CHALMERS
UNIVERSITY OF TECHNOLOGY

Finite element modelling and calibration of a shake rig for durability verification

Master's thesis in Applied Mechanics

Choudhury Dipesh Rohan
Shruti Laxmikant Vaidya

Department of Mechanics and Maritime Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Master's thesis 2019:36

Finite element modelling and calibration of a shake rig for durability verification

CHOUDHURY DIPESH ROHAN
SHRUTI LAXMIKANT VAIDYA

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and Maritime Sciences
Division of Applied Mechanics
Chalmers University of Technology
Gothenburg, Sweden 2019

Finite element modelling and calibration of a shake rig for durability verification

Supervisor: Hjalmar Sandberg, Durability and Safety-Cab and Chassis, Volvo GTT
Anders Nord, Durability and Safety-Cab and Chassis, Volvo GTT
Mladen Gibanica, Department of Mechanics and Maritime Sciences
Examiner: Thomas Abrahamsson, Department of Mechanics and Maritime Sciences

Master's Thesis 2019:36
Department of Mechanics and Maritime Sciences
Division of Applied Mechanics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: 7 post chassis durability shake rig on which the frame is mounted for experimental testing.

Typeset in LaTeX
Printed by [Chalmers University of Technology]
Gothenburg, Sweden 2019

Finite element modelling and calibration of a shake rig for durability verification

CHOUDHURY DIPESH ROHAN
SHRUTI LAXMIKANT VAIDYA
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Abstract

This study is about calibration of the finite element model of the 7-post durability shake rig at Volvo GTT, Lundby. The study was carried out to compare the dynamic behaviour of the physical rig with its finite element model and establish a correlation between them. The calibration is done against experimental data obtained from experimental modal analysis (EMA) by employing an optimization algorithm based on the minimization of the deviation metric between the transfer functions of the parametrized finite element model and the physical rig. The system is excited in two different setups: Hammer Test and 7-post rig test. In hammer test, force input is given while in 7-post rig test displacement input is given. Force is measured for corresponding displacement input using force transducers. Signal processing is done to experimental data. Frequency response functions are synthesized from the accelerance data of the EMA. Post-screening of the measured experimental data is done to cure test data of any errors in estimation. The system is identified using state-space method. The accuracy of the state-space model is improved by developing a MATLAB script that captured the relevant eigenmodes of the identified system. The finite element model is calibrated based on the identified state-space model. The dynamics of the calibrated model is found to be close to the physical rig. The calibrated model improved the correlation of fourth flexible eigenmode while retaining the behaviour of the first three eigenmodes.

KEYWORDS: Experimental Modal Analysis, Frequency response function, 7-post Test Rig, Hammer Test, system identification, calibration.

Acknowledgements

We want to express our gratitude to Volvo GTT for providing us the opportunity to conduct this study, which enhanced our knowledge of structural dynamics and modal analysis. We want to thank our supervisors at Volvo GTT, Hjalmar Sandberg, and Anders Nord for their advice and guidance, which was invaluable while performing the study. Their guidance kept the study in the right direction. We are indebted to our supervisor at Chalmers Mladen Gibanica, for his support and advice during the whole thesis. We want to thank him, for the MATLAB scripts he provided, for the fruitful discussions we had and for his help. Our special thanks to Anders Johansson for his support and help while doing the rig tests.

We want to express our appreciation to our Thesis Examiner, Professor Thomas Abrahamsson, for supporting us when we faced roadblocks in the thesis and guiding us to the successful completion of the thesis. We are also indebted to Jennie Sandhal for the support which helped in the successful completion of the project.

We also want to thank anyone who's reading this report; you're making the effort we put into writing it worthwhile.

CHOUDHURY DIPESH ROHAN, Gothenburg, August 2019
SHRUTI LAXMIKANT VAIDYA, Gothenburg, August 2019

Contents

List of Figures

List of Tables

1
Introduction

This chapter gives brief overview of the framework of the study. It contains five sections. Section 1.1 gives the background behind the issue to be investigated. Section 1.2 and 1.3 gives a brief description of the study and its intended outcome. Section 1.4 describes the structure of the work. The limitations of the research work are described in Section 1.5.

1.1 Background

The durability lab of Volvo Group Trucks Technology in Lundby, Gothenburg, has capabilities of performing highly advanced tests on different systems and components of the truck. A key facility is the chassis shake rig which is used to perform durability tests on complete cabs and chassis components such as battery boxes and fuel tanks.The chassis shake rig consists of 7 hydraulic actuators, the movement of which is controlled to reproduce a vibration response on the test object measured on a reference truck. The study is being carried out to compare the dynamic behaviour of the physical rig with its calculation model (finite element model) and establish a correlation between them.

Testing in physical rigs can be a both time consuming and expensive. Virtual rig tests can reduce the lead time of an actual rig test, providing more value to the test results. Virtual rigs allow the possibility of running different load case simulations and iterations on a finite element model of the rig. The most important simulations from these, can be chosen to be tested in the test rigs thereby reducing the time and effort needed for a test and thus placing the company ahead of its competitors.

1.2 Aim

The main aim of the thesis is to correlate and calibrate the finite element model of shake rig with the physical shake rig using frequency response functions obtained from experimental modal analysis.

1.3 Problem Description

The study involves development of a finite element model of the chassis shake rig for modal and transient dynamic simulation. Modal analysis will be performed on

finite element model of shake rig to synthesize frequency response. This frequency response will be compared with the frequency response obtained from experimental modal analysis (EMA). The resonant frequencies and mode shapes of the finite element model are to be correlated with the physical rig. The finite element model will be parametrized and calibrated to get a better correlation.

1.4 Method

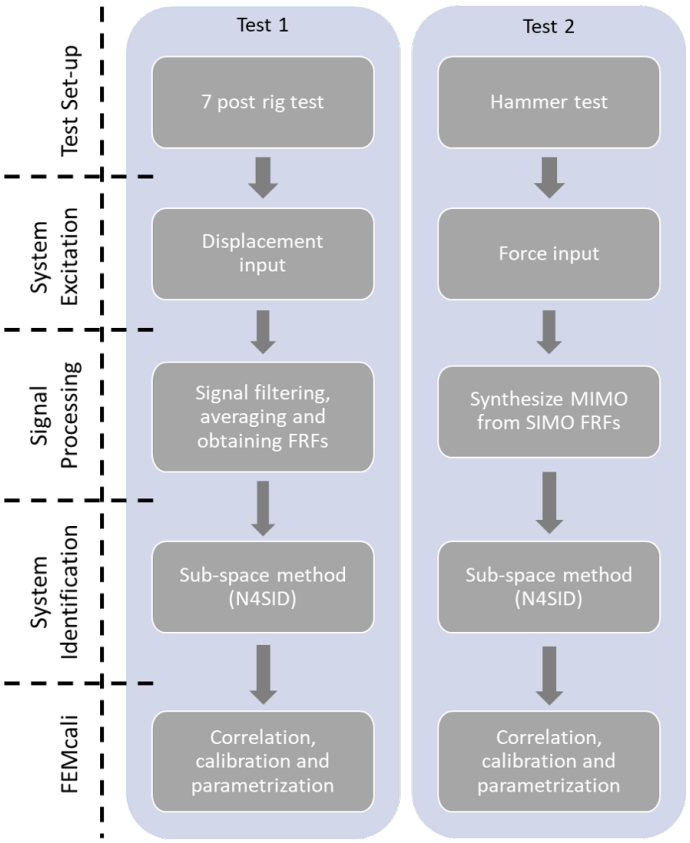

Figure 1.1: Overview

This section briefly describes the various steps that need to be followed to get the intended results. Figure 1.1 gives an overview of the study conducted. Two test setups will be evaluated. The finite element model for both the tests will be developed using ANSA and CVM. Experimental modal analysis (EMA) will be performed on the physical rig. Signal processing will be done to experimental data. Frequency response functions will be synthesized from the accelerance ($\frac{acceleration}{force}$) data of the EMA. Post-screening of the measured experimental data will be done to cure test data of any errors in estimation. The system will be identified using state-space method. The accuracy of the state-space model will be improved by developing a MATLAB script that captures the relevant eigenmodes of the identified system.

Correlation studies between the finite element model and the physical rig will be

performed by using MSC NASTRAN and MATLAB. First, a MATLAB script will be developed to do initial correlation and check for faulty channels. This script will also be used to plot eigenvectors and the mode shapes (See Appendix A). Then, a framework will be developed to submit NASTRAN runs from MATLAB (Windows environment) to the LINUX server of Volvo (See Appendix B). FEMCali will be used to submit NASTRAN runs and do eigenvalue calculation from the mass and stiffness matrices obtained from NASTRAN runs. Based on the experimental modal analysis results, the dynamic response of the finite element model will be correlated. A parameter study will be done to identify the parameters influencing the modal behavior of the rig. The parameters will be modified using FEMCali to obtain a calibrated finite element model of the rig. This section is divided into the following sub-sections which give a framework of work-flow for the research that will be done in this study.

1.4.1 Prestudy

In this phase all the engineering reports and relevant literature will be reviewed. Effort will also be towards getting hands-on practice of all the relevant tools needed in the study. The finite element model of the test rig will be setup in NASTRAN environment for static and modal simulation. Two load cases (prescribed displacement and applied force) will be evaluated for static and transient dynamic simulations. This will be done to understand how NASTRAN-deck works. The functionality of different property cards and material cards used in these simulations will be studied. Thesis planning and methodology for the thesis will be done in this phase.

1.4.2 Finite Element Modelling and Simulation

The virtual rig will be modelled using a combination of tools (CVM and ANSA). The main aim will be to setup a calculation model in NASTRAN environment that best represents the actual rig. The deck setup will be done for modal analysis (SOL103). The results obtained from the modal analysis will be used in correlation and calibration.

1.4.3 Experimental Modal Analysis

Experimental modal analysis is done to synthesize transfer functions. The transfer functions will be obtained using two ways. First, experimental data from hammer test (see section 4.2), which was conducted earlier, will be used to get transfer functions. Second, an experimental modal analysis of the frame structure will be carried out on 7 post durability shake rig. The system will be excited using periodic random excitation (white noise). Force transducers will be introduced in the rig for measuring force input. Response of the system in the form of acceleration from the accelerometers will be measured on the frame. We will also perform an experimental modal analysis of the frame with a battery box mounted on it. The accelerance data from this phase will be used to synthesize transfer functions in MATLAB (using *tfestimate()*).

1.4.4 Signal Processing

Signal processing will be done to experimental data. The raw test data from the physical rig is in the form accelerance data. From the hammer test data, Multi Input Multi Output(MIMO) transfer function will be synthesized using Single Input Multi Output(SIMO) transfer functions. These SIMOs will be obtained from accelerance data of hammer test.
Transfer functions will also be synthesized from the accelerance data of the EMA of the shake rig using MATLAB. Before that, the accelerance data will be analyzed. Several methods like windowing of time domain data, overlapping, mean of data, detrend data and/or using band-pass filters will be used to improve quality of data. Post-screening of the measured experimental data will be done after transfer function estimation. Estimation of FFT of signals, power spectral density of signals (P_{xx}) and singular value decomposition (SVD) will be done to investigate errors in estimation.

1.4.5 System Identification

The system identification will be done using using System Identification Toolbox in MATLAB. The quality of the state-space model will be improved by developing a MATLAB script. A script will be used to remove residual effects of modes outside the test frequency band. Then, the script will divide the frequency domain based on the location of eigenmodes on the frequency domain. The estimated state-space model (SSX) will be improved by doing more iterations [5].

1.4.6 Correlation

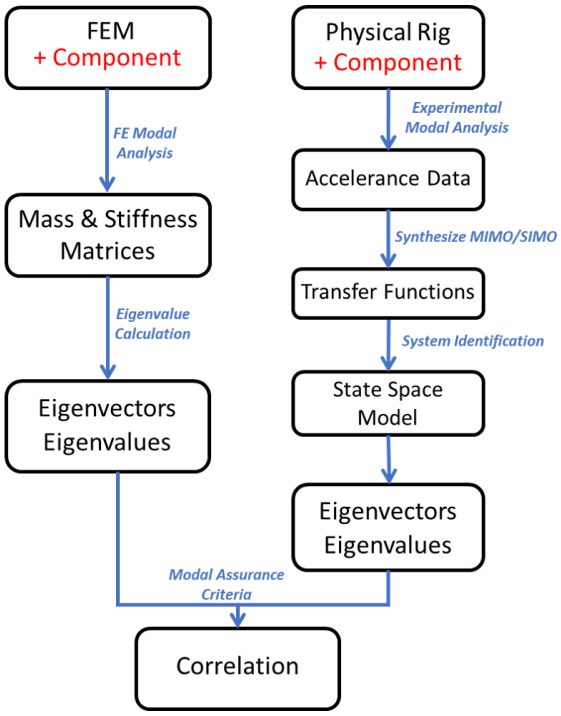

Figure 1.2: Correlation Process

A MATLAB script will be developed to do initial correlation by plotting experimental eigenvector elements against their analytical counterparts. Figure 1.2 shows the workflow used in the script. Modal assurance criteria will be used to do correlation. A framework will be developed to submit NASTRAN runs from MATLAB to the LINUX server of Volvo GTT. FEMCali will be used to submit NASTRAN runs and do eigenvalue calculation from the mass and stiffness matrices obtained from NASTRAN runs. The FEMCali code will be modified for our study.

1.4.7 Calibration

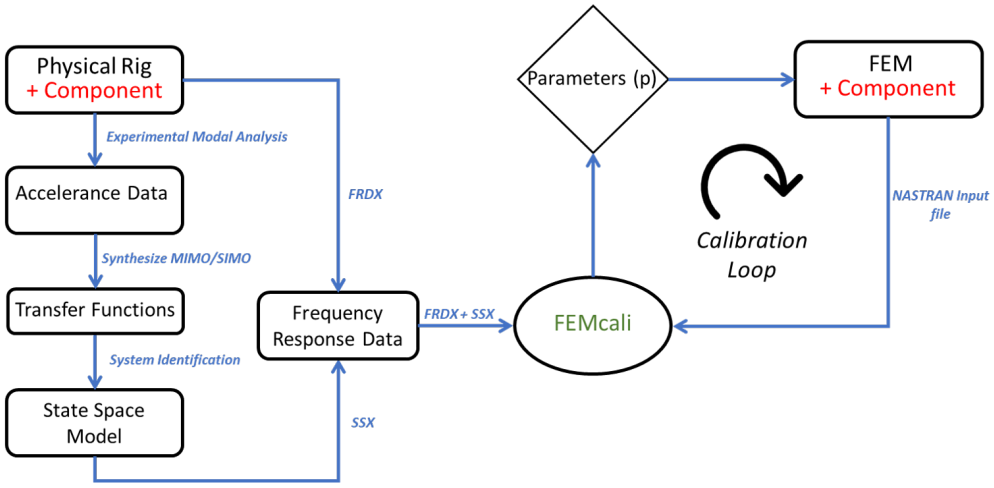

Figure 1.3: Calibration Process

Calibration of the FE model will be done based on the correlation output. Certain parameters will be taken (like thicknesses of the frame parts) and will be given as an input to the FEMCali application in MATLAB in the form of a NASTRAN input file. The second input to FEMCali will consist of the response data of the physical rig ($FRDX$) and the equivalent state-space model of the rig (SSX). This process will be a recurring one as it will involve iterations of the parameters required for calibration. The output of this phase will be a calibrated FE model. Figure 1.3 shows a flowchart of the calibration process. The calibration loop in FEMCali is described in Section 2.6.9 and Section 2.6.10. In Chapter 7 see Figure 7.1 for detailed understanding of how FEMCali works.

1.5 Limitations

Time

The study is limited to a short period. The goals are to develop a calculation model of the test rig, perform experimental modal analysis , correlate and parametrize the model and to test chassis components on the virtual rig. Depending on the progress of the tasks, the scope of the study is adjusted.

Availability of test rig

The shake rig in the durability lab of Volvo GTT at Lundby has continuous tests being carried out on the chassis and the cabs of the trucks. One part of the study involves carrying out measurements in the rig and will depend on the availability of the rig.

MATLAB-NASTRAN Framework

Since, FEMCali has never been used before at Volvo GTT before, a new framework is developed so that NASTRAN jobs can be submitted to the server from MATLAB in Windows environment. It would take time to analyze the present Volvo GTT system in place and modify FEMCali script to implement the new framework (See Appendix B).

Licences and Job Queues

The licences for System Identification Toolbox, Statistics Toolbox of MATLAB are limited. So, availability of these licences is a bottle-neck for correlation and calibration process. The NASTRAN jobs are submitted from FEMCali to the server at Volvo GTT. Since, multiple NASTRAN runs are submitted during the calibration process, availability of NASTRAN licences is also a bottle-neck.

2
Theory

In this chapter we focus on the basic theory of structural dynamics, system excitation, signal processing, system identification, finite element model parametrization, correlation and calibration.

2.1 Structural Dynamics

Linear-Viscous System

The following second order differential equation is used to represent linear viscous damped system.

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{V}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{Q}(t) \tag{2.1}$$

$\boldsymbol{M}$ is the mass matrix, $\boldsymbol{K}$ is the stiffness matrix and $\boldsymbol{V}$ is the viscous damping matrix. $\boldsymbol{q}$ represents the nodal displacement matrix of the system [1][2].

$$\boldsymbol{Q} = \boldsymbol{P}_u\boldsymbol{u} \tag{2.2}$$

$\boldsymbol{Q}$ can be represented as in equation 2.2. $\boldsymbol{Q}$ is the loading vector. $\boldsymbol{u}$ is the force stimuli. $\boldsymbol{P}_u$ is Boolean transformation matrix. The finite element model is parametrized using data vector $\boldsymbol{P}$. The data vector $\boldsymbol{P}$ contains all the parameters described in Section 7.1 in Chapter 7. The mass and stiffness properties of the finite element model is dependent on these parameters [1].

$$\boldsymbol{K} = \boldsymbol{K}(\boldsymbol{P}) \tag{2.3}$$

$$\boldsymbol{M} = \boldsymbol{M}(\boldsymbol{P}) \tag{2.4}$$

The parameter setting $\boldsymbol{P}$ is dependent on the nominal parameter setting $\boldsymbol{P_0}$ and normalized parameters $\boldsymbol{p}$. The normalized parameter is zero when nominal parameter setting is used [1].

$$\boldsymbol{P} = \boldsymbol{P}_0(1 + \boldsymbol{p}) \tag{2.5}$$

The mass and stiffness matrices are linear in some parameters like Young's Modulus. They are non-linear in some parameters like shell thickness. A Taylor series expansion linearization is used around the nominal values of the parameter to get a good approximation. The first-order differential equations (see equation 2.6 and equation 2.7) analogous to equation 2.1 can be formed as state-space equations [1] [10].

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \mathbf{B}\boldsymbol{u} \tag{2.6}$$

The output $\boldsymbol{y}$ is a linear combination of the state variable $\boldsymbol{x}$ and force stimuli $\boldsymbol{u}$.

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} \tag{2.7}$$

Where $\boldsymbol{x}$ is the state vector (see equation 2.8):

$$\boldsymbol{x} = \{\boldsymbol{q}; \dot{\boldsymbol{q}}\} \tag{2.8}$$

The state-space matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are related to mass, stiffness and damping matrices as follows :

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ -\boldsymbol{M^{-1}V} & -\boldsymbol{M^{-1}K} \end{bmatrix} \tag{2.9}$$

$$\boldsymbol{B} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{M^{-1}P_u} \end{bmatrix} \tag{2.10}$$

The harmonic force input $\boldsymbol{u}$ and harmonic output $\boldsymbol{y}$ can be seen in equation 2.12 and equation 2.11 respectively [10].

$$\boldsymbol{y} = \tilde{\mathbf{y}} exp(i\omega t) \tag{2.11}$$

$$\boldsymbol{u} = \tilde{\mathbf{u}} exp(i\omega t) \tag{2.12}$$

The relationship between the force input and output in the frequency domain is as follows [10]:

$$\tilde{\mathbf{y}} = \left[\boldsymbol{C}\left(i\omega\boldsymbol{I} - \boldsymbol{A}\right)^{-1}\boldsymbol{B} + \boldsymbol{D}\right]\tilde{u} \tag{2.13}$$

So, the transfer function for the dynamic system becomes:

$$\boldsymbol{H}(\boldsymbol{p}) = \boldsymbol{C}\left(\boldsymbol{p}\right)\left(i\omega\mathbf{I} - \boldsymbol{A}\left(\boldsymbol{p}\right)\right)^{-1}\boldsymbol{B}\left(\boldsymbol{p}\right) + D\left(\boldsymbol{p}\right) \tag{2.14}$$

The transfer function is estimated from experimental test data ($\boldsymbol{H^X}$) and FEM data ($\boldsymbol{H^A}$). During model calibration the oracle value of parameter ($\boldsymbol{p} = \boldsymbol{p}^*$) is found out such that there is least deviation between $\boldsymbol{H^X}$ and $\boldsymbol{H^A}$(p) [10].
A quadratic functional is used as the deviation metric:

$$\delta = \boldsymbol{\epsilon}^H\boldsymbol{\epsilon} \tag{2.15}$$

$\boldsymbol{\epsilon}$ is the deviation vector such that:

$$\boldsymbol{\epsilon}(\boldsymbol{p}) = \log_{10} vect(\boldsymbol{H}^A(\boldsymbol{p})) - \log_{10} vect(\boldsymbol{H}^X) \tag{2.16}$$

In order to make the calibration process less computationally heavy, a sampling strategy is used for discrete frequencies such that:

$$\triangle\omega_i = 2\zeta_i\omega_i \tag{2.17}$$

where ζ_i is the relative modal damping for the *ith* mode and $\triangle\omega_i$ is the half-bandwidth.

2.2 Experimental Testing

The quality of the experimental test data is important for good correlation and calibration. So, the choice of system excitation, testing hardware (force transducers and accelerometers) is very crucial to obtain good measurements. This section describes the important tools and system excitation methods used for testing.

2.2.1 Testing Software

There are various software programs used as platforms for data acquisition and management of test for durability, noise and vibration tests. Two software programs will be used for the two tests carried out during the thesis.

RPC Pro

This software package is used for the rig test which is carried out during the thesis. It is an advanced road simulation evaluation and signal processing tool developed by MTS System Corporation for conducting durability lab tests.It replicates vehicle response measured on test track.

The *RPC Pro* contains two modules *MODEL Pro* and *SIMULATE Pro*. *MODEL Pro* is used to drive rig using noise signals. *SIMULATE Pro* works on the output frequency response functions. It inverses the frequency response functions and multiplies them with the target response to get first drive signal estimate. It is a iterative process and is carried out till the difference between the actual and target response is minimized.

In the rig test, the data is be acquired using 21 channels. This data includes displacement inputs (7 channels) from actuators, force outputs (7 channels) from force transducers and acceleration outputs (7 channels) from accelerometers. The software will be used to give a white noise signal input and a random periodic amplitude modulation signal input.

LMS Test

The LMS test is a tool from the Siemens Simcenter testing solutions package. This tool is also used for durability and noise and vibration lab tests. The data obtained from the test can be processed and filtered in this tool as well.

The hammer test was carried out using this software, where the impact from the hammer on the frame was calculated as the force input and the acceleration output was recorded. The input force and the output acceleration data was processed in the software to estimate an accelerance transfer function.

2.2.2 Testing Hardware

The set up of a vibration test needs several hardware items. One of the primary hardware is a sensor which converts the system response into suitable electrical signals. The output signal from the sensor is sampled by the data acquisition system (DAQ) to a digital form in order to process it further. The modern multi channel DAQ may also need a signal conditioning device along with the sensors to match the voltage of the sensors with the DAQ. The DAQ also comprises of an analog signal filter. The filtering is done before sampling the signal. The signal above the desired frequency range is considered to be unwanted, spurious noise and eliminated using this high frequency stop band filter.

Sensors that are usually used for sensing response in a vibration test are accelerometers. However, strain gauges can also be used for sensing the response. Refer Abrahamsson, 2012 for more detailed explanation on the testing hardware.

2.2.3 Accelerometers

An accelerometer is a transducer that measures instantaneous acceleration of the location at which it is mounted, by converting kinetic energy to electrical signal. The accelerometers that will be used for the tests are based on piezoelectric effect. Piezoelectric materials accumulate electric charge when subjected to mechanical stress/strain. In a piezoelectric accelerometer, an inertial mass is mounted on a piezoelectric crystal. The inertial mass produces a counter-acting force on the piezoelectric crystal when acceleration is sensed by the transducer. This force produced by the inertial mass generates a proportional strain which results in an electric charge output.

There are two types of accelerometers: voltage mode and charge mode. Voltage mode acccelerometer is used for the tests carried out. A voltage mode accelerometer consists of internal electronics for signal conditioning and for converting the charge produced in the piezoelectric material to voltage. This type of accelerometers are good for noisy environments since the output is a low impedance signal, it can be transmitted easily using long cables and therefore good to be used in rig tests.

The technique used for mounting the accelerometers plays an important part in the accuracy of the response captured. The motion sensed by the accelerometer's piezoelectric material and the motion of the test object differ specially at higher frequencies. Two factors for proper mounting are proper surface contact and proper alignment of the local coordinate system of the accelerometer with the global coordinate system of the test object. Good surface preparation at the place where the accelerometers are to be mounted is achieved by making the surface flat and smooth. Silicon grease can also be used as it allows for a better surface contact. The fundamental eigen frequency is the critical frequency of the accelerometer. The frequency range of the test data should not be anywhere close this frequency as it may affect the results. Refer Abrahamsson, 2012 for further information on accelerometers.

2.2.4 Force Transducers

Force transducer is a type of sensor that measures the force by converting the strain produced due to the load in to electrical signal. Strain gauge based force transducers were used for measuring the forces in the rig tests. These type of force transducers have a spring element to which the load is applied. The spring deforms due to the force, thereby converting the force into strain.

The main sensing part in the force transducer is the strain gauge. This gauge comprises of holders (insulation layers) attached with a measuring grid. These gauges are attached to the spring elements. A typical strain gauge force transducer consists of four strain gauges placed in a way such that, when force is applied to the spring, two of the strain gauges are stretched and the other two are compressed. The gauges are connected to each other in a Wheatstone bridge circuit. There exists a difference in the voltage due to change in the resistance. This change in the resistance is caused due to the strain produced. The resulting electrical signal therefore depends on the applied force.Refer Abrahamsson, 2012 for further information.

2.2.5 System Excitation and Boundary Conditions

One of the important steps in testing is preparing the test setup. It is challenging to make the test object behave and respond dynamically on the test setup the same way it would during operational use. This is achieved by making sure the boundary conditions and system excitation mechanisms are chosen correctly. This also helps in removing unwanted dynamic effects from the surrounding and isolates the test object which helps in achieving better results from the test.

The input signal also needs to be modified to obtain a desired output signal. The excitation signal is decided based on a number of factors like the characteristics of the test object (system), the data usage and on the modal parameter estimation. A more detailed explanation about these factors is discussed in Phillips and Allemang, 2003.

Free Support Boundary Condition

A free support boundary condition means that the object is freely suspended in space and is not connected to the ground. Such a freely suspended structure will have rigid body modes (RB modes). Any structure in theory must have 6 RB modes having a natural frequency of 0 Hz each. These RB modes are defined by the mass and inertia properties which is an important data in vibration testing. This type of boundary condition helps to identify the RB modes of the structure.

A freely suspended state is not possible in reality. This can be achieved by supporting the object (structure) using spring like elements, example: bungee cords. Another way of achieving the free support condition is by mounting the object on a soft suspension like an air bellow or on fully deflated tires or special purpose gas

filled airbags.

This support system changes the frequencies of the RB modes. The natural frequencies of the RB modes which ideally need to be 0 Hz are close to 0 Hz but not zero. The highest RB mode frequency must be very low compared to the frequency values of the elastic modes. This can be ensured by attaching the suspension system at the nodal points.

Periodic Random Excitation

There are different excitation methods used in vibration tests. Depending on the time history the excitation signal can be divided as periodic and aperiodic. The periodic random excitation signal is most commonly used in vibration tests as it gives accurate and better results which are needed for model calibration. White noise or a colour frequency obtained through band pass filter are commonly used signals.

In periodic excitation the signal repeats after a fixed time interval called a cycle. The frequency spectrum of this type of excitation is characterized by random distribution of phase and random amplitude. The initial transient response of the system (caused due to the excitation signal) dies out after certain number of cycles making it a steady-state response having a good period repeatability.

Impact Excitation

An impact test like hammer test causes aperiodic excitation which is also a commonly used excitation method. Aperiodic excitation methods require less set-up work and are easy to carry out.

For impact tests, the hammer (impactor) has a force transducer or a force gauge attached to it to identify the force magnitude. The magnitude depends on the velocity of the impact and the mass of the hammer head. The mass of the hammer head along with the stiffness of the structure affects the structure's frequency range of excitation. In the time domain, the signal is in the form of a half sine shaped pulse [8]. See Ewins, 2000 for more information on hammer tests.

This excitation signal was used in the hammer test carried out at Vibration Testing Lab, Volvo GTT and more about it is discussed in Section 4.2.

2.3 Signal Processing

In vibration test, the signal processing part majorly involves conversion of time series data to the frequency domain using Fourier transformation. This depends on the input signal and the response of the object to the input excitation. There are however other factors that might arise while carrying out Fourier analysis which affect the results. Limiting the time history length and discretization can modify the data

and thus affect the result. This section discusses about the various terminologies in signal processing part of the thesis.

2.3.1 Band-Pass Filter

Digital band-pass filter performs mathematical operations on input signals such that it passes frequencies within a certain range and attenuates frequencies out of the given range. In this study, a digital band-pass filter is used to allow signals within a selected range of frequencies to be decoded, while preventing signals at unwanted frequencies (noise) from getting through. A band-pass filter also optimizes the signal to noise ratio [8]. Thus, it improves quality of signals and helps in better numerical estimation.

The digital band-pass filter is implemented by using MATLAB functions *designfilt()* and *filtfilt()*. They are used to design the digital band-pass filter and apply it on the accelerance data respectively.

2.3.2 Transfer-function estimation

System identification uses information of the spectra of test stimuli and the response of the system to this stimuli to estimate transfer functions. The output obtained from the sensors (accelerometers) is in the time domain. The useful information or the spectral properties however are in the frequency domain. This makes it crucial to convert this signal from time domain to the frequency domain. This conversion is done by Fourier transformation.

The equivalent Fourier series or Fourier transform of the time series data is obtained with the help of approximations produced and used by Discrete Fourier Transforms (DFT). The Fourier transform is discrete for a periodic signal. For a periodic time signal, the Fourier transform is a series of impulses. Due to the fact that the Fourier series is basically a series of exponential functions and when this exponential series is transformed using Fourier transforms, it results in a series of impulses in the frequency domain.

The time period of the time signal is divided into N samples. The maximum frequency corresponding to the maximum angular frequency is called the Nyquist frequency and it is a function of 2N samples. All these values help in obtaining the Fourier coefficients which are used to obtained the transfer function estimate.See Section 5.1.2 for more information on transfer function estimation in this study. Fast Fourier Transform developed by Cooley and Tukey in 1960 a widely used algorithm for the transfer function estimation. A detailed explanation of this procedure and all the equations used in the transfer function estimation are given in Abrahamsson, 2012.

2.3.3 Fast Fourier Transform

Fast fourier transform (FFT) is an algorithm that computes the discrete Fourier transform (DFT) of a signal. Fourier analysis converts a signal from time domain to frequency domain and vice versa. The DFT is obtained by decomposing a sequence of values into components of different frequencies. The transfer function estimate of the transfer path from the input u to the response y is determined from the estimated Fourier coefficients [8].

2.3.4 Singular Value Decomposition

Singular value decomposition expresses a matrix A (of order m x n) in terms of U, V' and S [12]. S is an m x n diagonal matrix with singular values of A on its diagonal. The columns of matrix U (of order m x m) are the left singular vectors for corresponding singular values. The columns of matrix V (of order n x n) are the right singular vectors for corresponding singular values.

In this study, MATLAB function *svd()* is used to compute the singular value decomposition of a matrix. This function calculates only the singular values of a matrix or both singular values and singular vectors in one function call [12]. Singular value decomposition is used in this study during signal processing to verify that the force inputs are linearly independent.

2.3.5 Power Spectral Density

In this study, power spectral density (PSD) (the measure of a signal's power content against a give frequency range) is used to evaluate the force signals and acceleration signals. PSD is used to illustrate spectral energy distribution [1]. MATLAB function *periodgram()* is used to get PSD of acceleration and force signals.

2.4 System Identification

The method of expressing a physical, dynamic system as a mathematical model by measuring input and output signal of the system is called system identification. It is a method used to identify a system in the form of a mathematical model.

Most of the currently used system identification methods are non iterative and are based on numerical linear algebra. Choosing a correct model order for the system is an important factor for these non iterative methods. One such method is the N4SID algorithm. The system identification toolbox in MATLAB uses the N4SID algorithm to build the state space model. More about the N4SID algorithm can be found in Abrahamsson, 2012.

2.4.1 State Space Model

State space model is a form of expressing the governing differential equations of the system we are studying (see equation 2.6 and equation 2.7). The state space form is very convenient to use for model validation purposes [1] [2].

