Instructional Shake Table
For Demonstrations in Structural Dynamics

by

Muhammed Ali Irfan Baig

B.Sc. Civil Engineering
University of Engineering and Technology,
Lahore, Pakistan, 2000

Submitted to the department of Civil and Environmental Engineering
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE IN CIVIL AND ENVIRONMENTAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEBRUARY 2003

Signature of Author...
Department of Civil and Environmental Engineering
August 24, 2002

Certified by...
Eduardo Kausel
Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by..
Oral Buyukozturk
Chairman, Departmental Committee for Graduate Studies

Instructional Shake Table
For Demonstration in Structural Dynamics

by

Muhammed Ali Irfan Baig

Submitted to the Department of Civil and Environmental Engineering
on August 24, 2002,
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Civil and Environmental Engineering

ABSTRACT

Instructional Shake tables are an effective pedagogical tool for use in class room and lab, for teaching structural dynamics at advanced undergraduate and graduate level. A detailed study is carried out for a miniature shake table, currently used by the University Consortium of Instructional Shake Tables. A detailed description of the apparatus is given, followed by the system operation.

Experiments that simulate different earthquakes are run, and response of a two degree of freedom structure model is noted. Issues regarding interpretation of the response are discussed. Finally an application of the apparatus is demonstrated by use of wavelets for extracting vibration modes of the structure and their damping.

Thesis Supervisor : Eduardo Kausel
Title : Professor of Civil and Environmental Engineering

ACKNOWLEDGMENTS

First of all I want to express my gratitude towards my advisor Professor Eduardo Kausel, who gave me the opportunity to work on this project, and guided me through it.

Thanks are also due to my fellow graduate students, especially Ragunathan Sudarshan who gave me valuable tips and advice whenever I needed it, and Hoang-Phong Nguyen who worked with me on this project and was the master mind behind all the Java and web related applications for this project.

My family has been a source of great support and strength for me throughout this last one and half year, especially my young son, whose presence with me was a source of joy and contentment. Without the support of my family, I could not have done this.

I am also grateful to my parents, especially my father who encouraged me more than anybody else, to try for the best, to believe in myself. He was one of the biggest driving forces in convincing me to try to come to MIT.

I also wish to thank Microsoft Corporation, who with their generous support to the I-campus project, of which the Shake Table is a part, made the project possible.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

CHAPTER 2

THE APPARATUS

CHAPTER 3

THE SOFTWARE

CHAPTER 4

EARTHQUAKE SIMULATION

CHAPTER 5

AN APPLICATION: ESTIMATION OF DAMPING

CHAPTER 6

CONCLUSION

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In 1998, University Consortium on Instructional Shake Tables was established, comprising of twenty three universities from three national earthquake research centers: Pacific Earthquake Engineering Research Center (PEER), Mid-America Earthquake Center (MAE) and Multidisciplinary Center for Earthquake Engineering Research (MCEER). The headquarters of the consortium are based in the Washington University at St. Loius.

The objective of this consortium is to "integrate earthquake engineering and structural dynamics into the undergraduate civil engineering curriculum " by introducing and promoting the use of small scale instructional shake tables into undergraduate curricula, and to give undergraduate students an opportunity to have hands-on experience in carrying out structural dynamics experiments. Such an opportunity augments the theory that is taught in classroom, along with introducing the students to experimental research in structural engineering.

1.2 MOTIVATION

The pedagogical value of such an instructional shake can be realized in the fact that it can be effectively used in a number of different courses at the undergraduate level. The flexibility of the system is such that different types of models can be tested, each experiment built around the different courses, highlighting the important and key concepts of that particular topic. In our case, the table is used to test a simple two degree of freedom steel structure, modeling a multi story building. The model is instrumented, and the whole system is used in conjunction with a data acquisition and control system. The whole setup not only gives the ability to record and save the response of the model due to the base excitation given by the vibrating table (the results of the experiment), but also allows for carrying out subsequent data processing and signal analysis procedures to extract information from the saved response, and to interpret results from it.

Such uses of instructional shake tables provide an excellent basis for an introduction to experimental methods in research in structural engineering. Much work has been done on this by Dyke, Caicedo, Johnson, etc. (2002), in this regard.

1.3 ORGANIZATION

In Chapter 2 we look at the hardware comprising the shake table, how the system is built, its various components and how they fit in the framework, the technical specifications. In Chapter 3 we go through the operation of the instructional shake table, and study the way the control software works, along with how experiments are made and run. In Chapter 4 we take a closer look at one of the experiments, namely a simulation of the famous earthquake that occurred in Kobe, Japan in 1995. We see how the experiment is made and what different SIMULINK blocks that it is composed of, actually mean. We also look at response of the model structure to this earthquake.

In Chapter 5 an example of the use of the instructional shake table is given. We show how the modal damping ratios of the model structure can be extracted from the free vibration response of the structure. Standard Discrete Wavelet decomposition is used (the filter bank approach). Lastly, we give the Matlab codes in the appendix.

CHAPTER 2

THE APPARATUS

2.1 OVERVIEW

The apparatus includes the shake table, the data acquisition and control unit, and a Pentium class computer. The shake table is a custom built precision equipment. The development of the system started at the University of Washington St. Louis, under the supervision of Prof. Shirley Dyke of Department of Civil Engineering. Once the system was refined and a prototype built, Quanser Consulting Inc. were selected to build operational units

In this chapter the apparatus of the whole setup, i.e., the shake table, the data acquisition and control unit, are succinctly explained, along with their specifications. Also discussed is the way they are hooked up together and some safety issues regarding their use.

2.2 COMPONENTS

The whole setup comprises of the following componensts :

i) A Shake Table

ii) A two floor model structure
iii) Power Amplifier with data channels
iv) Three Accelerometers
v) Data acquisition and control board (Multi-Q 3)
vi) Active Mass Driver
vii) Power Amplifier for the Active Mass Driver
viii) A portable pendant controller

Each of the above components is considered individually below, with pertinent specifications and uses.

2.2.1 SHAKE TABLE

The Shake Table is the primary component of the apparatus. The whole idea behind the system is to be able to have a "Shake Table" which can be controlled in any way, i.e., it can be shaken in any user defined manner, be it a real earthquake, or any displacements defined by any other function. Such tables have long been in use for carrying out tests on structural specimens. Indeed the Earthquake Simulator Laboratory at University of Californian at Berkeley have 20 ft by 20 ft concrete shake table for testing of full scale specimens for assessment of their response and structural performance under dynamic loads.

Full scale shake tables are hard to maintain and run, and are not a good choice at for demonstrations purposes. The instructional shake table is a small scale version of such equipment. It is small, easy to operate and portable, yet demonstrating clearly the basic principles of structural dynamics. With carefully designed experiments using small scale model structures, insight can be had into the structural response and behavior of different structural systems.

The shake table under consideration consists by a small 1 Hp brushless motor which drives a rotating ball nut coupled to an 18 inch by 18 inch table. The table itself is supported on two paraller shafts, with ball bearings in between. Models of different sizes and shapes can be screwed on top of the table. The following are the specifications (ref: UCIST Shake Table, by Quanser Consulting Inc.):

Parameter	**Value**	
Table Dimensions	18 " x 18 "	
Max. Payload	25	lbs
Peak velocity	33	in/sec
Peak Acceleration	2.5	g
Stroke	+/- 3	in.
Weight	60	lbs.
Ball screw efficiency	90	%

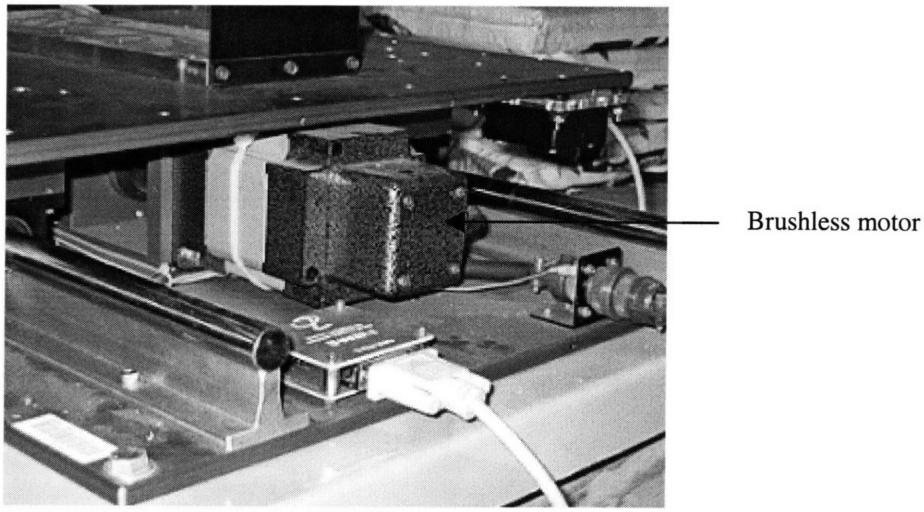

Figure 2.1: The shake table mounted on double shaft rail with brushless motor

2.2.2 MODEL STRUCTURE

The structure attached to the shake table is a two floor model building. The floors of the building are made much stiffer than the steel columns. This allows the structure to be modeled as a shear beam, with horizontal translations of the floors as the only degrees of freedoms. The deformation of the floors is extremely small as compared to those of the columns and hence can be safely neglected for all practical purposes. This approach has long been used in structural engineering practice to model multi story buildings, and the design correspondingly is such that floors are actually constructed stiffer than the columns. The mass of each floor is about 2 lbs.

The model structure can be extended to add more floors and increase the degrees of freedom. The top floor is equipped with a rack-and-pinion system to accommodate the Active Mass Driver, the details of which are explained later.

Fig. 2.2: The Model structure

2.2.3 POWER AMPLIFIER

The Shake Table is driven using a brushless motor amplifier housed in the Universal Power Module (UPM). It also has a separate power supply to power any other instrumentation (accelerometers in this case). Along with the power supply to accelerometers it also collects the data from these, which is then passed on to the Multi-Q3 data acquisition system. This is the Signal Conditioning sections of the system.

The following table gives some specifications for the Universal Power Module:

Parameter	Value
Power Amplifier	
Input Voltage	120 VAC
Maximum Current	25 Amperes
Continuous Current	12.5 Amperes
Signal Conditioning section	
DC output supply	+/- 12 VDC
Maximum Current	1 Ampere

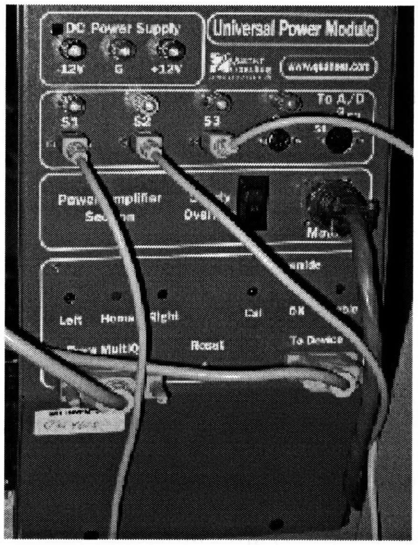

Fig. 2.3: The UPM

The UPM connects to the shake table to supply power and to send all the control signals. Also the X-axis channels of the Multi-Q board connect to the UPM. Up to four accelerometers can be connected. In the case of the shake table, there are a total of three accelerometers : two are mounted on the model structure, and one is mounted on the shake table it self.

2.2.4 ACCELEROMETERS

Accelerometers are the most common instrument used for determining response of structures, and change in dynamic state of a system. It is not easy to measure the displacement response of a vibrating structure with direct measurements. They are almost universally used for response measurements, and for recording any change in the dynamic state of a vibrating system.

The instrument essentially consists of mass connected to a spring and damper. Accelerometers can be described as a combination of two transducers -- the primary transducer, typically a single-degree-of-freedom vibrating mass which converts the acceleration into a displacement, and a secondary transducer which converts the displacement of the seismic mass into an electric signal. The actual construction varies according to the intended use and the range of acceleration and frequency to be measured. Given below is a brief description of various types used for various ranges of interest of acceleration and frequency

Piezoelectric

Piezoelectric transducers are often used in vibration-sensing accelerometers, and sometimes in shock-sensing devices. The piezoelectric crystals (often quartz or ceramic) produce an electric charge when a force is exerted by the seismic mass under some acceleration. The quartz plates (two or more) are preloaded so that a positive or negative change in the applied force on the crystals results in a change in the electric charge. Although the sensitivity of piezoelectric accelerometers is relatively low compared with other types of accelerometers, they have the highest range (up to 100,000 g's) and frequency response (over 20 kHz).

Potentiometric

The displacement of the spring-mass system is linked mechanically to a wiper arm, which moves along a potentiometer. The system can use gas, viscous, or magnetic damping to minimize acoustic noise caused by contact resistance of the wiper arm.

Potentiometric accelerometers typically have a frequency range from zero to 20 - 60 Hz, depending on the stiffness of the spring, and have a high-level output signal. They also have a lower frequency response than most other accelerometers, usually between 15 - 30 Hz.

Servo

In servo accelerometers, acceleration causes a seismic mass "pendulum" to move. When motion is detected by a position-sensing device, a signal is produced that acts as the error signal in the closed-loop servo system. After the signal has been demodulated and amplified to remove the steady-state component, the signal is passed through a passive damping network and is applied to a torque coil located at the axis of rotation of the mass. The torque developed by the torque coil is proportional to the current applied, and counteracts the torque acting on the seismic mass due to the acceleration, preventing further motion of the mass. Therefore, the current through the torque coil is proportional to acceleration. This device can also be used to measure angular acceleration as long as the seismic mass is balanced. Servo accelerometers provide high accuracy and a high-level output at a relatively high cost, and can be used for very low measuring ranges (well below 1 g).

Strain Gauge

Strain gauge accelerometers, often called "piezo-resistive" accelerometers, use strain gauges acting as arms of a Wheatstone bridge to convert mechanical strain to a DC output voltage. The gauges are either mounted to the spring, or between the seismic mass and the stationary frame. In the picture, strain gauge windings, which contribute to the spring action, are stressed (two in tension, two in compression), and a DC output voltage is generated by the four arms of the bridge that is proportional to the applied acceleration. These accelerometers can be made more sensitive with the use of semiconductor gauges and stiffer springs, yielding a higher frequency response and output signal amplitude. And unlike other types of accelerometers, strain gauge accelerometers respond to steady-state accelerations.

Capacitive

A change in acceleration causes a change in the space between the moving and fixed electrodes of a capacitive accelerometer. The moving electrode is typically a diaphragm-supported seismic mass or a flexure-supported, disk-shaped seismic mass. The element can act as the capacitor in the LC or RC portion of an oscillator circuit. The resulting output frequency is proportional to the applied acceleration.

Vibrating Element

In a vibrating element accelerometer, a very small displacement of the seismic mass varies the tension of a tungsten wire in a permanent magnetic field. A current through the wire in the presence of the magnetic field causes the wire to vibrate at its resonant frequency (like a guitar string). The circuitry then outputs a frequency modulation (deviation from a center frequency) that is proportional to the applied acceleration. Although the precision of such a device is high, it is quite sensitive to temperature variations and is relatively expensive.

Important characteristics of accelerometers include range of acceleration, frequency response, transverse sensitivity (i.e. sensitivity to motion in the non-active direction), mounting errors, temperature and acoustic noise sensitivity, and mass. High-frequency-response accelerometers are used mostly for vibration and shock testing. Shock testing includes automobile and aircraft crash testing, and studying the effects of explosions or earthquakes on buildings and other large structures. Uniaxial, biaxial or triaxial accelerometers are available. Also angular accelerations can be recorded.

One very important aspect in acceleration measurements using accelerometer is the DC offset. This means in essence that the measurements are recording a zero frequency component as well. This is non realistic for vibrating structures, and is primarily due to electrical drift within the instrument. Later it is described how to get around this problem

For the Shake Table, we have three accelerometers provided by Quanser Consulting Inc. Two of these are mounted on the intermediate and top floors of the model structure, and the third one is attached to the table itself to monitor the accelerations of the table, to keep a check on whether the table is actually vibrating according to the input signal or if there is a lag between the actual and desired response.

The range of the accelerometers is +/- 5 g with an output of +/- 5 volts.

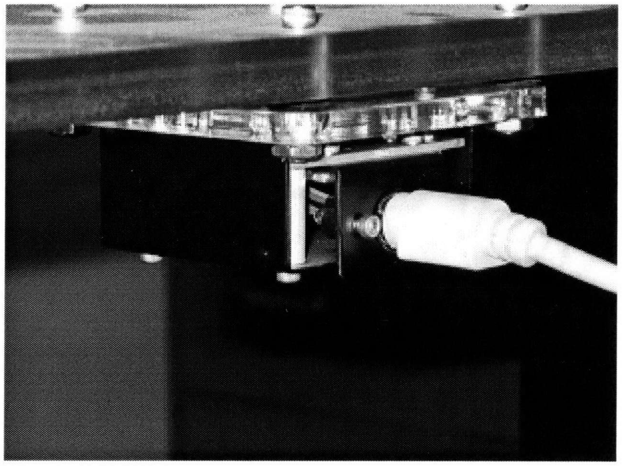

Fig 2.4: Accelerometer mounted on model structure

2.2.5 DATA ACQUISITION AND CONTROL BOARD (MULTI-Q)

The Data acquisition system used is the MULTI-Q 3. This is a general purpose data acquisition and and control board with 8 single ended analog inputs, 8 analog outputs, 16 bits of digital input and output each, and 3 programmable timer clocks. The whole system comprises of the MultiQ board which goes into the ISA slot of the Pentium class PC used for the control of the Shake Table, and a terminal board which has all the slots for inputs and outputs. The two are connected by a flat ribbon cable.

THE TERMINAL BOARD

The terminal board has two ports : the X axis and Y axis ports. These may be used to advantage for biaxial motions, by attatching one table to a second one, but this will deteriorate the bandwith of the lower table. In this instance, only the port labeled X- axis is used. Shown below is the terminal board for MultiQ-3.

