
Engineering Tripos Part IIA Module Experiment: 3C6

Digital Vibration Analysis, South Wing Mechanics Laboratory

1. Objectives

- To measure the calibrated transfer function of a vibrating structure using several common experimental methods;
- To predict the response of a coupled structure from the measured transfer functions of its decoupled subsystems;
- To compare the advantages and disadvantages of the different experimental methods.

2. Preparation

DO THIS FIRST: Turn on the PC and log in using your usual Teaching System ID and password. It can take a while for the windows profile to load.

If the power is not on at the benches, turn on the switches on the pillar at the north end of the bench, labelled "computers" and "bench sockets". You may need to plug in the two BNC cables carrying your input and output signals to the diecast box at the rear of the computer trolley. Your cables are labelled "3C6 CH1" and "3C6 CH2", to be plugged into the corresponding channels.

When using the magnets make sure they are well seated. If they are not, you will hear buzzing or rattling. Remove the magnet and wipe both surfaces with your finger to remove any grit. Moving the magnet slightly is often all that is needed to get rid of the buzz. If the buzz persists, try using a lower vibration level.

3. Experimental procedure

3.1 Sine wave measurements (30 minutes)

If the computer is not already on, switch it on, then log in using your usual Teaching System ID and password.

For the first part of the experiment you will measure a transfer function using the stepped-sine method:

1. Click '3C6_Notebook.ipynb' to open the notebook.
2. Attach the shaker to the bottom beam using the magnetic connector (align the rod to the shaker carefully before tightening the connection screws) and insert the brass coupling rod between the two beams if it is not already there.
3. Attach the accelerometer to the top beam directly above the coupling rod. Make sure the cables are clear of the beams.
4. Set the selector switch to "Sine".
5. Run the code below.

Remember:

- to actually run a cell of code, click inside the cell then press 'shift+enter'
- the cell is running while [*] is displayed
- the cell has finished running when it changes to a number

- under the 'view' menu above you can toggle off the header and toolbar, which is useful to give a bit more screen space
- you can control the cell output view by clicking on the left beside the cell: single click to expand/compress, double-click to hide.

```
In [ ]: import matplotlib
import numpy as np
import lab_3c6
import pydvm as dvma
```

```
In [ ]: %matplotlib notebook
```

```
In [ ]: # acquisition setup
settings = dvma.MySettings(channels=2,
                           fs=3000,
                           viewed_time=1,
                           device_driver = 'nidaq')
```

Now open a PC oscilloscope using your settings. This shows three plots:

- the top one is like a normal oscilloscope showing the signal (toggle on/off with 'T');
- the middle one shows the frequency spectrum of the signal (toggle on/off with 'F');
- the bottom one shows the signal amplitudes (toggle on/off with 'L');
- you can pause the data shown by pressing 'P' (press again to continue streaming data);
- you can toggle whether the window is always on top by pressing 'A'

Tap the top beam gently with a finger. Check that you can see a response in the oscilloscope.

Close the oscilloscope

```
In [ ]: # open oscilloscope
osc = dvma.Oscilloscope(settings)
```

In this part of the lab we want to measure the gain and phase shift from input to output at the specific frequencies of the generator. To do this we will use a simple interface embedded in the notebook.

1. Start the stepped-sine logging widget by running the code in the next cell;
2. Use the sine-wave generator to explore the system response in the frequency range 10–500 Hz;
3. For each frequency tested, allow a short time for the vibration to settle to a steady state, then click the "Measure" button on the computer;
4. A dot will be plotted on the screen showing the admittance at the measured frequency;
5. Explore up and down in frequency until you have dotted out a reasonable picture of the admittance (you should find at least four resonance peaks). Unwanted data points can be deleted using the "Delete Last Measurement" button. Be particularly careful to catch the peak values, where the system has resonances.

You don't have to wait long between measurements: you can get quite fast at changing frequency, pausing half a second, clicking "measure", and repeat.

Aim to get about 100 data points.

