Final Project

Electric and Magnetic Fields

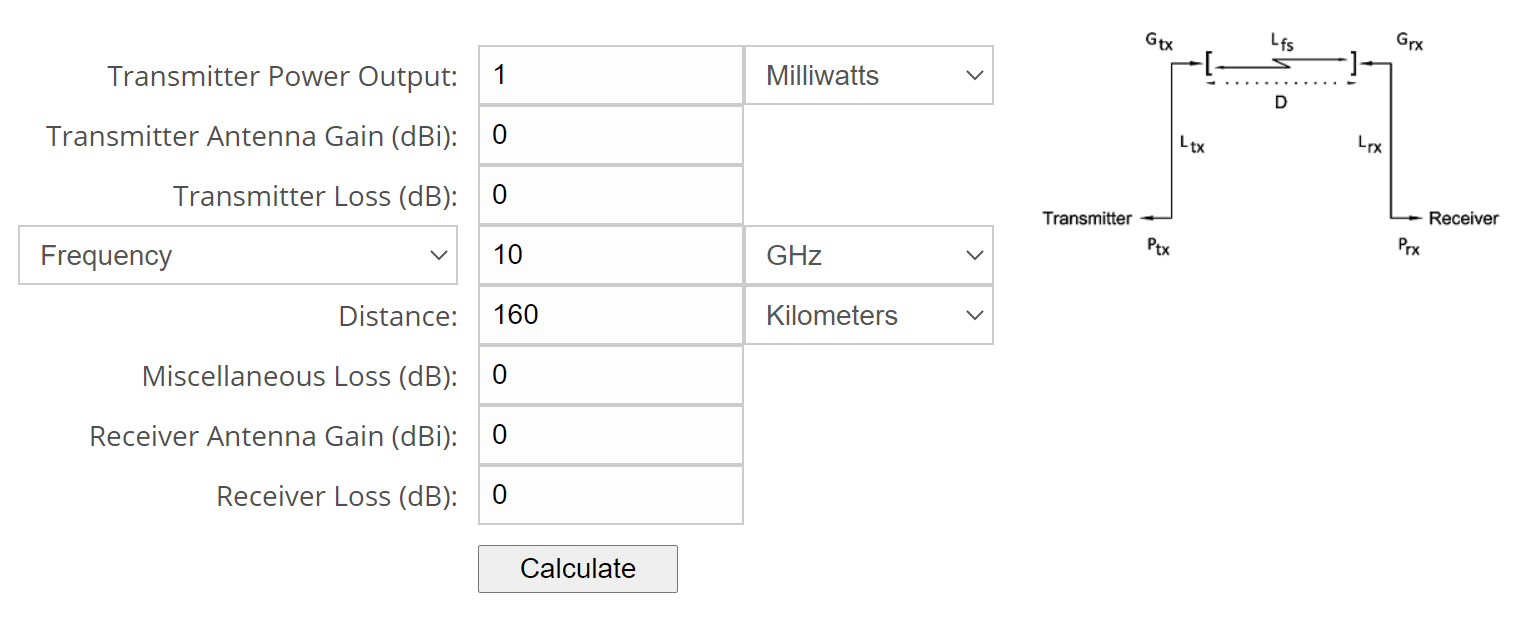
**Your Name**:

In this project, you are to write a Matlab program in order to calculate various performance aspects of a space-based radar operating at 10 GHz. The Matlab program should explicitly either a) output to the command window, or b) create a graph for each of the steps below. The output, if a number, should have a statement explaining what the quantity is. It is expected that you will continually add to the code snip provided with the assignment, as you proceed through each step that requires Matlab.

In several of the steps below, you are asked to provide answers to questions. Please type the answers into this document and include the document with your project submission materials.

The code snip ECE390\_Final\_Project\_2020 \_snip.m is your starting point. It contains much of the code that is needed for this project, but there are also additions you need to make to complete the assignment. I suggest you spend some time getting familiar with the code before you begin.

1. Calculate and plot the free-space path loss in dB at 10 GHz versus distance for d = 100 to 1000 **km.** This step is helpful in thinking about the Friis transmission equation. Note: you can check your calculations using the Link Budget calculator at this website - <https://www.pasternack.com/t-calculator-link-budget.aspx>. You can use the input parameters shown below, keeping in mind that 1 mW of output power is equal to 0 dBm (this information allows you to determine the path loss from the Received Power number displayed by the calculator). There are other calculators on the site that you might find interesting. Add the calculations for this step to the provided Matlab code.