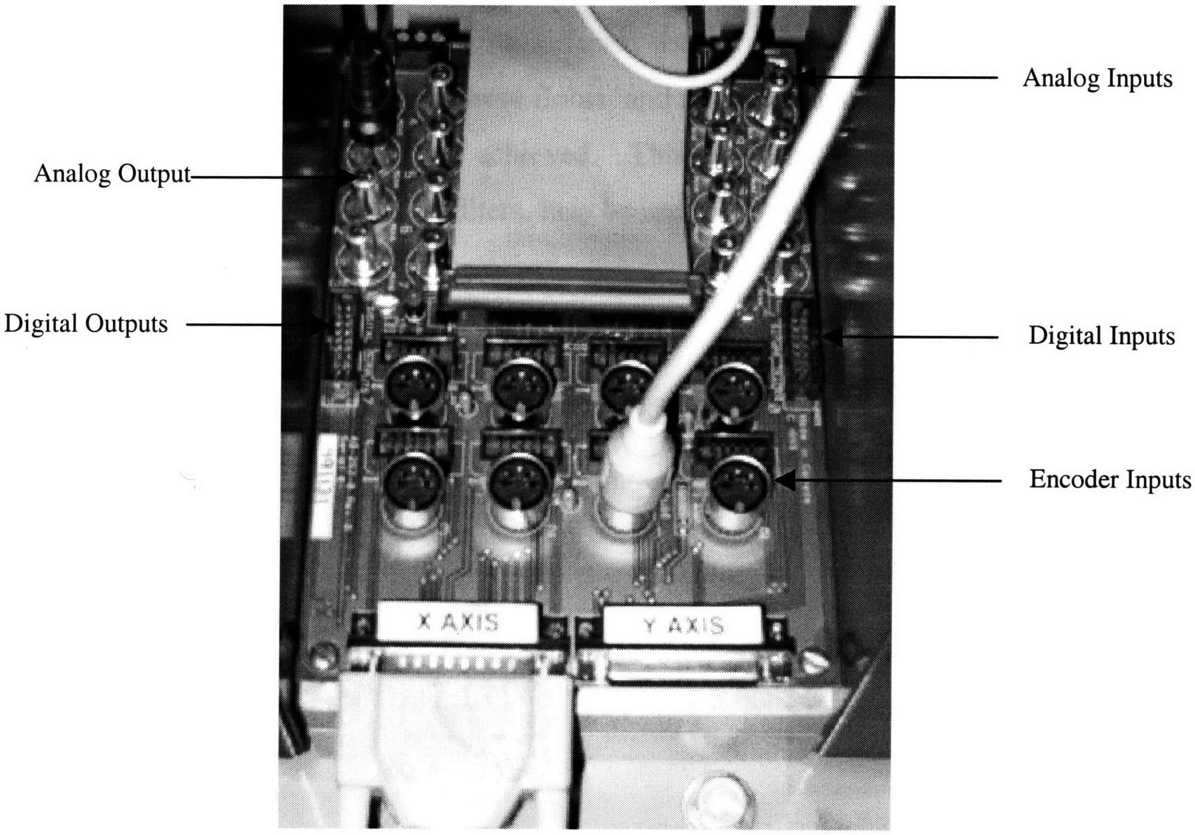

Fig. 2.5 : MultiQ-3: the terminal board

Inputs and outputs that relate to the table controller are accessible through the X-axis port as well. The details of the connectivity are described in later sections.

Encoder Inputs

In addition, the terminal board is also equipped with eight Encoder inputs. These are used to read off the number of revolutions of a motor, keep track of them and increment or decrement according to the actual revolutions.

2.2.6 THE ACTIVE MASS DRIVER

The Active Mass Driver is an optional accessory to the system. It is installed on the top of the model structure, where it sits on a serrated rack, sliding on a stainless steel rod for safety. The AMD can be programmed to move in any manner, but its main purpose to be installed on the building is to be a means of actively controlling the vibrations of the model structure. It is possible to write different control algorithms for the AMD, which incorporate the acceleration of different floors, and move the mass with a an acceleration such that the desired response is achieved. This is a nonlinear problem and active predictive filters, like the Kalman filters, may be used. Below is the picture of the AMD.

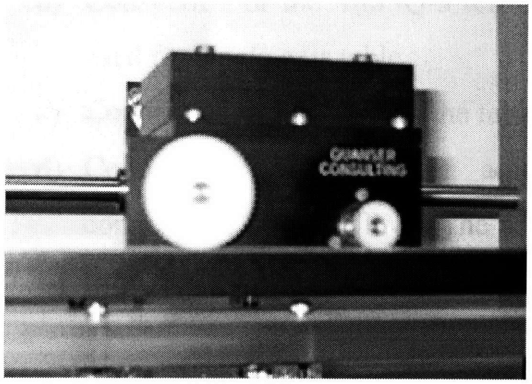

Fig. 2.6: The Active Mass Driver

Fig.2.7: The Power Module

The AMD is also manufactured by Quanser Consulting, Inc., and is a modified form of their IP01 cart, with extra mass added to make it more effective as a damper. The DC motor of the cart is powered by a separate UPM. This UPM also sends signals to the cart to update its postion, through a connection to the Analog output of the terminal board of MultiQ-3. The connectivity is explained in the next sectio

2.3 THE CONNECTIVITY

The table together with the Universal Power Module (UPM), the MultiQ-3 data acquisition and control board and the Active Mass Driver, together with the dedicated PC make up the whole system. In this section the connectivity of all the components is briefly explained.

The following connections are required for the Shake Table to be setup properly:

a) The power cable from the Universal Power Module to the motor. This is done using the 3-pin power cable.
b) Connection of the MultiQ-3 terminal board to the UPM. Channel for X axis is used for single axis table.
c) Connection of the UPM to the table circuit using a DB-15 cable.
d) Connections of the three accelerometers to the Analog-to-digital (A/D) connections on the UPM. The accelerometer from the table goes to S1, the one from intermediate floor goes to S2 and the one from top floor goes to S3.

Connecting the terminal board to UPM via X axis channel gives access to the relevant inputs and outputs. These, along with their functions are given in the following table.

D/A channel #0 (Analog output # 0):	*Drives the motor, driven by amplifier*
A/D channel #0 (Analog input # 0):	*Measures accelerometer on table*
A/D channel #1 (Analog input #1):	*Measures accelerometer on 1^{st} floor.*
A/D channel #2 (Analog input #2):	*Measures accelerometer on 2^{nd} floor.*
Encoder #0:	*Measures the revolutions of the table motor*
Digital input #0:	*Detects when the table reaches the left limit*
Digital imput #1:	*Detects when the table is at home position.*
Digital imput #2:	*Detects when the table reaches right limit*
Digital Output #0:	*Controls "calibrate" on table micro controller*
Digital Output #1:	*Controls "enable" on table micro controller*

Of course this means that these outputs and inputs can also be accessed from the X axis channel as well as the individual terminals on the board.

Active Mass Driver Connectivity

The AMD is run by a small DC motor, equipped with an encoder to count the number of revolutions. The motor is powered by a separate power module. The connections of these two are as follows

a) The cart is powered by the power module, by connecting to the "to load" terminal on the power module.

b) The card encoder is connected to the encode #1 on the terminal board. This will send the number of revolutions of the cart motor to the terminal board. The number of revolutions then can easily be translated into the linear displacement of the cart.

c) The power module is connected to the terminal boards by wiring the Analog output #1 to the "from D/A". This sends the signal from the MultiQ-3 to the cart. The resulting action is the appropriate displacement of the cart with the right velocity.

This completes the connectivity of the system.

2.4 SAFETY

There are some safety features built into the system to ensure that the equipment is not damaged. There is a switch on the main UPM labeled " Safety Overide". *This should be kept in the "OFF" position at all times.*

The table is equipped with *Limit switches.* These enable the table controller to ascertain if the end of travel limits are reached by the table. They deactivate the amplifier, so that the table can not go off limits. Also the table should be zeroed before starting so that the chances of exceeding the limits are minimized. Also any earthquake simulations that are run on the table, should be properly scaled down execution. This is discussed in the next chapters.

CHAPTER 3

THE SOFTWARE

3.1 OVERVIEW

The whole apparatus is controlled and data collected using a standard Pentium class computer. The flexibility of the system along with the ease of use of the control software make it an attractive choice for quick and easy demonstrations of principles of structural dynamics and earthquake engineering in real time. Not only that, but also the system is highly modular, enabling the user to design virtually any kind of experiments regarding dynamics, run them, and change many parameters while running the experiments in real time.

In this chapter a description is given of the different software packages that are used, how they link together, and how the system is managed using these. Also a succinct description is given of the pertinent drivers for different operations that are used to control the data acquisition card.

3.2 THE SOFTWARE PACKAGES

The system is controlled and run in real time by the following packages

i) MATLAB Computing Environment (from Mathworks Inc.)
Along with the following toolboxes:
    SIMULINK
    Real-Time Workshop
    Signal Processng Toolbox
    Control Systems Toolbox

ii) WinCon 3.1 (or 3.2) (from Quanser Consulting Inc.)
    WinCon Server
    WinCon Client

iii) Microsoft Visual C++ 6.0

Each of these programs is explained in the following sections, not necessarily in the order given above. We begin with MATLAB and SIMULINK, and then go on to explain the role they play in the creating model experiments and executing them. Lastly we discuss the control interface package WinCon.

3.3 MATLAB AND SIMULINK

MATLAB computing environment is at heart of the model experiment preparation and compilations. All the parameters that are to be set for the model are set in MATLAB. Also it is most convenient to carry out any subsequent data processing on the data collected sensors during the experiment execution. Not only the SIMULINK blocks, but also some functions, like the one to scale down the ground displacements for earthquakes, also are run using MATLAB. Hence a reasonable familiarity with this package is required for not only model preparation, but also for data collection and processing.

3.3.1 SIMULINK

SIMULINK, is a software package bundled with MATLAB, for modeling simulating and analyzing any dynamic linear or nonlinear system. The simulations can be modeled in discrete time or continuous time, or both. The attractiveness of SIMULINK lies in its graphical user interface. The package lets the user prepare models as block diagrams, with the standard windows click-and-drag operations. There is an extensive library of blocks for objects like sinks, sources, linear and nonlinear components. The blocks from the library can all the graphically connected together to make a compact, comprehensive model. Once a model has been drawn, parameters for each block can be set by clicking on blocks and entering appropriate values in relevant property sheets.

Models can be prepared in a very modular fashion. The whole model can be divided into small modules, each receiving a certain number of inputs and giving out some outputs, which are used by other components as inputs. This means that the models can be designed as a hierarchy of sub-models. This allows both top-down and bottom up approaches. The model can be viewed a higher level, and then each block can be opened up by clicking to see the details of that component.

Once a model has been defined, it can be simulated. Different integration methods can chosen for the solution, from the SIMULINK library. During run time, the out put of different blocks can be viewed using scope blocks. These allow the output to be plotted in real time. The simulation results can be put in MATLAB workspace for post processing and visualization.

3.3.2 THE TOOLBOXES: REAL-TIME WORKSHOP, SIGNAL PROCESSING TOOLBOX AND CONTROL SYSTEMS TOOLBOX

The models prepared using SIMULINK are PC simulations. In many instances, like running a model on a hardware like shake table, this model needs to be translated into a form that is readable by a hardware controlling device. In many cases this device runs C code. The Real Time Workshop toolbox is part of the MATLAB family of products. It integrates with SIMULINK, and enables models to be translated into C code, which is then compiled using a standard compiler. This compiled binary can be used to control the device.

In the case of shake table, the model is prepared in SIMULINK, and then the Real Time Workshop compiles it into a binary using the Microsoft Visual C++. The compiled form has an extension .wcl, which is discussed in the description of WinCon.

The Signal processing toolbox is not essentially required, but it is very useful if some data and signal processing is to be done on the data collected from the accelerometers mounted on the model structure. The Control Systems toolbox is useful if the Active Mass Driver option is to beused, and if further experiments are to be made and compiled. It is required to calculate the gains for the SIMULINK diagram implementing the AMD control algorithm.

3.4 WINCON CONTROL INTERFACE PACKAGE

WinCon control software is the actual interface used to run experiments on the Shake Table. The program incorporates "Real-Time Digital Signal Processing and Control under Windows OS using SIMULINK and TCP/IP technology". WinCon is a real time Windows application that runs SIMULINK generated code using Real Time Workshop on a PC. The software consists of two components, namely WinCon Server and WinCon client.

3.4.1 WINCON SERVER

This is the component that performs the following tasks:

- Converts a Simulink diagram to a WinCon Controller Library file (.wcl) using the Real Time Workshop.
- Compiles and links the code using Visual C++
- Downloads the (.wcl) file to the WinCon client
- Starts and stops WinCon client
- In case of remote accessing, maintains communications with WinCon client
- Makes changes to WinCon client parameters using Control Panels
- Plots the data streamed from WinCon client in real time.
- Saves data

3.4.2 WINCON CLIENT

This is the component that runs the code generated from the SIMULINK diagram. It performs the following tasks:

- Received controller code in the form of a .wcl file from the WinCon Server.
- Runs the controller in real time
- Maintains communication with the connected WinCon Server.
- Streams real time data to the WinCon server.

3.4.3 CONFIGURATION

The WinCon software can be installed in a number of configurations, depending how the apparatus is to be accessed. There are two most common configurations:

1) Single PC

In this configuration, only one PC is used to control the hardware. Both WinCon server and WinCon client are installed on the same computer. This is the most common configuration, and the one being currently used to control the shake table. When ever a WinCon controller library file is loaded by the WinCon server, it is automatically downloaded to the WinCon client without any user intervention. The model is ready to be run.

2) Remote Access

In this configuration the WinCon Server is installed on that PC from which the user wants to access the hardware. It is necessary to install MATLAB, SIMULINK and Visual C++ (if new models are to be generated remotely) on this machine. WinCon client is installed on the computer which is equipped with the data acquisition and control system and is actually connected to the hardware. In this case a communication is established between the two components using the TCP/IP protocol. The WinCon server connects to the WinCon client using the IP address of the client side computer. Once that is done, the WinCon controller library file (.wcl) is loaded by the WinCon server. This then as to be downloade to the WinCon client, to which the connection is already made using TCP/IP. After the controller has been downloaded to the client every operation is the same as for the first configuration

3.4.4 INTEGRATION WITH SIMULINK

MATALAB and SIMULINK should be installed on the computer before installing the WinCon Server. This is necessary because during installation, WinCon integrates itself SIMULINK. Indeed in all the SIMULINK models, the WinCon options appears. This facilitates the smooth integration of WinCon with SIMULINK, and the model can be compiled and linked in one smooth single operation. Infact all the SIMULINK blocks for using the Multi-Q board are installed as well and can be easily accessed for drag and drop operations just similar to the built-in SIMULINK blocks.

Compiling the Model

The SIMULINK model, once prepared and all the required constants put in the workspace, can be easily compiled, linked and converted to the WinCon controller library file (.wcl) from within SIMULINK. This is done by choosing the **WinCon** option from the menu and choose **Build**. This will carry out the compilation, linking, and generation of a wcl. file. The whole process is integrated seamlessly into one single action mentioned above.

3.4.5 CONTROL PANELS

A compiled model can be run using the WinCon server, by opening a WinCon controller Library file (.wcl). Once a model is opened and downloaded on to WinCon client, it can be run by simply invoking the *Start* option. Different parameters can be changed in real time while the experiment is running. To be able to do this, the model must have been compiled with the option *All Parameter Modifiable.* Control panels can be made in WinCon to change the values of variables. These are graphical user interfaces, which can be inserted by choosing the option from WinCon server menu. Once a control has been chosen, the variable of interest can be associated with it.

All the settings once made can be saved in a WinCon project (.wcp format). These saved projects can then be opened in future, to run the model with all the saved settings.

3.5 MICROSOFT VISUAL C++

Microsoft Visual C++ is the compiler that is used to compile the C code generated by the Real Time Workshop. It is necessary to have installed Microsoft Visual C++ ver. 5.0 or ver 6.0 in order to be able to make new real time simulations.

Care should be taken that in the SIMULINK model, under the WinCon option, Simulation parameters are correct. These can be set by choosing the Real-Time Workshop tab under the Simulation parameter.

CHAPTER 4

EARTHQUAKE SIMULATION

4.1 OVERVIEW

In this chapter we take a look at an experiment that simulates an earthquake: the 1995 Kobe earthquake. We look into the SIMULINK diagram closely, discuss the different blocks that are present in the model file: how they function how they link together and work to make the experiment happen in real time. Then a description is given of how the SIMULINK file is converted into an executable form suitable for running using the WinCon interface. We also discuss briefly how the Active Mass Driver can be used to control the structural response "actively". We compare the response recorded from the accelerometer for two simulations, i.e. with and without the Active Mass Driver running, and observe the difference in response.

Lastly, we discuss the very important issue of correcting the accelerometer readings for different artifacts that are introduced due to the instrument characteristics, namely noise and drift. A simple algorithm is presented that eliminates the drift (DC offset) from the acceleration record. The implementation of this scheme is derived in the end.

4.2 THE SIMULINK MODEL

The experiment is prepared in SIMULINK, using the different block-sets that come with the Multi-Q board, and those that are provided by Quanser Consulting Inc. for controlling the table. The following figure shows the complete SIMULINK diagram.

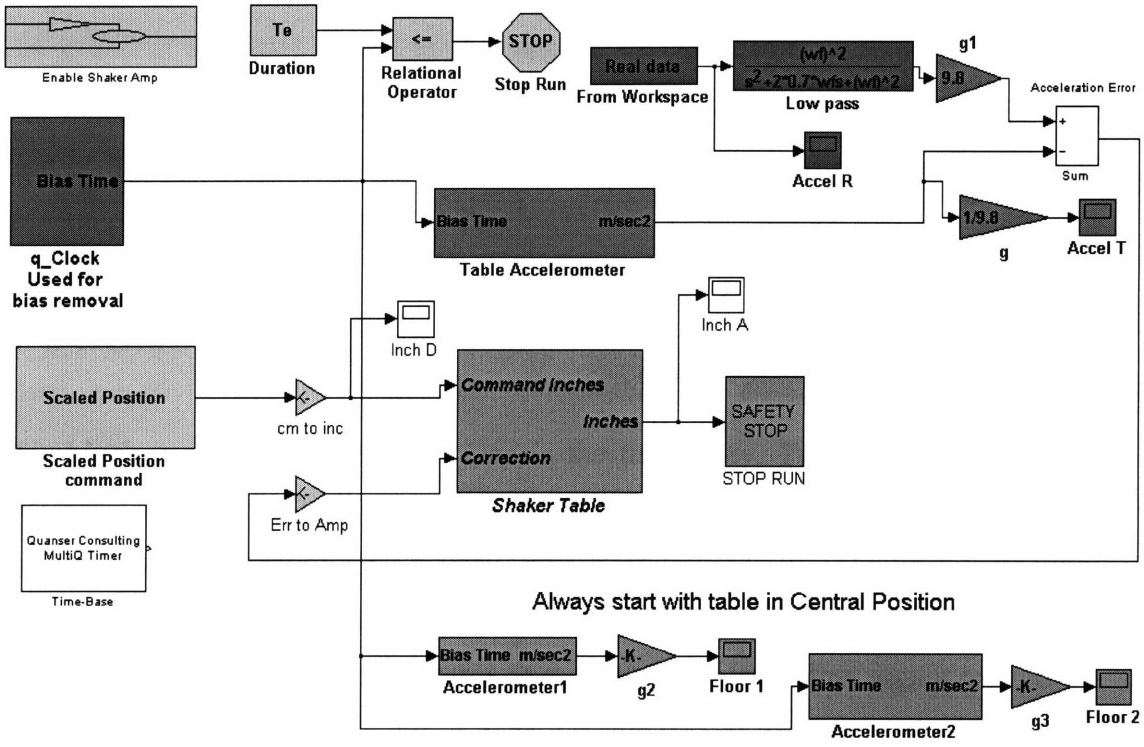

Fig. 4.1: The SIMULINK diagram for earthquake simulation

Given below is the detailed description of what each block does and relates to other blocks.

