Patch Antenna Rx

EE 4490 SPECIAL TOPICS INTRO TO ENGINEERING DESIGN TOOLS FINAL PROJECT.

Caleb Winfrey
KSU ID: | 07/28/2021

I. CST Simulation of Patch Antenna

Based on SID, the calculation for frequency to use:

$$86 \rightarrow 8.6 \rightarrow 0.86 \rightarrow 2.4 - 0.86 = 1.54GHz$$

Which creates the table:

Specification Table:	
SID #:	#####
Feed:	Inset
Frequency:	1.54GHz

Screen Captures of the CST simulation results:

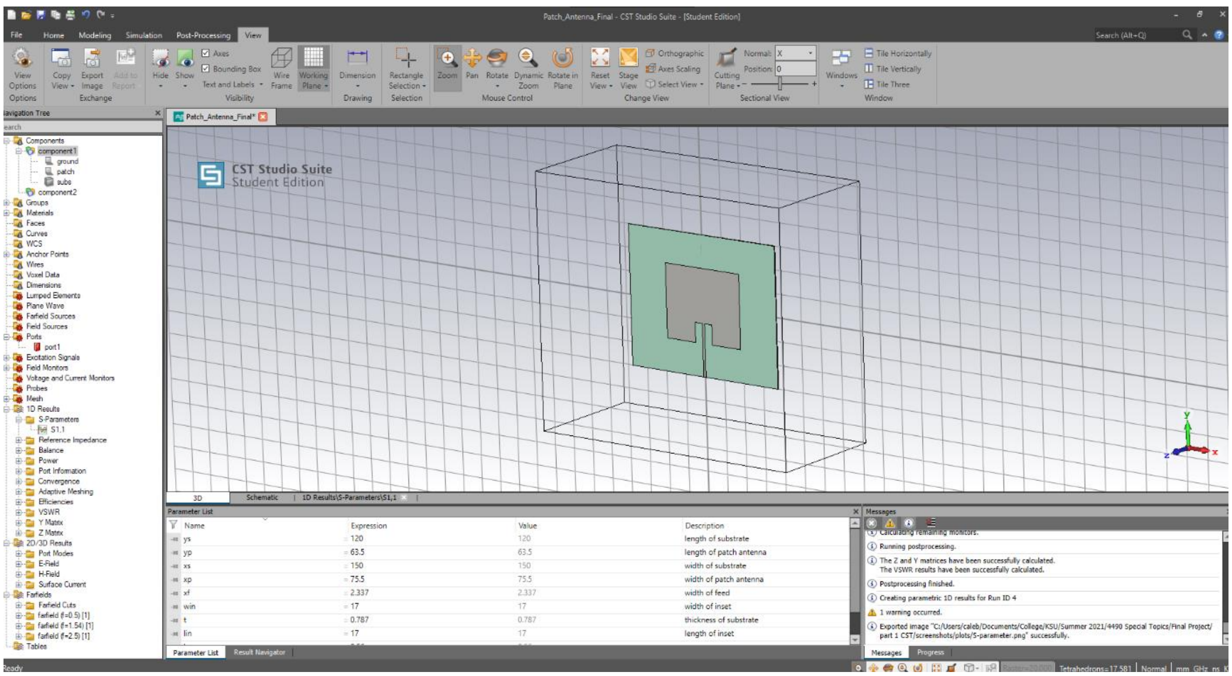

Figure 1. Finished Patch antenna with workspace shown.

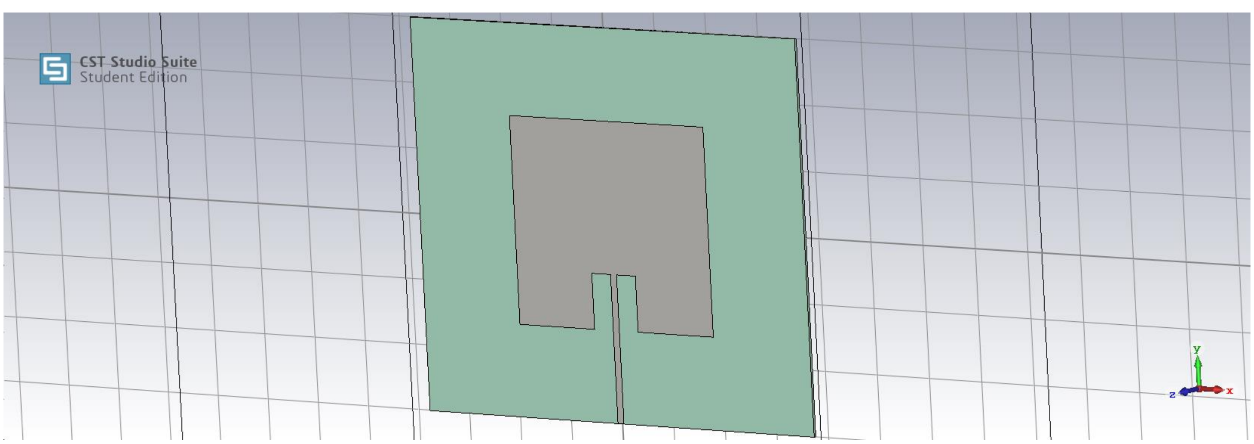

Figure 2. Finished patch antenna with inset feed, no workspace.

For the finished patch antenna above the final parameters are:

Final Parameters		
Parameter	**Value**	**Description**
xs	150	width of substrate
ys	120	length of substrate
t	0.787	thickness of substrate
xp	75.5	width of patch antenna
yp	63.5	length of patch antenna
xf	2.337	width of feed
k	8.56	port extension coefficient
win	17	width of inset
lin	17	length of inset

