

Optical Wireless Communication System

Final Assignment - NTUST Short Course

Martin Clinton Manullang (D10902809)

December 16, 2023

This document, including all graphics, code and content can be accessed from

1 Visible Light Communication

In this problem, we want to simulate the 3D received power based on the given condition below (The variable in the MATLAB script denoted in the bracket):

- LED Power (P_{LED}) = 2W
- LED Beam Profile (m_l) = 1 to 20 (This is looped from 1 to 20)
- Receiver Photodetector Size (A_{det}) = $1 * 10^{-4}$
- Field of View of the Receiver (FOV) = 140°
- Room Size / Space Size: width * length * height (l_x, l_y, h) = $2m * 2m * 3m$
- Position of LED at center of room (X_T, Y_T) = 0m, 0m

The above parameters is written down in the MATLAB script as below:

```
1 P_LED = 2; % transmitted optical power by individual LED
2 A_det = 1 * 10 ^ -4; % detector physical area of a PD
3 Ts = 1; % gain of an optical filter
4 index = 1.5; % refractive index of a lens at a PD
5 FOV = 140; % FOV of a receiver
6 G_Con = (index ^ 2) / (sind(FOV) .^ 2); % gain of an optical concentrator
7 lx = 2; ly = 2; h = 3; % room dimensions in meters
8 XT = 0; YT = 0; % Position of the single LED
9 resolution = 20;
10 Nx = lx * resolution; Ny = ly * resolution;
11 x = linspace(-lx / 2, lx / 2, Nx);
12 y = linspace(-ly / 2, ly / 2, Ny);
13 [XR, YR] = meshgrid(x, y);
```

1.1 Plot of 3D received power

Instead of plotting for $m_l = 1$ and $m_l = 20$, I iterate the loop from $m_l = 1$ to $m_l = 20$ to make it easier to answer the next question.

Here's the plot results of $m_l = 1$ and $m_l = 20$ in Watt and dBm:

