Schumann Folder User Guide

/data/procdata/detchar/envnoise/Schumann

VIR-0365A-15

Melissa Guidry

This is a guide for navigating the Matlab scripts and data associated with my work on correlated magnetic noise across Virgo (see VIR-0364A-15 for more details). If you have questions about a specific process, please contact me at maguidry@email.wm.edu.

Contents

(1)-How to Process Data

(a)-Centaur digitizer miniseed files

The script I used for processing miniseed files is called **Asd_Csd_all.m** and can be found in the **scripts** folder. It computes and saves the ASDs for each channel of the digitizer, the CSDs between each channel and a channel of interest (for me, this was the magnetometer channel), the raw count values, and the time values. You simply have to save the miniseed files (in their respective day folders) in the **top_directory**.

1) Go to line 46 of the script. Make sure that you have your day folders full of miniseed files in this directory. In line 47, modify the numbers to reflect the day folders.
2) Modify line 43 to save where you would like to save.
3) In lines 26-37, we have values that you may like to change.
   a. calib_sensor_* --- The calibration values for converting our data
   b. Fsb --- The input voltage range
   c. len --- The length of the files being processed, I usually did an hour so 3600 seconds
   d. T --- The time window for each PSD calculated
4) Go to lines 111-114. The values in I(n) are associated with channel numbers. Change them as you may need. At the top of the script, I have commented a guide – the real channel number is on the left, the converted index number *n* is on the right in parenthesis.
5) Run the script. You now have Pxx_n mat files with groups of five 1-hour PSDs. To merge them into one list, and then one long string of PSDs, use **PSDs_merge.m**. You must modify the directory in line 3 from which the files are stored, as well as the channel of interest in line 21 (e.g.: "P66" for the magnetometer channel). Also modify the names in lines 12, 23, 30, 34, 35, and 38. Add the local start time of the first miniseed file to line 55, and change the length of each PSD (line 52). time_merge will be a vector of GPS times matching the PSD points. After generating these, I generally save all my merges under one merges.mat file in the directory.
6) The PSDs_merge.m script may be modified to merge the time-series data similarly.
7) The code for actually finding coherence is in **plot_all.m** lines 157 – 167.

(b)-Building magnetometer data

I used two scripts for processing environmental sensor data, each with a different purpose: **coh_buildings_field.m** and **sideband_finder.m**.

If you want to eventually find coherence between environmental sensors and data you have already collected, use **coh_buildings_field.m**.

If you want to look at environmental channel PSDs spanning far back in time (without the need to compute CSDs), use **sideband_finder.m**.

1) Enter your "save_to" destination in line 5.
2) In line 24, specify which channel you are interested in.
3) Lines 29 and 32, determine how long each file (pixel) should be and how much long the pwelch averaging time window should be. Greater T allows greater frequency resolution but less averages.
4) Line 35: The sampling frequency you want in the end
5) Line 48: How many PSDs should you save at a time?
6) Change your start and end times in lines 52 and 53
7) I was interested in tracking the amplitude of a sideband at 47 Hz. If you aren't interested in this, comment out all references to "amp_list". If you want to look at a different sideband, modify the frequency range in lines 94-95.
8) I only wanted to save PSD data between 45 and 51 Hz. If you want a different saved frequency window, change the values in lines 97-98.

   a. You may run into a problem where your frequency resolution does not line up with the frequencies you are cutting at (k1, k2, b, or c would appear as empty lists). Use values that are in your generated frequency list F.

9) In the end, you get different .mat files filled with 300 PSDs. To merge them into one file, use **sideband_merge.m**. Change the directory to where your .mat files are saved. Go to line 50 and set p = 1:(whatever your value for p in line 48 of **sideband_finder.m** is). You now have merged files!

(c)-CoCo80 files

I never developed a formal script for processing CoCo80 files, but it would be easy to do so.

