ECE 693 Project 1

Trent Collins

2/5/2025

Introduction

This Project implemented three different approaches to path loss. Each implementation accounts for different factors. First, free space path loss, then refraction (according to the knife model), and finally reflection. The programs implement equations and concepts from lectures to complete the calculations. Figure 1 shows diagrams representing the models used for each section (drawn in photoshop). First, is a one ray free space model shown in the uppermost part of the figure. Next, is a diffraction example in the middle figure showing the signal going “around” the object. Finally, is a two-ray model, showing the signal reflecting. Matlab was used to create these applications using their “App Builder” tool kit. The programs allow users to input variables, and have a reference diagram assisting in the visualization of each parameter. There is also a description of the program that explains its functionality. Finally, the box on the bottom right outputs the desired information to the user. Each program was built in Matlab and then validated with hand calculations for multiple input values.

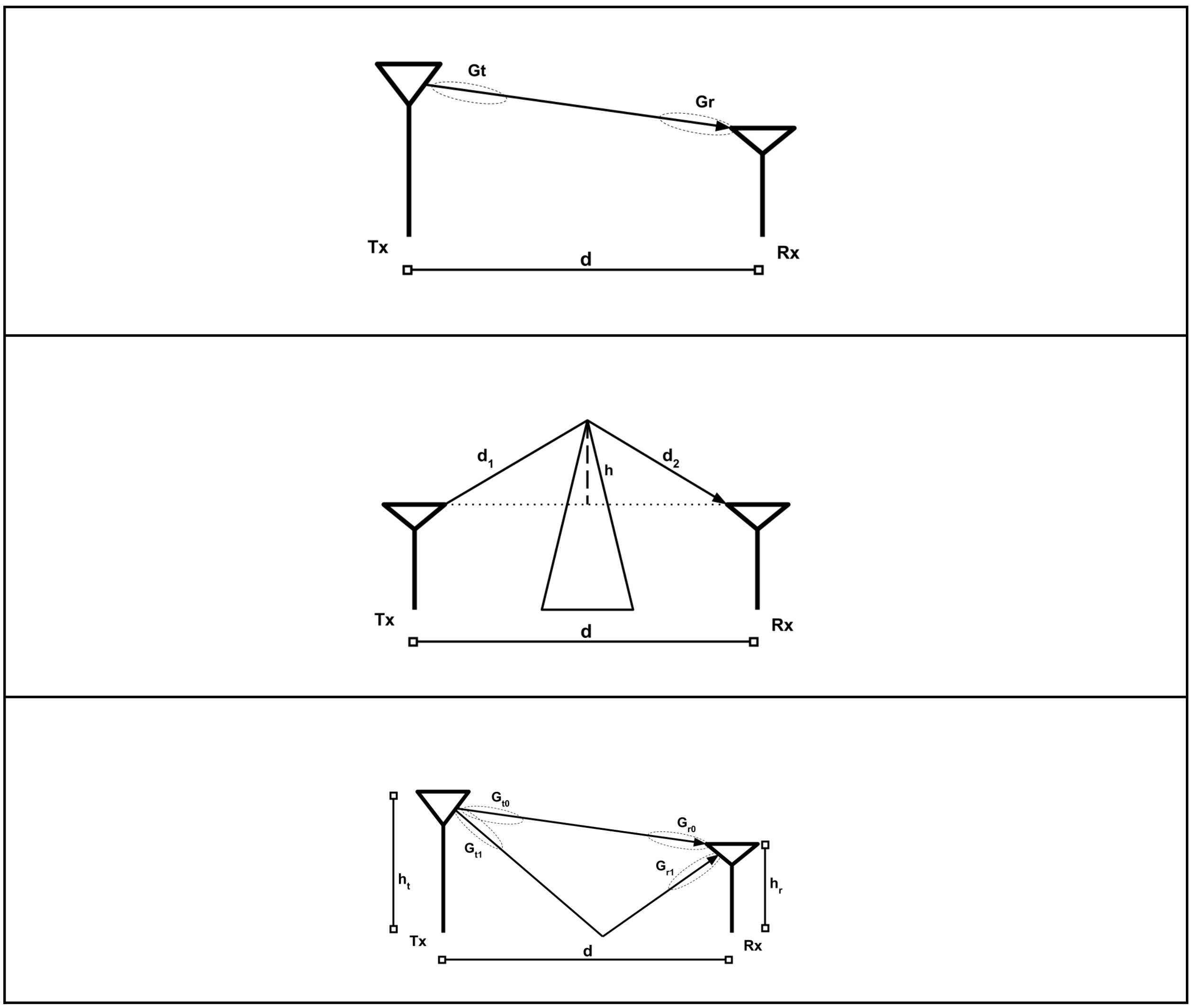

Figure 1. Diagrams Implemented

<u>Part I</u>

In this portion, the free space model for path loss was implemented. This implemented the equation $P_{LdB} = 10log((\frac{4\pi d}{\lambda})^2 * (\frac{1}{G_t G_r}))$. The code to implement the calculations can be seen in Figure 2. Figure 3 shows the input-output for two different examples, highlighting the GUI. It should be noted that both outputs yield the expected results in Figure 3. As for the left side $P_{LdB} = 10log((\frac{4\pi*100m}{c/5GHz})^2 * (\frac{1}{1})) = 86.42$ and for the right side $P_{LdB} = 10log((\frac{4\pi*10m}{c/5GHz})^2 * (\frac{1}{1})) = 66.42$.

