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 discontinuity.
1.2	04/30/2025	“”	• Add wind-speed 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.

Introduction

At Jennette's Pier, the Coastal Studies Institute maintains a wave energy testing facility. At this site, hydrophones were deployed periodically from ??? to ??? to record ??? The hydrophone's recordings were stored as wav files.

In this project, we analyzed the hydrophone data from Jennette's Pier, providing plots of power spectral density (PSD) as well as the decidecadal power spectrum. The spectral analysis adheres to the IEC Technical Specification (IEC TS 62600-40, Edition 1.0 2019-06).

We grouped the recordings into bins based upon environmental factors that contribute to ambient noise, specifically wind-speed and rain-rate. The bins were defined as in Ma and Nystuen (2005). In this first release of the software, we excluded recordings during rain.

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 this project. Here, we document the design (process and data), implementation, and testing.

Process views

Overview

The 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 bins. Concatenate the PSD arrays for all mat files in a wind-speed bin. Then, generate the plots of PSD and decidecadal spectrum per wind-speed bin.

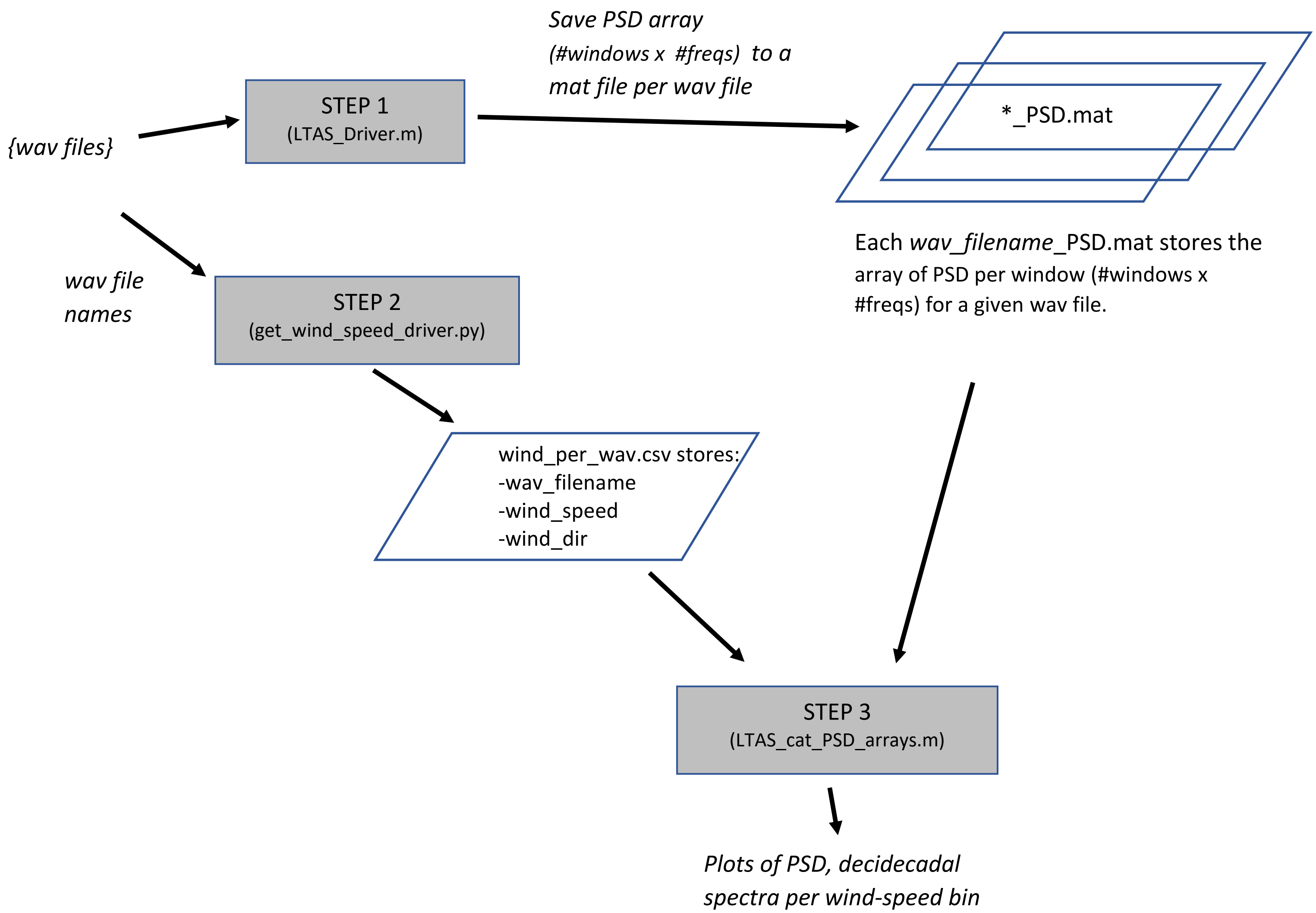

Figure 1(a): Overview flow diagram for Steps 1-3 of the spectral analysis code for Jeanette's Pier data. Data files are drawn as parallelograms.

