
Engineering Tripos Part IIB Module Experiment: 4C6

Experimental Modal Analysis, South Wing Mechanics Laboratory

The logged data will be saved to a local folder on the desktop. At the end of the session you must transfer your data to you and your lab partner. If you do not do this your data may be deleted by the next students attending the lab.

1. Objectives

- To explore the technique of experimental modal analysis on a structure of your choice;
- Obtain high quality transfer function estimates across a grid of test sites on a structure;
- Find the mode shapes corresponding to the first few resonant frequencies;
- Quantify the modal parameters of the first few modes;
- Obtain and compare damping estimates using the 3dB bandwidth, circle fitting, and sonogram decay fitting methods.

2. Introduction

Experimental modal analysis is a modern technique of vibration investigation which has become very popular. Provided the modal overlap of a system is low, it is reasonably straightforward to recognise resonances by peaks in the vibration response and to map out the corresponding mode shapes by determining how the amplitude and phase of that modal contribution varies over the surface of the system. This information can be used to validate theoretical models of the structure, or to allow design modifications to be undertaken on a rational basis when a vibration problem has to be solved.

The usual method is to measure transfer functions from a grid of points distributed over the system. One envisages applying a force at a fixed point and measuring the vibration response (displacement, velocity or acceleration) at the grid of points. In practice, it is often easier to take advantage of the reciprocal theorem: a fixed sensor, usually an accelerometer, is used, and force is applied to the structure at each of the grid points using a force-transducer hammer. This is the approach to be used in this experiment. The system you will use has much in common with commercial modal analysers, but you will be required to do certain stages by hand rather than having the computer do everything automatically.

Your report should describe the key results of the experiments, with photographs of your test structure and plots of relevant results. Please include photographs of the test object from a variety of angles, including the measurement grid markings, and also include a photo of it when suspended/supported for measurements. A number of specific questions to be addressed are contained in the detailed instructions in Section 4. Your completed report should be submitted online as a PDF file using the 4C6 Moodle page before the last lecture of term when a feedback session on the experiment will be held.

The logged data will be saved to a local folder on the desktop. At the end of the session you must transfer your data to you and your lab partner. If you do not do this your data may be deleted by the next students attending the lab.

3. Apparatus

The apparatus comprises a frame with bungees ready to suspend your test structure. Your chosen test structure should be:

- under 4kg (health and safety)
- without sharp hazardous protrusions (health and safety)
- not valuable (the department won't replace any damaged items)
- ok to use a sharpie pen to mark a grid of measurement locations
- small enough to suspend over the South Wing Mechanics Lab benches within the frame
- light enough damping to obtain mode shapes with isolated peaks
- simple enough shape to map out a grid of points and (in the report) sketch mode shapes
- complex enough to give interesting results
- good examples include (but not limited to): saucepans / dinner plates / breakfast bowls / ...
- bad examples include (but not limited to): ruler (too simple) / laptop (too expensive) / Stradivarius violin / ...

You will be provided with:

- an instrumented hammer to apply controlled forces to it together with a selection of different tips so that you can select an appropriate one for your structure (e.g. considering input bandwidth)
- an accelerometer to measure the response and some wax to fix it to your structure
- charge amplifier with variable gain (adjust the dial on the top rather than the push-buttons on the bottom)
- a datalogging computer with pydvma for acquisition and analysis

The logged data will be saved to a local folder on the desktop. At the end of the session you must transfer your data to you and your lab partner. If you do not do this your data may be deleted by the next students attending the lab.

If the computer is not already on, switch it on, then log in using your usual Teaching System ID and password. If the power is not on, the switch for the computer is on a pillar at the north end of the bench, and the switch for the bench apparatus is somewhere on the central white trunking; follow the wires back to find the correct place. **You may need to plug in the two BNC cables** carrying your input and output signals to the diecast box on the bottom shelf of the computer trolley. Your cables are labelled "4C6 CH1" and "4C6 CH2", to be plugged into the corresponding channels. Note these numbers will correspond to channels 0 and 1 in the legend of your plots.

4. Experimental procedure

4.1 Measuring one transfer function

The first stage of the experiment is to measure one transfer function, to find out roughly how the system behaves and to familiarise yourself with the equipment and software.

First attach the accelerometer to somewhere on the structure, the exact location doesn't matter much at this stage, but consider putting it either near an edge, or off-centre of a plate section of the structure.