When you have gathered sufficient data points, click on the "Save Data" and "Save Fig" buttons: you will be prompted for a file name. You need to save your data as well as the figure so you can read it back again later.

Any files you save during this experiment should be saved to where this notebook has already been copied to and is the default location for all saved data. Do not put files on the local disk of the PC you are using.

```
In [ ]: # log data
logger1 = lab_3c6.measure_stepped_sine(settings)
```

Make sure you remembered to save both the data and the figure!

CHECK: Have you saved the data (you should have a *.npy file)

CHECK: Have you saved the figure (you should have both a *.pdf and *.png file - the 'Save Fig' button saves both at once)

You will need the data later: to check that you have saved it, you can reload and replot it. Note that the individual points are now plotted as a continuous line. There's no need to save the figure below as well, having saved it above.

```
In [ ]: data_stepped_sine = dvma.load_data()  
data_stepped_sine.plot_tf_data()
```

For the report:

1. Comment on your general observations of the data.
 2. Comment on the advantages and disadvantages of this method.
 3. Comment on the accuracy of this method.
-
-

3.2 Random noise excitation (30 minutes)

Switch to "Noise" on the selector, and you will hear the sound of the cantilever system responding to such a signal, generated by an analogue noise generator.

The data acquisition used for the rest of the lab will use the 'pydvma' logger interface that is more general than the embedded widget above. Open the logger window by running the next cell (shift+enter).

Familiarise yourself with the logger window:

- the centre panel contains the figure - note the zooming / panning tools under the figure.
- when data is shown, you can click on the legend lines to select which channels are shown.
- when using the box-zoom tool (from under the figure), then left-click-drag will zoom to the rectangle, and right-click-drag will zoom out.
- the left panel controls what is plotted (you can select which plot to show using the top dropdown).
- use the 'auto x' and 'auto y' buttons to automatically adjust the x and y axes to fit the selected data ('auto y' changes the y-limits to fit the selected data, 'auto x' changes the x-limits to the whole data range).
- the right panel provides data analysis tools (you can select which tools are displayed using the top dropdown).
- the centre top panel is for basic data management (logging / loading / deleting).
- the centre bottom panel is to save data or figures.
- the max signal levels are shown for convenience in the title bar of the window for quick reference.
- in the top-left panel you can open the oscilloscope by clicking on the small icon button.

```
In [ ]: # acquisition setup  
settings = dvma.MySettings(channels=2,  
                           fs=3000,  
                           stored_time=2,  
                           device_driver = 'nidaq')  
logger2 = dvma.Logger(settings,  
                      test_name = 'noise_input_' + str(settings.stored_time) + 's',  
                      default_window='hann')
```

We will explore the effect of some different logging options, starting with 'stored_time=2' seconds. Open the logger interface:

1. Press **'Log Data'** (**top-middle panel**) to log data using the specified settings
2. You should have a single measurement, shown as two lines in the figure.
3. If you have made more measurements then use **'Delete All'** (**top-middle panel**) and carry out a single measurement.
4. Press **'Save Figure'** (**bottom-middle panel**) to save the time data plot as a *.png and *.pdf file
5. Press **'Calc FFT'** (**right panel**) to calculate the FFT using the window function specified beneath it: the default 'hann' is appropriate here
6. Press **'Save Figure'** (**bottom-middle panel**) to save the plot

7. Press '**Calc TF**' (**right panel**) to calculate the Transfer Function: the defaults window='hann' and average='None' are appropriate here. We will use the averaging option later.
8. Zoom in on the frequency range of interest (0-500 Hz) and adjust the y-scale appropriately using '**Auto Y**' (**left panel**)
9. Press '**Save Figure**' to save the plot. It will save a *.png and *.pdf of the figure as seen on the screen:
 - You can adjust the x-y scales using the box zoom (under the plot) and the axis tools (left panel)
 - You can choose the legend location (left panel)
 - You can toggle which lines are displayed by clicking on the lines inside the legend (when 'off' they will be still visible as transparent lines, and you can choose which line is on 'top' by deselecting and reselecting it)
10. Press '**Save Dataset**' to save all the data (it will save the data that has been calculated so far, so press it after you have calculated the FFT and Transfer Functions)

Next repeat the steps above, but logging for 30 seconds instead. You will need to change the settings line to 'stored_time=30' and re-run the logger.