2.5 Finite Element Model

The finite element model of the rig is modelled based on number of cross-members, type of cross-members, relative position of cross-members and position of components(e.g. battery box) among other factors. The finite element model for hammer test and rig test (periodic random excitation) are different. The connections between the various components of the model are done in order to make the model as close to reality as possible. CVM and ANSA are used to model the rig. CVM creates default connections between different components based on the output settings. The validity of these connections are verified and modified to get a model with a dynamic response similar to physical rig. The NASTRAN-deck is set up to get the desired output files in the prescribed format (mass and stiffness matrices).

2.5.1 Type of Elements

RBE2 and RBE3 are two common types of Rigid Body Elements (RBEs) that are used in finite element modelling. They are multi point constraint (MPC) elements. Both RBE2s and RBE3s are often used to connect one node to several nodes.

RBE2 Element

RBE2 element is a rigid body with independent degrees of freedom specified at one node (master node) and with dependent degrees of freedom at an arbitrary number of nodes (slave nodes). It is used to model the rigid spiders for bolts as can be seen in Figure 2.1.

RBE3 Element

RBE3 element defines the motion at a reference node as the weighted average of the motions at a set of other nodes. This is used to model bolts, shown in Figure 2.1. See NASTRAN Manual for further explanations. RBE2 connections are generally stiffer than RBE3 connections, because all the slave nodes have the same motion as the master node. In RBE3 connections, the motion of the reference node is defined by all the other connected nodes [16].

RBAR Element

RBAR defines a rigid bar with six degrees of freedom at each end [16]. This allows the user to constrain the degrees of freedom that would better represent the physics of the problem. RBAR elements are created by default in CVM if the nodes of the

parts are aligned.

CBAR Element

CBAR is a simple beam element connection. In this thesis, CBAR is used to define simple beam element between two RBE2 spider to represent a bolt. The beam properties are defined based on previous calculation models. These properties can be parametrized while doing model calibration [16].

Shell Elements

The finite element model of the rig has mostly components that are modelled using shell element. CQUAD4 and CTRIA3 are the shell elements that are used. CQUAD4 is a quadrilateral plate element connection. CTRIA3 is a triangular element connection. Both define a plane strain plate element that can be used for components that have small thicknesses.

Solid Elements

Solid elements are used to model some brackets connecting cross-members with the side rail. CHEXA, CPENTA and CTETRA are solid elements that are used. CHEXA defines the connections of a six-sided isoparametric solid element with eight to twenty grid points [16]. CPENTA defines the connections of a five-sided isoparametric solid element with six to fifteen grid points [16]. CTETRA defines the connections of a four-sided isoparametric solid element with four to ten grid points [16].

2.5.2 Boundary Conditions

Every dynamic system has stiffness and mass properties. Mass introduces inertia due to kinetic energy and stiffness comes from potential energy, which is a function of boundary condition. Different boundary condition introduces different reactions at the support, accordingly the stiffness of the vibrating system changes, mass being the same. So it is necessary to have a calculation model that has the correct degrees of freedom.

The NASTRAN file is set-up in ANSA. SOL103 solver type is used in this thesis while doing correlation and calibration. In SOL103, there is no need to constrain the model using Single Point Constraints (SPCs) or enforced displacements. In SOL112 and SOL109 we need to define single point contraints, force input or enforced displacement to get a contrained calculation model.

2.5.3 Modelling of Bolts

Bolts are modelled using RBE2 elements and beam element. A RBE2 spider is modelled at connection location (hole) and the spiders are connected using CBAR

element (see Figure 2.1). CBAR elements represent the beam. The beam properties are defined based on assumptions and prior use in other models.

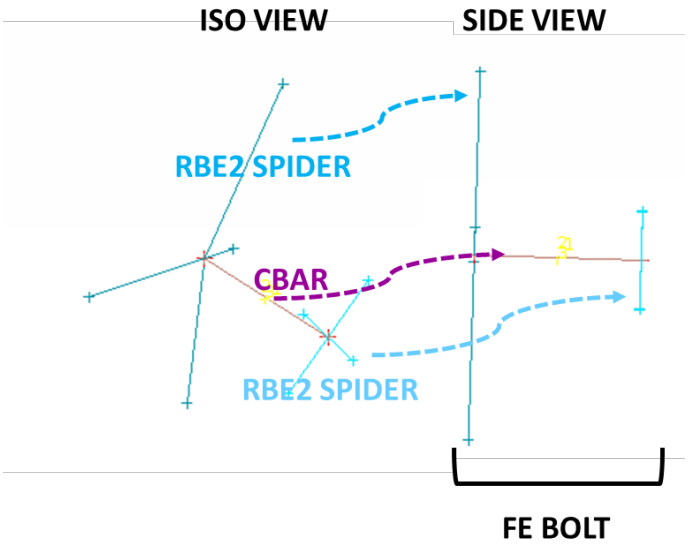

Figure 2.1: Bolt Connection in the FE Model

2.5.4 Cross Member Connections

The cross-members are connected to the side rail and the inner liner in CVM tool. These connections are based on the relative location of the the nodes on the cross-members and that on the side rail/inner liner. CVM creates RBAR connections when the connection locations (nodes) are aligned (i.e. they are co-linear on the YZ-plane and XY-plane). It creates RBE2 spiders and CBARs when the connection locations are not aligned.

2.6 Correlation and Calibration

This section contains the essential definitions related to correlation and calibration that would help in understanding the process.

2.6.1 Model Verification

Model verification is an assessment of computational model wherein, the model and the code implementing it are checked for numerical errors such that they accurately represent the underlying mathematical model and its solution [1].

2.6.2 Model Validation

Model Validation is the process that corroborates that a computational model in its domain of applicability, possesses a satisfactory range of accurate consistency with

the intended use of the model.The credibility of the model is assessed by using a deviation metric prescribed by a suitable validation criteria [1].

2.6.3 Model Calibration

Model Calibration is a process in which the parameters of a parametrized calculation model are iterated such that the results from simulation closely represent the experimental results.

2.6.4 Correlation Indicators

A wide range of indicators can be used to compare the modal behaviour of the calculation model. The eigenvalues of the finite element model and resonant frequencies of the physical rig can be compared and used as a correlation indicator [11]. The eigenmodes of the finite element model and the physical rig can be plotted and compared. This strategy helps in realizing the mode shapes of the flexible modes. When experimental eigenvectors are plotted against their analytical counterparts, they should ideally be lined up along a straight line in the plot for a good correlation. Another, non-graphical, correlation measure is to calculate the angle between the experimental and analytical eigenvectors.

Three commonly used methods for model correlation using modal parameters are Modal Assurance Criterion (MAC), Percent frequency difference and Pseudo orthogonality.

Modal Assurance Criterion

The Modal Assurance Criterion (MAC) is a statistical indicator that measures the degree of consistency between mode shapes [17]. It is used to pair modes shapes derived from finite element model with those obtained experimentally. The MAC value varies from zero to one, with one indicating fully consistent mode shapes and a value close to zero suggests that the mode shapes are inconsistent. MAC is most sensitive to large differences and relatively insensitive to small variations in the mode shapes [17]. It is to be noted that MAC takes into consideration only modal shapes which mean that a separate frequency comparison must be used in conjunction with the MAC values to determine the correlated mode pairs [17]. See Section 8.2.3 for results obtained in this study using MAC.

$$MAC = \frac{|\{\psi_{tr}\}^H \{\psi_{vr}\}|^2}{(\{\psi_{tr}\}^H \{\psi_{tr}\})(\{\psi_{vr}\}^H \{\psi_{tr}\})} \tag{2.18}$$

Percent Frequency Difference

In Percent frequency difference method we compute the error percentage between analytical natural frequency and experimental natural frequency.

$$\sum_n frequency = \frac{f_{tr} - f_{ar}}{f_{ar}} \times 100\% \tag{2.19}$$

Pseudo-orthogonality

Modal pseudo-orthogonality is a check in which the mode shape matrices from calculation model and experimental model that represent the same dynamic system. This is accomplished through a matrix computation, which maps the systems mass or stiffness matrices into modal space. This is useful because the modal mass and stiffness matrices are theoretically known.

Most often the systems mass matrix is used because it will ideally result in an identity matrix (depending on modal vector scaling), making the results easier to analyze.

$$[\Phi_a]^T[M][\Phi_t] = [M_{modal}] \approx [I] \tag{2.20}$$

2.6.5 Model Reduction

The finite element model is reduced by using model reduction techniques [3]. See Craig and Kurdila, 2006 for further details. Model reduction helps in reducing the degree of freedom of the system. A reduced model is less computationally heavy without compromising on the accuracy of results. During the calibration process in FEMCali model reduction helps in reducing the computation time, so more iterations can be run in the same time subject to other factors.

2.6.6 Modal Decomposition

Modal decomposition expresses the state-space equation as a linear combination of the various modes of the system. Since, there is transformation to the modal domain, we get diagonal state-space matrices. This reduced model contains all relevant information regarding the model [1].

2.6.7 Fisher Information Matrix

Fisher Information Matrix is the inverse of the variable covariance matrix [1], which measures the amount of information that an observable random variable carries about an unknown parameter $\boldsymbol{p}$. The determinant of Fischer Information Matrix is used to scale the contribution of the variables with respect to the dynamics of the system [1].

2.6.8 Deviation Metric

A frequency domain deviation metric is used in this study, see equation 2.16. It is the log of the difference of transfer function from finite element model and transfer function from experiment.

2.6.9 FEMCali

FEMCali is a finite element model calibration MATLAB tool. It is a finite element calibration procedure that uses frequency response data from test and damping equalization [10]. In this study, the calibration of the finite element model of durability rig would be done using this tool.

FEMCali uses gradient based minimization technique and frequency response function based deviation metric to do calibration [1] [10]. The calibration process is a nonlinear minimization procedure where the deviation metric is nonlinear in the given parameters [10]. The Levenberg-Marquardt algorithm is used to perform such minimisation. See Abrahamsson, 2012 for further explanation.

The damping values found in the experimental testing can be applied to the calculation model. However, mapping these experimental modes to the numerical model is cumbersome and difficult [10]. So damping equalization method is used to overcome this problem. In this method equal damping is assigned for all the modes, to avoid mode paring issues [10]. See Abrahamsson and Kammer, 2015 for further discussions on this topic. A numerical value of modal damping is assigned to both the finite element model and the analytical model obtained after experimentation. Then, calibration is performed on the finite element model towards the experimental model [10].

The calibration process is iterative in nature, in which calibration is done with multiple start values for the parameters till an optimal value is found. The finite element model is reduced to a surrogate model to decrease computation time. The same surrogate model is used during the whole calibration process in which parameter setting is modified for every iteration. Finite difference approximation method is used to calculate gradients of the structural matrices of the finite element model. The first term of the Taylor series expansion of the reduced-order model is used as the surrogate model [10]. The obtained surrogate model is an accurate representation of the finite element model where the reduced-order model and its gradients are established from the original finite element model. The complete finite element model is not evaluated later. This decreases the computational effort.

2.6.10 Work-flow in FEMCali

FEMCali takes input arguments like the calculation model file (∗.dat), frequency response data and corresponding identified state space model (∗.mat). We also need to provide the degree of freedom number at the sensor locations and force input (actuator) locations along with their direction. The frequency band that we are studying is also provided. During the calibration process, the mass and stiffness matrices are obtained from the finite element model [10]. The surrogate model is obtained by estimating the modal damping matrix [11] according to the damping equalization method mentioned in Section 2.6.9. The response of the dynamic system to infinitesimal variations in parameters is obtained by extracting corresponding structural matrices [10]. These structural matrices are extracted from the finite ele-

ment model where the parameters are varied infinitesimally. Damping equalization is used to modify the state-space model obtained from experimental data [11].
Then, FEMCali compares the surrogate model and the modified state space model. Lavenberg-Marquardt algorithm is used for the above purpose. It searches for the minimum deviation metric. The calibration procedure uses randomised start values for parameters to obtain the final calibrated parameter values [10]. The minimum deviation metric obtained while comparing the numerical model to the experimental model gives the final calibration solution [11]. FEMCali gives the calibrated parameters and various graphical results of the calibration process as output. The calibration loop is represented graphically in Figure 7.1 in Chapter 7.

3

Finite Element Model

This chapter gives information about the methodology followed while modelling finite element models of the experimental setup. It describes the commands and tools that were used for finite element modelling. The finite element models were developed using ANSA. The finite element components were assembled using CVM (See Section 3.2).

3.1 Finite Element Model

Finite Element Model Simplifications

While modelling the finite element models certain simplifications were used to make the problem less computationally demanding while not compromising on the physics of the problem. Figure 3.1 illustrates how rigid elements have been used to represent the actuators without actually modelling the actuators.

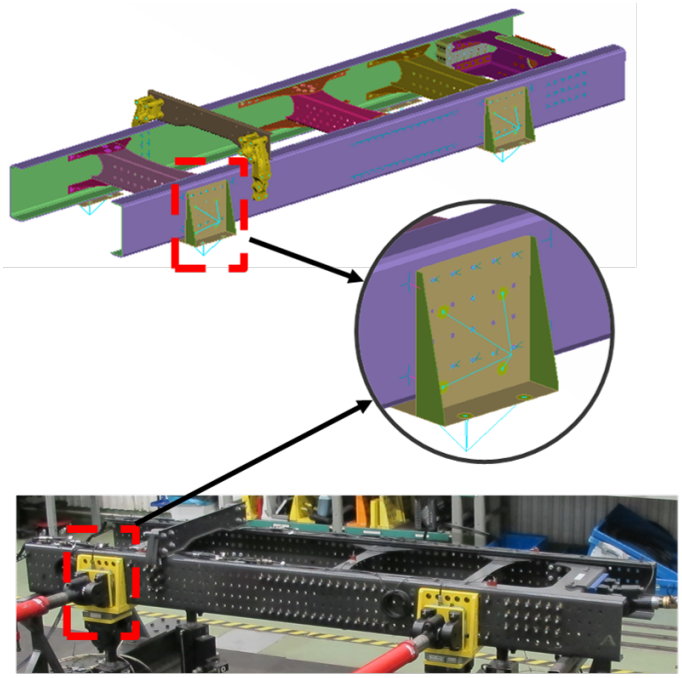

Figure 3.1: FE Model simplification

Finite Element Model Mesh

The finite element models of the rig uses shell elements (CQUAD4, CTRIA3) for components like side rail, inner liner and cross-members and solid elements (CHEXA, CPENTA, CTETRA) for some brackets connecting the cross-members to side rail (see Figure 3.2). See Section 2.5 in Chapter 2 for more details.

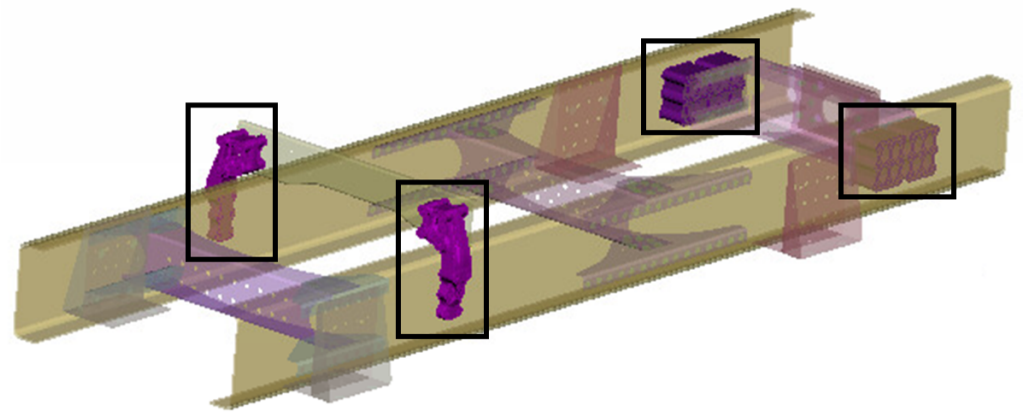

Figure 3.2: Solid Element Parts in the FE Model

In the finite element model CBAR, RBAR, RBE2 and RBE3 elements are used for making connections. CBAR and RBAR can be used interchangeably depending on the usage. CBAR elements have more parameters like cross section area and Young's modulus which can be parametrized in calibration process (See Section 7.1) and give us more control over the modal behaviour. RBAR can also be used for such connections but without any parameters. RBE2 and RBE3 are used to model rigid connections, especially while modelling bolts (see Section 2.5 and Figure 2.1 for more details). RBE2 connections equally distribute the force to dependent nodes which makes the structure stiffer. RBE3 elements distribute the force, by using weighted average, to dependent nodes. They do not make the structure stiffer.

3.2 Finite Elemenet Model Assembly

Complete Vehicle Model Tool

CVM is a GUI based interface that is used in UNIX environment to integrate FEA models of components of intrest. It assembles (by creating connections) the required FEA model. CVM is used to model the virtual rig. It is important to note that ANSA can be used to assemble different FE models but CVM makes it more user friendly. CVM can also be used to setup solver type (eg. SOL103, SOL112, SOL109) which can also be done by editing the NASTRAN(∗.dat) file. CVM is also used for running complete vehicle simulations.

3.2.1 Rig Test: FEM

Virtual rig tests can reduce lead time of actual rig tests, by providing more ease in testing and value to the test results. It allows to test more freely for a wide range of load-cases. The work of virtual testing starts from making a virtual model of the structure to be tested.
Two finite element models of the rig test set-up were made: one without the battery box (see Figure 3.3) and one with it (see Figure 3.4). These finite element models have five cross-members.

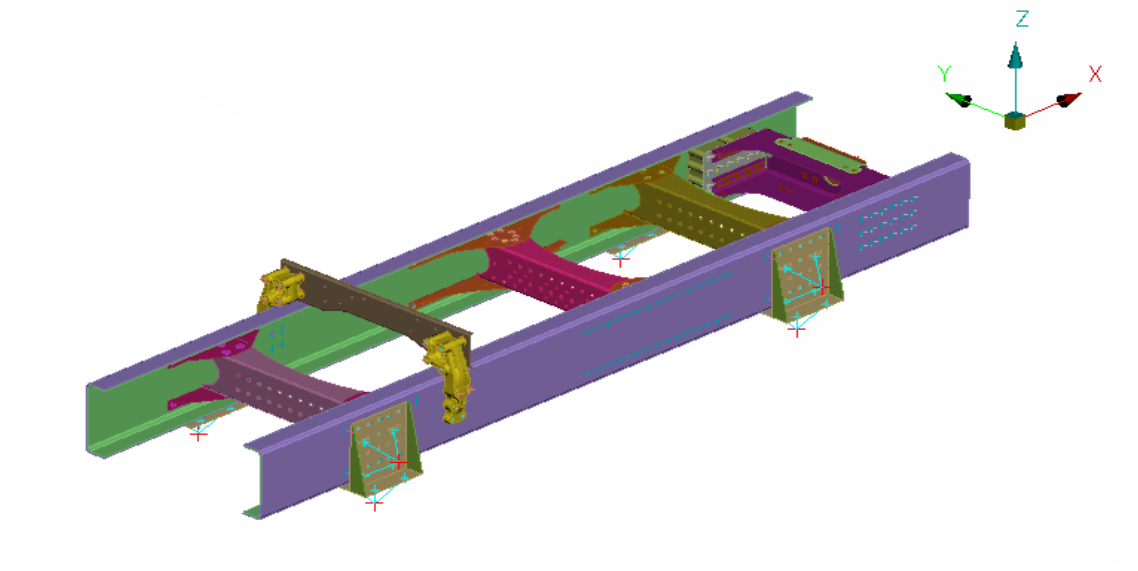

Figure 3.3: Finite Element Model of Rig Test

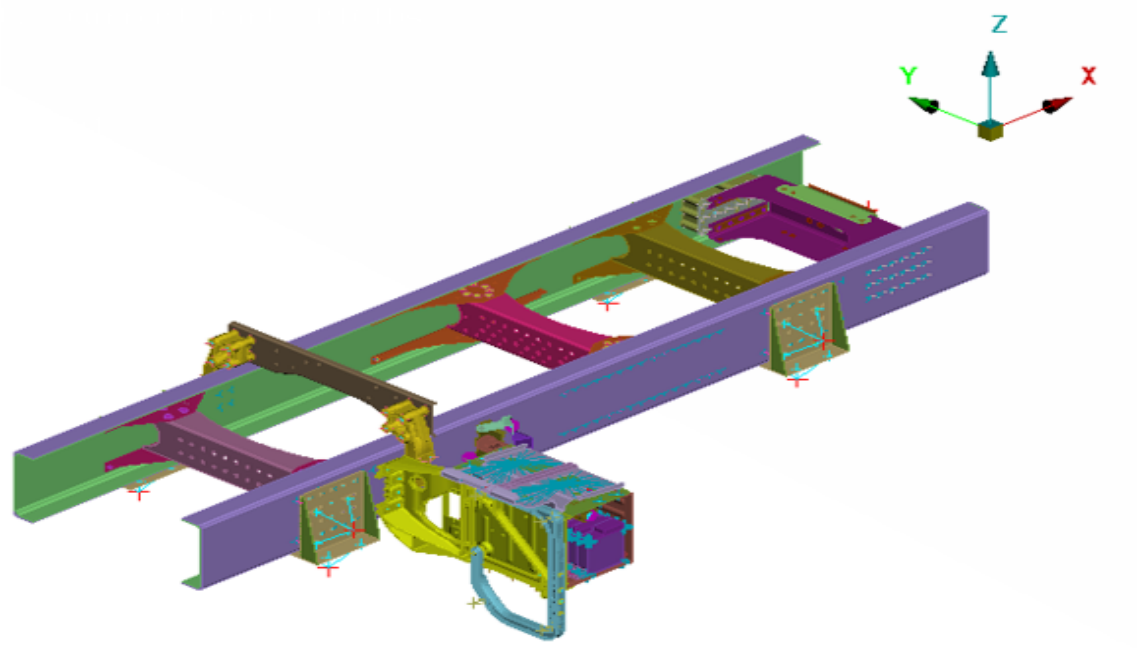

Figure 3.4: Finite Element Model of Rig Test with Battery Box

3.2.2 Hammer Test

One finite element model of the hammer test set-up was modelled. This set-up has four cross-members (see Figure 3.5).

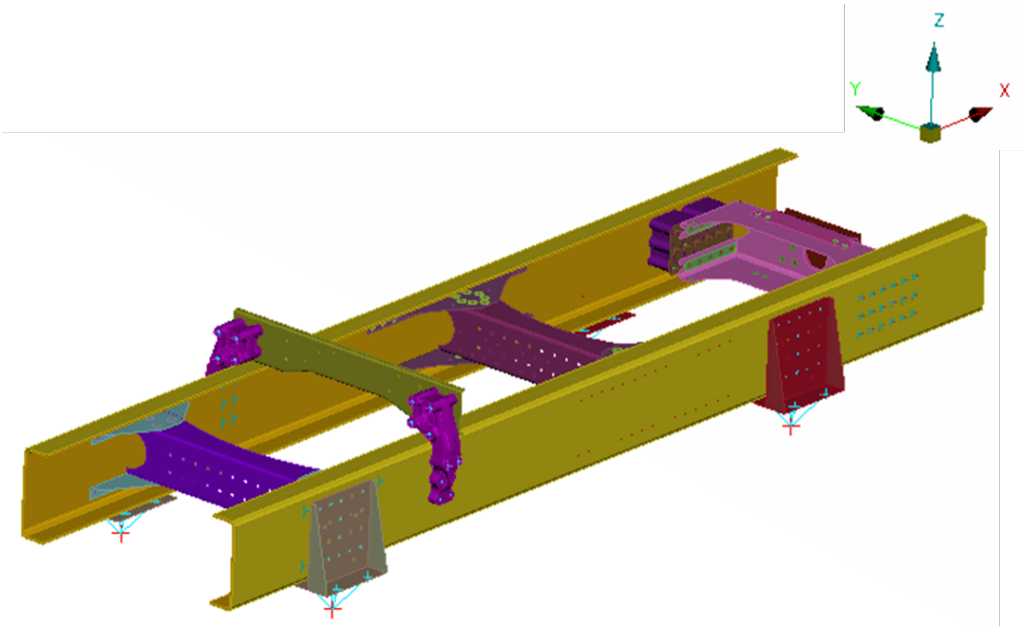

Figure 3.5: Finite Element Model of Hammer Test

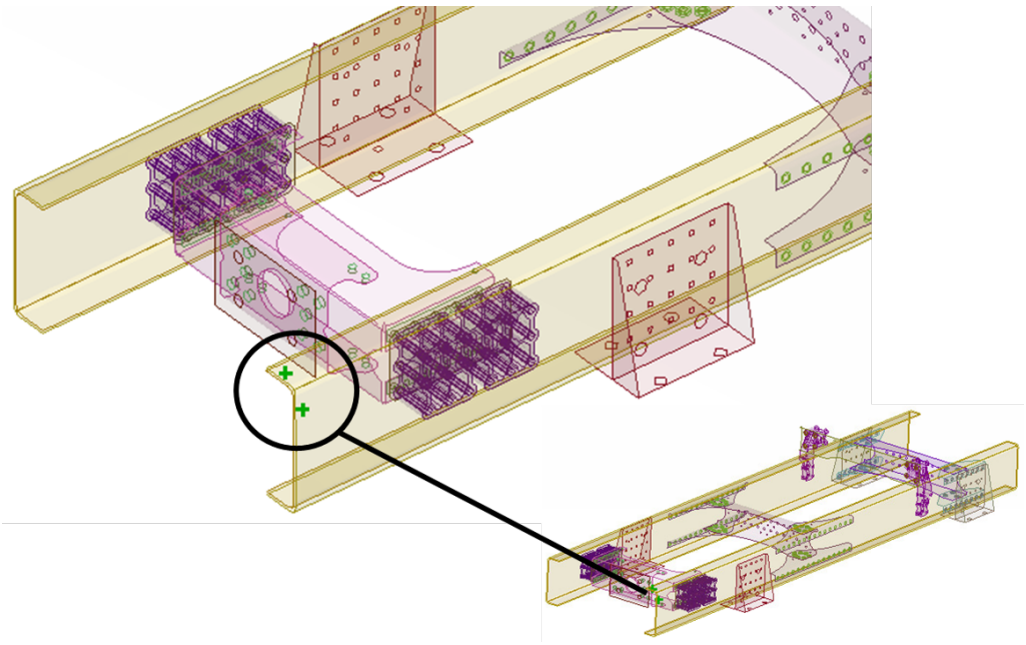

Figure 3.6: Hammer Test: Excitation Locations

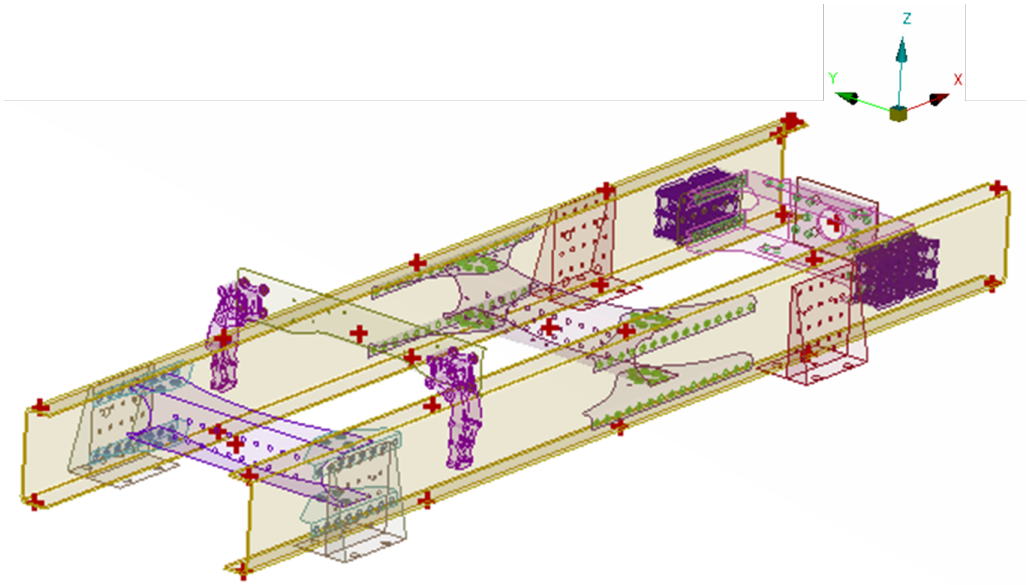

Figure 3.7: Hammer Test: Accelerometer Locations

The force input locations can be seen in Figure 3.6. The accelerometer locations can be seen in Figure 3.7. It is necessary to correctly model these locations because they are given as input to FEMCali. The grid points are not always exactly located at these positions. So, the grid points are translated, without changing the geometry of the model, such that these locations and corresponding grid points coincide.

4

Experimental Testing

This Chapter discusses the procedure of the tests that were carried out to obtain experimental transfer functions. These transfer functions were later used in the study during system identification and correlation. Calibration and correlation processes require a reference to which the simulation data is compared. The response obtained from the finite element model was evaluated with the results from the experimental tests in order to calibrate the finite element model.
Two tests were carried out to record the dynamic behavior of the frame. In rig test (See Section 4.1), the frame was mounted to the 7-post durability test rig, which uses displacement controlled input. In hammer test (See Section 4.2) the frame is mounted on a fixture with force input at 2 locations. Hammer test was conducted prior to commencement of the study.

4.1 Rig test

The 7-post shake rig testing facility at Volvo GTT, Lundby is used to perform tests on truck frame to record the dynamic response of the system. In this study, the test was conducted in two phases: One with only the frame mounted on the rig (See Figure 4.1) and the second phase had a component (battery box) mounted on the frame.

4.1.1 Background

One of the objectives of the study is to measure the force values at actuator locations, such that it can be used in transfer function synthesis (See section 5) later in the study. The idea behind this was based on a study conducted earlier at Volvo GTT documented in the report "7 poster chassis shaker simulation" [15]. A brief summary of this method is explained in this section.

The vehicle simulations were carried out using CVM (See Section 3.2 for more details about CVM) as a database and NASTRAN as the solver. Different load cases corresponding to the different events at the Hällared proving ground (HPG) are stored in CVM. The acceleration response was calculated from test track simulations (vehicle simulations) in CVM and these results were used for comparison with the rig simulations. There were two load cases defined for the rig simulations: one was having a displacement-controlled input and acceleration as the output and the

other was having a force-controlled input with acceleration as the output.

A finite element model of the rig was developed using the part database in CVM and the simulation of the rig model was carried out in NASTRAN to obtain the acceleration response of the system. The acceleration output from these rig simulations were compared with the acceleration output from the vehicle simulations. An optimization loop, developed in MATLAB, was used to improve the correlation between the acceleration output from vehicle simulations and rig simulations for both the load cases. The optimization loop optimized the input for both the rig simulation load cases to get close correlation between the output channels at a number of predefined locations on the frame model.

Power spectral density, level crossing graphs and potential damage measures were used to check the correlation of the rig simulations with the vehicle simulations. It was observed that the force input rig simulation gave better correlation than the traditional displacement input simulation.

4.1.2 Test Setup

Calibration study requires reliable test data based on which the finite element model is calibrated. Having good test data is therefore a crucial first step. The test facility at Volvo GTT is used to conduct tests on the truck frame and record their dynamic response. The tests are generally carried out taking into consideration the actual road load cases.

Figure 4.1: Rig test set-up

The shake rig consists of seven hydraulic cylinders in a controlled environment. These hydraulic cylinders are used to excite the system using displacement con-

trolled input. Four out of the seven actuators provide input in the vertical direction (positive z), two actuators provide input in the lateral direction (positive y) and one in the longitudinal direction (negative x). The sensor(accelerometer) locations on the chassis frame, mounted on the shake rig, have been predefined by Volvo. These locations are chosen in a way that they can provide maximum information of the dynamic response of the system. The obvious locations for sensor placement on the frame are the points close to the excitation location possibly within 40 cm. This helps in capturing more accurate values of the corresponding output accelerations from the sensors. Figure 4.1 shows the test setup with the actuators. See Section 2.2 for more information on the test setup.

The chassis frame structure was mounted on the shake rig. The test object was a stripped straight frame consisting of a siderail and an inner liner. The siderail and the inner liner have a hole pattern throughout the 3526 mm length. The structure also consisted of 5 cross members made of steel.

The shake rig allows input to be provided in different ways. The input signal can either be a single input, where only one actuator excites the system or multiple input where all actuators are used simultaneously to excite the system. The input signal can also be of various types (See Section 2.2.5 for more information on input signal). The input can be a random excitation signal or an actual load case from the proving ground. This input is provided to the rig using a software where the amplitude and frequency of the signal can be changed and altered according to the requirement. Two of the commonly used software packages at the durability lab at Volvo GTT are RPC PRO and LMS. These software programs also allow different types of outputs according to the requirement. Refer Section 2.2.1 for more information on the rig control software.

4.1.3 Displacement controlled test

The shake rig is displacement controlled. However, as reasoned earlier in Section 4.1.1, using force values during correlation process would give better results. Hence, the existing displacement controlled rig was modified to obtain force value for corresponding displacement input. This was achieved by introducing force transducers at the interface between actuator and rigid attachement (The yellow part in Figure 4.2). Figure 4.1 shows the 7 post test rig setup.

The force transducers (See section 2.2.4 for more information on force transducers) were mounted at the interface between the hydraulic actuators and rigid attachments (See figure 4.2). The rigid attachment is bolted to the chassis frame. The force transducers measure force corresponding to the input displacement. This is not exactly a force controlled method since the force is calculated from the rig as an output from the force transducers. This output force is then used to synthesize transfer functions which will be discussed in Chapter 5.

