end
   Sxy=(sumSxy)./(blocks*wp);
   Syy=(sumSyy)./(blocks*wp);
   Sxx=(sumSxx)./(blocks*wp);

   ref_signal_mag=sqrt(Syy(f_min:f_max));       % Reference signal for
                                                % multiplication
   if concept==1
         acc_temp=(Sxy)./(Syy);                 % H1 concept
   elseif concept==2
         acc_temp=(Sxx)./(Sxy);                 % H2 concept
   elseif concept==3
         acc_temp=sqrt(Sxx).*exp(i*angle(Sxy));
   end
   if kk==n_resp+1
      rec_acc_freq=acc_temp(f_min:f_max);
   else
      vali_acc_freq=acc_temp(f_min:f_max);
   end
end

clear force_measure

sum_a_hat=zeros(length(f),n_resp);

for kk=1:n_resp
   x=temp_acc(:,kk);
```

```
    w=hanning(N);
    w=w(:);

    f_sxy=[0:N/2]'*Fs/N;

    sumSxy=zeros(size(f_sxy));
    sumSyy=zeros(size(f_sxy));
    sumSxx=zeros(size(f_sxy));

    blocks=floor(length(x)./N);

    wp=sum((w').^2)./N;

for ii=1:blocks
        x_fft=fft(w.*x((ii-1)*N+1:N*ii));
        y_fft=fft(w.*y((ii-1)*N+1:N*ii));

        sumSxy=2*(conj(x_fft(1:N/2+1)).*(y_fft(1:N/2+1)))./(. . .
                Fs*N)+sumSxy;
        sumSyy=2*((abs(y_fft(1:N/2+1))).^2)./(Fs*N)+sumSyy;
        sumSxx=2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N)+sumSxx;

        inst_Sxx=sqrt(2*((abs(x_fft(1:N/2+1))).^2)./(Fs*N*wp));
        inst_acc(:,ii)=inst_Sxx;
    end
    Sxy=(sumSxy)./(blocks*wp);
    Syy=(sumSyy)./(blocks*wp);
    Sxx=(sumSxx)./(blocks*wp);
    if concept==1
            acc_temp=(Sxy)./(Syy);              % H1 concept
    elseif concept==2
            acc_temp=(Sxx)./(Sxy);              % H2 concept
    elseif concept==3
            acc_temp=sqrt(Sxx).*exp(i*angle(Sxy));
    end

    a_hat(:,kk)=acc_temp(f_min:f_max);
    measur_std(:,kk)=(std((inst_acc).'))';
end


a_hat_full=a_hat;
a_hat_full(:,n_resp+1)=vali_acc_freq;

max_std=1;
std_div=0.05;
divisions=20;

iteration_divion=[250 500 1000 2000 4000 8000 16000 32000 . . .
               64000 128000 256000 512000 512000*2 512000*4]';

for ii=1:length(f)
    for jj=1:n_ex
       H_acc_full(:,jj)=H_hat(ii,(jj-1)*(n_resp)+1:jj*(n_resp)).';
   end
       set2=[1:n_resp].';
       a_hat_freq=a_hat(ii,:).';
    for ocv_count=1:n_resp
       set1=find(set2~=ocv_count);
       for nn=1:n_resp-1
          H_acc(nn,:)=H_acc_full(set1(nn),:);
       end
       H_valid=H_acc_full(ocv_count,:);
       if concept==3
          a_valid=a_hat_freq(ocv_count);
          a_ocv=(a_hat_freq(set1).');
```

```
else
   a_valid=a_hat_freq(ocv_count)*ref_signal_mag(ii);
   a_ocv=(a_hat_freq(set1).')*ref_signal_mag(ii);
end