4.2.1 ENABLE SHAKER AMP

The block basically enables the shake table amplifier by sending signal to analog output #0 of the MultiQ-2 terminal board. This makes it possible for the table to run.

4.2.2 DURATION

The block is actually holds a constant value, Te which is eventually read from the Matlab workspace. This value is the total time duration of the simulation. As soon as Te seconds are passed, the simulation is stopped.

4.2.3 BIAS TIME

The block has a SIMULINK clock, which is used for keeping track of the time duration of the simulation. The time kept by this clock is used for accelerometers and also for comparing against the value of Te.

4.2.4 SCALED POSITION

This block contains two vectors. One is the time array, Tc, and the other is the displacements array, Xc. These displacements are those that are put in the workspace and correspond to the acceleration time history. The displacements are calculated to be input to the table, by scaling down the actual time history of the earthquake, in time. This is done so that the maximum ground (table in our case) displacements do not exceed the stroke length of the table. When this is done, the duration of the earthquake is naturally reduced in order to keep the actual acceleration values. Once this is done, the modified time history can be integrated twice to get the corresponding displacements. These along with the scaled down time are placed in this block. The displacements, after conversion to inches from centimeters, is fed as input to the *Shake Table* block, which we discuss in the following.

4.2.5 REAL DATA FROM WORKSPACE

This block, as the name implies, contains the real data in the workspace. This is the scaled down time history, (scaling down temporally). There are two vectors, Tc, same as in the previous block, and Ac, the accelerations. These accelerations are passed through a low pass filter to throw away frequencies that are higher than the operational bandwidth of the table motor. These values are normalized to *g*, so they are multiplied by 9.81 to get in m/sec^2.

4.2.6 ACCELEROMETER BLOCKS

Only one of the three blocks need to be discussed since they are all the same, except for the different locations where the accelerometers are installed. This block takes acceleration data from Analog outputs of the Multi-Q terminal board, combine it with the time signal from the *Bias Time* to give output as acceleration in m/sec^2.

4.2.7 SHAKER TABLE

There are two inputs to this block. The first is the displacements in inches as computed from the `Xc` displacement array in the workspace, and the other is the difference in the reading of the table accelerometer and the filtered accelerations taken from the `Ac` (the real accelerations). The output of the block is the actual displacement of the table. We look at the details of this block in the diagram below.

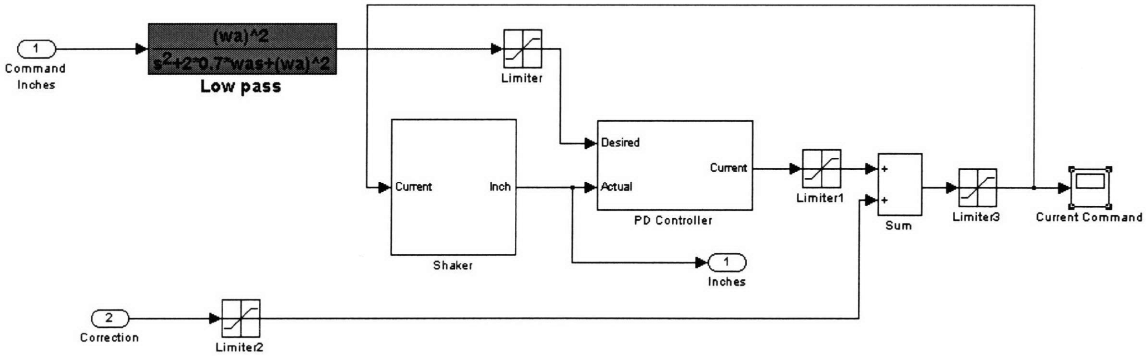

Fig. 4.2: The Shaker Table block

The first input is the displacement in inches read from the workspace. This data is filtered to remove the high frequency component beyond the operational bandwidth of the table. The data, after passing through a limiter (basically setting a limit to the magnitude) becomes the first input to the block *PD Controller* (acronym for Proportional Derivate Controller). This corresponds to the desired position of the table. The second input to *PD Controller* is the actual position of the table. This actual position is the output of the block *Shaker*, and is read from the encoder input #0. The detail of this block is shown in fig 4.3. The second input to *Shaker Table* is the difference in the actual and measure accelerations. The output from the *PD Controller* is then added to this difference and after passing through usual limiter blocks this is taken to the be current desired position (after taking into account the previous errors). The *current command* is given as the input to the *Shaker* block. This is read by the Multi-Q and the values passed onto the table using Analog Output. Shown below is the detail of this block.

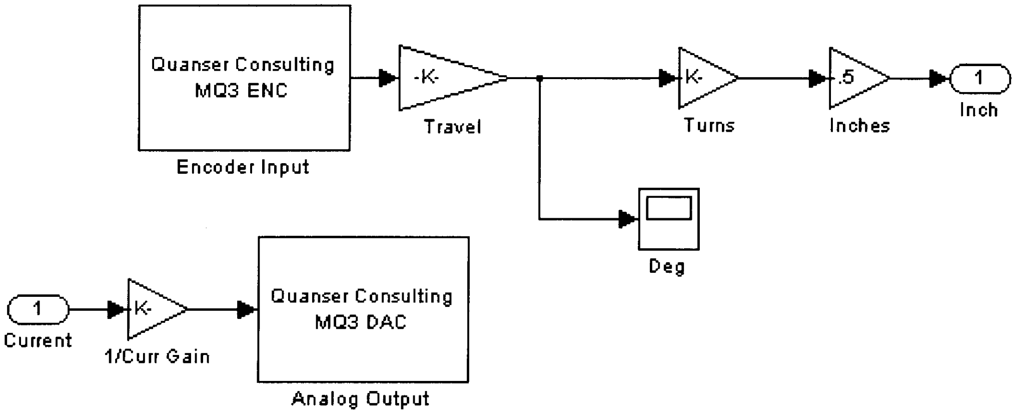

fig. 4.3: The detail of Shaker *block, showing table position being read by* Encoder Input *and current position sent via* Analog Output.

4.3 RUNNING THE MODEL

The SIMULINK model needs to be compiled to be able to run in real time on the shake table. This is accomplished by using the Real Time Workshop, to generate C code. This code is then compiled using Microsoft Visual C ++. This code talks to the data acquisition and control system, to send and receive signals to and from shake table. The whole process is done in once smooth operation. But before this can be done, the acceleration time history of the earthquake that is to be simulated must be present in the MATLAB workspace. As discussed earlier, the earthquake is to be scaled down spatially, and in order to keep the accelerations the same as that of original earthquake, there has to be a corresponding time scaling.

4.3.1 SCALING THE EARTHQUAKE

The the function `q_scale` is used to scale down the time history. This function comes with WinCon and has the following arguments and outputs:

```
[Tc, Xc, Ac, Te]  =  q_scale (t, a, xmas)
```

`t`	array containing the time values at equal samping intervals in seconds, and
`a`	array containing the acceleration values in g's corresponding to the t.
`xmas`	maximum displacement of the table from zero position, in centimeters.

The function returns the followng:

`Tc`	Command time array
`Xc`	Position command array
`Ac`	Acceleration array, obtained by differentiating `Xc` twice. This should exactly match with the input array a.
`Te`	Duration of the simulation

`Tc` and `Xc` are the time and displacements corresponding to scaled down time history. But since scaling is carried out both spatially and temporally, the acceleration Ac matches exactly with the array `a`. The scaling operation is done below for the earthquake that struck Kobe, Japan on January 17, 1995. The North-South component recorded at Kobe Japanese Meteorological Agency (JMA) station, is used. The magnitude on the Richter scale was 7.2. This is shown in fig. 4.4.

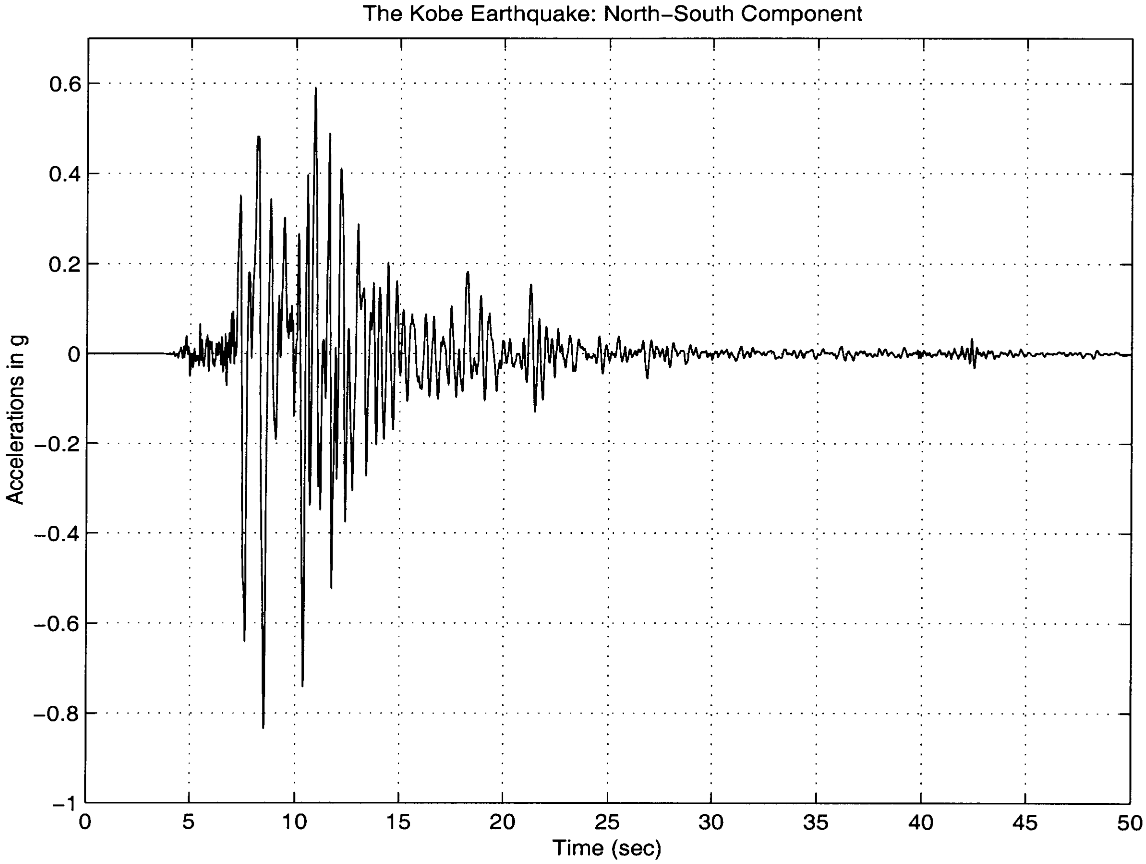

Fig. 4.4: The North-South component of Kobe Earthquake

The total duration of the earthquake record is about 50 seconds, and the maximum acceleration is about 0.82 g. This earthquake is to be scaled down now, for running on the shake table. A safe value of 3 cm is chosen for `xmas`. The actual earthquake accelerations and time values are stored in arrays `a` and `t`, as discussed. After passing these as arguments to the function `q_scale`, we get the scaled down earthquake record. This is plotted in fig. 4.5 and 4.6,

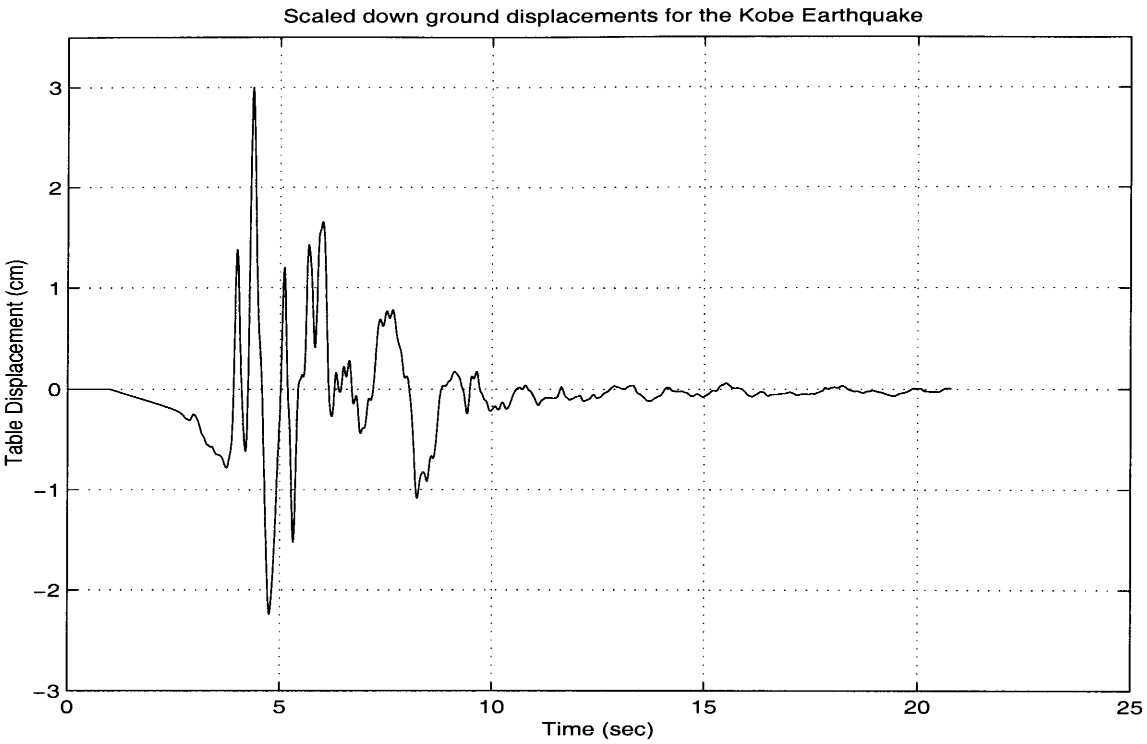

Fig.4.5: The computed displacements for a max displacement of 3 cm

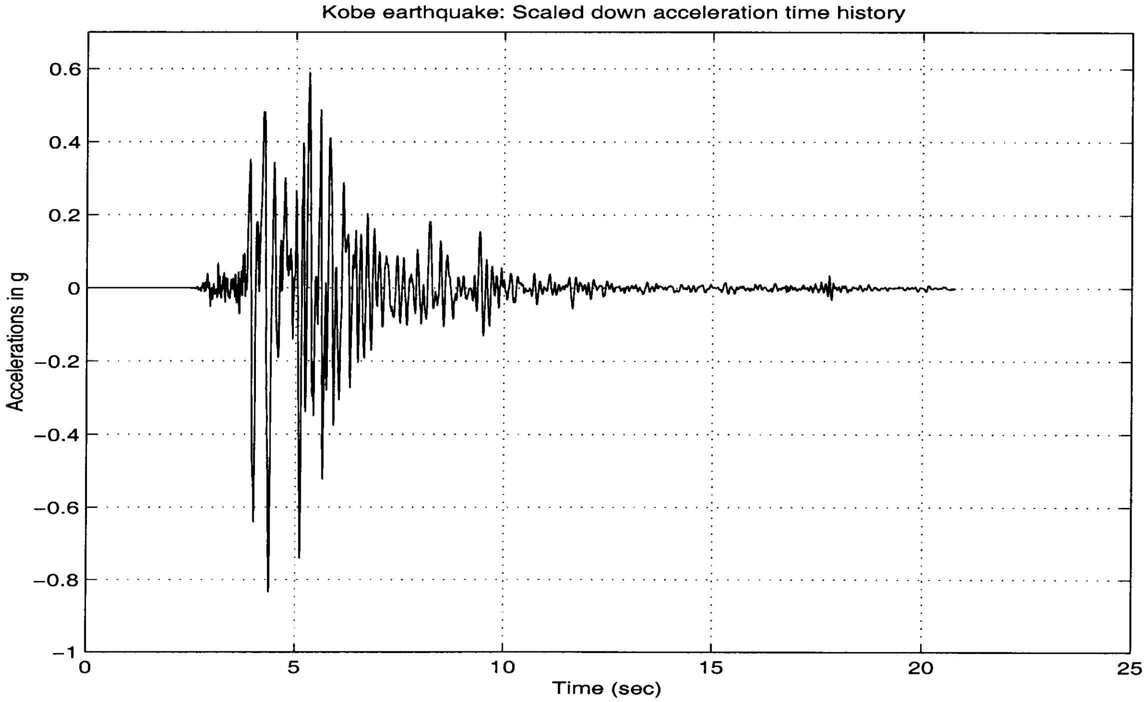

Fig 4.6: The accelerations corresponding to the displacements in fig. 4.5

For fig. 4.6, it is clear that the accelerations have not changed. But the maximum duration of the record as come down to about 22 seconds.

4.3.2 COMPILING THE MODEL

Once the scaled down earthquake record, in terms of displacements, is in the workspace, the earthquake can be compiled. Before that some gain values (constants) for the PD controller must be put in the workspace. These values, along with the earthquake data are picked from workspace by the SIMULINK model. These are computed by running a MATLAB script provided by Quanser. This computes the required gains. The command to run this is `q_gain`.

To compile the model, choose the option *Wincon* → *Options*. Care should be taken to ensure that under the Solver tab, Fixed Step size is selected. Now the *build* option can be chosen. This will start the compilation process, and may take a few seconds. Once the compilation is complete, the WinCon server loads up automatically and the model is ready to run.

4.3.3 RESPONSE FROM ACCELEROMETERS

The response from the accelerometers mounted on the model structure can be observed and recorded. Plot windows can be opened from WinCon and the response observed in real time. Later the response can be saved either to MATLAB workspace, or to a MAT file. The advantage of saving the response is that it can be later plotted in MATLAB environment, processed and analyzed. The figures 4.7 and 4.8 show the response from the intermediate and top floor accelerometer.