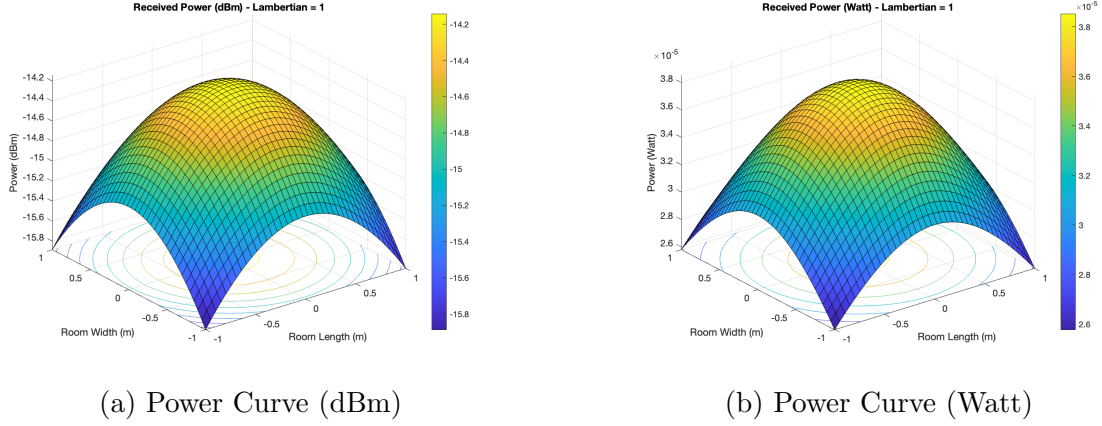


Figure 1: Side-by-side plot of power curves with $m_l = 1$

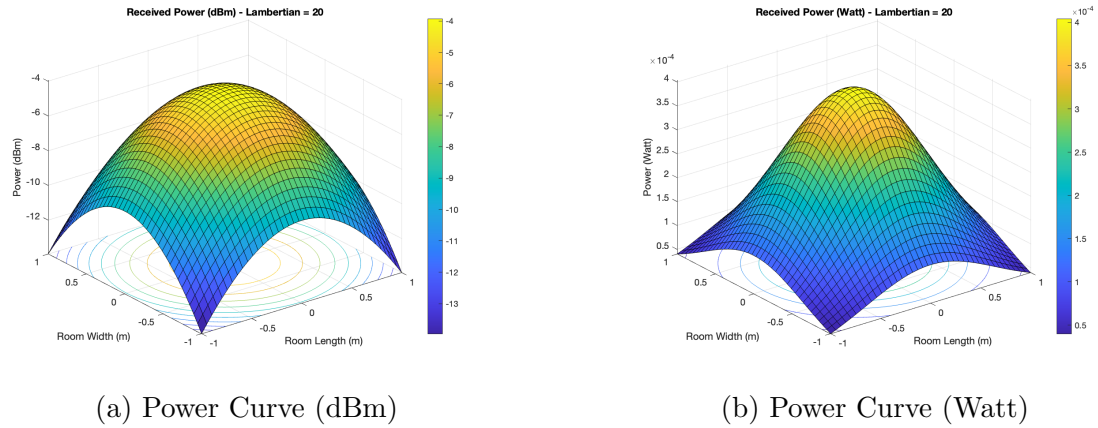


Figure 2: Side-by-side plot of power curves with $m_l = 20$

1.2 Optimum Lambertian Order

To answer this question, we modify our script to store the power at the center and corner for every m_l from 1 to 20 as follows:

```

1   for ml = 1:20 % Lambertian order range
2
3       D = sqrt((XR - XT) .^ 2 + (YR - YT) .^ 2 + h ^ 2); % Distance vector
4       cosphi = h ./ D; % Angle vector
5       receiver_angle = acosd(cosphi);
6       H = (ml + 1) * Adet .* cosphi .^ (ml + 1) ./ (2 * pi .* D .^ 2); % Channel
          DC gain
7       P_rec = P_LED .* H .* Ts .* G_Con; % Received power
8       P_rec(abs(receiver_angle) > FOV) = 0;
9       P_rec_dBm = 10 * log10(P_rec * 1000);
10
11      % Find the received power at the center and one of the corners
12      P_center = P_rec(Ny/2, Nx/2); % Center of the room
13      P_corner = P_rec(1, 1); % Top-left corner of the room
14
15      % Convert powers to dBm for comparison
16      P_center_dBm = 10 * log10(P_center * 1000);
17      P_corner_dBm = 10 * log10(P_corner * 1000);
18
19      % Store the powers for each Lambertian order
20      power_at_center = [power_at_center P_center_dBm];
21      power_at_corner = [power_at_corner P_corner_dBm];
22
23      % Plot in Watt
24      figure(1)
25      surf(x, y, P_rec);
26      xlabel('Room Length (m)');
27      ylabel('Room Width (m)');
28      zlabel('Power (Watt)');
29      title(['Received Power (Watt) - Lambertian = ', num2str(ml)]);
30      colorbar
31      drawnow
32      % Save as PNG and SVG
33      saveas(gcf, ['3D_power_curve_Watt_', num2str(ml), '.png']);
34      saveas(gcf, ['3D_power_curve_Watt_', num2str(ml), '.svg']);
35      % Append to GIF
36      frame = getframe(gcf);
37      im = frame2im(frame);
38      [imind, cm] = rgb2ind(im, 256);
39      if ml == 1
40          imwrite(imind, cm, filenameWatt, 'gif', 'Loopcount', inf, 'DelayTime',
41                  0.5);
42      else
43          imwrite(imind, cm, filenameWatt, 'gif', 'WriteMode', 'append', '
44                  DelayTime', 0.5);
45      end
46
47      % Plot in dBm
48      figure(2)
49      surf(x, y, P_rec_dBm);
50      xlabel('Room Length (m)');
51      ylabel('Room Width (m)');
52      zlabel('Power (dBm)');
53      title(['Received Power (dBm) - Lambertian = ', num2str(ml)]);
54      colorbar
55      drawnow
56      % Save as PNG and SVG

```

```

55     saveas(gcf, ['3D_power_curve_dBm_', num2str(m1), '.png']);
56     saveas(gcf, ['3D_power_curve_dBm_', num2str(m1), '.svg']);
57     % Append to GIF
58     frame = getframe(gcf);
59     im = frame2im(frame);
60     [imind, cm] = rgb2ind(im, 256);
61     if m1 == 1
62         imwrite(imind, cm, filenameDBm, 'gif', 'Loopcount', inf, 'DelayTime',
63             0.5);
64     else
65         imwrite(imind, cm, filenameDBm, 'gif', 'WriteMode', 'append', 'DelayTime',
66             ', 0.5);
67     end
68 end
69
70 % Find the difference that close to 6 dB between the center and the corner
71 power_difference = power_at_center - power_at_corner;
72 % Find the Lambertian order that gives the closest to 6 dB difference
73 [~, index] = min(abs(power_difference - 6));
74
75 fprintf('The Lambertian order that gives the closest to 6 dB difference is %d\n',
76     , index);
77 fprintf('The received power at the center is %f dBm\n', power_at_center(index));
78 fprintf('The received power at the corner is %f dBm\n', power_at_corner(index));
79 fprintf('The difference between the center and the corner is %f dB\n',
80     power_difference(index));

