Jennette’s Pier Hydrophone Spectral Analysis:

Software Design, Implementation, and Testing

Steve Lockhart

sblockhart.zzero@gmail.com

Contents

Version History

Version#	***Date***	***Author***	***Description***
1.0	06/24/2022	S.Lockhart	• Initial release generates sample plots (PSD, decidecadal spectrum) as per IEC Technical Specification, for a specified wav file
1.1	07/02/2022	“”	• Added a quality check on time series, skipping windows that seem to have a data gap.
1.2	05/22/2025	“”	• Add wind-speed/wind-dir bins. • Although the software has been extended to work with OOI data, this version of the document focuses on the CSI data from Jennette’s Pier. It also focuses on characterizing the *baseline* soundscape.

Introduction

The Coastal Studies Institute (CSI) conducted baseline soundscape characterization according to International Electrotechnical Commission Technical Standard 62600-40 (Marine energy – Wave, tidal, and other water current converters – Part 40: Acoustic characterization of marine energy converters).

To establish baseline conditions, CSI deployed Ocean Sonics icListen High Frequency hydrophones at the seaward end of the pier, where the average water depth is 3-4 meters. The hydrophones were deployed 1 meter above the sandy seabed. CTD casts were also conducted on each sampling occasion to collect data for sound speed profile calculations. The hydrophones were deployed on several days with various wind and wave conditions, from July 2022 through March 2024. As per the IEC specifications, this dataset is not yet complete.

For each deployment, the hydrophone recorded underwater sound for 5 minutes at 512,000 samples/second. The hydrophone's recordings were stored as wav files.

Analyzing the hydrophone data, we grouped the recordings into bins based upon environmental factors that contribute to ambient noise, specifically wind-speed, wind direction, and rain-rate. For each wind-speed/wind-dir/rain-rate[1] bin, we generated plots of the power spectral density (PSD) as well as the decidecadal power spectrum. Our spectral analysis adheres to the IEC Technical Specification (IEC TS 62600-40, Edition 1.0 2019-06).

For each hydrophone deployment, we also analyzed the recording to detect biological[2] sounds, including snapping shrimp noise and marine mammal vocalizations (clicks and whistles). These detections help us interpret the binned spectra.

We generated plots of power spectral density (PSD) and decidecadal power spectrum per wind-speed bin. (The software can also generate these plots per wav file.)

The purpose of this document is to describe the software that we developed for the spectral analysis. We call the software LTAS, as it was originally designed to calculate the Long-Term Average Spectrum. Here, we document the design (process and data), implementation, and testing of LTAS. The software for detecting biological sounds will be documented elsewhere.

---

[1] In this first release of the software, we excluded recordings during rain.

[2] In the first release of this software, the biological signals are included in the spectra. In a future release, we will have the option to exclude them.

Process views

Data exploration/pre-processing

Before performing the spectral analysis, we preview the data to identify any data quality issues, including the following:

- If a hydrophone started recording before the device was placed in the water (or continued recording after the device was removed from the water), we would need to extract the appropriate subset of the wav file—only the samples acquired when the device was in the water.
- The LTAS software assumes all wav files have the same sample rate. The data exploration/pre-processing identifies wav files that do not have the standard sample rate.
- The LTAS software assumes wav filenames are unique. The data exploration/pre-processing identifies duplicate file names.

Currently, the data exploration does not fix issues; it just helps to identify them.

Overview of LTAS Spectral Analysis

The LTAS software includes three components or processing steps:

1) For each wav file in a folder, calculate the power spectral density (PSD) for each 1-second window. Store the PSD array (of dimension #windows x #frequencies) in a mat file (1 mat file per wav file).
2) Get the wind speed and direction associated with each wav file. Store this information in a csv file.
3) Using the outputs of the previous two steps, assign the wav files to wind-speed/wind-dir bins. Concatenate the PSD arrays for all mat files in a wind-speed/wind-dir bin. Then, generate the plots of PSD and decidecadal spectrum per wind-speed/wind-dir bin, showing median, 25^{th} and 75^{th} percentile curves for the bin.

