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 receieved 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 (`ml`) = 1 to 20 (This is looped from 1 to 20)
- Receiver Photodetector Size (`Adet`) = $1 * 10^-4$
- Field of View of the Receiver (`FOV`) = $140°$
- Room Size / Space Size: width * length * height (`lx, ly, h`) = $2\text{m} * 2\text{m} * 3\text{m}$
- Position of LED at center of room (`XT, YT`) = $0\text{m}, 0\text{m}$

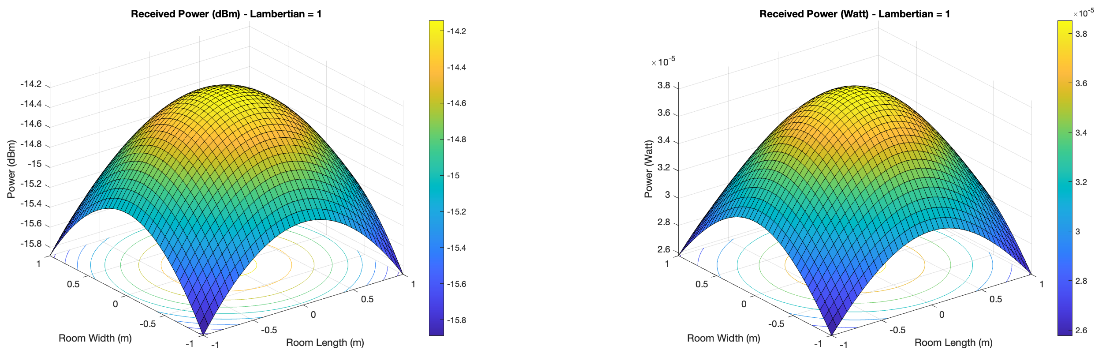
The above parameters is written down in the MATLAB script as below:

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

1.1 Plot of 3D receieved power

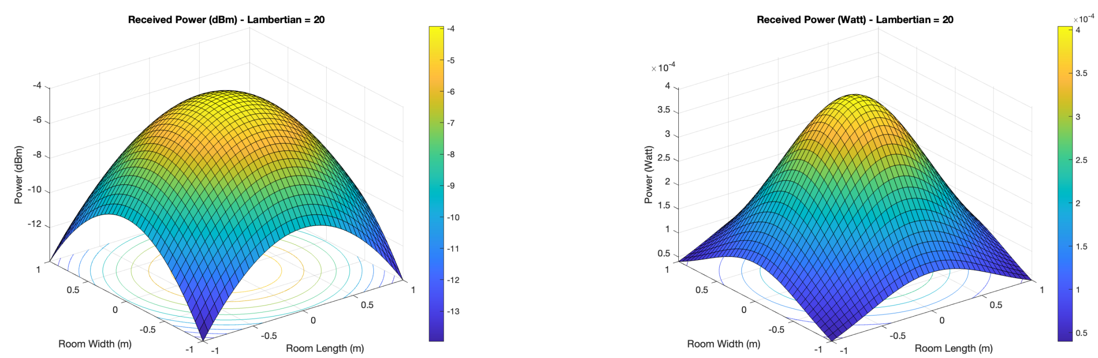
Instead of plotting for `ml = 1` and `ml = 20`, I iterate the loop from `ml = 1` to `ml = 20` to make it easier to answer the next question.

Here's the plot results of `ml = 1` and `ml = 20` in Watt and dBm:



(a) Power Curve (dBm) (b) Power Curve (Watt)

Figure 1: Side-by-side plot of power curves with `ml = 1`



(a) Power Curve (dBm) (b) Power Curve (Watt)