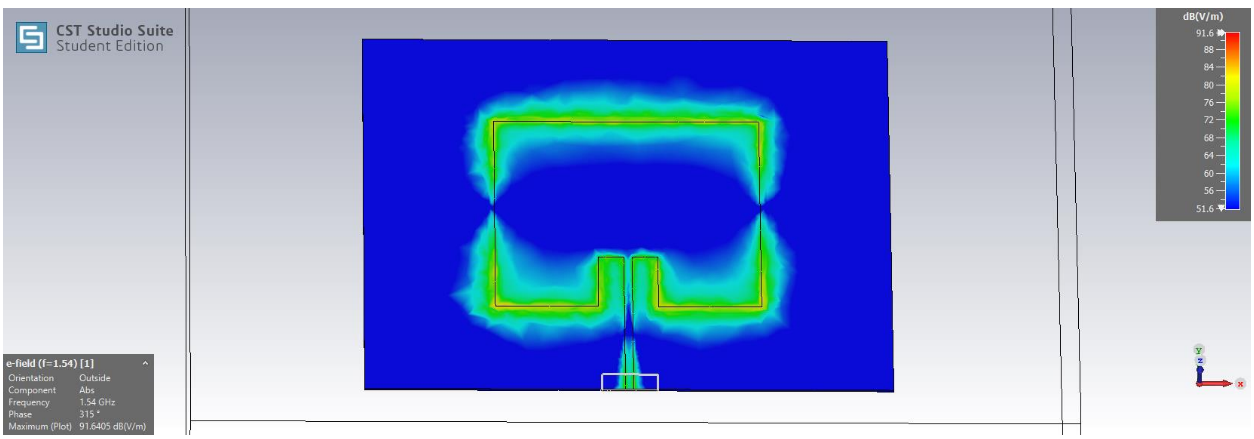

Figure 3. E-field plot (contour) of patch antenna at peak output, no standing waves, minimal reflections.

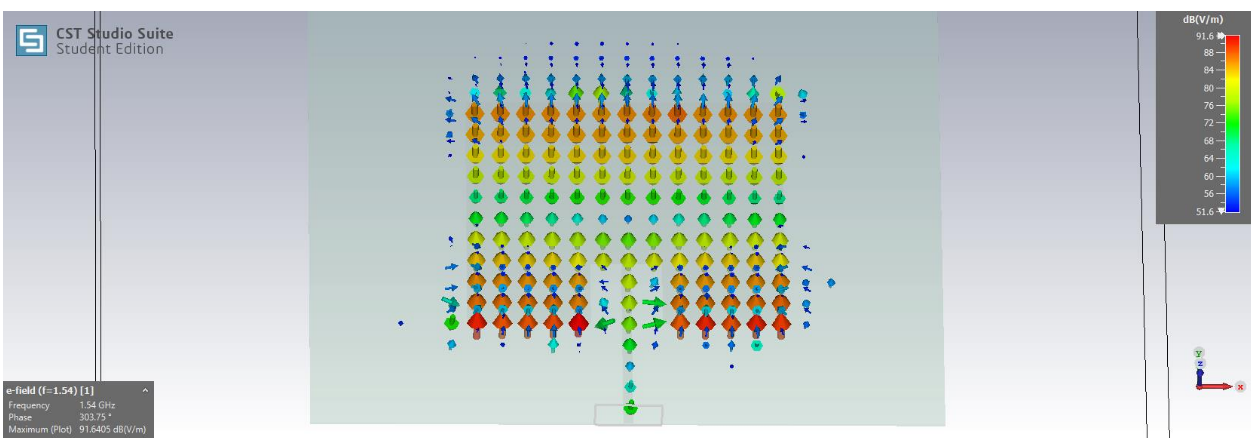

Figure 4. E-field plot (arrow) of patch antenna at peak output no standing waves, minimal reflections.

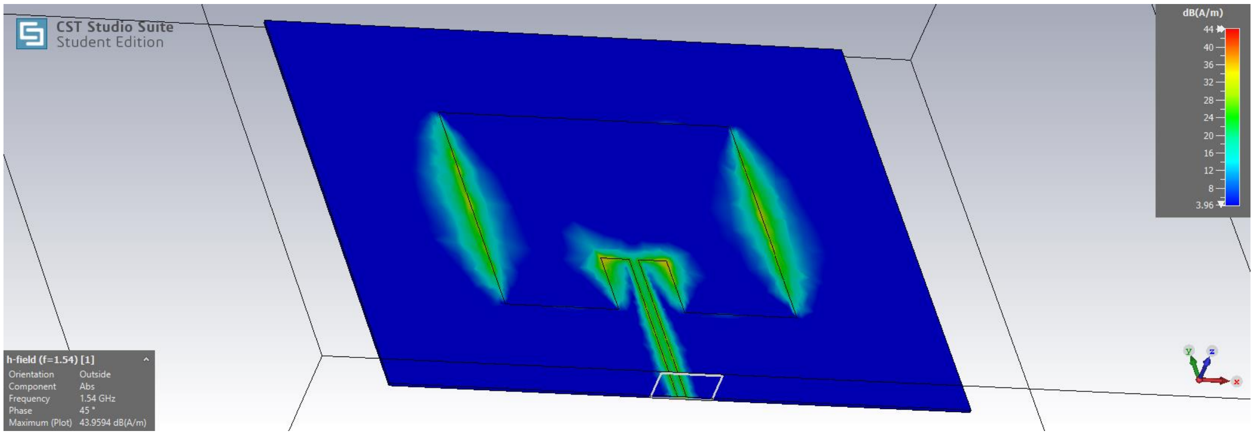

Figure 5. H-field plot (contour) of patch antenna at peak output no standing waves, minimal reflections. Set at an angle for better view.

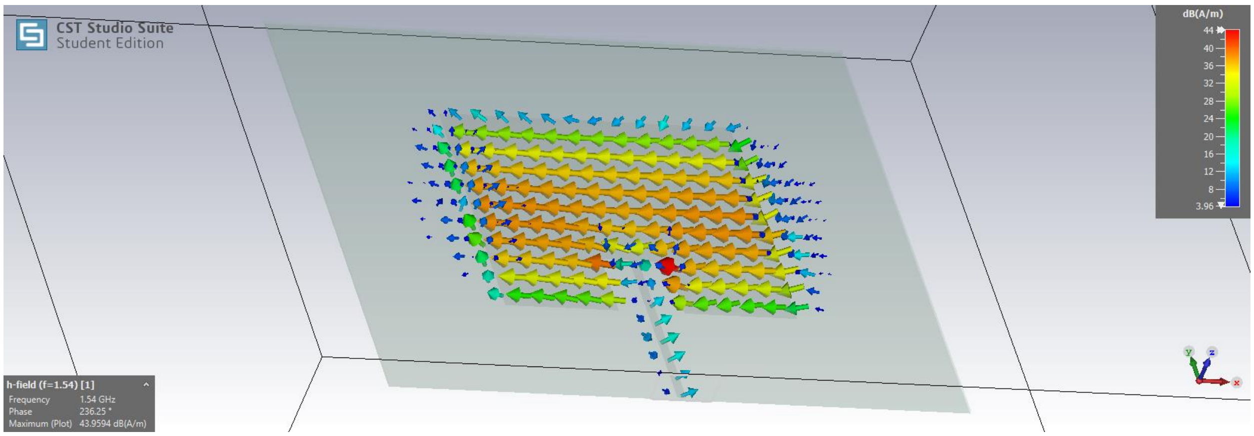

Figure 6. H-field plot (arrow) of patch antenna at peak output no standing waves, minimal reflections. Set at an angle for better view.