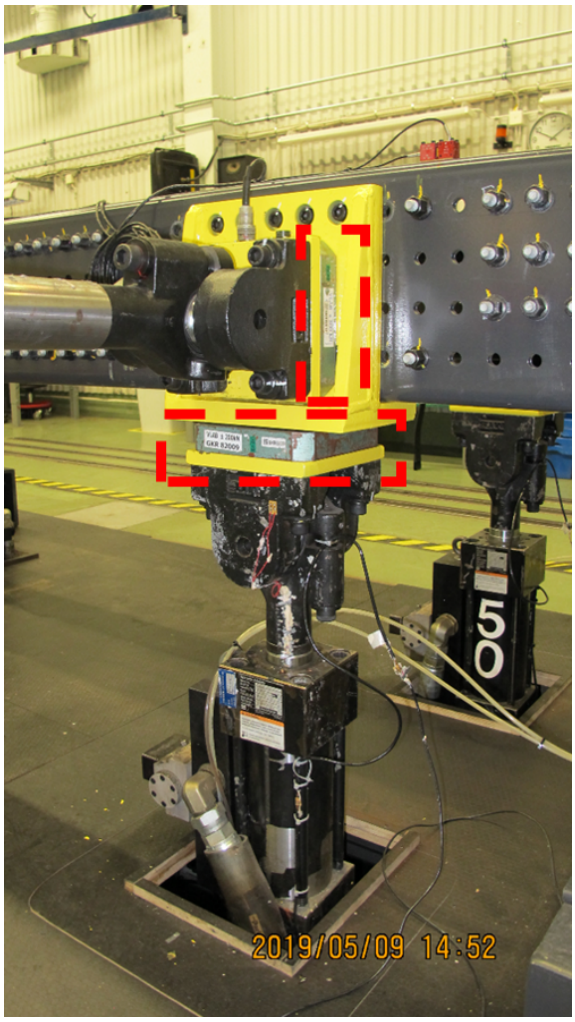

Figure 4.2: Force Transducer Location on the Shake Rig

The seven force transducers record force values for corresponding displacement input from the actuators. Each of them was mounted between the actuators and the frame. Seven accelerometers were mounted on the frame. See figure 4.3. The sensors used for the test were triaxial accelerometers (See Section 2.2.3 for more information on accelerometer). These accelerometers measure the accelerations in all the three directions. The sensors were placed in a way that the local coordinate system of these sensors was aligned with the global coordinate system. All this information was captured in the DAQ system (See Section 2.2.2 for more information on DAQ system) through twenty-one channels. The first seven channels recorded the displacement input, the next seven recorded force values and last seven channels recorded acceleration output.

Once the set up for the test was done, the system was excited using white noise of varying amplitude having a time period of 600 seconds. This signal was used to excite the system and the results for the test were studied. The data obtained from this test did not give good transfer function estimates. See Section 8 for more information on these transfer function estimates. So, the test was carried out using

a different signal.

In the second test a periodic random signal was used for exciting the system for 30 cycles. The signal was a modulated displacement amplitude wave having a time period of 136 seconds. See Section 8 for more information on the transfer function estimates obtained from this test.

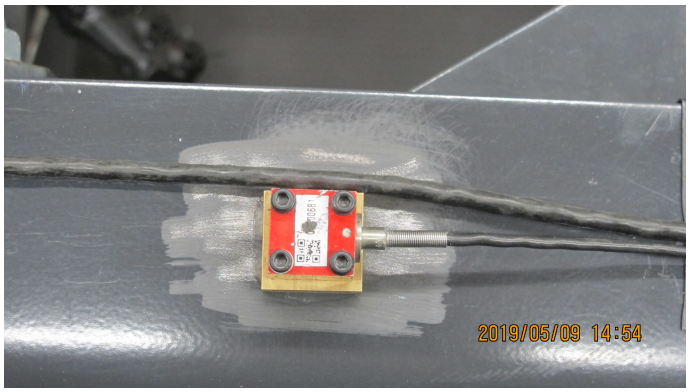

Figure 4.3: Sensor placement for the rig test

Besides recording the dynamic response of the frame, the dynamic response of the battery box attached to the frame was also recorded. Three additional acceleration channels were used for this purpose. Therefore, two tests were conducted, one with only the frame mounted on the rig and in the second test battery box was attached to the frame. The battery box had an accelerometer placed on it for measuring the accelerations in all the three directions. The results obtained from the test were used to synthesize transfer function estimates. The methodology used for transfer function estimation is discussed in Chapter 5, and the results are described in Chapter 8.

4.2 Hammer Test

This test was experimental modal analysis of a frame using a force controlled input technique. This test was conducted previously at Volvo GTT. The test results were noted in an engineering report , 'Experimental modal analysis of chassis component shaker rig frame for correlation with durability virtual FE rig model'. The procedure of the test is summarized below.

4.2.1 Test setup

Hammer test carried out on a frame structure. The frame consited of two side rails, two inner liners and four cross members attached to it. It had a through hole pattern similar to the test object in the rig test. The major difference between the rig test object and this EMA test object were the number of cross members and their attachment positions on the frame. The frame had slightly different dimensions then

the frame in the rig test. The dimensions of the test object can be seen in the figure 4.4. The hole to hole distance in the horizontal direction is 50 mm and the hole to distance in the vertical direction is 60 mm. The figure also shows the placement of the cross members. The cross members are numbered 1 to 4 according to their placement on the frame.

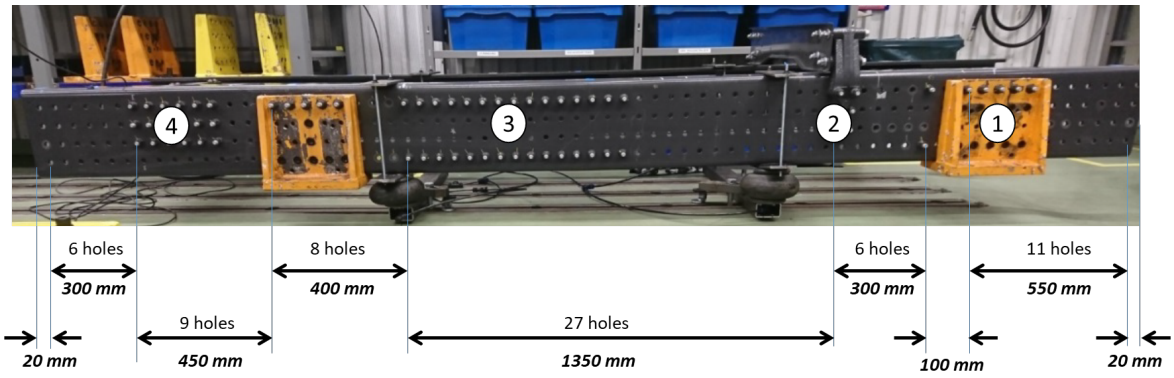

Figure 4.4: Dimensions and cross member positions

In order to simulate a free condition of a structure, it is mounted on soft rubber isolators filled with gas or the structure is suspended using flexible bungee chords. In our test the frame was mounted on a fixture and supported by air bellows. A free-free response is a modal response of a structure when the structure is assumed to be suspended freely in space and not constrained in anyway. Such a structure when subjected to an external excitation will give the eigenmodes of the structure.

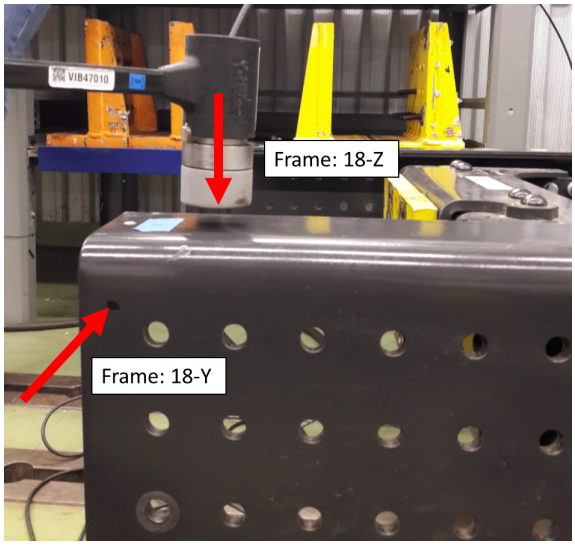

Figure 4.5: Force input points

The sensor placement locations and the force input location can be seen in figures 4.5 and 4.6 respectively.

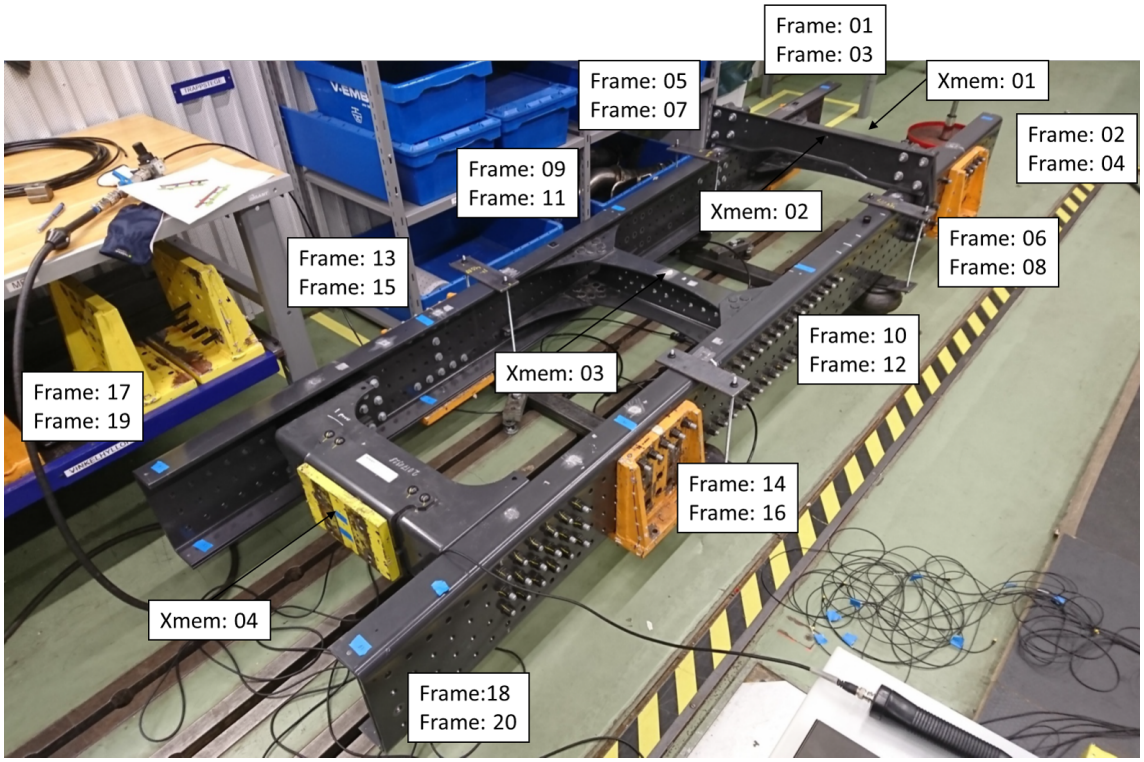

Figure 4.6: Sensor placement positions

4.2.2 Force controlled test

The input for the test was an impact from a hammer in two directions (Y and Z). The idea was to have a single input multiple output (SIMO) signal to excite all the modes in the desired frequency range of 0 to 100 Hz. This approach however did not capture all the modes and hence an approach to excite the system in two directions was used. The test object was first excited using a force in the Y direction and acceleration data from various sensor positions on the frame were captured. This SIMO was then added to the second SIMO signal obtained by force input in the Z direction.

There were 24 sensor positions chosen for the test. A rowing accelerometer approach was used for it. In this seven triaxial accelerometers were placed at the first seven positions out of the twenty four sensor positions and the tests (excitation in Y and Z direction) were carried out. This data was recorded and the sensors were then removed and mounted on to the next seven positions and the system was excited again. This was done till data from all the twenty four sensor locations was captured.

LMS software was used to capture and process the data. The data obtained from the test was in the form of a frequency response function (FRF) and had to be processed in order to use it in FEMCali. The procedure of the processing of the test data is explained in Chapter 5.

The rig test was carried out to validate the finite model of the frame structure mounted on the rig as mentioned in Section 4.1. The test did not work out as

anticipated and hence the results obtained from the Hammer test were used in the study.

5

Signal Processing

This chapter discusses the various signal processing tools and methods adopted in this study. The data acquired from any test is not free from noise even if the test is performed under controlled conditions. This noise can be due to a number of factors around the test object. In both the vibration tests, the response from the tests had noise that needed to be eliminated. It is due to this noise, that the test data cannot be used directly for comparison with the finite element model. Moreover, the test data needs to be filtered and synthesized in order to get it in a suitable, desired format needed for comparison with the finite element model.

MATLAB was used for processing the data from both the tests. MATLAB is a good tool for signal processing as it provides many possibilities with its different functions made specially for processing signals. The MATLAB code for both the tests can be found in the Appendix A.

The two tests mentioned in the previous chapter had different output formats. The first test which was a rig test had multiple format outputs (force and acceleration) and the hammer test (experimental modal test) had acceleration output. The processing techniques used for the two outputs were different. Section 5.1 explains about the signal processing done for the rig test.

5.1 Rig test

Digital Fourier analysis if not carried out properly can lead to errors in the results. This section discusses about the processing done on the signal obtained from the rig test. The test data obtained from the rig test was in the time domain. The signal processing for the rig test data can be divided into two parts: signal pre-processing which describes the processing done on the time series data and the second part is signal post-processing. This part describe the conversion of the time series data into frequency domain and the processing done after.

5.1.1 Filtering

In the periodic random excitation, the signal obtained was in the time series domain. This data was processed in order to extract better information from it. The processing of the signal was done in MATLAB. A code for signal processing of both

the white noise and the periodic random excitation was developed (See Appendix A). The 30 cycles time series data was first averaged out using *mean()* function. The trend in the signal was removed using *detrend()*. This function helps in removing any polynomial trend from the data. Once this was done, a band pass filter was used to filter out the noise. It filters out unwanted data and frequencies.

5.1.2 Transfer Function Estimate

It is efficient to treat test data in the frequency domain when using it for model calibration purposes as it is easy to extract information about the structures dynamic behaviour.

Transfer functions are estimated from input and output signals. In this study, transfer functions were estimated for each channel by using force signals and acceleration signals obtained from the DAQ system. The MATLAB function *tfestimate()* was used for this purpose. The function *tfestimate()* uses Welch's averaged periodogram method.The accelerance data was divided into segments by using windowing. This helps in getting better estimation of the transfer function.

The relationship between the input x and output y is modeled by the linear, time-invariant transfer function txy in MATLAB [13]. In the frequency domain, Y(f) = H(f)X(f). Either 'H1' or 'H2' transfer function estimators can be used for estimating the transfer functions. 'H1' estimator is used when the noise is uncorrelated with the input signals [13]. 'H2' estimator is used when the noise is uncorrelated with the output signals [13]. In this study, there were two input signals and 72 output signals. See Section 2.3.2 for more information on transfer function estimation.

Once the transfer functions were estimated, the processed data was checked for force uncorrelation using the singular value decomposition. Other checks were done using the Power spectral density plot and by calculating the Fast fourier transform of the processed data. Refer section 8.1 for the results of the rig test.

5.2 Hammer test

As discussed in the previous chapter, this test was carried out using the LMS software and the output was in the form of frequency response functions (FRFs). A MATLAB code was developed to carry out the filtering and synthesizing of the test data (See Appendix A).
The test was carried out in a way that at a time only seven sensors were mounted on the frame. The test was repeated a number of times by remounting the sensors at different points on the frame. This rowing accelerometer approach was used due to the availability of limited number of multiaxial sensors.

Two sets of single input multiple output (SIMO) FRFs were obtained due to two input directions. There were a total of seventy two output accelerations (twenty four accelerometer locations each having acceleration measured in all three directions). These two sets of SIMO signals were first combined to make a multiple input multiple output (MIMO) set. This new MIMO signal was stored as a 3-dimensional array of the order 72 x 2 x frequency. The cutoff frequency was 150 Hz since the desired frequency range is from 10 to 100 Hz. The reason for keeping the extra 50 Hz is explained in Chapter 6 where the effect of frequencies outside the desired frequency range are discussed. The system identification procedure for the MIMO signal is explained in the next chapter.

6

System Identification

This chapter describes how the synthesized transfer functions were used to identify the system (See Figure 6.1). System identification was used to build mathematical models based on the dynamic behaviour of the system. Information about the dynamic behaviour was obtained from experimental modal analysis (EMA). These models were used to calibrate and validate finite element model of the system. Therefore, it is a necessary to get good identified models, so that it can be used in calibration and validation process [6] [4].

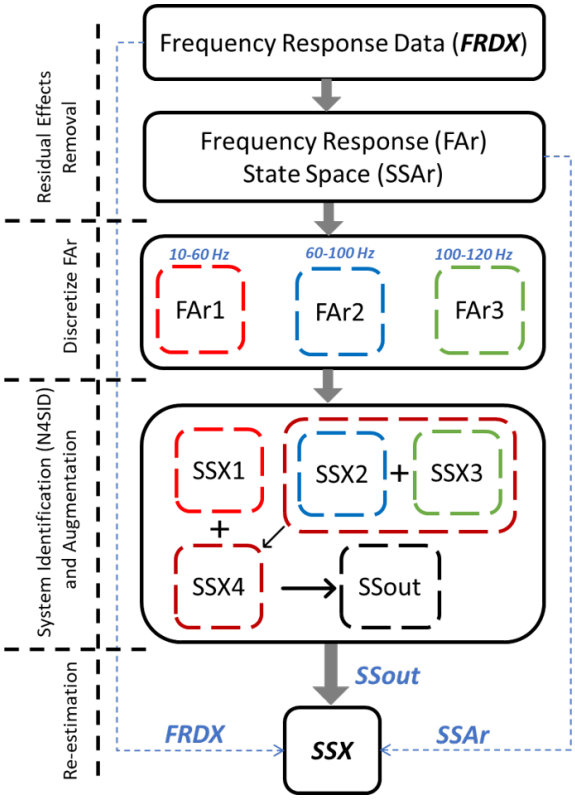

Figure 6.1: Overview of System Identification Process

In this study, linear state-space model structure was used in system identification. These state-space models (state-space system matrices) were estimated by using N4SID algorithm (Subspace method). The N4SID algorithm gives flexibility while positioning poles and the identified poles are not constrained to lie in the frequency range of test data [4]. The identified state-space models were optimized using different techniques as discussed in section 6.1, section 6.2 and section 6.3. The system

response from EMA, FEA and state-space model were compared for each channel. The location of local maxima(resonances) and local minima(anti-resonances) were observed and were used to improve the model (by improving individual channels, identifying faulty channels).

6.1 Residual States

The frequency response data object obtained after signal processing has data from 0-120 Hz (see chapter 5 for details). The frequency band under study is from 0-100 Hz. From the EMA data it is observed that there are four rigid body modes (RBM<4 Hz) and four flexible modes in the frequency range 0-100 Hz. There are two more modes between 100-120 Hz. The strongly participating modes outside the frequency range affect the modal behaviour in the test frequency band [4].

A MATLAB script was used to remove these residual effects from the frequency band we used to model our system. The MATLAB function *removeresidualeffectsNEW (FA,beta,gamma)* takes the frequency response data object and two pole positions as input. The location of the two poles is decided by using factors γ and β. Alternatively, the values of γ and β can be predetermined based on the modal behaviour of the rig. The first pole was placed at 1.43 times the upper cut-off frequency (i.e. γ=1.43). The second pole was placed far off the cut-0ff frequency(i.e. β=100). The location of first pole was chosen such that it does not lie close to the 5th flexible mode. This was done to avoid issues during state-space modelling. The location of second pole was chosen in order to have residual flexibility [4]. The function creates zero-pole model using zpk and frequency response data object and gives it as output.

6.2 State Space Modelling

After, the effects of residual states were removed, the new frequency response object was divided into frequency bands. The division into these frequency bands was based on the location of eigenfrequencies. N4SID algorithm was used to model state-space of each of these frequency bands. Then these discrete state-space models were combined using *parallel()* funtion in MATLAB. The parallel function concatenates these smaller state-space models to get a final augmented state-space model. The final augmented state space model was then re-estimated.

6.3 Re-estimation with more Iterations

The estimated state space model was further improved by more iterations with the estimation data [5]. Further iterations on the state-space system matrices: B, C and D improved the properties of the model.

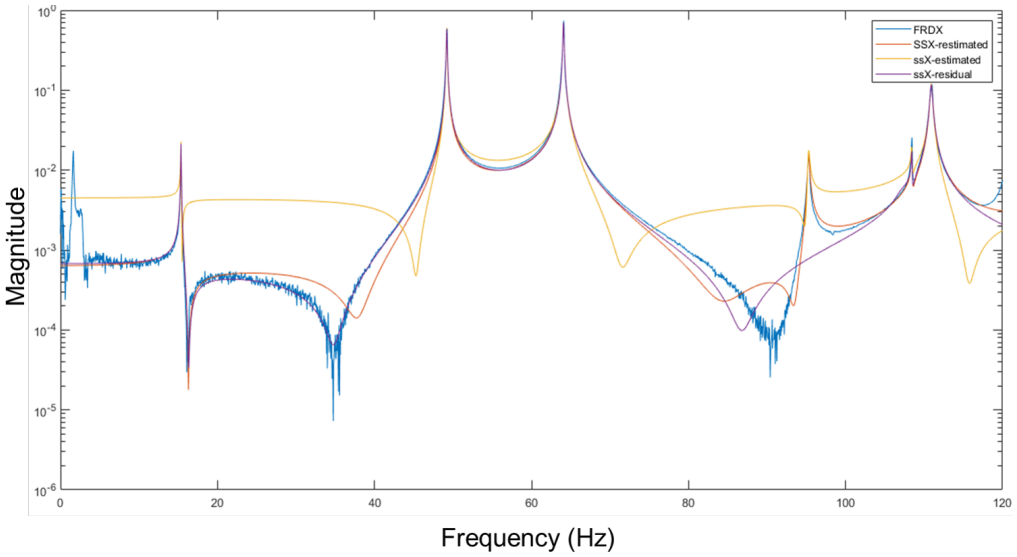

Figure 6.2: State-space Models obtained during System Identification

In Figure 6.2 the frequency response data(FRDX) of one channel is plotted along with estimated state-space models. The state-space model ***SSX-restimated*** captures the resonances and anti-resonances better than other state-space models. The other state-space models either fail to capture the resonances or poorly capture anti-resonances. See Appendix C for frequency response data for all channels.

7

FEMcali

This Chapter gives insight into how the finite element model was parametrized and the calibration process in FEMCali. FEMCali takes frequency response data and state-space model as one of the input arguments. The frequency range under study is specified as 10-100 Hz in the FEMCali toolbox. The grid point numbers of the actuator and sensor locations are entered with their degree-of-freedom. The parametrized NASTRAN file is also an input argument (See Section 7.1 for more information about parametrization). In this study, a reduced-order model (surrogate model with the first 100 flexible modes of the original model) is used for the calibration process. FEMCali runs the calibration loop as described in Section 2.6.9 and Section 2.6.10 in Chapter 2. The overview of the calibration process can be seen in Figure 7.1.

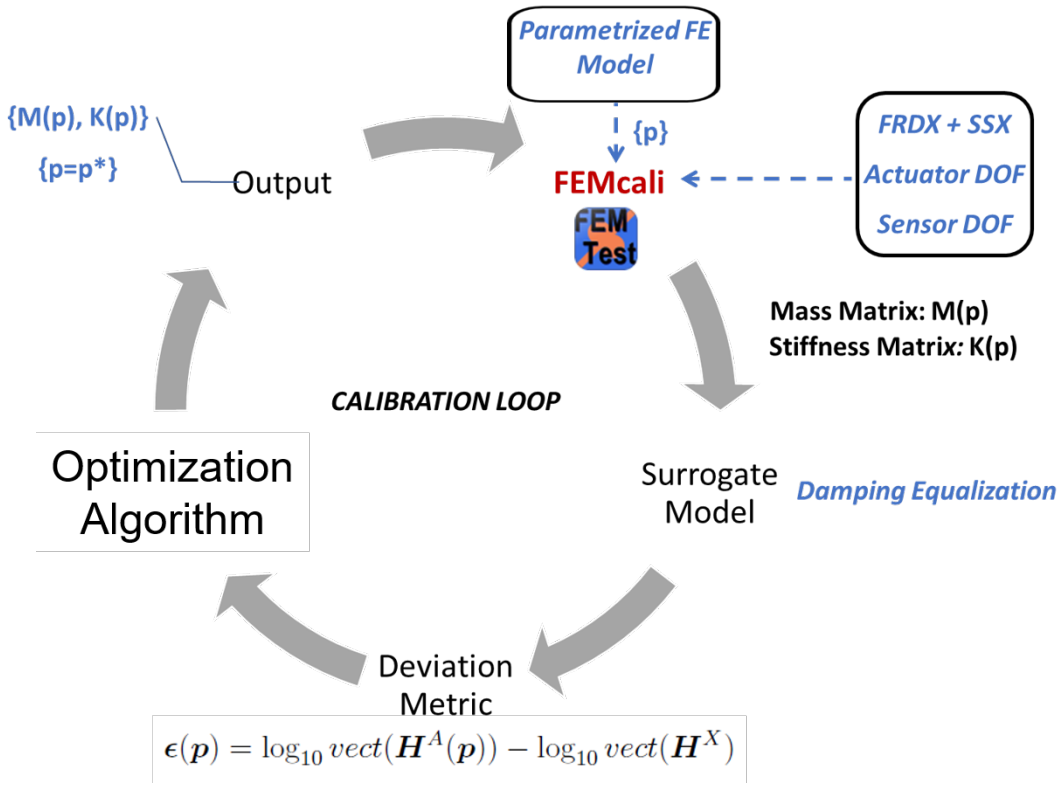

Figure 7.1: Calibreation Process Overview

7.1 Parameterization

The finite element model was parametrized before using it in FEMCali. The modal behavior of the finite element model is sensitive to variation of the parameter values. It is essential to identify those parameters that are most affecting the modal behavior.

The parametrization of the finite element model is done in two steps. First, the finite element model is parametrized by selecting a large set of parameters in the beginning. Then, based on the Cramer-Rao lower bound values the final selection of parameters is done. See Section 2.6.7 for more information on Fisher Information Matrix and Section 8.2.2 for results of the parameter study. These parameters are chosen based on understanding of how they affect the structural and geometrical properties of the finite element model. For example, changing the cross section area of CBAR elements changes the stiffness of joints (See Section 2.5.1 for details about finite element connections).

Thirty seven parameters were chosen initially, thirty of which were finalized as shown in Table 7.1. These thirty parameters have been used throughout the study. The other seven parameters are not included in the study due to high Cramer-Rao lower bound values (See also Section 8.2.2 for more information).

The upper and lower bounds of the parameters are set according to the type of parameter we are dealing with. Parameters like density and Young's modulus are not varied beyond 40% of the nominal value. This is done to avoid unrealistic values for these parameters during the calibration process. Other parameters like cross section area and shell thickness can be varied based on requirements. CBAR elements have cross section area as one of the input arguments. This was parametrized for some CBAR elements.

7.2 Model Calibration

Table 7.1 shows the parameters which were chosen after identifying parameters which affect the dynamic behavior of the rig. See Section 2.6.7 for more information about Fischer Information Matrix. The table shows with their nominal values and the upper and lower bounds chosen for each parameter. The table also shows the type of property that is being parametrized along with the component.
The first 9 parameters in the list are the thicknesses of the frame members. The shell thickness influences the stiffness of the structure and hence these parameters were chosen. The next set of parameters seen in the list are the cross sectional areas of the CBAR elements. CBAR elements are used as connections between the members. Increasing the contact area of the connection will make the structure more stiff and thereby shift the modes further ahead. The last set of parameters contain the Young's modulus of the frame members. This set also contains Young's modulus of some CBAR elements. These were chosen in order to check if the material of the

CBARS (the connections) had any effect on the modes.

The model is calibrated using FEMCali Toolbox (as explained in in Section 2.6.9 and Section 2.6.10 in Chapter 2). The parameters(p1-p30) to which the modal behaviour is sensitive are identified first. The inverse of the Fisher information matrix for each parameter is computed in FEMCali. The sensitivity of the finite element model to any variation of these parameters is illustrated by by the Fisher information matrix. The inverse of this matrix is also calculated for the combination of each parameter with each of the other free parameters [10].

Table 7.1: Parameter List

Parameter	Property	Part	Nominal	LL	UL
p1	Shell Thickness	SIDE RAIL	8.00	6.40	9.60
p2	Shell Thickness	INNER LINER	5.00	4.00	6.00
p3	Shell Thickness	XMEMB04	10.00	8.00	12.00
p4	Shell Thickness	XMEMB01	4.00	3.20	4.80
p5	Shell Thickness	XMEMB01 AT	6.00	4.80	7.20
p6	Shell Thickness	XMEMB03 AT	7.00	5.60	8.40
p7	Shell Thickness	XMEMB03	7.00	5.60	8.40
p8	Shell Thickness	XMEMB02	5.00	4.00	6.00
p9	Cross-section Area	X1 AT	254.47	203.58	305.36
p10	Cross-section Area	X3 AT	314.16	251.33	376.99
p11	Cross-section Area	FRA	176.71	141.37	212.06
p12	Cross-section Area	X1-IL2SR	176.71	141.37	212.06
p13	Cross-section Area	X12SR	176.71	141.37	212.06
p14	Cross-section Area	X2-IL2SR	176.71	141.37	212.06
p15	Cross-section Area	X22SR	176.71	141.37	212.06
p16	Cross-section Area	X4-IL2SR	176.71	141.37	212.06
p17	Cross-section Area	X42SR	176.71	141.37	212.06
p18	Youngs Mod	X4 CBAR	210000.00	126000.00	294000.00
p19	Youngs Mod	X4 main	210000.00	168000.00	252000.00
p20	Youngs Mod	X1	210000.00	168000.00	252000.00
p21	Youngs Mod	X1 attach CBAR	210000.00	126000.00	294000.00
p22	Youngs Mod	X3	210000.00	168000.00	252000.00
p23	Youngs Mod	X3 attach CBAR	210000.00	126000.00	294000.00
p24	Youngs Mod	X2 main	210000.00	168000.00	252000.00
p25	Youngs Mod	X2 attach	164000.00	131200.00	196800.00
p26	Youngs Mod	Innerliner	210000.00	168000.00	252000.00
p27	Youngs Mod	Siderail	210000.00	168000.00	252000.00
p28	Youngs Mod	FRA CBAR	210000.00	126000.00	294000.00
p29	Youngs Mod	X1 CBAR	210000.00	126000.00	294000.00
p30	Youngs Mod	X2 CBAR	210000.00	126000.00	294000.00

8

Results and Discussion

This chapter discusses the results obtained from seven post rig test (See Section 4.1) and the hammer test (See Section 4.2). Section 8.1 deals with rig test results and Section 8.2 deals with hammer test results.

8.1 Rig Test

The transfer function estimates obtained from the rig test did not capture the resonant frequencies of the rig. Therefore these transfer function estimates could not be used for state-space modelling, see Section 5.1.2 for more information on the method used to carry out the transfer function estimates. As a result of this, several signal post-processing techniques were used to evaluate the input and output signals. These post-processing techniques were used to identify the reasons behind the poor quality of transfer function estimates.

Singular Value Decomposition of Force Signal

Singular value decomposition (SVD) of the force signal was carried out. It was observed from the singular value matrix (S) that the forces were linearly independent and uncorrelated (see S matrix below). It can be seen that the diagonal values are close to each other and their value is decreasing.

$$S = \begin{bmatrix} 1376.317 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 407.774 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 319.381 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 235.510 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 206.890 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 175.432 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 125.142 \end{bmatrix}$$

Such linearly independent forces are desired for tests and thus the input provided to the *tfestimate* function was correct. Section 9.1 discusses the reasons that might have caused these poor quality of transfer function estimates. This issue could not be investigated further due to time constraints.

8.2 Hammer Test

This section discusses the hammer test results. The discussion is divided into four sections. Section 8.2.1 discusses the eigenmodes of the frame obtained from experimental modal analysis and compares the eigen vectors from test and simulation. Section 8.2.2 discusses the results obtained from parameter study for each of the four flexible modes. Section 8.2.3 discusses the MAC plots for pre-calibration and post-calibration finite element models. Section **??** discusses the effect of the cross member placements on the eigenmodes of the system.

8.2.1 Eigenmodes from EMA and FEA

This Section discusses the eigenmodes of system obtained from experimental modal analysis and finite element analysis. It compares the mode shapes obtained from tesing and finite element analysis of pre-calibration model. The eigenvectors from both the data sets were checked for alignment by plotting eigenvectors (See Section 2.6.4 for more details). Eigenvectors at sensor location points were found out and plotted to illustrate their directions. The mode shape of each mode was constructed using these eigenvectors.

Four flexible eigenmodes were captured in the frequency range of 10-100 Hz. The table 8.1 shows the eigenfrequencies of the four flexible modes obtained from the experimental modal analysis and the finite element analysis. Finite element analysis was done using *SOL103* in *NASTRAN*. It can be seen from the table that the eigenfrequency values are close for the first, second and fourth eigenmodes.