1) After the CoCo80 files have been converted to .txt files, I save them in my directory.
2) Using MATLAB, I do "Import Data". In the import window, choose "Matrix" for the imported data format. Select the range: I always started with A28 to bypass the extra table information. Click "Import Selection".
3) Once the matrix (for example *MatrixB* which is a 512x2 double) has been imported, you can use the following lines for later ease of use:
   a. F = MatrixB(:, 1); mags = MatrixB(:, 2);
4) Remember to convert your magnitudes. If they are in the V^2 format, you need to take the square root, convert from Volts to Other Units (using a sensor specific calibration constant), and divide by the square root of the frequency bin size.

(d)-LIGO magnetometer data

Due to my being a Virgo summer student, I didn't have non-GUI access to LIGO data. So I downloaded all LIGO data through their ldvw GUI: https://ldvw.ligo.caltech.edu/

I downloaded raw data from a channel in the form of .wav files, usually with a 3600 second duration (shorter was more work, longer usually made the site crash). It is important to note that the .wav files will have a different amplitude than the originally recorded data (.CSV). Thus, we developed the following conversion using a downloaded .CSV dataset:

csvMAG = 0.109303396 * wavMAG - 3581.7

I developed the script **ligo_Asd_Csd.m** for calculating the PSDs for the LIGO set and the CSDs for coherence studies. It returns data, PSD, and CSD lists for that LIGO magnetometer. It is run under the assumption that the comparison raw data vector has already been computed.

I save the PSDS all to the same .mat file because with only three lists we aren't as slow as the usual >5 lists. This can be modified similar to the **Asd_Csd_all.m** script.

1) Make sure all of the .wav files are saved IN ORDER in the top_directory (line 10). To guarantee this, I saved my .wav files with this example format: 20150622-100000.wav
2) Change your save_to directory (line 7) to where you wish

3) Line 44, the duration of your files (3600 s was what I chose)
4) Lines 47-48: T is the time window averaging for the PSD generation. FSb is the voltage range for the sensor. This should be specified on the ldvw site for the channel you are looking at.
5) Line 57-58: Sorry, this part is less user friendly. In line 57 I decimate the LIGO channel sample rate to something closer to our sample rate of comparison: 8192 Hz to 256 Hz. Then in line 58 I finally change it to 250 Hz. (This is all because our Villa Cristina data were sampled at 250 Hz and the sampling rate should be the same when computing CSDs)
6) "dat6" stands for "data for channel 6, the magnetometer channel". In line 61, it is assumed that dat66_list was imported. You must import your raw data vector for comparison and change the name accordingly.
7) After running the script, we have lists of PSDs. These can be merged with the following code, given a list Pxx_list:
    a. Pxx_merge = [];
    b. for q = 1:length(Pxx_list)
        i. Pxx_merge = [Pxx_merge Pxx_list{q}];
    c. end

    You can generate the time_merge file using code similar to that used in **Asd_Csd_all.m**.

(2)-General Cleaning

Once you have a merged matrix of PSD values, cleaning it is easy. I used the script **PSDs_cleaning.m**. This script outputs a list of "clean" indices for the merged PSD list of interest.

1) Line 3: Which frequencies should be used for determining the "cleaning" threshold? For example, if you want a dataset that is cleaner around 8 Hz, you should clean between 7 and 9 Hz. The indices can be identified by looking at the indices for frequencies of interest in your frequency vector associated with your PSD merge (saved in constants.mat).
2) Line 4: Start with a divide_thresh of 1. If you want to cut more, keep increasing this number and re-running the cutting.
3) Line 8: enter the name of your PSD merge
4) You can uncomment PSDs2 and all its associated code if you want to clean two merges together (very important for coherence analysis)
5) After running, the script returns "clean". To use these indices, for example, I could plot a spectrogram:
    a. Imagesc(time_merge(clean), F, log10(sqrt(Pxx_merge(:, clean))));
6) "dirty" is also returned, which is the indices that were cut.