For the calibration of the inset size to reach the target frequency, S-parameter was used:

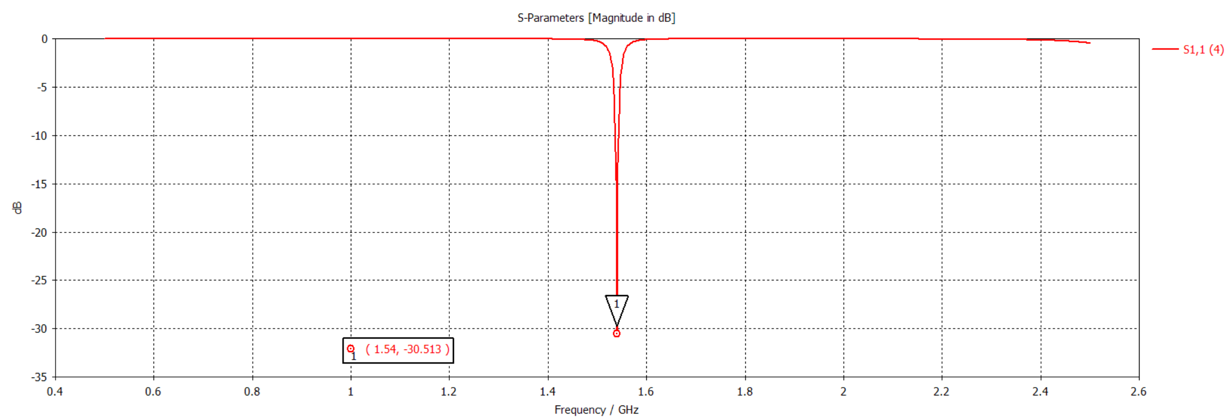

Figure 7. Final S1,1 plot after fine tuning the insets for the patch antenna.

Results were: 1.54GHz at -30.513 dB, well within the -15dB threshold requirement.

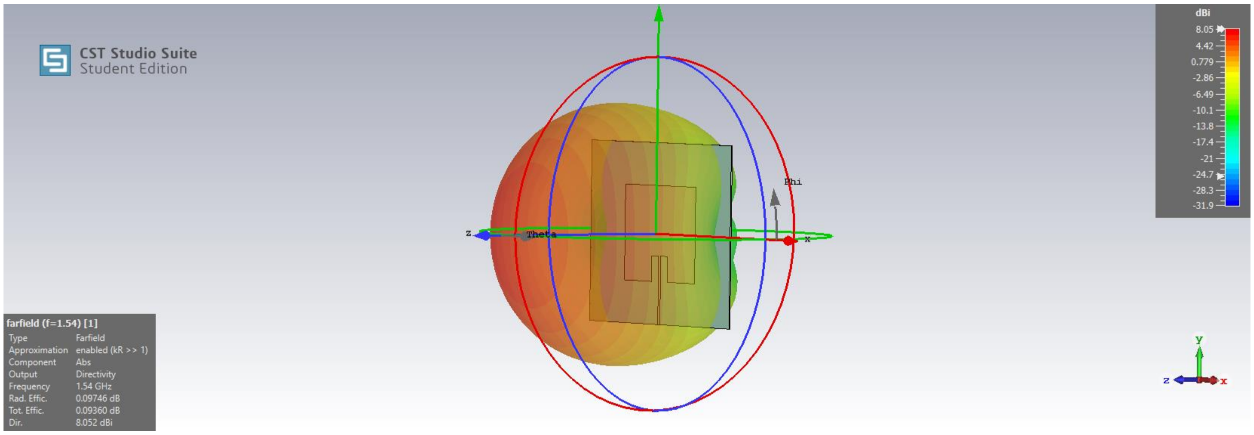

Figure 8. 3D radiation pattern plot for the patch antenna, even "bubble" pattern. Minimal dorsal pattern.

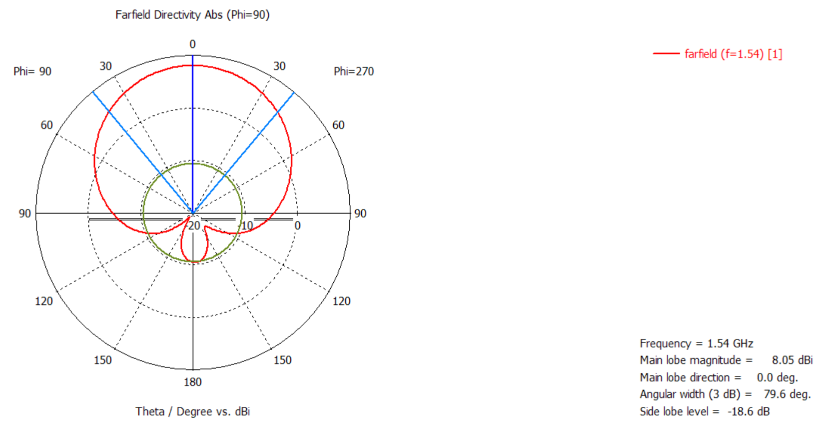

Figure 9. 1D radiation pattern plot, very even pattern.

Results: max magnitude: 8.05 dBi for main lobe @ 1.54 GHz.

Summary:

Based on the CST simulation results, the calculated target frequency of 1.54GHz was achieved at -30dB by changing the inset values up/down until 17 length and width. The contour and arrow plots of the E-field and H-field showed minimal reflections, which means there is low loss of power in this design. Also, the radiation pattern plots showed an even main lobe radiating out of the patch with a minimal lobe on the dorsal side. Lastly, the 3D and 1D plots reported a magnitude of 8.05 dBi for the main lobe.