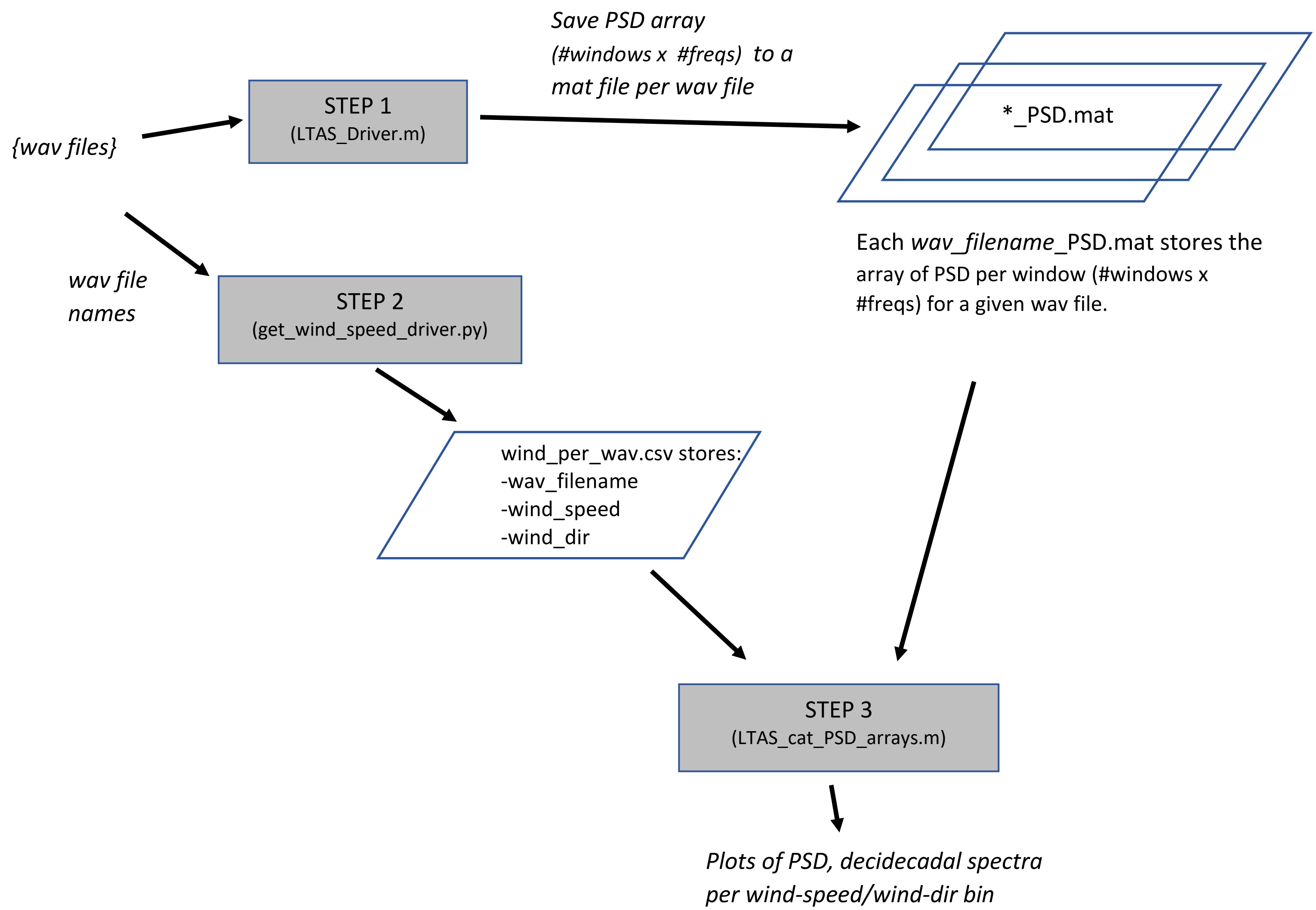

Figure 1(a): A high-level overview of steps 1-3 of the spectral analysis code for Jeanette's Pier data, showing the main program for each step. Data files are drawn as parallelograms.

Step 1: Calculate Power Spectral Density (PSD) per window, per wav file

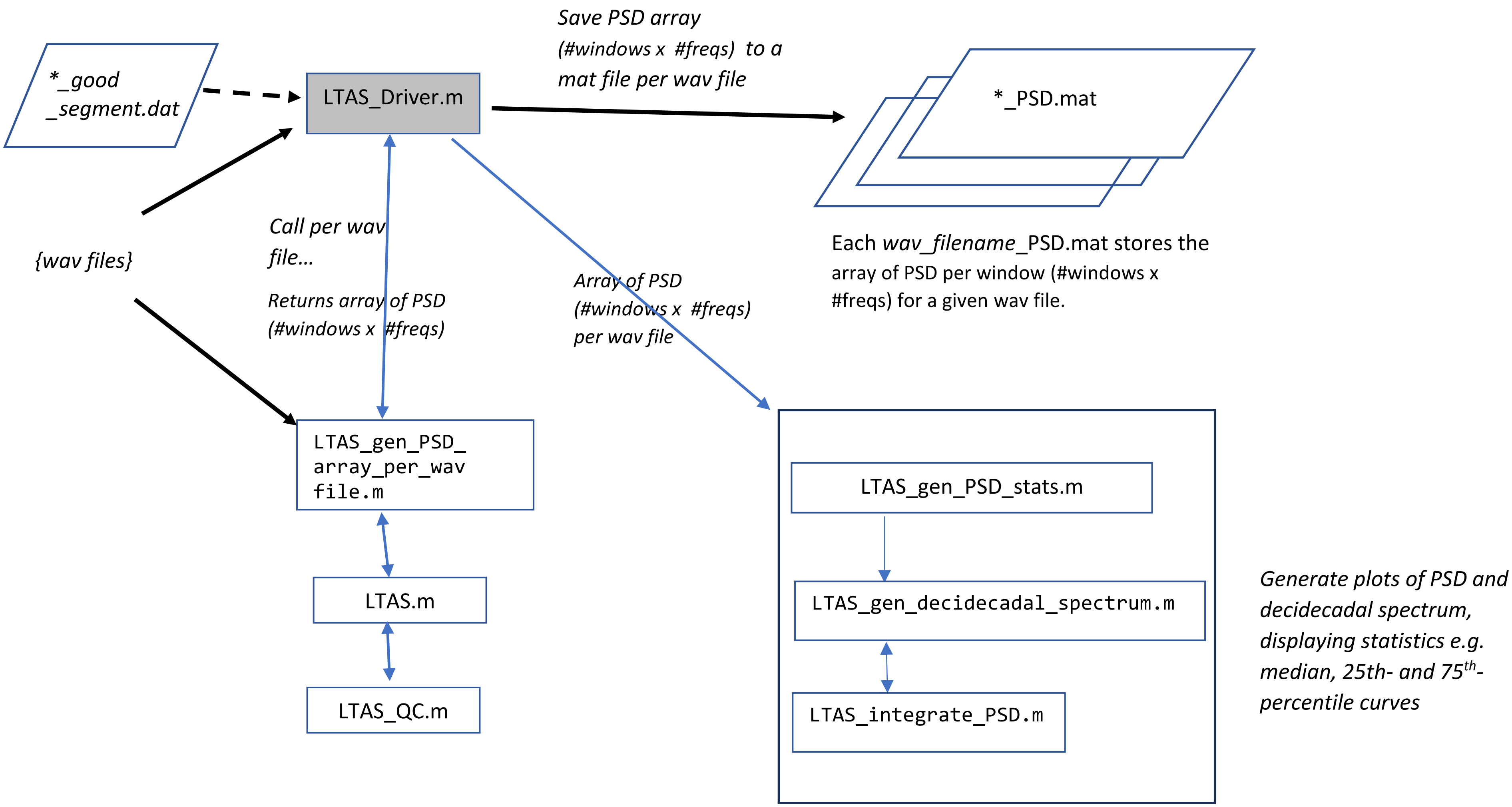

Figure 1(b): A more detailed flow diagram for "Step 1" of the spectral analysis code for Jeanette's Pier data. Each rectangle represents a matlab function or script. A blue arrow indicates a call from one program to another. A black arrow indicates data flow. The gray shading indicates a main program. The output mat file is drawn as a parallelogram.