[U,S,V]=svd(H_acc);
[U1,S1,V1]=svd(H_acc_full);
c_numb(ii,:)=cond(H_acc);
I=eye(n_resp-1,n_ex);
I=I*I';
psi_iterate=zeros(size(S'));
psi_valid=psi_iterate;
var_inv_PI=zeros(size(U));
dont_run=1;
loop_run=1;
loop_break=1000000;
starter_sum=0;
nn=0;
pp=1;
opti_std_logic=0;
old_freq=f(ii)-f(2);
measur_std=zeros(size(a_ocv.'));
while (opti_std_logic==0)

 if pp > (divisions-1)
  break
 end
 measur_std=(max_std-(pp-1)*std_div).*ones. . .
             (size(a_ocv.')).*max(abs(a_ocv.'));
 std_iter(pp,:)=(max_std-(pp-1)*std_div);
 const_b=1; %0.5;
 beta(pp,:)=const_b./(S(1,1)^2);
 norm_std(pp,:)=max(measur_std);
 run_if=1;
 kk_break=100000000;
 kk=1;
 break_qq=0;
 cross_over=0;
for qq=1:length(iteration_divion)
   for mm=1:n_ex
      var_inv_PI(mm,mm)=(1-(1-beta(pp,:). . .
                         *S(mm,mm).^2)^(iteration_divion(qq)));
    end
    delta_k=U*var_inv_PI*U';
    J11_temp(qq,:)=(U*(var_inv_PI-I)*U'*abs(measur_std). . .
               ).'*conj(U*(var_inv_PI-I)*U'*abs(measur_std));

    J_cost1_temp(qq,:)=(U*(var_inv_PI-I)*U'*abs. . .
          (a_ocv).').'*conj(U*(var_inv_PI-I)*U'*abs(a_ocv. . .
          ).')-J11_temp(qq);
    J_cost2_temp(qq,:)=((delta_k)*measur_std).'*conj((. . .
      delta_k)*measur_std);
   if (J_cost2_temp(qq,:) >=J_cost1_temp(qq,:))
      cross_over=1;
   end
 if qq==1
   total_j_cost_temp_old=J_cost1_temp(qq,:)+J_cost2_temp(qq,:);
   total_j_cost_temp_new=0;
 else
   total_j_cost_temp_old=J_cost1_temp(qq-1,:)+. . .
                            J_cost2_temp(qq-1,:);
   total_j_cost_temp_new=J_cost1_temp(qq,:)+J_cost2_temp(qq,:);
 end

if (total_j_cost_temp_new >=total_j_cost_temp_old & . . .
                            cross_over==1)
```

```
      resol=2.^(qq-1);
      clear J_cost2_temp J_cost1_temp J11_temp
      break;
   elseif (qq==length(iteration_divion))
      resol=2.^(qq-1);
      clear J_cost2_temp J_cost1_temp J11_temp
      break
   end
   if break_qq==1
        break
   end

end
   resol_pp(pp)=resol;
   a_hat_iter=a_hat(ii,:);

   const_b=1;
   cross_over=0;
   while (loop_run==1)
      for mm=1:n_ex
         var_inv_PI(mm,mm)=(1-(1-beta(pp,:). . .
                           *S(mm,mm).^2)^(resol*(kk)+nn));
     end
     delta_k=U*var_inv_PI*U';

      %Bias error

      J11(kk,:)=(U*(var_inv_PI-I)*U'*measur_std. . .
         ).'*conj(U*(var_inv_PI-I)*U'*measur_std);
      J_cost1(kk,:)=(U*(var_inv_PI-I)*U'*abs(a_ocv. . .
         ).').'*conj(U*(var_inv_PI-I)*U'*abs(a_ocv).')-J11(kk,:);