Table 8.1: Eigenfrequency comparison between EMA and FEA results

Modes	**EMA Modal Frequency [Hz]**	**FEA Modal Frequency [Hz]**
Mode 1	15.35	13.19
Mode 2	49.21	46.70
Mode 3	64.10	54.46
Mode 4	95.28	95.04

First Flexible Eigenmode

The first flexible eigenmode has torsional mode shape at 13.19 Hz (from FEA). Figure 8.1 shows the mode shape for first flexible eigenmode obtained from finite element analysis. Figure 8.2 illustrates the eigenvectors and compares the first flexible mode shape obtained from tesing and finite element analysis of pre-calibration model. The pink dashed line shows the mode shape of the experimental testing and the black dashed line shows the mode shape from simulation. It can be seen from Figure 8.2 that the mode shapes match well. The eigenvectors have small deviation in the angle at the ends of the frame but otherwise it is a good match. It is to be

noted that mode shape after calibration is similar to the mode shape obtained from pre-calibration model.

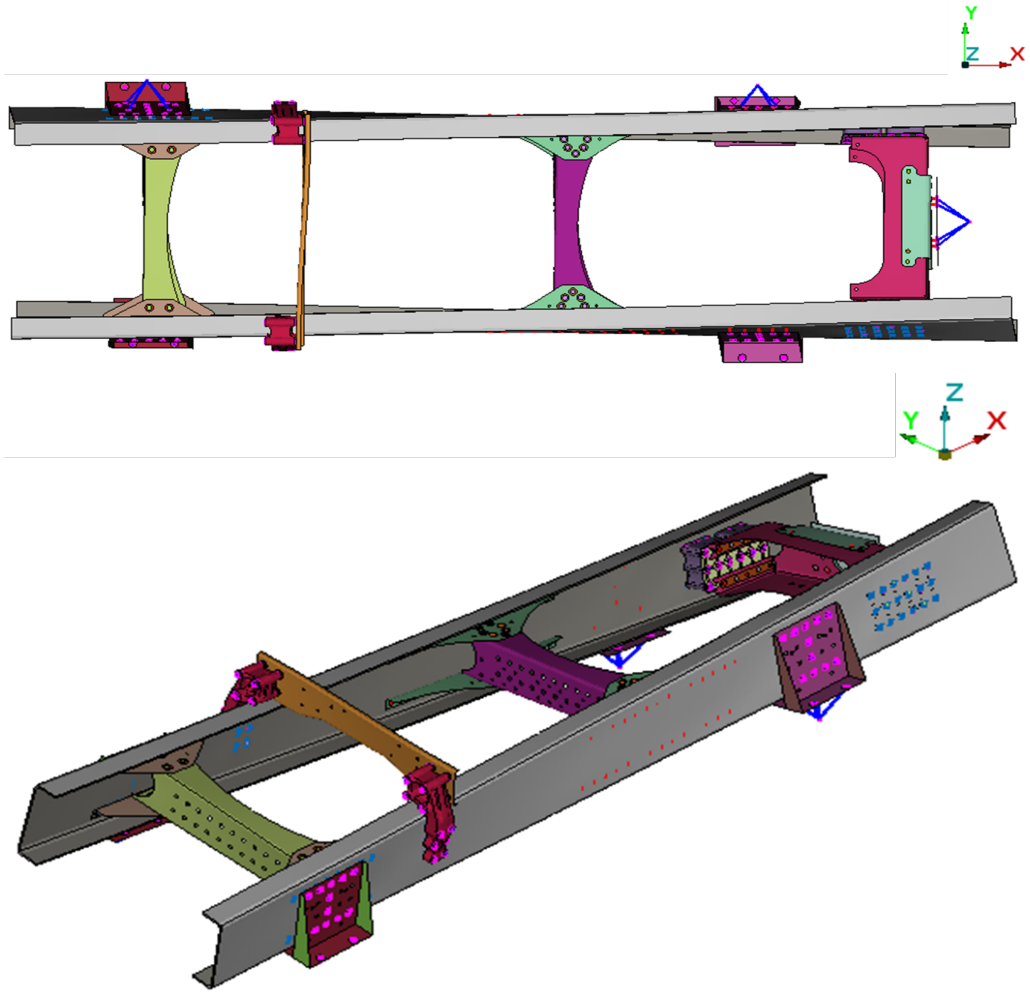

Figure 8.1: First Flexible Eigenmode of Pre-calibration Model

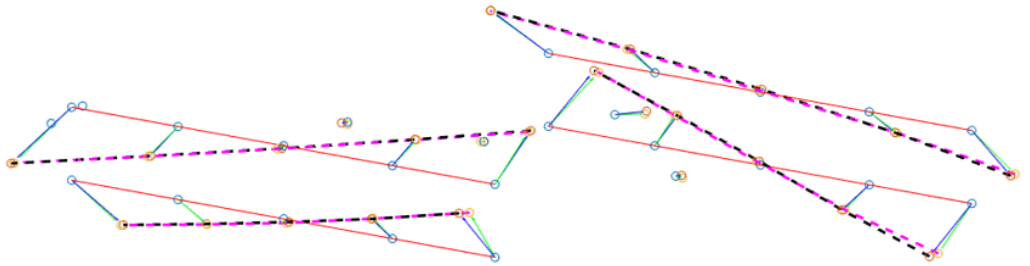

Figure 8.2: First Mode Shape Comparison: EMA and FEA of Pre-calibration Model

Second Flexible Eigenmode

The second flexible eigenmode has bending mode shape (in XY plane) at 46.70 Hz (from FEA). Figure 8.3 shows the mode shape for second flexible eigenmode obtained from finite element analysis. From Figure 8.3, it is observed that, the maximum deflection occurs at the centre of the frame between second cross member and third cross member.

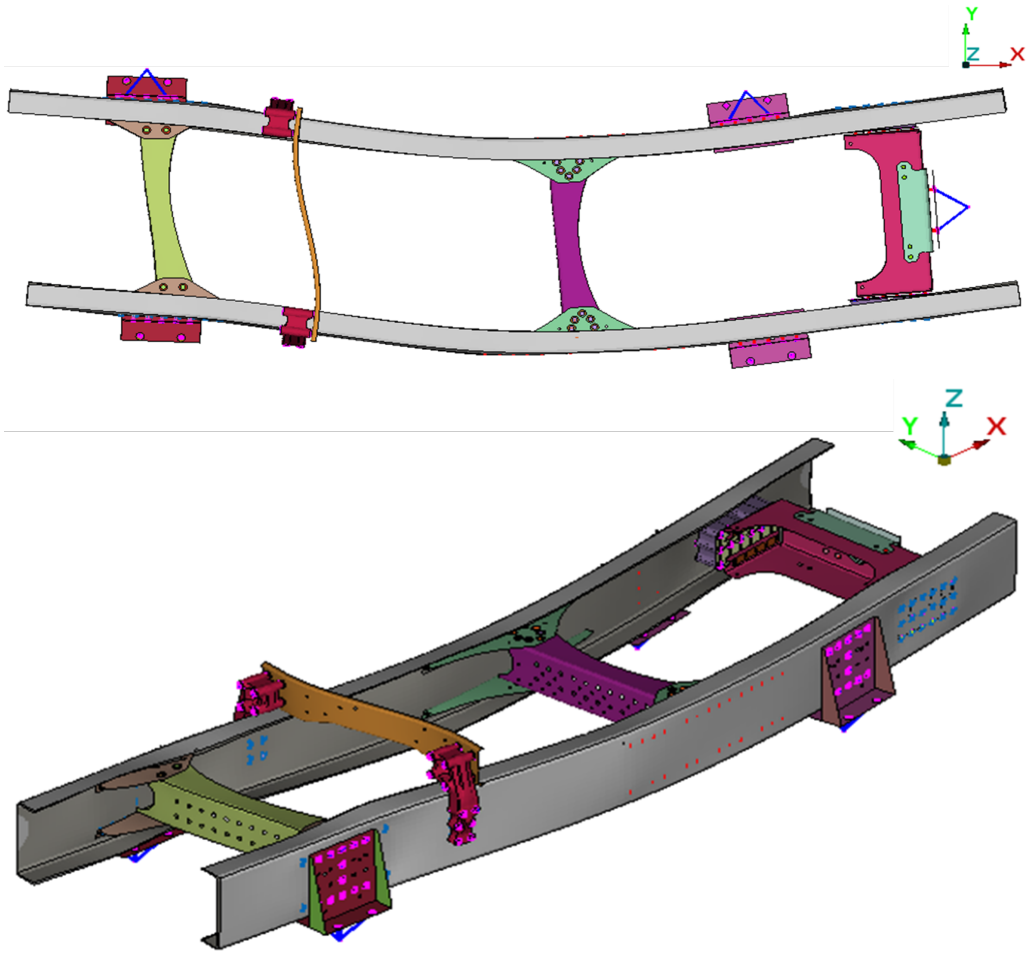

Figure 8.3: Second Flexible Eigenmode of Pre-calibration Model

Figure 8.4 illustrates the eigenvectors and compares the second flexible mode shape obtained from testing and finite element analysis of pre-calibration model. The pink dashed line shows the mode shape (constructed using the eigenvectors) of the experimental testing and the black dashed line shows the mode shape from simulation. It can be seen from Figure 8.4 that the mode shapes match well with small deviations. However, the eigenvectors have no deviation in the angle at the ends of the frame. It is to be noted that mode shape after calibration is similar to the mode shape obtained from pre-calibration model.

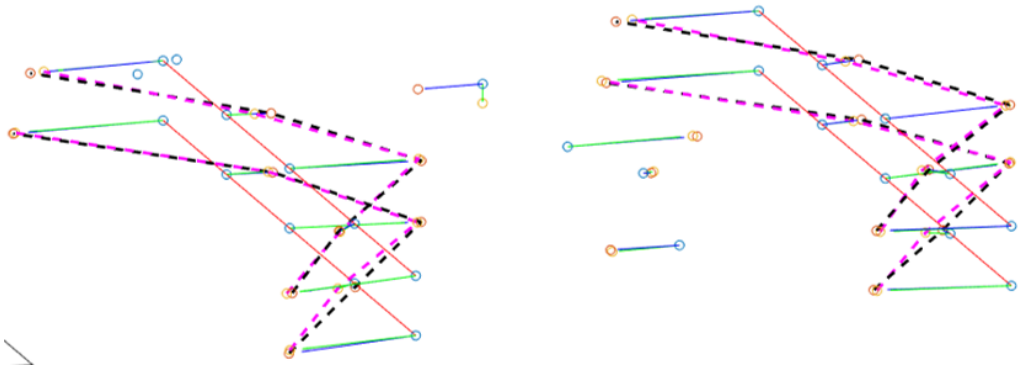

Figure 8.4: Second Mode Shape Comparison: EMA and FEA of Pre-calibration Model

Third Flexible Eigenmode

The third flexible eigenmode has bending mode shape (in XY plane) at 54.46 Hz (from FEA). Figure 8.5 shows the mode shape for third flexible eigenmode obtained from finite element analysis. From Figure 8.5, it is observed that, the maximum deflection occurs between third cross member and rig connection brackets at the rear. Therefore, this bending mode shape differs from the second flexible mode shape.

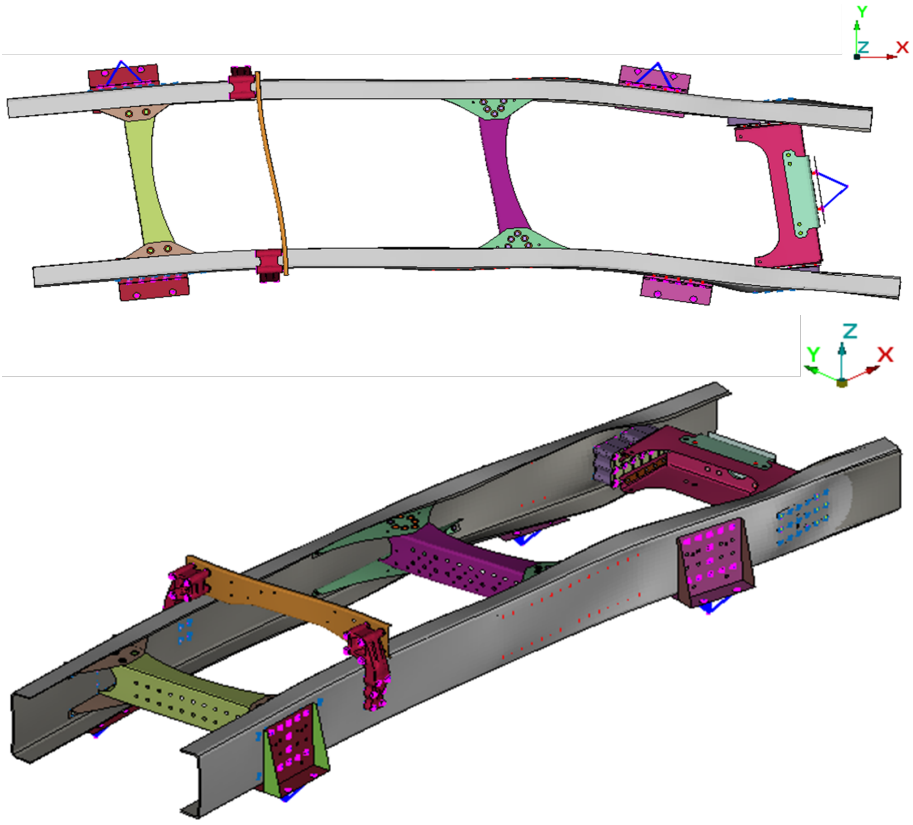

Figure 8.5: Third Flexible Eigenmode of Pre-calibration Model

Figure 8.6 illustrates the eigenvectors and compares the third flexible mode shape obtained from tesing and finite element analysis of pre-calibration model. The pink dashed line shows the mode shape of the experimental testing and the black dashed line shows the mode shape from simulation. It can be seen from Figure 8.6 that the mode shapes do not match in the region where there is maximum deflection. It is observed that,the finite element model deflects more than the test data. This can be attributed to the test set up. The frame structure was supported by a cross support placed near third cross member (See Figure 4.6). It can be due this support that the deflection was seen to be less in test data.

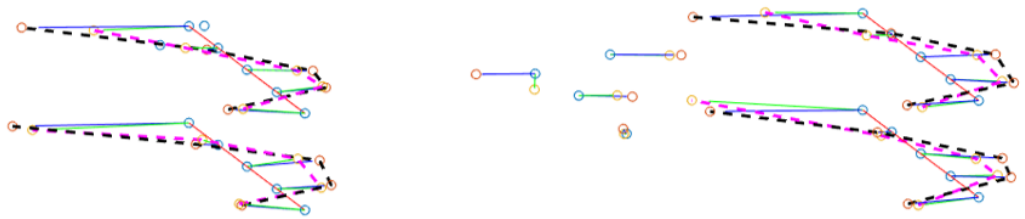

Figure 8.6: Third Mode Shape Comparison: EMA and FEA of Pre-calibration Model

Fourth Flexible Eigenmode

The fourth flexible eigenmode has local bending mode shape close to the rear rig attachment brackets at 95.04 Hz (from FEA). Figure 8.7 shows the mode shape for fourth flexible eigenmode obtained from finite element analysis.

From Figure 8.8, it is observed that, the mode shapes are similar. However, there is considerable deviation at both the ends of the frame and also at the sensor location points. The deflection at the rear end of the frame is observed to be inwards for the experimental test data where as the deflection is outwards for the finite element model. The eigenvectors have deviation in the angle between them.

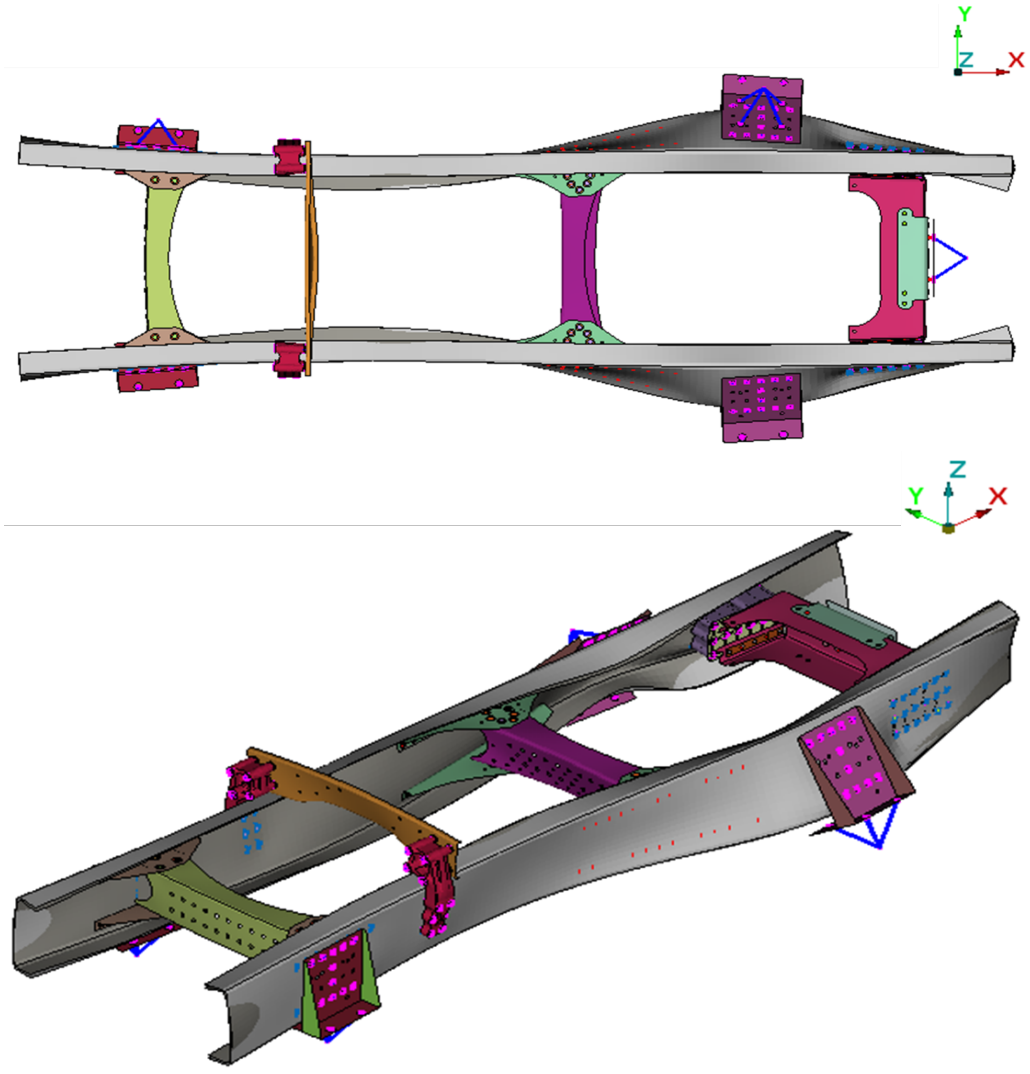

Figure 8.7: Fourth Flexible Eigenmode of Pre-calibration Model

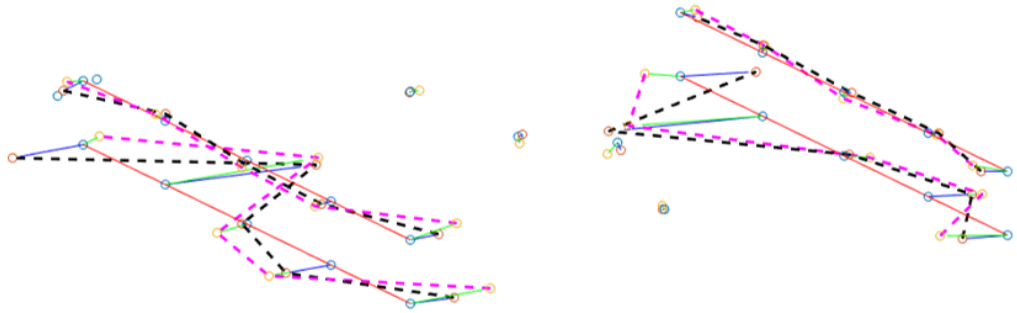

Figure 8.8: Fourth Mode Shape Comparison: EMA and FEA of Pre-calibration Model

8.2.2 Parameter study

In finite element model calibration, the model parameters need to be chosen carefully. There were thirty seven parameters that were initially chosen. When a FEmcali run was carried out using these thirty seven parameters, it was observed that seven of the parameters (which were the cross sectional areas of CBAR elements) did not affect the modes. Therefore, 30 parameters are used in the study. See Section 7.1 for more information on parametrization. These parameters are analysed in this section. Table 7.1 shows the list of thirty parameters used in the pre-calibration model.

Cramer-Rao Lower Bound Plots

Parameter study in FEMCali is done by calculating the inverse of the Fisher information matrix. FEMCali plots a colour coded matrix of Cramer-Rao lower bound values (CRLB). The diagonal of the matrix shows the influence of the parameters. The upper and the lower triangles shows the covariance between the parameters of the matrix. It is the lightest square (lowest CRLB value) along the diagonal that has the most effect and the darkest square (highest CRLB value) has the least effect on the modal behaviour. According to Cramer-Rao lower bound, more the negative value more is the effect of that parameter. The scale of these plots should be taken into consideration while interpreting these plots.

The parameter study was carried out for the frequency range of 10-100 Hz and also for individual modes. This was done by running multiple simulations in FEMCali with different frequency ranges. This was done to ascertain the influence of these parameters on each individual mode.

Influence of Parameters on First Flexible Eigenmode

Figure 8.9 shows the influence of the chosen parameters on first flexible eigenmode. The frequency range taken for this simulation was from 10-20 Hz. It is observed from Figure 8.9 that two parameters influence the first flexible mode more compared to other parameters. Parameter ***p20*** which is the Young's modulus of first cross member (X1) has the least CRLB value ($\approx$ -2.5) and parameter ***p6*** which is the

thickness of the attachment bracket of third cross member (XMEMB03 AT) also has negative CRLB value (≈ -1.5). Therefore, it can be concluded that these two parameters influence the first flexible mode.

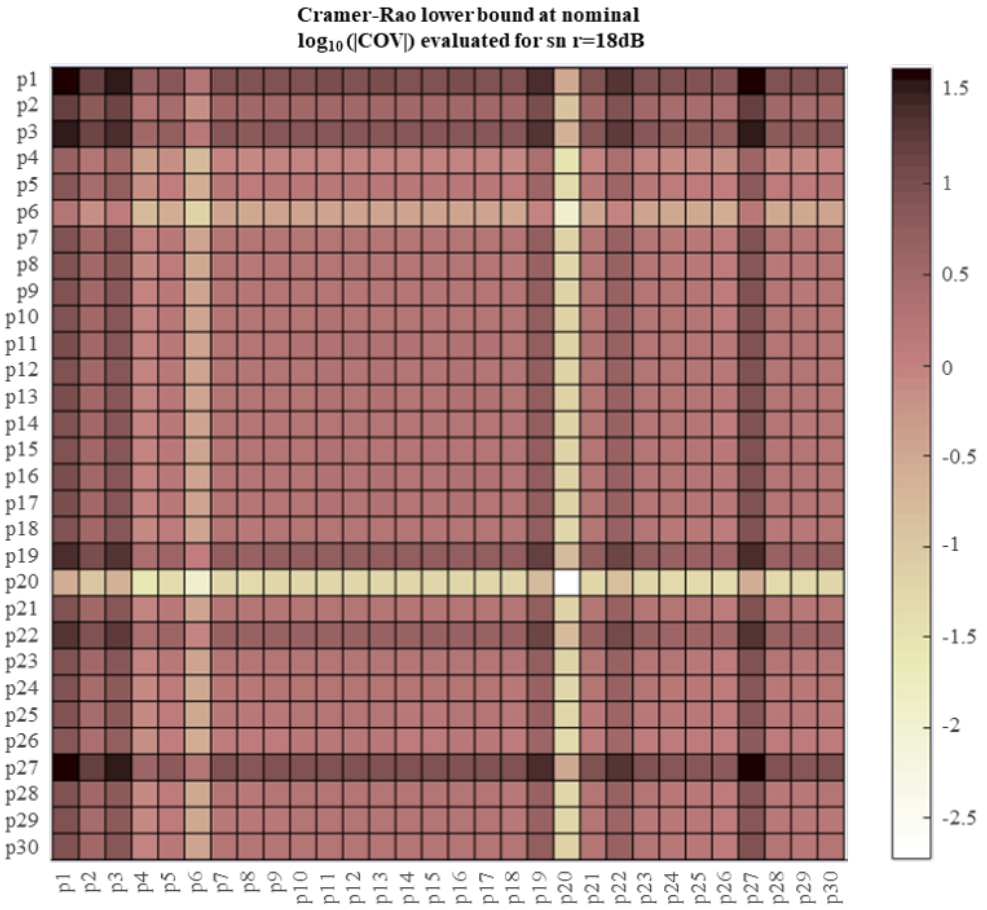

Figure 8.9: Parameter study for First Flexible Mode (10-20 Hz)

It can be seen from Figure 8.9 that the parameters from ***p9*** to ***p17*** (cross section areas of the CBAR elements) do not have considerable effect on this eigenmode. The parameters that have least effect are ***p1*** (thickness of the siderail), ***p3*** (thickness of fourth cross member), ***p27*** (Young's modulus of siderail), ***p19*** (Young's modulus of fourth cross member) and ***p22*** (Young's modulus of third cross member). It can be observed that thickness of siderail and fourth cross member do not influence the first flexible eigenmode.

Influence of Parameters on Second Flexible Eigenmode

Figure 8.10 shows the influence of the chosen parameters on second flexible eigenmode. The frequency range taken for this simulation was from 30-50 Hz. Low CRLB values (along the diagonal) can be seen from ***p9-p17*** (cross section areas of CBAR elements) and also from ***p23-p30*** (Young's modulus). In the thickness parameter set, thickness of second and fourth cross members (***p8*** and ***p3*** respectively) have low CRLB values. The Young's modulus (***p19*** and ***p24***) of the same two cross

members influence this eigenmode.

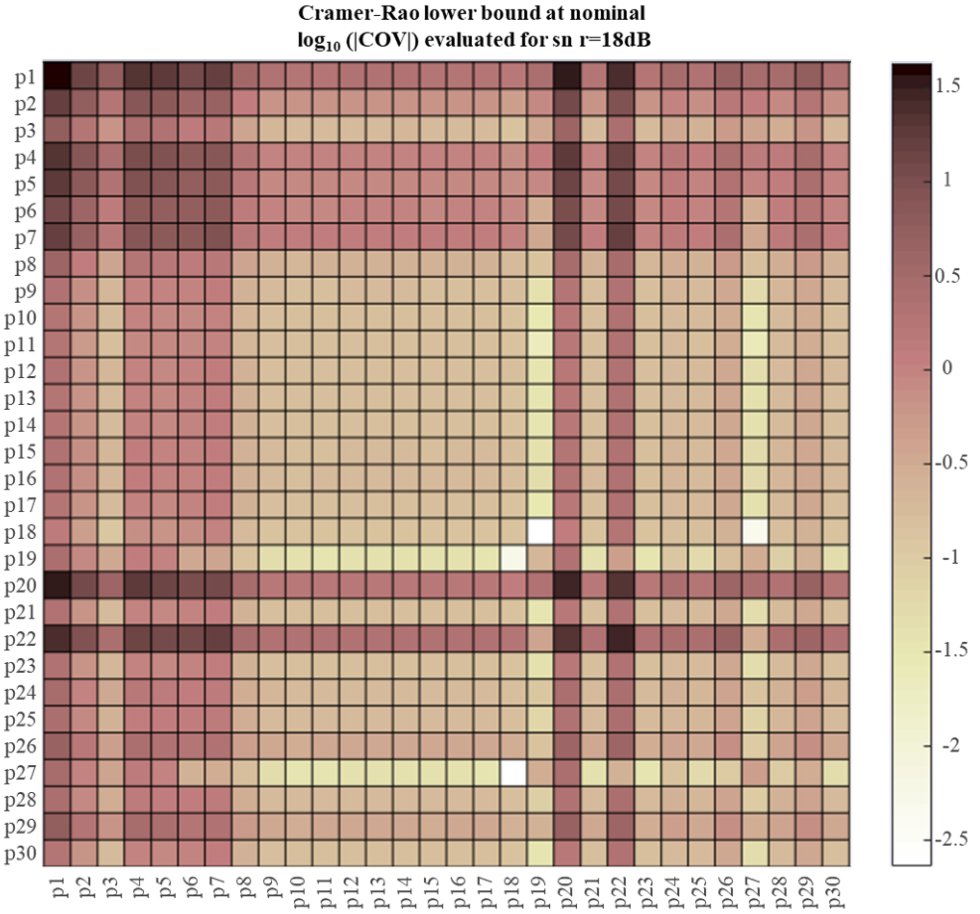

Figure 8.10: Parameter study for Second Flexible Mode (30-50 Hz)

The parameters ***p1*** (thickness of siderail), ***p20*** and ***p22*** (Young's modulus of first and third cross member) have higher CRLB values along the diagonal compared to other parameters, implying low effect on the eigenmode.

Influence of Parameters on Third Flexible Eigenmode

Figure 8.11 shows the CRLB matrix for the third flexible eigenmode. The frequency range for the simulation was 52-70 Hz. The parameters not affecting third flexible eigenmode can be prominently seen in Figure 8.11. The Young's modulus of the siderail ***p27*** has the highest CRLB value, followed by parameters ***p1*** (thickness of siderail), ***p26*** (Young's modulus of inner liner) and ***p22*** (Young's modulus of third cross member).

As seen in Figure 8.11, there are many parameters affecting third flexible eigenmode. However, thickness of the attachment bracket of third cross member (***p6***) and Young's modulus of fourth cross member (***p19***) have the lowest CRLB values.

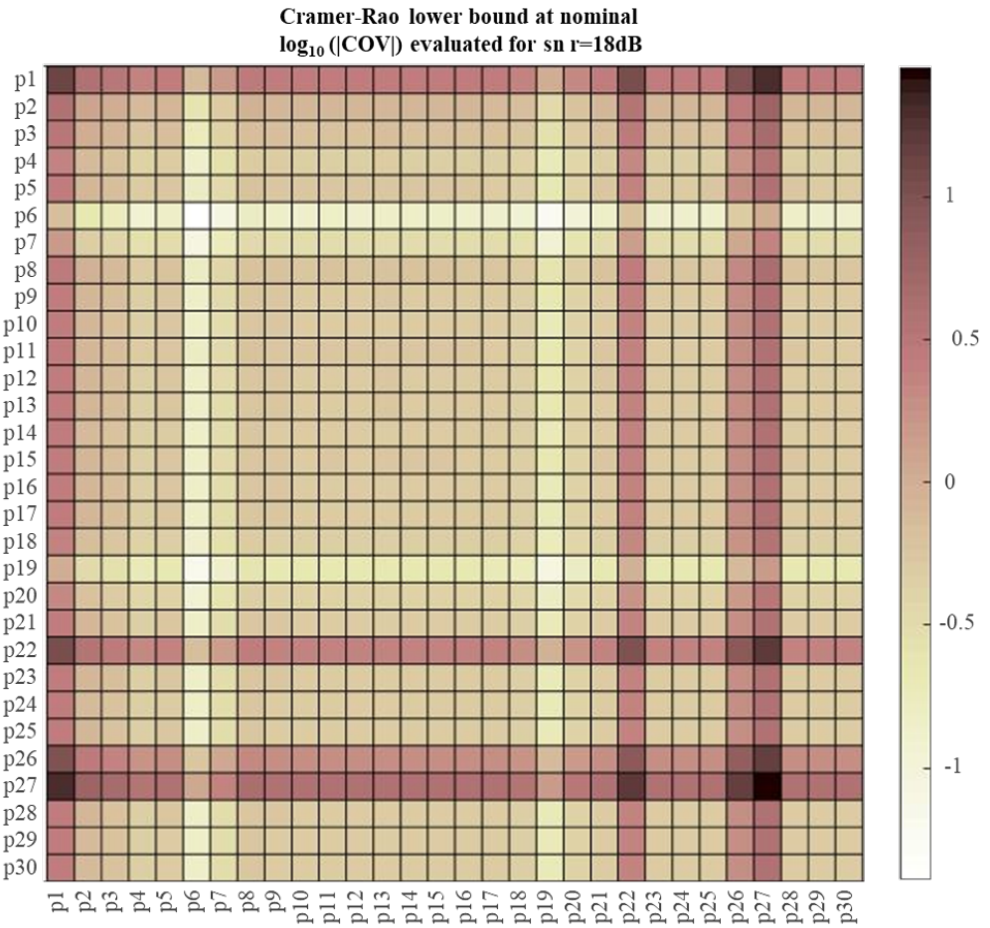

Figure 8.11: Parameter study for Third Flexible Mode (52-70 Hz)

Influence of Parameters on Fourth Flexible Eigenmode

Figure 8.12 shows the CRLB matrix for the fourth flexible eigenmode. The frequency range for the simulation was 90-102 Hz. As seen in the figure, red square in the center covers all the cross sectional parameters (***p9-p17***). These parameters do not have effect on fourth flexible eigenmode. The scale for this colour coded matrix have positive values. Therefore, it can be concluded that these parameters can be ignored for this eigenmode as their effect on the mode is almost negligible. The lowest CRLB values are seen for parameters ***p1***, ***p2*** corresponding to the thickness of siderail and inner liner respectively. The Young's modulus of the siderail and the inner liner (***p27*** and ***p26*** respectively) also have low CRLB values.