```
% Button down function: LeftPanel
function LeftPanelButtonDown(app, event)
    d = app.DistancemEditField.Value;
    l = (3*10^8)/(app.FrequencyGHzEditField.Value * 10^9);
    Gt = app.GtEditField.Value;
    Gr = app.GrEditField.Value;
    app.PathLossdBEditField.Value = 10*log10((((4*pi*d)/l)^2) * (1/Gr*Gt));
end
```

Figure 2. Code For Part I

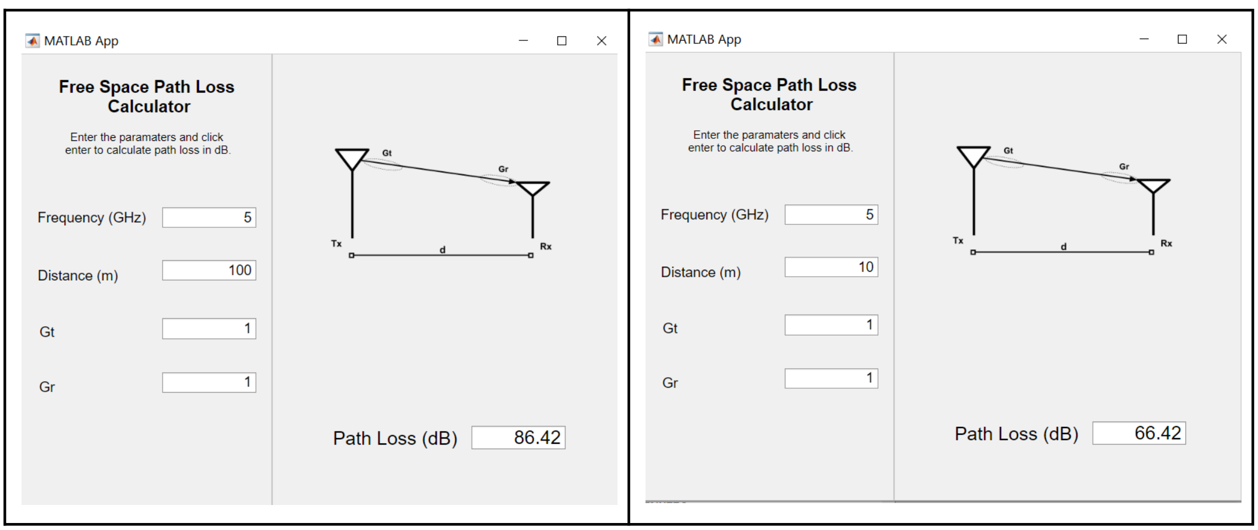

Figure 3. Example I/O For Part I

Part II

Part II implemented refraction with the Knife Model. This models the refraction of the signal as it passes over the “edge” of an obstacle. This model first calculates a diffraction parameter using the equation $v = h * \sqrt{\frac{2*(d^{*}+d^{*})}{\lambda*d^{*}*d^{*}}}$. Next, depending on the value calculated, there are five possible equations that can be implemented in a piecewise function to find the path loss. These equations are shown in Figure 4 [1]. The implementation of this is seen in Figure 5. First, the diffraction parameter was calculated. Then, using an if-else block, the loss was calculated with the respective equation for each parameter value. Figure 6 shows the GUI, as well as examples for each possible range. The figure in the GUI illustrates the program's application and describes each variable. The program also explains each parameter and its meaning to avoid confusion. Table I shows the hand calculations for these values. When values are compared from Table I and Figure 6, the results are validated.

$$L(v)\mathrm{dB} = \begin{cases} 0, & v < -1, \\ 20\log_{10}[.5 - .62v], & -0.8 \le v < 0, \\ 20\log_{10}[.5e^{-.95v}], & 0 \le v < 1, \\ 20\log_{10}[.4 - \sqrt{.1184 - (.38 - .1v)^2}], & 1 \le v \le 2.4, \\ 20\log_{10}[.225/v], & v > 2.4. \end{cases}$$