To use the wax (in a tiny pot): if there is none on the accelerometer already, then scrape out a tiny quantity, warm it by rubbing in your fingers, then squash three dots onto the contact-circle area of the accelerometer. Then firmly squash the accelerometer on to the chosen location. The wax cools and hardens making a good but temporary contact. Repeat later in the experiment when you choose your 'fixed' location for the grid measurements.

Next check that there is no 'lab_4C6' folder on the desktop (but keep any folders with custom names like 'lab_4C6_CRSID'). If there is a 'lab_4C6' folder with data inside, then the previous students may have forgotten to transfer their data. Please rename the folder to 'lab_4C6_yyyymmdd' then continue.

Double-click on the 'Start 4C6_new.bat' file on the desktop. This copies across a fresh copy of this notebook to the folder 'lab_4C6' on the desktop, then it opens Jupyter Notebooks starting in this folder. You may be prompted to overwrite the existing files - if so then abort and check the note above, renaming the folder if it contains previous student data. If you need to (e.g. if you accidentally deleted part of this notebook), then you can get a fresh copy by running 'Start_4C6.bat' again.

Click '4C6_Notebook.ipynb' to open.

The next cells import the necessary modules: pydvma is a python package written for data acquisition at CUED.

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 pydvma as dvma
```

```
In [ ]: %matplotlib widget
```

Choose your acquisition settings. Try these settings initially, but you are likely going to need to adjust them depending on your structure and the input bandwidth of the hammer. A bit of trial and error is needed. Make a note of your chosen settings.

- channels=2 (number of channels to record)
- fs=10000 (sampling rate in Hz)
- stored_time=5 (time in seconds to store data)
- device_driver = 'nidaq'

```
In [ ]: # acquisition setup
settings = dvma.MySettings(channels=2,
                           fs=10000,
                           stored_time=5,
                           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 structure gently with the instrumented hammer in a few locations. Check that you can see a response in the oscilloscope: the top plot shows the time data, the middle shows the FFT, and the bottom shows the levels. If you get no data at all, check that you've plugged the cables into the computer and that everything is switched on. Adjust the gains of the two channels on the charge amplifier (hammer and accelerometer) until you get a good size of signal without clipping.

Once you have checked that the signal levels are ok, **close the oscilloscope**. You can reopen the oscilloscope again later from within the logger window, but the logger window won't open unless you close the standalone one first.

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

Now open the logger window by running the next cell (shift+enter). The settings now include "pretrig_samples=100". This will store the data that starts 100 samples **before** the impulse is detected (after pressing "Log Data").

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.

```
In [ ]: # acquisition setup
settings = dvma.MySettings(channels=2,
                           fs=10000,
                           stored_time=5,
                           pretrig_samples=100,
                           device_driver = 'nidaq')
logger1 = dvma.Logger(settings)
```

Next:

1. Press "Log Data". The logger will wait for you to use the instrumented hammer to tap the structure (20 seconds before timeout).
2. Tap the structure gently with the instrumented hammer at a chosen position (consider the driving point first).
3. The measured data will appear in the figure when logging completes after 5 seconds.
4. You can examine the results closely by zooming, using either the tools at the bottom of the figure, or the axis options in the left panel.
5. You can display either channel separately, or both together, by clicking the legend lines. You can hide or show all lines using the buttons in the left panel. You can cycle through your selection using the arrow buttons..
6. Check your results. Are they clipped? Was there a single, clean impact or multiple impacts? Be prepared to practise your tapping technique until you get good results. If you get no data at all, check that you've plugged the cables into the computer and that everything is switched on.
7. If you want to redo a measurement, then press "Delete Last" or "Delete All", and re-measure. You should end up with a single good measurement set.
8. If you think that the logger has stopped responding, you can restart this notebook by selecting 'Restart & clear output' from the 'Kernel' menu of the notebook.

Obtain the corresponding frequency response using the "Calculate FFT" button (right panel), which takes the Discrete Fourier Transform of both channels of data, using the Fast Fourier Transform algorithm. The "window" dropdown should be set to "None".

The results will appear in the figure. Note that the top left dropdown shows the view is now "FFT Data". The display will initially show the log magnitude, but you can also plot the phase, the real or imaginary parts, or the Nyquist plot (real part against imaginary part in the complex plane) via the menu option "Plot type" in the left panel.

After examining the details, you can obtain a transfer function using the "Calculate TF" button, with "window" and "averaging method" both set to "None". This will divide the accelerometer spectrum by the force spectrum, to obtain a transfer function which should be independent of the precise forcing used to obtain it.