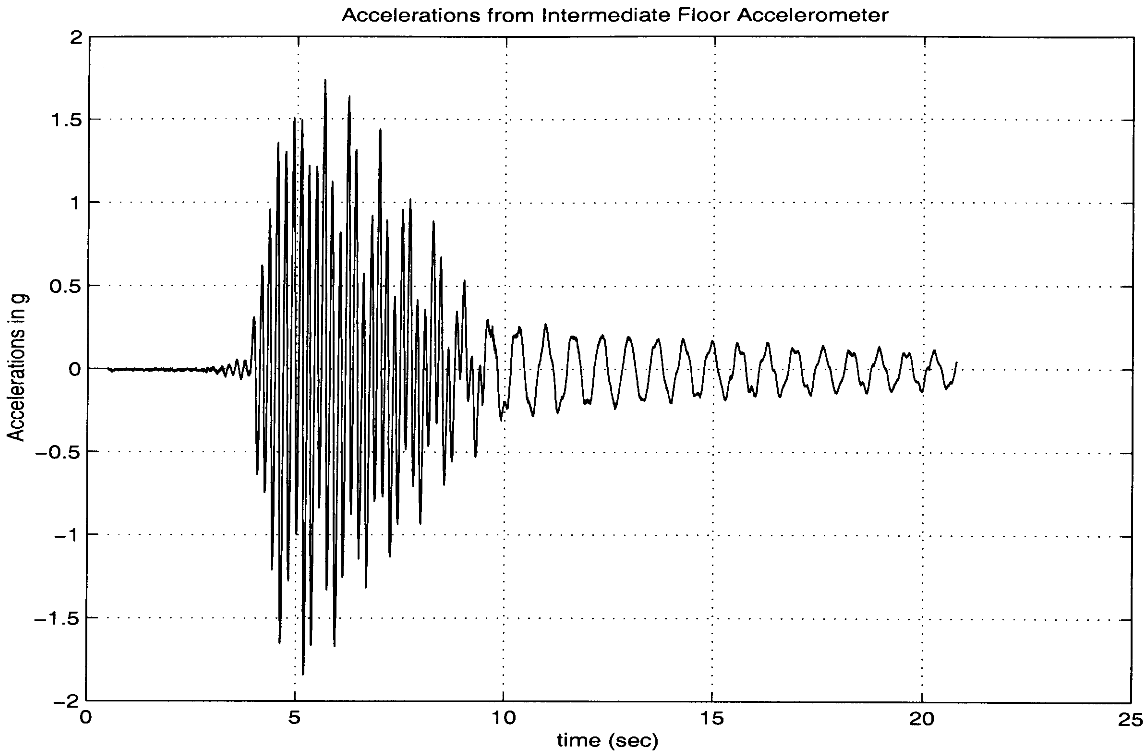

Fig. 4.7: Accelerations from intermediate floor of the model structure

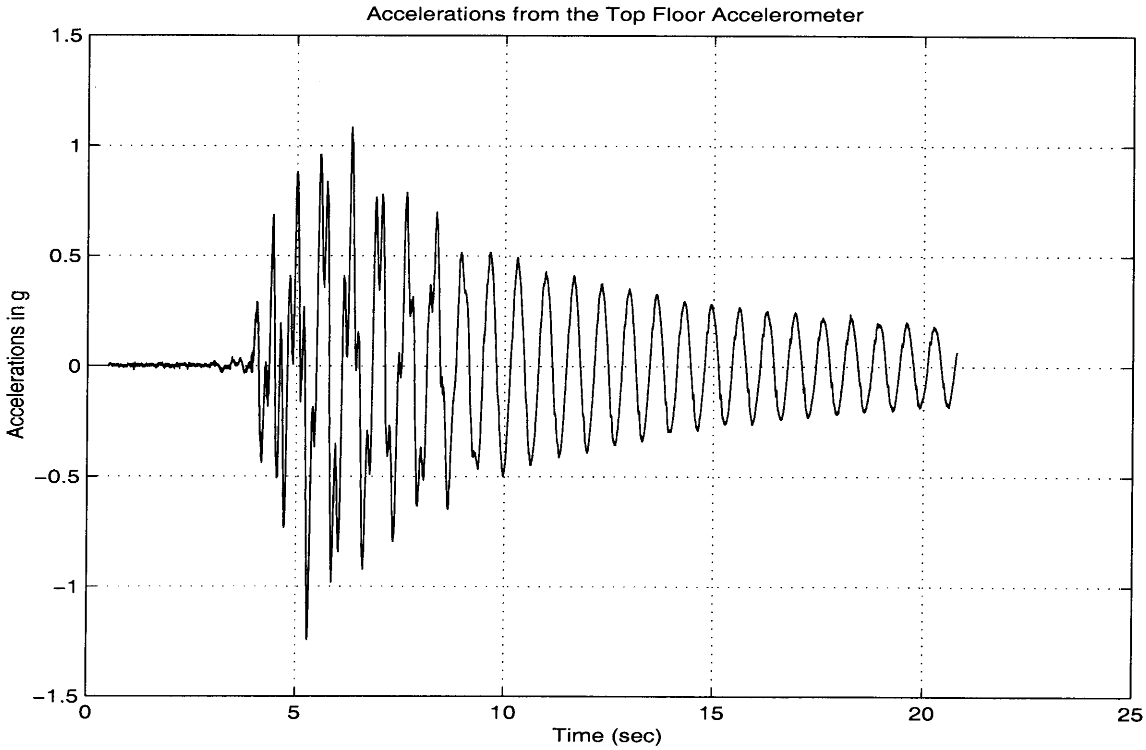

Fig. 4.8: Accelerations from the top floor of the model structure

4.4 ACTIVE MASS DRIVER

Traditionally the earthquake resistant design philosophy for civil structures, like multistory buildings, has relied on the ability of the structural material to dissipate energy during seismic loading. This depends on the development of plastic zones at appropriate locations in the structure. This philosophy, although quite successful, does permit some large displacements e.g. inter story drifts, to occur. This produces large amount of yielding in steel structures, and extensive cracking in concrete.

Active damping is one of the solutions to overcome the problems of large displacements. The whole idea hinges on feedback control theory and predictive control of structure. The estimations (predictions) are calculated on the basis of the previous response measured, resulting in a non-linear feedback-control loop.

The data acquisition and control board can be used in conjunction with an Active Mass Driver, as already mentioned in chapter 2. The driver is a small cart, with mass screwed on top of it, resting on the top floor of the model structure. It runs on a rack and pinion system, the number of revolutions of the cart wheel motor being read by Encoder # 1. This allows the exact position of the cart with respect to the floor to be calculated. Based on accelerations read from the top floor, new displacements along with speed are calculated by the control algorithm. This signal is sent via Analog output #1. Due to the movement of the cart of rack, a reaction is developed, this reaction (a force) tends to balance out the dynamic forces that are acting to produce accelerations. The SIMULINK diagram for this is shown in figure 4.9

4.4.1 SIMULINK BLOCKS

The SIMULINK diagram shown in fig 4.9 is used for controlling the Active Mass Driver. This diagram can be easily added to any existing earthquake model, to perform active control for the structural response during seismic excitation.

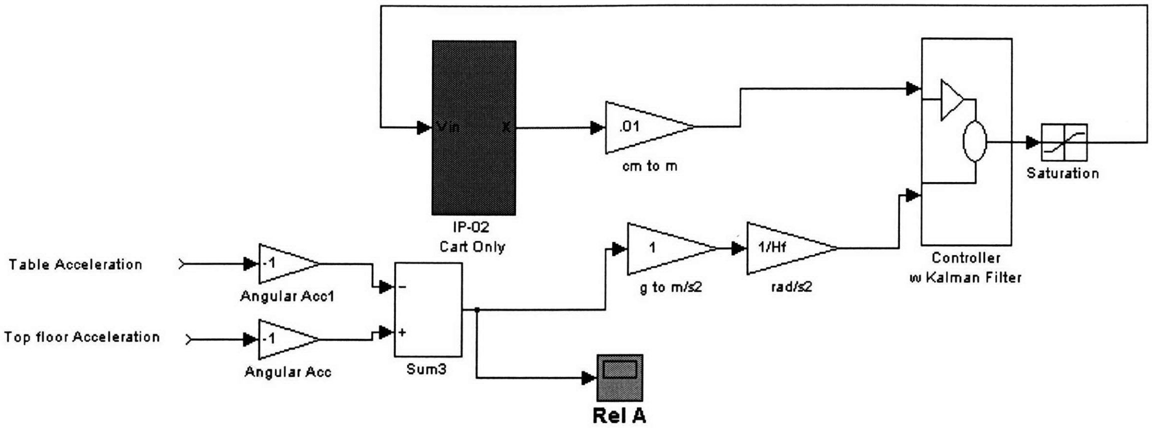

Fig. 4.9: SIMULINK blocks for AMD

Accelerations from the table accelerometer and the top floor accelerometer are taken and converted to angular accelerations. The difference of the two values is taken and sent to the Kalman filter (non linear predictive filter). The other input to the filter block is the position of the cart. This is the output of the block *IP-02*. Estimate of the desired cart velocity is made and the value sent to the the block *IP-02* as input. We see the performance of the control algorithm in figures 4.10 and 4.11. These show the intermediate and top floor acceleration response with the AMD working during the Kobe earthquake simulation, along with the response recorded without the AMD for comparison.

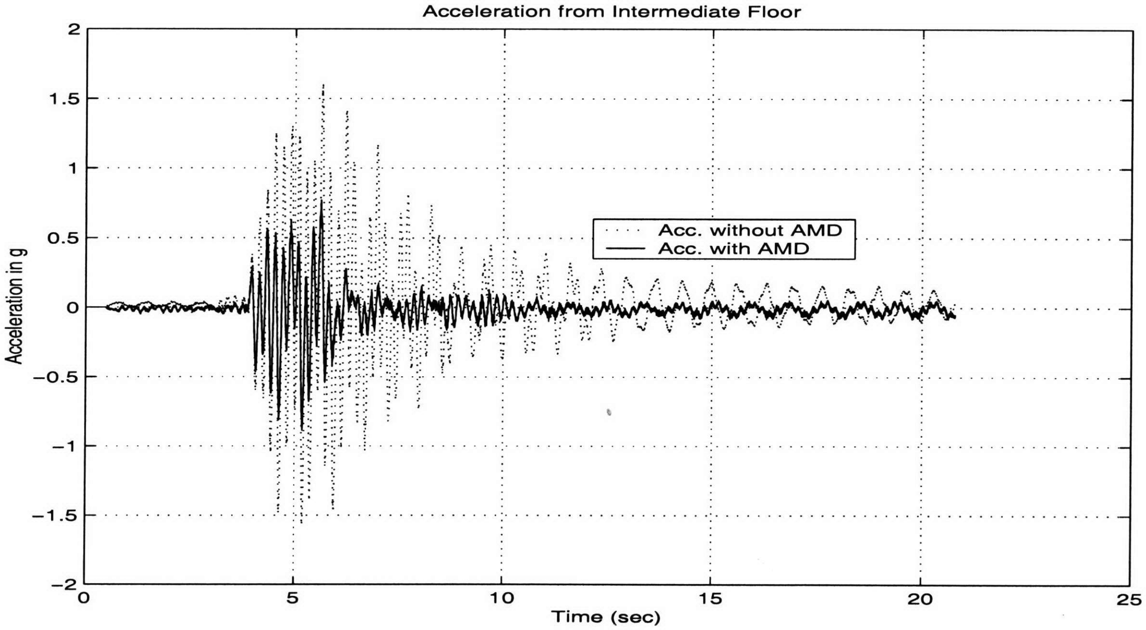

Fig. 4.10: Acceleration response of intermediate floor, with the Active Mass Driver working.

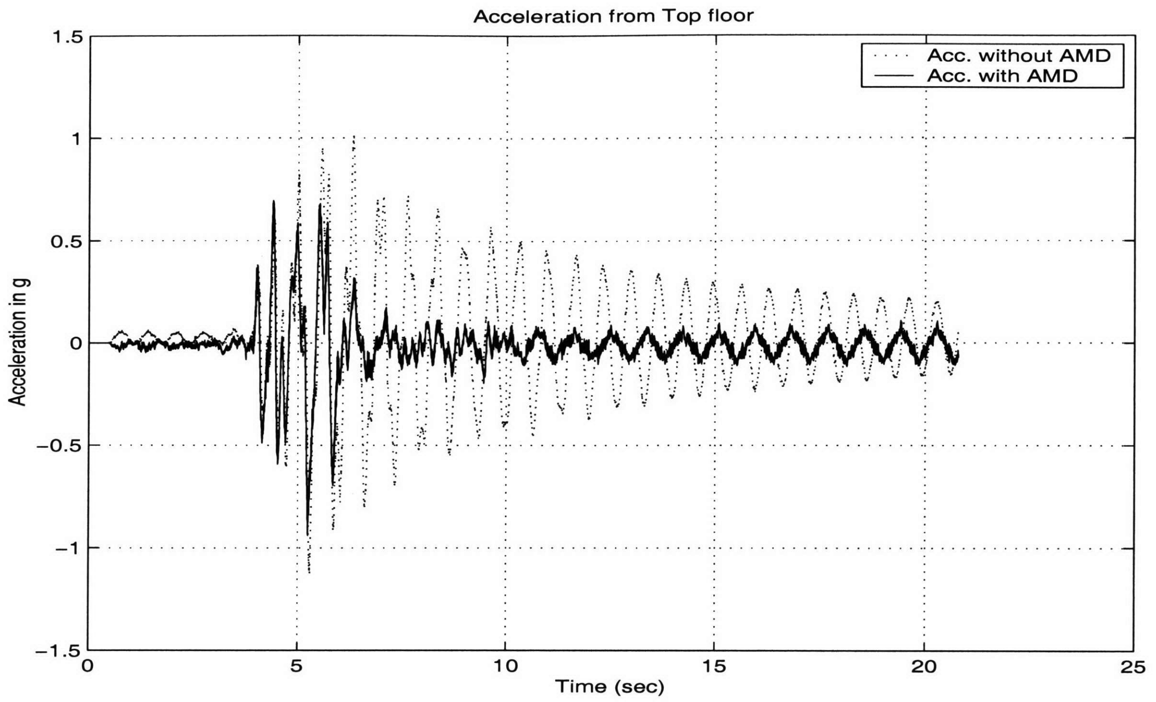

Fig. 4.11: Acceleration response of the top floor with the Active Mass Driver working.

For the intermediate floor, the response initially is not much less than what it is for uncontrolled structure. But once the impulse component is passed, the reduction is much more pronounced: the maximum acceleration is about 0.2 g as compared to about 1.25 g when the AMD was not used. Similarly for the top floor the performance of AMD for an initial short period is not very good. This short period is the "response time" of the controller, during which the acceleration response is virtually the same as when the AMD was not working. Once the impulsive component of the earthquake has passed, there is a marked reduction in response, with max acceleration about 0.25 *g* as compared to 0.75 *g* without the AMD.

4.4.2 DIFFICULTIES IN ACTIVE CONTROL

Active control of structures looks attractive when the results for small scale structures are studied. But the fact remains that the methodology is still far from applicable in real structures, at least on a scale to actually play a major role in earthquake resistance of the structure. For large massive structures, large masses are required for effective active control, and for those large masses, a huge amount of power is needed to actuate them within a reasonable short period of time. The response time of the system should be as small as possible, and that requires a lot of power. For this reason active control though effective in theory, has not been able to find a place in regular earthquake engineering practice.

The other problem associated with active control is that that of noise in the data sensing and recording system. For large magnitude of response, this noise can be muffled, but once the force of excitation has died out the noise begins to dominate the actual data. For the model structure that was tested, this actually had a detrimental effect, because the control system began to respond to the noise, and actually began to excite the structure. This kind of problem can be overcome by the use of effective real time filtering of the data recorded.

4.5 BASELINE CORRECTION

Fig. 4.12 shows the frequency spectrum of the data collected from intermediate floor accelerometer, for the Kobe earthquake simulation.

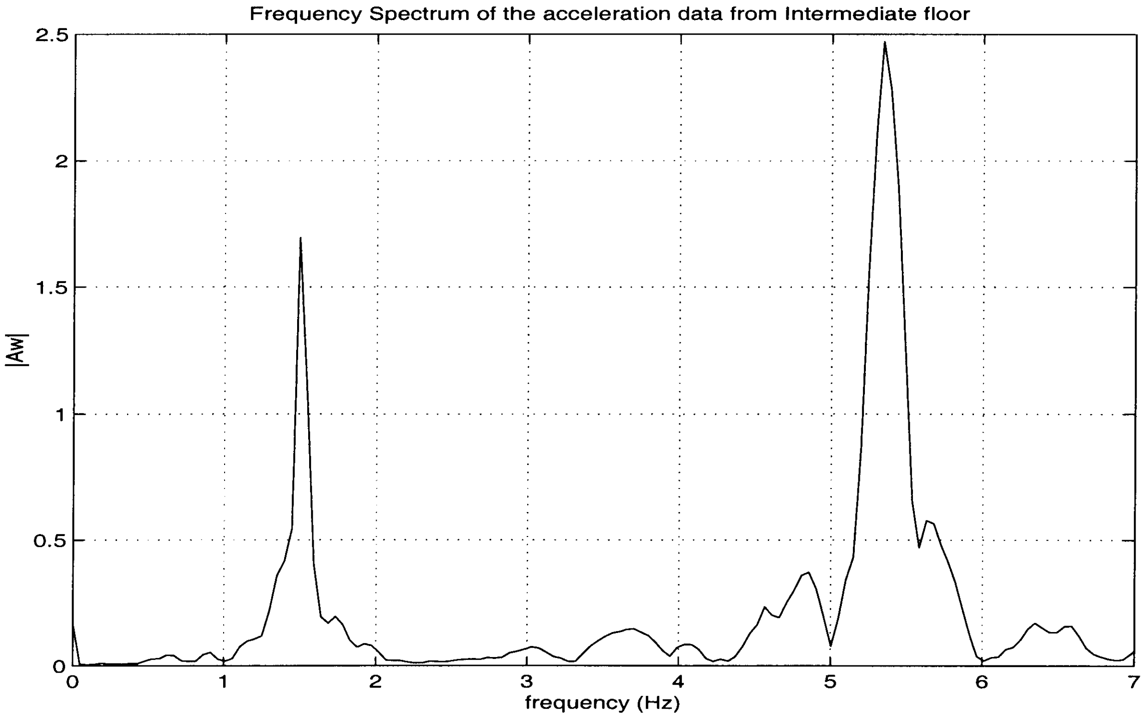

Fig. 4.12: The frequency Spectrum of acceleration from intermediate floor

The first and second modal frequencies can easily be detected from this curve. But it is interesting to note that for zero frequency, the magnitude of the Fourier coefficient is non-zero. Physically that does not make sense, as the implication is that the structure is moving in one direction with some constant acceleration. Of course this is not the case. This is an artifact introduced into the data due to the instrument itself, and is known as the *drift of the accelerometer*. This drift effect can accumulate in the instrument over time. Accelerometers usually have an option of adjustment against this DC offset, but still this drift can be observed in instruments that are corrected for this. All the acceleration records from seismological stations are corrected for this. Here a simple algorithm is shown that works very well for data that is collected from accelerometers that are mounted on vibrating structures, i.e. the accelerations do not die out immediately after the ground has come to rest.