Step 2: Get wind speed and direction per wav file

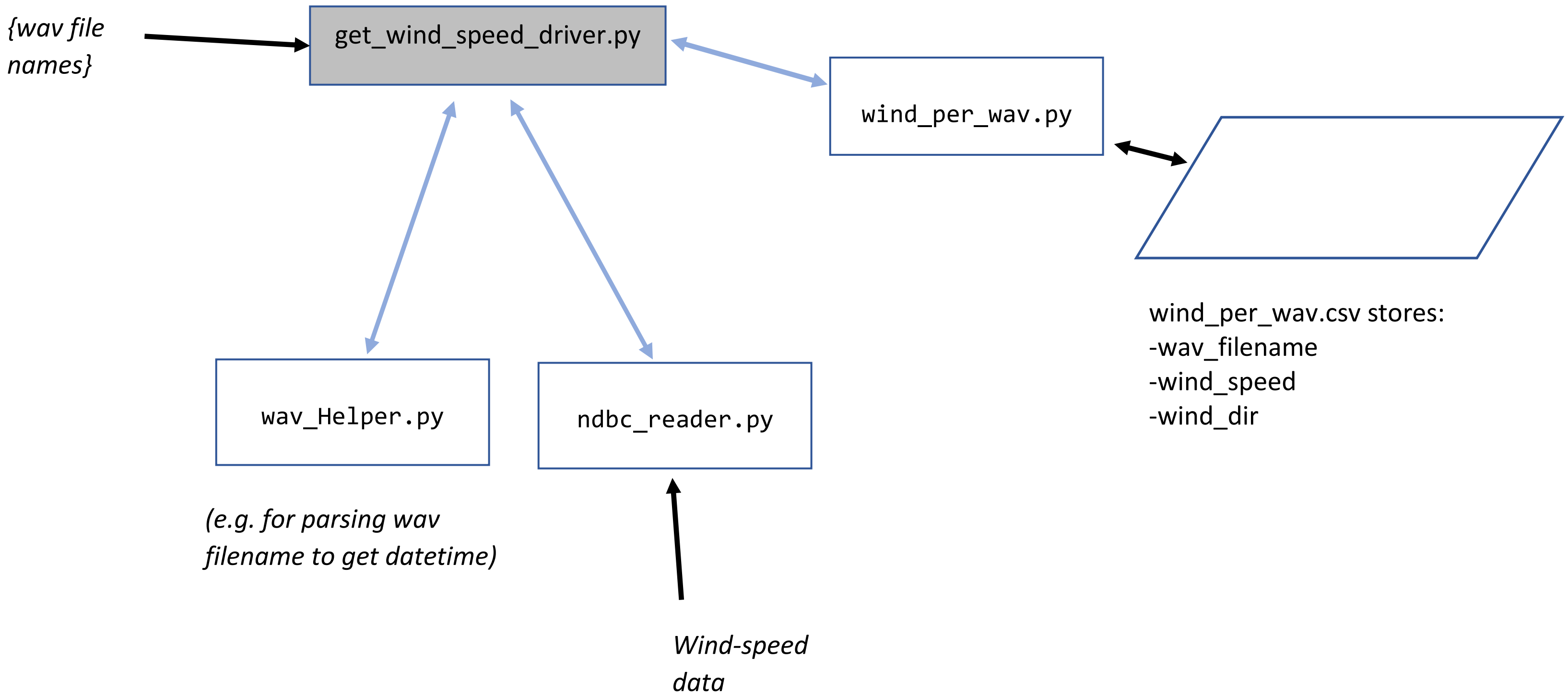

Figure 1(c): Flow diagram for "Step 2" of the spectral analysis code for Jeanette's Pier data. Each rectangle represents a python program. A blue arrow indicates a call from one program to another. A black arrow indicates data flow. The gray shading indicates a main program. The output csv file is drawn as a parallelogram.

Step 3: Power Spectral Density (and spectrum) per wind-speed/wind-dir bin

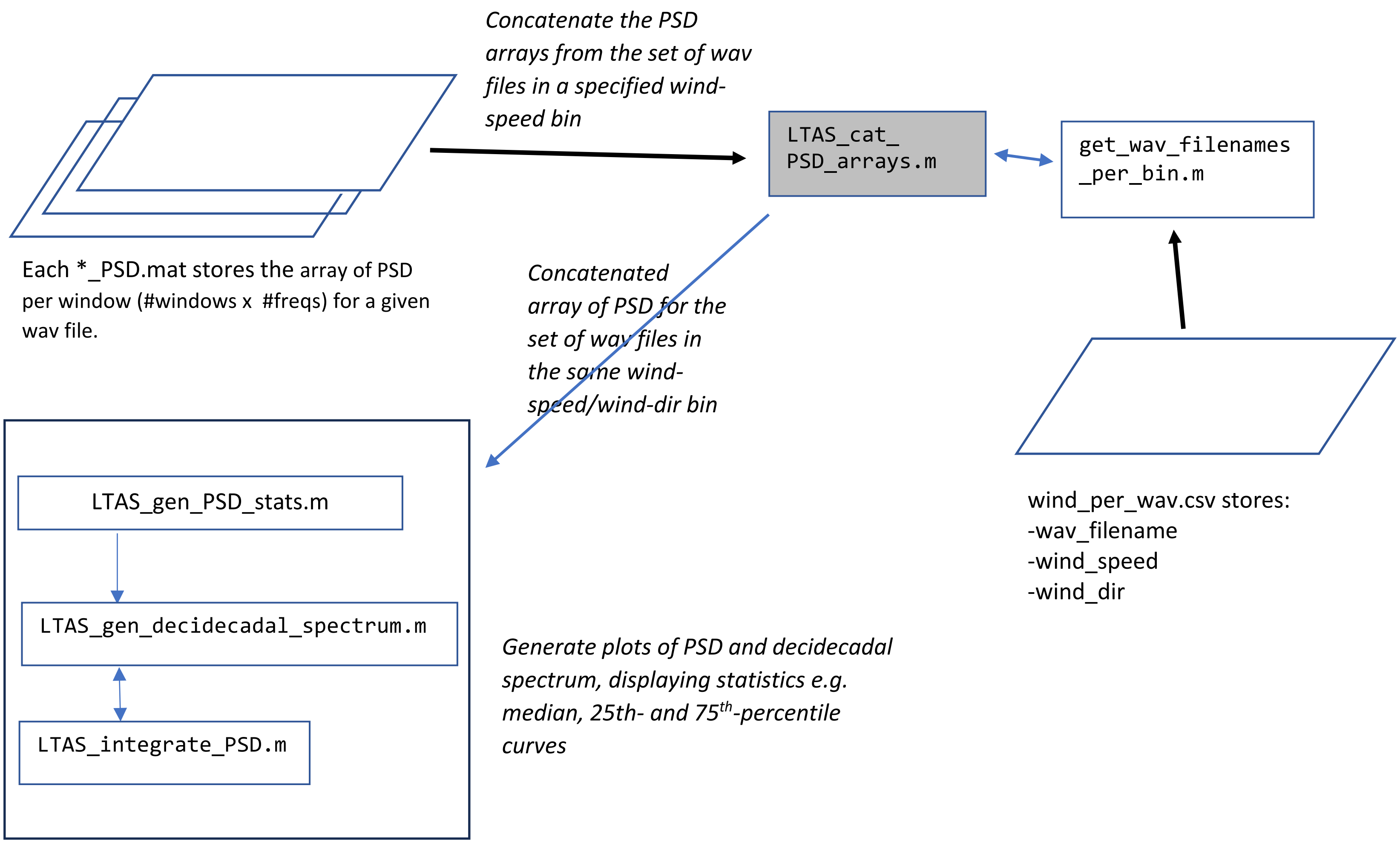

Figure 1(d): Flow diagram for “Step 3” of the spectral analysis code for Jeanette’s Pier data. Each rectangle represents a matlab function or script. A blue arrow indicates a call from one program to another. A black arrow indicates data flow. The gray shading indicates a main program. The mat file is drawn as a parallelogram.