- Press 'Save Figure' to save the plot: it will save a PNG and PDF of the figure as seen on the screen.
- Press 'Save Dataset' to save all the data to disk (it will save the data that has been calculated so far, so press it after you have calculated the FFT and Transfer Functions)

Save your figures at each stage of the lab so that you can insert them directly into your report.

The transfer function you have measured will show evidence of noise, making the plot less clear than you might ideally like. One source of this noise can be investigated easily. In the right panel, press "Clean Impulse". The signal in Channel 0, the hammer force, will be processed by detecting the time range when there is significant force, and tapering the signal down to zero outside that range. Then press "Calculate TF" again. This should result in a cleaner transfer function. Discuss why this works in your report. Use this "Clean Impulse" button before calculating transfer functions throughout this experiment.

You will need to do some trial and error measurements: use different settings so that you obtain a good clean transfer function estimate over a bandwidth that contains a set of clear distinct modal peaks (in the range 5-20 peaks is suitable). You will need to adjust sampling frequency, hammer tip, charge amp gains, and low pass filters if needed, until you obtain good results. Once you have the settings about right, then the next portion of the experiment will study these modes, to obtain accurate estimates of the frequencies, damping factors and mode shapes. To introduce the method you should work through one case by hand.

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

You can check you have saved your data by loading and replotting the results:

```
In [ ]: data_single_measurement = dvma.load_data()
data_single_measurement.plot_time_data()
data_single_measurement.plot_freq_data()
data_single_measurement.plot_tf_data()
```

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: [<ModalData>]
  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 time, frequency and transfer function data types, and a <ModalData> set (which you will use later). 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 average' 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.

4.2 Determining modal frequencies and damping

You will need to keep the logger window open. If you have already closed it then **don't worry** you can re-open it, without losing any data, using:

```
In [ ]: logger1.show()
```

Don't forget to save your data and figures so you can use them for your report.

Examine the log magnitude plot carefully in the vicinity of the strong resonance at approximately 250 Hz. Estimate its frequency and bandwidth (to the 3 dB points). One way to do this is to:

1. Zoom into the peak using the box zoom at the bottom of the figure
 - You can zoom in by left-click-dragging rectangles, or zoom out by right-click-dragging rectangles
 - Press "Auto Y" in the left panel
 - Use the text box to set ymin to 3dB less than ymax
 - Iterate xmin / xmax until it fills the figure
 - You might want to turn the legend off using the legend controls in the left panel so that it's not in the way.

From the bandwidth estimate the modal damping factor. To what accuracy do you think you can obtain the damping factor by this method?

Now select a frequency range in the vicinity of this resonant peak, i.e. widen the displayed frequency to include around 20–40 dB from the peak value (making sure just one peak is in view). Using the "Plot type" dropdown select "Nyquist", and you should see a portion of a circle, as was explained in the lectures.

Now use the tool selection dropdown (top-right) to choose "Mode Fitting". Check that "TF Type" is set to "Acceleration" and tweak the frequency range if needed using the fmin / fmax settings. Then press "Fit". This will attempt to fit a single term of a modal summation to the displayed peak: if successful you should now see a computed circle that matches the data reasonably well, and a message that gives a summary of the fitted modal parameters (natural frequency f_n , damping factor z_n , the Q-factor $Q_n = 1/(2z_n)$, modal coefficient a_n , and phase p_n - the phase is the angle from the origin to the centre of the circle). If the fit looks like a poor match, then try again using a different frequency range.

Compare your estimate above with the modal fit.

CHECK 1: Did you remember to clean the impulse data before calculating the transfer function?

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?

CHECK 4: Did you remember to check that you are happy with the saved figures? (find the files using windows explorer and open the pdf's)

Remember to save your data as well as all figures

4.3 Measuring the grid and finding the mode shapes

You are now ready to investigate the mode shapes. For simplicity we will start afresh (and re-do the previous measurement). Make sure you have saved your data, then close your first logger window before starting the new one.

Choose and draw on a grid of points for your measurements. A ruler and sharpie is provided. Aim to span most of the surface of your structure with your grid, and try to choose a suitable resolution so that you can capture a range of mode shapes. If your grid is too fine then it will take a very long time to do all the measurements, if the grid is too coarse then you won't have a clear view of the mode shapes. Somewhere in the range of 20-40 positions is likely to be sufficient.

Label your measurement positions from 0 to N, and carry out your measurements in this sequence so that the logger legend labels will match!