4.5.1 PROCEDURE FOR BASELINE CORRECTION

The baseline correction procedure that is followed is based on minimizing the sum of integral of velocities. This corresponds to minimizing the energy. We proceed to develop the algorithm as follows:

Let

a = Measured Accelerations

$\bar{a}$ = Corrected Accelerations

then

$v = \int a \, dt$; velocities from uncorrected accelerations

$\bar{v} = \int \bar{a} \, dt$; velocities from corrected accelerations

We apply a parabolic correction to the actual accelerations,

$$\bar{a} = a + At^2 + Bt + C \quad ; \qquad \text{where } t \text{ is time}$$

$$\bar{v} = v + \frac{1}{3}At^3 + \frac{1}{2}Bt^2 + Ct + D$$

$$v(0) = \bar{v}(0) = 0 \qquad ; \qquad \text{assuming the structure to be at rest initially}$$

$$\Rightarrow \quad \bar{v} = v + \frac{1}{3}At^3 + \frac{1}{2}Bt^2 + Ct$$

The given condition is

Minimize the integral $\quad I = \int \bar{v}^2\,dt$

$$\Rightarrow \quad \frac{dI}{dA} = 0; \qquad \frac{dI}{dB} = 0; \qquad \frac{dI}{dC} = 0$$

$$\frac{dI}{dA} = \frac{d}{dA}\int_0^T \bar{v}^2\,dt = \int_0^T 2\bar{v}\,\frac{d\bar{v}}{dA}\,dt = 0$$

$$\frac{dI}{dA} = \int 2\left(v + \frac{1}{3}At^3 + \frac{1}{2}Bt^2 + Ct\right)\frac{t^3}{3}dt \quad ; \qquad \text{where } \frac{d\bar{v}}{dA} = \frac{t^3}{3}$$

$$\Rightarrow \quad \frac{2}{3}\int_0^T vt^3\,dt + \frac{2}{63}AT^7 + \frac{1}{18}BT^6 + \frac{2}{15}CT^5 = 0$$

$$\Rightarrow \quad \boxed{\frac{2}{63}AT^2 + \frac{1}{18}BT + \frac{2}{15}C = J_1 \quad ; \qquad \text{where } J_1 = -\frac{2}{3T^5}\int_0^T vt^3\,dt} \qquad (\text{I})$$

Similarly

$$\frac{dI}{dB} = \frac{d}{dB}\int_0^T \bar{v}^2\,dt = \int_0^T 2\bar{v}\,\frac{d\bar{v}}{dB}\,dt = 0$$

$$\frac{dI}{dB} = \int 2\left(v + \frac{1}{3}At^3 + \frac{1}{2}Bt^2 + Ct\right)\frac{t^2}{2}\,dt \quad ; \qquad \text{where } \frac{d\bar{v}}{dA} = \frac{t^2}{2}$$

$$\Rightarrow \quad \int_0^T vt^2\,dt + \frac{1}{18}AT^6 + \frac{1}{10}BT^5 + \frac{1}{4}CT^4 = 0$$

$$\Rightarrow \quad \boxed{\frac{1}{18}AT^2 + \frac{1}{10}BT + \frac{1}{4}C = J_2 \quad ; \qquad \text{where } J_2 = -\frac{1}{T^4}\int_0^T vt^2\,dt} \qquad (\mathrm{II})$$

$$\frac{dI}{dC} = \frac{d}{dC}\int_0^T \bar{v}^2\,dt = \int_0^T 2\bar{v}\,\frac{d\bar{v}}{dC}\,dt = 0$$

$$\frac{dI}{dB} = \int 2\left(v + \frac{1}{3}At^3 + \frac{1}{2}Bt^2 + Ct\right)t\,dt \quad ; \qquad \text{where } \frac{d\bar{v}}{dA} = t$$

$$\Rightarrow \quad 2\int_0^T vt\,dt + \frac{2}{15}AT^5 + \frac{1}{4}BT^4 + \frac{2}{3}CT^3 = 0$$

$$\Rightarrow \quad \boxed{\frac{2}{15}AT^2 + \frac{1}{4}BT + \frac{2}{3}C = J_3 \quad ; \qquad \text{where } J_3 = -\frac{2}{T^3}\int_0^T vt\,dt} \qquad (\mathrm{III})$$

Hence from these three conditions, we have three equations, with three unknowns, *A*, *B* and *C* to solve for. These can be written in matrix form as

$$\begin{bmatrix} \frac{2}{63} & \frac{1}{18} & \frac{2}{15} \\ \frac{1}{18} & \frac{1}{10} & \frac{1}{4} \\ \frac{2}{15} & \frac{1}{4} & \frac{2}{3} \end{bmatrix} \begin{bmatrix} AT^2 \\ BT \\ C \end{bmatrix} = \begin{bmatrix} J_1 \\ J_2 \\ J_3 \end{bmatrix}$$

The solution of these three simultaneous equations gives the following result:

$$A = \frac{1}{T^2}[\,7087.5\,J_1 - 6300\,J_2 + 945\,J_3]$$

$$B = \frac{1}{T}[\,-6300\,J_1 + 5760\,J_2 - 900\,J_3]$$

$$C = 945\,J_1 - 900\,J_2 + 150\,J_3$$

where

$$J_1 = -\frac{2}{3T^5}\int_0^T v t^3\,dt; \qquad J_2 = -\frac{1}{T^4}\int_0^T v t^2\,dt \;; \qquad J_3 = -\frac{2}{T^3}\int_0^T v t\,dt$$

The method is implemented in MATLAB, and the related code is given in the appendix. The procedure was applied to the data already observed to get a frequency spectrum which is free of any DC offset (drift). This is shown in fig. 4.13.

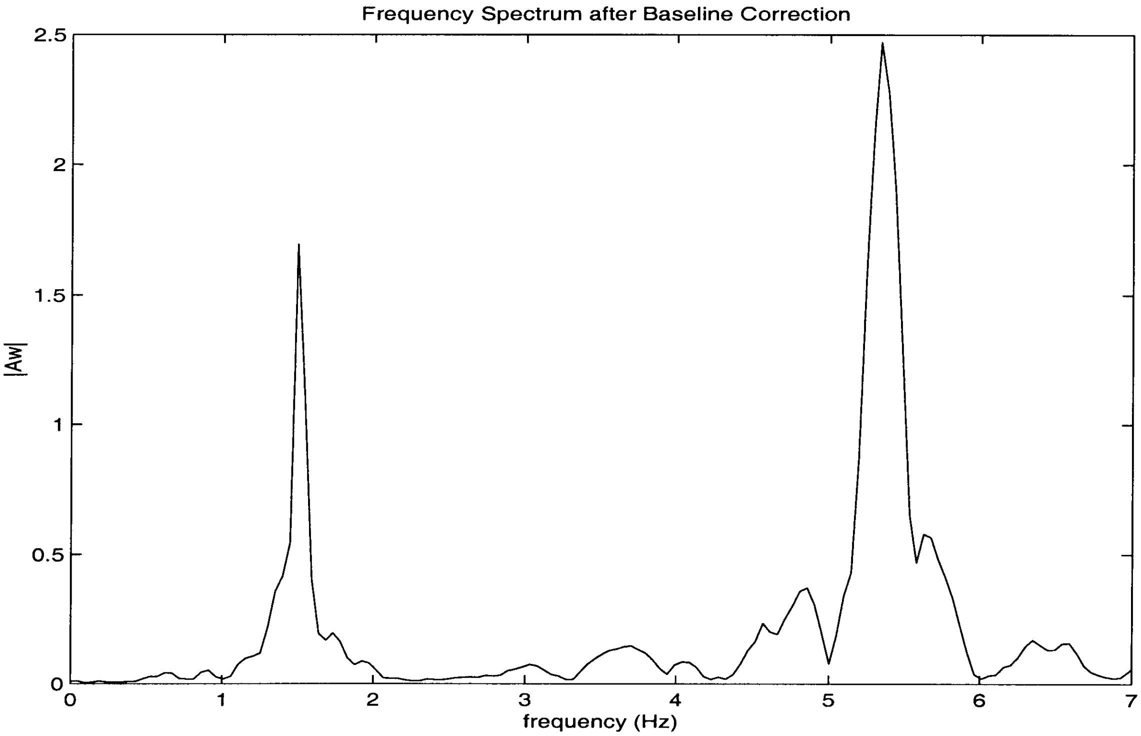

Fig. 4.13: Frequency spectrum of Acceleration data from intermediate floor, after Baseline correction

CHAPTER 5

AN APPLICATION: ESTIMATION OF DAMPING

5.1 OVERVIEW

Sensors are attached to civil engineering structures, to monitor the response due to various external loads. Accelerometers are typically used to record the response of the structure due to dynamic loads like earthquakes, wind, explosions, etc. The recorded signal can have various components, depending on the structure, namely the degrees of freedom. In addition there can be electrical noise from the data acquisition setup, DC offset caused by "drift" in the accelerometers, etc.

The aim of this chapter is to demonstrate the usefulness of shake table to carry out structural dynamics experiments to gain insight into behavior of structures and to explore methods to do so. Wavelet decomposition is carried out on a signal from an accelerometer mounted on the model structure, to separate out the relevant signal corresponding to the degrees of freedom of the structure, and to filter out the noise that corrupts the signal. Moreover, damping is estimated for the different modal components which are isolated. We start out with a brief discussion of damping and then go on to perform discrete wavelet transform, to extract modal components and determine the damping.

5.2 DAMPING

We consider a single degree of freedom structural system, with mass m, stiffness k and damping constant c. The response of the system is the displacement u. The governing differential equation is :

$$m\frac{d^2u}{dt^2} + c\frac{du}{dt} + ku = p(t)$$

In the above equation $p(t)$ is the forcing function. For a free vibration case this is equal to zero. The above equation can be written as :

$$\frac{d^2u}{dt^2} + 2\xi\omega\frac{du}{dt} + \omega^2 u = 0$$

where ω is the natural frequency, and ξ is the damping ratio of the system. They are defined as,

$$\omega = \sqrt{\frac{k}{m}} \qquad \text{and} \qquad \xi = \frac{c}{2m\omega}$$

The solution for the above differential equation is then, subject to the initial conditions $u(0)$ = A and $\frac{du(0)}{dt} = 0$, is

$$u = A.e^{-\xi\omega_d t}\cos(\omega_d t) \text{ where } \quad \omega_d = \omega\sqrt{1-\xi^2}$$

$\omega_d \approx \omega$ (for all practical purposes in structural eng.)

For multi degree of freedom structures, we use modal mass, stiffness, and frequency, in the above equation for each mode. The modal mass can be obtained by pre and post multiplying the mass matrix with the modal vector. The same is true for modal stiffness

If

M = Mass matrix

K = Stiffness matrix

C = Damping matrix and

Φ = Matrix with modal vectors as columns,

then

$\Phi^T \mathbf{M} \Phi = diag(\mu_j)$ where μ_j = jth modal mass

$\Phi^T \mathbf{K} \Phi = diag(\kappa_j)$ where κ_j = jth modal stiffness

$\Phi^T \mathbf{C} \Phi = diag(\eta_j)$ where η_j = jth modal damping

and **C** is assumed to be proportional

Modal damping ratio is then defined as for single degree of freedom structures, with modal mass, stiffness and damping using instead;

$$\xi_j = \frac{\eta_j}{2\sqrt{\mu_j \kappa_j}}$$

5.2.1 DETERMINING DAMPING RATIO

We consider a simply harmonic function given by the solution given on the last page. The envelope of the function is given by $Ae^{-\xi\omega t}$. So if we have this function we can determine the value of ξ, provided we know the frequency ω. This is done as following:

Let

$$f = Ae^{-\xi\omega t}$$

$$\log f = \log A - \xi\omega t$$

If we know the function f for the whole range then we also know $\log A$. Hence the damping ratio ξ is only the negative of the slope of the above straight line, divided by the frequency (which is known).

5.2.2 DETERMINING THE ENVELOPE FUNTION

Consider a function *f(t),* whose Fourier transform is *F(ω).* Then the Hilbert Transform of a function is defined as,

$$\mathfrak{H}(f(t)) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)\, i\, \mathrm{sgn}(\omega)\, e^{i\omega t}\, d\omega$$

where $$\mathrm{sgn}(\omega) = \begin{cases} +1 \ ; \ \omega > 0 \\ -1 \ ; \ \omega < 0 \end{cases}$$

We are considering here real input signal *f(t),* hence its Fourier transform *F(ω)* is a complex function. Multiplying *F(ω)* with i sgn(ω) still keeps the product complex, and hence taking its inverse Fourier transform gives a real function. What it does essentially is to produce a phase shift of 90 degrees to the input signal. Hence, it gives sines if cosines are given in, and vice versa.

Consider a harmonic signal *f(t), decaying exponentially,*

$$f(t) = Ae^{-\xi\omega t}\cos(\omega t); \qquad \text{then}$$

$$\mathfrak{H}(f(t)) = Ae^{-\xi\omega t}\sin(\omega t)$$

If we square the above two relations and take square root we are left with the envelope function.

The above procedure has been implemented using the `Hilbert` function in `Matlab.`
If given a real function as an argument, it returns a complex function, the real part of which is the real input, and the imaginary part is the Hilbert transform of the input.

5.3 THE SIGNAL ANALYSED

The signal that was analyzed was data recorded from an accelerometer mounted on the top floor of the model structure (fig. 5.1). The structure was a 2-degree-of-freedom system, and the response recorded was that for free vibration. Considering the structure, two modal frequencies are of interest, and those are expected to have most of the energy of the signal. The signal was corrected for any DC offset before being used for this project.

A Fast Fourier transform was carried out to find out the modal frequencies. Based on these frequencies, it was determined up to what level the wavelet decomposition is required. The following figure shows the frequency spectrum of the input signal. Note only the frequency range of interest is shown. ***The sampling rate of the signal was 1 kHz, and hence the Nyquist frequency was 500 Hz.***

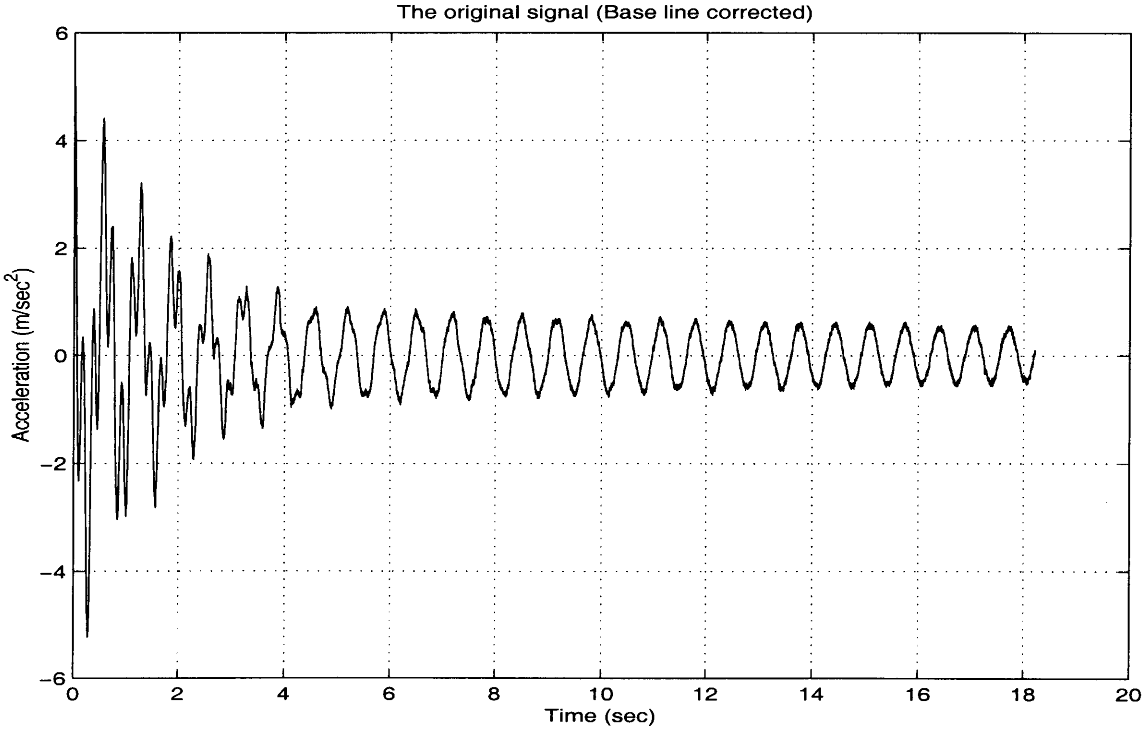

Fig. 5.1: The original signal to be analyzed

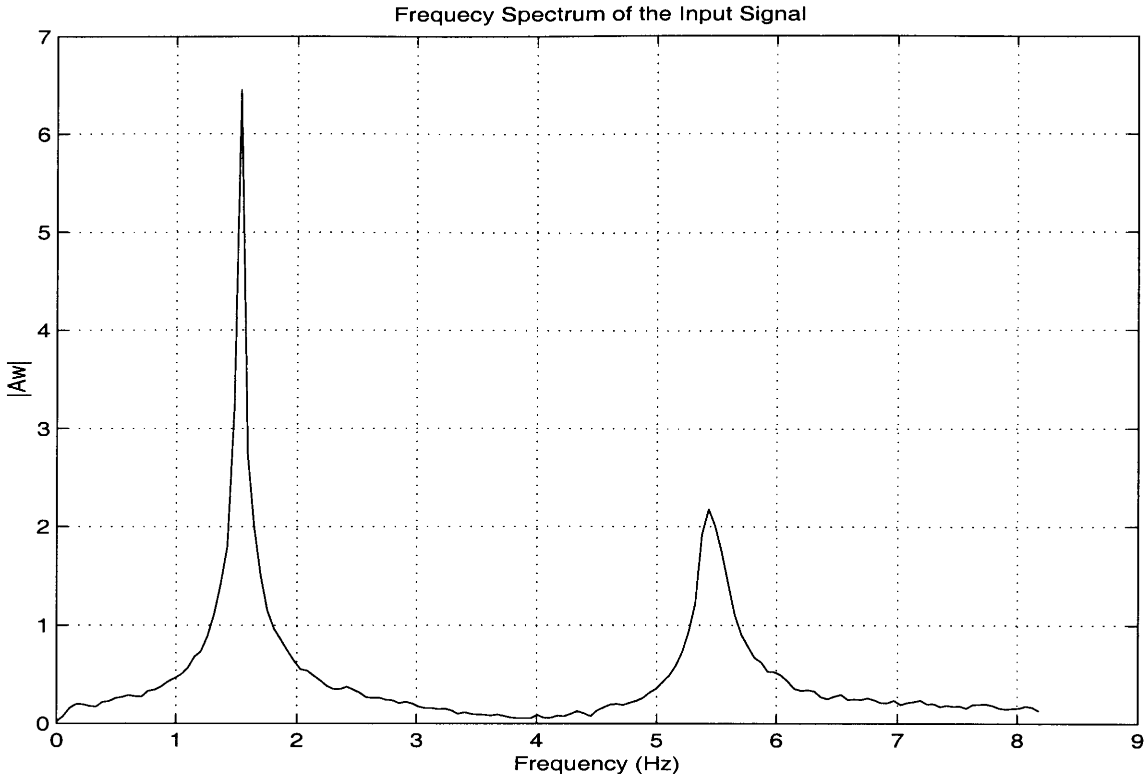

Fig. 5.2: The frequency spectrum of the signal

From fig. 5.2 above, we can see the two dominant frequencies, corresponding to the two modes of the structure.

