ENG 6 Solar Survey Individual Report

The function I played within the group was to construct the graphic user interface that is enables the user to collect sunlight. The other role I had was assisting in the debugging of the other interface used to interpolate and plot the solar data.

The interface for collecting the sunlight includes functions that are not our own work. These files that contain the functions are the a*rduino.m*, *ginput.m,* and *getAvailableComPort.m* . The ginput2 function is what allows one to zoom in, zoom out and select their location at the same time. Originally, the function is created by Carlos Adrian Vargas Aguilera as indicated by this link: https://www.mathworks.com/matlabcentral/fileexchange/20645-ginput2-m-v3-1-nov-2009. The reason why ginput2 is useful is because while using the standard ginput, the user cannot zoom in or out without first inputting their coordinates. The arduino function is taken from Lab 7 of the Engineering 6 Labs. The main use of this function deals with the connection between the computer and the Arduino module. Lastly, the getAvailableComPort function allows the user to find out which serial port their Arduino is connected to more easily than the instructions that Lab 7 provide. This function is written by Daniel Lavoie as indicated by the link: http://www.mathworks.com/matlabcentral/fileexchange/9251-get-available-com-port.

There was no major problems that arose during the debugging process of the interface used to collect the solar data was the issue of storing the data for multiple locations. The main idea is to set the class *trial,* that is located within the file called *trial.m,* equal to an object called *TRIAL. TRIAL* contains the properties: location_coordinates (the coordinates of the location), morning_open (the Open Circuit Sweep), morning_load (the Load Sweep), noon_open (the noon version of Open Circuit Sweep), noon_load, late_afternoon_open, late_afternoon_load, date and location_name(the approximate location name). The date, location_name are string inputs and the data sweep data are stored accordingly depending on what time of day the user indicated. The time of day specification is done with a group of radio buttons labeled *Morning, Noon, and Late Afternoon*. There is only one TRIAL object and it is a dynamic variable meaning that it just gets rewritten as one uses the program. Specific instances of TRIAL are stored into a cell array called ALL_TRIALS. The user specifies where to store the specific instance by typing the number in Trial Number box and then pressing Store Trial push button. The only errors that came up had to do with misplaced code within the wrong code blocks. An example would be that I placed the assignin function under the Store Trial push button code block. This lead to the variable ALL_TRIALS being called incorrectly by the Plot Interface GUI. The GUI could not call it correctly because it stored each instance of TRIAL as a 1x1 structure when it should have been stored as “1x1 trial”.

I was also involved in the debugging and the development of the Plot Interface GUI which interpolates and plots the contour plots for the user data. The major problems with this devlopment process was getting the data to work with the function TriScatteredInterp. The main reason for this issue was due to the data not being scaled properly. My main contribution to this part was locating syntax errors and constructing the algorithm to plot the Max Power of the raw data and the Average Voltage of the raw data. In conclusion, my role dealt majorly with debugging and GUI development.