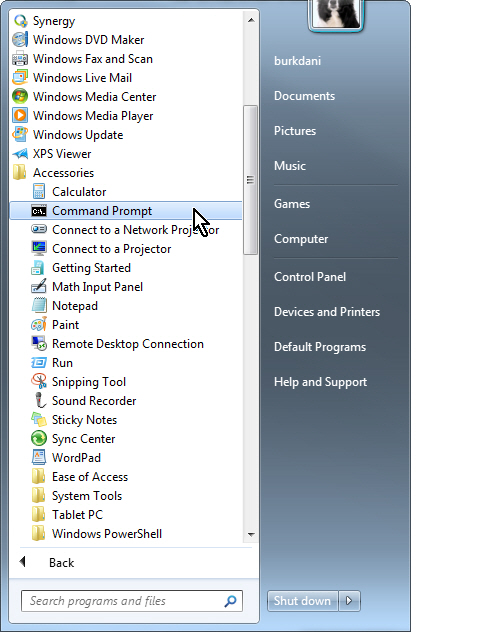
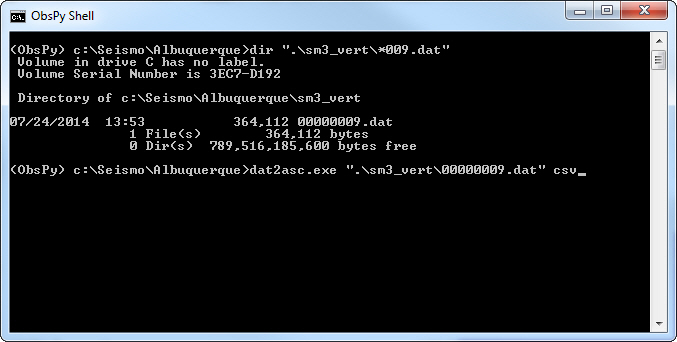
The Calibration process cookbook

STEP 1: Figure out the process of turning the data acquisition data into ASCII, comma separated files.

 The utility called “DAT2ASC.EXE” will convert Symmetric Research files into a comma separated file format which is used for processing all of the calibration data. This is the only format currently supported by this process.

This process will be used to convert all DAT files used for calibrating all parts of the system, and is run from the command prompt. In Windows, find the command prompt window which is generally found under “accessories”:

DAT2ASC.EXE is provided in the utilities from Symmetric Research. The specific version necessary is dated from 10/5/2010 and is 171,008 bytes in length. This program will seek the target file that you direct in the command prompt, and it will create a .CSV file containing the data. The program will also convert and append all consecutive files that are in the same directory as your target file. If you wish to convert only one file, you must isolate it from any other files with consecutive names.



Once each file is converted, Dat2ASC will create a file called “Dat2asc-301-Data.csv” that contains your output data. Rename this file before converting subsequent files or it will be overwritten.

C:\mydirectory>RENAME Dat2asc-301-Data.csv mycalibrationfile.csv

STEP2: Calibrate the system constants

Before Sigcal can create a calibration curve for the sensor, a few calibration constants must be defined:

* The station name
* The sensitivity of each channel of the Analog/Digital Converter in terms of microvolts per count
* The sensitivity of the laser position sensor in terms of millivolts per micron of ground motion
* The ratio of the length of fulcrum to center of mass, divided by the length from fulcrum to laser target. This is known as “lcalconst” and is required in order to resolve laser displacement with actual seismometer mass displacement.
* The total damping ratio “h” for the seismometer as a response to an impulse
* The seismometer free-period resonance frequency.

These calibration constants must be measured and recorded accurately if we wish to reconcile the mass movement to actual ground motion and seismometer coil signal output. Therefore, be sure to accurately calibrate your digitizer by applying a set voltage step to each channel and measure the resulting change in counts . Divide the voltage by the number of counts offset, then multiply by 1,000,000 to yield a calibration constant of microvolts per count. Calibrate also your laser position sensor to determine its actual voltage output in terms of millivolts per micron. You may then generate your calibration constant file by running the python application, “calconst.py”

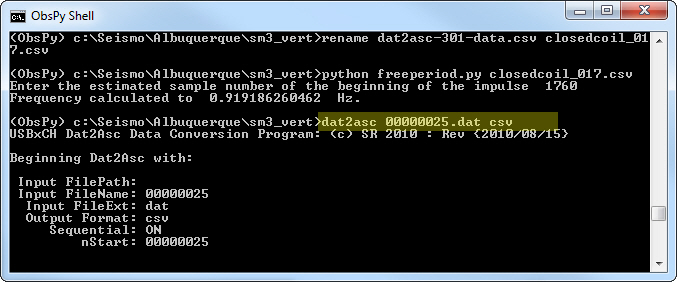
STEP 2 part a: Determining Free period for the sensor

Useful Tip # 1

When managing the data files, be sure to create a separate directory within called “archive” in which you move all your non-important .dat files. Then create another directory called “processed”. Each test that you convert from .dat should go into this directory after you are done converting the files into .csv format.