- **CHECK 1: Did you remember to re-run the test using stored_time=30 seconds?**
- **CHECK 2: Did you remember to save your data after calculating the transfer function?**
- **CHECK 3: Did you remember to save the figures that you want to keep for the report?**

Remember to save your data as well as all figures to the teaching system

Understanding the logger:

When you press 'Save Dataset', the logger saves a <DataSet> object which has the following structure:

```
<DataSet> class:
    time_data_list: [<TimeData>, <TimeData>, <TimeData>]
    freq_data_list: [<FreqData>, <FreqData>, <FreqData>]
    cross_spec_data_list: []
    tf_data_list: [<TfData>, <TfData>, <TfData>]
    modal_data_list: []
    sono_data_list: []
    meta_data_list: []
```

Each type of data is arranged in 'sets', e.g. each measurement will add another <TimeData> set to time_data_list. This example has three sets of each data type. Similarly each time you load data it will add the loaded sets to the appropriate data list.

- When you press 'Calc FFT' then a <FreqData> item is calculated for each <TimeData> set.
- When you press 'Calc TF' then a <TfData> item is calculated for each <TimeData> set.
- When you press 'Calc TF' with average='across sets' then a single <TfData> item is calculated averaging across all <TimeData> items, with an assumption that the time data sets all have the same settings.

We can compare the transfer functions from the two measurements as follows:

1. Keep your 30 second noise test data (don't delete it)
2. Press 'Load Data' (top-middle panel) and open your data from the 2 second noise test
3. Select 'TF Data' from the 'Figure selection' dropdown (left panel) to compare the datasets from the two measurements

Note that pressing 'Calc TF' (right panel) will also work here as the settings used to calculate both transfer functions are the same. *BUT later you will want to compare transfer functions computed using different settings, in which case you will need to load the data separately and use the 'Figure selection' dropdown.*

Remember to save your comparison data and figures to the teaching system

If you accidentally close the logger window, you can re-open it using:

```
In [ ]: logger2.show()
```

To smooth out the transfer functions we can average the data: recall that the transfer function is given by:

$$G(\omega) = \frac{E[YY^*]}{E[XX^*]}$$

where X and Y are the FFTs of the input and output respectively, and $E[\cdot]$ denotes the expectation (or average). If we take the whole time series then no averages are taken. But we can split the time data into sections (or 'frames'), apply a window to remove discontinuities at the edges of each frame, calculate the FFT of each frame, then take the averages of those.

1. Press 'Load Data' and find the 30 second noise data file
2. Under 'Calc TF' select window='hann' and average='within each set'
3. Use the 'N_frames' slider to investigate the effect of averaging with different numbers of frames (you can also use the text box)
4. Save plots for the cases $N=1$, $N=30$ and your choice of N that gives the 'best' transfer function estimate
5. Be careful to watch the coherence near the peaks as N increases.

Note that you can use the matplotlib viewing tools underneath the plot (e.g. box zoom), or for more careful control you can use the axes controls in the left panel. Note that the coherence axis is labelled on the right-y-axis and is controlled by 'co. min' and 'co. max' in the left panel, separately to the transfer function y-axis scale.

If you accidentally close the logger window, you can re-open it using:

```
In [ ]: logger2.show()
```

- **CHECK 1: Did you remember to save your data after calculating the transfer function using your best choice of N_{frames} ? You will need this later for the comparisons.**
- **CHECK 2: Did you remember to save the figures that you want to keep for the report?**

Remember to save your data as well as all figures to the teaching system.