The first modal frequency ω_1 = 1.537 Hz

The second modal frequency ω_2 = 5.434 Hz

5.4 WAVELET DECOMPOSTION AND EXTRACTION OF DAMPING RATIO

5.4.1 WAVELET DECOMPOSITION USING FILTERBANK

Wavelet decomposition can be effectively carried out using a filter bank. We assume for the moment, that the filters are ideal. The goal in a filter bank (2 channel) is to decompose the signal into low frequency component and high frequency component.

This can be carried out by passing the signal through the low and high pass filters simultaneously. This gives us the low frequency and high frequency components, the length equal to that of the actual signal. We overcome this by downsampling the signals by 2, i.e. we drop every other component of the signal. This cuts down the length of both the signals in half. At the same time, this also brings down the Nyquist frequency of both signals by half. There are two advantages in this. First, the quantity of data remains the same, and second, making the Nyquist frequency of the components half allows us to pass the signal from the low and high pass filter again. Doing this repeatedly we can get finer and finer resolution of the signal. The low pass coefficients are the *approximation coefficients*, i.e. they approximate the signal, without having the high frequency, finer detail. Similarly the high pass coefficients are the *detail coefficients,* i.e. they contain all the fine high frequency details of the signal. At each level, the approximation coefficients represent contents with frequencies from the lower half of the frequency spectrum of the previous level approximation coefficients. Similarly the detail coefficients at each level represent contents with frequencies from the higher half of the frequency spectrum of the previous level approximation coefficients.

The second of the filter bank is the synthesis part. Here the coefficients are first *upsampled* and then passed through low and high pass filters and finally adding the two components. This is done on as many levels as on which the analysis was carried out.

The following figure shows a simple implementation of a filter bank

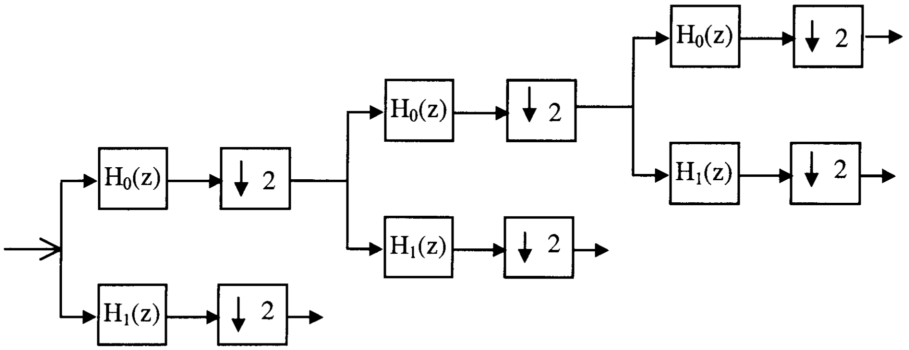

Fig. 5.3(a): Standard Filterbank Implementation: The Synthesis bank

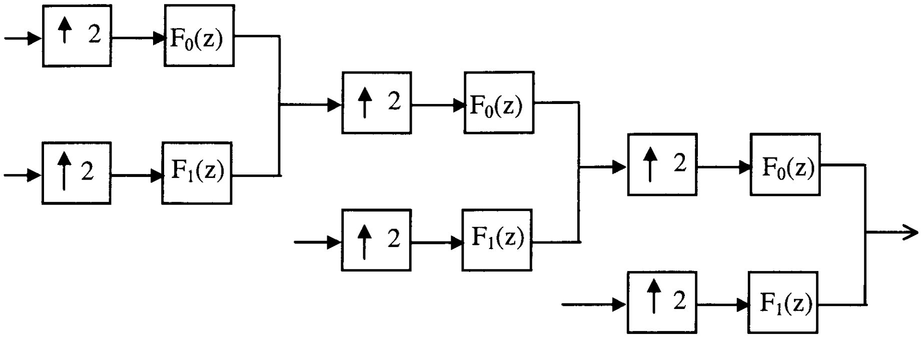

Fig. 5.3(b): Standard Filterbank Implementation: The Analysis bank

Even though we are down sampling at every level, still if all the coefficients are preserved, the original signal can be reconstructed perfectly. This is achieved by taking care that the analysis filters are related to the synthesis filters by the following relations:

$$F_0(z)H_0(-z) + F_1(z)H_1(-z) = 0$$

$$F_0(z)H_0(z) + F_1(z)H_1(z) = 2z^{-l}$$

where *H(z)* and *F(z)* are the z-transforms of the analysis and synthesis filters, and *l* is the delay. These are called the conditions for perfect reconstruction. As long as these are met, we have a perfect reconstruction filterbank.

Since the Nyquist frequency is 500 Hz, we need to do a seven level decomposition (assuming ideal filters). The discrete Meyer filter was used for this purpose, since this gave optimal frequency separation for this case. (Ref: Wavelet Toolbox, Mathworks Inc.)

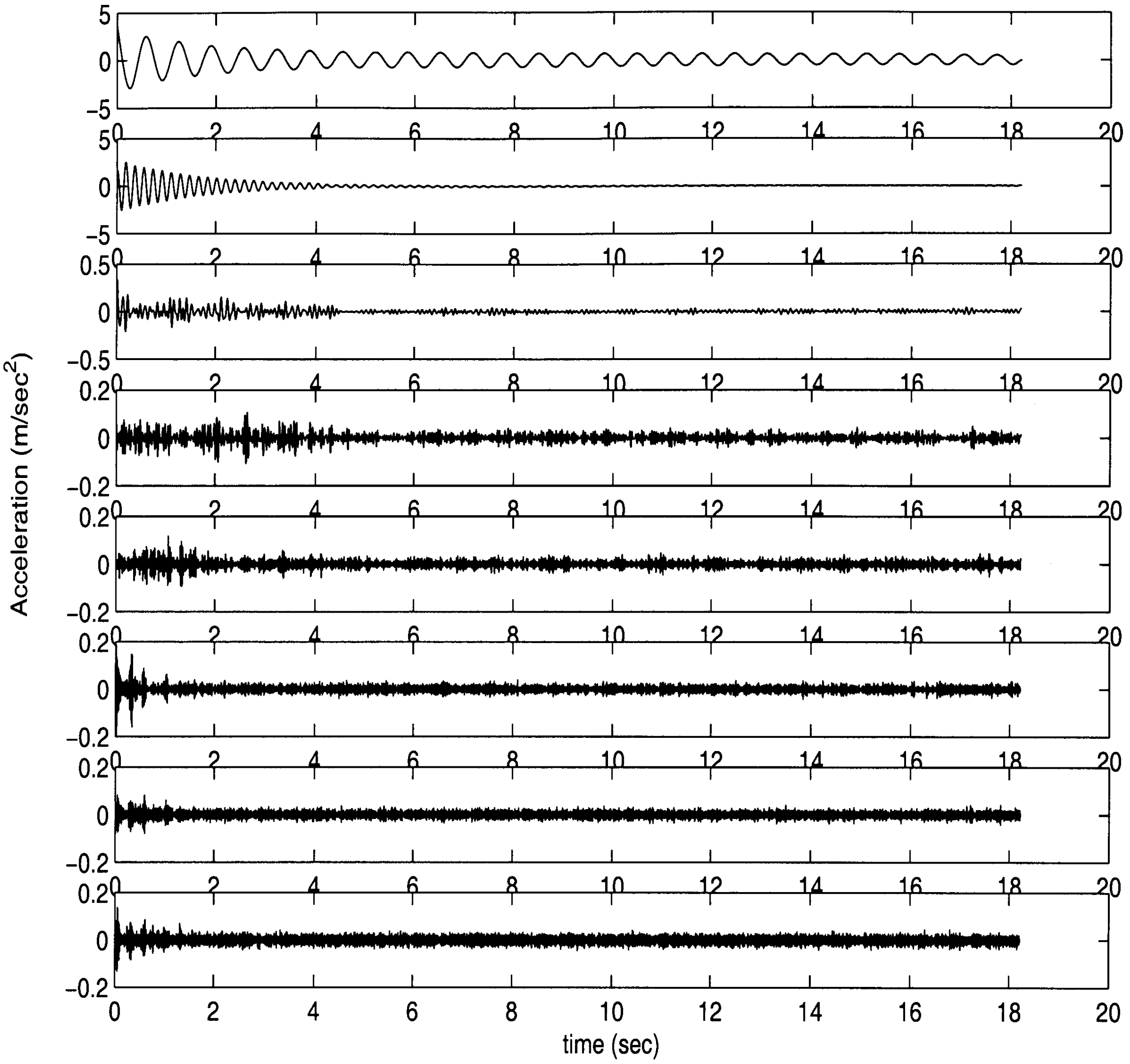

Fig. 5.4: Signal reconstructions from individual coefficients

In the plots above, reconstructed from approximation and detail coefficients, its is clearly seen that from D1 to D5, the signal content is mainly high frequency noise, originating from a number of sources, like accelerometers themselves, the data acquisition system, dangling wires, etc. the D6 component shows high some high frequency part which is decaying out with time. This corresponds to the high frequency waves traveling in the columns of the model and are independent of the modal response from the discretized system so they are overlooked in this case. We look at the signal reconstructed from approximation first, in figures 5.5 and 5.6.

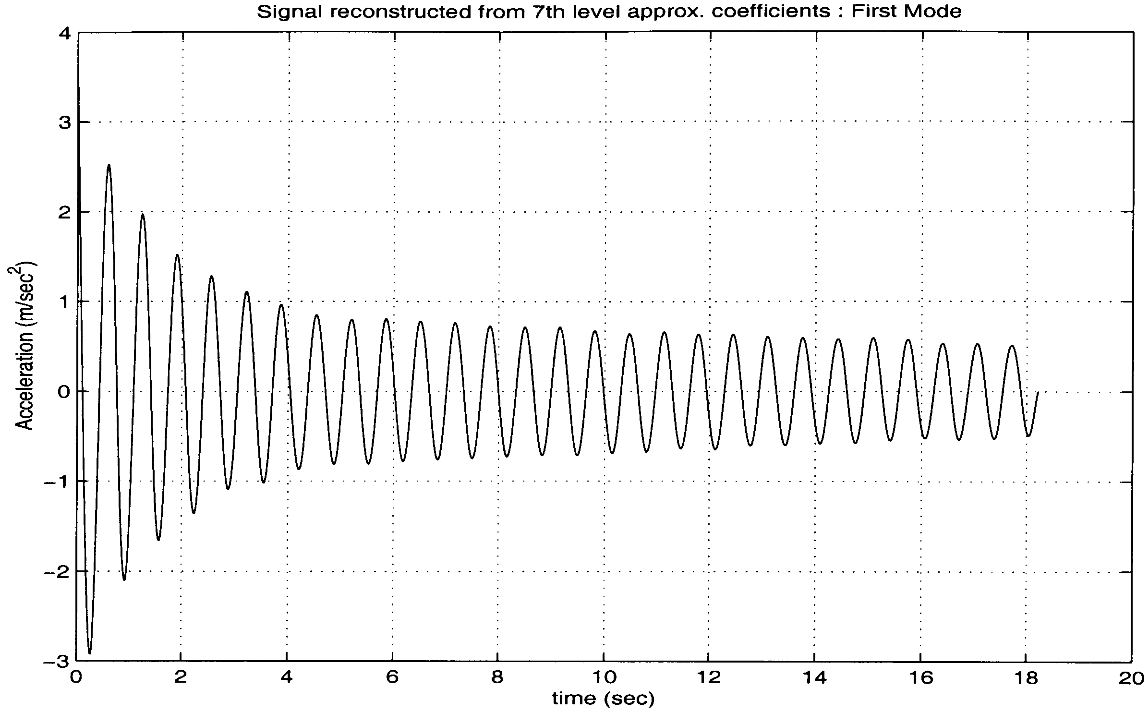

Fig 5.5: Signal reconstructed from 7th level approximation coefficients. This represents the first mode quite accurately.

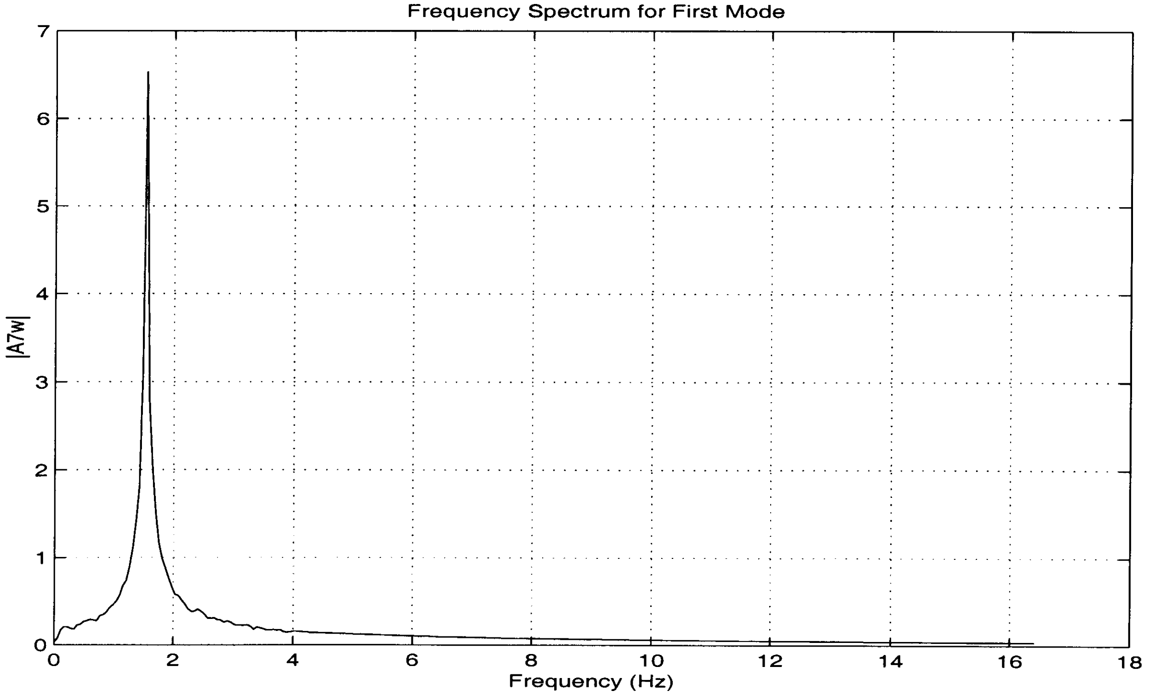

Fig 5.6: The frequency spectrum of the signal reconstructed from 7th level approx. coefficients. The peak frequency corresponds to the first modal frequency.

The signal has a frequency of 1.53 Hz and turns out to be the first mode, of the structural response. We verify this in the frequency domain by carrying out a Fourier tranform of this reconstructed signal. We can see that the first modal frequency is cleanly extracted from all the other frequencies.

Similarly the second mode is entirely contained in the signal reconstructed from the seventh level detail coefficients. This is shown in fig. 5.7 . The frequency spectrum of this component is shown in fig. 5.8, clearly exhibiting only the second mode frequency. Hence we have the two required modes from only the 7th level wavelet coefficients.

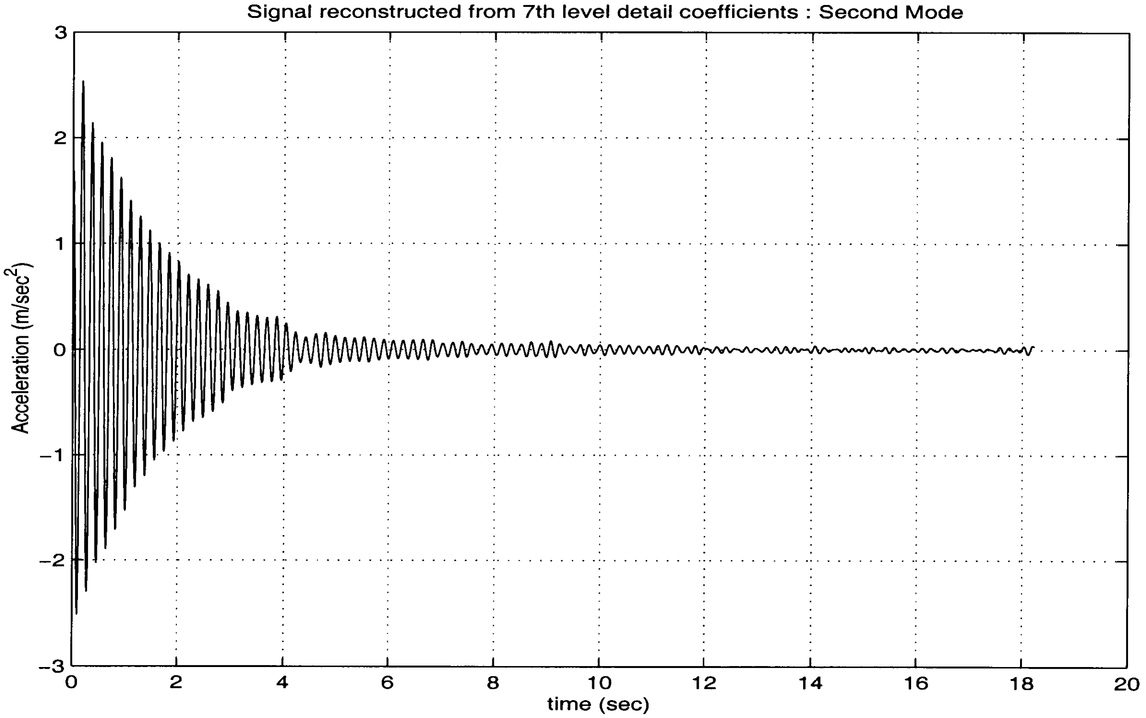

Fig 5.7: Signal reconstructed from 7th level detail coefficients. This represents the first mode quite accurately.

