Progress Report II: The Coupling Function Calculator LIGO-T1600387

Julia Kruk [1]

Abstract— **The seventh week of the SURF program concludes with the expanded coupling code entering it's final stages of debugging. Many additions to the code were made, including functions that calculate coherence, improve DARM calibration, allow for easier handling of important variables in the configuration file, plot the super-imposed spectra along with the coupling functions, etc. Many new options were also added to increase the program's flexibility.**

I. INTRODUCTION

Since the submission of the first report, the coupling function program has nearly doubled in size and it's capabilities have expanded significantly. By the fourth week, the code was capable of taking a single injection time and a single quiet time for DARM data, and either one specific channel, or a list of channels specified in a text file. Now it has the capacity to accept csv channel files that list channel names and their unique injection times. It can plot multiple channel data on the same figure, as well as recognize when different injection times are provided for different frequency bands of the same sensor channel. Calibration coefficients are extracted from the PEM.ligo.org database and used in the calibration of sensor amplitude density spectra. Therefore the calibration information will only need to be update in a single place, increasing the program's user friendliness. This code is designed with two specific objectives in mind. It allows the user to run bulk amounts of data to determine which injection times produces the best coupling functions. Then it is capable of taking specific times for each individual channel and producing coupling functions either on individual plot figures, or many on the same figure for the purpose of comparison or presentation. When we begin making PEM injections before O2, this code will significantly expedite the process of attaining good coupling functions that will be used to better understand the effect of ambient noise on various locations along the interferometer and the process of vetting gravitational waves.

II. ADDED FEATURES

A. Functions

The paragraphs below give a quick summary of the program's new capabilities made possible by the addition on new functions. Each of them can be prompted by a specific set of variables in the configuration file and/or through the options parser. The only function that is completely automatic is Spectrum Calibration:

- Spectrum Calibration: This function extracts calibration coefficients from a text file that holds all the data in the PEM.ligo.org database. There will be a smaller code written that will automatically overwrite the text file whenever the PEM.ligo.org database is updated. If the online database does not report a coefficient for a particular channel, then the program automatically sets the coefficient equal to 1.
- DARM Calibration: This function will calibrate DARM data one of two ways. Either it will use the strain calibrated channel and multiply the strain at each bin by the effective interferometer length (4000 m), or it will use the DELTAL–EXTERNAL channel and apply the same calibration code that is used for the monitors in the control room to get displacement. The desired calibration method must be specified in the configuration file.
- Coherence: Upon specification in the configuration file, the coherence is calculated bin by bin for every sensor channel that is run through the program, and is reported in the csv output. There is also an option that will prompt the program to output coherence plots for each channel. These options are featured in the configuration file as variables that accepts 'on' or 'off' inputs.
- Time Specificity: The program is now able to recognize when the text file with the list of channel names includes specific injection times beside each channel. This function is intended to save the user a lot of time, so that instead of running the program again with every different injection time, everything can be run at once. Also helpful when using the Multi-plot function for the purpose of comparing different channel data.
- Multi-Plotting: This function can be called upon when multiple channels need to be plotted on the same figure, or when a channel is meant to be decomposed into multiple frequency bands and plotted in different colors to survey possible up-conversion or down-conversions. This output can be requested via the output parser, but the program also recognizes when it needs to decompose channel data into frequency bands with their own injection times without direct specification. Like the Data Export function that plots coupling functions for channels individually, Multi-Plotting has been engineered to accept variations of input. The user may provide multiple channels and the same injection and

*This work was supported by the 2016 Caltech SURF program and the LIGO Collaboration.

[1]H. Kwakernaak is a physics student at Cornell University, Ithaca NY 14850.

quiet time for all, input a list of channel names with their individual injection times, or the user can provide one channel name and a list of times in the configuration file corresponding to a list of band names.
- Spectrum Plotting: This is a feature that was built into the two existing plotting functions discussed above. If the input for the variable 'spectrum–plotting' is 'on', then the program will include plots of the super-imposed DARM and sensor spectra during injection times. Code is currently being written to plot the spectrum of channel data that had unique injection times for various frequency bands. This function has been made optional for the sake of run time; for if all the capabilities of this code were rendered active at the same time, the program will take approximately 15 minutes to run.

B. Added Options

A few of the options created recently are briefly described below. These are minor additions that either improve outputted plots or impact the way that data is packaged and exported:

- DARM 10: This gives the user the option to plot the 'DARM divided by 10' line on the outputted plots as a reference. It would allow the user to better understand whether a particular injection is producing significant enough coupling.
- Upper Limits: Typing '-u' in the options parser, or agreeing to this in interactive mode, will produce plots that do not show upper limit coupling factors.
- Directory: Either through the options parser or interactive mode, a directory name could be provided which the program can create if the name is unique, or direct the produced figures and data to once all is complete.
- Multi-plotting limit: This variable in the configuration file controls how much channel data can be plotted on a single figure at one time.
- Colors: This is a list of colors that will be used in the plotting function, in the order they are provided in the configuration file.

III. Interactive Mode

User-friendliness is very important in this project. As this code grows larger and more complex, it is very important that it is made easily usable by those who will need to operate it in the coming months, and the student that may want to use or build upon it next summer. Therefore, the configuration file includes many comments that explain what inputs should look like, and anything that could be entered via the options parser, can also be specified in interactive mode. By type '-i' after the script name when running the program, all information will be attained by the code through a user interface. This interactive feature clearly explains what the correct inputs are, and includes easily understandable error messages that will direct the user on fixing an error. The interactive mode, options parser and configuration files are all intended to make this program easy to use and keep it a black-box for as long as possible.

IV. Goals

The debugging procedures that are currently being conducted show that the plots generated by this code closely match those that were made by Robert Schofield and Anamaria Effler last September. The slight deviations that exist are being closely investigated. The Spectrum plotting feature is the last thing that needs to be polished before this code can be declared finished. Once this is done, two much smaller codes will be written. The first will be designed to overwrite the sensor calibration text file containing all the calibration coefficients whenever the PEM.ligo.org database is updated. The second, which has already been drafted, will be able to take the csv files produced by the coupling function calculator, and create the lowest coupling function for a channel out of all the data it's provided. After these codes will be completed over the next two weeks, the final week of the program will be devoted to composing extensive documentation on these programs.