For the report:

1. Comment on the 2s vs 30s data.
 2. Why do you think the transfer function estimates are noisy without averaging?
 3. What do you observe about the transfer function smoothness and the coherence as N increases?
 4. Justify your choice of your 'best' value for N .
-
-

3.3 Impulsive excitation (20 minutes)

1. Remove the magnet and rod which attaches the shaker to the beams.
2. Move the accelerometer to the underneath of the beams to where the shaker was previously connected.
3. Switch the selector to "Hammer".
4. Using a new logger window with new settings (next cell below), click on 'Log Data';
5. Tap gently on the top of the beams vertically above the accelerometer. Tap within 20 seconds of pressing 'Log Data' to avoid a timeout. The logger will automatically save the signal starting 100 samples just before the tap (specified by the 'pretrig_samples' variable).
- 6. Data will be collected by the computer for 6 s: hold the hammer still and don't put it down on the bench until it is finished.**

7. Check your hammer impulse in the time-domain display window.

- Is it clipped (amplitude=1)?
- Is the signal too small (amplitude<0.1)?
- Was there a single, clean impact or multiple impacts?
- Was there any vibration before the impulse? Make sure the beams were not still vibrating from a previous impulse.
- Did you hear any rattling or buzzing sounds? Try a lighter tap.
- Be prepared to practice your tapping technique until you get good results. Don't proceed until you've got a single clean impulse.

8. Zoom in on the time signals so you can see the individual samples of the impulse and before the impulse. This will be needed for some of the questions on this section. Save this figure.

9. Calculate frequency spectra using 'Calc FFT' (check that window is set to 'None'). Save this figure.

10. Calculate a transfer function using 'Calc TF' to compare with those obtained by the earlier methods. You should find that the result is satisfactory, with sharp clear peaks and troughs, without any need for averaging. If it is not, try again. You may have to develop your hammering technique a little to obtain best results. If you find it hard to get a clean transfer function then try moving the magnet, and checking that the brass coupling rod between the beams is well seated. It takes very little time to repeat the impulse response procedure. **You will find the write-up much easier with good data.** Save this figure.

- **CHECK 1: Did you remember to save your data after calculating the transfer function?**
- **CHECK 2: Did you remember to save the figures that you want to keep for the report?**

Remember to save your data (after calculating the FFT and transfer function) and figures to the teaching system

```
In [ ]: # acquisition setup
logger2.close() # close the previous logger and start a new one
settings = dvma.MySettings(channels=2,
                           fs=3000,
                           stored_time=6,
                           pretrig_samples=100,
                           device_driver = 'nidaq')
logger3 = dvma.Logger(settings,
                      test_name = 'hammer_input_' + str(settings.stored_time) + 's',
                      default_window=None)
```

- **CHECK 1: Did you remember to save your data after calculating the transfer function?**
- **CHECK 2: Did you remember to save the figures that you want to keep for the report?**

Remember to save your data (after calculating the FFT and transfer function) and figures to the teaching system

Now load the data from each of your previous measurements, and plot the transfer functions.

- **DO:** use 'Delete All' to clear existing data
- **DO:** use 'Load Data' to load your chosen datasets
- **DO:** load your hammer data first: for reasons that become clear later
- **DO:** load your best 'averaged' smooth data from the noise-input section
- **DO:** use the 'Plot selection' dropdown to choose 'TF Data' for comparisons
- **DON'T:** press 'Calc TF' otherwise the transfer functions will all be recalculated using the same settings. You don't want this: you want to compare your transfer functions that have each been calculated using the right choices of windows and averaging.

If you accidentally close the logger window, you can re-open it using:

In []: `logger3.show()`

The most obvious difference should be that the hammer results are shifted vertically relative to the noise results. This is because the two input signals need different calibration factors. The computer can calculate an appropriate scale factor to allow direct comparison:

Zoom in to the frequency range 0–500 Hz then press 'Best Match (to ref)'. The scale factor is printed above the plot, and the factor is applied to the data. All datasets are scaled to match the dataset that was loaded first, which is why you should load this first. If you didn't then you can use the 'Ref set / chan' options and specify which 'set' corresponds to the reference hammer data (look at the legend labels to see the set numbers).