The dat2csv program looks for consecutive files. Therefore if you have two tests with consecutive filenames, such as 006 and 007, you should convert file 007 first, and then move it to the “processed directory” so that it doesn’t get sucked into the 006 ascii file when you do your next conversion.

In order to calculate the response of the system to the upcoming series of sinusoidal signals we employ to measure response, an accurate measurement of the free-period oscillation of the seismometer must be made. We have created a small python-based analysis tool called “Freeperiod” that accomplishes this task. Freeperiod.py takes a csv input file, and performs signal analysis in order to calculate free-period.

The method of actually collecting the free period data is covered elsewhere: This document deals with the processing of the data.

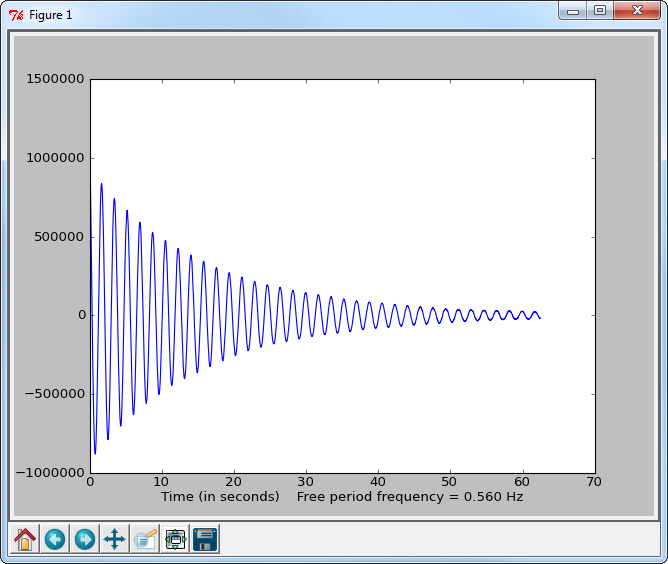
First, convert your file to csv format:

Next, rename your file to something memorable, like “Freeperiod.csv”

Now, analyze the ascii file with the Python utility, “Freeperiod.py”. This program requires that a file called “calcontrol.cal” exist in your existing directory that contains preliminary calibration information. You must first have calibrated both your digitizer and your laser position sensor before you can accurately measure amplitude or calculate sensitivity curves. However, for the calculation of the free period and the damping ratio, it will be okay to have estimates of these values within your calibration control file.

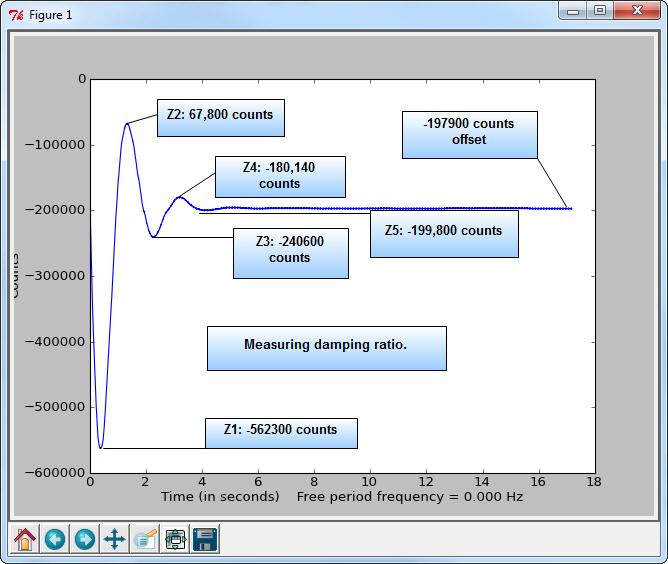
Step 2, part a continued: Analyzing for free period and damping ratio