II. MATLAB Plots & Calculations

Part 1: Calculating and Plotting Distance vs R_X Power

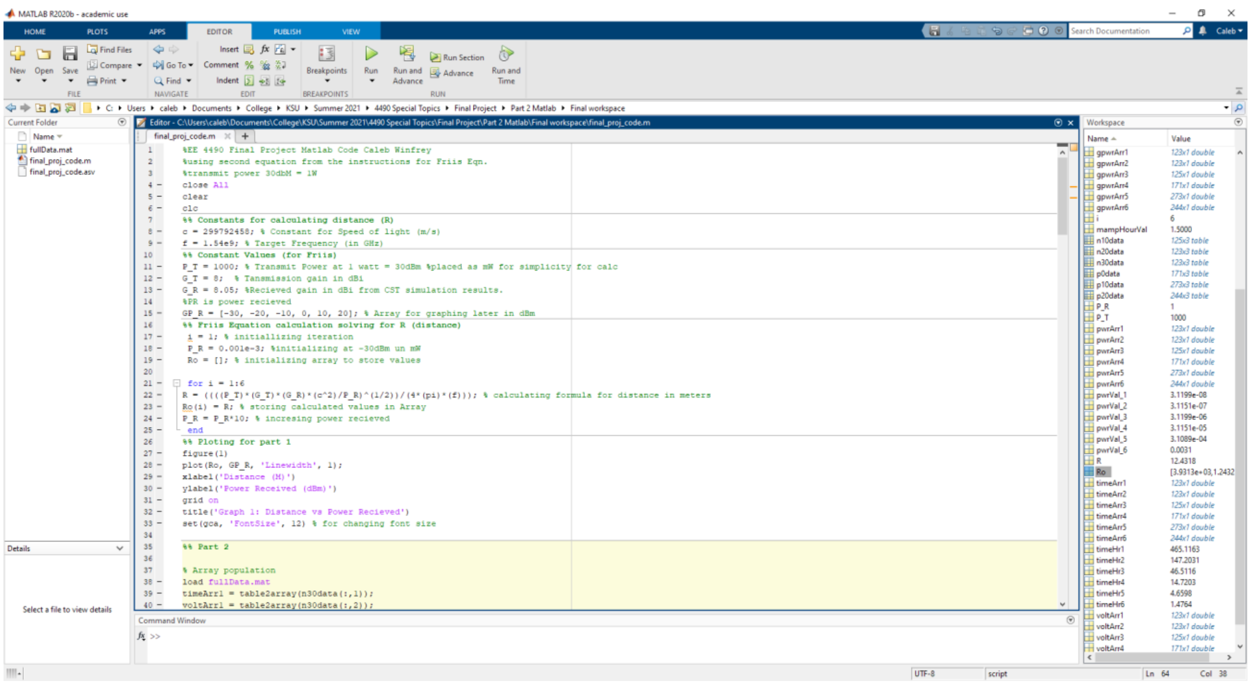

Figure 10. MATLAB Workspace with code for Part 1.

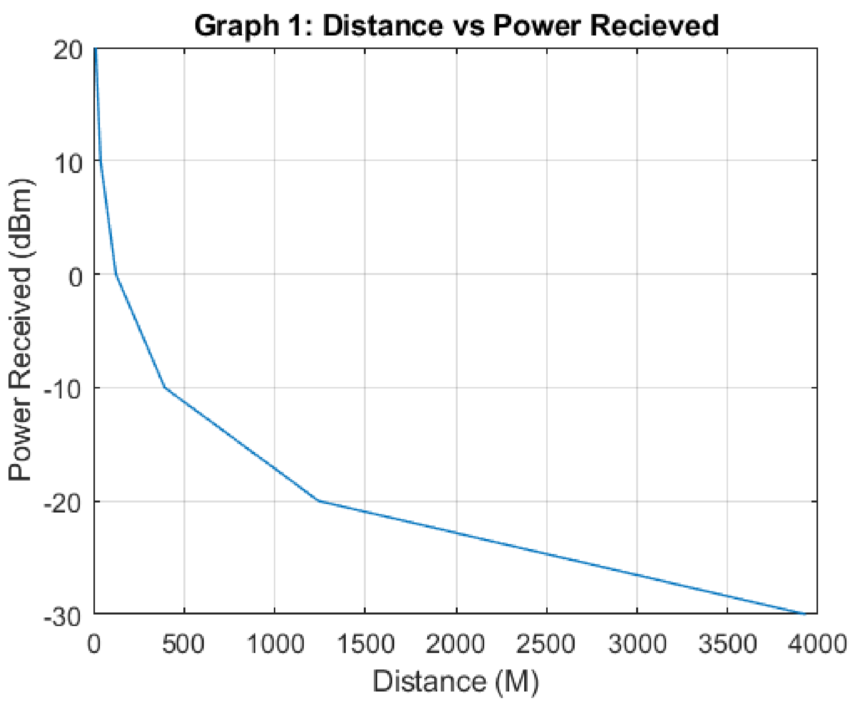

Figure 11. Plot of calculated distance for given power received.

Part 2. Calculating output power from each received power.

Note: For this calculation, ADS was used to simulate a modified circuit from the midterm project. All data was exported and plotted with MATLAB. *See included .m file and back of report for full code.*