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

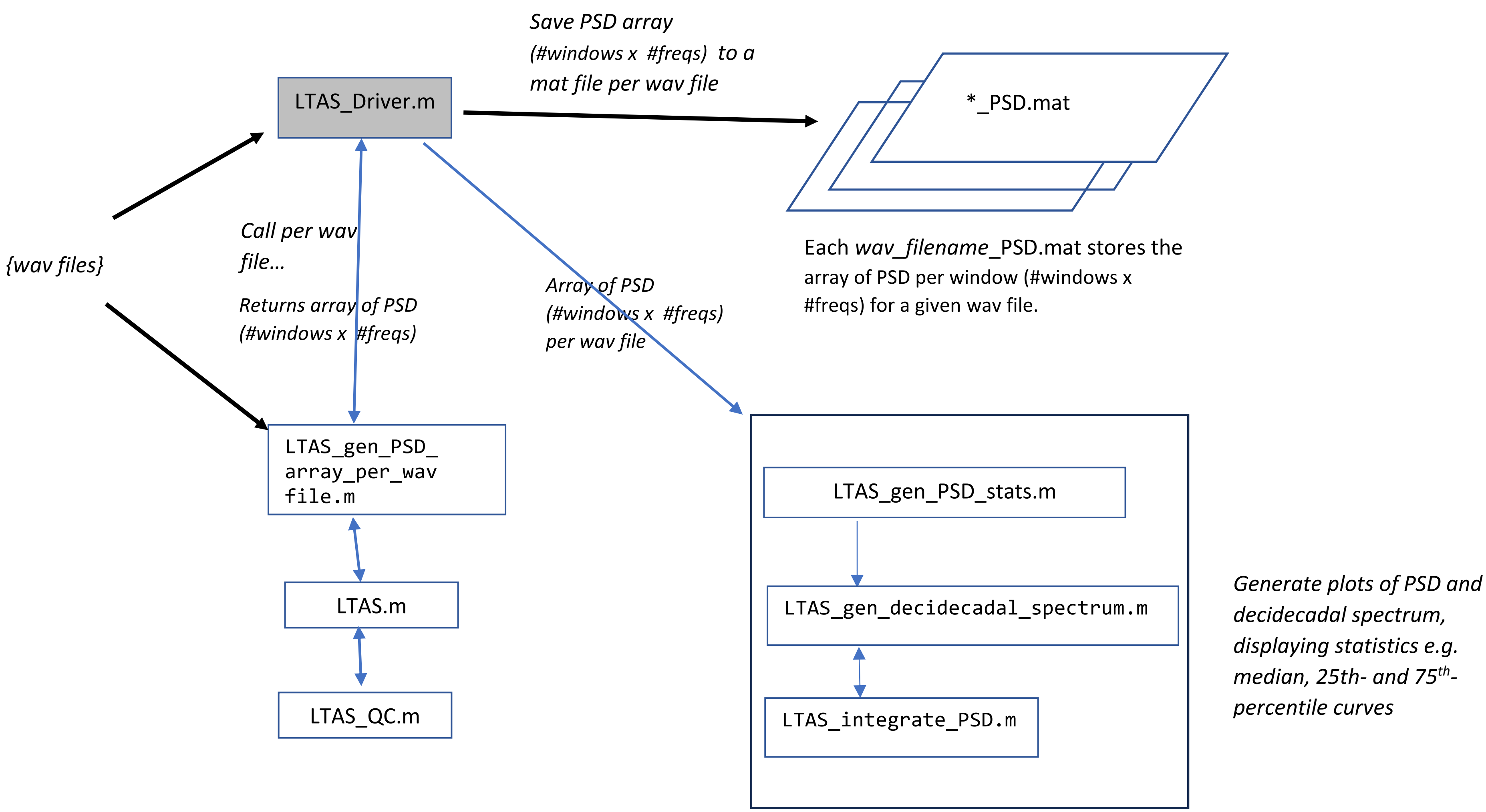

Figure 1(b): Flow diagram for “Step 1” of the spectral analysis code for Jeanette’s Pier data. Each rectangle represents a matlab function or script. A blue arrow indicates a call from one program to another. A black arrow indicates data flow. The gray shading indicates a main program. The output mat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