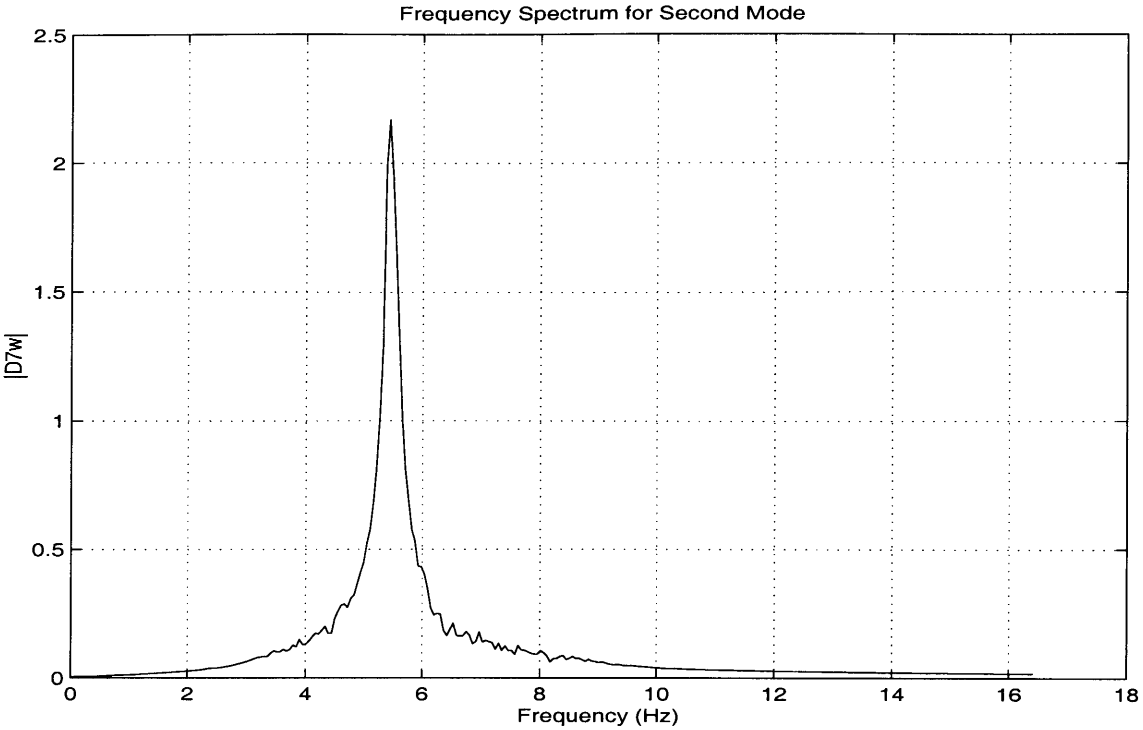

Fig 5.8: The frequency spectrum of the signal reconstructed from 7th level detail coefficients. The peak frequency corresponds to the first modal frequency.

5.4.2 RECONSTRUCTED SIGNAL

The plot shown in figure 5.9 shows the the actual signal along with the signal reconstructed solely from 7th level approximation and 7th level detail coefficients only. This is done by setting all the other coefficients equal to zero and reconstructing the signal. We can see that the seventh level detail and approximation components do reproduce quite faithfully the actual signal, but without the high frequency components, which were mainly noise. Hence these signals can be relied upon to actually represent the first and second mode of the structure.

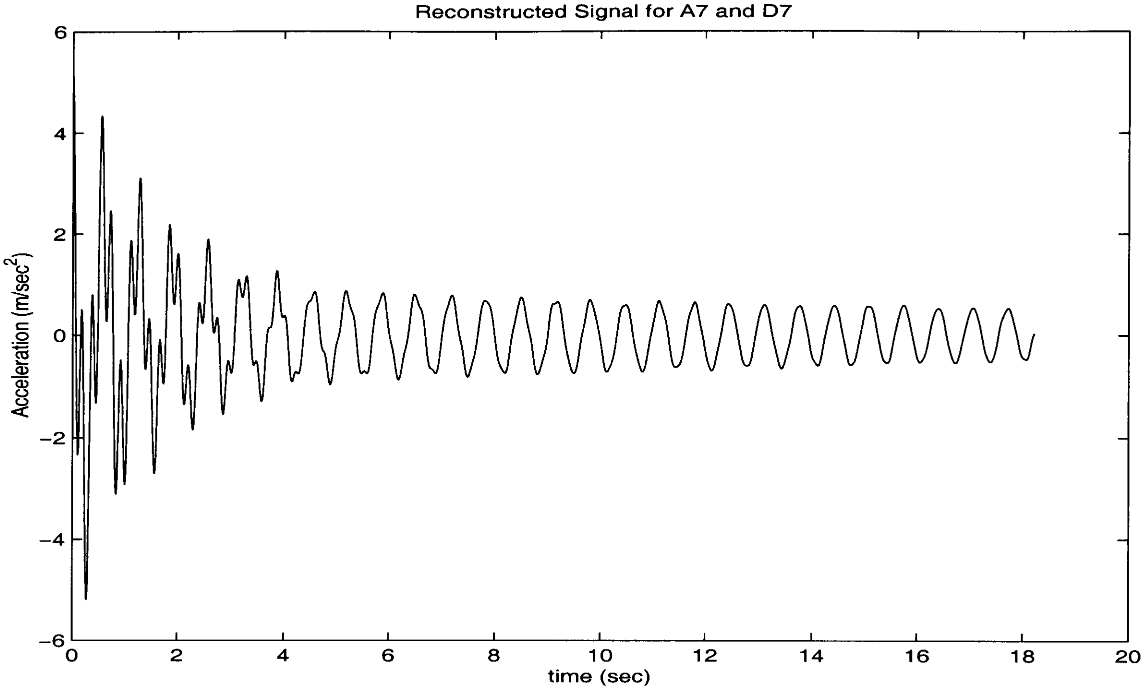

Fig 5.9: Reconstructed signal with only the 7th level approximation and detail coefficients

5.4.3 THE ENVELOPE FUNCTION

We calculate the envelope function using the methodology explained earlier. The following plots show the envelopes extracted for both the modes and the damping calculated from them. Figure 5.10 shows the envelope curve determined for the first mode, using the Hilbert transform function in `Matlab`.

The plot in fig. 5.11 shows how the damping changes with amplitude (which in turn decreases with time). This is clearly supported by the plot in fig. 5.10, which shows that that the rate of decay has decreased (over all) as the amplitude is decreasing. This damping curve is obtained by taking the slope of the envelope function at each instant.

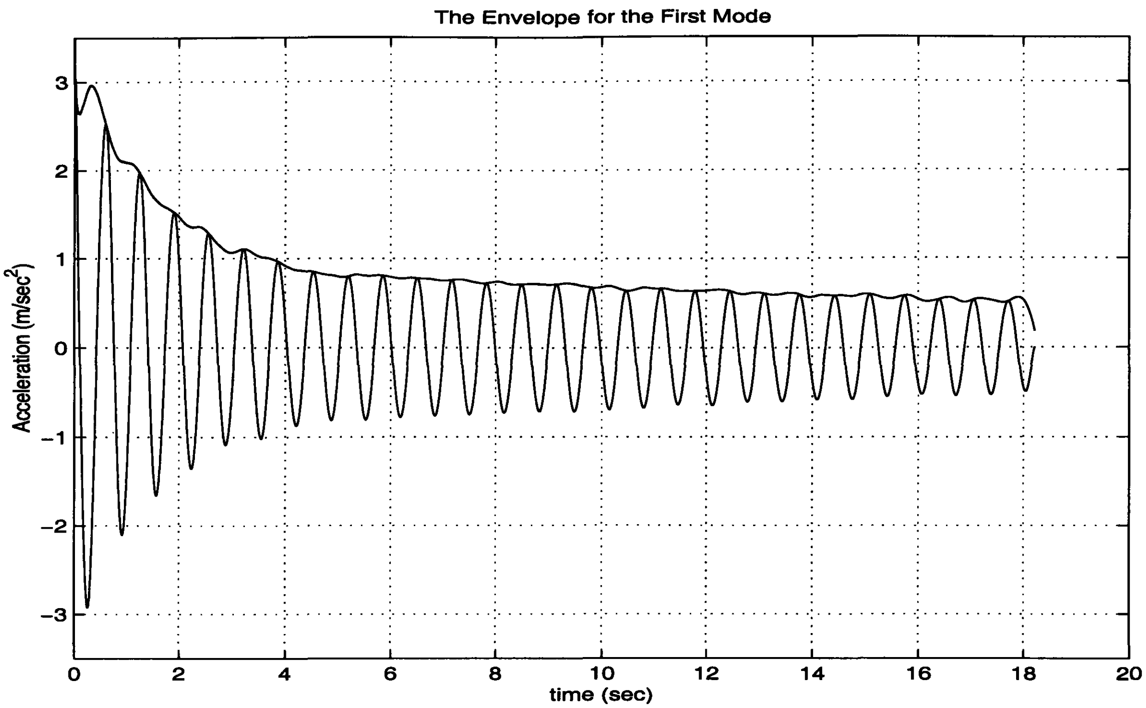

Fig 5.10: The envelope for first mode

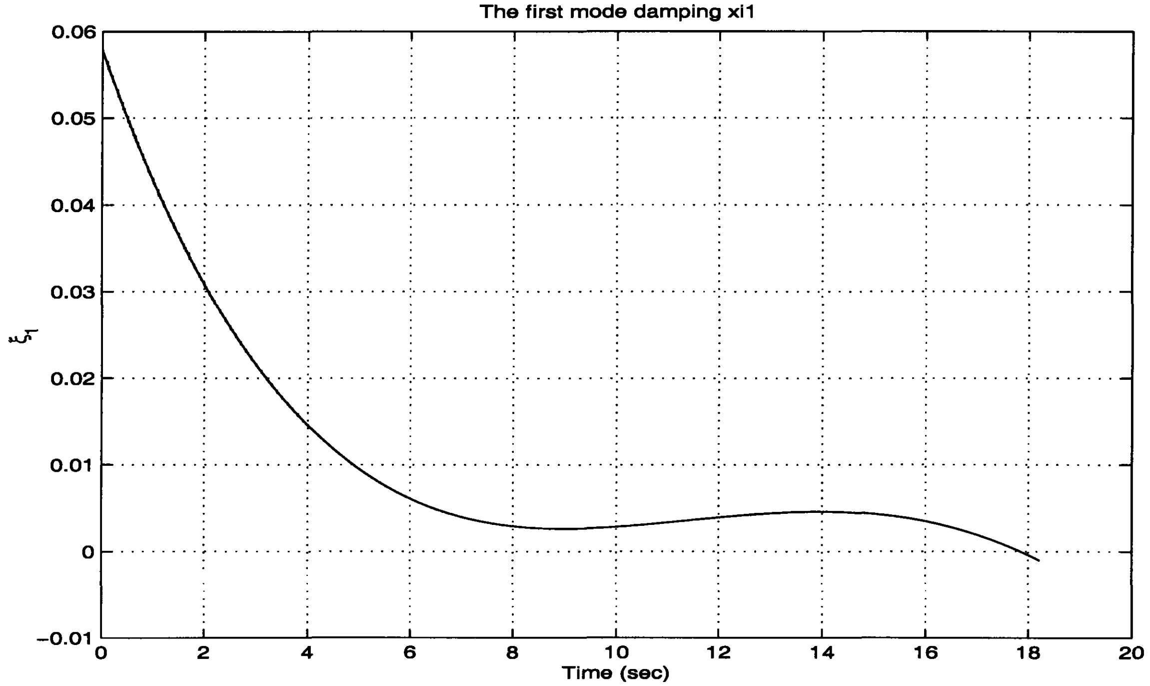

fig. 5.11: The variation in first modal damping ratio

Since the envelope function is not exactly smooth, (because of discretization), taking its log does not produce a smooth function. That has to be approximated by an equivalent polynomial fit. This is done using MATLAB basic fitting utility.

Similarly, for second mode, the figures on next page show the envelope function and the damping extracted.

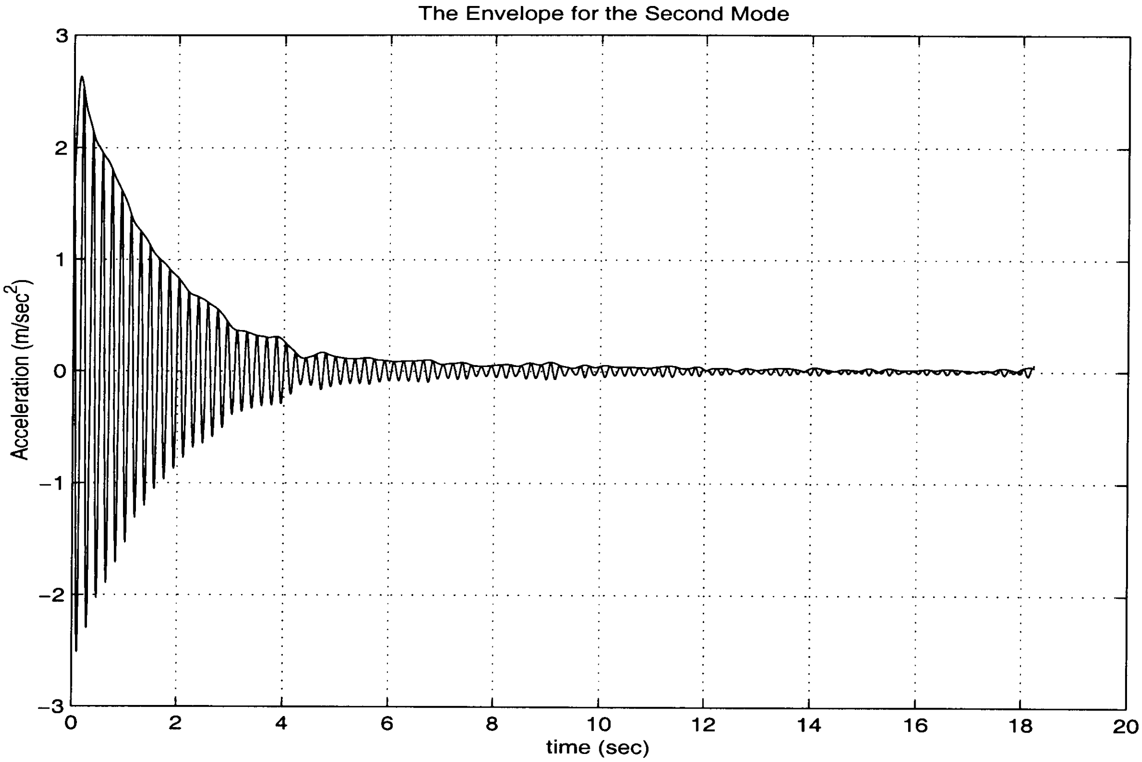

Fig. 5.12: The envelope function for second mode

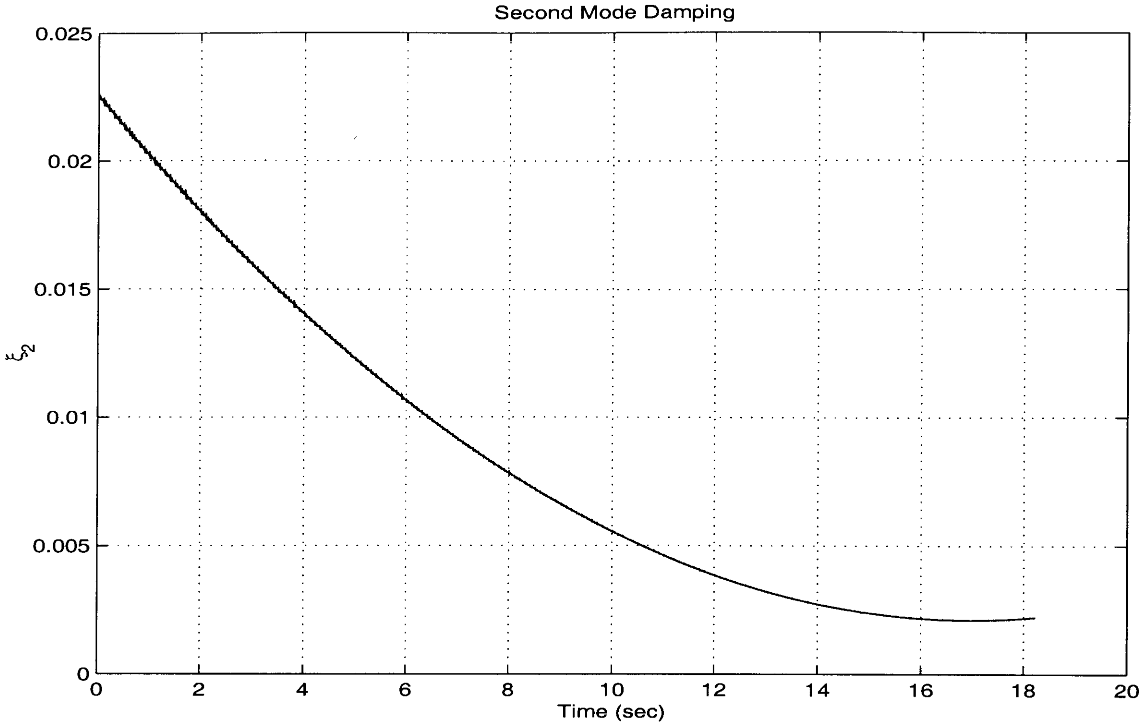

Fig. 5.13: The variation in second damping ration

5.5 CONCLUSION

The damping ratios decrease, as can be seen, as the amplitude of vibration dies out. This can interpreted in the light of the fact that structural damping, although modeled as viscous, is actually mainly due to internal material friction, which depends on the amplitude. Hence it changes with decay.