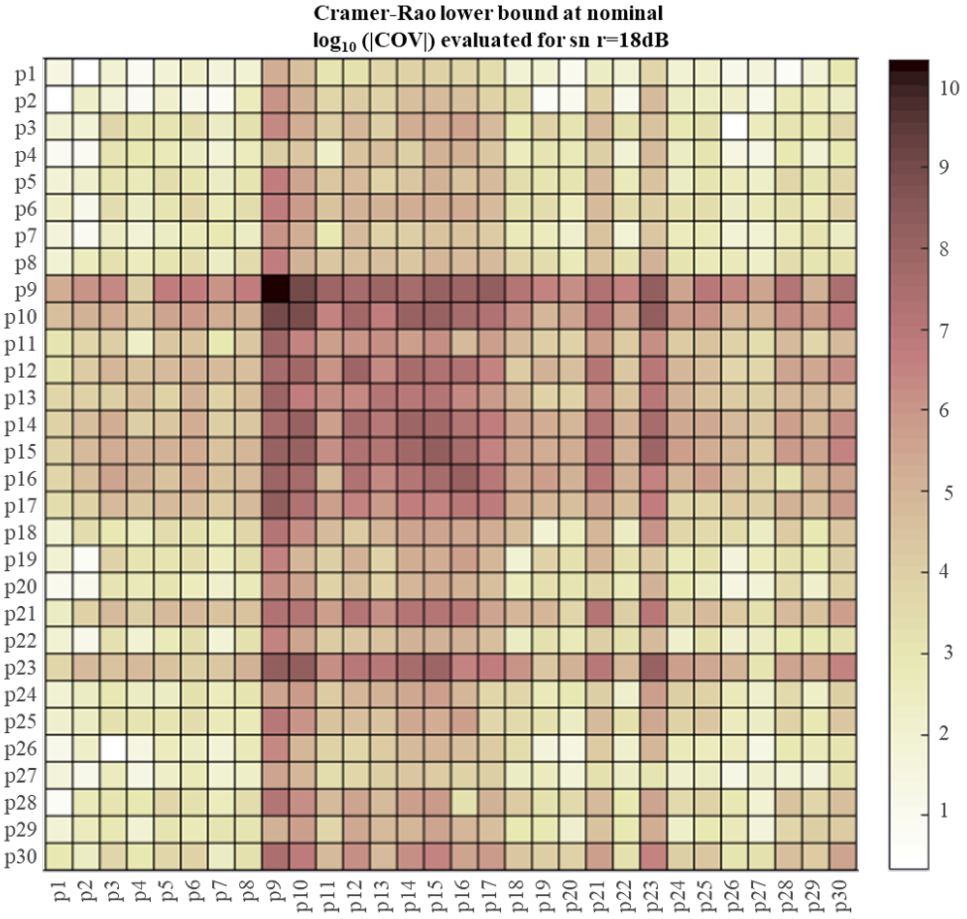

Figure 8.12: Parameter study for Fourth Flexible Mode (90-102 Hz)

Influence of Parameters on all Eigenmodes

The figure 8.13 shows CRLB matrix for the frequency range of 10-100 Hz. This matrix looks similar to that of the fourth flexible eigenmode. This can be attributed to accelerance data, which is biased towards higher frequency. The parameter ***p9*** has no effect on the eigenmodes of the structure. The CRLB values of all the cross section parameters (***p9-p17***) is higher than 8 (See scale on Figure 8.13). This shows

that these parameters have negligible effect on the eigenmodes.

The thickness parameters (***p1-p8***) have low CRLB values. These parameters affect the eigenmodes considerably. The lowest CRLB values can be seen for parameters ***p1*** (thickness of siderail) and ***p2*** (thickness of inner liner) in the thickness set. The Young's modulus values (***p18-p30***) are also low. The lowest CRLB values in the Young's modulus set are those of the siderail (***p27***) and the inner liner (***p26***).

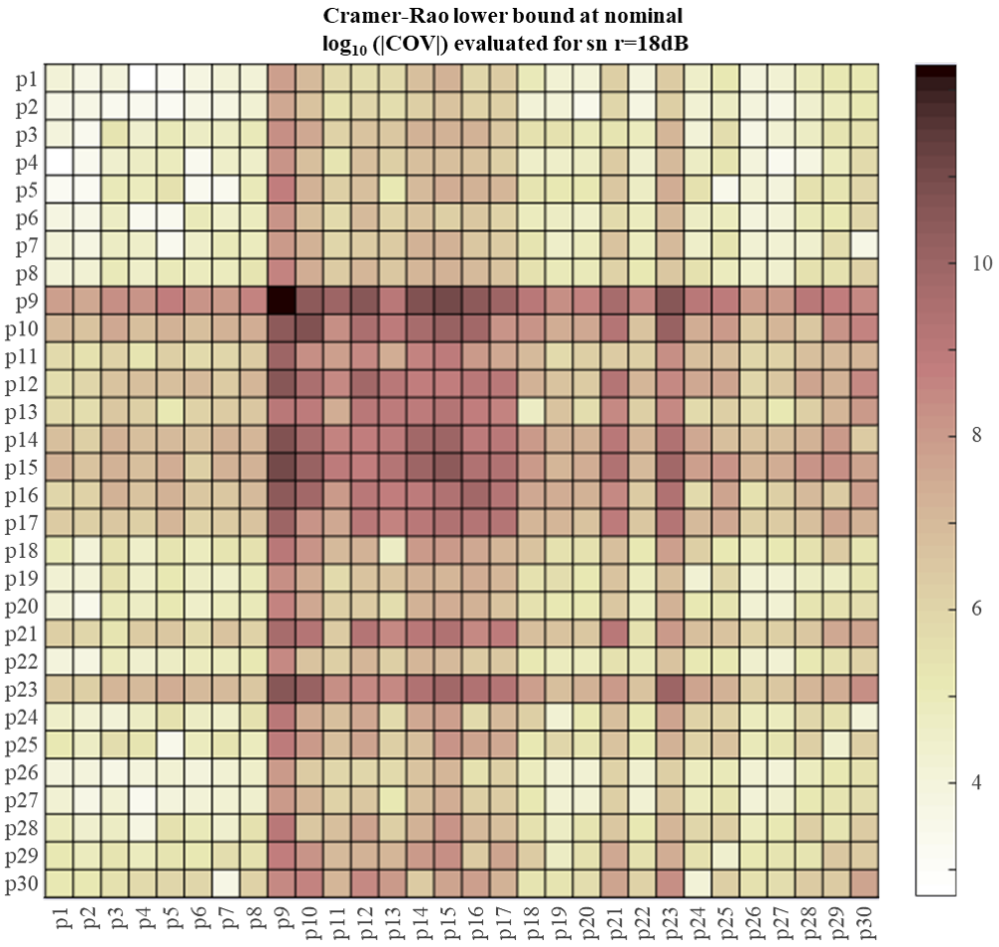

Figure 8.13: Parameter study for frequency range of 10-100 Hz

It can be concluded that parameters influencing the modal behaviour for the frequency range of 10-100 Hz are the thicknesses and the Young's modulus of all the components. Among these parameters, thickness and Young's modulus of the siderail and the inner liner affect the modal behaviour the most. It is to be noted that the scale used in Figure 8.9, Figure 8.10, Figure 8.11, Figure 8.12 and Figure 8.13 are different with different highest and lowest CRLB values for each matrix.

8.2.3 Modal Assurance Criterion (MAC) Plots

The Modal Assurance Criterion (MAC) is a statistical indicator which is used in this study to pair mode shapes from the analytical and test data (See Section 2.6.4 for more information on MAC). In this section, the MAC plots obtained from FEMCali simulations are discussed. Two FEMCali simulations with different frequency ranges were executed. The results are discussed below.

MAC Plots from FEMCali Simulation (10-100 Hz)

From Figure 8.14 and Figure 8.15, it can be observed that the MAC values of first three flexible eigenmodes are high compared to the fourth flexible eigenmode in both pre-calibration and post-calibration models. Furthermore, The fourth flexible eigenmode is not captured in the post-calibration plot as it goes out of the frequency range (10-100 Hz). So, another FEMCali simulation is executed using 10-102 Hz frequency range.

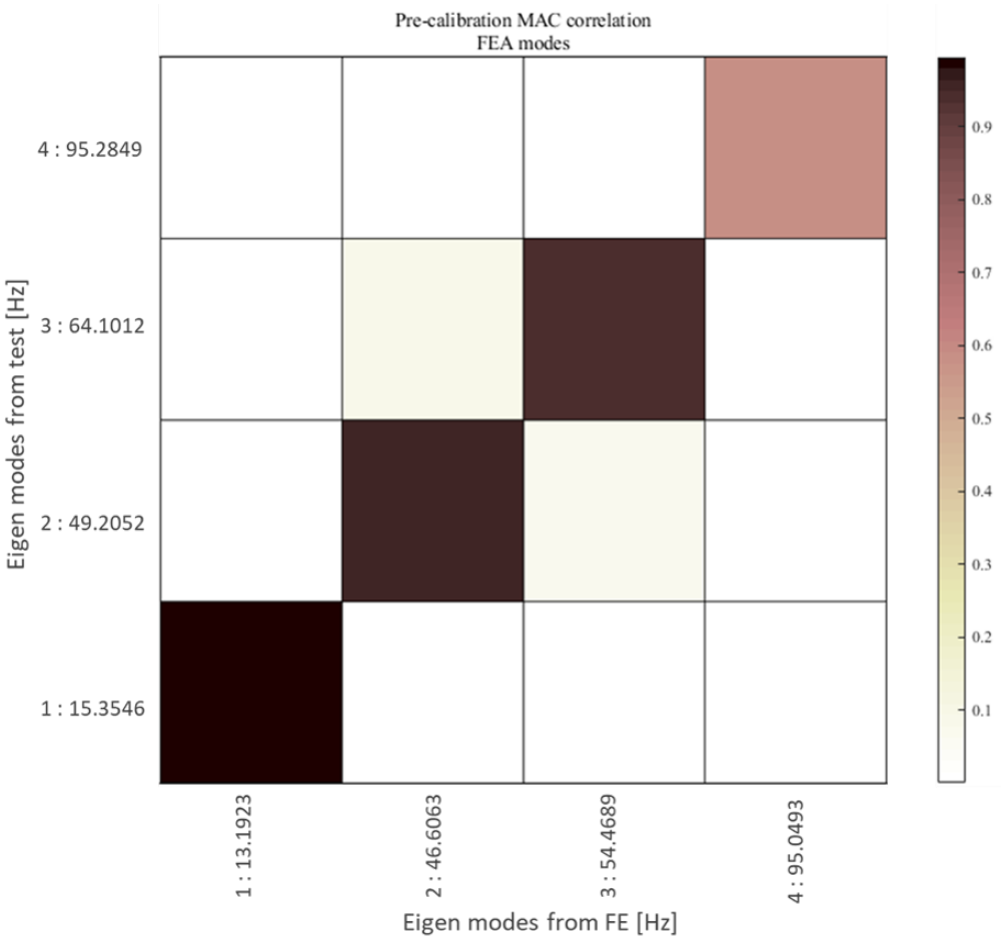

Figure 8.14: MAC Plot for Pre-calibration Model

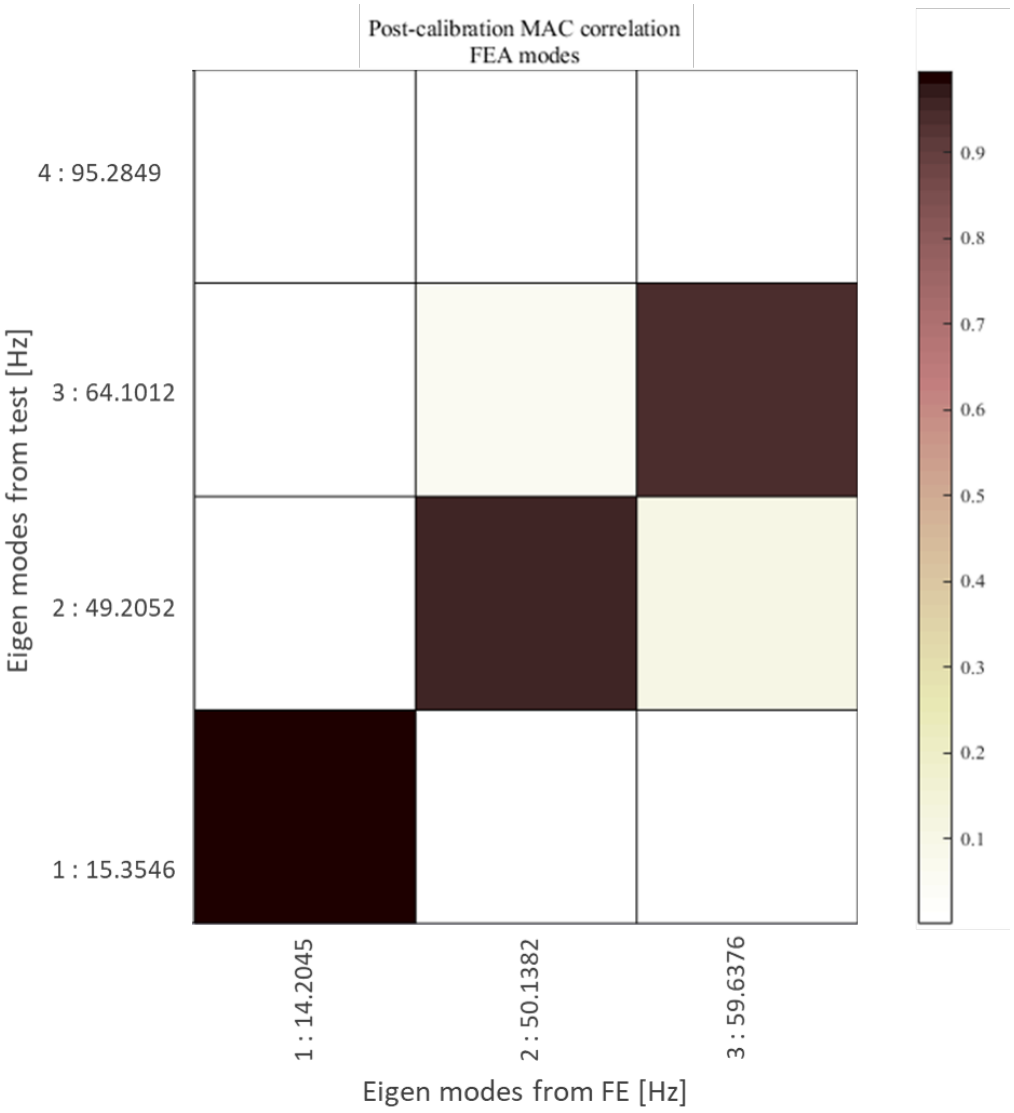

Figure 8.15: MAC Plot for Post-calibration Model

MAC Plots from FEMCali Simulation (10-102 Hz)

From Figure 8.16 and Figure 8.17, it can be observed that the MAC values of first three flexible eigenmodes are high compared to the fourth flexible eigenmode. The fourth flexible eigenmode is captured in the post-calibration plot by changing the frequency range to 10-102 Hz. The MAC values for each mode can be seen in Table 8.2. The MAC value of the fourth mode increases from 0.584 to 0.701 after calibration, but the correlation is not good. This result is consistent with the results from the eigenmode shape plots for fourth flexible eigemode (See Figure 8.8).

Table 8.2: MAC values for Pre-calibration and Post-calibration Models (10-102 Hz)

Flexible Modes	Pre-calibration MAC	Post-calibration MAC
1	0.994	0.995
2	0.958	0.961
3	0.941	0.950
4	0.584	0.701

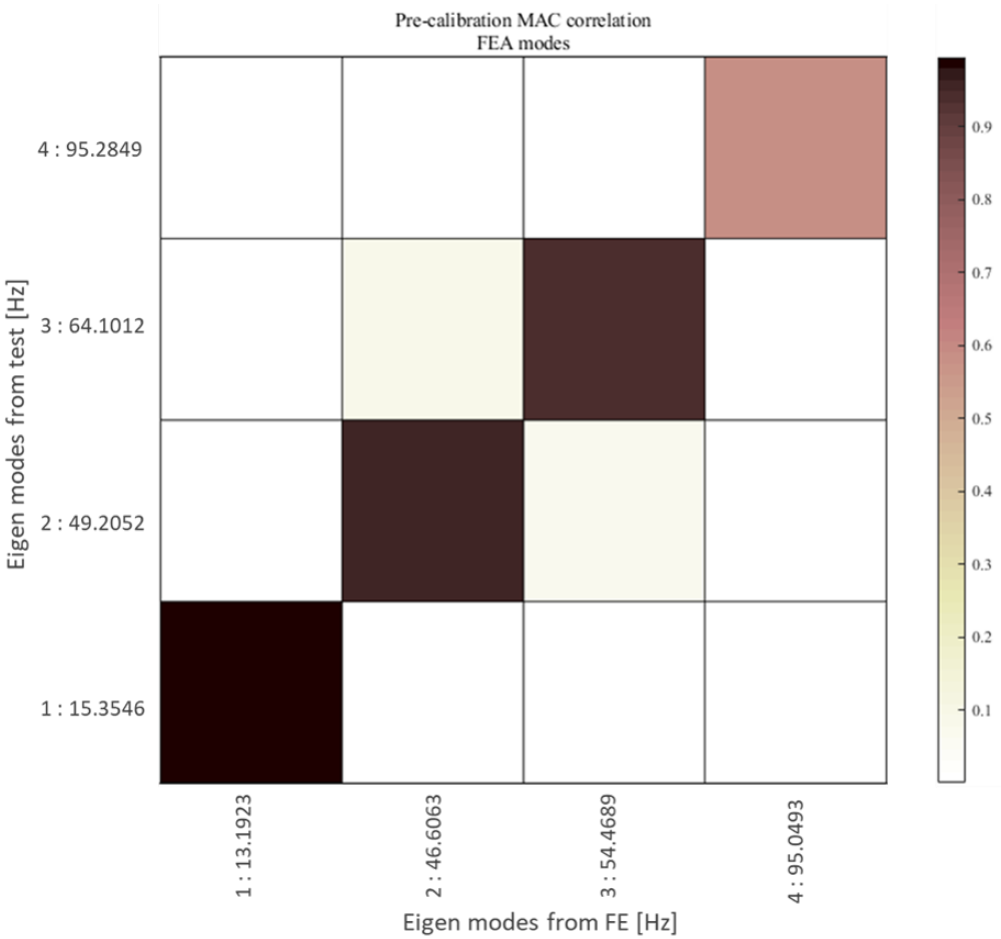

Figure 8.16: MAC Plot for Pre-calibration Model

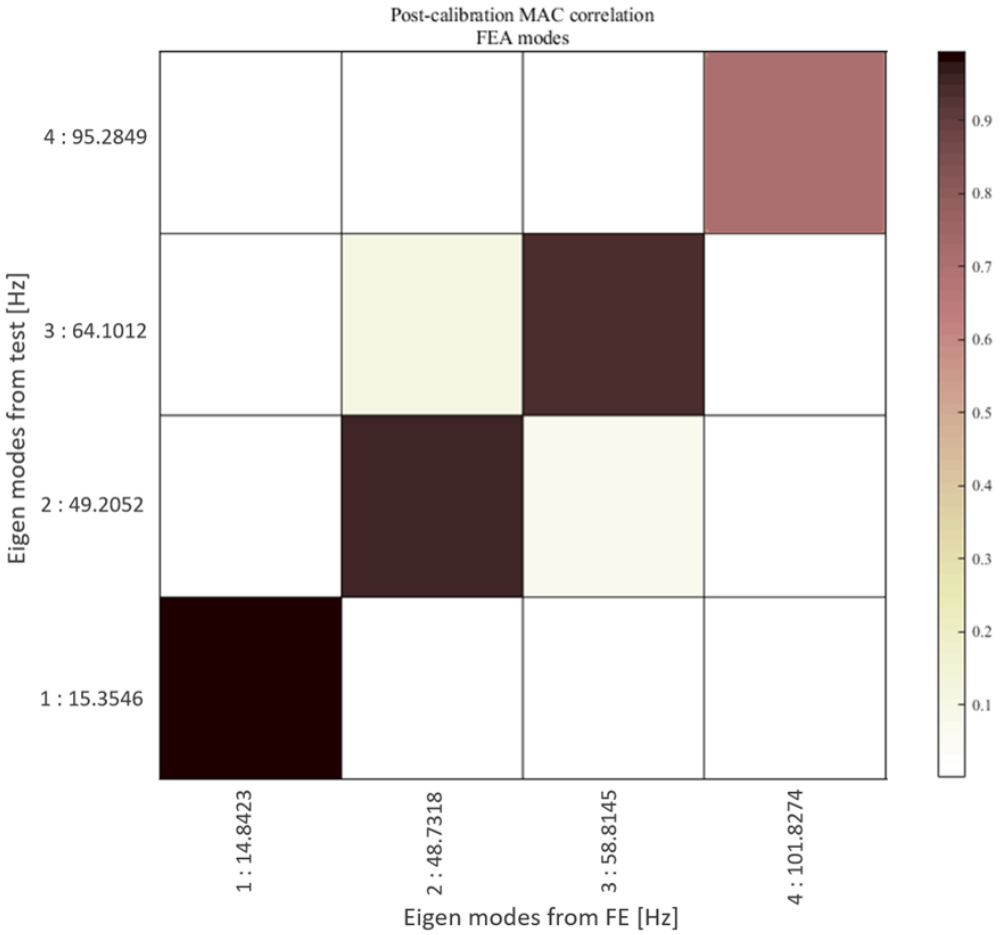

Figure 8.17: MAC Plot for Post-calibration Model

Table 8.3: Eigenfrequency from EMA and FEA

Flexible Modes	**EMA Modal Frequency [Hz]**	**Pre-calibration Modal Frequency [Hz]**	**Post-calibration Modal Frequency [Hz]**
1	15.35	13,19	14,84
2	49.20	46,60	48,73
3	64.10	54,47	58,81
4	95.28	95,05	101,83

Table 8.3 shows a comparison of the eigenfrequency values from experimental modal analysis and finite element analysis (for finite element models before and after calibration). This can be observed in the MAC plots as well (See Figure 8.14, Figure 8.15, Figure 8.16 and Figure 8.17).

8.2.4 Influence of Placement of Third Cross Member

The pre-calibration model was modified to study the influence of position of third cross member on the dynamic behaviour of the system. The position of the third cross-member was changed along X-axis (see Figure 8.18) to study the sensitivity of the modal behaviour of the system to its position. L is the distance of the third cross memeber from the front end of the frame.

The results of finite element modal analysis is tabulated in Table 8.4. It is observed that the eigenfrequency of first and second flexible eigenmode is not affected much by changing the position of third cross-member (See Figure 8.19). The third and fourth flexible modes are more sensitive to the change in position. The fourth flexible eigenmode gets affected more because changing the position of third cross member changes the eigenmode shape.

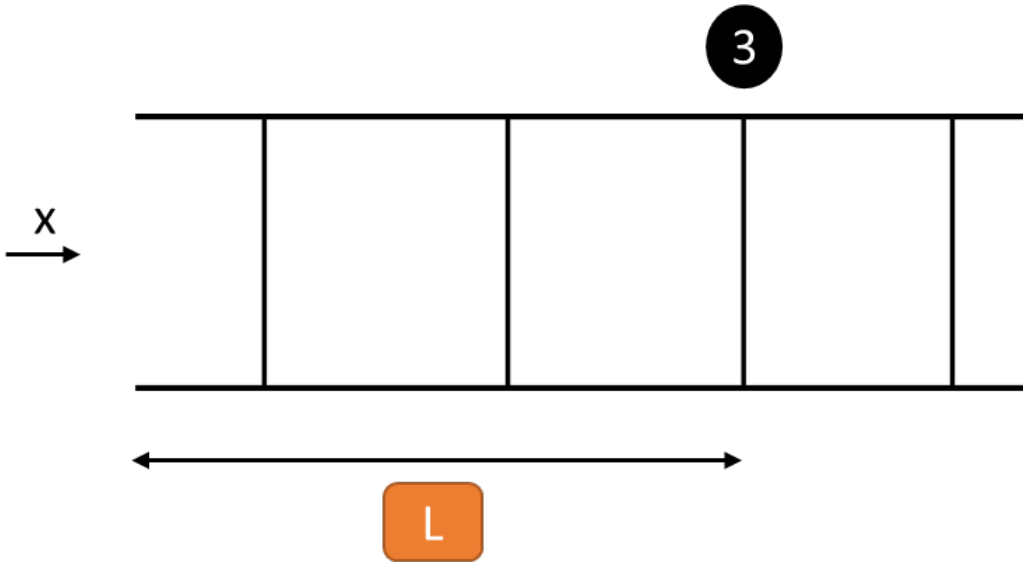

Figure 8.18: Position Variation of 3rd cross-member

Table 8.4: Iteration showing the effects of cross member 3 position

Iterations	Distance from start of the frame (mm)	Mode 1	Mode 2	Mode 3	Mode 4
1	2475	13.75	46.36	58.70	98.57
2	2425	13.49	47.06	57.18	99.83
3	2375	13.38	47.74	55.85	99.16
4	2325	13.32	48.32	54.77	96.48
5	2275	13.27	48.69	54.06	93.46

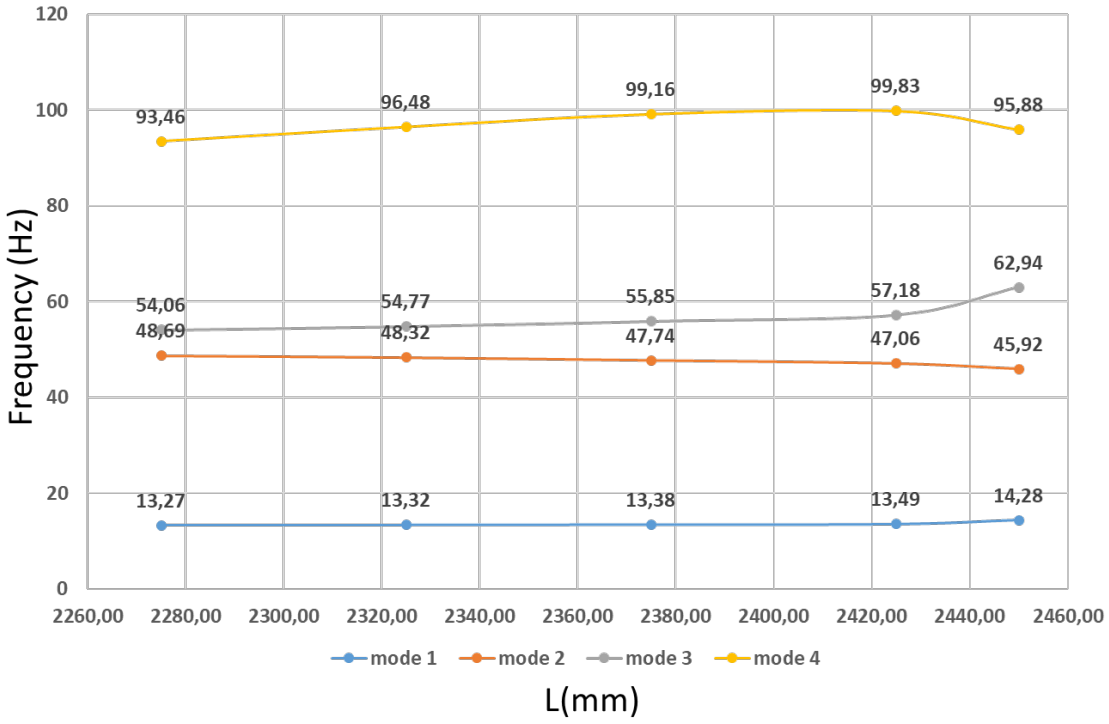

Figure 8.19: Influence of Position of Third Cross member on the Eigenfrequencies

9
Conclusions

Model calibration involves data from experimental testing and finite element model simulations. Therefore, it is extremely important that the experiments are conducted with great accuracy and care [11]. Hence, the success of the calibration will depend on the accuracy of measurements. The accuracy of the finite element model will also affect the final calibration result [11].

9.1 Rig Test

The measurements from the rig test were used to synthesize transfer function estimates as described in Section 5.1. It was observed that the transfer function estimates have poor quality (See Section 8.1).
The rig setup constrains the frame at locations where it is mounted to the rig. When input from only one actuator is provided, the other actuators act as supports for the frame. The force values are recorded using force transducers at the interface between the actuators and the frame. The force transducers, which are used in the experimental testing, recorded forces in only one direction. Since they do not record spurious forces in other directions, the net force recorded can be misleading. These discrepancies in force values can lead to errors while synthesizing the transfer function estimates.

9.2 Hammer Test

For hammer test, the correlation is good for first three flexible eigenmodes which is evident from various correlation indicators. There is mismatch of eigenvectors in fourth flexible eigenmode (MAC=0.70). When the frequency range is 10-100 Hz, the fourth flexible eigenmode is not captured in the given frequency range (see Figure 8.14 and Figure 8.15). The frequency range is changed to 10-102 Hz based on simulations to capture the fourth flexible eigenmode (see Figure 8.16 and Figure 8.17). Different parameters affecting the modal behaviour of the system were identified and model was calibrated based on these parameters (See Section 8.2.2 for results of parametrization).
The low MAC value for fourth flexible eigenmode can be attributed to eigenvectors at the rear end of the frame. It is observed in Figure 9.1 that the eigenvector from FEA points outward while the eigenvector from EMA points inward. So, there is an angle of approximately 180between these two eigenvectors resulting in low MAC values.

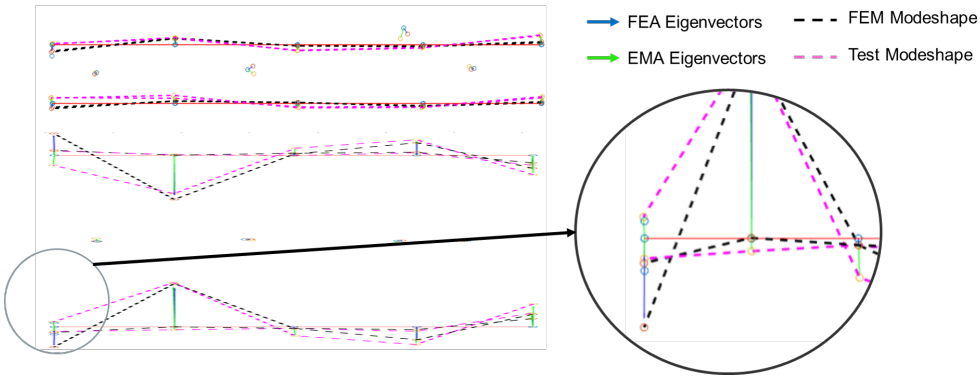

Figure 9.1: Mode-Shape Difference for 4th mode

The difference in eigenvectors of the fourth mode can be attributed to the test setup that is used during hammer test. The fixture for air-bellows and the cross-member attaching both the air-bellows (marked in red in Figure 9.2) gives additional rotational stiffness to the setup which is undesirable.

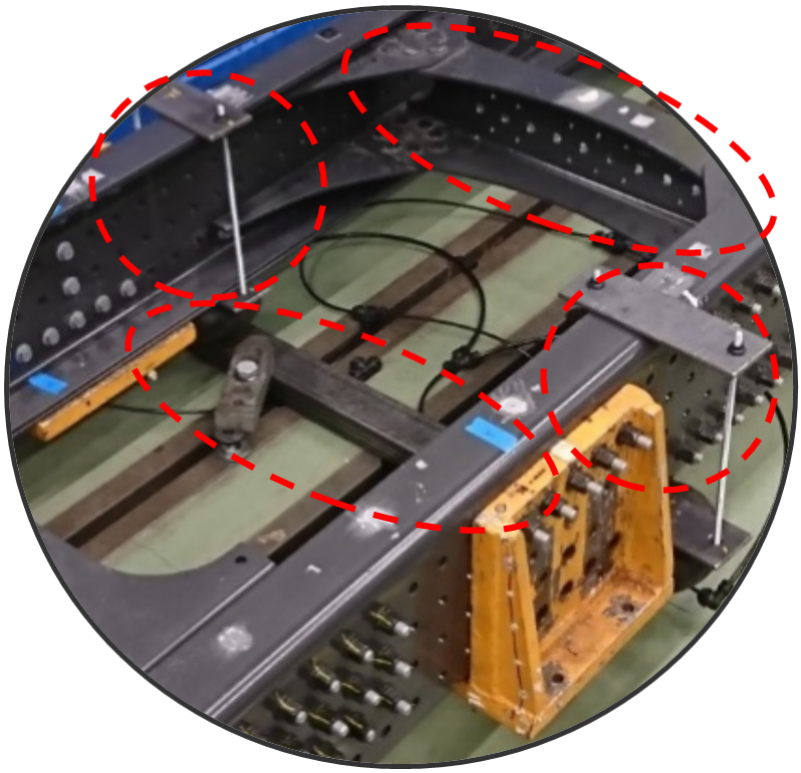

Figure 9.2: Fixture for Hammer Test

More work needs to be done for the displacement controlled rig to behave as a force controlled one. This allows good future scope for research in this field.

10
Future Work

FEMcali was used to calibrate the parameters in the Finite element model and with the current set of parameters the model cannot be improved further. However, several modifications in the test methodology can give us better experimental data. With better experimental data the transfer function estimates can be improved. The new data can be used in calibration to improve results.

The test methodology used in the rig test needs to be evaluated to get desirable transfer functions. Type of force input needs to be investigated further.The way in which force data is collected can be investigated further. In hammer test the test setup constrains the system and makes it more rotationally stiff along x direction. This needs to be investigated further.

Bibliography

[1] Abrahamsson, Thomas (2012). Calibration and validation of structural dynamics models. Chalmers – Applied Mechanics, Gothenburg.