(3)-Other Useful Scripts

Schumann_amp_over_time.m – A script for monitoring the amplitude of Schumann resonances over time in a merged PSD dataset. Simply have your merges in the workspace and modify the Pxx_merge name in line 3. The indices in lines 15-19 will depend on your Frequency vector resolution, so change it accordingly.

TF_APPLY.m – a script for applying the factory magnetometer transfer function to PSD merges. The transfer function should be imported from: **CHON_OFF_matfiles.mat**

(4)-Data Organization

(a)-LIGO_mags

This file represents all of the raw and processed data for LIGO magnetometers that we were interested in.

1. H1:PEM-CS… -- magnetometeter channel names
    a. July1-3 – days of data
        i. 1hr_wavs – the raw .wav files
        ii. data – constants.mat, merges.mat, PH1.mat… processed mat files
    b. fullmerges.mat – a merged matrix for all days for this magnetometer

(b)-LocationB

This file represents all of the raw and processed data for Location B (1 km down the West Arm).

1. July16-17 – days of data
    a. miniseed – all the miniseed files in day folders
        i. 16 – the day in the month when the data was collected
    b. PSD – contains all processed files
2. noise_jump – contains preliminary attempt to understand sidebands
3. tot_merges.mat – a merged matrix for all days for this magnetometer

(c)-metal_tube_test

This file represents all of the raw and processed data for our first short test with the magnetometer housed in the metal tube near the Central Building.

1. Full Spectro – contains the processed PSD merges for the magnetometer and y-axis seismometer
2. miniseed – contains all the miniseed files

(d)-MFS-06_calib

This file contains everything relating to the transfer function determination of the MFS-06 magnetometer.

1. CHON_OFF_matfiles.mat – the data for the transfer functions. See **TF_APPLY.m**

(e)-MFS-06_noisefloor

This file contains the noise floor data for the MFS-06. **noise_magV2** is in units of $nT/sqrt(Hz)$

(f)-NEfield

This file contains the raw and processed data for the North End field installation.

1. June12-15
    a. cutting – a file containing ASDs for different cutting frequency thresholds
        i. 1-10 – defining a cutting threshold between 1-10 Hz
    b. miniseed – contains the miniseed files in the day folders
    c. PSD – processed NE field data
    d. seismo_coh – contains coherence between the seismometer and magnetometer
    e. WEB_compare – files for comparing NE field with WEB magnetometers
        i. Coh – coherence files with field (lots of figures)
            1. glitches – cutting out glitches
        ii. spectros – spectrograms for different NEB and WEB magnetometers during this time

(g)-sideband_tracking

Contains PSDs for WEB V stretching back to February to look for sideband occurrence.

1. Feb – data stretching back to February. Mysteriously missing July.
    a. PSDs – the processed data files
    b. constants.mat – frequency vector
    c. merges.mat – merged PSDs and time
2. full_spectro.fig – the full spectrogram since February

(h)-VillaCristina

All the Villa Cristina raw and processed data.

1. coco_test1 – the first test we ever did with the CoCo80 outside and inside of the house
2. coco_test2 – second test we did later (analyzing NS, EW orientation in the living room)
3. June22-25
    a. cutting – first attempts at cutting out glitches
    b. mat_files – the data mat files
    c. miniseed
    d. powersupply_coh – looking at coherence with IPS
    e. PSD – processed PSDs
    f. seis_coh – coherence with seismometers
    g. spectro_ASD – spectrograms and ASD figures
    h. WEB_coh – coherence with the WEB magnetometers

(i)-Virgo_CoCositetests

All of the CoCo80 surveys we did around Virgo.

1. maps – ASD values at different sites
2. mat_files – frequency and magnitude vector for each .mat location file. The magnitude is V^2 and must be converted (see part 4 of processing CoCo80 data above).
3. text_files – all of the original text files