Choose one of the locations for your accelerometer: it doesn't matter exactly which one, but aim for close to where you have done previous measurements so that you are less likely to need to change your settings. Put the accelerometer on one of the grid locations on the opposite face of the structure, so when you reach the driving point then you can tap on the grid point directly.

Note: as you do all the measurements you may find that the settings need a bit of adjustment (e.g. the gain). If so, then make the necessary adjustments, and start again!

```
In [ ]: # acquisition setup
logger1.close()
settings = dvma.MySettings(channels=2,
                           fs=10000,
                           stored_time=5,
                           pretrig_samples=100,
                           device_driver = 'nidaq')
logger2 = dvma.Logger(settings)
```

For each measurement position in turn, from 0 to N, do the following:

1. Press "Log Data"
2. Use the hammer to deliver an impulse to the appropriate position: **strike the grid intersections as accurately as possible**
 - if you see a warning about clipped data then press "undo" and "Log Data" again
 - if you think you have not done a good measurement (e.g. double tap or too gentle hit), then press "Delete Last" and "Log Data" again
 - the triggering mechanism will time-out after 20 seconds if the threshold is not detected, so don't wait too long before tapping. Be careful to hold the hammer still during the logging time, i.e. don't put it down on the bench during this time!
3. Repeat until you have a single good measurement for each point. Note that the legend labels each measurement as a 'Set', which should correspond to the grid point labels you have measured.

Press "Save Dataset" to store the Time Data you have just logged.

To reset the figure axes at any stage, press 'Auto X' then 'Auto Y'.

You will have a large collection of data: before continuing, maximise the logger window if you haven't already, then zoom in on the initial impulse. Press "Show Chan Only" which will only select the lines from Channel 0 of each set. Make sure that they are all a reasonable amplitude (~0.3 or above), and with no double-hits. If any show a double hit, identify which 'Set' it belongs to, then choose "Edit Dataset" from the tool selection dropdown (top-right), enter the set to replace into the text box, then press "Log & Replace Selected Set". This will log data as usual, but the data will directly replace the chosen set.

Once you have a good set of time data, go back to "Standard Tools" (top-right dropdown) and press "Clean Impulse". This will clean all the impulse data channels. Then press "Calculate FFT" and "Calculate TF".

The result will be N sets of transfer-function data, corresponding to the N grid points that you labelled. Look at the transfer function set carefully - do all the channels look good, or do some of them look wrong in one way or another? If so, identify the doubtful sets, and use the "Edit Dataset" tool to re-log data for the corresponding positions. Don't forget to "Clean Impulse", "Calculate FFT" and "Calculate TF" afterwards.

To see individual channels, press 'None' under 'line selection' in the left panel, then use the legend to select a line. You can then use the previous (<) and next (>) arrows to cycle through the selection.

Press "Save Dataset" to store all the data you have logged and calculated.

If for any reason you have to restart, you can read these transfer functions back in quickly using the "Load Data" option.

Now proceed to the modal analysis proper. Display all channels in "Amplitude (dB)" form, and select a frequency range in the vicinity of a resonant frequency, just as you did in the previous stage when you performed a modal fit. Select "Nyquist" from the plot type dropdown to reveal a set of portions of circles. Now select "Mode Fitting" and press "Fit", and the logger will work through all N channels, trying to fit a single frequency and damping factor for the dataset, and amplitude and phases for each position. (The phase is the angle from the origin to the centre of the circle.)

After processing each channel, the logger will list the results in the message area, as well as show the locally reconstructed transfer functions. The reconstructions are placed in a single new set with N channels. In the left panel under "Line Selection" press the "None" button then use the legend to select specific lines to compare each reconstruction with the corresponding measurement.

You know that a given mode manifests itself in all of your measured transfer functions and moreover that it should in theory always have the same frequency and damping factor, but the amplitude will vary from place to place (make sure you know why). If certain sets show poor fits, then something has gone wrong with the fitting procedure. Look at the data carefully in any such cases. Most often, this problem arises when the relevant peak in the transfer function is very tiny, implying that the corresponding point on the structure is near to a node line of that mode. This is useful to know for sketching mode shapes later on. In other cases, the problem may arise because the optimisation routine used to fit the modal parameters has not converged. Try taking a slightly different frequency range and repeating the calculation. If the problem won't go away then simply ignore any rogue data for the purposes of forming your damping factor estimate.