Remember to save your data and figures (showing the three measured transfer functions together) to the teaching system

For the report:

1. Comment on the zoomed-in data of the impulse:
 - How long is the impulse? What does that mean for the spectra of the input and output signals?
 - Why are the accelerometer and hammer signals slightly non-zero before the impulse? There may be different reasons for each.
 - Using 'Calc FFT' look at the spectra of the input and output signals - how does this compare with what you would expect from an ideal impulse? Why does it tail off at both low and high frequencies? And why should no window function be used for this test?
 2. Compare the results from the three measurement methods (stepped sine, noise, and impulse hammer), using the scaled data so they are all aligned.
 3. Comment on similarities and differences:
 - Why are the natural frequencies not identical for the different methods?
 - Can you explain why there are more significant differences at low frequencies? Note that the hammer method shows peaks that are not visible in the shaker data. Can you explain why? (Hint: the shaker is applied between the base the cantilevers, while the hammer is not).
 - What else is similar / different and can you suggest reasons why?
 - Consider suggesting how you could test your suggested reasons using further experiments (no need to carry out these tests).
-
-

3.4 Calibration (20 minutes)

The final stage of the measurement is to calibrate your transfer functions to give results in physical units.

1. Move the accelerometer to the suspended mass behind the cantilevers.
2. Start a new logger with the settings below.
3. Use the hammer to measure a good transfer function of this mass, swinging in a straight line.
4. Log several measurements.
5. Convert the transfer function to show acceleration/force instead of velocity/force. To do this:
 - After logging the data, press 'Calc TF' with window=None and average='across sets'
 - Press 'x (iw)', i.e. multiplication by $i\omega$
 - The transfer function should now show a near-horizontal line over a range of frequencies.
6. Make your best estimate of the value of this constant level: to convert the decibel value back to a linear value note that:

$$Y_{\text{dB}} = 20 \log_{10} |Y_{\text{linear}}|$$

7. The transfer function for a free mass is very simple: just rearrange $F = ma$, and the mass (weight plus accelerometer) is given on a label on the apparatus: approximately 0.3 kg.
8. Hence deduce the factor by which the measured transfer function must be multiplied in order to give a calibrated result.
- The calibration factor is defined here to be the factor by which the measured data should be multiplied in order to give the actual values.
 - The calibration factor is not the value of the horizontal line: you need to compare this value with what you expect for the transfer function of a free mass.

```
In [ ]: # acquisition setup
logger3.close() # close the previous logger and start a new one
settings = dvma.MySettings(channels=2,
                           fs=3000,
                           stored_time=6,
                           pretrig_samples=100,
                           device_driver = 'nidaq')
logger4 = dvma.Logger(settings,
                      test_name = 'calibration_' + str(settings.stored_time) + 's',
                      default_window=None)
```

Remember to save your data and figures to the teaching system

Verify that your procedure is correct by loading your data, applying the factor you deduced and plotting the results:

```
In [ ]: calibration_factor = ### enter or calculate your value here
print(calibration_factor)
```

```
In [ ]: # load data
data_calibration = dvma.load_data()
```

```
In [ ]: data_calibration.tf_data_list.set_calibration_factor(factor=calibration_factor)
p = data_calibration.plot_tf_data()
```

```
In [ ]: dvma.save_fig(p)
```

Now apply your calibration factor to your measured transfer function from the impulsive excitation test and resave your calibrated data using a different filename:

```
In [ ]: # load data
data_hammer = dvma.load_data()
```

```
In [ ]: # Apply factor
data_hammer.tf_data_list.set_calibration_factor(factor=calibration_factor)
```

```
In [ ]: # Save data using new filename
data_hammer.save_data()
```

Note that this will only work if you saved your datasets above after calculating the transfer functions. If you didn't, then you can go back to any of the logger tools, recalculate what you need, and resave your data. You don't need to restart each logger, you can continue wherever you left off using:

```
In [ ]: loggerN.show() # for N=2,3,4 corresponding to the different sections of the lab
```

Calibrated Comparison

Now load, calibrate and compare your data from all three measurements as the culmination of this part of the experiment. Use your best data from each section (i.e. don't forget to use your averaged noise input data).