      % Amplified variance

      J_cost2(kk,:)=((delta_k)*measur_std).'*conj((delta_k. . .
                     )*measur_std);
   if (J_cost2(kk) >=J_cost1(kk))
       cross_over=1;
    end

    if kk==1
       total_j_cost_old=J_cost1(kk,:)+J_cost2(kk,:);
       total_j_cost_new=0;
    else
       total_j_cost_old=J_cost1(kk-1,:)+J_cost2(kk-1,:);
       total_j_cost_new=J_cost1(kk,:)+J_cost2(kk,:);
    end

    allowed_error=100*abs(total_j_cost_old-total_j_cost_new. . .
                 )./(measur_std'*measur_std);
    if (allowed_error < 0.25 & run_if==1 & cross_over==1)
         if pp==1
            if f(ii)~=old_freq
               [' Condition number= ' num2str(c_numb(ii,:)). . .
             ' Frequency= ' num2str(f(ii)) ' Left out number='. .
             num2str(ocv_count) ]
               old_freq=f(ii);
            end
         end
         kk_break=kk+50;
         kk
         opti_valid(pp,:)=kk;
         run_if=0;
         cross_over=1;
      end
      if kk==kk_break
```

```
            break
         end
         kk=kk+1;
      end

      for mm=1:n_ex
         psi_valid(mm,mm)=(1-(1-beta(pp,:)*S(mm,mm).^2)^. . .
                              (resol*opti_valid(pp)+nn))./S(mm,mm);
      end
      F_valid=V*psi_valid*U'*a_ocv.';
      mic_valid1=H_valid*F_valid;
      valid_diff(pp,ocv_count)=100*(abs(abs(mic_valid1)-abs(. . .
                                  a_valid))./abs(a_valid));

      if pp==1
         J_cost_temp1=J_cost1;
         J_cost_temp2=J_cost2;
      end
      pp=1+pp;
      clear J_cost1 J_cost2

   end

end   %end of leave one out



press=(sum(valid_diff.')).'/n_resp;
[min_m min_n]=min(abs(valid_diff));
[min_valid min_ind]=min(min(abs(valid_diff)));
index_valid=min_n(min_ind);
opti_ve(:,ii)=min_valid;
opti_std(ii,:)=std_iter(index_valid);

% Program for force identification

cross_over=0;
I=eye(n_resp,n_ex);
I=I*I';
var_inv_PI=zeros(size(U1));
if concept==3
   a_hat_freq1=(abs(a_hat_freq));
else
   a_hat_freq1=(abs(a_hat_freq))*ref_signal_mag(ii);
end


      measur_std=opti_std(ii,:).*ones(size(a_hat_freq)).*max. . .
                  (abs(a_hat_freq1));
      const_b=1;
      beta_opti=const_b./(S1(1,1)^2);
      run_if=1;
      kk_break=100000000;
      kk=1;
      break_qq=0;
      for ss=1:length(iteration_divion)
         for mm=1:n_ex
            var_inv_PI(mm,mm)=(1-(1-beta_opti*S1(mm,mm).^. . .
                              2)^(iteration_divion(ss)));
          end
          delta_k=U1*var_inv_PI*U1';

          J11_temp1(ss,:)=(U1*(var_inv_PI-I)*U1'*measur_std. . .
                      ).'*conj(U1*(var_inv_PI-I)*U1'*measur_std);

         J_cost1_temp1(ss,:)=(U1*(var_inv_PI-I)*U1'*abs. . .
```

```
        (a_hat_freq1)).'*conj(U1*(var_inv_PI-I)*U1'*abs. . .
        (a_hat_freq1))-J11_temp1(ss);
        J_cost2_temp1(ss,:)=((delta_k)*measur_std).'. . . .
                                    *conj((delta_k)*measur_std);

   if ss==1
      total_j_cost_temp_old=J_cost1_temp1(ss,:). . .
                                    +J_cost2_temp1(ss,:);
      total_j_cost_temp_new=0;
   else
      total_j_cost_temp_old=J_cost1_temp1(ss-. . . .
                                    1,:)+J_cost2_temp1(ss-1,:);
      total_j_cost_temp_new=J_cost1_temp1(ss,:)+. . . .
                                    J_cost2_temp1(ss,:);
   end


     if (total_j_cost_temp_new >=total_j_cost_temp_old)

       resol=2.^(ss-1);
       clear J_cost2_temp1 J_cost1_temp1 J11_temp1
       break
     elseif (ss==length(iteration_divion))
       resol=2.^(ss-1);
       clear J_cost2_temp1 J_cost1_temp1 J11_temp1
     end
    end
    const_b=1;
while (loop_run==1)

     for mm=1:n_ex
        var_inv_PI(mm,mm)=(1-(1-beta_opti*S1. . . .
                             (mm,mm).^2)^(resol*(kk)+nn));
    end
    delta_k=U1*var_inv_PI*U1';