Data Views

wav files

The wav files are downloaded from dropbox into the local file system, where each hydrophone deployment has a separate folder, containing one or more wav files.

Source(s) of wind-speed

For wind speed, we use data from NDBC site ORIN7 at https://www.ndbc.noaa.gov/station_page.php?station=orin7

For each year of interest, we follow the links to the historical data e.g. for 2021:

https://www.ndbc.noaa.gov/download_data.php?filename=orin7h2021.txt.gz&dir=data/historical/stdmet/

From there, we downloaded the gzip file,and gunzipped it.

The python function ndbc_reader.load_ndbc_file parses the space-delimited file, performs some quality control, converts from knots to meters per second, and converts to a height of 10 meters above mean sea-level (Johnson, 1999).

Source(s) of rain-rate

We do not have a good source of rain rate. Instead, we will manually review the notes for each deployment. If the notes indicate that there is rain, we will exclude the files for that deployment. (In other words, do not run steps 1 and 2 on those folders.)

wind_per_wav.csv

This csv file represents the data that is communicated from Step 2 to Step 3. It stores the following fields:

- wav_filename_sans_ext
- wind_speed (in meters per second, measured at 10 meters above mean sea-level)
- wind_dir (in degrees, measured clockwise from North)

Sample contents of wind_per_wav.csv:

```
wav_filename_sans_ext,wind_speed,wind_dir
SCW1984_20210421_132000,2.25129502121495,186.33333333333334
SCW1984_20210421_142000,6.1,160.1
```

PSD mat file

This mat file represents the data that is communicated from Step 1 to Step 3. There is a PSD mat file per wav file. It stores the following:

- PSD_per_window_cal is an array of dimension #windows x #freqs, storing the magnitude of the PSD in $\mu Pa^2/Hz$ (after applying the calibration).
- frequency_Hz is the associated frequency values, of dimension #freqs x 1

The file naming convention for the PSD mat file is to use the wav file name (minus the .wav extension) and add _PSD.mat. Therefore, when we group wav files into bins by wav filename, we know which PSD mat files to concatenate together.

All *_PSD.mat files are stored in the same directory in the file system. This makes it easier to concatenate them together.

The concatenated PSD array is not saved; it exists only in memory.

good segment mat file

This mat file is optional. It is used to identify a custom segment of the wav file to process. It stores the following

- start_sample
- end_sample

(By running the LTAS_Driver.m in preview_mode, one can see which segment is good.)

The file naming convention for this mat file is to use the wav file name (minus the .wav extension) and add _good_segment.mat. Currently, it must be manually generated.

Assumptions/issues regarding the data

The table below documents assumptions made regarding the data as well as potential issues.

Short Description	***Long Description***
wav sample rate	To generate spectral plots per wind-speed/wind-dir/rain-rate bin, the appropriate PSD arrays (of dimension #windows x #freqs) are concatenated. Therefore, the wav files must all have the same sample rate. (Hopefully, this issue would be identified in the data exploration/pre-processing step.)
wav file names	We assume wav file names are unique. This is relevant because we use the wav file name as part of the PSD mat file name, and we store all of the PSD mat files in the same folder. (Hopefully, this issue would be identified in the data exploration/pre-processing step.)
wind_per_wav.csv	How to avoid duplicate rows in this file (e.g. if we rerun step 2)? Ideally, want to update if the wav_filename already exists in the file; otherwise, insert. This is accomplished in get_wind_speed_driver.py one row at a time—if the insert fails, do the update.

Data quality checks

NDBC file for wind speed

In the file from NDBC, values for wind speed = 99 and wind direction = 999 indicate “no data”. Of course, these rows were excluded.

Wav files

Here, we list the quality checks performed on the wav files for this release of the code (i.e. for characterizing the baseline soundscape). In the future, when analyzing the sound from marine energy conversion (MEC) devices, we will need to add quality checks to remove windows having biological sounds.

Out of water

There may be sections of each wav file where the hydrophone was recording data before and/or after the device is deployed in the water. Hopefully, this issue would be identified in the data exploration/pre-processing step. Currently, the data exploration/pre-processing step does not automatically fix the issue. Instead, one must create a *_good_segment.dat mat file, where the * is replaced by the wav filename (without the extension). See “good segment mat file” above in the Data views section for the format.

Clipping

The LTAS program calculates the power spectral density for each 1-second window. The windows are 50% overlapping.

If a 1-second window contains samples having a value close to or at the wav file’s limit (a value of +/- 1), the window is skipped due to clipping.

Data gap/jump in mean value

The LTAS program calculates the power spectral density for each 1-second window. The windows are 50% overlapping.

If a 1-second window contains a jump in the mean value which is above a threshold, the window is skipped. Figure 2 shows an example.

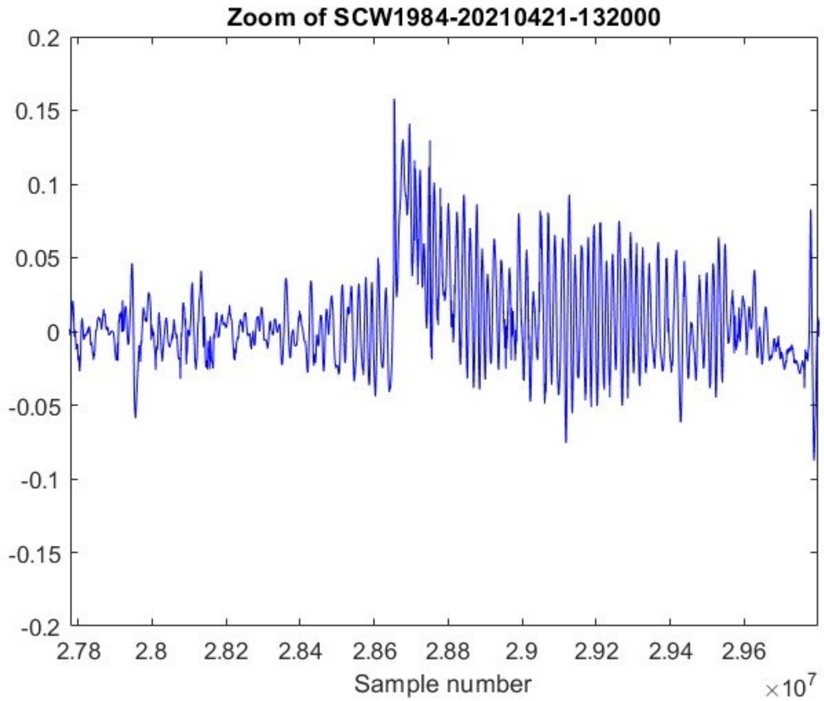

Figure 2: An example of a jump in the mean value in the recording.