[2] Berbyuk, V. (2014). Structural dynamics control. Chalmers – Applied Mechanics, Gothenburg. 2nd ed.

[3] Craig, R., Kurdila, A. (2006). Fundamentals of structural dynamics. Hoboken: John Wiley Sons.

[4] Gibanica, Mladen, Abrahamsson, Thomas J. S. & Allemang, Randall J. Residual States for Modal Models Identified from Accelerance Data,p195-p206. Topics in Modal Analysis & Testing, Volume 9, 2019. Springer International Publishing.

[5] Gumussoy, S., Ozdemir, A., McKelvey, T., Ljung, L., Gibanica, M., & Singh, R. (2018). Improving Linear State-Space Models with Additional Iterations. IFAC-Papersonline, 51(15), 341-346. doi: 10.1016/j.ifacol.2018.09.158

[6] Gibanica, M., Abrahamsson, T., & McKelvey, T. (2018). Physically motivated rank constraint on direct throughput of state-space models. IFAC-Papersonline, 51(15), 329-334. doi: 10.1016/j.ifacol.2018.09.156

[7] Vold, Håvard, John Crowley, and G. Thomas Rocklin. "New Ways of Estimating Frequency Response Functions." Sound and Vibration. Vol. 18, November 1984, pp. 34–38.

[8] Ewins, D. J. (2000). Modal testing: Theory, Practice and Application. Research Studies Press Ltd, Baldock, England.

[9] Phillips, A., Allemang, R. (2003). An overview of MIMO-FRF excitation/averaging/processing techniques. Journal Of Sound And Vibration, 262(3), 651-675. doi: 10.1016/s0022-460x(03)00116-0

[10] Abrahamsson, T. and Kammer, D. (2015). Finite element model calibration using frequency responses with damping equalization. Mechanical Systems and Signal Processing, 62-63, pp.218-234.

[11] Granado, I.E. (2015). Model calibration of a vehicle tailgate using frequency response functions.

[12] Singular value decomposition - MATLAB svd- MathWorks Nordic. (2019). Retrieved 7 October 2019, from https://se.mathworks.com/help/matlab/ref/double.svd.html;

[13] Transfer function estimate - MATLAB tfestimate- MathWorks Nordic. (2019). Retrieved 7 October 2019, from https://se.mathworks.com/help/signal/ref/tfestimate.html

[14] Ågren, T. Experimental modal analysis of chassis component shaker rig frame for correlation with durability virtual FE rig model. Volvo GTT Engineering Report ER-668568.
[15] Mori, M. and Öijer, F. 7 poster chassis shaker simulation
[16] MSC Software Corporation. MSC Nastran 2013.1.1 Quick Reference Guide. URL https://simcompanion.mscsoftware.com/infocenter/index.
[17] Pastor, M., Binda, M., Harčarik, T. (2012). Modal Assurance Criterion. Procedia Engineering, 48, 543-548. doi: 10.1016/j.proeng.2012.09.551
[18] Mathworks.com. (2019). Load rpc files (of type BINARY SHORTINTEGER TIMEHISTORY) - File Exchange - MATLAB Central. [online] Available at: https://www.mathworks.com/matlabcentral/fileexchange/9357-load-rpc-files-of-type-binary-shortinteger-timehistory [Accessed 29 Oct. 2019].

A

Appendix A: MATLAB Code

A.1 Hammer test

```
0   %% ------------------------Hammer Test-----------------------------------
1
2   close all;
3   clear all;
4   clc;
5
6   %----------------------Load frequency response functions-----------------
7
8   Y_excitation_pos8_14            = load('Y_excitation_pos8_14.mat');
9   Y_excitation_pos1_7             = load('Y_excitation_pos1_7.mat');
10  Y_excitation_pos15_20_xmemb1    = load('Y_excitation_pos15_20_xmemb1.mat');
11  Y_excitation_xmemb2_4           = load('Y_excitation_xmemb2_4.mat');
12  Z_excitation_pos1_7             = load('Z_excitation_pos1_7.mat');
13  Z_excitation_pos8_14            = load('Z_excitation_pos8_14.mat');
14  Z_excitation_pos15_20_xmemb1    = load('Z_excitation_pos15_20_xmemb1.mat');
15  Z_excitation_xmemb2_4           = load('Z_excitation_xmemb2_4.mat');
16
17  %--------------------------Extract FRF-----------------------------------
18
19  frf1 = Y_excitation_pos1_7.FRF.y_values.values;
20  frf2 = Y_excitation_pos8_14.FRF.y_values.values;
21  frf3 = Y_excitation_pos15_20_xmemb1.FRF.y_values.values;
22  frf4 = Y_excitation_xmemb2_4.FRF.y_values.values;
23  frf5 = Z_excitation_pos1_7.FRF.y_values.values;
24  frf6 = Z_excitation_pos8_14.FRF.y_values.values;
25  frf7 = Z_excitation_pos15_20_xmemb1.FRF.y_values.values;
26  frf8 = Z_excitation_xmemb2_4.FRF.y_values.values;
27
28  %-------------------------Synthesize MIMO--------------------------------
29
30  cut = 3000;                             %change to change # of data points
31
32  frf_y = [frf1(1:cut,:),frf2(1:cut,:),frf3(1:cut,:),frf4(1:cut,:)];
33  frf_z = [frf5(1:cut,:),frf6(1:cut,:),frf7(1:cut,:),frf8(1:cut,:)];
```

```
34  ResponseData(:,:,1) = frf_y;
35  ResponseData(:,:,2) = frf_z;
36  ResponseData        = permute(ResponseData,[2 3 1]);
37
38  %-----------------------------Frequency-----------------------------------
39
40  Frequency    = [];
41  Frequency(1) = 0;
42  for i=1:(Y_excitation_pos15_20_xmemb1.FRF.x_values.number_of_values-1)
43      Frequency(i+1) = Frequency(i)+0.05;% frequency increment
44  end
45  Frequency   = Frequency';
46
47  %------------------------Angular Frequency--------------------------------
48
49  aFrequency  = 2*pi*Frequency;                                        % w=2pif
50  aFrequency  = aFrequency(1:cut,:);
51
52  %-----------------------------Mobility data-------------------------------
53
54                    % use loop to obtain mobility data from accelerance data
55
56  % for i=1:length(aFrequency)
57  %     ResponseData(:,:,i) = ResponseData(:,:,i)/(1i*aFrequency(i));
58  % end
59
60  %----------------------------FRD/IDFRD Object-----------------------------
61
62  % FRDX        = frd(response,frequency,Ts)
63  % IDFRDX      = idfrd(ResponseData,aFrequency,Ts);
64  %
65  % For continuous-time models, Ts = 0. For discrete-time models Ts is a
66  % positive scalar representing the sampling period.
67  % This value is expressed in the unit specified by the TimeUnit property of
68  % the model. To denote a discrete-time model with unspecified sample time,
69  % set Ts = -1.
70
71  FRDX                = frd(ResponseData,aFrequency,0);
72  FRDX                = fselect(FRDX, 0*2*pi, 120*2*pi);                %FRDX
73  FRDX.ResponseData   = FRDX.ResponseData*1000;
74
75  %----------------------Residual Effect removal---------------------------
76
77  % function [FAr,SSAr] = removeresidualeffectsNEW(FA,beta,gamma)
78  % This funtion creates FAr and SSAr
79  % FAr  : Frequency response data
```

```
80  % SSAr : State Space model of
81  % FA   : FRD object
82  % beta : first pole
83  % gamma: second pole
84  [FAr, SSAr]  = removeresidualeffectsNEW(FRDX);
85
86  % split FAr to two parts
87  FAr1 = fselect(FAr, 10*2*pi, 60*2*pi);
88  FAr2 = fselect(FAr, 60*2*pi, 100*2*pi);
89  FAr3 = fselect(FAr, 100*2*pi, 120*2*pi);
90
91  %--------------------- State Space model estimation-----------------------
92
93  % systemIdentification;
94
95  Options                     = n4sidOptions;
96  Options.Display             = 'off';
97  Options.EstimateCovariance  = 0;
98  Options.EnforceStability    = 1;
99
100 ssX1           = n4sid(FAr1, (2)*2, 'Feedthrough', [true true], Options);
101 ssX2           = n4sid(FAr2, (2)*2, 'Feedthrough', [true true], Options);
102 ssX3           = n4sid(FAr3, (2)*2, 'Feedthrough', [true true], Options);
103
104 ssX4            = parallel(ssX2, ssX3);
105 ssX             = parallel(ssX1, ssX4);
106
107 % if exist('ssX.mat', 'file')==2
108 %  delete('ssX.mat');
109 % end
110 %
111 % save('ssX.mat');
112 % load('ssX.mat');
113
114 %-------------------------------------------------------------------------
115 % SSout = (SS1, SS2, FRD, J, constrain, mr);
116
117 % SSout:
118 % SS1: state space from FAr using n4sid subspace method
119 % SS2: state space SSAr
120 % FRD: original frequency response data
121 % J:any value (10-100)
122 % constrain and mr not required
123
124 SSout = parallelWithIterations(ssX, SSAr, FRDX, 100);
125 SSX     = SSout;                                                    %SSX
```

```
126 FRDsys  = frd(SSX, FRDX.Frequency);                                %FRD of SSX
127
128 load ('SS0.mat'); %FRD FEM
129 FRDfe  = frd(SS0, FRDX.Frequency);
130 %% phase and magnitude
131
132 k=1;
133 for I=1:2
134     for J=1:72
135         h=figure;
136         opt.hold=false;
137         magphase([I J; 0 inf],FRDX,opt);
138         opt.hold=true;
139         magphase([I J; 0 inf],FRDsys,opt);
140         title(['in' ,int2str(I),'out' ,int2str(J)]);
141 %         legend('FRDX','FRDfe','Location','NorhtEastOutside');
142 %         saveas(h,sprintf('FIG%d.png',k));
143         k=k+1;
144         pause;
145     end
146 end
147 %% comparison of FRD of test, state space, fem
148 %-------------------------------------------------------------------------
149 k=1;
150 for j=1:2                                                    %Input Channel
151 for i = 1:72                                                %Output Channel
152     h=figure;
153     semilogy(FRDX.Frequency/2/pi, abs(squeeze(FRDX.ResponseData(i,j,:))),...
154         'LineWidth',1);
155     hold on;
156     semilogy(FRDX.Frequency/2/pi, abs(squeeze(FRDsys.ResponseData(i,j,:))),...
157         'LineWidth',1);
158 %     semilogy(FRDX.Frequency/2/pi, abs(squeeze(FRDfe.ResponseData(i,j,:))),...
159 %     'LineWidth',1);
160     title(['In:',int2str(j),'Out:',int2str(i)]);
161     legend('FRD: EMA','State Space (n4sid)','Location', 'NorthEast');
162     print('-dpng','-r500');
163     saveas(h,sprintf('FIG%d.png',k));
164     hold off;
165     k=k+1;
166 end
167 end
168
169 %----------------------Eigenvalues and Eigenvectors----------------------
170
171 [Wn,zeta]   = damp(SSout.A); % damping and eigen frequency
```

```
172  fn          = Wn/(2*pi); % EIGEN FREQUENCIES
173
174  % EIGS   Find a few eigenvalues and eigenvectors of a matrix
175  % A must be square and should be large and sparse.
176  % [V,D] = EIGS(A) returns a diagonal matrix D of A's 6 largest magnitude
177  % eigenvalues and a matrix V whose columns are the corresponding
178  % eigenvectors.
179  % EIGS(A,K) return the K largest magnitude eigenvalues.
180
181  [V,eval]  = eigs(SSout.A,19);
182  % [V,eval]  = eig(ssX.A);
183
184  % EIGENVALUES
185  eigenValue        = diag(eval);
186
187  %------------------------------------------------------------------------
188
189  Z = imag(eigenValue)/(2*pi);
190
191  [eigenval,I,I2] = unique(abs(Z));
192
193  EigVec=V(:,I);
194
195  eigenVector=(SSout.C)*EigVec;
```

A.2 Rig test

```
0   close all; clear; clc;
1   %--------------------------- Data Extraction -----------------------------
2
3   currentDir = pwd; %uigetdir; %gets directory
4   timFiles = dir(fullfile(currentDir,'*.tim')); % gets all tim files in struct
5   data=[];
6   %
7   % for k = 1:length(timFiles)
8   %   baseFileName = timFiles(k).name;
9   %   [time_,data_,fformat] = rpc3(baseFileName);
10  %   data=[data;data_];
11  % end
12  clear data
13  for k = 1:length(timFiles)
14    baseFileName = timFiles(k).name;
15    [time_,data_,fformat] = rpc3(baseFileName);
16    data(:,:,k)=data_;
17  %   data=[data;data_];
18  end
```

```
19  data = mean(data,3);
20  DT = detrend(data);
21
22  time(1,1)=0;
23  for k = 2:length(data)
24      if rem(k,2)==0
25          time(k,1)=time(k-1,1)+0.020;
26      else
27          time(k,1)=time(k-1,1)+0.019;
28      end
29  end
30
31  DeltaT              = fformat.dt;
32  samplingFreq        = fformat.fsample;
33  nyquistFreq         = samplingFreq/2;
34  Ndata               = fformat.data_length;
35
36                                                  % Design bandpass filter
37  d1 = designfilt( 'bandpassfir',...
38                   'FilterOrder',20,...
39                   'CutoffFrequency1',0.5,...
40                   'CutoffFrequency2',30,...
41                   'SampleRate', samplingFreq);
42                                                              % Filtering
43  y  = filtfilt(d1,DT);                                    % filtered data
44  %-------------------------------------------------------------------------
45
46  Fo = y(:,8:14)*1e3;
47  % Fo2 = y2(:,8:14)*1e3;
48
49  % plotting power spectral density for the forces
50  [Pxx,F] = periodogram(Fo(:,5),[],1024,512);
51  close all,plot(F,Pxx); hold on;
52  % [Pxx2,F] = periodogram(Fo2(:,5),[],1024,512);
53  % plot(F,Pxx2);
54
55  legend('with filter','without filter');
56
57  %-------------------------------------------------------------------------
58  force_chls          = y(:,8:14)*1e3;                    % input channel
59  acceleration_chls   = y(:,15:21);                       % output channel
60
61  % fft
62  i=1;j=1;
63  [ f1,M1 ] =  CalcFFT( samplingFreq, force_chls(:,i));
64  [ f2,M2 ] =  CalcFFT( samplingFreq, acceleration_chls(:,j));
```

```
65  % segments_           = 30;       % number of segments for averaging
66
67  %----------------------------Synthesize MIMO-----------------------------
68
69  % create iddata object by specifying output signal, input signal and
70  % sampling time
71  % data=iddata(output,input,sample_time);
72  % estimate a transfer function with 2 poles and 1 zero
73  % sys=tfest(data,2,1);
74
75  % [txy,normFrequency] = tfestimate(force_chls,acceleration_chls,...
76  %                                                      30,[],'mimo');
77
78  FRF = [];
79  for k=1:size(acceleration_chls,2)
80  %[FRF(II,1,:),f] = tfestimate(u,y(:,II),ones(Ndata,1),0,Ndata,Fs);
81  %[FRF(:,:,k),f] = tfestimate(force_chls,acceleration_chls(:,k),...
82  %                  Ndata,0,Ndata,samplingFreq);
83   [FRF(:,:,k),f] = tfestimate(force_chls,acceleration_chls(:,k),...
84                    ones(Ndata,1),0,Ndata*30,samplingFreq);
85  end
86
87  ResponseData  = permute(FRF,[3 2 1]);
88
89  %  ResponseData  = permute(txy,[2 3 1]);             % MIMO transfer function
90  %  aFrequency    = pi*normFrequency*samplingFreq; % angular frequency [rad/s]
91  aFrequency    = 2*pi*f;
92
93  %----------------------------FRD/IDFRD Object----------------------------
94
95  % FRDX        = frd(response,frequency,Ts)
96  % IDFRDX      = idfrd(ResponseData,aFrequency,Ts);
97  %
98  % For continuous-time models, Ts = 0. For discrete-time models Ts is a
99  % positive scalar representing the sampling period.
100 % This value is expressed in the unit specified by the TimeUnit property of
101 % the model. To denote a discrete-time model with unspecified sample time,
102 % set Ts = -1.
103
104 FRDX        = frd(ResponseData,aFrequency,0);
105 FRDX        = fselect(FRDX, 0*2*pi, 120*2*pi);                    %FRDX
106
107 %----------------------Residual Effect removal---------------------------
108                                       % This funtion creates FAr and SSAr
109
110 % function [FAr,SSAr] = removeresidualeffectsNEW(FA,beta,gamma)
```

```
111  % FAr  : Frequency response data
112  % SSAr : State Space model of
113  % FA   : FRD object
114  % beta : first pole
115  % gamma: second pole
116  [FAr, SSAr]  = removeresidualeffectsNEW(FRDX);
117
118                                                    % split FAr to two parts
119  FAr1 = fselect(FAr, 10*2*pi, 60*2*pi);
120  FAr2 = fselect(FAr, 60*2*pi, 100*2*pi);
121  FAr3 = fselect(FAr, 100*2*pi, 120*2*pi);
122
123  %--------------------- State Space model estimation-----------------------
124
125  % systemIdentification;
126
127  Options                      = n4sidOptions;
128  Options.Display              = 'off';
129  Options.EstimateCovariance   = 0;
130  Options.EnforceStability     = 1;
131
132  ssX1  = n4sid(FAr1, (2)*2, 'Feedthrough', [true true true true true true...
133                                                          true], Options);
134  ssX2  = n4sid(FAr2, (2)*2, 'Feedthrough', [true true true true true true...
135                                                          true], Options);
136  ssX3  = n4sid(FAr3, (2)*2, 'Feedthrough', [true true true true true true...
137                                                          true], Options);
138
139  ssX4  = parallel(ssX2, ssX3);
140  ssX   = parallel(ssX1, ssX4);
141
142  % if exist('ssX.mat', 'file')==2
143  %  delete('ssX.mat');
144  % end
145  %
146  % save('ssX.mat');
147  % load('ssX.mat');
148
149  %-------------------------------------------------------------------------
150  % SSout = (SS1, SS2, FRD, J, constrain, mr);
151
152  % SSout:
153  % SS1: state space from FAr using n4sid subspace method
154  % SS2: state space SSAr
155  % FRD: original frequency response data
156  % J:any value (10-100)
```

```
157 % constrain and mr not required
158
159 SSout = parallelWithIterations(ssX, SSAr, FRDX, 100);
160 fsys  = frd(SSout, FRDX.Frequency);                          %FRD of SSout
161 SSX   = SSout;                                                    %SSX
162
163 fsysAr  = frd(SSAr, FRDX.Frequency);
164
165 %----------------------------------SS Plot---------------------------------
166
167 for j=1:7                                                    % Input Channel
168 for i = 1:7                                                 % Output Channel
169
170     semilogy(FRDX.Frequency/2/pi, abs(squeeze(FRDX.ResponseData(i,j,:))));
171     hold on;
172     semilogy(FRDX.Frequency/2/pi, abs(squeeze(fsys.ResponseData(i,j,:))));
173 %    semilogy(FAr1.Frequency/2/pi,abs(squeeze(FAr1.ResponseData(i,2,:))));
174 %    semilogy(FAr2.Frequency/2/pi,abs(squeeze(FAr2.ResponseData(i,2,:))));
175 %    semilogy(FRDX.Frequency/2/pi,abs(squeeze(fsysAr.ResponseData(i,2,:))));
176 %    legend('test','SSout','FAr1','FAr2','SSAr');
177     txt = {'Channel:',num2str(i),'x',num2str(j)};
178     text(4,0.5,txt)
179     pause;
180     hold off;
181 end
182 end
183
184 %----------------------Eigenvalues and Eigenvectors----------------------
185
186 [Wn,zeta]   = damp(SSout.A); % damping and eigen frequency
187 fn          = Wn/(2*pi); % EIGEN FREQUENCIES
188
189 % EIGS   Find a few eigenvalues and eigenvectors of a matrix
190 % A must be square and should be large and sparse.
191 % [V,D] = EIGS(A) returns a diagonal matrix D of A's 6 largest magnitude
192 % eigenvalues and a matrix V whose columns are the corresponding
193 % eigenvectors.
194 % EIGS(A,K) return the K largest magnitude eigenvalues.
195 [V,eval]  = eigs(SSout.A,19);
196 % [V,eval]  = eig(ssX.A);
197
198 % EIGENVALUES
199 eigenValue        = diag(eval);
200 %-------------------------------------------------------------------------
201
202 Z               = imag(eigenValue)/(2*pi);
```

```
203   [eigenval,I,I2] = unique(abs(Z));
204   EigVec          = V(:,I);
205   eigenVector     = (SSout.C)*EigVec;
```

A.3 Functions

The following functions are used in Hammer test and Rig test MATLAB codes. Refer A.1 and A.2.

A.3.1 CalcFFT

```
0   function [ f,M ] = CalcFFT( samplingFreq, y_response)
1
2       % fast fourier transform
3       L    = length(y_response);
4       NFFT = 2^nextpow2(L); % Next power of 2 from length of y
5       Y    = fft(y_response,NFFT)/L;
6       f    = samplingFreq/2*linspace(0,1,NFFT/2+1);
7       M    = 2*abs(Y(1:NFFT/2+1));
8   end
```

A.3.2 removeresidualeffectsNEW

Refer [4] for understanding the function.

A.3.3 parallelWithIterations

Refer [5] for understanding the function.

A.3.4 rpc3

This function is used to load data from rpc format. Refer [18].

B

Appendix B: MATLAB-NASTRAN Framework

A framework was developed during the study for data sharing between MATLAB and NASTRAN by modifying *mscnastran.m* file in the FEMCali package. This provides a framework for submitting FEA runs from MATLAB (Windows Environment) to NASTRAN (LINUX Environment) and importing FEA simulation data from NASTRAN into the MATLAB environment to access its programming and mathematical capabilities (See Figure B.1). Ganymed SSH-2 was used for this purpose (See line 20 in the code below). Ganymed SSH-2 for Java is a library which implements the SSH-2 protocol. It allows one to connect to SSH servers from MATLAB. It supports SSH sessions i.e. remote command execution and shell access. The SSH path edited into the code (See line 30 in the code below). The username and password also need to provided in the code (See line 36 and line 37 in the code below). The location of the NASTRAN file (See line 35 in the code below) also needs to be provided.

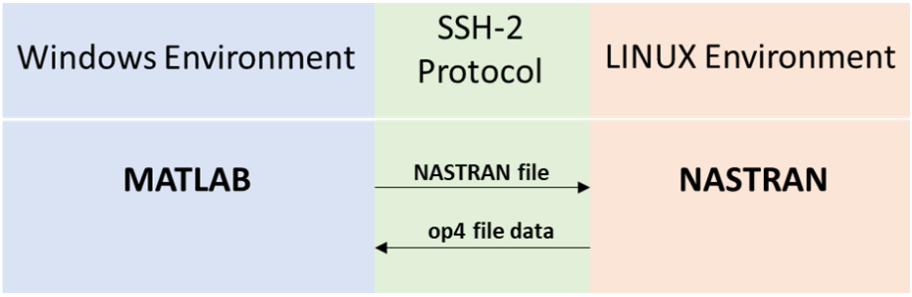

Figure B.1: MATLAB-NASTRAN Framework

B.1 Modified Code mscnastran.m

```
0   function mscnastran(infile,tidyuplevel,sdir,TempDir)
1   %MSCNastran. Helper function to execute MSC.Nastran quietly. It hangs until
2   %            job is done. The environmental variable MSCNAST_EXE needs to
3   %            be defined and point to the Nastran executable prior to a call
4   %            to this function.
5   %Inputs: infile   - Name of input file
6   %        tidyuplevel  - 0 No tidying
7   %                       1 delete=yes
```

```
8   %                       2 delete=yes scr=mini
9   %
10  %Call:   mscnastran(infile,verbose)
11
12  %Written: 2011-12-26: Thomas Abrahamsson
13
14  % ---default---
15  % infile=datfile
16  % tidyuplevel=2
17  % sdir='..'
18  global executionTimeData RunIndex
19  %%% Add Java SSH package to path
20  javaaddpath('C:\Users\a314495\Downloads\ganymed-ssh2\ganymed-ssh2-build250.jar')
21
22  if nargin<2,tidyuplevel=100;end
23  if nargin<3,sdir=[];end
24
25  if ~exist(infile,'file')
26      error(['The Nastran input file ' infile ' does not exist'])
27  end
28
29  paths.sambapath     = 'got-hpc-samba.got.volvo.net';  % samba path
30  paths.sshpath       = 'got-hpc10-i011.got.volvo.net'; % ssh path
31  paths.nasexepath    = '/caeapps/caesys/bin/';         % nastran path
32  paths.nasexecmd     = 'batch_nastran -Case';          % nastran submit
33  % infile              = 'sol103.dat';
34  % location_          = '';
35  execu               = ['cd /nobackup/ve_chassis_02/a314495/FEMcali/Runs/r19/', T
36  user                = 'a314495';
37  pass                = 'DeepSpace@18';
38  SSHch               = sshfrommatlab(user,paths.sshpath,pass);
39
40  % ------------------------------Windows----------------------------------
41  if ispc
42
43     exestr=[' old=no jid=' infile];
44    if ~isempty(sdir),exestr=[' sdirectory=' sdir exestr];end
45  %-------------------------------------------------------------------------
46  if tidyuplevel<=0
47      %     evalstr=['dos(''' Nast_exe exestr ''')'];
48
49      [~, result1]  =  sshfrommatlabissue(SSHch,'pwd')
50      % Execute
51      [~, result2]  =  sshfrommatlabissue(SSHch,execu)
52      tic
53      % check if simulation is done
```

```
54      [~, result3]  =  sshfrommatlabissue(SSHch,'q')
55
56      dotpos=strfind(infile,'.');
57      dotpos=dotpos(end);
58      op4file=[infile(2:dotpos-1) '.op4'];
59
60      pause on
61      disp('Waiting for the Nastran job...');
62      while exist(op4file,'file') == 0% exist(pchfile,'file')== 2 if file exists
63          pause(2);
64  %         disp('Waiting for the Nastran job...');
65      end
66
67      disp('Nastran job iteration done');
68
69      executionTimeData.nastranrun(RunIndex) = toc;
70      RunIndex                               = RunIndex+1;
71
72      pause off
73
74  %-------------------------------------------------------------------------
75
76  elseif tidyuplevel==1
77      evalstr=['dos(''' Nast_exe ' scr=mini notify=no ' exestr ''')'];
78
79  %-------------------------------------------------------------------------
80  elseif tidyuplevel==2
81
82      [~, result1]  =  sshfrommatlabissue(SSHch,'pwd')
83      % Execute
84      [~, result2]  =  sshfrommatlabissue(SSHch,execu)
85      tic
86      % check if simulation is done
87      [~, result3]  =  sshfrommatlabissue(SSHch,'q')
88
89      dotpos=strfind(infile,'.');
90      dotpos=dotpos(end);
91      op4file=[infile(1:dotpos-1) '.op4'];
92
93      pause on
94      disp('Waiting for the Nastran job...');
95      while exist(op4file,'file') == 0% exist(pchfile,'file')== 2 if file exists
96          pause(2);
97  %         disp('Waiting for the Nastran job...');
98      end
99      disp('Nastran job iteration done');
```

```
100
101       executionTimeData.nastranrun(RunIndex) = toc;
102       RunIndex                               = RunIndex+1;
103
104       pause off
105
106   %-------------------------------------------------------------------------
107   else
108       evalstr=['dos(''' Nast_exe ' scr=yes notify=no ' exestr ''')'];
109   end
110
111   %evalc(evalstr);
112
113
114
115   %-------------------------------------------------------------------------
116   elseif isunix
117       Nast_exe='module load untested nastran/2012-1;nast20121 batch=no ';
118       if tidyuplevel<=0
119           evalstr=['unix(''' Nast_exe ' old=no jid=' infile ''')'];
120       elseif tidyuplevel==1
121           %     evalstr=['unix(''' Nast_exe ' delete=yes old=no notify=no jid=' in
122           evalstr=['unix(''' Nast_exe ' old=no notify=no jid=' infile ''')'];
123       else
124           %     evalstr=['unix(''' Nast_exe ' delete=yes scr=yes old=no notify=no
125           evalstr=['unix(''' Nast_exe ' scr=yes old=no notify=no jid=' infile ''')
126       end
127       % %   evalc(evalstr);
128   end
```

C

Appendix C: FRF

The figures C.1 to C.20 show frf comparison plots of experimental test data obtained from rig test and state space model from the FE model. The figures C.1 to C.10 show the frf results from 72 output channels for input from channel 1. Figures C.11 to C.20 show the frf results from 72 channels for input from channel 2.