     %Bias error

     J11(kk,:)=(U1*(var_inv_PI-I)*U1'*measur_std).'*. . .
              conj(U1*(var_inv_PI-I)*U1'*measur_std);

     J_cost1(kk,:)=(U1*(var_inv_PI-I)*U1'*abs(a_hat_freq1). . .
     ).'*conj(U1*(var_inv_PI-I)*U1'*abs(a_hat_freq1))-J11(kk,:);
     J_cost2(kk,:)=((delta_k)*measur_std).'*conj(. . .
           (delta_k)*measur_std);

   if kk==1
      total_j_cost_old=J_cost1(kk,:)+J_cost2(kk,:);
      total_j_cost_new=0;
   else
      total_j_cost_old=J_cost1(kk-1,:)+J_cost2(kk-1,:);
      total_j_cost_new=J_cost1(kk,:)+J_cost2(kk,:);
   end

   if (J_cost2(kk) >=J_cost1(kk))
      cross_over=1;
   end
  allowed_error=100*abs(total_j_cost_old-. . .
                total_j_cost_new)./(measur_std'*measur_std);

  if (allowed_error < 0.1 & run_if==1 & cross_over==1)
        kk_break=kk+10;
        opti_kk(ii,:)=kk;
        run_if=0;
        cross_over=1;
     end
```

```
            if kk==kk_break
               break
            end
            kk=kk+1;
         end

   index_j=opti_kk(ii);
   psi_iterate=zeros(length(V1(1,:)),length(U1(:,1)));

   for mm=1:n_ex
      psi_iterate(mm,mm)=(1-(1-beta_opti*S1(mm,mm).^2)^. . .
                        (resol*(index_j)+nn))./S1(mm,mm);
   end

   if (f(ii)>166 & f(ii)<175)
         for rr=1:n_ex
             filter_char(rr,:)=(1-(1-beta_opti*S1(rr,rr. . .
                              ).^2)^(resol*(index_j)+nn));
          end
         singular_vales_number=[1:n_ex].';
         figure
         plot(singular_vales_number,filter_char,'r')
         title(['Filter characteristic at ' num2str(f(ii)) ' Hz'])
         ylabel(' Weighting ')
         xlabel('Singular value')
         set(gca,'xtick',[1 2 3]');
         set(gca,'xticklabel',['1';'2';'3']);
         set(gca,'fontname','symbol');
         axis([1 n_ex 0 1])
   end


   F_opt=V1*psi_iterate*U1'*a_hat_freq;
   mic_resp(ii,:)=transfer_path(ii,:)*F_opt;
   F_r(ii,:)=F_opt.';
   clear J_cost  J_cost1  J_cost2
   if ii==1
      %break
   end

end % frequency

break
clear temp_acc temp_signal_gen x y acc_temp Fs
clear N Sxy Syy sumSxy f_sxy sumSyy w freq wp x_fft y_fft

%Response reconstruction
%
%

if concept==3
   F_r=F_r;
   est_mean=mic_resp;
else
   for ii=1:n_ex
      F_r(:,ii)=F_r(:,ii).*ref_signal_mag;

   force_measure_freq(:,ii)=force_measure_freq(:,ii).*ref_signal_mag;
   end
   rec_acc_freq=rec_acc_freq.*ref_signal_mag;
   est_mean=mic_resp.*ref_signal_mag;
end
```