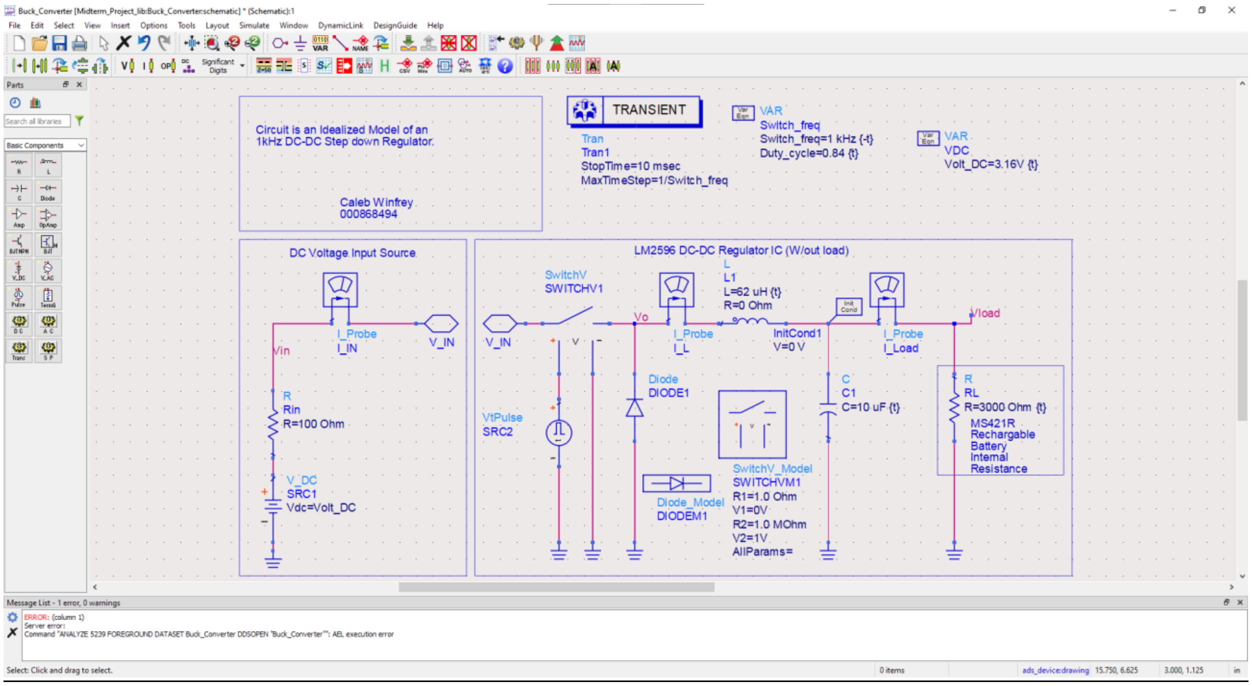

Figure 12. Screenshot of ADS workspace for the modified circuit.

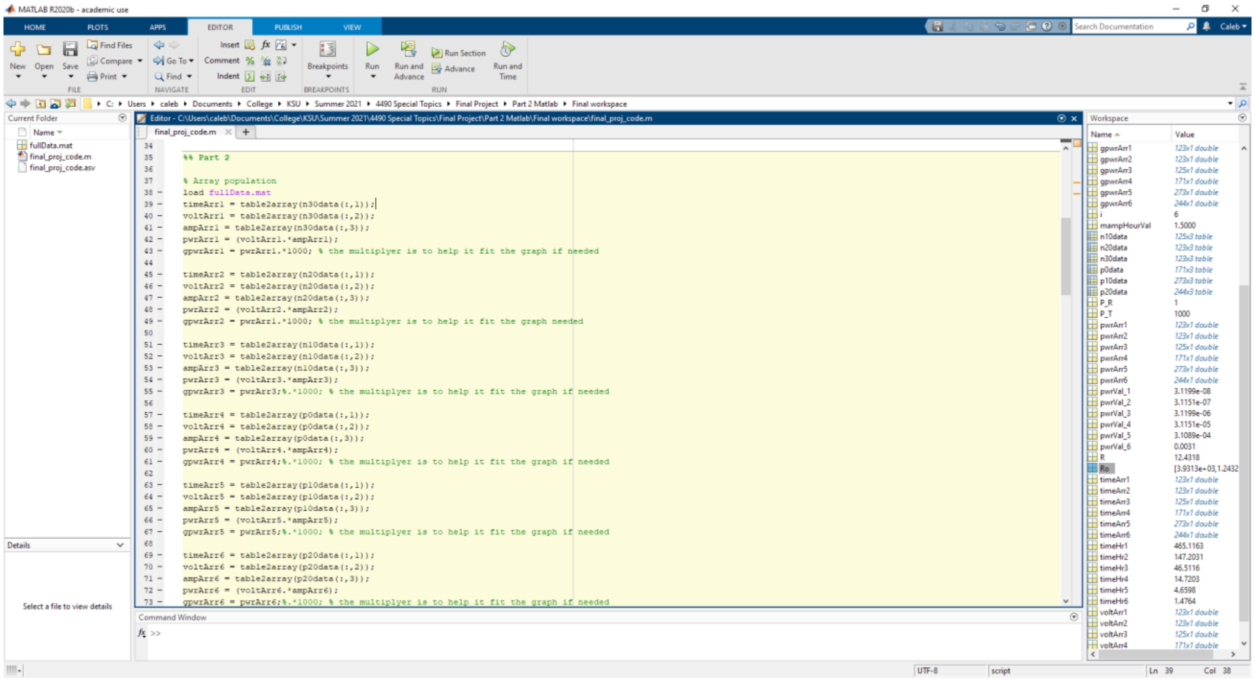

Figure 13. Workspace with portion of code, for part 2. Code shown is populating different arrays from imported data.

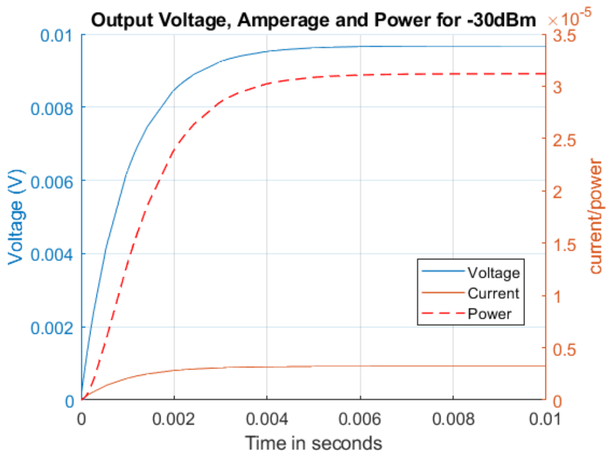

Figure 14. Output power graph with Voltage and Current.