Figure 2: Side-by-side plot of power curves with `ml = 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 `ml` from 1 to 20 as follows:

```matlab
    for ml = 1:20 % Lambertian order range

        D = sqrt((XR - XT) .^ 2 + (YR - YT) .^ 2 + h ^ 2); % Distance vector
        cosphi = h ./ D; % Angle vector
        receiver_angle = acosd(cosphi);
        H = (ml + 1) * Adet .* cosphi .^ (ml + 1) ./ (2 * pi .* D .^ 2); % Channel DC gain
        P_rec = P_LED .* H .* Ts .* G_Con; % Received power
        P_rec(abs(receiver_angle) > FOV) = 0;
        P_rec_dBm = 10 * log10(P_rec * 1000);

        % Find the received power at the center and one of the corners
        P_center = P_rec(Ny/2, Nx/2); % Center of the room
        P_corner = P_rec(1, 1); % Top-left corner of the room

        % Convert powers to dBm for comparison
        P_center_dBm = 10 * log10(P_center * 1000);
        P_corner_dBm = 10 * log10(P_corner * 1000);

        % Store the powers for each Lambertian order
        power_at_center = [power_at_center P_center_dBm];
        power_at_corner = [power_at_corner P_corner_dBm];

        % Plot in Watt
        figure(1)
        surfc(x, y, P_rec);
        xlabel('Room Length (m)');
        ylabel('Room Width (m)');
        zlabel('Power (Watt)');
        title(['Received Power (Watt) - Lambertian = ', num2str(ml)]);
        colorbar
        drawnow
        % Save as PNG and SVG
        saveas(gcf, ['3D_power_curve_Watt_', num2str(ml), '.png']);
        saveas(gcf, ['3D_power_curve_Watt_', num2str(ml), '.svg']);
        % Append to GIF
        frame = getframe(gcf);
        im = frame2im(frame);
        [imind, cm] = rgb2ind(im, 256);
        if ml == 1
            imwrite(imind, cm, filenameWatt, 'gif', 'Loopcount', inf, 'DelayTime', 0.5);
        else
            imwrite(imind, cm, filenameWatt, 'gif', 'WriteMode', 'append', 'DelayTime', 0.5);
        end

        % Plot in dBm
        figure(2)
        surfc(x, y, P_rec_dBm);
        xlabel('Room Length (m)');
        ylabel('Room Width (m)');
        zlabel('Power (dBm)');
        title(['Received Power (dBm) - Lambertian = ', num2str(ml)]);
        colorbar
        drawnow
        % Save as PNG and SVG
```

```
        saveas(gcf, ['3D_power_curve_dBm_', num2str(ml), '.png']);
        saveas(gcf, ['3D_power_curve_dBm_', num2str(ml), '.svg']);
        % Append to GIF
        frame = getframe(gcf);
        im = frame2im(frame);
        [imind, cm] = rgb2ind(im, 256);
        if ml == 1
            imwrite(imind, cm, filenameDBm, 'gif', 'Loopcount', inf, 'DelayTime', 0.5);
        else
            imwrite(imind, cm, filenameDBm, 'gif', 'WriteMode', 'append', 'DelayTime', 0.5);
        end
    end

    % Find the difference that close to 6 dB between the center and the corner
    power_difference = power_at_center - power_at_corner;
    % Find the Lambertian order that gives the closest to 6 dB difference
    [~, index] = min(abs(power_difference - 6));

    fprintf('The Lambertian order that gives the closest to 6 dB difference is %d\n', index);
    fprintf('The received power at the center is %f dBm\n', power_at_center(index));
    fprintf('The received power at the corner is %f dBm\n', power_at_corner(index));
    fprintf('The difference between the center and the corner is %f dB\n', 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 `ml = 11` at the center is -6.366077 dBm
- The received power when `ml = 11` at the corner is -12.462148 dBm
- The difference when `ml = 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:

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

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

First, we calculate the atmospheric Loss as below:

$$\text{atm\_loss} = \exp\big(-0.4\,\mathrm{km}^{-1} \times 1.2\,\mathrm{km}\big) = 0.6188\,\mathrm{mW} = -2.0846\,\mathrm{dB} \tag{1}$$

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

$$\text{coverage\_area} = \pi \times (1200 \times \tan\left(\frac{0.0035}{2}\right)^2) = 13.7806 \tag{2}$$

$$\text{collection\_area} = \pi \times (0.3)^2 = 0.2827 \tag{3}$$

Then, we can calculate the geometry loss as below:

$$\text{geometry\_loss} = \frac{\text{collection\_area}}{\text{coverage\_area}} = 0.0205\,\mathrm{mW} = -16.8788\mathrm{dB} \tag{4}$$

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

- $P_t = 40\,\mathrm{mW} = 16.0206\,\mathrm{dBm}$
- $\text{atm\_loss} = 0.6188\,\mathrm{mW} = -2.0846\,\mathrm{dB}$
- $\text{geometry\_loss} = 0.0205\,\mathrm{mW} = -16.8788\mathrm{dB}$
- $rec_{sens} = -30\,\mathrm{dBm}$

$$\begin{aligned}\text{link\_budget} &= P_t + \text{atm\_loss} + \text{geometry\_loss} - rec_{sens} \\ &= 16.0206\,\mathrm{dBm} + (-2.0846\,\mathrm{dB}) + (-16.8788\mathrm{dB}) - (-30\,\mathrm{dBm}) \\ &= 27.06\,\mathrm{dB}\end{aligned} \tag{5}$$

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:

$$\text{fog\_loss} = 28.9 \times 1.2 = 34.68\,\text{dB} \tag{6}$$

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

$$\begin{aligned} \text{available\_link\_power} &= \text{link\_budget} - \text{fog\_loss} \\ &= 27.06\,\text{dB} - 34.68\,\text{dB} \\ &= -7.62\,\text{dB} \end{aligned} \tag{7}$$

We can conclude that the available link power on foggy weather condition is $-7.62\,\text{dB}$ which is insufficient to support the link.

Appendix A - First Matlab Code