```
if concept==3
   F_r=F_avg;
   est_mean=est_avg;
else
   for ii=1:n_ex
      F_r(:,ii)=F_avg(:,ii).*ref_signal_mag;

   force_measure_freq(:,ii)=force_measure_freq(:,ii).*ref_signal_mag;
   end
   rec_acc_freq=rec_acc_freq.*ref_signal_mag;
   est_mean=est_avg.*ref_signal_mag;
end
if correction==1
   title_select=' Responses -by S V rejection'
   if coh_noise==1
      std_select=' +/- one std devn';
   elseif coh_noise==3
      std_select=' +/- 3 std devn';
   end
else
   title_select=' '
   std_select=' '
end

combi=([num2str(resp_number), ' - ',num2str(receiver_number)])

max_f(1)=max(max(abs(force_measure_freq)));
max_f(2)=max(max(abs(F_r)));
min_f(1)=min(min(abs(force_measure_freq)));
min_f(2)=min(min(abs(F_r)));

figure
subplot(2,1,1)
for ii=1:n_ex
   if ii==2
      title([' Reconstructed Forces - ',num2str(n_ex),' . . .
         forces and ',num2str(n_resp),title_select std_select])
      subplot(2,1,2)
   elseif ii==4
      title([' Reconstructed Forces - ',num2str(n_ex),' . . .
         forces and ',num2str(n_resp),title_select std_select])
      subplot(2,1,2)
   end
   loglog(f,abs(F_r(:,ii)),'b--',f,abs(force_measure_freq(:,ii)),'r')
   legend('Reconsructed','Measured')
   axis([min(f) max(f) min(min_f)*1.25 max(max_f)])
   ylabel(['Force ',num2str(ii),' [ N ]'])
   if ii==2
      xlabel(' Frequency Hz')
      figure
      subplot(2,1,1)
   elseif ii==4
      xlabel(' Frequency Hz')
   end
end

normal_force=(sqrt(sum((abs(force_measure_freq))'.^2)))';
figure
subplot(2,1,1)
for ii=1:n_ex
   if ii==2
```

```
      title([' Reconstructed Forces - ',num2str(n_ex),' . . .
         forces and ',num2str(n_resp),title_select std_select])
      subplot(2,1,2)
   elseif ii==4
      title([' Reconstructed Forces - ',num2str(n_ex),' . . .
         forces and ',num2str(n_resp),title_select std_select])
      subplot(2,1,2)
   end
   loglog(f,abs(F_r(:,ii))./normal_force,'b--',f,abs(. . .
      force_measure_freq(:,ii))./normal_force,'r')
   legend('Reconsructed','Measured')
   axis([min(f) max(f) 0.01/max(normal_force) 100./max. . .
      (normal_force)])
   ylabel(['Normalized Force ',num2str(ii)])
   if ii==2
      xlabel(' Frequency Hz')
      figure
      subplot(2,1,1)
   elseif ii==4
      xlabel(' Frequency Hz')
   end
end


if pressure==1
   max_v(1)=max(max(abs(est_mean./(2*10^(-5)))));
   max_v(2)=max(max(abs(rec_acc_freq./(2*10^(-5)))));
   min_v(1)=min(min(abs(est_mean./(2*10^(-5)))));
   min_v(2)=min(min(abs(rec_acc_freq./(2*10^(-5)))));


   figure
   subplot(2,1,1)
   loglog(f,abs(est_mean./(2*10^(-5))),'b--',f,abs(. . .
      rec_acc_freq./(2*10^(-5))),'r')
   legend('Reconsructed','Measured')
   title([' Velocity response - ',num2str(n_ex),' . . .
      forces and ',num2str(n_resp),title_select std_select])
   ylabel(' Velocity m/s')
   %xlabel(' Frequency Hz')
   axis([min(f) max(f) min(min_v) max(max_v)])