Since the device does not record a timestamp for each sample, a data gap may be manifested as a sudden jump in the mean value. This could happen in the following scenario:

- The device does not record a timestamp for each sample; instead, we just *assume* the time series is continuous—that consecutive samples are separated by the sample interval.
- The hydrophone is recording at a high sample rate; however, it fails to store a buffer of data before the next buffer is ready to be stored. In this case, it may skip a buffer (or part of a buffer). The skipped buffer is a data gap.
- There is a low-frequency signal that is not filtered out. For example, the period of ocean waves may be several times the size of the 1-second window. In this case, a data gap could be manifested as a jump in the mean value.

When we define a jump in the mean, we need to tune the duration of the mean; we wouldn’t want to confuse a data gap with a brief biological “click” sounds e.g. from echolocation or from snapping shrimp noise.

Implementation views

Programming language

The current implementation uses Matlab for steps 1 and 3, python for step 2.

The software was developed on Matlab version 24.1.0.2578822 (R2024a) Update 2 and python version 3.13.2.

Code availability

The Matlab code (for steps 1 and 3) is available in github at: https://github.com/sblockhartzzero/CSI_JP_SpectralAnalysis_Matlab

The python code (for step 2) is available in github at: https://github.com/sblockhartzzero/CSI_JP_SpectralAnalysis_Python

Filesystem usage

The following table documents the files/folders you will need to specify when customizing the code to your environment.

File/Folder Description	***Comments***
wav_folder • in step 1's LTAS_Driver.m • in step 2's get_wind_speed_driver.py	• Folder storing a set of wav files • Currently, you need to run both step 1 and step 2 for each wav_folder.
wind_per_wav_fullpath • in steps 2's get_wind_speed_driver.py • in step 3's LTAS_cat_PSD_arrays.m	• Full path to the wind_per_wav.csv file • The folder should exist; if the file does not yet exist, step 2 will create it—as long as the folder exists.
PSD_matfile_folder • in step 1's LTAS_Driver.m • in step 3's LTAS_cat_PSD_arrays.m	• Folder storing the *_PSD.mat files (for all wav files)
csv_fullpath = csv_folder + csv_filename • in step 2's ndbc_reader.py	• Path to NDBC ORIN7 space-delimited file

Python environment

Starting with my base python (version 3.13.2), I performed the following steps, using conda:

- Created an env e.g. **conda create –name try_ndbc1**
- **conda activate try_ndbc1**
- **conda install** the following:
    - netcdf4
    - jupyter
    - xarray
    - matplotlib
    - numpy
    - windrose

Note that netcdf4 and xarray were needed only for a jupyter notebook.

How to run

Before running the code, perform the following prerequisite steps:

- Download the Jennette's Pier hydrophone wav files from dropbox onto your local filesystem
- Obtain the source data files for wind-speed. (See "Data views" section above.)
- Customize the code, editing the main programs, (LTAS_Driver.m, get_wind_speed_driver.py, LTAS_cat_PSD_arrays.m) as well as ndbc_reader.py to point to the folders on your file system. (See "Filesystem usage" above.)

For each step, run the appropriate main program, as detailed below.

How to run data exploration for wav files

- This data exploration is enabled by turning on the preview_mode in **LTAS_Driver.m**. Run the Matlab program **LTAS_Driver.m**, making sure to switch preview_mode to true.

How to run data exploration for wind speed (NDBC file(s))

How to run step 1

- For step 1, run the Matlab program **LTAS_Driver.m**, making sure to switch preview_mode to false.
- This program processes all the wav files in a folder. If you have a folder per hydrophone deployment (as we did), then you'll need to run this program once per folder.
- Remember to check for rain during a deployment. Do not run step 1 on the deployment's folder if there was rain during the deployment.
- In general, it's OK to rerun step 1 on a folder, as the *_PSD.mat files get overwritten.

How to run step 2

- For step 2, set up your python environment. (See "Python Environment" above.)
- Run **python get_wind_speed_driver.py**.
- Similar to step 1, this program processes all the wav files in a folder. If you have a folder per hydrophone deployment (as we did), then you'll need to run this program once per folder. Each time, the program appends to the wind_per_wav.csv file (performing either an inset or update, as needed).
- Remember to check for rain during a deployment. Do not run step 2 on the deployment's folder if there was rain during the deployment.
- It's OK to rerun step 2, as the wind_per_wav.csv file will be updated.

How to run step 3

- For step 3, run the Matlab program **LTAS_cat_PSD_arrays.m**.
- This program processes a single, specified wind-speed/wind-dir bin, generating plots. You'll need to run this program for each wind-speed bin, saving the plots for each run. Currently, it doesn't label the plots with information identifying the bin, so you should edit the plots manually.

Sample Plots

Data Exploration of NDBC file for wind speed, wind direction

The jupyter notebook probe_ndbc_wind.ipynb explores the wind-speed data in a specified NDBC data file, ultimately generating a windrose plot of the wind-speed, as in Figure 3 below.