Step 2: Get wind speed and direction per wav file

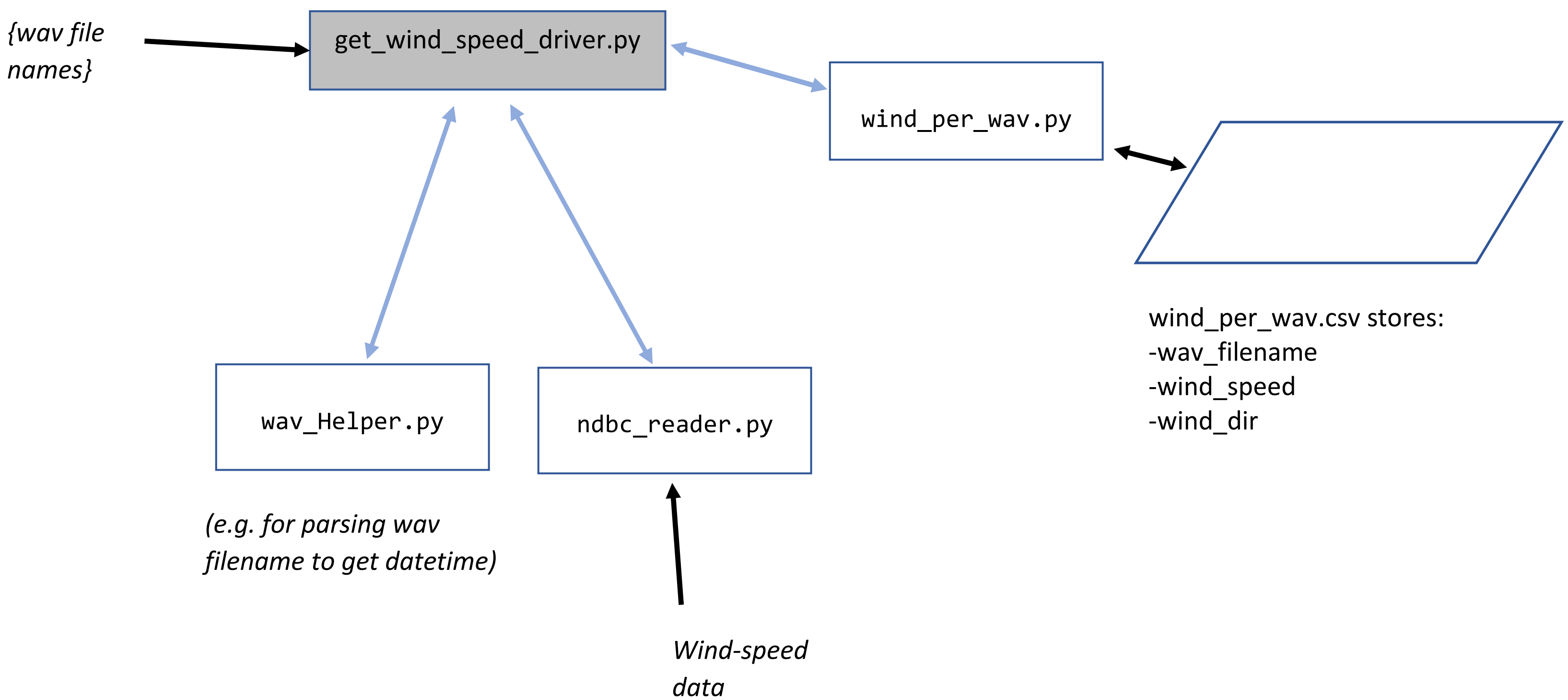

Figure 1(c): Flow diagram for "Step 2" of the spectral analysis code for Jeanette's Pier data. Each rectangle represents a python program. A blue arrow indicates a call from one program to another. A black arrow indicates data flow. The gray shading indicates a main program. The output csv file is drawn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