1. Use 'Load Data' to load each of the datasets to be compared:
 - LOAD THE CALIBRATED HAMMER DATA FIRST
 - Then load the uncalibrated data from the stepped sine and noise input tests
 - When loading the noise input dataset choose the one with your best choice of N_frames
2. Select 'TF Data' from the 'Figure selection' dropdown to see the transfer function data.

3. When all three transfer functions are shown, then zoom into the frequency range 0-500 Hz and press 'Best Match (to ref)'. This finds and applies a calibration factor to match the amplitudes of the visible datasets to the amplitude of the first dataset: this is why you needed to load the calibrated hammer data first.

- **CHECK 1: Did you remember to save your data calibrating the data?**
- **CHECK 2: Did you remember to save the figures that you want to keep for the report?**

Remember to save your calibrated comparison data and figures to the teaching system

For the report:

1. Show your working and calculated value for the calibration constant
 2. Show a plot of your final calibrated set of transfer functions. Don't forget to use the calibration constant on the hammer data then match the shaker data to the hammer data, not the other way around.
 3. Comment on the accuracy of your calibration.
-
-

3.5 Coupling of subsystems (20 minutes)

Finally, we will measure the transfer functions of the uncoupled beams to predict the coupled response.

Using the hammer method obtain admittances for the two cantilevers separately (by removing the coupling rod, removing the magnetic shaker attachment, and moving the accelerometer appropriately — you may need to practice your upside-down hammering technique!).

1. Start a new logger below, and change the 'test_name' as appropriate for each beam.
2. Get the best result you can for each beam.
3. Save the dataset for each beam after calculating the transfer function.

Remember to save your data and figures to the teaching system

```
In [ ]: # acquisition setup
logger4.close() # close the previous logger and start a new one
settings = dvma.MySettings(channels=2,
                           fs=3000,
                           stored_time=6,
                           pretrig_samples=100,
                           device_driver = 'nidaq')
logger5 = dvma.Logger(settings,
                      test_name = 'upper beam',
                      default_window=None)
```

Now you can use these measurements to predict the response of the coupled system. The admittance of the coupled system is related to the admittances $Y_1(\omega)$, $Y_2(\omega)$ of the two separate systems by the formula:

$$\frac{1}{Y_{\text{coupled}}} = \frac{1}{Y_1(\omega)} + \frac{1}{Y_2(\omega)}$$

which is derived in Section 7 of the lecture notes, and is similar to the formula for parallel resistors.

To calculate the coupled admittance using these measurements, load the data then use the "lab_3c6.coupled_TF" function (code below).

Note that this will only work if you saved your datasets above after calculating the transfer functions. If you didn't, then you can go back to each logger tool and resave your data. You don't need to restart each logger, you can continue wherever you left off.

```
In [ ]: data_lower = dvma.load_data()
```

```
In [ ]: data_upper = dvma.load_data()
```

```
In [ ]: data_coupled = lab_3c6.coupled_TF(data_lower,data_upper)
```

```
In [ ]: data_coupled.save_data()
```

Load the saved data using the logger:

1. Start the logger using 'logger5.show()' in the next cell.
2. Use 'Delete All' to clear the data.
3. Use 'Load Data' to load your coupled prediction from above, and your uncalibrated hammer data from earlier.