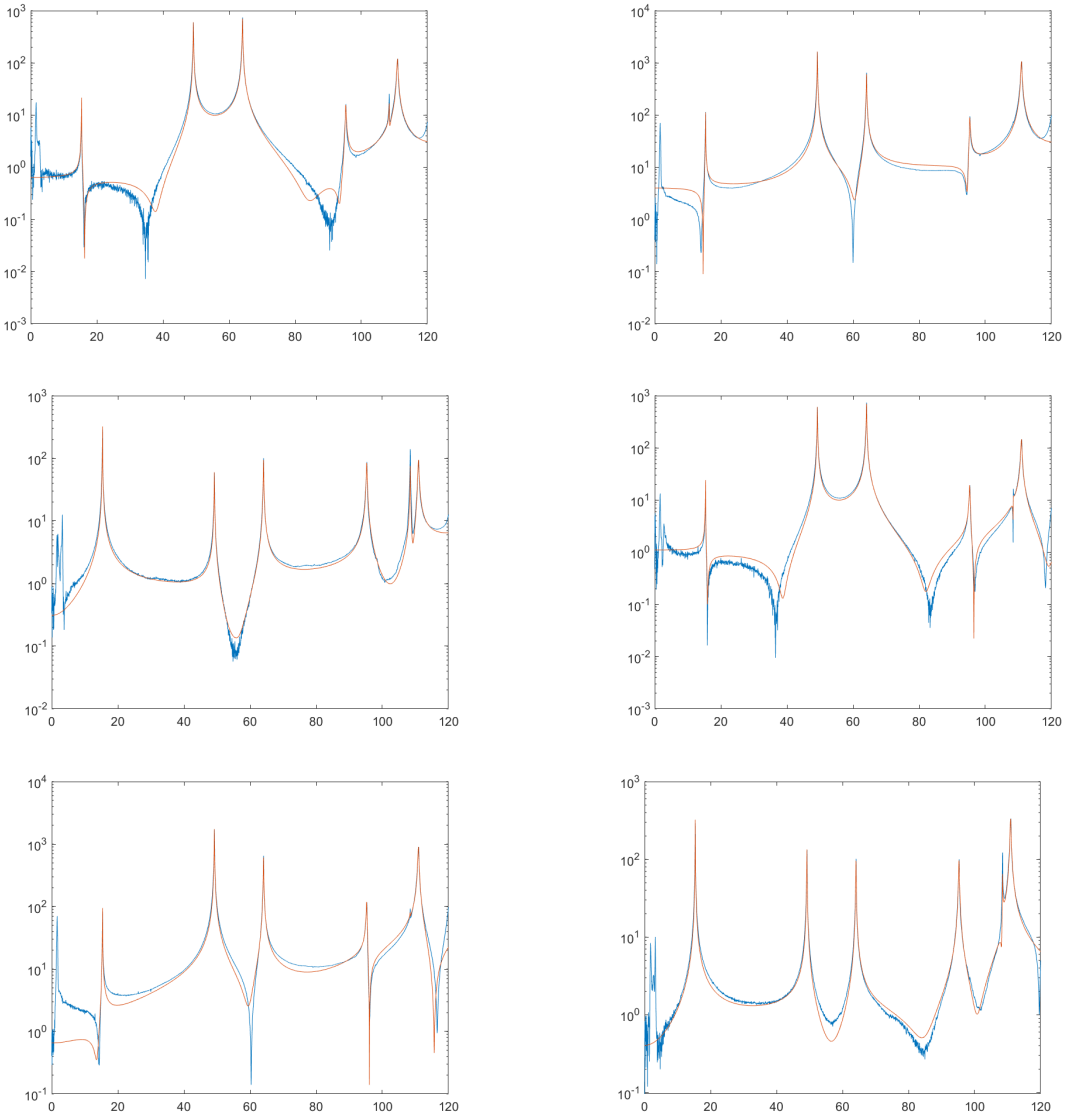

Figure C.1: FRF plots input channel 1, output channels 1 to 6

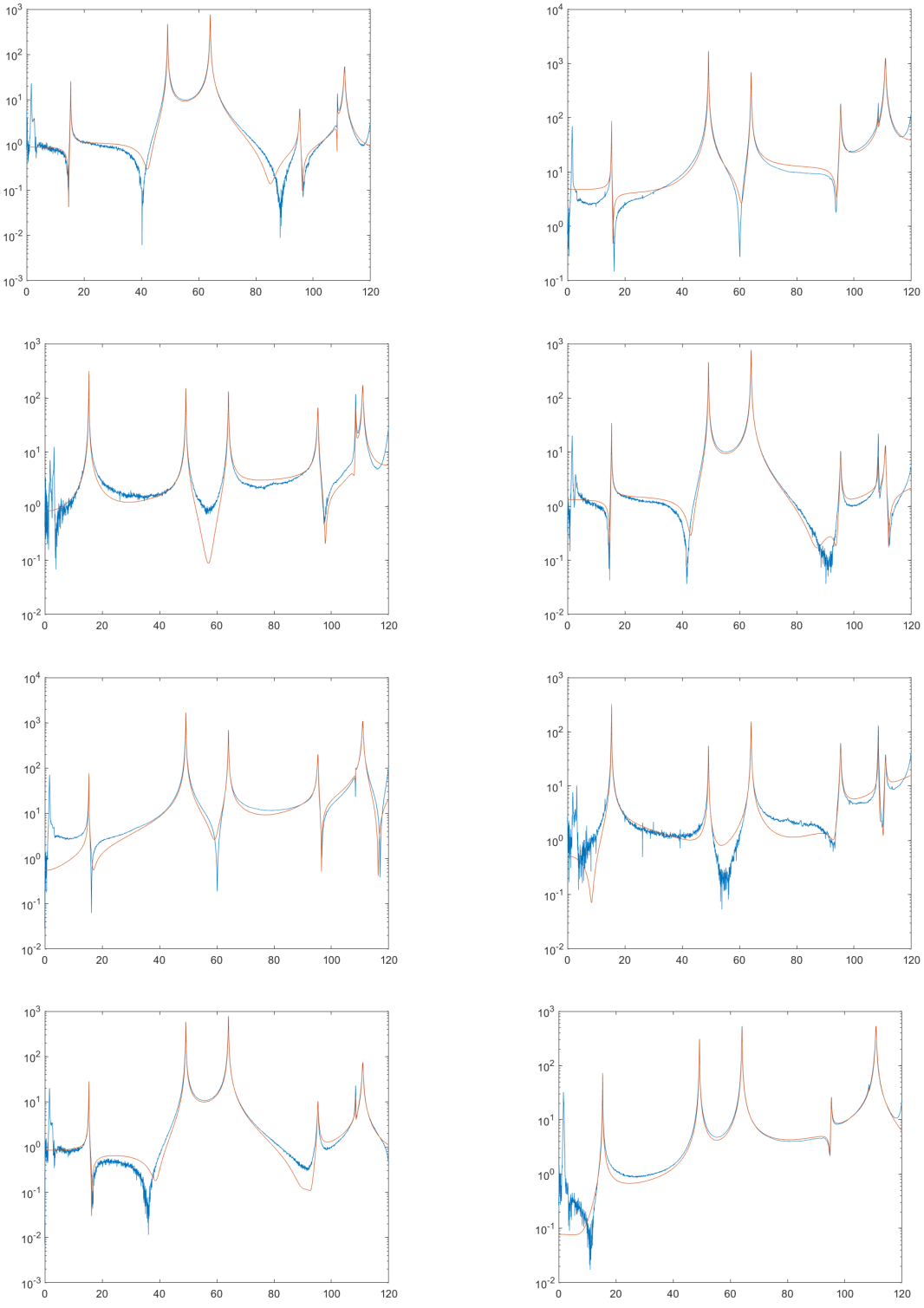

Figure C.2: FRF plots input channel 1, output channels 7 to 14

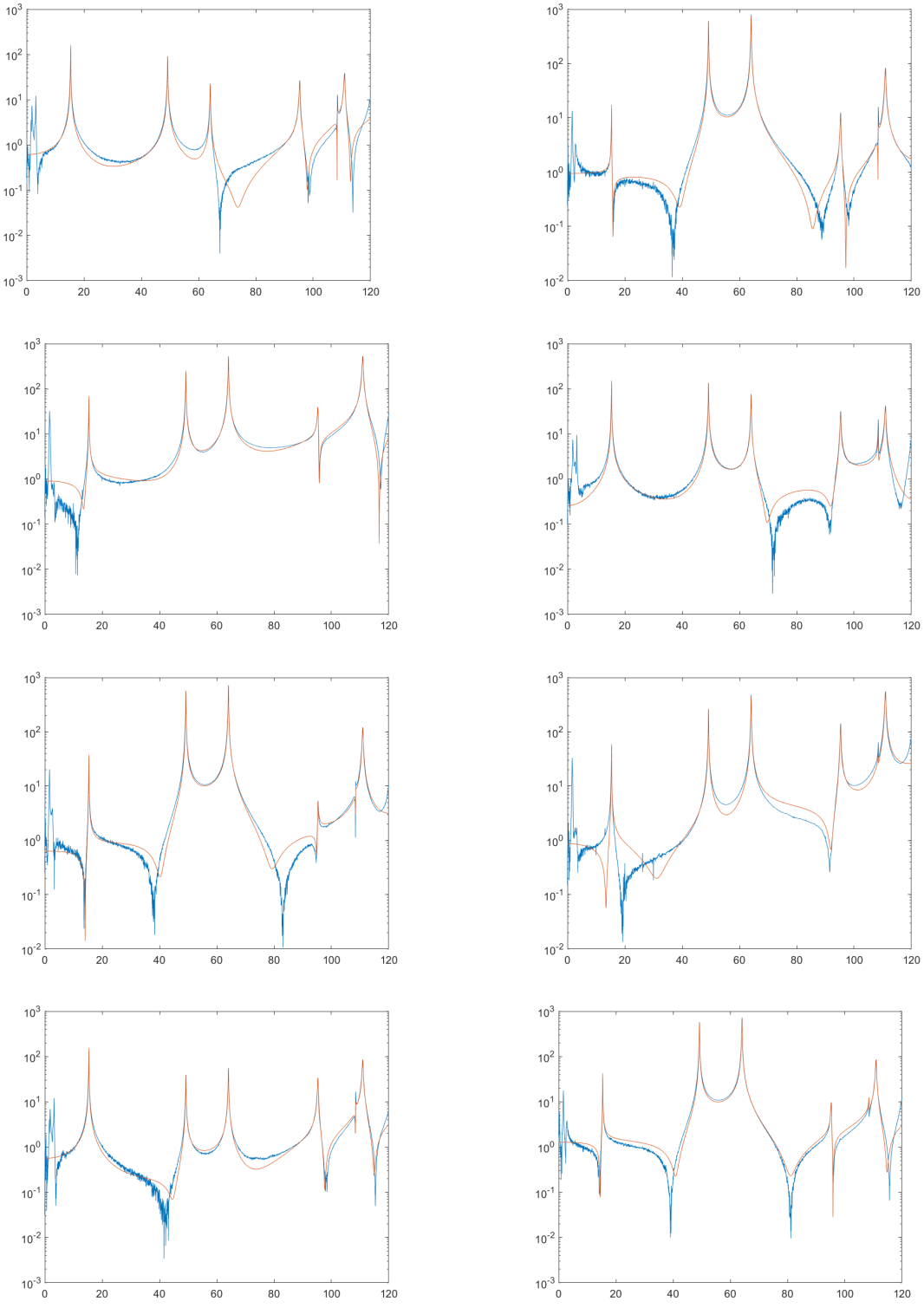

Figure C.3: FRF plots input channel 1, output channels 15 to 22

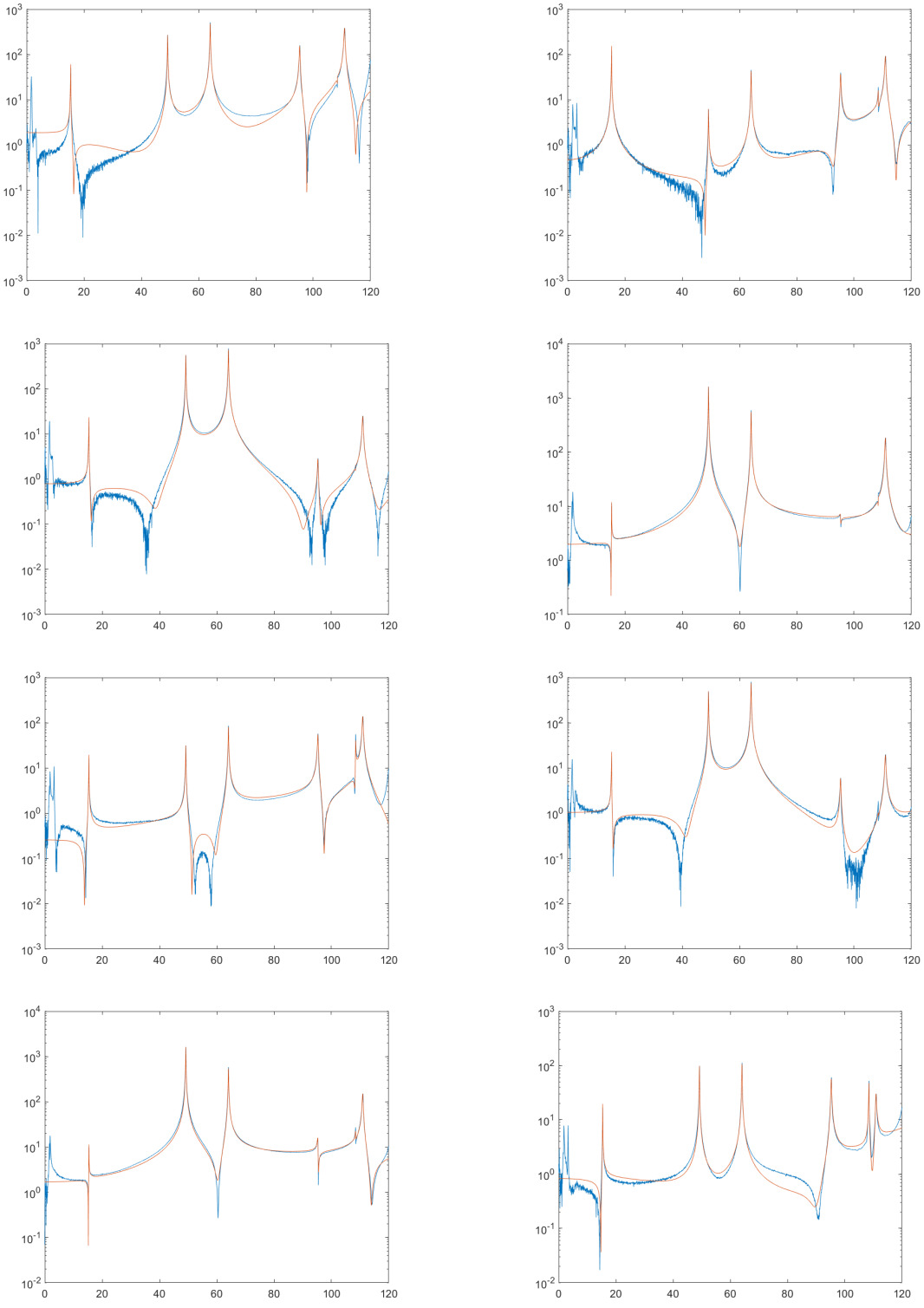

Figure C.4: FRF plots input channel 1, output channels 23 to 30

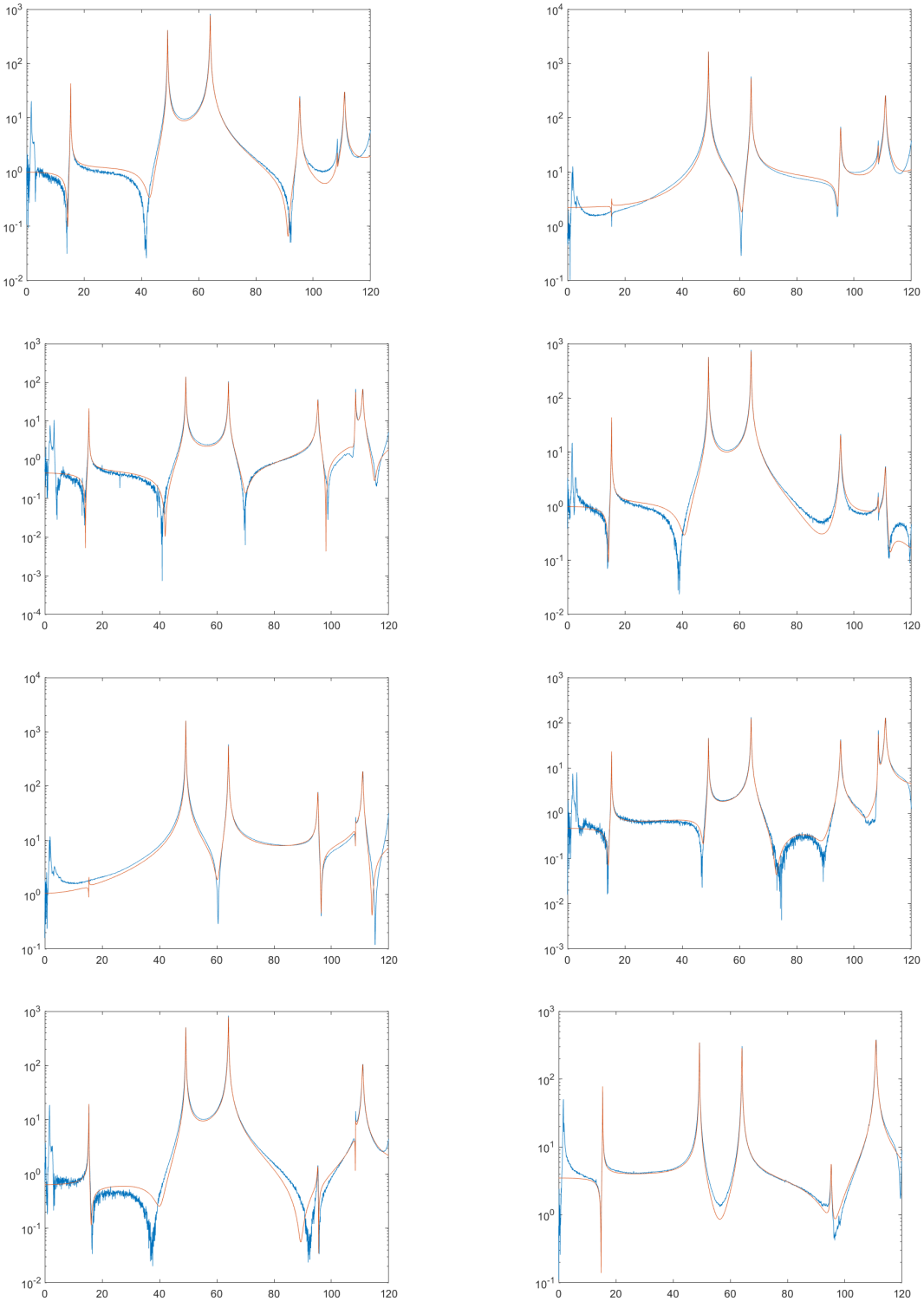

Figure C.5: FRF plots input channel 1, output channels 31 to 38

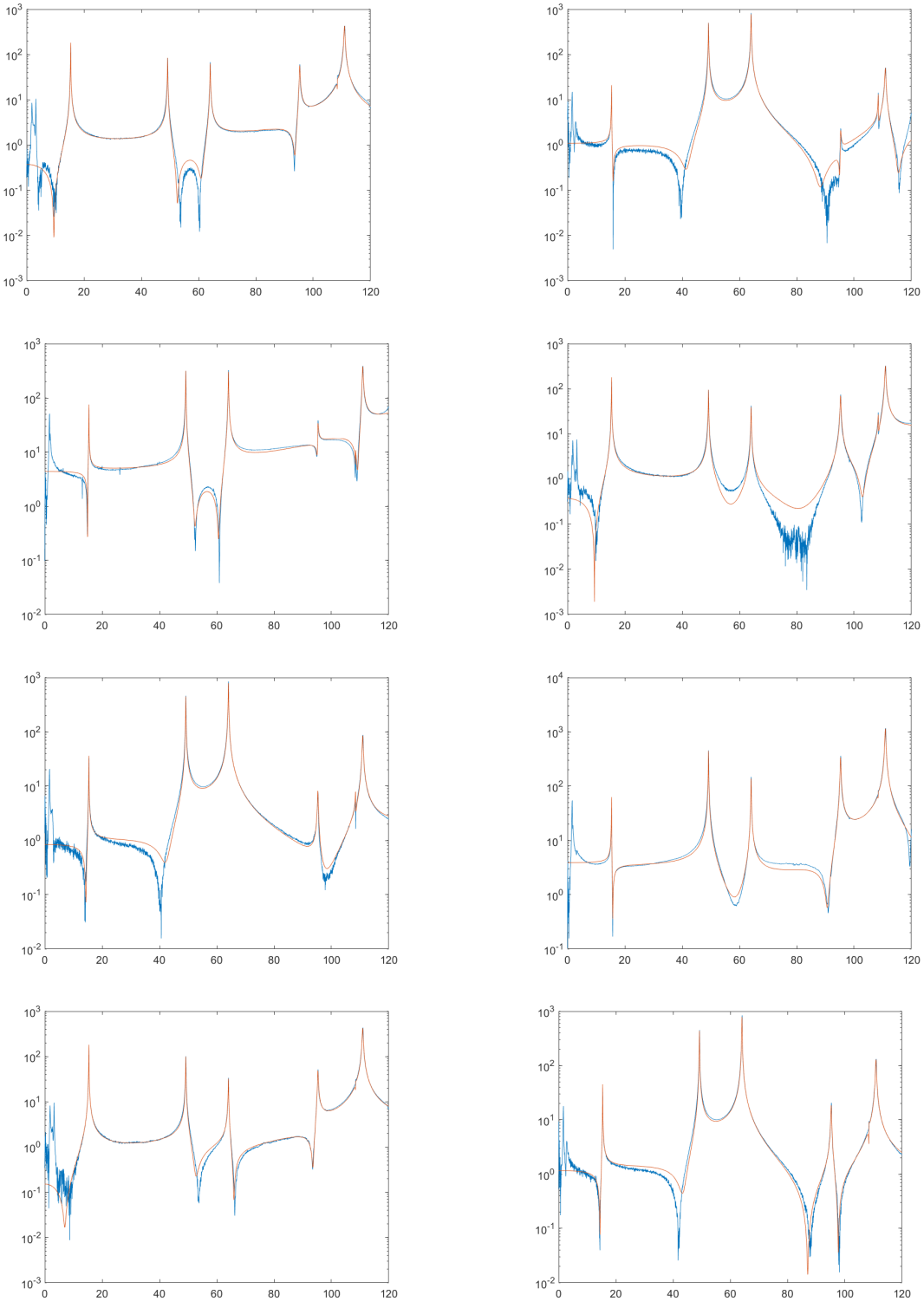

Figure C.6: FRF plots input channel 1, output channels 39 to 46

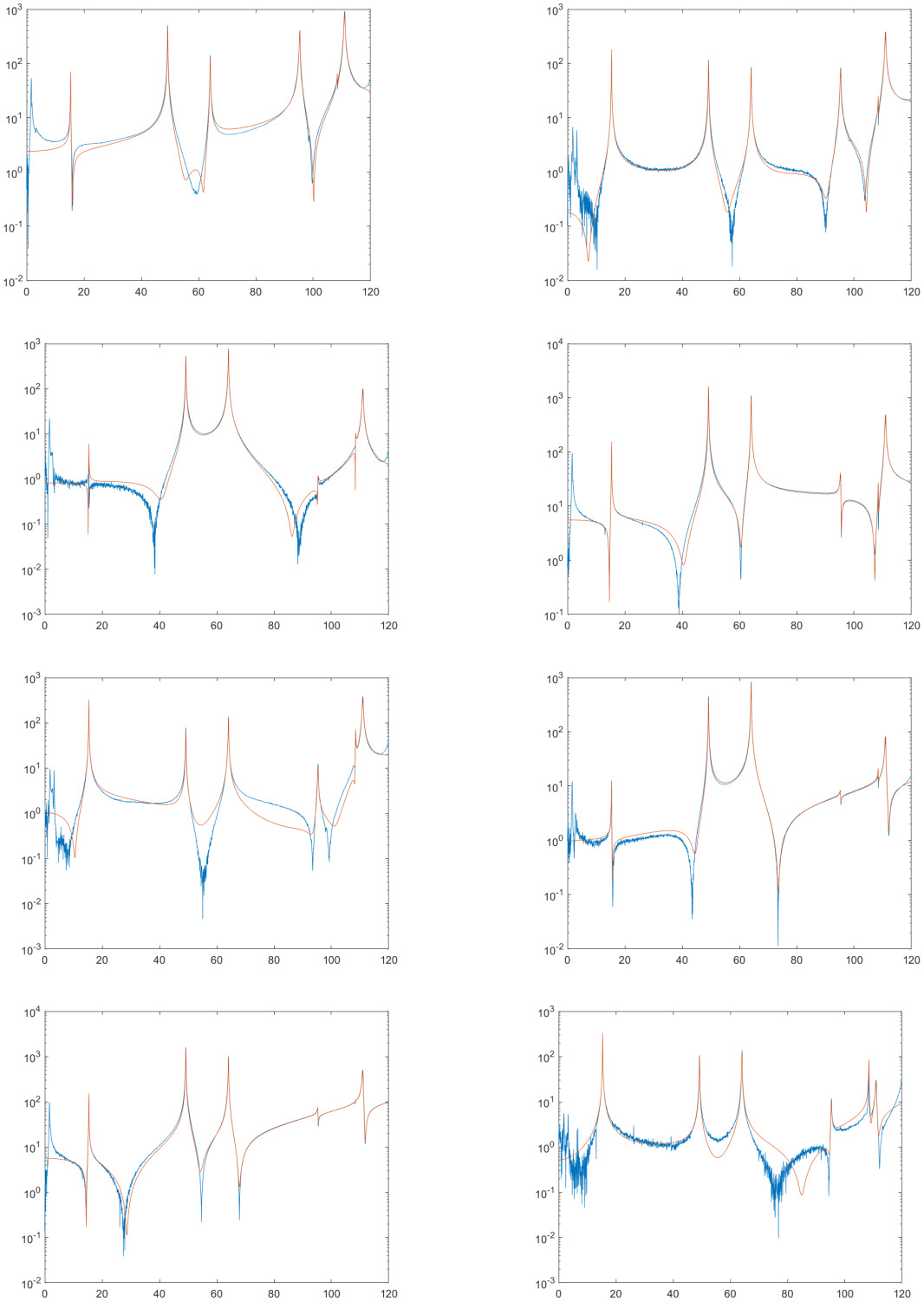

Figure C.7: FRF plots input channel 1, output channels 47 to 54

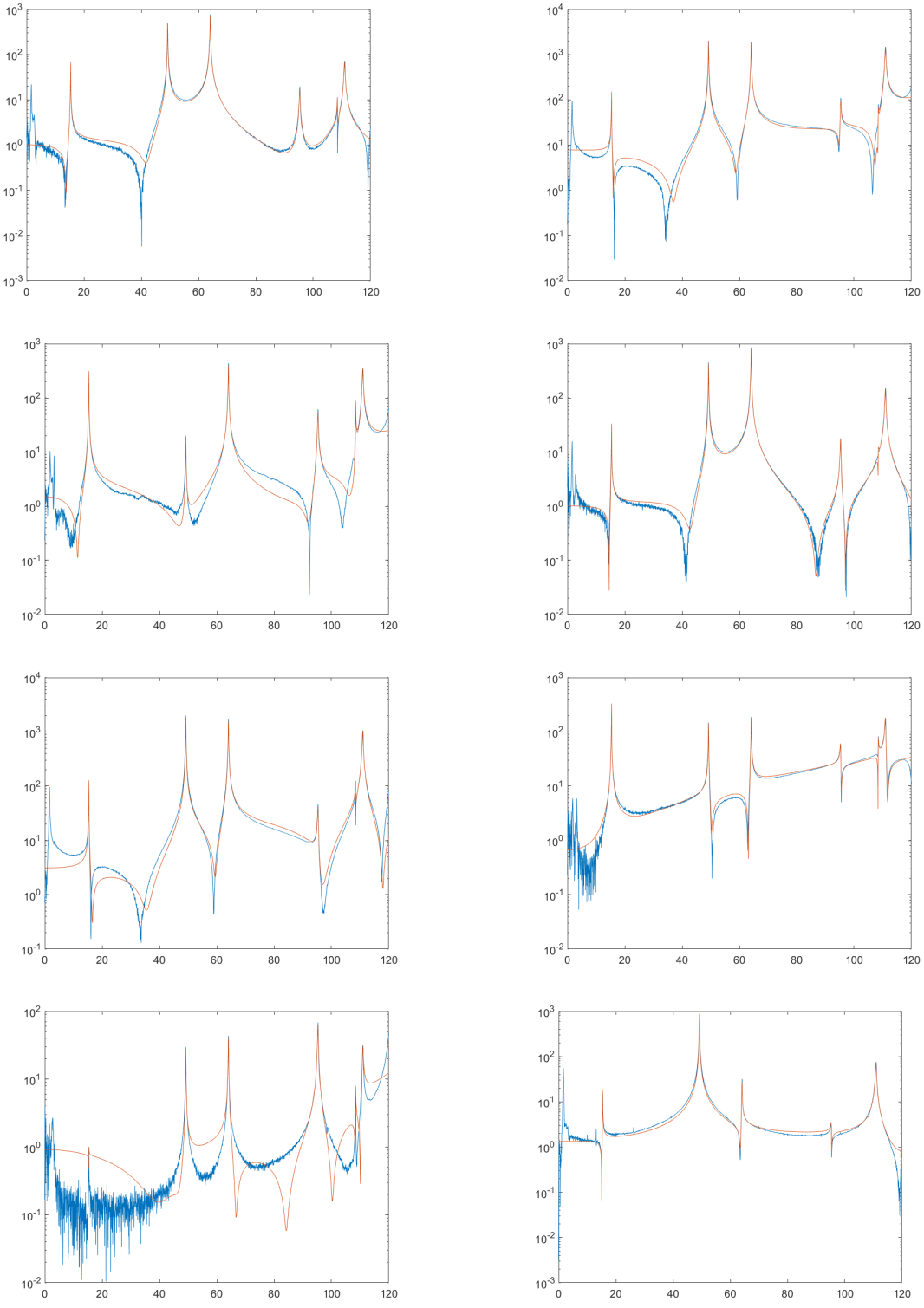

Figure C.8: FRF plots input channel 1, output channels 55 to 62

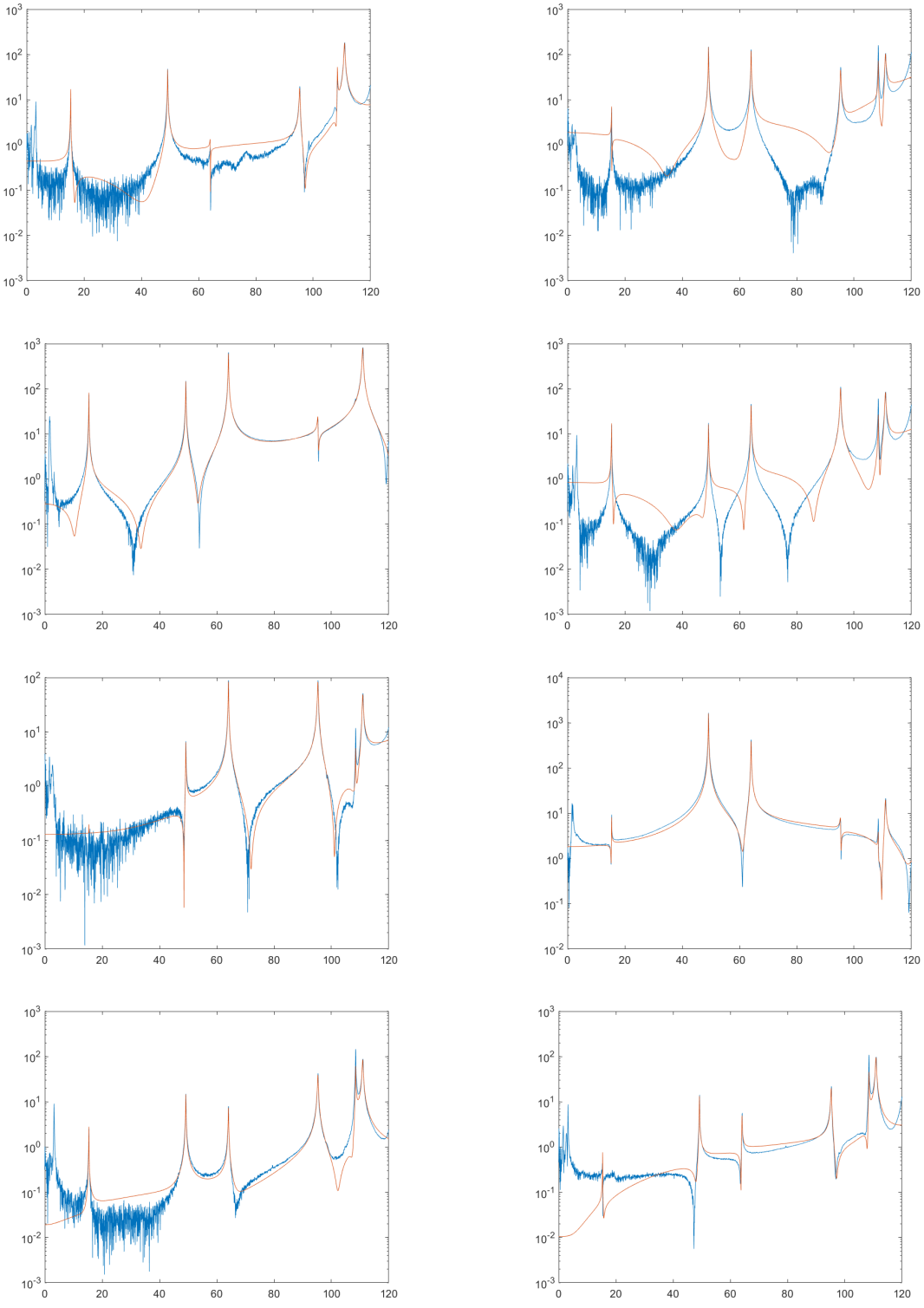

Figure C.9: FRF plots input channel 1, output channels 63 to 70

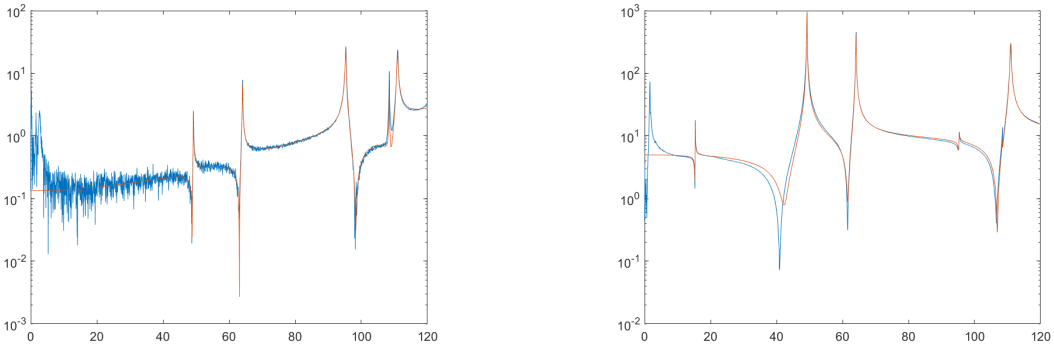

Figure C.10: FRF plots input channel 1, output channels 71 and 72