Figure 4. Function To Calculate L for Part II

```
% Button down function: LeftPanel
function LeftPanelButtonDown(app, event)
    %%Input Variables
    h = app.hmEditField.Value;
    d1 = app.d1mEditField.Value;
    d2 = app.d2mEditField.Value;
    %%Input Lambda (c/f)
    l  = (3*10^8) / (app.FrequencyGHzEditField.Value * 10^9);
    %%Calculate v
    if ( (l > 0) && ( d1 > 0 ) && (d2 > 0) )
        v = h * sqrt( (2*(d1+d2)) / (l * d1 * d2));
    else
        v = 0; %%Condition to avoid dividing by zero
    end
    %%Find Loss depending on v
    if (v < - 1)
       PL = 0;
    elseif( (v >= -0.8) && (v < 0) )
       PL = 20*log10(0.5 - 0.62*v);
    elseif( (v >= 0) && (v < 1) )
       PL = 20*log10(0.5*exp(-0.95*v));
    elseif( (v >= 1) && (v <= 2.4) )
       PL = 20*log10(0.4 - sqrt(0.1184 - (0.38-0.1*v)^2));
    elseif (v > 2.4)
       PL = 20*log10(0.225/v);
    end
    %%Update Knife Parameter Button
    app.KnifeParamterEditField.Value = v;

    %%Update Path Loss Parameter Button
    app.PathLossdBEditField.Value = PL;
end
```