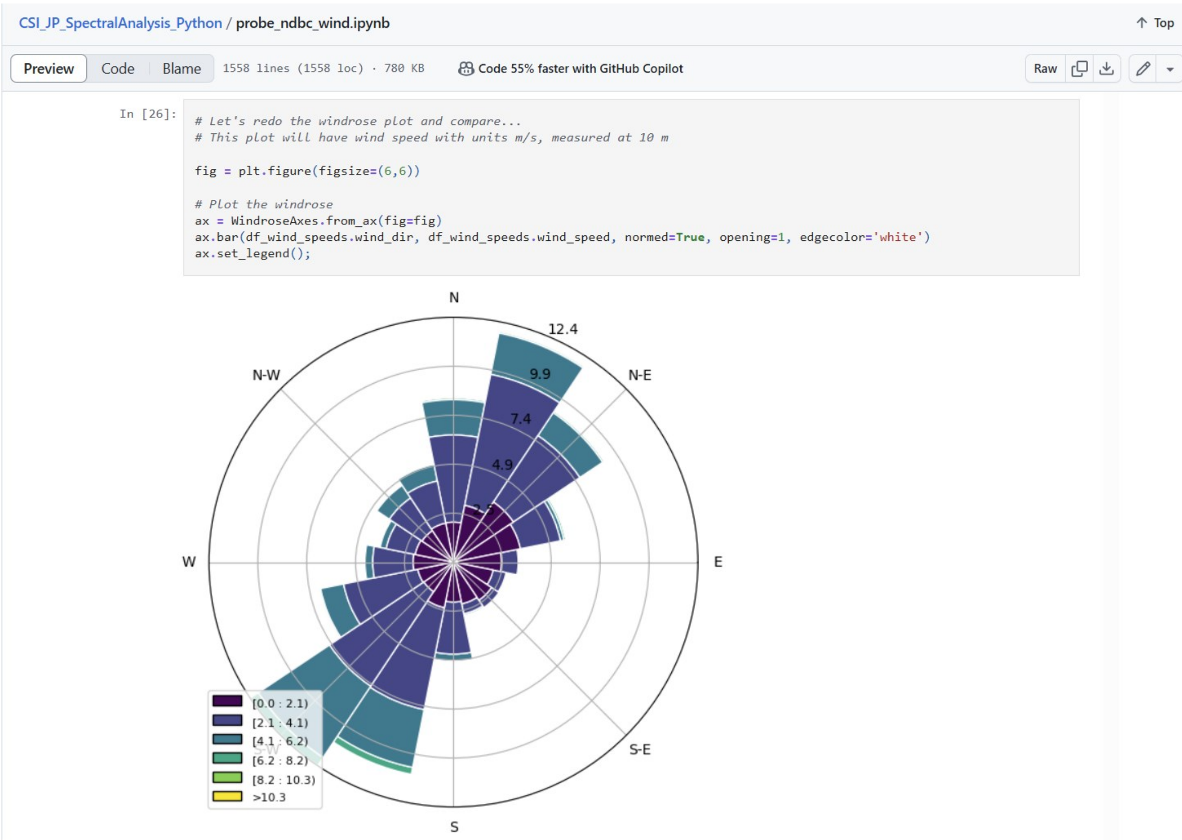

Figure 3: windrose plot from jupyter notebook probe_ndbc_wind.ipynb.

Step 1: Calculate Power Spectral Density (PSD) per window, per wav file

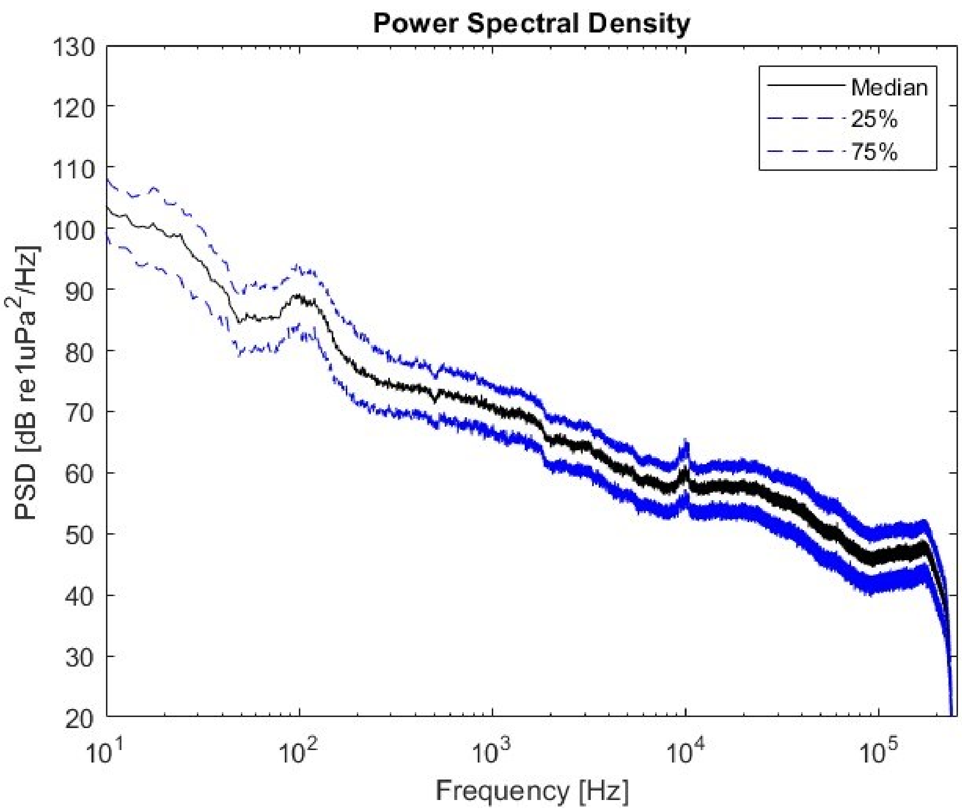

Figure 4: Sample plot of power spectral density for SCW1984_20210423_134500.wav.