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

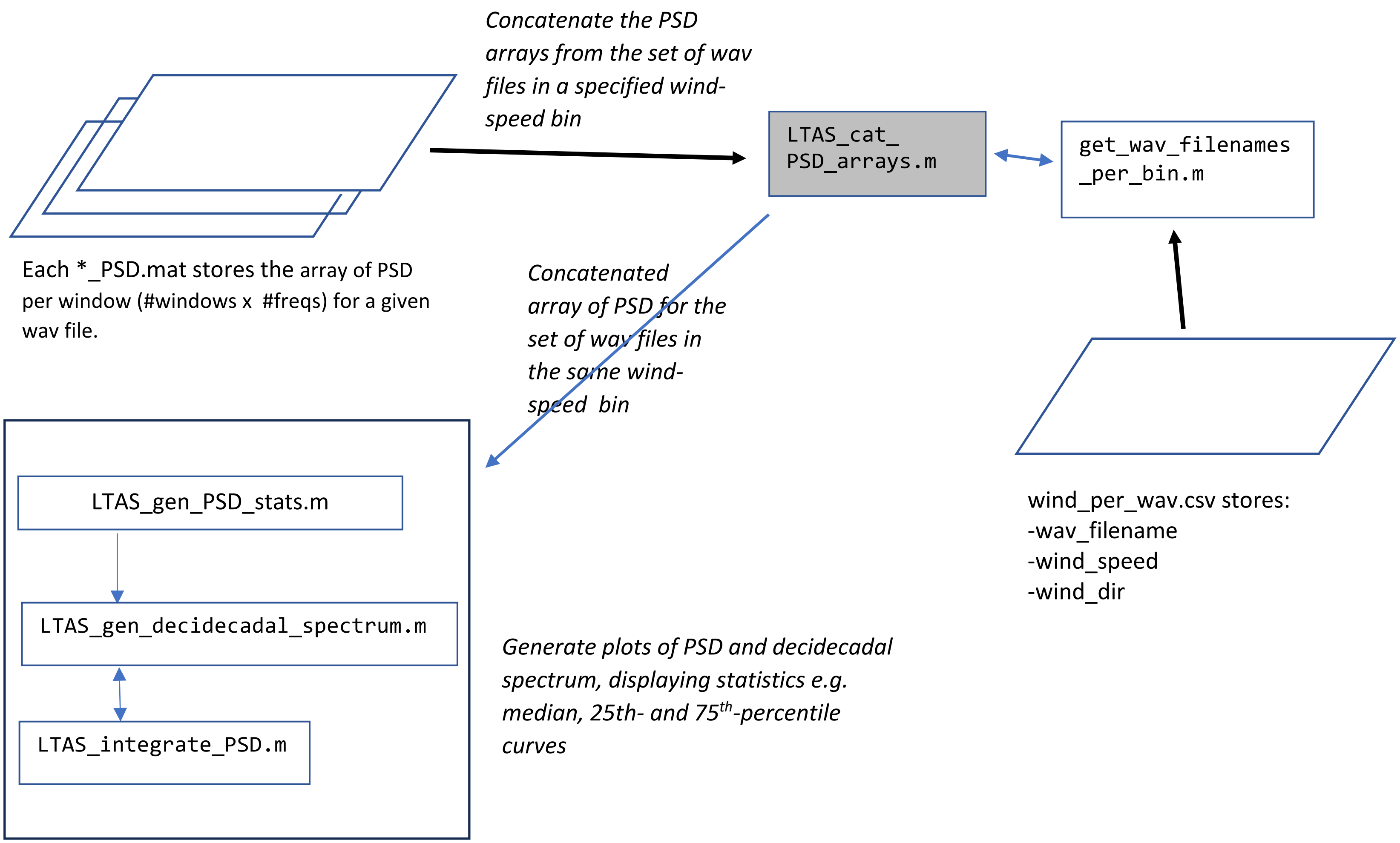

Figure 1(d): Flow diagram for “Step 3” of the spectral analysis code for Jeanette’s Pier data. Each rectangle represents a matlab function or script. A blue arrow indicates a call from one program to another. A black arrow indicates data flow. The gray shading indicates a main program. The mat file is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

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/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. If this is not the case, the software generates a warning and skips the wav file.
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.
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

Discontinuity

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

However, if a 1-second window contains a discontinuity which is above a threshold, the window is skipped. Figure 2 shows an example of a discontinuity.

