How to Model Gain and Phase using Spherical Harmonics

Suren R Gourapura

July 21, 2017

1 Files

Needed Files:

a. `data.xlsx` : For a given frequency, this includes the gain and phase values for all directions in 5 degree increments

b. `eff_Data_Gen.cpp` : This script allows the arrangement of excel data into a Mathematica-friendly format

c. `fit_Sph_Har.nb` : This notebook allows us to plot the data and fit with $L = 0 - 12$ Spherical Harmonics

d. `general_AvgDev_Chi2.cpp` : This script allows us to test any fit with $L = 0 - 12$ Spherical Harmonics

Additional Files:

e. `visual_Rep_Of_Sph_Har` : To better see how additional spherical harmonics improve the fit (demonstrated with gain for 83.33 MHz)

f. `m_NotEqualTo_0_NoEff` : Proof that non-zero m values do not improve the fit (at least with gain for 83.33 MHz)

2 How to Format Data for Mathematica

Copy a column of gain or phase data from the `data.xlsx` file. This should include 2664 values. Note that I have converted the gain data to v/v and phase data to radians for the first frequency, although this is not necesary. Run the `eff_Data_Gen.cpp` file. Paste the column into the terminal. The output will have the correct curly braces and commas for Mathematica. Open `fit_Sph_Har.nb`, select the current data, and paste in the formatted data from the output of the C++ file. Make sure that the final comma of the output is removed, so that it looks like this:

$dataPhase$ ={`output here without the last comma`}

Evaluating the file should now allow the data to be plotted and fitted. The fit coefficients should also be outputted. If not, remove the semicolon from the line assigning the coefficient variables and evaluate again. The coefficients are in order from Y00 to Y012.

3 How to Find Chi^2 and the Average Deviation

First, open `general_AvgDev_Chi2.cpp` and edit the output file name to your choosing. To stay organized, I suggest:

FitTest.(G for gain or P for Phase)(frequency).Y00-Y(largest m-value being used, default is 0)(largest L-value being used, default is 12)

If you desire, further spherical harmonics can be added in the file below the Y012 equation, or current harmonics can be commented out for a simpler fit. Don't forget to change numCoeff in the file to however many harmonics you have decided to use. Next, obtain the fit coefficients from the previous step. You will need to enter each one-by-one into the `general_AvgDev_Chi2.cpp` file. Note that the script cannot handle scientific notation, so I suggest placing each coefficient into an excel column with an enter in front to evaluate the scientific notation and then copy-pasting the entire column into the C++ script. Next, the file needs the gain or phase data. Copy and paste the 2664 values from the excel worksheet into the script. Finally, enter the directory you have saved the C++ script in and find the file you have named above. In it, you will find the Average Deviation and the Chi^2.

4 Additional Information

To see how the number of spherical harmonics improves the fit, check out the `visual_Rep_Of_Sph_Har` file.

To see how the non-zero m values affect a set of data, check out the `m_NotEqualTo_0_NoEff` file. Note that I have already put the gain data of 83.33 MHz and by looking at the coefficients attached to the non-zero m spherical harmonics, one can conclude that they are not necessary. This may not be true for all gain and phase patterns!

5 Future Work

I have mapped out and fit the gain and phase of the first frequency (83.33 MHz). The rest need to be mapped and fit to ensure that this method will fit well all gain and phase patterns of our antenna. I believe that the phase data will cause the most problems, as they are not symmetric about the z-axis near the poles. Secondly, our fits need to be put into ARASim to see if they perform as well as the actual antenna data.