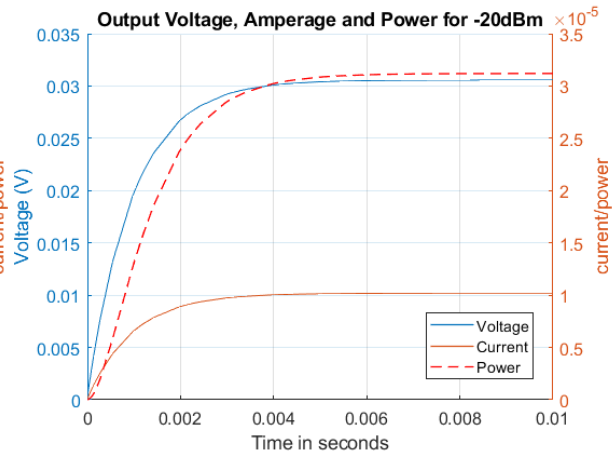

Figure 15. Output power graph with Voltage and Current.

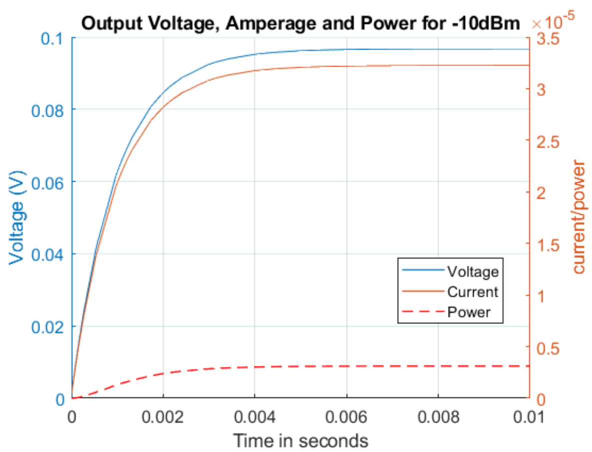

Figure 16. Output power graph with Voltage and Current.

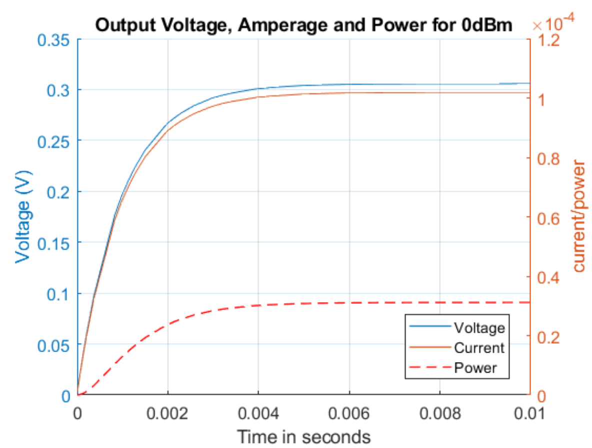

Figure 17. Output power graph with Voltage and Current.

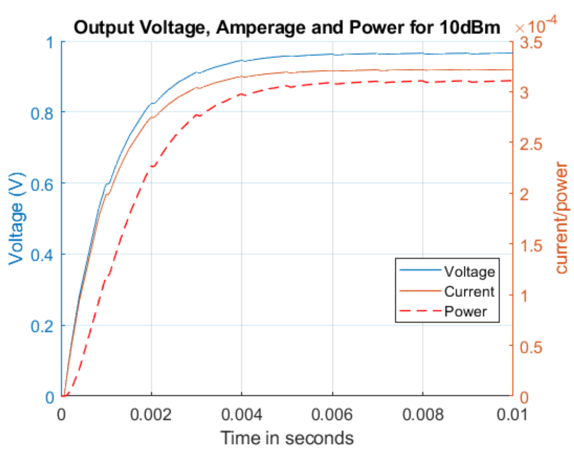

Figure 18. Output power graph with Voltage and Current.

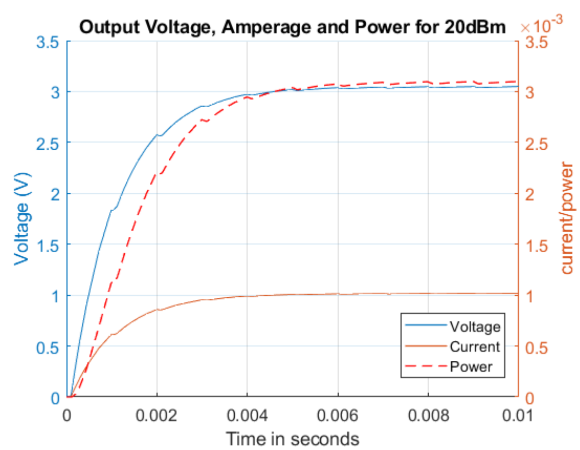

Figure 19. Output power graph with Voltage and Current.

Part 3: Charging Times for battery given Received powers.