```

At the end of the code, we obtained the result as below:

- The Lambertian order that gives the closest to 6 dB difference is 11
- The received power when $m1 = 11$ at the center is -6.366077 dBm
- The received power when $m1 = 11$ at the corner is -12.462148 dBm
- The difference when $m1 = 11$ between the center and the corner is 6.096071 dB

I also tried to animate the changes and save it as .gif files that can be seen on [this link dbm — Watt](#).

2 Free Space Optical Wireless Communications

2.1 The link budget in clear weather condition

From the question, we obtain some value such below:

- $L = 1.2 \text{ km} = 1200 \text{ m}$
- $\gamma_t(\lambda) = 0.4 \text{ km}^{-1}$

- $\alpha = 0.2^\circ = 0.0035 \text{ rad}$
- $P_t = 40 \text{ mW} = 16.0206 \text{ dBm}$
- $r = 30 \text{ cm} = 0.3 \text{ m}$
- $rec_{sens} = -30 \text{ dBm}$

First, we calculate the atmospheric Loss as below:

$$\text{atm_loss} = \exp(-0.4 \text{ km}^{-1} \times 1.2 \text{ km}) = 0.6188 \text{ mW} = -2.0846 \text{ dB} \quad (1)$$

Next, we calculate the geometry loss. Before that, we need to get the coverage area and the collection area.

$$\text{coverage_area} = \pi \times (1200 \times \tan\left(\frac{0.0035}{2}\right))^2 = 13.7806 \quad (2)$$

$$\text{collection_area} = \pi \times (0.3)^2 = 0.2827 \quad (3)$$

Then, we can calculate the geometry loss as below:

$$\text{geometry_loss} = \frac{\text{collection_area}}{\text{coverage_area}} = 0.0205 \text{ mW} = -16.8788 \text{ dB} \quad (4)$$

Finally, we have all of the value to calculate the link budget. Which are:

- $P_t = 40 \text{ mW} = 16.0206 \text{ dBm}$
- $\text{atm_loss} = 0.6188 \text{ mW} = -2.0846 \text{ dB}$
- $\text{geometry_loss} = 0.0205 \text{ mW} = -16.8788 \text{ dB}$
- $rec_{sens} = -30 \text{ dBm}$

$$\begin{aligned} \text{link_budget} &= P_t + \text{atm_loss} + \text{geometry_loss} - rec_{sens} \\ &= 16.0206 \text{ dBm} + (-2.0846 \text{ dB}) + (-16.8788 \text{ dB}) - (-30 \text{ dBm}) \\ &= 27.06 \text{ dB} \end{aligned} \quad (5)$$

2.2 The link budget in foggy weather condition

Given the Figure 2, we obtained the moderate fog visibility 500m is having $\gamma_t(\lambda) = 28.9\text{db/Km}$. By using that, we can calculate the fog loss as below:

$$\text{fog_loss} = 28.9 \times 1.2 = 34.68 \text{ dB} \quad (6)$$

Finally, we can calculate the available link power on foggy weather condition as follows:

$$\begin{aligned} \text{available_link_power} &= \text{link_budget} - \text{fog_loss} \\ &= 27.06 \text{ dB} - 34.68 \text{ dB} \\ &= -7.62 \text{ dB} \end{aligned} \quad (7)$$

We can conclude that the available link power on foggy weather condition is -7.62 dB which is insufficient to support the link.