Although the results are quite encouraging, and the damping ratios obtained are very realistic, there is some refinement required at this stage, mainly in the area of satisfactory curve fitting of the functions obtained by taking the log of the envelopes obtained.

CHAPTER 6

CONCLUSION

6.1 ADVANTAGES OF USING INSTRUCTIONAL SHAKE TABLE

The use of Instructional Shake Table system for demonstration of concepts in structural dynamics and earthquake engineering is a very convenient and effective way of bringing to life many theoretical aspects taught in these disciplines. The usual curricula that are followed in most colleges traditionally focus heavily on theoretical aspects, going into the mathematical details. While these mathematical and theoretical aspects are extremely important and cannot be dispensed with, the fact remains that without actual hands on experience, they mostly tend to remain mere mathematical abstractions. This is especially true in the case of those undergraduate students who have had no prior exposure to structural dynamics. Without actually experiencing the physical meaning of these concepts and seeing how the underlying mathematics relates to the physics of the problem, it is hard for the student to have a good knowledge grasp.

The Instructional Shake Table can be used to devise various experiments that bring forward and highlight the actual "physics" of the problem, helping students to catch up on the theory that is taught in the classroom.

Shake tables have also been used at various universities of the consortium, notabl=ye University of Wisconsin at St. Louis, for bringing children from k-12 in contact with basics of structural dynamics, explaining the behavior of buildings due to seismic loads. This helps in not only college, but pre-college educations, sparking interest in structural analysis.

PUBLIC AWARENESS TO EARTHQUAKE HAZARDS

The Shake Table can be a very effective tool in increasing public awareness regarding earthquake hazards. Earthquake models can bring to light, the ways building responds and how changing the structural properties of a building, can change the response to an earthquake, rendering the building safer or more hazardous. Any major earthquake usually sparks up a lot of public interest and there is a general desire to know more about the hazards and effects of earthquake on structures, especially buildings. Shake tables can be used to great advantage for such public education, thus potentially promoting construction methods and materials that are suitable to seismically active regions, resulting in better earthquake resistant structures. Such outreach activities can be a great source of public education regarding earthquake hazard mitigation.

6.2 PROBLEMS ENCOUNTERED

The whole system has been custom built for the University Consortium of Instructional Shake Tables, by Quanser Consulting Inc. The system prototype was conceived at University of Wisconsin at St. Louis, and Quanser Consulting then proceeded to make the operational units with some modifications for better. Nevertheless it was a unique project and carried out for the first time at this level. This naturally resulted in some bugs that were overlooked before the apparatus were commissioned.

6.2.1 DOCUMENTATION ERRORS

The biggest problem that was faced was that of proper documentation. The system does come with some manuals, but these gave scattered information, certainly not in a coherent manner. Also the manner in which the manuals are written, especially the one for the Multi-Q data acquisition system, is more oriented towards an individual already expert in electronics and data processing through hardware. The targeted end used of this system is supposed to be a civil engineer, structural engineer most probably. Much time is lost in deciphering the electrical engineering jargon used in the manual for Multi-Q. There is also some inaccuracy in which the connectivity of the Active Mass Driver is explained in the manual : some extra cables are instructed to be connected between the Universal Power Module and the terminal board. The connectivity was setup true to the instructions given. It was observed that a lot of noise began to corrupt the accelerometer readings. This became most evident when it was observed that the accelerometer for the table, was given high magnitude readings even when the table itself was not moving. The magnitude was such that it could not be dismissed as mere random, electrical noise. Also the reading of the first floor accelerometer appeared to be half wave rectified. After careful and logical reasoning the culprit cables were removed, and after than the readings became cleaner, and made sense.

6.2.2 SOFTWARE PROBLEMS

As stated in earlier chapters, the system comes with WinCon 3.1 control software. This software is also from Qunaser Consulting, and makes the job of controlling the table quite simple. But it has some problems, although not very serious, which need to be looked into. The following observations are made about WinCon:

- Although working smoothly, the software's server part gets stuck at times, for no apparent reason. This happens sometimes when an attempt is made to load .wcp project files. The program has to be closed using the Windows OS Task Manager. Occasionally even that fails to work and the computer has to be restarted. This is a problem which should be resolved, since the system can crash at most crucial times, spoiling the whole demonstration sequence.
- In the manuals, Quanser Consulting Inc. talk about an "External Interface Window", which could be used to write C++ or Java applications to directly talk to WinCon. There is given a list of some parameters that are exchanged between such an "Outside Application" and WinCon, but no examples are given. Quanser has, however, back tracked and no longer support this feature. This makes the control software WinCon, virtually a black box, making it extremely difficult to be able to communicate with it via a user written program.

6.2.3 HARDWARE PROBLEMS

The apparatus hardware is robustly built and has hardly ever given any problem as yet. The only thing that might be improved upon is that when the Universal Power Module is switched off, its capacitors do not immediately change state. Indeed if it is turned on soon after switching it off, it does not work. This problem can be easily over come by only taking care that there is some suitable interval between switching the power module on again.

6.3 CONCLUSION

The use of Instruction Shake Table for demonstrations in structural dynamics proved to be a good experience overall. After testing of equipment and making some modifications in the standard earthquake simulations, an experiment was developed which incorporated the use of Active Mass Driver to control the vibrations of the model structure. This experiment proved to be quite successful. It has started to be used effectively used at MIT to demonstrate the idea of active structural control, by Prof. J. J. Connor at the

Department of Civil and Environmental Engineering at MIT, to students of one his graduate classes. The course involved advanced structural analysis and control of structures. The use of shake table put forward the ideas clearly and students are actually able to see the implementation of the theory they study in class.

Apart from this, the shake table can be used for various other courses involving different fields, like geotechnical engineering. The table can be effectively used to demonstrate the liquefaction of sand and soft soils under dynamic loads. Smaller model structures can built and soil structure interaction under seismic conditions demonstrated.

APPENDEX A

MATLAB CODE FOR BASE LINE CORRECTION

The algorithm that was developed in chapter 4 is implemented in MATLAB, as a function named `bsline`. The function takes in two arguments, namely the time array and the measured acceleration array, and gives out one array containing the corrected acceleration values. This array is of the same size as the input array of accelerations.

Given below is the MATLAB code for the function

```
function [acc] = bsline(time,DATA)
%%  BASE LINE CORRECTION TO ACCELERATION RECORDS
%%  METHOD OF MINIMUM OF INTEGRAL OF SQUARED VELOCITIES : without assuming that %%  the
velocity is zero at the end of the accleration record
%%  APPLICABLE FOR ACCELERATION RECORDS FROM ACCELEROMETERS MOUNTED ON VIBRATING %%
STRUCTURES
%%  This script applies base line correction to accelertion record saved from wincon

%%  Written by:
%%                  Muhammed Ali Irfan Baig
%%                  Graduate Student
%%                  Department of Civil and Environmental Engineering
%%                  Massachusetts Institute of Technology


len = size(DATA);
%%   Declaring an array which will hold the       velocity, found by numerical
%%   integration of the record
v = [1:len(1)];
```

```
v = v';

%%   Assuming structure was at rest when the excitation started
v(1) = 0;

plot(time,DATA);
pause;
%DATA=DATA.*9.81;

%%   Performing numerical integration on uncorrected data now

for i = 2:1:len(1);
    v(i) = v(i-1) + (0.5*(DATA(i-1)+DATA(i))*(time(i)-time(i-1)));
end


subplot(2,2,1) : plot(time,DATA);
title('UNCORRECTED ACCELERATIONS')
subplot(2,2,3) : plot(time, v);
title('UNCORRECTED VELOCITIES');
%pause;

%%   Storing vT, T

vT = v(len(1));
T  = time(len(1));

%%  Calculating J1 and J2
J1 = 0;                %% since the initial velocities are considered to be zero.
J2 = 0;
J3 = 0;
for i = 2:1:len(1)
    a  = v(i-1)*(time(i-1))^3;
    b  = v(i)*(time(i))^3;
    J1 = J1 + 0.5*(a + b)*(time(i)-time(i-1));

    c  = v(i-1)*(time(i-1))^2;
    d  = v(i)*(time(i))^2;
    J2 = J2 + 0.5*(c + d)*(time(i)-time(i-1));

    e  = v(i-1)*(time(i-1));
    f  = v(i)*(time(i));
    J3 = J3 + 0.5*(e + f)*(time(i)-time(i-1));
end

J1 = -J1 * 2 /(3*(T^5));
J2 = -J2 / (T^4);
J3 = -J3 * 2 / (T^3);

%%  Calculating coefficients A, B, C, for baseline correction function

A   =   (7087.5 * J1 -  6300 * J2 + 945 * J3)/(T^2);
B   =   (-6300  * J1 +  5760 * J2 - 900 * J3)/T;
C   =      945  * J1 -   900 * J2 + 150 * J3;
```

```
%% Performing baseline correction

for i = 1:1:len(1)
    acc(i) = DATA(i) + (A*time(i) + B)*time(i) + C;
end

subplot(2,2,2) : plot(time, acc);
title('CORRECTED ACCELEROGRAM');
%%pause
```

APPENDEX B

MATLAB CODE FOR ESTIMATION OF DAMPING

In chapter 5, it was shown that modal damping ratios from the accelerometer records could be estimated using discrete wavelet decomposition. The relavant modal components can be extracted using this technique, and then envelope functions estimated for each component using the Hilbert Tranform. This MATLAB code is an implementation of the algorithm discussed in chapter 5. The data processed is from the top floor accelerometer of the model structure for a free vibration test.

```
%%   Determination of Modal Damping from a signal using
%%   Wavelet Transform
%%
%%   Written by :
%%                  Muhammed Ali Irfan Baig
%%                  Graduate Student
%%                  Department of Civil and Environmental Engineering
%%                  Massachusetts Institute of Technology


%%   We use the following wavelet for signal decomposition decomposition

wlt = 'dmey';       %%%%%%%   Discrete Meyer Wavelet

%%   Loading the signal from .mat file
```

```
load data3;                    %   Loads acceleration data from top floor
accelerometer, for free vibration
time = time';
len =   length(time);          %   Determining the length of the signal

dt  =   0.001;               %   dt = time step

%%   Now we perform the discrete fourier transform to find the dominant modal frequencies

nyq =   1/2/dt;              %   Nyquist Frequency
T   =   time(len);
df  =   1/T;
freq=0:df:nyq;

Aw  =   abs(fft(A*dt));            %   Performing FFT on the signal
plot(time,A); xlabel('Time (sec)'); ylabel('Acceleration (m/sec\sq)');
title('The original signal (Base line corrected)');
pause;
plot(freq(1:150),Aw(1:150)); xlabel('Frequency (Hz)'); ylabel('|Aw|');
title('Frequecy Spectrum of the Input Signal')
pause;

%%%
%%%     From the frequency spectrum we see that the dominant frequencies of the 2-dof
%%%     system are 1.537 Hz and 5.434 Hz.  The sampling period of the original signal is
%%%     0.001 sec, corresponding to a nyquist freq of 500 Hz.  Therefore we do a seven
%%%     level wavelet decomposition, assuming initially ideal low and high pass filtering
%%%

w1 = 1.537 *2*pi;
w2 = 5.434 *2*pi;

%       Performing seven level discrete wavelet transform

[C L]=  wavedec(A,7,wlt);

%       Rebuiling signal from individual approximation and detail coefficients
%       and ploting them along with their frequency spectra

A7  =   wrcoef('a',C,L,wlt,7);
A7w =   abs(fft(A7*dt));

D7  =   wrcoef('d',C,L,wlt,7);
D7w =   abs(fft(D7*dt));

D6  =   wrcoef('d',C,L,wlt,6);
D6w =   abs(fft(D6*dt));

D5  =   wrcoef('d',C,L,wlt,5);
D5w =   abs(fft(D5*dt));

D4  =   wrcoef('d',C,L,wlt,4);
D4w =   abs(fft(D4*dt));
```

```
D3  =   wrcoef('d',C,L,wlt,3);
D3w =   abs(fft(D3*dt));

D2  =   wrcoef('d',C,L,wlt,2);
D2w =   abs(fft(D2*dt));

D1  =   wrcoef('d',C,L,wlt,1);
D1w =   abs(fft(D1*dt));

subplot(8,2,1):plot(time,A7);
subplot(8,2,2):plot(freq(1:300),A7w(1:300));

subplot(8,2,3):plot(time,D7);
subplot(8,2,4):plot(freq(1:300),D7w(1:300));

subplot(8,2,5):plot(time,D6);
subplot(8,2,6):plot(freq(1:300),D6w(1:300));

subplot(8,2,7):plot(time,D5);
subplot(8,2,8):plot(freq(1:300),D5w(1:300));

subplot(8,2,9):plot(time,D4);
subplot(8,2,10):plot(freq(1:300),D4w(1:300));

subplot(8,2,11):plot(time,D3);
subplot(8,2,12):plot(freq(1:300),D3w(1:300));

subplot(8,2,13):plot(time,D2);
subplot(8,2,14):plot(freq(1:300),D2w(1:300));

subplot(8,2,15):plot(time,D1);
subplot(8,2,16):plot(freq(1:300),D1w(1:300));

pause;
close;

%       Rebuilding signal from individual approximation and detail coefficients
plot(time,A7); xlabel('time (sec)'); ylabel('Acceleration (m/sec^2)');
title('Signal reconstructed from 7th level approx. coefficients : First Mode')
pause;
plot(freq(1:300),A7w(1:300)); xlabel('Frequency (Hz)'); ylabel('|A7w|');
title('Frequency Spectrum for First Mode')
pause;

plot(time,D7); xlabel('time (sec)'); ylabel('Acceleration (m/sec^2)');
title('Signal reconstructed from 7th level detail coefficients : Second Mode')
pause;
plot(freq(1:300),D7w(1:300)); xlabel('Frequency (Hz)'); ylabel('|D7w|');
title('Frequency Spectrum for Second Mode')
pause;

%%%     Reconstruct the signal using A7 and D7

rA  =   A7 + D7;
```

```
plot(time,rA,'-r'); xlabel('time (sec)'); ylabel('Acceleration (m/sec^2)');
title('Reconstructed Signal for A7 and D7');
pause
hold
plot(time,A);
pause;
hold

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%    FINDING THE DAMPING OF FIRST MODE    %%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

env1 = abs(Hilbert([A7 zeros(1,10000)]));
env1 = env1(1:len);
plot(time,A7); xlabel('time (sec)'); ylabel('Acceleration (m/sec^2)');
hold
plot(time,env1,'-g')
title('the envelope for the first mode');
hold
pause;

logenv = log(env1);

p1 = -7.0275e-006
  p2 = 0.00041059
  p3 = -0.0095383
  p4 = 0.10944
  p5 = -0.65098
  p6 = -1.4985

logenv = p1*time.^5 + p2*time.^4 + p3*time.^3 + p4*time.^2 + p5*time + p6;

%%  Finding the slope of the log curve to find modal damping ratio %%   %%%%%%%%%

%%%%%% logenv = log(env1fit/env1fit(1));

damping(1) = (logenv(2)-logenv(1))/dt;                                  %%%%%
for i=2:1:len-1
   damping(i)=(logenv(i+1)-logenv(i-1))/2/dt;
end
damping(len) = (logenv(len)-logenv(len-1))/dt;

for i=1:1:len
     damping(i) = -damping(i)/w1;
end

plot(time,damping); xlabel('Time (sec)'); ylabel('xi1');
title('The first mode damping xi1')
pause;
%%  close;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%    FINDING THE DAMPING OF SECOND MODE    %%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

env2 = abs(Hilbert([D7 zeros(1,10000)]));
env2 = env2(1:len);
plot(time,D7);xlabel('time (sec)'); ylabel('Acceleration (m/sec^2)');
hold;
plot(time,env2,'-g')
title('the envelope for the second mode');
hold;
pause;


logenv2 = log (env2);

p1 = -0.0014423;
  p2 = 0.051502;
  p3 = -0.78801;
  p4 = -0.1081;


logenv2 = p1*time.^3 + p2*time.^2 + p3*time + p4;

%%  Finding the slope of the log curve to find modal damping ratio %%

%%%%%logenv2 = log (env2fit/env2fit(1));
damping2(1) = (logenv2(2)-logenv2(1))/dt;
for i=2:1:len-1
    damping2(i)=(logenv2(i+1)-logenv2(i-1))/(2*dt);
end
damping2(len) = (logenv2(len)-logenv2(len-1))/dt;

for i=1:1:len
    damping2(i) = -damping2(i)/w2;
end

plot(time,damping2); xlabel('Time (sec)'); ylabel('xi1');
title('Second Mode Damping')
```

REFERENCES

[1] *WinCon 3.1 User's Manual* ,Quanser Consulting.

[2] *UCIST Shake Table,* Quanser Consulting.

[3] *MULTIQ-3 Programming Manual,* Quanser Consulting.

[4] *www.quanser.com*

[5] *MATLAB, User's Guide.* Mathworks Inc.

[6] *SIMULINK, User's Guide.* Mathworks Inc.

[7] *Advanced Structural Dynamics,* Eduardo Kausel and Jose Manuel Roesset

[8] *Structural Dynamics: Theory and Applications,* J.W. Tedesco, W.G. McDougal and C.A. Ross. Addison Wesley Longman, Inc., CA, 1999.

[9] *Dynamics of Structures,* R.W. Clough and J. Penzien. 2nd edn, McGraw-Hill Book Co., NY, 1993.

[10] *Dynamics of Structures:Theory and Applications to Earthquake Engineering,* A. K. Chopra. Prentice Hall, 1995.

[11] *Advancing Earthquake Engineering Education Through a Cooperative Effort Based on Instructional Shake Tabl,.* Shirley J. Dyke, Scott M. Johnson, R.T Ranf, Juan M. Caicedo and M. Soto-Fournier. 7th US National Conference on Earthquake Engineering, Boston, MA.

[12] *Creating SIMULINK Models for use with the Instructional Shake Table,* Scott Johnson. University Consortium of Instructional Shake Tables

[13] *Wavelets and Filterbanks,* Gilbert Strang and Truong Guyen. Cambridge Wellesley Press.

[14] *Discrete Digital Signal Processing,* Alan V. Oppenheim, Ronald W. Schafer, John R. Buck, Prentice Hall, 1998.