   [rec_acc_freq_oct c_freq]=oct3rdlg(rec_acc_freq./(. . .
      2*10^(-5)),f,13);
   [est_mean_oct c_freq]=oct3rdlg(est_mean./(2*10^(-5)),f,13);


   mean_error=sum(abs(est_mean_oct-rec_acc_freq_oct. . .
      ))./length(est_mean_oct);

   subplot(2,1,2)
   semilogx(c_freq,(est_mean_oct),'b--',c_freq,(. . .
      rec_acc_freq_oct),'r')
   legend('Reconsructed','Measured')
   title([' Velocity response in 1/3 octave bands - Average . . .
      error =' num2str(mean_error) ])
   ylabel(' Velocity level dB [re 1m/s]')
   xlabel(' Frequency Hz')
   axis([min(f) max(f) 40 80])

%Individual contribution from forces
   for ii=1:n_ex

   Actual_contri(:,ii)=transfer_path(:,ii).*force_measure_freq(:,ii);
   Estimate_contri(:,ii)=transfer_path(:,ii).*F_r(:,ii);
```

```
    [est_contri_oct c_freq]=oct3rdlg(Estimate_contri(:,ii)./(. . .
          2*10^(-5)),f,13);
      est_oct(:,ii)=est_contri_oct;
    end
    figure
     semilogx(c_freq,(est_oct(:,1)),'r',c_freq,(est_oct(:,2)),'. . .
       b--',c_freq,(est_oct(:,3)),'g-.')
     legend('Force 1','Force 2', 'Force 3')
     title([' Velocity contribution in 1/3 octave bands from. . .
       individual Forces ' ])
      ylabel(' Vel dB [re 1m/s]')
      xlabel(' Frequency Hz')
      axis([min(f) max(f) 30 80])

    for ii=1:n_ex

    Actual_contri(:,ii)=transfer_path(:,ii).*force_measure_freq(:,ii);
    Estimate_contri(:,ii)=transfer_path(:,ii).*F_r(:,ii);
    [est_contri_oct c_freq]=oct3rdlg(Estimate_contri(:,ii)./(2*10. . .
       ^(-5)),f,13);
      if length(find(Actual_contri(:,ii)~=0))==0
        Act_contri_oct=zeros(size(est_contri_oct));
     else
        [Act_contri_oct c_freq]=oct3rdlg(Actual_contri(:,ii)./(2*10. .
          ^(-5)),f,13);
     end
     mean_error=sum(abs(est_contri_oct-Act_contri_oct). . .
       )./length(est_contri_oct);
     if ii==1
        figure
     end
       subplot(n_ex,1,ii)
       semilogx(c_freq,(est_contri_oct),'b--',c_freq,(. . .
          Act_contri_oct),'r')
       legend('Reconsructed','Measured')
       if ii==1
          title([' Velocity contribution in 1/3 octave bands. . .
           from Force ' num2str(ii) ' and Avg error ='. . .
           num2str(mean_error) ])
       else
         title(['                                              . . .
          Force ' num2str(ii) ' and Avg error =' num2str. . .
          (mean_error) ])
       end
       ylabel(' Vel dB [re 1m/s]')
       if ii==n_ex
         xlabel(' Frequency Hz')
       end
        axis([min(f) max(f) 30 80])

    end


else


omega=f*2*pi;

max_v(1)=max(max(abs(est_mean./(j*omega))));
max_v(2)=max(max(abs(rec_acc_freq./(j*omega))));
min_v(1)=min(min(abs(est_mean./(j*omega))));
min_v(2)=min(min(abs(rec_acc_freq./(j*omega))));


figure
subplot(2,1,1)
```

```
loglog(f,abs(est_mean./(j*omega)),'b--',f,abs(. . .
   rec_acc_freq./(j*omega)),'r')
legend('Reconsructed','Measured')
title([' Velocity response - ',num2str(n_ex),' forces and. . .
   ',num2str(n_resp),title_select std_select])
ylabel(' Velocity m/s')
axis([min(f) max(f) min(min_v) max(max_v)])