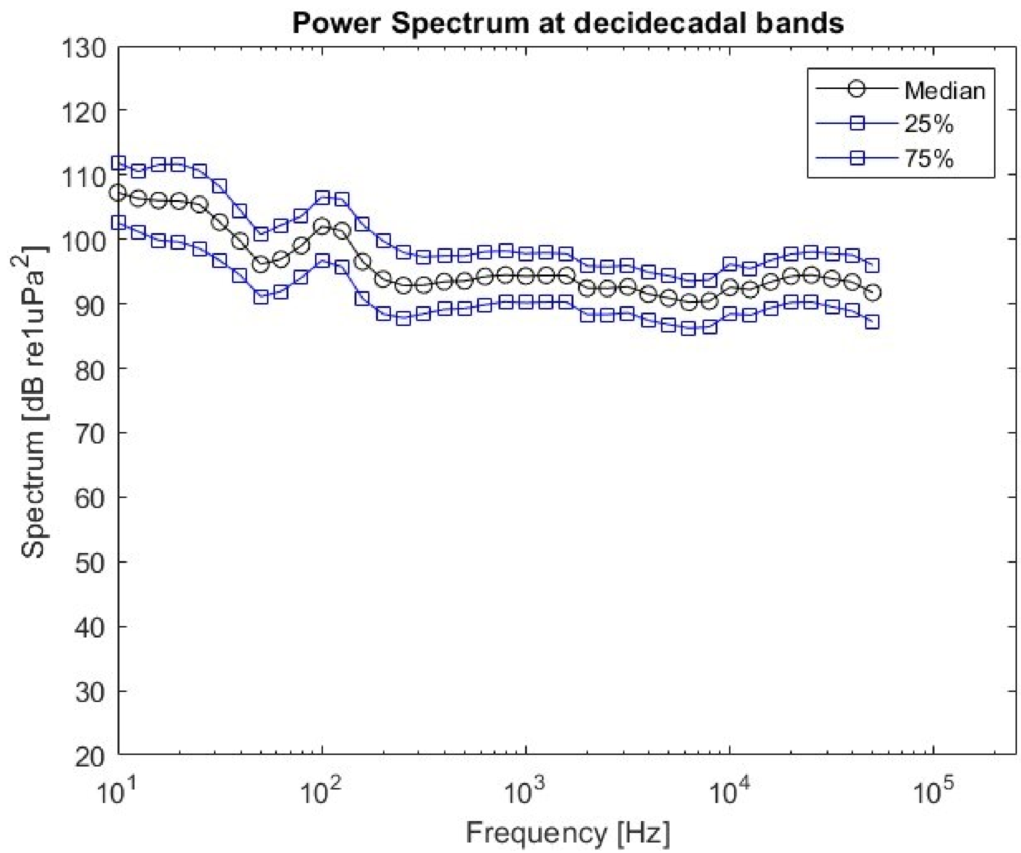

Figure 5: Sample plot of power spectrum over decidecadal bands for SCW1984_20210423_134500.wav.

Step 2: Get wind speed and direction per wav file

After running step 2 (and before running step 3), you can see how the wind-speed/wind-dir bins are populated by running the jupyter notebook view_wind_per_wav.ipynb. This will let you know which bins are actually populated before running step 3. This notebook provides a histogram, as in Figure 6 below. The bin edges are set to the bin edges of the wind-speed/wind-dir bins.

Figure 6:

Step 3: Power Spectral Density (and spectrum) per wind-speed/wind-dir bin

The plots from step 3 are similar to the plots from step 1. Whereas step 1 plots PSD and decidecadal spectra per wav file, step 3 generates these plots per wind-speed/wind-dir bin.

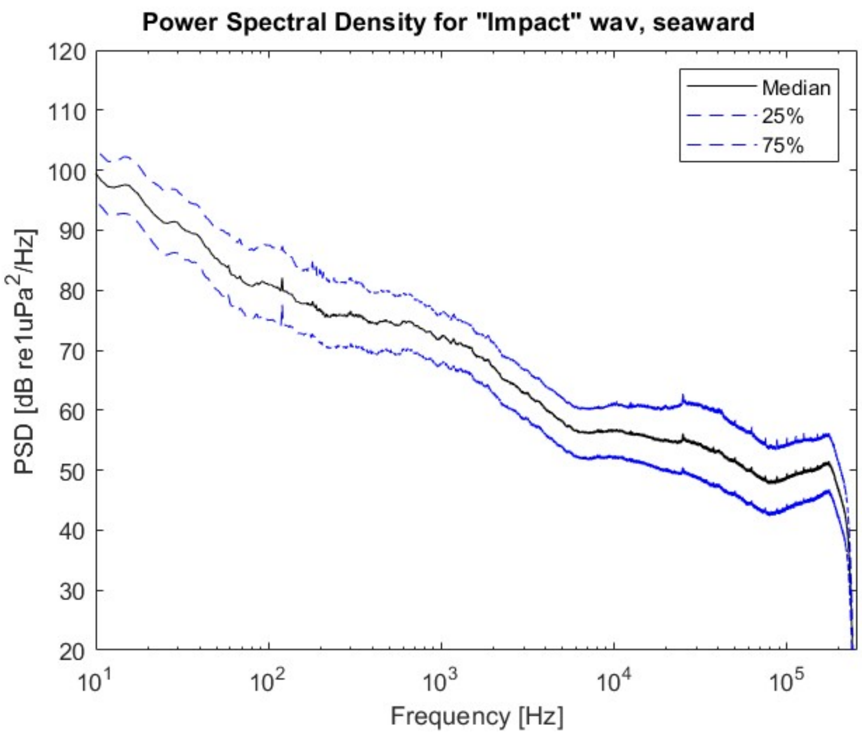

Figure 7: Plot of PSD generated in step 3, showing median (black) as well as 25th and 75th percentile curves for the “impact” wav files that were in the seaward (wind direction) bin. (Wind-speed bin is 2-4 m/s @10m above sea-level.)

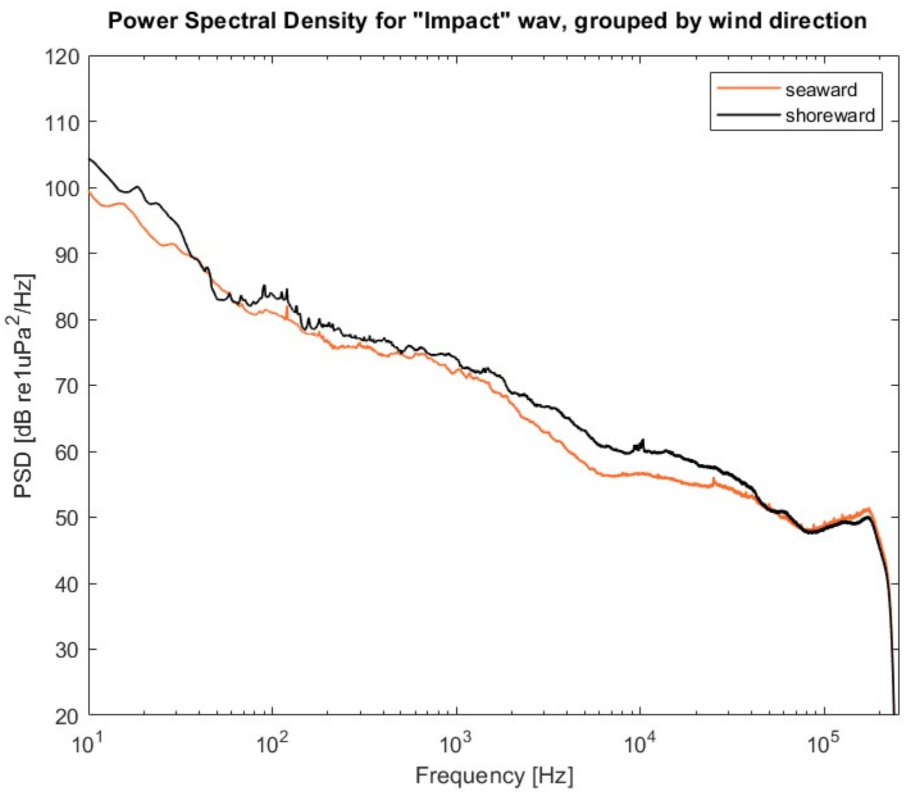

Figure 8: Plot comparing the median value of PSD for two bins—seaward and shoreward. This plot was manually crafted from the PSD plots for seaward and shoreward. Wind speeds were between 2-4 m/s for all wav files.