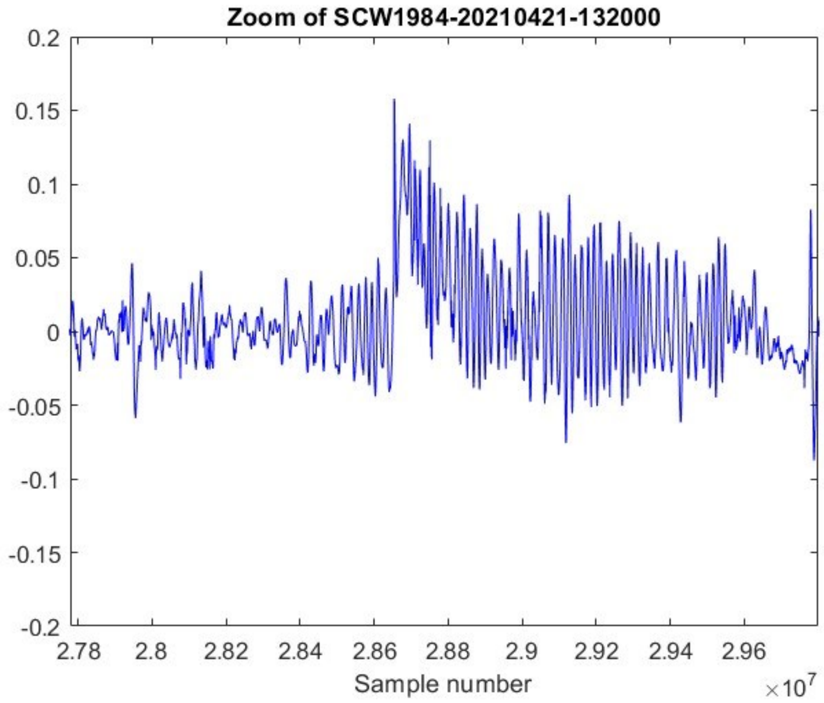

Figure 2: An example of a discontinuity in the recording.

A discontinuity in the time series may exist in the following scenario:

- 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 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.

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
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

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:

How to run step 1

- For step 1, run the Matlab program **LTAS_Driver.m**.
- 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 range, generating plots. You'll need to run this program for each wind-speed bin, saving the plots for each run.