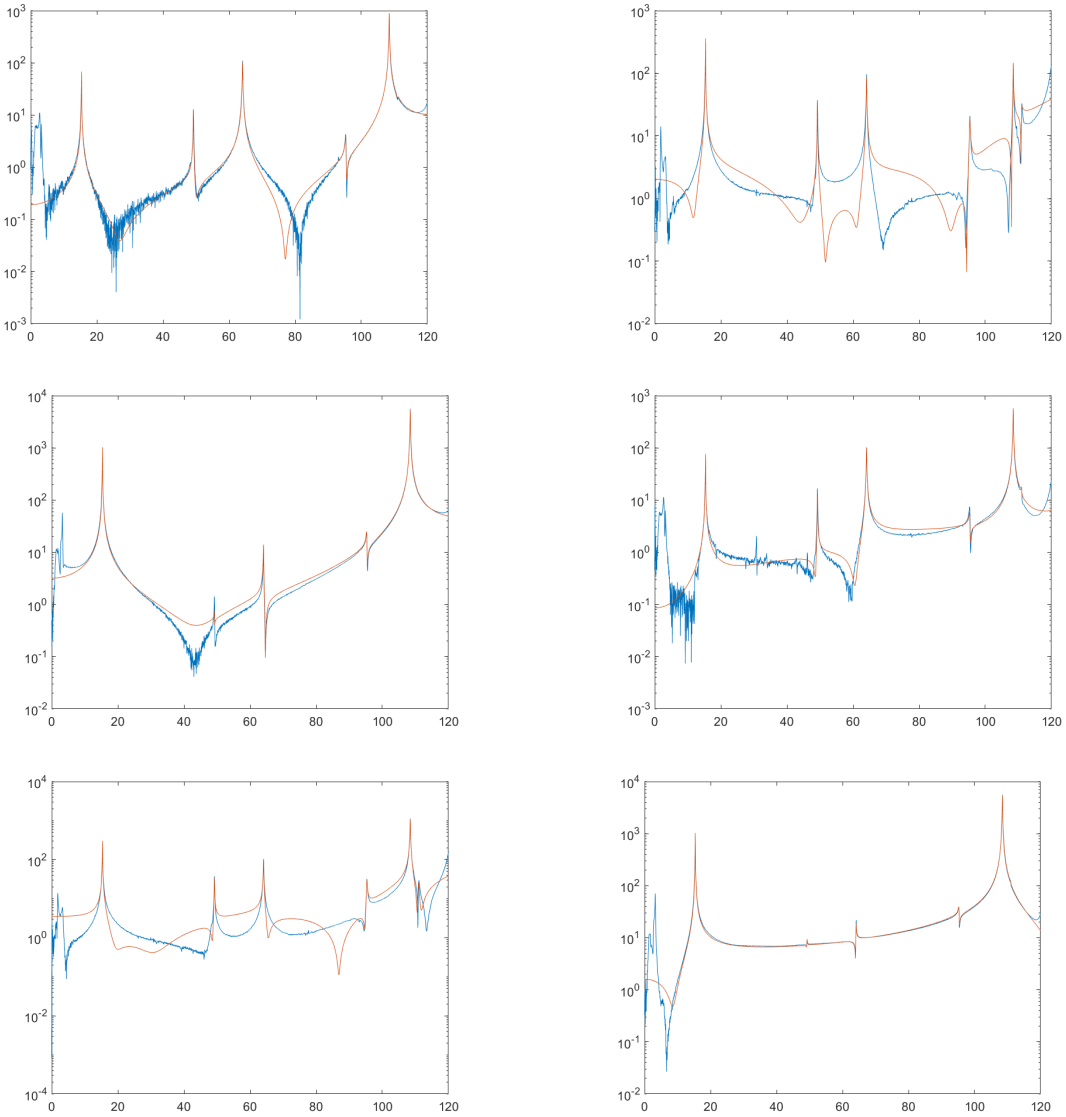

Figure C.11: FRF plots input channel 2, output channels 1 to 6

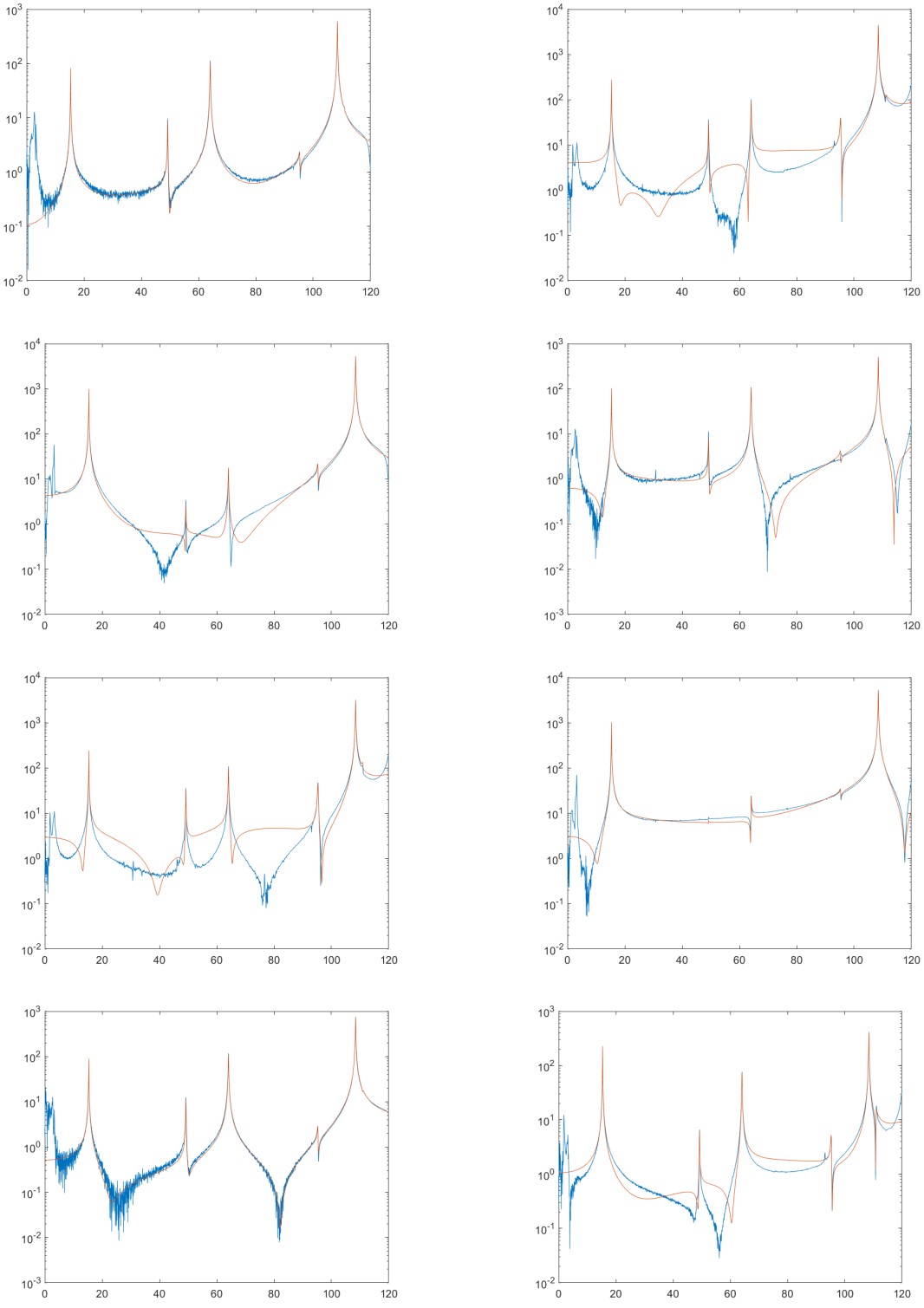

Figure C.12: FRF plots input channel 2, output channels 7 to 14

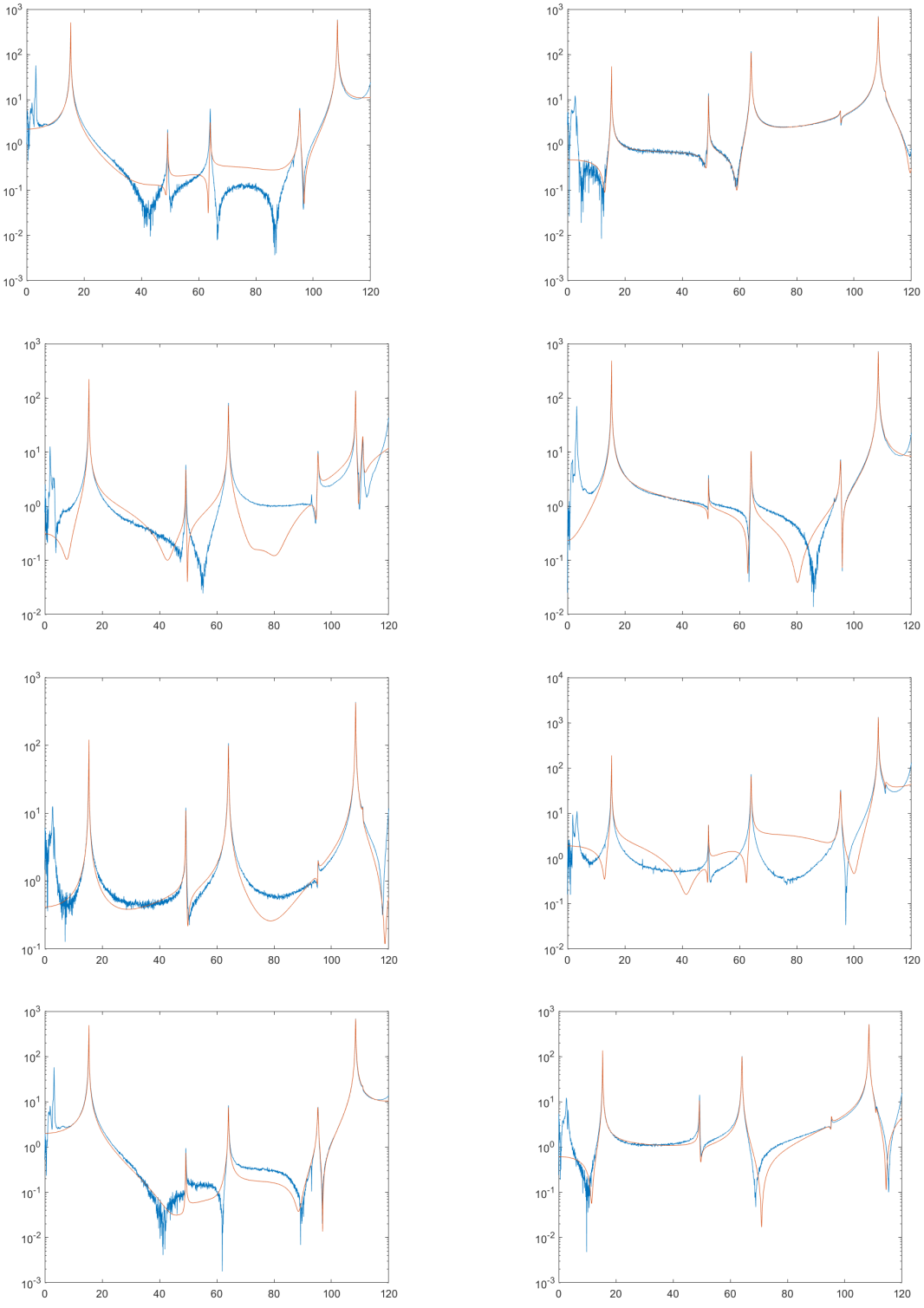

Figure C.13: FRF plots input channel 2, output channels 15 to 22

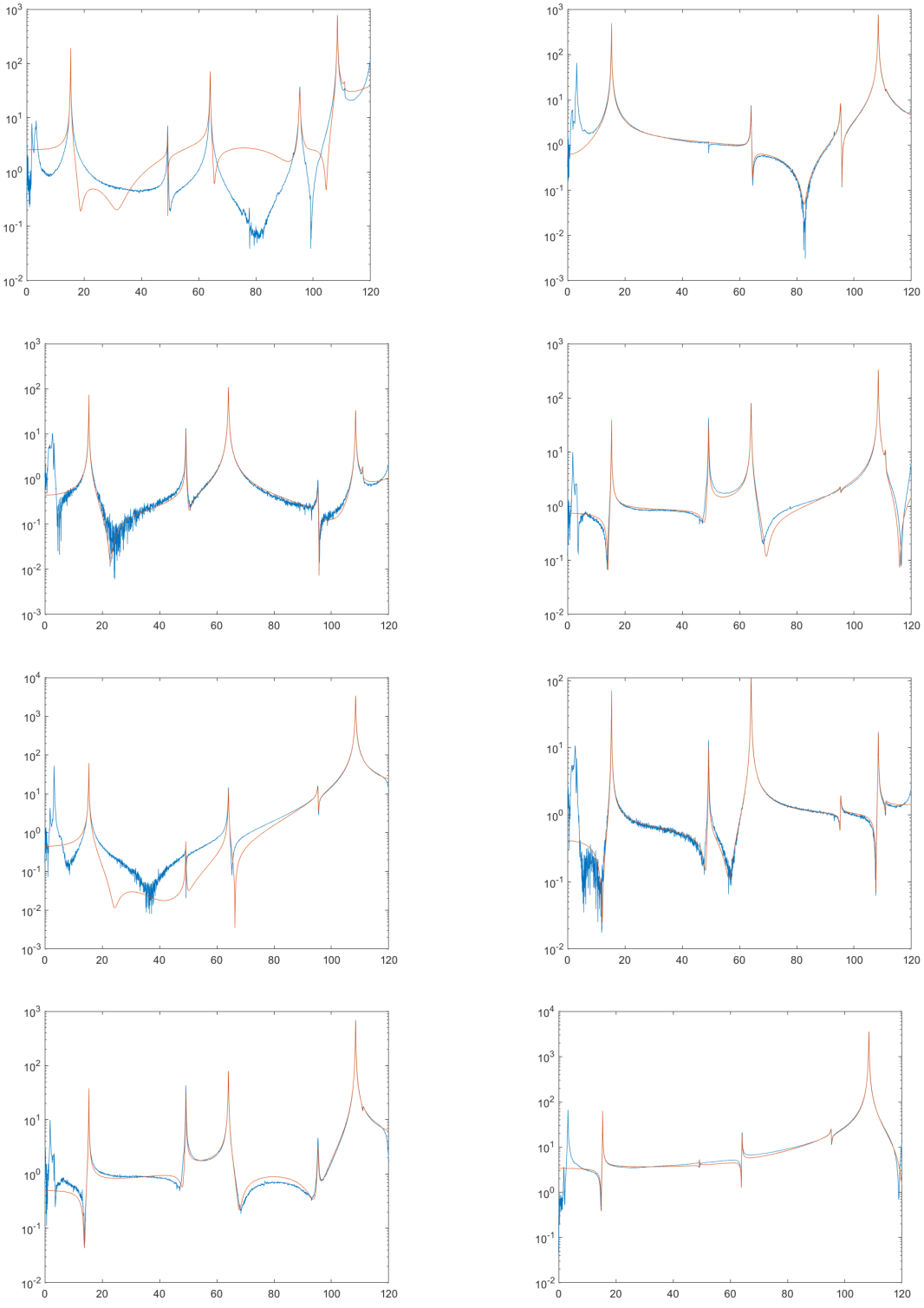

Figure C.14: FRF plots input channel 2, output channels 23 to 30

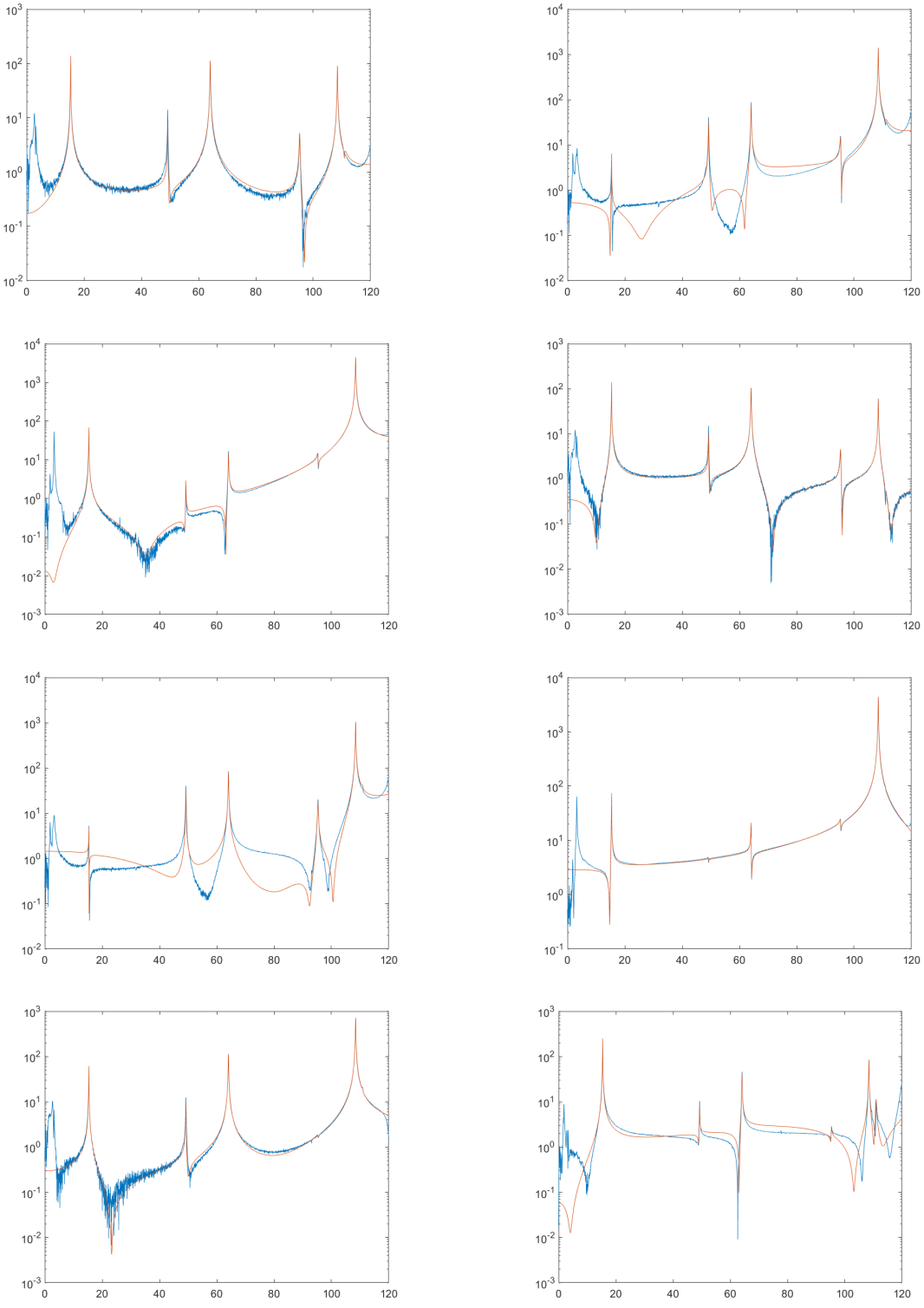

Figure C.15: FRF plots input channel 2, output channels 31 to 38

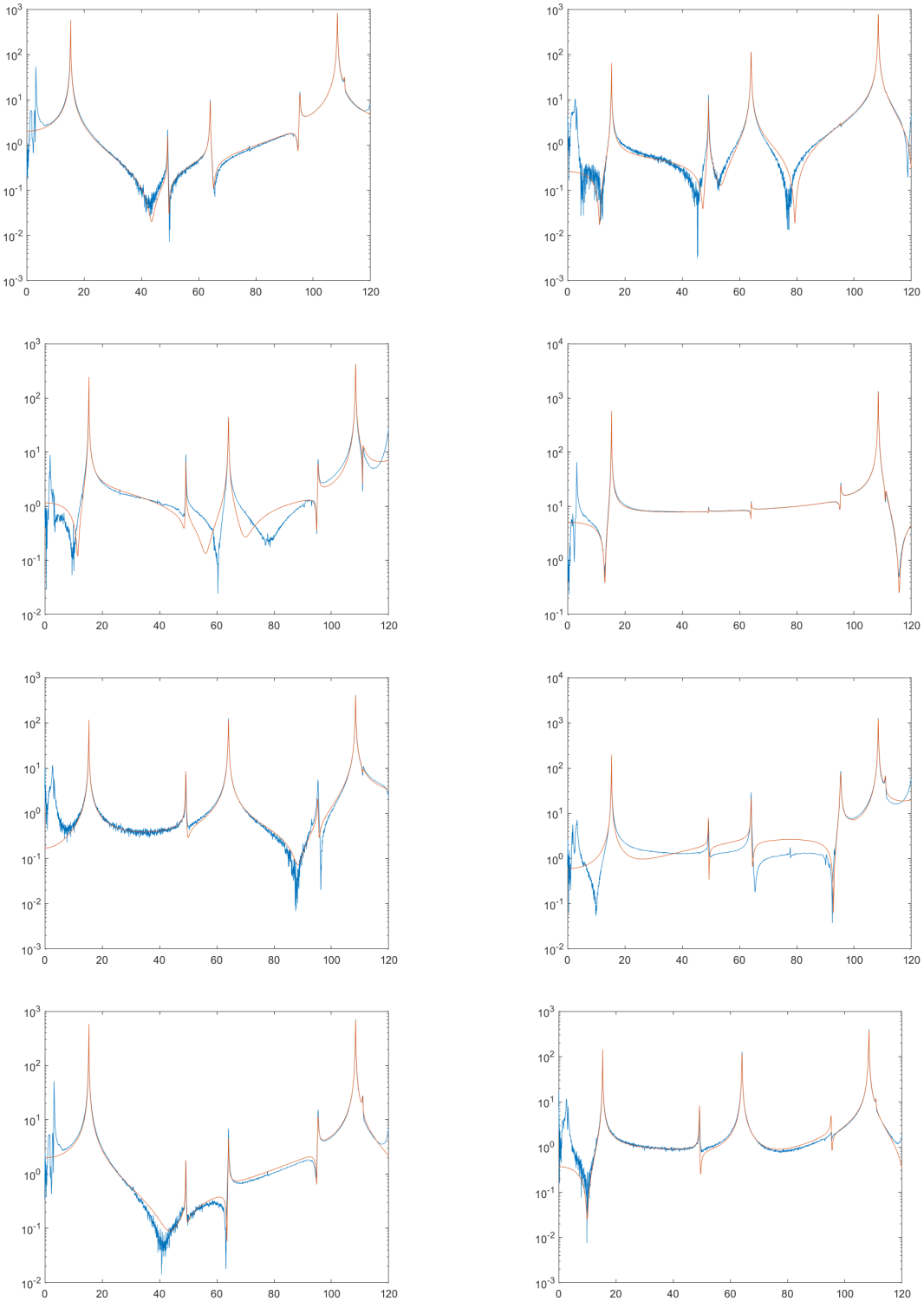

Figure C.16: FRF plots input channel 2, output channels 39 to 46

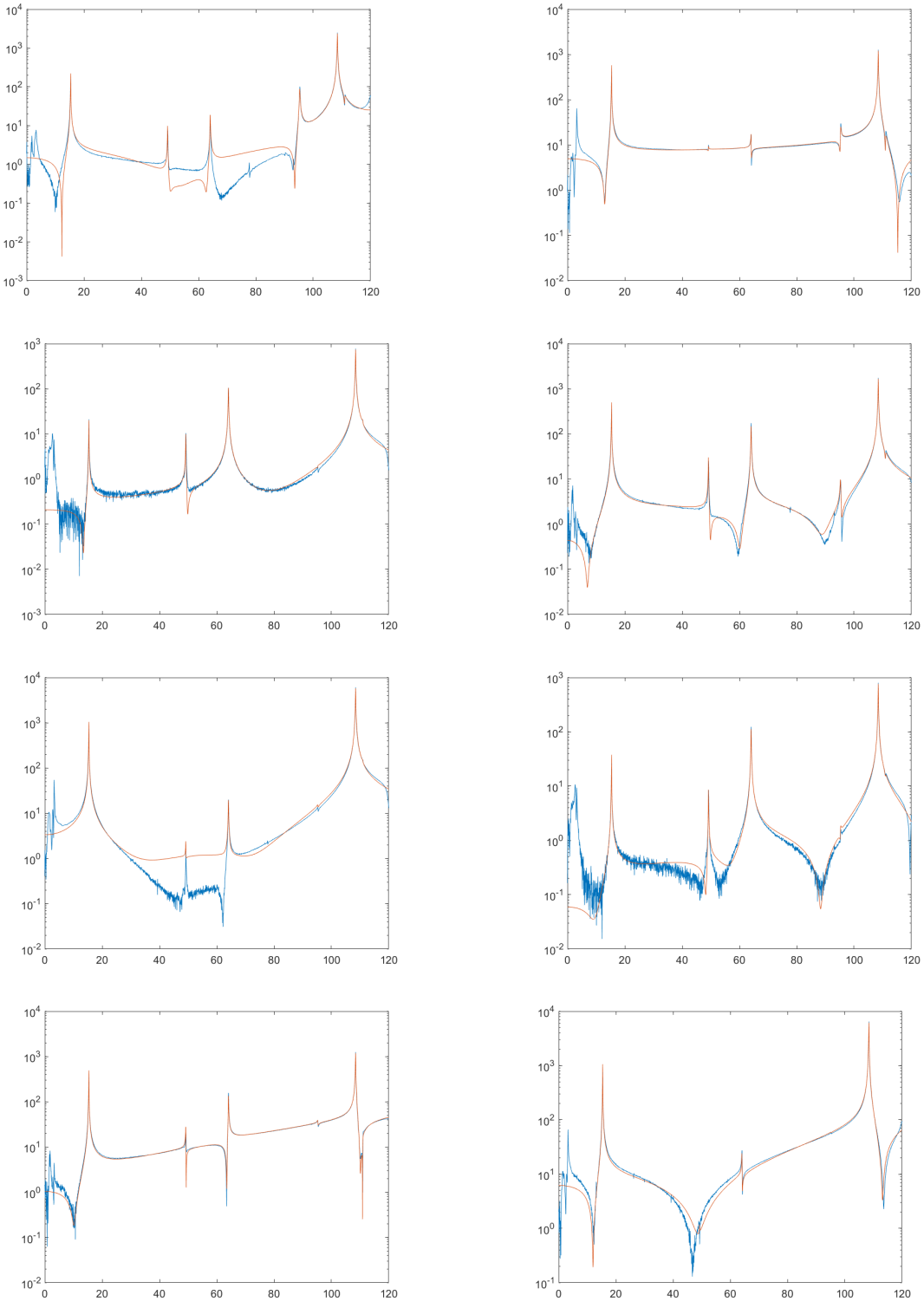

Figure C.17: FRF plots input channel 2, output channels 47 to 54

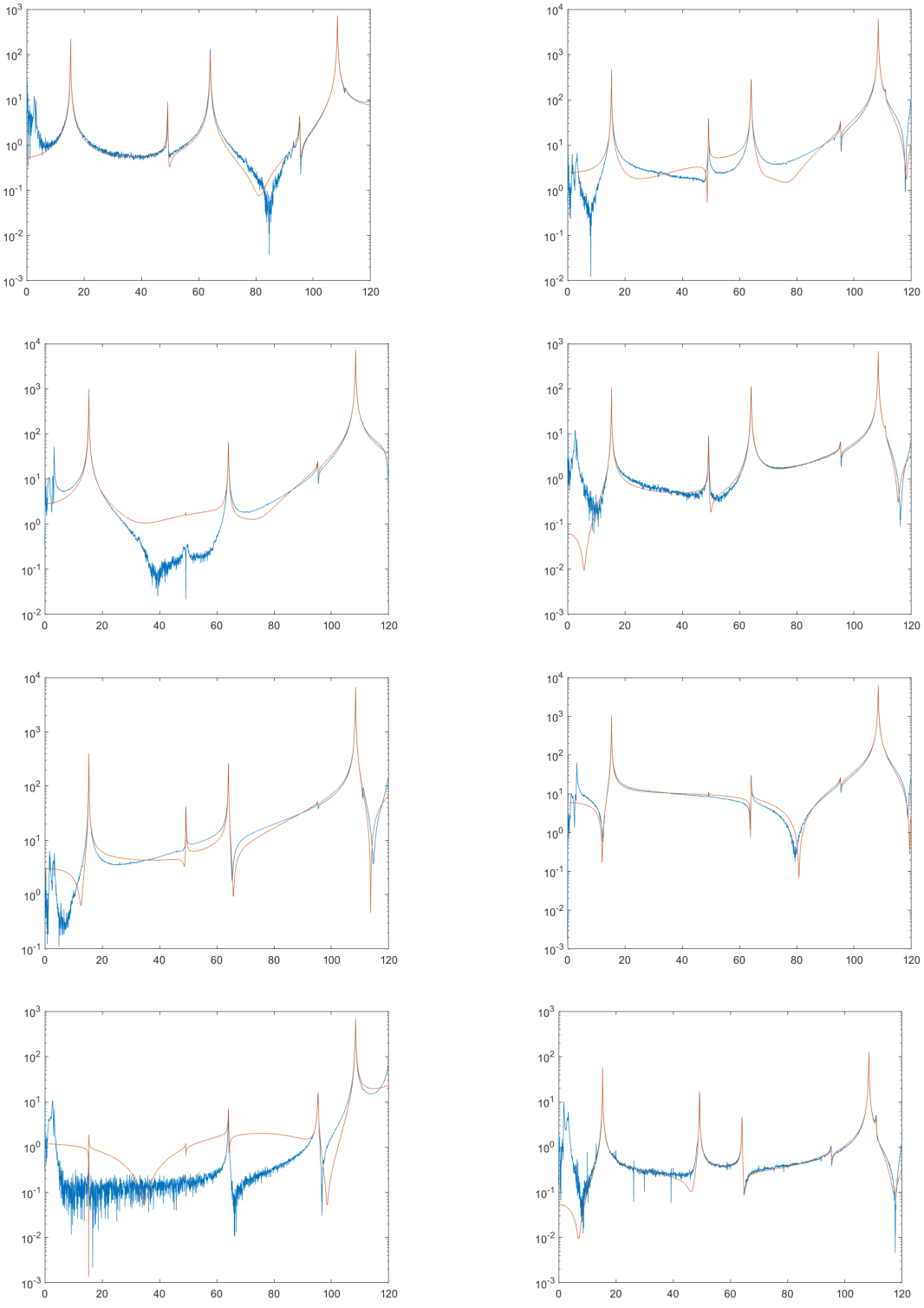

Figure C.18: FRF plots input channel 2, output channels 55 to 62

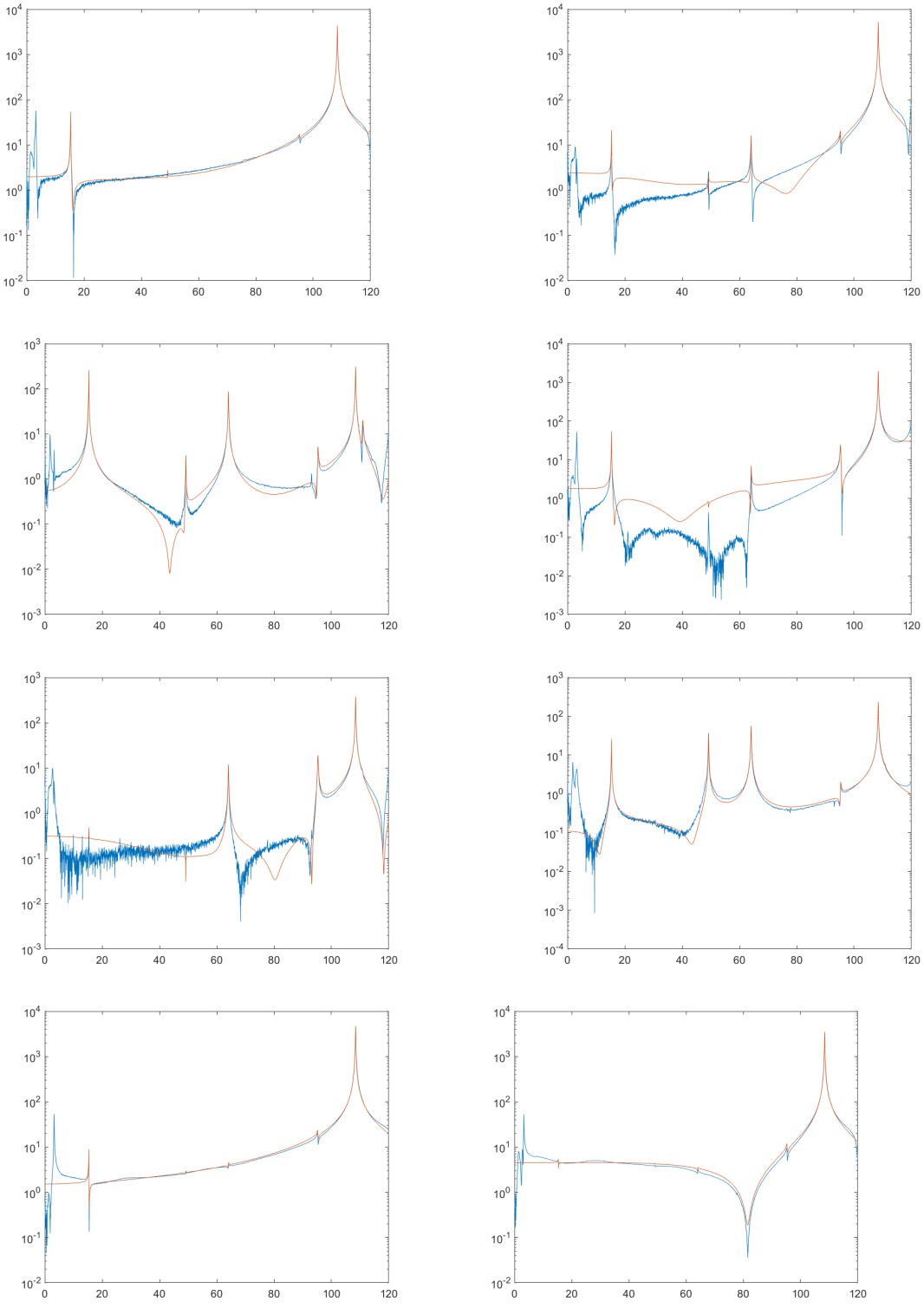

Figure C.19: FRF plots input channel 2, output channels 63 to 70

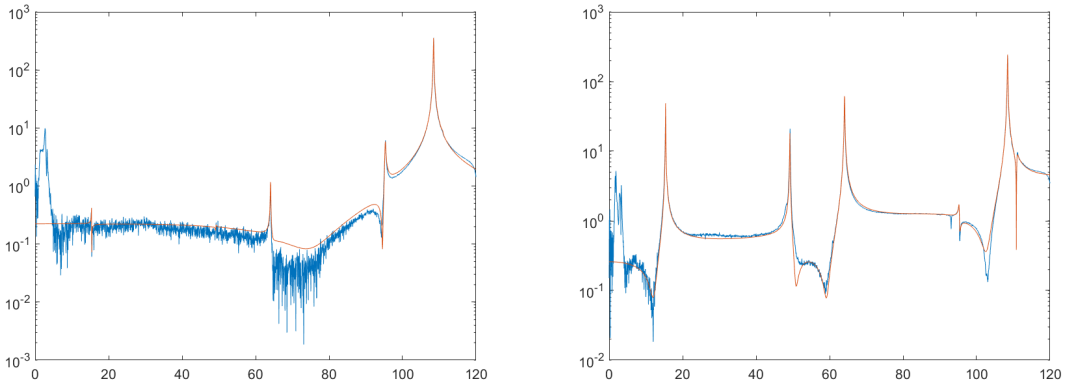

Figure C.20: FRF plots input channel 2, output channels 71 and 72