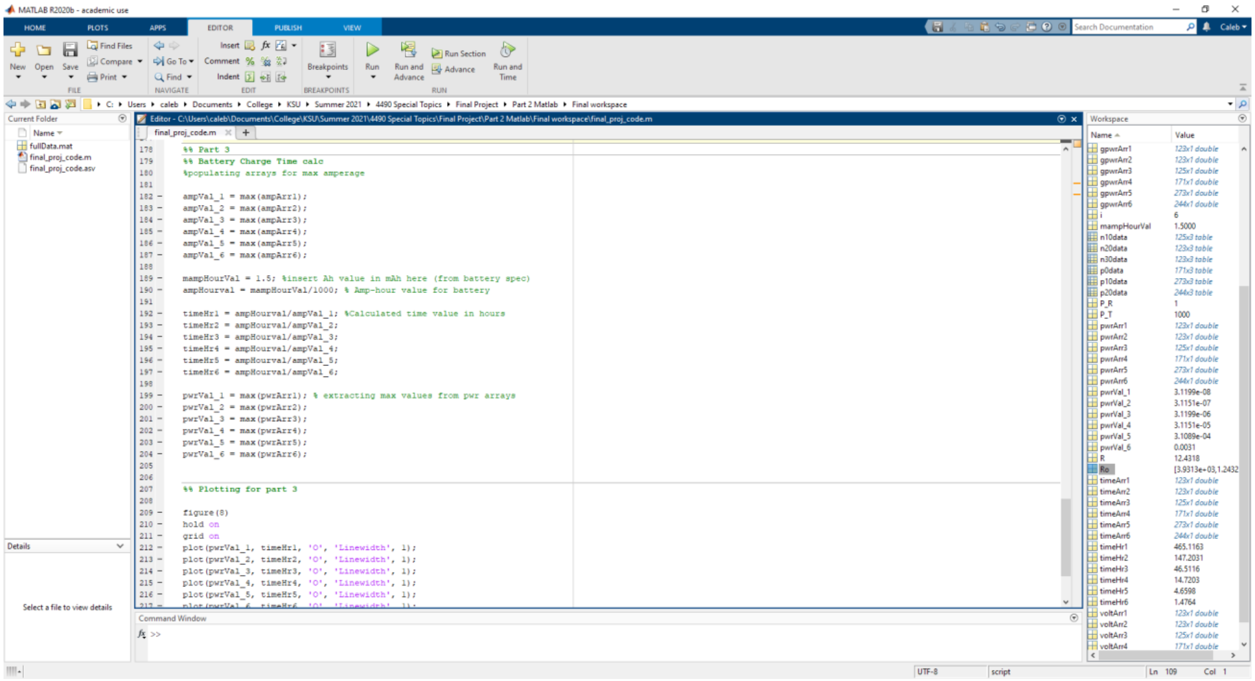

Figure 20. Workspace with portion of code, showing calculating charge times and populating needed constant Values.

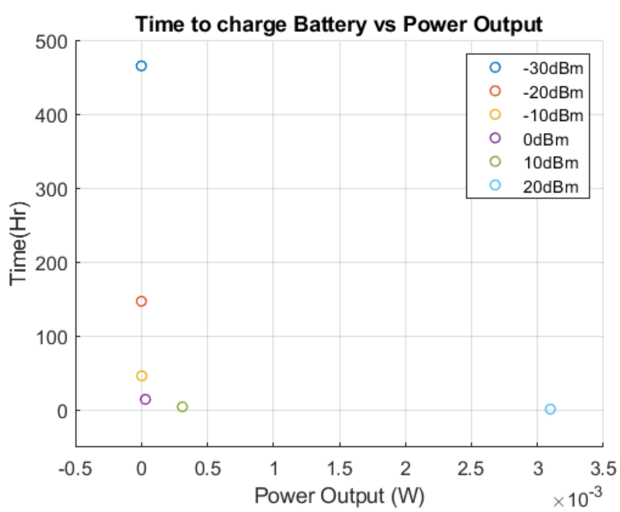

Figure 21. Plot of Each Power output vs Time in hours. Note, the legend shows which received power correlates.

RX Power (dBm)	Charge Time (Hr)	Distance (M)
-30	465.1	3931
-20	147.2	1243
-10	46.51	393.1
0	14.72	124.3
10	4.659	39.31
20	1.476	12.43

Figure 22. Table showing Rx power, charge time and distance.

Summary:

This portion of the project had its challenges but displayed correlating results. For the first part, the Friis formula was rearranged to solve for distance at different received powers. The plot of the data showed that with a larger distance, less power is received which was expected. For part 2, simulating from ADS was used to get the needed data, the power received needed to be at the equivalent watts for the dBm values. To do this a 100Ω series resistor was added and stayed constant throughout the project, this allowed for the voltage to be changed to meet power requirements. Also, the switching frequency, duty cycle and load resistor were amended to match the requirements for the selected battery from the midterm project, (which is 1.5mAh). Exporting the data, and cleaning it up for MATLAB, the plots showed that power received correlates power output, (as expected).

For the third part, the charge time was calculated using the amp-hour rating spec of the battery, using MATLAB's max function managed to take the maximum amperage output to divide with the amp-hour rating to get the charge time for each amperage value. Then, the values were all plotted on a single graph to show that charge times decreased with the more power received and output, which was expected.

Special Thanks:

Special thanks to Morgan Winters for creating a specialized Python program to help with cleaning up the exported data from ADS.