Sample Plots

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

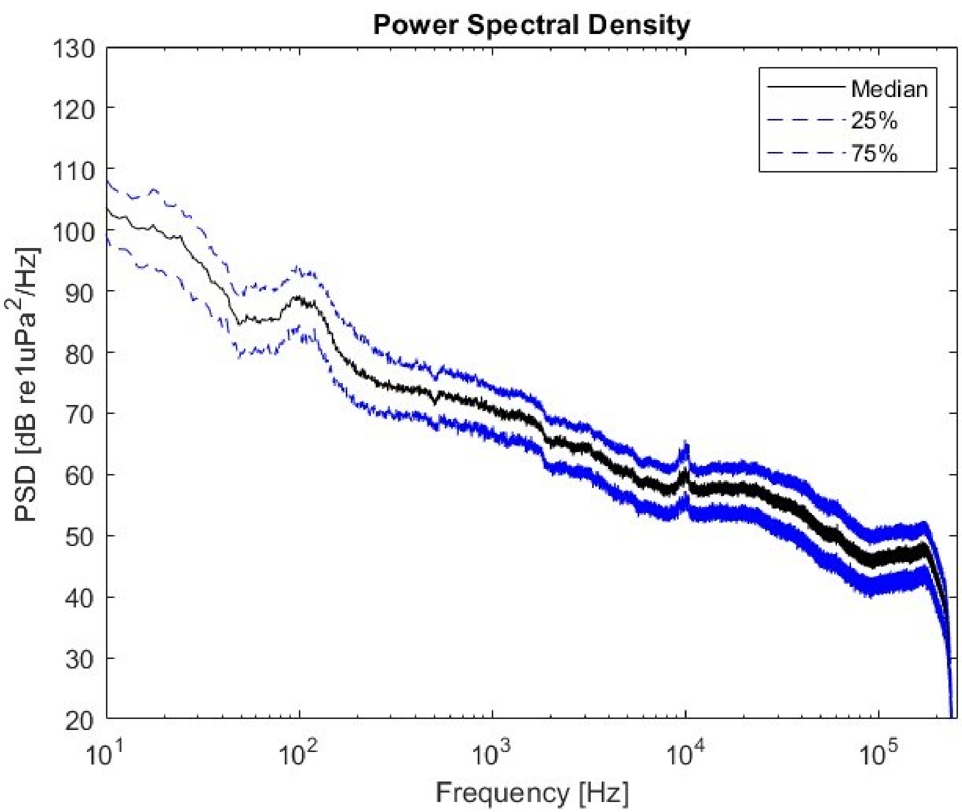

Figure 3a: Sample plot of power spectral density for a specified wav file.

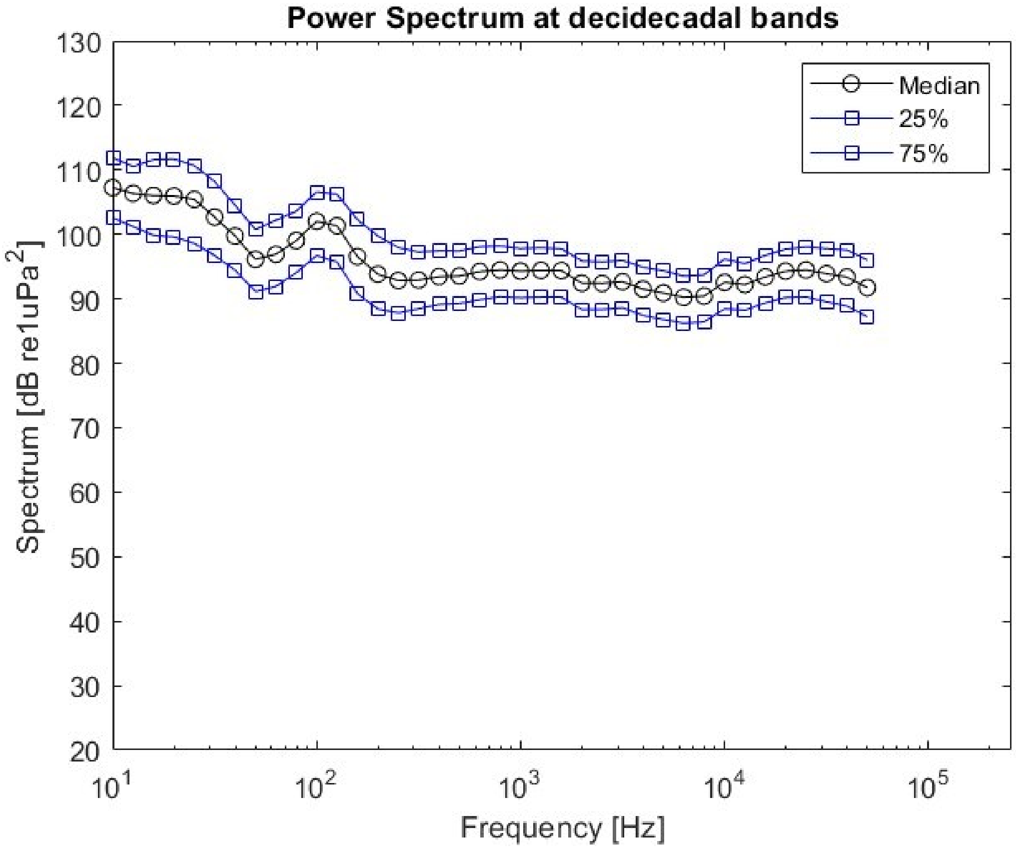

Figure 3b: Sample plot of power spectrum over decidecadal bands for a specified wav file.