- **CHECK 1: Did you remember to save your comparison data?**
- **CHECK 2: Did you remember to save the figures that you want to keep for the report?**

Remember to save your comparison data and figures to the teaching system

```
In [ ]: logger5.show()
```

For the report:

1. Compare your predicted transfer function with the directly measured transfer function. Does this method give a reliable way to deduce the behaviour of coupled systems? If not, why not?
 2. Comment on similarities and differences, suggesting reasons for any differences observed.
-

AT THE END OF THE EXPERIMENT

- **check you have saved your data and figures to the teaching system**
- **share your data with lab-group partners and your own devices (e.g. using [Firefox Send](#) or [wetransfer](#))**
- **sign-out of the pc**

NOTE:

- Please note that the state of the loggers is not saved along with the notebook itself, so when you close the notebook then you can no longer use logger.show()
- Therefore, please remember to save data and figures as needed for the report before leaving the lab
- The pydvm logger is opensource, so you can install it on your own device (windows/mac/linux). Contact tb267 if you have any problems.

4. Write-up for laboratory report (3 hours)

The write-up of this experiment as coursework for module 3C6 should be short. Download the template from the 3C6 moodle site (bit.ly/cued3c6). As a guide, the text should be no more than three A4 sides, plus appropriate plots of experimental results. Your report should include representative results from all the stages of the experiment, including the final calibrated comparison of all measurement methods. It need only give a very brief description of the experimental procedure, but it should contain answers to the various questions asked in this sheet, and a critical discussion of the results which might take the form of short bullet points in reference to the experimental results.

When writing up, please concentrate on the following:

- Use diagrams to illustrate your points and tables or figures to summarise data.
- Comment on interesting features of your plots (frequency and amplitude of peaks, units of vertical axis, unexpected features in the data, etc.).
- Make sure you discuss the major features of the measured data (the main peaks, comparison between methods, validity of system-coupling theory).

- There is no need to regurgitate chunks of Part I material. Refer to such material in passing while making particular observations on measured data.

The report should be submitted online using the 3C6 Moodle page as a PDF file, within the usual time allowed for lab hand-ins. You should then attend the next available feedback session for discussion — please allow at least 24 hours for report marking between hand-in and attending a feedback session. Feedback sessions last for 30 minutes — the feedback dates and locations are on Moodle. Attendance at the feedback session is very strongly advised — but if you can't make the designated session and there is a later one available, you can attend that instead. If you intend to submit an FTR on this experiment, you are advised to do the experiment early enough in the term that you can attend a feedback session well before the FTR hand-in deadline.

5. Write-up for Full Technical Report (10 additional hours)

Guidance on the preparation of Full Technical Reports is provided both in Appendix I of the General Instructions document and in the CUED booklet A guide to report writing, with which you were issued in the first year. The experimental plots from your Laboratory Report should be included. The written material from your Laboratory Report will be superseded by this fuller account, perhaps corrected in the light of information from the feedback session. Remember that this report will be assessed both on technical content and on quality of presentation. It should be written as a technical report using a full sentences and paragraphs (not just bullet point lists).

The following additional questions should be discussed in your Full Technical Report:

1. Does the form of the transfer functions you have measured agree with what you expect based on the lecture notes for the module?
2. What governs the shape of the force pulse from the hammer? How does this then determine the bandwidth of the frequency spectrum of the force? *Hint: look in the Information Data Book.* Why does the hammer have a rubber tip?
3. How well do the measured and 'predicted' admittances of the coupled cantilevers agree? Are the predicted frequencies the same? Are there other discrepancies? How do these discrepancies arise? How could the measurement procedure be improved to reduce discrepancies?
4. What are the main advantages and drawbacks of the three approaches to forcing (sinusoidal, random and impulsive)? For each of the three, suggest a type of application for which it might be the best. Think about size, power, method of attachment, test speed, frequency range, signal-to-noise ratio Do not fall into the trap of thinking that the 'best' test method is the one which has an input force 'most similar' to the type of forcing in the real application — remember that provided linear theory is applicable, any type of input force should theoretically give the same transfer function.
5. What do your measurements tell you about the damping of the cantilever system? *Hint: refer to your Mechanics Data Book.* Why might you expect more damping in the coupled system?

The report should be submitted online using the 3C6 Moodle page as a PDF file.