```
clear all; close all; clc;

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

% Prepare for animation
filenameWatt = '3D_power_curve_Watt.gif';
filenameDBm = '3D_power_curve_dBm.gif';

power_at_corner = [];
power_at_center = [];

for ml = 1:20 % Lambertian order range

    D = sqrt((XR - XT) .^ 2 + (YR - YT) .^ 2 + h ^ 2); % Distance vector
    cosphi = h ./ D; % Angle vector
    receiver_angle = acosd(cosphi);
    H = (ml + 1) * Adet .* cosphi .^ (ml + 1) ./ (2 * pi .* D .^ 2); % Channel DC
        gain
    P_rec = P_LED .* H .* Ts .* G_Con; % Received power
    P_rec(abs(receiver_angle) > FOV) = 0;
    P_rec_dBm = 10 * log10(P_rec * 1000);

    % Find the received power at the center and one of the corners
    P_center = P_rec(Ny/2, Nx/2); % Center of the room
    P_corner = P_rec(1, 1); % Top-left corner of the room

    % Convert powers to dBm for comparison
    P_center_dBm = 10 * log10(P_center * 1000);
    P_corner_dBm = 10 * log10(P_corner * 1000);

    % Store the powers for each Lambertian order
    power_at_center = [power_at_center P_center_dBm];
    power_at_corner = [power_at_corner P_corner_dBm];

    % Plot in Watt
    figure(1)
    surfc(x, y, P_rec);
    xlabel('Room Length (m)');
    ylabel('Room Width (m)');
    zlabel('Power (Watt)');
    title(['Received Power (Watt) - Lambertian = ', num2str(ml)]);
    colorbar
```

```
    drawnow
    % Save as PNG and SVG
    saveas(gcf, ['3D_power_curve_Watt_', num2str(ml), '.png']);
    saveas(gcf, ['3D_power_curve_Watt_', num2str(ml), '.svg']);
    % Append to GIF
    frame = getframe(gcf);
    im = frame2im(frame);
    [imind, cm] = rgb2ind(im, 256);
    if ml == 1
        imwrite(imind, cm, filenameWatt, 'gif', 'Loopcount', inf, 'DelayTime', 0.5);
    else
        imwrite(imind, cm, filenameWatt, 'gif', 'WriteMode', 'append', 'DelayTime',
            0.5);
    end

    % Plot in dBm
    figure(2)
    surfc(x, y, P_rec_dBm);
    xlabel('Room Length (m)');
    ylabel('Room Width (m)');
    zlabel('Power (dBm)');
    title(['Received Power (dBm) - Lambertian = ', num2str(ml)]);
    colorbar
    drawnow
    % Save as PNG and SVG
    saveas(gcf, ['3D_power_curve_dBm_', num2str(ml), '.png']);
    saveas(gcf, ['3D_power_curve_dBm_', num2str(ml), '.svg']);
    % Append to GIF
    frame = getframe(gcf);
    im = frame2im(frame);
    [imind, cm] = rgb2ind(im, 256);
    if ml == 1
        imwrite(imind, cm, filenameDBm, 'gif', 'Loopcount', inf, 'DelayTime', 0.5);
    else
        imwrite(imind, cm, filenameDBm, 'gif', 'WriteMode', 'append', 'DelayTime',
            0.5);
    end
end