Step 2: Get wind speed and direction per wav file

Before running step 3, you can see how the wind speed bins are populated by running the jupyter notebook view_wind_per_wav.ipynb. This notebook provides a histogram, as in Figure 4 below. The bin edges are set to the bin edges of the wind speed bins.

Figure 4:

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

Calibration

Testing

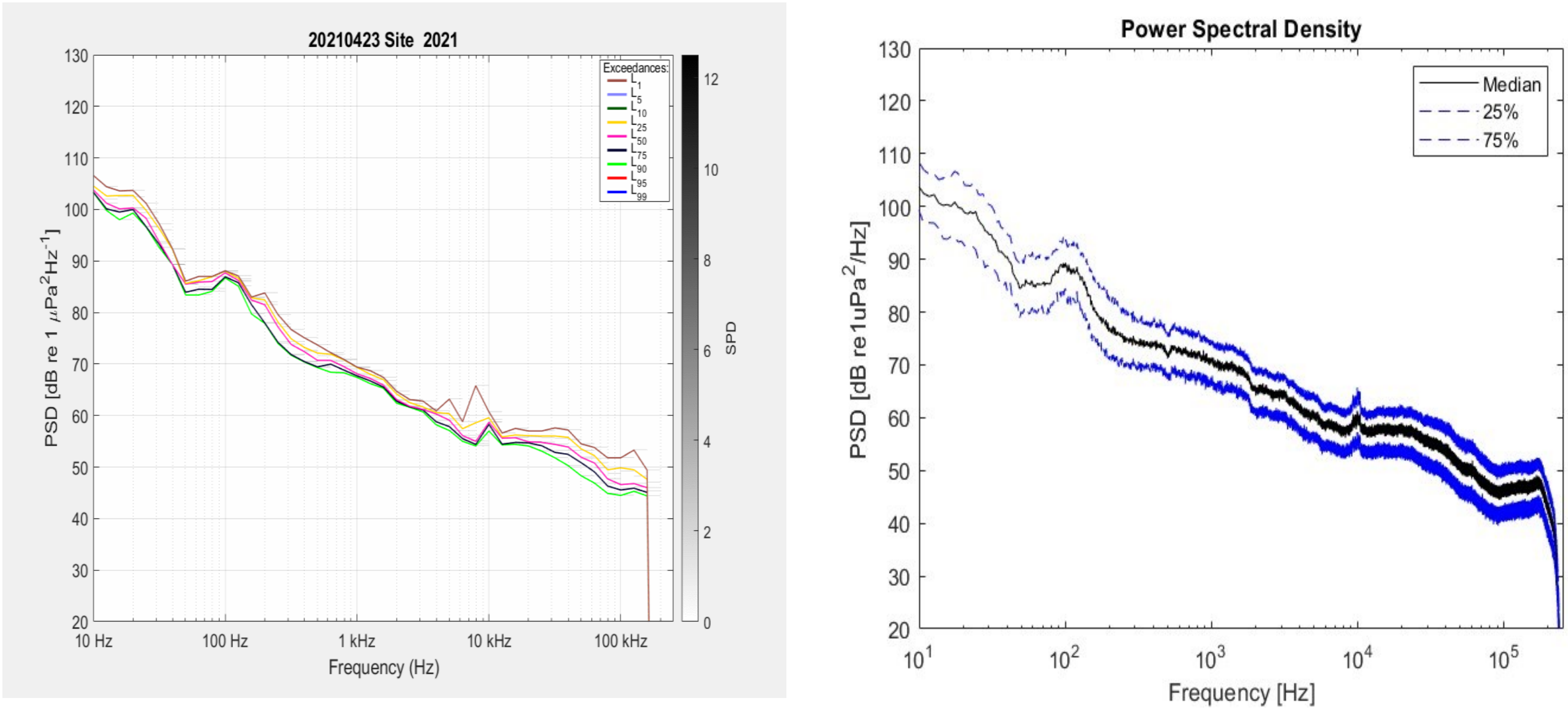

Figure 5: A comparison of power spectral density from MANTA (on the left) and from our Matlab-based LTAS code (on the right) for wav file SCW1984_20210423_134500.wav.

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 5 shows that the comparison is good. (In doing so, 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.)

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