[rec_acc_freq_oct c_freq]=oct3rdlg(rec_acc_freq./(j*omega),f,13);
[est_mean_oct c_freq]=oct3rdlg(est_mean./(j*omega),f,13);


mean_error=round(10*sqrt(sum((abs(est_mean_oct-rec_acc_freq_oct. .
   )).^2)./length(est_mean_oct)))/10

subplot(2,1,2)
semilogx(c_freq,(est_mean_oct),'b--',c_freq,(rec_acc_freq_oct),'r')
legend('Reconsructed','Measured')
title([' Velocity response in 1/3 octave bands - Average error. . .
   =' num2str(mean_error) ])
ylabel(' Velocity level dB [re 1m/s]')
xlabel(' Frequency Hz')
axis([min(f) max(f) -50 -20])

%Individual contribution from forces
for ii=1:n_ex
   Actual_contri(:,ii)=transfer_path(:,ii).*force_measure_freq(:,ii);
   Estimate_contri(:,ii)=transfer_path(:,ii).*F_r(:,ii);
   [est_contri_oct c_freq]=oct3rdlg(Estimate_contri. . .
      (:,ii)./(j*omega),f,13);
   est_oct(:,ii)=est_contri_oct;
end
figure
   semilogx(c_freq,(est_oct(:,1)),'r',c_freq,(est_oct(:,2)). . .
      ,'b--',c_freq,(est_oct(:,3)),'g-.')
   legend('Force 1','Force 2', 'Force 3')
   title([' Velocity contribution in 1/3 octave bands from . . .
      individual Forces ' ])
   ylabel(' Vel dB [re 1m/s]')
   xlabel(' Frequency Hz')
   axis([min(f) max(f) -75 -5])

for ii=1:n_ex
   Actual_contri(:,ii)=transfer_path(:,ii).*force_measure_freq(:,ii);
   Estimate_contri(:,ii)=transfer_path(:,ii).*F_r(:,ii);
   [est_contri_oct c_freq]=oct3rdlg(Estimate_contri. . .
      (:,ii)./(j*omega),f,13);
   if length(find(Actual_contri(:,ii)~=0))==0
      Act_contri_oct=rec_acc_freq_oct;
      mean_error=round(10*sqrt(sum((abs(est_contri_oct-. . .
         Act_contri_oct)).^2)./length(est_contri_oct)))/10;
   else
      [Act_contri_oct c_freq]=oct3rdlg(. . .
         Actual_contri(:,ii)./(j*omega),f,13);
      mean_error=round(10*sqrt(sum((abs(est_contri_oct. . .
         -Act_contri_oct)).^2)./length(est_contri_oct)))/10;
   end
   contri_error(:,ii)=mean_error;
   if ii==1
      figure
   end
   subplot(n_ex,1,ii)
   semilogx(c_freq,(est_contri_oct),'b--',c_freq,. . .
      (Act_contri_oct),'r')
   legend('Reconsructed','Measured')
```

```
   if ii==1
      title([' Velocity contribution in 1/3 octave bands from. . .
       Force ' num2str(ii) ' and Avg error =' num2str(mean_error) ])
   else
      title(['                                                   . . .
       Force ' num2str(ii) ' and Avg error =' num2str(mean_error) ])
   end
   ylabel(' Vel dB [re 1m/s]')
   if ii==n_ex
      xlabel(' Frequency Hz')
   end
   axis([min(f) max(f) -75 -5])

end
contri_error
end

if correction==1
   figure
   subplot(2,1,1)
   loglog(f,c_numb,'r')
   title([' Condition number '])
   ylabel(' Condition number')
   axis([min(f) max(f) 0 max(c_numb)*1.3])

   subplot(2,1,2)
   semilogx(f,sv_used,'r')
   title([' Singular values used '])
   ylabel(' Rank of matrix')
   xlabel(' Frequency Hz')
   axis([min(f) max(f) 0 max(sv_used)])
else
   figure
   loglog(f,c_numb,'r')
   title([' Condition number ( ' combi ' )'])
   ylabel(' Condition number')
   xlabel(' Frequency Hz')
   set(gca,'ytick',[1 10 100 500]');
   set(gca,'yticklabel',['  1';' 10';'100';'500']);
   set(gca,'fontname','symbol');
   axis([min(f) max(f) 1 500])
end
```