% Find the difference that close to 6 dB between the center and the corner
power_difference = power_at_center - power_at_corner;
% Find the Lambertian order that gives the closest to 6 dB difference
[~, index] = min(abs(power_difference - 6));

fprintf('The Lambertian order that gives the closest to 6 dB difference is %d\n',
    index);
fprintf('The received power at the center is %f dBm\n', power_at_center(index));
fprintf('The received power at the corner is %f dBm\n', power_at_corner(index));
fprintf('The difference between the center and the corner is %f dB\n',
    power_difference(index));
```

Appendix B - Second Matlab Code

```
clear all; clc; close all;
```

```

% A line of sight free space optical communication system has a link
% with a range (L) of 1.2 km

L_in_km = 1.2; % in km
L_in_m = L_in_km * 1000; % in m

% The system is used in both clear weather and fog condition.
% In clear weather the attenuation coefficient of the atmospheric channel
% \gamma_{t}(\lambda) is 0.4 km^{-1}

gamma_t = 0.4; % in km^{-1}

% Figure 2 shows the international visibility specification under different fog condition
% The system includes a transmitter using a laser having a very
% narrow full-angle beam divergence \alpha of 0.2 degree

angle = 0.2; % in degree
angle_rad = deg2rad(angle); % in radian

% and emitted power P_{t} of 40 mW

P_t = 40; % in mW
P_t_dbm = 10 * log10(P_t); % in dBm

% At the receiver end a concentration lens with a radius r = 30 cm

r = 30; % in cm
r_in_meter = r / 100; % in meter

% is used to focus the light to the photodetector.
% The receiver has a sensitivity of -30 dBm

rec_sensitivity = -30; % in dBm

% We need to investigate the link gain and loss under both clear and foggy weather conditions.
% Please work out the following questions

% 1. In clear weather condition (no fog), determine the link budget (20%)

% Link Budget = Gain - Loss
% Link Budget = emitted power - recieved sensitivity - atmospheric loss - beam divergence loss

% STEP 1: Calculate the Atmospheric Loss
atm_loss = exp(-gamma_t * L_in_km); % in mW
atm_loss_db = 10 * log10(atm_loss); % in dB

% STEP 2: Calculate the Beam Divergence Loss / Geometry Loss

area_coverage = pi * (L_in_m * tan(angle_rad / 2))^2; % in m^2
area_collection = pi * (r_in_meter)^2;
geometry_path_loss = area_collection / area_coverage;
geometry_path_loss_dbm = 10 * log10(geometry_path_loss); % in dB

% STEP 3: Calculate the Link Budget
```

```
total_gain = P_t_dbm - rec_sensitivity;
total_loss = atm_loss_db + geometry_path_loss_dbm;

link_budget = total_gain + total_loss; % Total Loss is negative

% print the detail of every component of link budget
fprintf('Power Transmitted (dBm) = %.2f dBm\n', P_t_dbm);
fprintf('Recieved Sensitivity = %.2f dBm\n', rec_sensitivity);
fprintf('Atmospheric Loss = %.2f dB\n', atm_loss_db);
fprintf('Geometry Path Loss = %.2f dB\n', geometry_path_loss_dbm);
fprintf('Total Gain = %.2f dB\n', total_gain);
fprintf('Total Loss = %.2f dB\n', total_loss);
fprintf('Link Budget in Clear Weather Condition = %.2f dB\n', link_budget);

% NEXT QUESTION
% Using Figure 2, calculate the atmospheric power loss under moderate fog.
% Comment on whether the available link power margin is sufficient when we have
    moderate fog in the channel?

% Moderate Fog
% Visibility: 500 m
% Attenuation Coefficient: 28.9 dB/Km


att_coeff = 28.9; % in dB/km
fog_loss = att_coeff * L_in_km; % in dB

fprintf('\n\n');
fprintf('Atmospheric Loss in Moderate Fog = %.2f dB\n', fog_loss);
fprintf('Link Budget in Moderate Fog = %.2f dB\n', link_budget - fog_loss);

fprintf('Is it sufficient? ');
if (link_budget - fog_loss) > 0
    fprintf('Yes\n');
else
    fprintf('No\n');
end
```