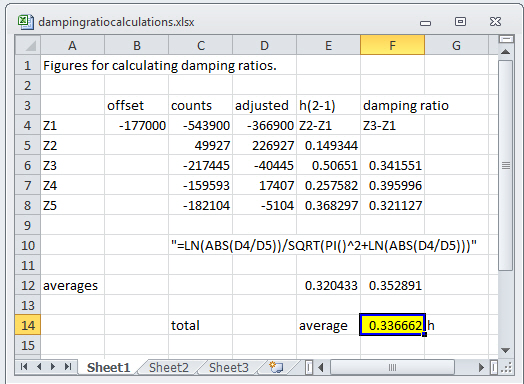
When Freeperiod.py launches, it will open the csv file and display the laser position sensor (ADC channel 4) on a screen. Identify the start of the resonance cycle, which occurs when the mass is released. Use the zoom function to identify the starting sample number. Once recorded, shut down the graphics window to return to the command prompt window. The program will prompt you to type in this value. A second window will now open to display the waveform, including the resonance frequency at the bottom of the screen. You may screenshot this window for recording in the log.



Step 2: Part b: Measure Damping Ratio.

The software for calculating damping ratio is still in development. However you may use the python code “Freeperiod.py” to view the waveform in order to record the waveform impulses for the calculation of damping ratio. There is a spreadsheet included called “dampingratio.xls” that can be used to calculate the damping ratio. There will be several points that you need to hand-record, such as the amplitudes of the three to five peaks in the signal, as well as the DC offset that may be present.



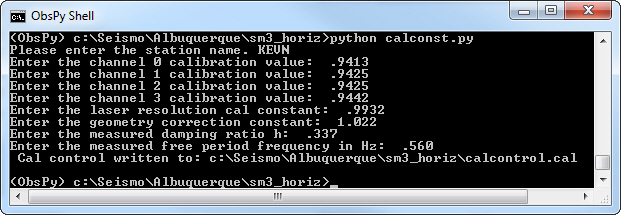


These numbers should be applied to the spreadsheet called “dampingratiocalculations.xls”. The average value should be your damping ratio. Most systems should have just three impulses but some may exhibit five. If noise is present in your signal, you may have to estimate the appropriate value that lies in the median of the noise at the top of the peak. If noise is a huge issue and you cannot with reasonable accuracy measure Z3 through Z5, estimate damping ratio with the first two impulses.

Step 2, part c: writing out the calibration constants

It is assumed that the technique for calculating the true sensitivity of the analog-digital converter has been employed to determine the exact measurement of microvolts/count. It is also assumed that the laser output has been measured by using a micrometer or some other method of determining the number of millivolts for one micron of displacement. It is also assumed that the moment arm ratio between the center of the mass and the measurement point of the laser has been calculated. (This is called the laser cal constant, typically something like 0.535 for the SM3)

Gathering your constants, enter them into the program called calconst.py in order to generate the new calibration constants for processing your data.



The calibration values for channels 1 through 4 represent the number of microvolts for one count of the analog-digital converter.

The laser resolution cal constant represents millivolts / micron.

The geometry correction factor represents the difference between the center of the velocity coil and the actual measurement location. This is important if the laser is mounted a significant distance from the center of the coil. It is calculated by measuring the distance from the pivot point of the seismometer pendulum and the locations to the laser as well as the center of the coil. As the system swings through it’s movement, the displacement of points farther away from the pivot point will be greater than points located closer to the pivot. The ratio of distances is represented by this equation:

Lcalconst = Length of moment arm to the center of the coil / length of moment arm to center of laser pickup point.

The damping ratio and free period frequency have been measured in the previous paragraph.

Once these constants have been entered into the program, the control file is ready for use in calculating the sensitivity curve for the seismometer . This is accomplished by converting the signal files into .csv ascii files, then storing them in a separate directory for processing with sigcal. Each frequency that was tested should be separated into its own ascii.csv file as the program depends on the data being stationary and clean, with no signal offsets or changes in frequency during each test.

It is important that, when collecting the calibration data, you change frequencies BETWEEN logged file runs. The digitized files which contain he frequency change should be moved OUT of the directory structure and stored within a separate “archive” directory before you run the DAT2ASC program. Once your directory has been cleaned, your .DAT structure should have only one, perhaps two files per frequency. These files should now be converted to ascii .CSV using the DAT2ASC.EXE utility. The result should be several dozen .csv files within the directory structure that represents the instrument response to frequency. NO OTHER CSV FILES should be present within the directory: Store your damping ratio and free period ascii files in a separate subdirectory. All that should be present within this directory are the .csv files that you created, and the calcontrol.cal file that you generated a few minutes ago.

Once the files are present, open the command prompt and point it to the working directory that contains the files. Then type “python sigcal.py”. If your paths are set correctly, and if you have the python script located within the working directory, the program should parse through the directory and locate all of your csv files. The result should be a series of status lines as it processes each file, followed by the construction of a calibration output file as well as a graph of the response curve.