Appendix A - First Matlab Code

```

1  clear all; close all; clc;
2
3  % Constants
4  P_LED = 2; % transmitted optical power by individual LED
5  Adet = 1 * 10 ^ -4; % detector physical area of a PD
6  Ts = 1; % gain of an optical filter
7  index = 1.5; % refractive index of a lens at a PD
8  FOV = 140; % FOV of a receiver
9  G_Con = (index ^ 2) / (sind(FOV) .^ 2); % gain of an optical concentrator
10 lx = 2; ly = 2; h = 3; % room dimensions in meters
11 XT = 0; YT = 0; % Position of the single LED
12 resolution = 20;
13 Nx = lx * resolution; Ny = ly * resolution;
14 x = linspace(-lx / 2, lx / 2, Nx);
15 y = linspace(-ly / 2, ly / 2, Ny);
16 [XR, YR] = meshgrid(x, y);
17
18 % Prepare for animation
19 filenameWatt = '3D_power_curve_Watt.gif';
20 filenamedBm = '3D_power_curve_dBm.gif';
21
22 power_at_corner = [];
23 power_at_center = [];
24
25 for ml = 1:20 % Lambertian order range
26
27     D = sqrt((XR - XT) .^ 2 + (YR - YT) .^ 2 + h ^ 2); % Distance vector
28     cosphi = h ./ D; % Angle vector
29     receiver_angle = acosd(cosphi);
30     H = (ml + 1) * Adet .* cosphi .^ (ml + 1) ./ (2 * pi .* D .^ 2); % Channel DC
        gain
31     P_rec = P_LED .* H .* Ts .* G_Con; % Received power
32     P_rec(abs(receiver_angle) > FOV) = 0;
33     P_rec_dBm = 10 * log10(P_rec * 1000);
34
35     % Find the received power at the center and one of the corners
36     P_center = P_rec(Ny/2, Nx/2); % Center of the room
37     P_corner = P_rec(1, 1); % Top-left corner of the room
38
39     % Convert powers to dBm for comparison
40     P_center_dBm = 10 * log10(P_center * 1000);
41     P_corner_dBm = 10 * log10(P_corner * 1000);
42
43     % Store the powers for each Lambertian order
44     power_at_center = [power_at_center P_center_dBm];
45     power_at_corner = [power_at_corner P_corner_dBm];
46
47     % Plot in Watt
48     figure(1)
49     surf(x, y, P_rec);
50     xlabel('Room Length (m)');
51     ylabel('Room Width (m)');
52     zlabel('Power (Watt)');
53     title(['Received Power (Watt) - Lambertian = ', num2str(ml)]);
54     colorbar

```

```

55     drawnow
56     % Save as PNG and SVG
57     saveas(gcf, ['3D_power_curve_Watt_', num2str(m1), '.png']);
58     saveas(gcf, ['3D_power_curve_Watt_', num2str(m1), '.svg']);
59     % Append to GIF
60     frame = getframe(gcf);
61     im = frame2im(frame);
62     [imind, cm] = rgb2ind(im, 256);
63     if m1 == 1
64         imwrite(imind, cm, filenameWatt, 'gif', 'Loopcount', inf, 'DelayTime', 0.5);
65     else
66         imwrite(imind, cm, filenameWatt, 'gif', 'WriteMode', 'append', 'DelayTime',
67             0.5);
68     end
69     % Plot in dBm
70     figure(2)
71     surf(x, y, P_rec_dBm);
72     xlabel('Room Length (m)');
73     ylabel('Room Width (m)');
74     zlabel('Power (dBm)');
75     title(['Received Power (dBm) - Lambertian = ', num2str(m1)]);
76     colorbar
77     drawnow
78     % Save as PNG and SVG
79     saveas(gcf, ['3D_power_curve_dBm_', num2str(m1), '.png']);
80     saveas(gcf, ['3D_power_curve_dBm_', num2str(m1), '.svg']);
81     % Append to GIF
82     frame = getframe(gcf);
83     im = frame2im(frame);
84     [imind, cm] = rgb2ind(im, 256);
85     if m1 == 1
86         imwrite(imind, cm, filenameDBm, 'gif', 'Loopcount', inf, 'DelayTime', 0.5);
87     else
88         imwrite(imind, cm, filenameDBm, 'gif', 'WriteMode', 'append', 'DelayTime',
89             0.5);
90     end
91 end
92 % Find the difference that close to 6 dB between the center and the corner
93 power_difference = power_at_center - power_at_corner;
94 % Find the Lambertian order that gives the closest to 6 dB difference
95 [~, index] = min(abs(power_difference - 6));
96
97 fprintf('The Lambertian order that gives the closest to 6 dB difference is %d\n',
98     index);
99 fprintf('The received power at the center is %f dBm\n', power_at_center(index));
100 fprintf('The received power at the corner is %f dBm\n', power_at_corner(index));
101 fprintf('The difference between the center and the corner is %f dB\n',
102     power_difference(index));