Figure 5. Code For Part II

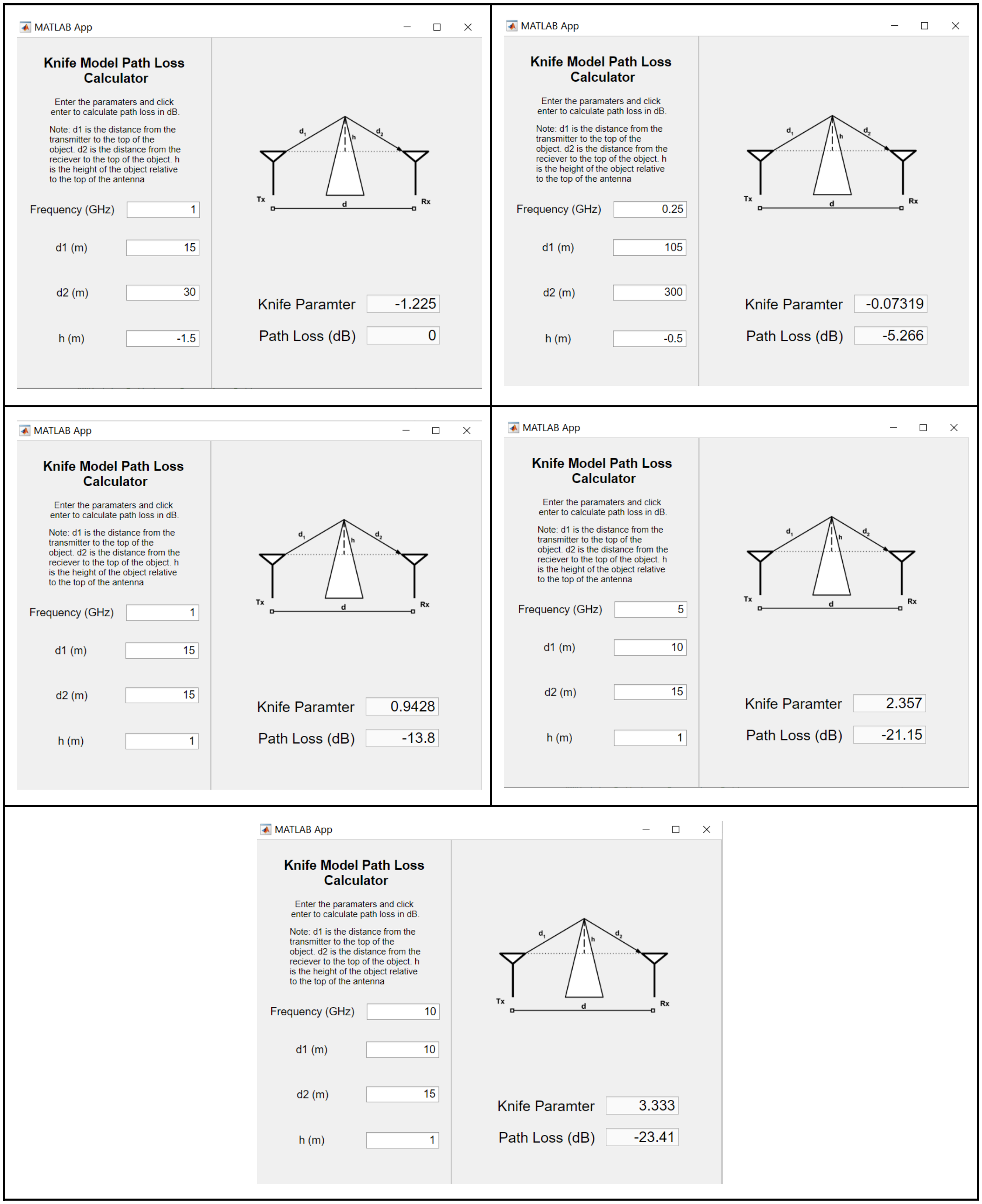

Figure 6. Example I/O For Part II

Table I. Hand Calculations for Part II

F (GHz)	D_1(m)	D_2(m)	H(m)	Knife Parameter	L(dB)
1	15	30	-1.5	-1.225	0
0.25	105	300	-0.5	-0.07	-5.27
1	15	15	1	0.94	-13.8
5	10	15	1	2.357	-21.15
10	10	15	1	3.333	-23.41

Part III

This final portion implemented a two-ray model for calculating path loss in dB. This implemented the function $P_{LdB} =- 10log(G * (\frac{h_t*h_r}{d^2})^2)$ to calculate path loss. Figure 7 shows this equation implemented into the program. Figure 8 shows the input and outputs for this model. The program takes in the antenna gain, the heights of the antennas, and the distance between the two and outputs the path loss in dB. There is also a figure and a description accompanying the program to explain its functionality and each parameter. The outputs for each example result in the expected results as, $P_{LdB} =- 10log(1 * (\frac{10*5}{100^2})^2) = 46.02dB$ and $P_{LdB} =- 10log(1 * (\frac{20*10}{50^2})^2) = 21.94dB$ which match the outputs of the program.

```
% Button down function: LeftPanel
function LeftPanelButtonDown(app, event)
    %Pass in Values
    G = app.ApproximateAntennaGainEditField.Value;
    hr = app.RecieverHeightmEditField.Value;
    ht = app.TransmitterHeightmEditField.Value;
    d = app.DistancemEditField_2.Value;
    %Calculate Path Loss
    Pl = d^4/(G * (ht * hr)^2);
    %Convert to dB
    app.PathLossdBEditField.Value = 10*log10(Pl);

end
```

Figure 7. Code For Part III

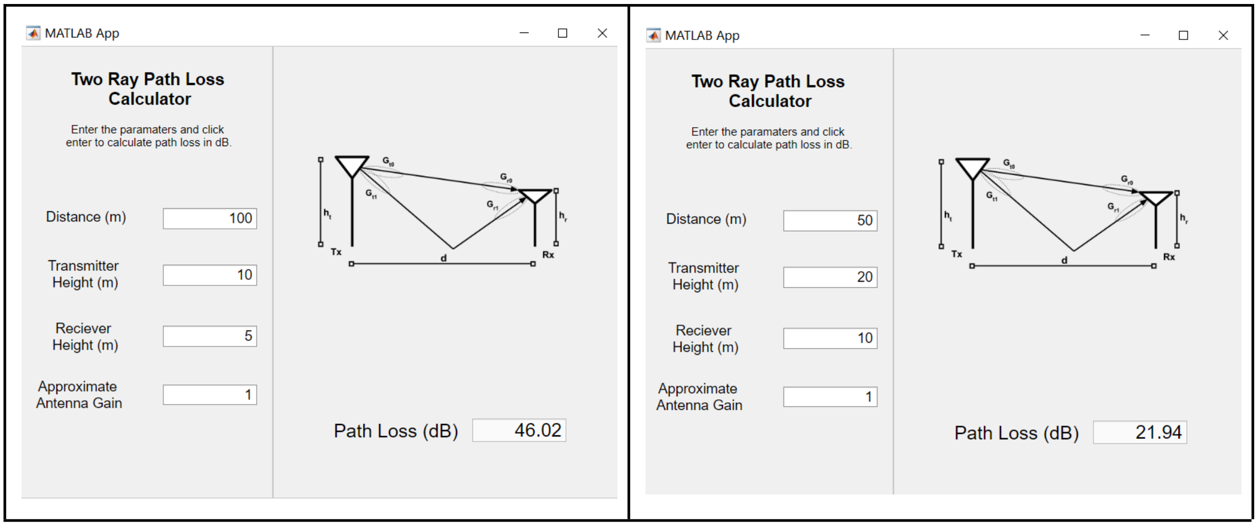

Figure 8. Example I/O For Part III

Conclusion

This project implemented different path loss concepts into a software approach. Modeling these situations helps give an intuition to how different parameters affect path loss. Additionally, these models can be implemented to validate results or to simplify calculations. They can also be used as learning tools for future students.

References

[1] Goldsmith, A. (n.d.). *Wireless Communications* (2nd ed.). pp.51, March 2020.