1. Calculate the real and imaginary relative permittivity of soil versus frequency from 1.5 GHz up to 10 GHz. Plot these values for volumetric moisture content values from 0 to 80% assuming the temperature is 20C. Use the code in ECE390\_Final\_Project\_2020 \_snip.m. You need to specify the frequency parameters (in Hz!!), the moisture increment step size and the temperature. \*\* **Do not change values in the “Other Soil Parameters and Constants” section of the file**. The provided code will run and generate the graphs of r’ (real) and r’’ (imaginary) relative permittivity versus frequency automatically. *Data check: for 20% moisture content, you should find the real and imaginary parts to be approximately 12.9 and 3.6, respectively, at 10 GHz*. *Interesting back-story: The equations in the Matlab code for calculating the soil permittivity come from a paper published by Craig Dobson, a graduate student who worked for Professor Fawwaz Ulaby, the author of the textbook we are using*.
2. Repeat the previous step for temperatures of 0C and 32C. Comment on the differences in the results.
   1. Do you think a radar could be used to measure soil temperature? Explain your answer.
3. Now run the code to generate a plot of r’ (real) and r’’ (imaginary) relative permittivity versus moisture **at 10 GHz**. (The code for generating these graphs is already included in the ECE390\_Final\_Project\_2020 \_snip.m file. Confirm that the plot is done at the proper frequency by checking the figure caption in the plot window.) The ratio of r’’ to r’ is known as the loss tangent; a larger value for the loss tangent indicates that the material absorbs more energy as a signal passes through it.
   1. What are the loss tangent values at 10%, 40% and 80% moisture levels? (You can add code for these calculations or just read the values from the plot.)
   2. What does this tell you about power absorption in the soil as moisture content increases?
4. Calculate the reflection coefficient, assuming a signal is incident from free space onto the soil, for the permittivity values in the previous step, i.e. at 10 GHz and for volumetric moisture content from 0% to 80%. Plot the magnitude of the reflection coefficient as a linear value (use the ‘*abs’* function) versus moisture content. Also, plot the corresponding return loss in dB versus moisture content. Refer to the equations below for help, and the course lecture notes. *Data check: at 10% moisture, you should find the reflection coefficient magnitude to be approximately 0.47 and the return loss to be approximately 6.6 dB.*
   1. How does the reflection coefficient change versus moisture content, and how does this affect the amount of power reflected from the surface of the soil?

er\_c(i) = er\_p(i)+er\_pp(i)\*j; %complex permittivity er' + er''\*j

Z\_soil(i) = 377/sqrt(er\_c(i)); %wave impedance

Gamma(i) = (377-Z\_soil(i))/(377+Z\_soil(i)); %reflection coefficient

1. Plot the percentage of the incident power that is absorbed in the soil versus moisture content at 10 GHz. Note that sum of the percentages of reflected power and incident power must equal 100%. Also, the ratio of reflected power to the incident power is ||2. *Data check: at 10% moisture, you should find approximately 78% of the power is absorbed.*
2. Now, for the radar calculation: calculate the power received at the radar receiver, assuming the radar illuminates a piece of metal that has a radar cross section of |⋅⋅R2 where R is the distance from the radar to the metal plate. Assume R = 160 km. For the metal plate, the magnitude of the reflection coefficient is 1. Assume the radar transmitter power is 100 W, and that the gain of both the receive and transmit antennas is 10,000 (40 dB). Display the received power in dBW in the Command Window in Matlab with an appropriate text label. *Data check: the power received back from the metal plate should be less than 1 micro-Watt (-60 dBW) and greater than 1 nano-Watt (-90 dBW).*
3. The 10 GHz complex permittivity values for the different moisture levels have been mapped to matrices that are in the files er\_real.txt and er\_imag.txt; these files are included in the Final Project module on the Canvas page. Each element in these matrices corresponds to a pixel that has been measured by your hypothetical radar. The size of the pixel is approximately 0.4 miles by 0.4 miles. The values in the matrices are the soil permittivity for the corresponding pixel. Much of the Matlab code that you will need for these steps is included in the file below; note that portions of the code have been removed and you will need to complete those lines. In this step of the project, you are to:
   1. Read in the matrices with the permittivity values.
   2. Calculate the complex permittivity, the corresponding wave impedance of the soil and the reflection coefficient assuming free space incidence.
   3. Use the above to calculate the radar cross-section for each pixel and the power received back at the radar (assuming the same distance, transmit power and antenna gains from the step #7).
   4. Plot the received power to create a radar image comprised of all the individual pixels; there are a few steps needed to normalize and scale the data so that a nice image is achieved. As mentioned above, note that the matrix reading operation and the image making commands are included in the code sample below.
   5. Do you recognize the image that has been generated? Explain.
   6. By examining the range of values for the power received back at the radar, what is the approximate range of moisture levels among all the pixels in the image?

ermatrixreal=readmatrix('er\_real.txt'); %read matrix of real permittivity values

ermatriximag=readmatrix('er\_imag.txt'); %read matrix of imaginary permittivity values

[row,col] = size(ermatrixreal); %determine the number of rows and columns in the matrices

Pt = X; %transmit power

Gt = X; %40 dB gain transmit antenna

Gr = X; %40 dB gain receive antenna

for i = 1:row

for j = 1:col

er\_c\_radar(i,j) = ermatrixreal(i,j)+ermatriximag(i,j)\*j; %complex permittivity

Z\_soil\_radar(i,j) = X; %wave impedance of soil

Gamma\_radar(i,j) = X; %reflection coefficient

RL\_radar\_dB(i,j) = X; %return loss

sigma = abs(Gamma\_radar(i,j))^2\*X; %radar cross-section

Prec = Pt\*Gt\*Gr…X; %power received at radar

Prec\_dB(i,j) = 10\*log10(Prec);

end

end

figure

maxPrec = max(abs(Prec\_dB),[],'all'); %find max value for normalization

minPrec = min(abs(Prec\_dB),[],'all'); %find min value for normalization

%disp(maxPrec);

%disp(minPrec);

for i = 1:row

for j = 1:col

Prec\_dB(i,j) = (-Prec\_dB(i,j)-minPrec)/(maxPrec-minPrec)\*255; %normalize and scale max value to 255

% there is a minus sign on "Prec\_dB" in the previous line in order

% for the plotting to work.

int8\_Prec\_dB(i,j) = uint8(Prec\_dB(i,j)); %convert to unsigned 8-bit integer

end

end

imshow(int8\_Prec\_dB); %display the image