```

Appendix B - Second Matlab Code

```

1 clear all; clc; close all;

```



```

2
3 % A line of sight free space optical communication system has a link
4 % with a range (L) of 1.2 km
5
6 L_in_km = 1.2; % in km
7 L_in_m = L_in_km * 1000; % in m
8
9 % The system is used in both clear weather and fog condition.
10 % In clear weather the attenuation coefficient of the atmospheric channel
11 %  $\gamma_t(\lambda)$  is 0.4 km-1
12
13 gamma_t = 0.4; % in km-1
14
15 % Figure 2 shows the international visibility specification under different fog
16 % condition
17 % The system includes a transmitter using a laser having a very
18 % narrow full-angle beam divergence  $\alpha$  of 0.2 degree
19
20 angle = 0.2; % in degree
21 angle_rad = deg2rad(angle); % in radian
22
23 % and emitted power  $P_t$  of 40 mW
24
25 P_t = 40; % in mW
26 P_t_dbm = 10 * log10(P_t); % in dBm
27
28 % At the receiver end a concentration lens with a radius r = 30 cm
29
30 r = 30; % in cm
31 r_in_meter = r / 100; % in meter
32
33 % is used to focus the light to the photodetector.
34 % The receiver has a sensitivity of -30 dBm
35
36 rec_sensitivity = -30; % in dBm
37
38 % We need to investigate the link gain and loss under both clear and foggy weather
39 % conditions.
40 % Please work out the following questions
41
42 % 1. In clear weather condition (no fog), determine the link budget (20%)
43
44 % Link Budget = Gain - Loss
45 % Link Budget = emitted power - recieved sensitivity - atmospheric loss - beam
46 % divergence loss
47
48 % STEP 1: Calculate the Atmospheric Loss
49 atm_loss = exp(-gamma_t * L_in_km); % in mW
50 atm_loss_db = 10 * log10(atm_loss); % in dB
51
52 % STEP 2: Calculate the Beam Divergence Loss / Geometry Loss
53
54 area_coverage = pi * (L_in_m * tan(angle_rad / 2))^2; % in m^2
55 area_collection = pi * (r_in_meter)^2;
56 geometry_path_loss = area_collection / area_coverage;
57 geometry_path_loss_dbm = 10 * log10(geometry_path_loss); % in dB
58
59 % STEP 3: Calculate the Link Budget

```

```

57 total_gain = P_t_dbm - rec_sensitivity;
58 total_loss = atm_loss_db + geometry_path_loss_dbm;
59
60 link_budget = total_gain + total_loss; % Total Loss is negative
61
62 % print the detail of every component of link budget
63 fprintf('Power Transmitted (dBm) = %.2f dBm\n', P_t_dbm);
64 fprintf('Recieved Sensitivity = %.2f dBm\n', rec_sensitivity);
65 fprintf('Atmospheric Loss = %.2f dB\n', atm_loss_db);
66 fprintf('Geometry Path Loss = %.2f dB\n', geometry_path_loss_dbm);
67 fprintf('Total Gain = %.2f dB\n', total_gain);
68 fprintf('Total Loss = %.2f dB\n', total_loss);
69 fprintf('Link Budget in Clear Weather Condition = %.2f dB\n', link_budget);
70
71 % NEXT QUESTION
72 % Using Figure 2, calculate the atmospheric power loss under moderate fog.
73 % Comment on whether the available link power margin is sufficient when we have
    moderate fog in the channel?
74
75 % Moderate Fog
76 % Visibility: 500 m
77 % Attenuation Coefficient: 28.9 dB/Km
78
79
80 att_coeff = 28.9; % in dB/km
81 fog_loss = att_coeff * L_in_km; % in dB
82
83 fprintf('\n\n');
84 fprintf('Atmospheric Loss in Moderate Fog = %.2f dB\n', fog_loss);
85 fprintf('Link Budget in Moderate Fog = %.2f dB\n', link_budget - fog_loss);
86
87 fprintf('Is it sufficient? ');
88 if (link_budget - fog_loss) > 0
89     fprintf('Yes\n');
90 else
91     fprintf('No\n');
92 end

```