Calibration

Testing

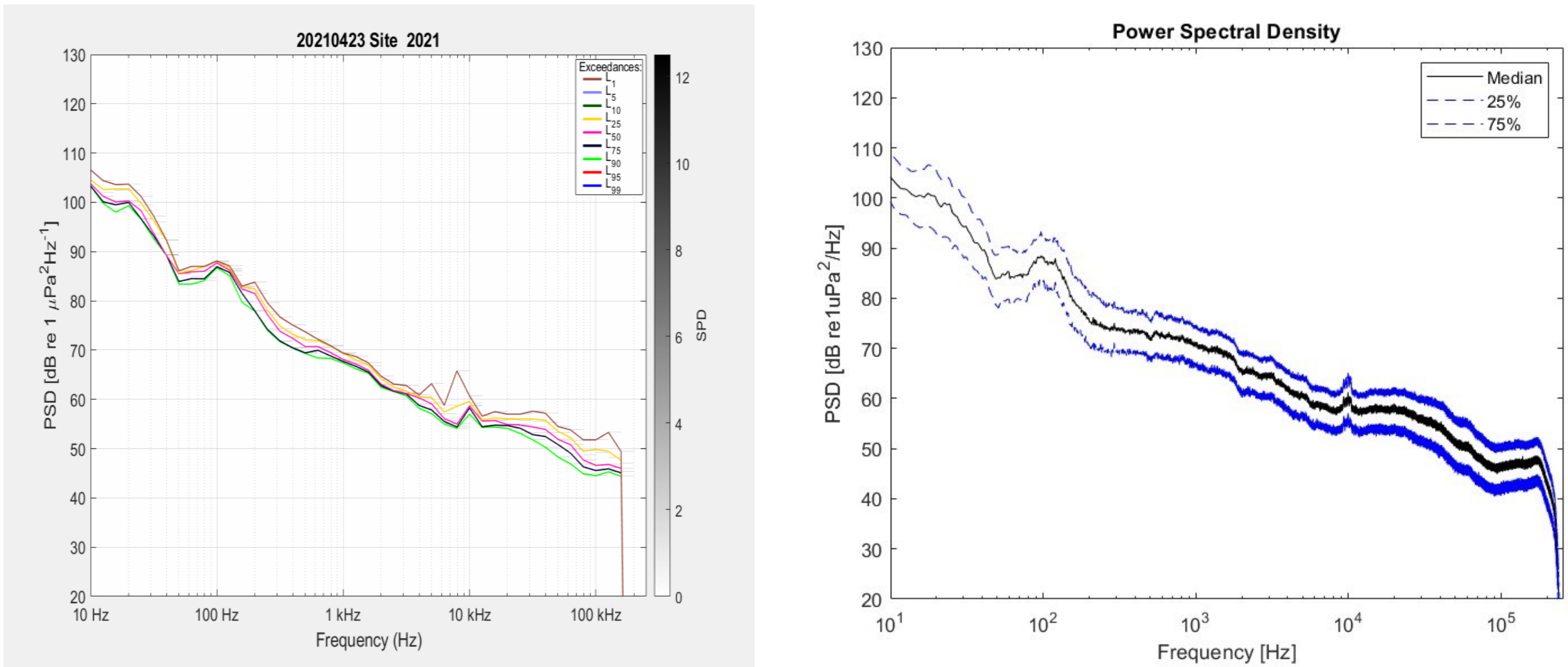

Figure 9: A comparison of power spectral density from MANTA (on the left) and from our Matlab-based LTAS code (on the right) for wav file SCW1984_20210423_134500.wav.

To test our Matlab-based LTAS code, we compared the PSD to one generated using MANTA (Miksis-Olds et al., 2021)--using the same wav file (SCW1984_20210423_134500.wav), same calibration data, and same peak voltage setting. Figure 9 shows that the comparison is good[3,4].

Future enhancements

Our initial focus is to characterize the baseline soundscape. In the future, this effort may be expanded to analyze sound emitted by marine energy conversion (MEC) devices.

Also, since the sample rate for the “background” wav files was too low, I just ran the software on the "impact" wav files from April of 2021. This is just a test. To get "background", I still need to do the following:

1) See if I can extract "background" from these "impact" recordings i.e. exclude periods when the MEC device is operating (as mentioned in previous email)

2) Exclude periods when there is biological "noise" e.g. whistles and echolocation clicks

3) Exclude periods when there is other anthropogenic noise (e.g. ships)

---

[3] When doing the comparison, we discovered that Matlab’s pwelch command returns PSD in units of power ***per radian***. So, we fixed this by modifying LTAS to convert to power ***per Hz*** to get good agreement with MANTA.

[4] The quality control checks did not find any issues with this wav file SCW1984_20210423_134500.wav; therefore, the LTAS code and MANTA processed the same set of 1-second windows.

References

IEC TS 62600-40, "Marine energy – Wave, tidal and other water current converters – Part 40: Acoustic characterization of marine energy converters", Edition 1.0 2019-06.

Johnson, H. K. (1999). Simple expressions for correcting wind speed data for elevation. *Coastal Engineering (Amsterdam)*, *36*(3), 263–269. https://doi.org/10.1016/S0378-3839(99)00016-2

Ma, B. B., & Nystuen, J. A. (2005). Passive Acoustic Detection and Measurement of Rainfall at Sea. *Journal of Atmospheric and Oceanic Technology*, *22*(8), 1225–1248. https://doi.org/10.1175/JTECH1773.1

Miksis-Olds, J. L., Dugan, P. J., Martin, S. B., Klinck, H., Mellinger, D. K., Mann, D. A., Ponirakis, D. W., & Boebel, O. (2021). Ocean Sound Analysis Software for Making Ambient Noise Trends Accessible (MANTA). *Frontiers in Marine Science*, *8*. https://doi.org/10.3389/fmars.2021.703650