MATLAB Code:

```
%EE 4490 Final Project Matlab Code Caleb Winfrey
%using second equation from the instructions for Friis Eqn.
%transmit power 30dbM = 1W
close All
clear
clc
%% Constants for calculating distance (R)
c = 299792458; % Constant for Speed of light (m/s)
f = 1.54e9; % Target Frequency (in GHz)
%% Constant Values (for Friis)
P_T = 1000; % Transmit Power at 1 watt = 30dBm %placed as mW for simplicity for calc
G_T = 8;  % Transmission gain in dBi
G_R = 8.05; %Received gain in dBi from CST simulation results.
%PR is power received
GP_R = [-30, -20, -10, 0, 10, 20]; % Array for graphing later in dBm
%% Friis Equation calculation solving for R (distance)
 i = 1; % initializing iteration
 P_R = 0.001e-3; %initializing at -30dBm un mW
 Ro = []; % initializing array to store values

 for i = 1:6
R = ((((P_T)*(G_T)*(G_R)*(c^2)/P_R)^(1/2))/(4*(pi)*(f))); % calculating formula for distance in meters
Ro(i) = R; % storing calculated values in Array
P_R = P_R*10; % increasing power received
 end
%% Plotting for part 1
figure(1)
plot(Ro, GP_R, 'Linewidth', 1);
xlabel('Distance (M)')
ylabel('Power Received (dBm)')
grid on
title('Graph 1: Distance vs Power Recieved')
set(gca, 'FontSize', 12) % for changing font size
```

```
%% Part 2
% Array population
load fullData.mat
timeArr1 = table2array(n30data(:,1));
voltArr1 = table2array(n30data(:,2));
ampArr1 = table2array(n30data(:,3));
pwrArr1 = (voltArr1.*ampArr1);
gpwrArr1 = pwrArr1.*1000; % the multiplier is to help it fit the graph if needed

timeArr2 = table2array(n20data(:,1));
voltArr2 = table2array(n20data(:,2));
ampArr2 = table2array(n20data(:,3));
pwrArr2 = (voltArr2.*ampArr2);
gpwrArr2 = pwrArr1.*1000; % the multiplier is to help it fit the graph needed

timeArr3 = table2array(n10data(:,1));
voltArr3 = table2array(n10data(:,2));
ampArr3 = table2array(n10data(:,3));
pwrArr3 = (voltArr3.*ampArr3);
gpwrArr3 = pwrArr3;%.*1000; % the multiplier is to help it fit the graph if needed

timeArr4 = table2array(p0data(:,1));
voltArr4 = table2array(p0data(:,2));
ampArr4 = table2array(p0data(:,3));
pwrArr4 = (voltArr4.*ampArr4);
gpwrArr4 = pwrArr4;%.*1000; % the multiplier is to help it fit the graph if needed

timeArr5 = table2array(p10data(:,1));
voltArr5 = table2array(p10data(:,2));
ampArr5 = table2array(p10data(:,3));
pwrArr5 = (voltArr5.*ampArr5);
gpwrArr5 = pwrArr5;%.*1000; % the multiplier is to help it fit the graph if needed

timeArr6 = table2array(p20data(:,1));
voltArr6 = table2array(p20data(:,2));
ampArr6 = table2array(p20data(:,3));
pwrArr6 = (voltArr6.*ampArr6);
gpwrArr6 = pwrArr6;%.*1000; % the multiplier is to help it fit the graph if needed

%% Plotting figures
figure(2)  %-30dBm
grid on
hold on
yyaxis left
plot(timeArr1,voltArr1, 'Linewidth', 1)
xlabel('Time in seconds')
ylabel('Voltage (V)')
set(gca, 'FontSize', 12) % for changing font size

yyaxis right
plot(timeArr1,ampArr1, 'Linewidth', 1)
plot(timeArr1,gpwrArr1, 'r--', 'Linewidth', 1)
ylabel('current/power')
title('Output Voltage, Amperage and Power for -30dBm')
legend('Voltage', 'Current', 'Power')
hold off

figure(3) %-20dBm
grid on
hold on
yyaxis left
plot(timeArr2,voltArr2, 'Linewidth', 1)
xlabel('Time in seconds')
ylabel('Voltage (V)')
set(gca, 'FontSize', 12) % for changing font size

yyaxis right
plot(timeArr2,ampArr2, 'Linewidth', 1)
plot(timeArr2,gpwrArr2, 'r--', 'Linewidth', 1)
ylabel('current/power')
title('Output Voltage, Amperage and Power for -20dBm')
legend('Voltage', 'Current', 'Power')
hold off

figure(4) %-10dBm
grid on
hold on
yyaxis left
plot(timeArr3,voltArr3, 'Linewidth', 1)
xlabel('Time in seconds')
ylabel('Voltage (V)')
set(gca, 'FontSize', 12) % for changing font size

yyaxis right
plot(timeArr3,ampArr3, 'Linewidth', 1)
plot(timeArr3,gpwrArr3, 'r--', 'Linewidth', 1)
ylabel('current/power')
title('Output Voltage, Amperage and Power for -10dBm')
legend('Voltage', 'Current', 'Power')
hold off
```

```
figure(5) %0dBm
grid on
hold on
yyaxis left
plot(timeArr4,voltArr4, 'Linewidth', 1)
xlabel('Time in seconds')
ylabel('Voltage (V)')
set(gca, 'FontSize', 12) % for changing font size

yyaxis right
plot(timeArr4,ampArr4, 'Linewidth', 1)
plot(timeArr4,gpwrArr4, 'r--', 'Linewidth', 1)
ylabel('current/power')
title('Output Voltage, Amperage and Power for 0dBm')
legend('Voltage', 'Current', 'Power')
hold off

figure(6) %10dBm
grid on
hold on
yyaxis left
plot(timeArr5,voltArr5, 'Linewidth', 1)
xlabel('Time in seconds')
ylabel('Voltage (V)')
set(gca, 'FontSize', 12) % for changing font size

yyaxis right
plot(timeArr5,ampArr5, 'Linewidth', 1)
plot(timeArr5,gpwrArr5, 'r--', 'Linewidth', 1)
ylabel('current/power')
title('Output Voltage, Amperage and Power for 10dBm')
legend('Voltage', 'Current', 'Power')
hold off

figure(7) %20dBm
grid on
hold on
yyaxis left
plot(timeArr6,voltArr6, 'Linewidth', 1)
xlabel('Time in seconds')
ylabel('Voltage (V)')
set(gca, 'FontSize', 12) % for changing font size

yyaxis right
plot(timeArr6,ampArr6, 'Linewidth', 1)
plot(timeArr6,gpwrArr6, 'r--', 'Linewidth', 1)
ylabel('current/power')
title('Output Voltage, Amperage and Power for 20dBm')
legend('Voltage', 'Current', 'Power')
hold off
%% Part 3
%% Battery Charge Time calc
%Populating arrays for max amperage
ampVal_1 = max(ampArr1);
ampVal_2 = max(ampArr2);
ampVal_3 = max(ampArr3);
ampVal_4 = max(ampArr4);
ampVal_5 = max(ampArr5);
ampVal_6 = max(ampArr6);
mampHourVal = 1.5; %insert Ah value in mAh here (from battery spec)
ampHourval = mampHourVal/1000; % Amp-hour value for battery
timeHr1 = ampHourval/ampVal_1; %Calculated time value in hours
timeHr2 = ampHourval/ampVal_2;
timeHr3 = ampHourval/ampVal_3;
timeHr4 = ampHourval/ampVal_4;
timeHr5 = ampHourval/ampVal_5;
timeHr6 = ampHourval/ampVal_6;
pwrVal_1 = max(pwrArr1); % extracting max values from pwr arrays
pwrVal_2 = max(pwrArr2);
pwrVal_3 = max(pwrArr3);
pwrVal_4 = max(pwrArr4);
pwrVal_5 = max(pwrArr5);
pwrVal_6 = max(pwrArr6);

%% Plotting for part 3
figure(8)
hold on
grid on
plot(pwrVal_1, timeHr1, 'O', 'Linewidth', 1);
plot(pwrVal_2, timeHr2, 'O', 'Linewidth', 1);
plot(pwrVal_3, timeHr3, 'O', 'Linewidth', 1);
plot(pwrVal_4, timeHr4, 'O', 'Linewidth', 1);
plot(pwrVal_5, timeHr5, 'O', 'Linewidth', 1);
plot(pwrVal_6, timeHr6, 'O', 'Linewidth', 1);
xlim([-0.5e-03, 3.5e-03]) % these settings are to make the points on the graph look better
ylim([-50, 500])
xlabel('Power Output (W)')
ylabel('Time(Hr)')
title('Time to charge Battery vs Power Output')
legend('-30dBm', '-20dBm', '-10dBm', '0dBm','10dBm','20dBm')
set(gca, 'FontSize', 12) % for changing font size
hold off
```