Repeat this exercise for each of the resonances that are well defined / well separated. This is easiest using the "Amplitude (dB)" view, where adjusting the frequency range is more straightforward: **you don't need to have "Nyquist" view selected for the fit to work.** To replace a previous fit then simply zoom into the mode to replace, a message will appear if you have already fitted this mode, and press "Fit". To delete mode fits then select a frequency range containing the modes to delete, a message will appear to summarise the modes fitted within that range, and press "Reject".

Spend a little bit of time getting a good fit for each mode. If you deal with oddities now the quality of your mode shapes will be much improved which will make things easier for you later on.

Remember to save your data (after fitting all the modes) and figures to the teaching system

Once you have processed each mode the mode shapes can be visualised simply by drawing a sketch of the structure with its grid of points, and writing the modal amplitudes on it. Do this for each mode, and try to deduce the mode shape: sketch in the position of any nodal lines, and describe the motion briefly in words.

The modal amplitudes are in a 2D numpy array, with each row showing the 12 amplitudes for a given mode. The actual values are awkward to read, so for convenience we can scale them to be in the range [-100,+100], and print the result:


```
In [ ]: modal_data = logger2.dataset.modal_data_list[0]
for n_mode in range(len(modal_data.fn)):
    fn=modal_data.fn[n_mode]
    zn=modal_data.zn[n_mode]
    an=modal_data.an[n_mode,:]
    print('Mode {}: fn={:.3f}, zn={:.5g}, Qn={:.1f}'.format(n_mode,fn,zn,1/2/zn))
    scaled_amplitudes = an/np.max(np.abs(an)) * 100
    print('scaled amplitudes={}'.format(np.array2string(scaled_amplitudes,precision=0)))
    print('_____')
```

Make a note of these values, or save this notebook, or copy-paste the text in an email to yourself!

4.4 Reconstructing Transfer Functions

One standard way to check the accuracy of the modal parameters determined by the procedure you have just followed is to reconstruct some of the transfer functions using the theoretical expression together with the parameter values you have found. If the transfer functions match, the analysis is confirmed, while any mismatch will give clues as to what is missing in the analysis.

Press the "Reconstruction" button in the "Mode Fitting" tool. This will use the data from your previous mode fits in the modal summation formula to reconstruct all 12 transfer functions. Recall that the reconstructions are placed in a single new set with 12 channels. In the left panel under "Line Selection", press the "None" button and use the legend to select specific lines to compare a reconstruction with its corresponding measurement. Once you have selected a pair for comparison, you can then use the arrow selection buttons to increment the comparison.

Once the reconstruction is complete, you should compare it directly with the measurement. How well do they fit? Comment in your write-up. Does the driving point measurement agree with expectations (antiresonance patterns)?

Remember to save your data (after doing the reconstruction) and figures to the teaching system

CHECK 1: Have you remembered to save the figures you need for your report?

CHECK 2: Have you saved your dataset after carrying out all the processing?

CHECK 3: Have you remembered to save the figures you need for your report?

4.5 Alternative damping estimates

Choose a single transfer function measurement set from your grid. For the modes identified above, use the 3dB bandwidth method to obtain an alternative estimate of the modal Q-factors for one of the transfer function sets. How well do they agree?

For the chosen set, use the sonogram tool to obtain a sonogram-decay estimate for the modal damping. How well do these values agree? How linear do the damping decay curves appear for each mode? What are the dominant sources of damping for your structure?

Note that you can re-open any of the logger instances you have previously started, and it will re-open where you left off including with any data so far stored. For example, to restart logger2 use:

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

AT THE END OF THE EXPERIMENT

- **check you have saved your data and figures to the local folder on the desktop**
- RENAME YOUR 'lab_4C6' folder to 'lab_4C6_CRSID' with your CRSID
- share your data with yourself and lab-group partners (e.g. using [Firefox Send](#) or [wettransfer](#))
- YOUR DATA WILL BE DELETED AT THE END OF TERM, BUT THERE IS NO GUARANTEE THAT IT WON'T BE DELETED ACCIDENTALLY BEFORE THEN
- 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 pydvma logger is opensource, so you can install it on your own device (windows/mac/linux), however it is still in development so may not work.

Installation instructions

1. Install anaconda by downloading from <https://www.anaconda.com/distribution/>

- Open Anaconda prompt and type:

```
conda install numpy scipy jupyter matplotlib pyqtgraph ipympl ipywidgets jupyterlab  
pip install sounddevice pydvma
```

- If you get an error about not having the correct permissions, then try opening Anaconda prompt by right-clicking and choosing 'run as administrator'
- then open Jupyter notebooks from the windows start menu, and this notebook should work.

In []: