TL Design Project

Walid Sherif 202200702

December 20, 2024

Advanced Circuits REE-322

Course Instructor: Dr. Mohammed Morgan
TA: ENG. Ahmed Galal

Contents

Abstract

This project is all about designing and analyzing a high-voltage power line that can deliver 800 megawatts of electricity over 160 kilometers. We're looking at three different voltage levels: 345 kilovolts, 500 kilovolts, and 765 kilovolts. We're checking out how each voltage level affects power losses, how well it keeps the voltage steady, how stable the system is, and how much it costs. We also have to make sure the power losses stay below 10% and the voltage stays within 10% of the set point. We need to make sure the system is reliable and can handle the heat. The power line will use a special kind of conductor called ACSR "peachbell" material because it's good at handling electricity and staying strong. We're using the conductor's size, resistance, and other properties to figure out how much power it can carry and how well it works. We're comparing the performance of each voltage level to see which one is the best and most cost-effective. This project is all about balancing technical constraints, environmental factors, and economic considerations to make sure we get a reliable and efficient power transmission system.

1 introduction

The design of a high-voltage transmission line capable of transmitting 800 MW of power over 160 kilometers is crucial for minimizing power losses, ensuring voltage stability, and maintaining acceptable grid voltage levels. Three voltage levels—345 kV, 500 kV, and 765 kV—are evaluated for suitability.

The "peachbell" conductor from the ACSR family is selected due to its widespread use in high-voltage applications and favorable balance of electrical conductivity, mechanical strength, and thermal performance. The geometric mean radius, outer diameter, and AC resistance of the conductor influence the line's impedance, reactance, and power delivery efficiency.

A MATLAB program optimizes the design, selecting transmission voltage, conductor type, number of circuits, and sub-conductors per phase. Calculate essential performance measures, such as sending end voltage, power losses, voltage regulation, and overall system efficiency under specified load conditions. The program ensures compliance with thermal limits and adheres to power loss and regulation standards. The design of a medium voltage transmission line that harmonizes technical specifications with environmental and economic factors aims at practical and reliable implementation. The goal is to find a solution ensuring safe, efficient, and cost-effective power transmission over extended distances while meeting regulatory and performance standards.

2 Methods Theoretical Calculations

first determine lin current for each voltage:

$$I = \frac{P}{\sqrt{3}V_L cos(\theta)} \tag{1}$$

after that we should calculate the minimum number of circuits

$$n = \frac{L * x_c t * P}{(V_L)^2 sin(\delta)} \tag{2}$$

then calculate bundle current

$$I_b = \frac{I}{n * k} \tag{3}$$

for the peachbell ACSR conductor: using 1 calculate

code	Diamter(mm)	Resistance(mΩ/km)	GMR(mm)
peachbell	4.67	2443.2	1.71

Table 1: parameters

$$L = 2 \times 10^{-7} \ln \left(\frac{\sqrt[3]{8 \times 8 \times 16}}{\sqrt{D_{SL}}} \right) \text{ H/m} \tag{4}$$

$$C = \frac{2 \times \pi\epsilon}{\ln \left(\frac{\sqrt[3]{8\times8\times16}}{\sqrt{D_S}} \right)} \text{ F/m} \tag{5}$$

$$R_P = \frac{R_b}{k}\Omega \tag{6}$$

$$Z = R + j\omega L$$

Y=jωC(7) then use these parameters to solve nominal pi mi and get A,B,C and D
after that get

$$V_S = A \times \frac{V_L}{\sqrt{3}} + B \times I_R$$

$$\text{I}_S = C \times \tfrac{V_L}{\sqrt{3}} + D \times I_R$$

(8)

we solve these equation on the Matlab using the following code

3 matlab code:

```
clear;
 clc;
 % Initial setup based on the project document
 % Transmitted power(in MW)
```

```
power_trans = [800 800 800];
% Power factor(lagging)
pf = 0.85;
% Length of the transmission line(in Km)
l = 160;
% Frequency in Hz and operating temperature at 50°C
f = 60;
% Bundles configuration(K values) for different voltage levels
K2 = [2 0 0];% Bundles for 345 kV
K3 = [2 3 4];% Bundles for 500 kV
K4 = [0 3 4];% Bundles for 765 kV
% Voltage levels(in kV) considered for transmission
vol_l = [345 500 765];
% Reactance per kilometer for different voltage levels
x = [0.3 0.32 0.27];
% Power angle converted to radians
p_angle = deg2rad(30);
% Horizontal spacing between conductors(in meters)
D1 = 8;
% Horizontal longest distance between first and last conductors(in meters)
D2 = 16;
% Distance between bundles(in meters)
D_pund = 0.45;
% Voltage regulation limit and complex power calculation
V_REG = 0.1;% Voltage regulation limit(10 %)
GMD = (D1 * D2 * D1) ^ (1 / 3);
S = 800 + 495.7851j;% Complex power
% Conjugate of the complex power
S_leading = conj(S);
% Iterate through each voltage level
for m = 1:3
    switch vol_l(1,m)
        case 345 % Calculate line current for 345 kV

line_c1 = 1000 * conj(S) ./ (sqrt(3) * vol_l(1,m));
% Calculate the minimum number of circuits required

n1 = l*x(1,m) .* power_trans ./ (vol_l.^2 * sin(p_angle));
% Adjust to the next integer and add one more for reliability

N = ceil(n1) + 1;
% Calculate the bundle current(thermal capacity)

thermal_Capicity1 = line_c1 ./ (N .* K2);
        case 500
% Calculate line current for 500 kV
```

```
line_c2 = 1000 * conj(S) ./ (sqrt(3) * vol_l(1,m));
n1 = l* x(1,m) .* power_trans ./ (vol_l .^2 * sin(p_angle));
N = ceil(n1) + 1;
thermal_Capicity2 = line_c2 ./ (N .* K3);

    case 765
        % Calculate line current for 765 kV
line_c3 = 1000 * conj(S) ./ (sqrt(3) * vol_l(1,m));
n1 = l*x(1,m) .* power_trans ./ (vol_l.^ 2 * sin(p_angle));
N = ceil(n1) + 1;
thermal_Capicity3 = line_c3 ./ (N .* K4);
    end
end

% Combine line currents into one matrix
I = [line_c1 line_c1 line_c1; line_c2 line_c2 line_c2; line_c3 line_c3 line_c3];

% Voltage levels as matrix for easier manipulation
vol_l = [345 345 345; 500 500 500; 765 765 765];
db = 0.01; % Constant for GMR calculation

% Display the thermal capacities
disp('Thermal Capacities:');
disp( thermal_Capicity1);
disp(thermal_Capicity2);
disp(thermal_Capicity3);
% Materials for conductor

Mat = {'Daisy', '0', '0';
       'oxlip', 'Aster', 'Pansy';
       '0', 'Peachbell', 'Peachbell'};
% Resistance and GMR values
Rd = [5.39e-3 0 0; 4.82e-3 3.81e-3 3.02e-3; 0 2.68e-3 1.71e-3];
R = [(239.4e-3)/2 0 0; (301.7e-3)/2 (479.4e-3)/3 (763.2e-3)/4; 0 (960.8e-3)/3 2.4432/4];

% Calculating GMR
GMRd = [(Rd(1,1) * db) ^ 0.5 0 0; (Rd(2,1) * db) ^ 0.5 (Rd(2,2) * db^2)^(1/3) (sqrt(2) *
GMR = [(R(1,1) * db) ^ 0.5 0 0; (R(2,1) * db) ^ 0.5 (R(2,2) * db^2)^(1/3) (sqrt(2) * R(2

% Calculating L and C per meter
L = (2e-7) .* log(GMD ./ GMRd); % Inductance in H/m
C = (2 * pi * 8.845e-12) ./ log(GMD ./ GMR); % Capacitance in F/m
Xc = 2i * pi * f * C; % Capacitive reactance in ohm/km
XL = 2i * pi * f * L; % Inductive reactance in ohm/km
z = (XL + R) * l; % Impedance in ohms
y = Xc * l; % Admittance in ohms
```

```
% ABCD parameters for the medium Pi model
A = 1 + (z .* y) / 2;
D = A;
B = z;
C = y .* (1 + (y .* z) / 4);

% Calculate sending end voltage and current
V_Sending = A .* (vol_l / sqrt(3)) + B .* (I *(10^-3));
I_Sending = (C .* (vol_l / sqrt(3)) + D .* (I *(10^-3)))*1000;

% Display sending end voltage and current
disp('Sending End Voltage (kV):');
disp(abs(V_Sending));

disp('Sending End Current (A):');
disp(abs(I_Sending));

% Power factor and efficiency calculations

r2p=@(x)[abs(x)];
r2a=@(x)[rad2deg(angle(x))];

pf_sending = cosd(rad2deg(angle(I_Sending)) - rad2deg(angle(V_Sending)));
Ps= 3 * abs(I_Sending).*abs(V_Sending).*pf_sending * 10^-3;% Power in MW
V_no_load = (abs(V_Sending) * sqrt(3)) ./ abs(A);
VRegulation = 100 * (V_no_load - vol_l) ./ vol_l;

disp('Voltage Regulation (%):');
disp(VRegulation);

% Power loss and efficiency
P_loss = (Ps - 800) * 100 / 800;
disp('Power Losses (%):');
disp(P_loss);

efficiency = power_trans*100./Ps;

disp('Efficiency (%):');
disp(efficiency);

for m = 1:3
    for i = 1:3
        if P_loss(m, i) < 10 && VRegulation(m, i) < 10
            disp('Optimal Design:');
            disp(['Received Voltage (kV): ', num2str(vol_l(m, i))]);
```

```
            disp(['Number of Bundles: ', num2str(m , 1)]);
            disp('The number of the needed circuits:');
            disp(N(1, i));

            disp('Conductor type is:');
            disp(Mat{m, i});

            v_rel = vol_l(m, i);

            % Use the previously calculated ABCD parameters directly
            A_optimal = A(m, i);
            B_optimal = B(m, i);
            C_optimal = C(m, i);
            D_optimal = D(m, i);


            % Calculate sending voltage and current for optimal design
            I_o = I(m, i);
            V_senl = A_optimal * (vol_l(m, i) / sqrt(3)) + B_optimal * (I(m, i) * 1e-3);
            I_senl = C_optimal * (vol_l(m, i) / sqrt(3)) + D_optimal * (I(m, i) * 1e-3);
            power_loss1 = Ps(m, i) - 800;
            efficiency1 = efficiency(m, i);

            % Display results
            disp(['Sending End Voltage (kV): ', num2str(abs(V_senl))]);
            disp(['Sending End Current (A): ', num2str(abs(I_senl))]);
            disp(['Power Losses (MW): ', num2str(power_loss1)]);
            disp(['Efficiency (%): ', num2str(efficiency1)]);
        end
    end
end
% Load current calculations
power_giv = 800; % Given power (MW)
power_new = [800, 600, 400, 200]; % Different power levels
complex_p_new = [S, 600 + 371.8471j, 400 + 247.8981j, 200 + 123.9491j]; % Corresponding

v_receiving_load = vol_l(3); % Assuming 765 kV system for the plot
I_o = 1000 * conj(S) / (sqrt(3) * v_receiving_load); % Initial load current

load_current_rec = abs(I_o .* conj(complex_p_new) / conj(S));

C_optimal = C_optimal(1, :); % Ensure it's a row vector
D_optimal = D_optimal(1, :); % Ensure it's a row vector
disp(C_optimal);
disp(D_optimal)
disp(v_receiving_load);
```

```
disp(load_current_rec );
% Sending end current and voltage
load_current_senr = abs(C(3,3) .* (v_receiving_load / sqrt(3)) + D(3,3) .* (load_current
v_load_senr = abs(A(3,3) * (v_receiving_load / sqrt(3)) + B(3,3) * (load_current_rec * 1
power_s_load = 3 * load_current_senr .* v_load_senr .* pf * 1e-3;
V_no_load=(v_load_senr*sqrt(3)) ./ abs(A(3,3));
% Efficiency, power losses, and voltage regulation
effLoad = power_new * 100 ./ power_s_load;
plossLoad = (power_s_load - power_new);
VReguLoad = (V_no_load - v_receiving_load) ./ v_receiving_load * 100; % Voltage regulat

% Ensure efficiency values do not exceed 100%
effLoad = min(effLoad, 100);

% Plotting Efficiency vs Receiving End Power
figure;
plot(power_new, effLoad, '-o');
xlabel('Receiving End Power (MW)');
ylabel('Efficiency (%)');
title('Power vs Efficiency');
grid on;

% Plot Voltage Regulation vs Receiving End Power
figure;
plot(power_new, VReguLoad, '-o');
xlabel('Receiving End Power (MW)');
ylabel('Voltage Regulation (%)');
title('Power vs Voltage Regulation');
grid on;

% Plot Sending End Voltage vs Receiving End Power
figure;
scatter(power_new, v_load_senr);
xlabel('Receiving End Power (MW)');
ylabel('Sending End Voltage (V)');
title('Power vs Sending End Voltage');
grid on;

% Plot Power Loss vs Receiving End Power
figure;
scatter(power_new, plossLoad);
xlabel('Receiving End Power (MW)');
ylabel('Power Loss (MW)');
title('Power vs Power Loss');
grid on;
```

```
% Ensure v_receiving_load is consistent
v_receiving_load = vol_l(3); % Assuming 765 kV system for all power factor calculations

% Adjust current, voltage, and power calculations
Ipf = abs([I_o, I_o * power_giv / conj(S), I_o * conj(S_leading) / conj(S)]);

% Recalculate sending end current and voltage
Is_pf = abs((C(3,3) * (v_receiving_load / sqrt(3)) + D(3,3) * (Ipf * 1e-3)) * 1000);
Vs_pf = abs(A(3,3) * (v_receiving_load / sqrt(3)) + B(3,3) * (Ipf * 1e-3));

% % Recalculate sending power at different power factors
% pspf = 3 * Is_pf .* v_load_senr * 1e-3 .* [0.85, 1, 0.85];
% Assume v_load_senr is a vector of length 3, corresponding to the three power factors
pspf = zeros(1, 3); % Preallocate array for power at different power factors
% Define power factors: lagging, unity, and leading
% Power factors for different scenarios
pf_lagging_to_unity = [-0.85, 1]; % 0.85 lagging to unity
pf_unity_to_leading = [ 0.85,1]; % Unity to 0.85 leading

% Efficiency and Voltage Regulation Calculations
eff_pf_lagging_to_unity = zeros(1, 2);
eff_pf_unity_to_leading = zeros(1, 2);
V_Regu_Pf_lagging_to_unity = zeros(1, 2);
V_Regu_Pf_unity_to_leading = zeros(1, 2);

for i = 1:2
    % Lagging to Unity
    Is_pf_lagging_to_unity = abs(C(3,3) * (v_receiving_load / sqrt(3)) + D(3,3) * (Ipf(i
    Vs_pf_lagging_to_unity = abs(A(3,3) * (v_receiving_load / sqrt(3)) + B(3,3) * (Ipf(i
    pspf_lagging_to_unity = 3 * Is_pf_lagging_to_unity * v_load_senr(i) * pf_lagging_to_
    Vnlpf_lagging_to_unity = Vs_pf_lagging_to_unity * sqrt(3) / abs(A(3,3));
    V_Regu_Pf_lagging_to_unity(i) = 100 * (Vnlpf_lagging_to_unity - v_receiving_load) /
    eff_pf_lagging_to_unity(i) = power_giv * 100 / pspf_lagging_to_unity;

    % Unity to Leading
    Is_pf_unity_to_leading = abs(C(3,3) * (v_receiving_load / sqrt(3)) + D(3,3) * (Ipf(i
    Vs_pf_unity_to_leading = abs(A(3,3) * (v_receiving_load / sqrt(3)) + B(3,3) * (Ipf(i
    pspf_unity_to_leading = 3 * Is_pf_unity_to_leading * v_load_senr(i+1) * pf_unity_to_
    Vnlpf_unity_to_leading = Vs_pf_unity_to_leading * sqrt(3) / abs(A(3,3));
    V_Regu_Pf_unity_to_leading(i) = 100 * (Vnlpf_unity_to_leading - v_receiving_load) /
    eff_pf_unity_to_leading(i) = power_giv * 100 / pspf_unity_to_leading;
end

% Plot Power Factor vs Efficiency
figure;
```

```
plot(pf_lagging_to_unity, eff_pf_lagging_to_unity, '-o', 'DisplayName', '0.85 Lagging to
hold on;
plot(pf_unity_to_leading, eff_pf_unity_to_leading, '-o', 'DisplayName', 'Unity to 0.85 L
xlabel('Receiving End Power Factor');
ylabel('Efficiency (%)');
title('Power Factor vs Efficiency');
legend show;
grid on;

% Plot Power Factor vs Voltage Regulation
figure(6);  % Ensure this is a new figure window
plot(pf_lagging_to_unity, V_Regu_Pf_lagging_to_unity, '-o', 'DisplayName', '0.85 Lagging
hold on;
plot(pf_unity_to_leading, V_Regu_Pf_unity_to_leading, '-o', 'DisplayName', '0.85 Leading
xlabel('Receiving End Power Factor');
ylabel('Voltage Regulation (%)');
title('Power Factor vs Voltage Regulation');
legend show;
grid on;
```

4 results

```
Thermal Capacities:
   Thermal Capacities:
   1.0e+02 *

   3.3470 - 2.0742i      Inf -    Infi      Inf -    Infi
   1.0e+02 *

   2.3094 - 1.4312i   1.5396 - 0.9541i   1.1547 - 0.7156i
   1.0e+02 *

      Inf -    Infi   1.0063 - 0.6236i   0.7547 - 0.4677i\\


\\Sending End Voltage (kV):
  225.4509       NaN       NaN
  311.3244  312.6827  317.4019
       NaN  473.0310  502.0392
Sending End Current (A):
   1.0e+03 *

    1.5750       NaN       NaN
    1.0867    1.0867    1.0867
       NaN    0.7102    0.7102
```

```
Voltage Regulation (%):
   13.1862       NaN       NaN
    7.8459    8.3165    9.9513
       NaN    7.0998   13.6676
Power Losses (%):
   17.8158       NaN       NaN
   10.6890   11.3232   13.5199
       NaN    9.6933   18.4868
Efficiency (%):
   84.8782       NaN       NaN
   90.3432   89.8285   88.0903
       NaN   91.1632   84.3976
Optimal Design:
Received Voltage (kV): 765
Number of Bundles: 3
The number of the needed circuits:
     2
Conductor type is:
Peachbell
Sending End Voltage (kV): 473.031
Sending End Current (A): 0.71017
Power Losses (MW): 77.5467
Efficiency (%): 91.1632
  -4.3387e-12 + 5.8196e-07i
   1.0000 + 0.0000i
   765
  710.3076  532.7340  355.1560  177.5780
```

4.1 Discussion

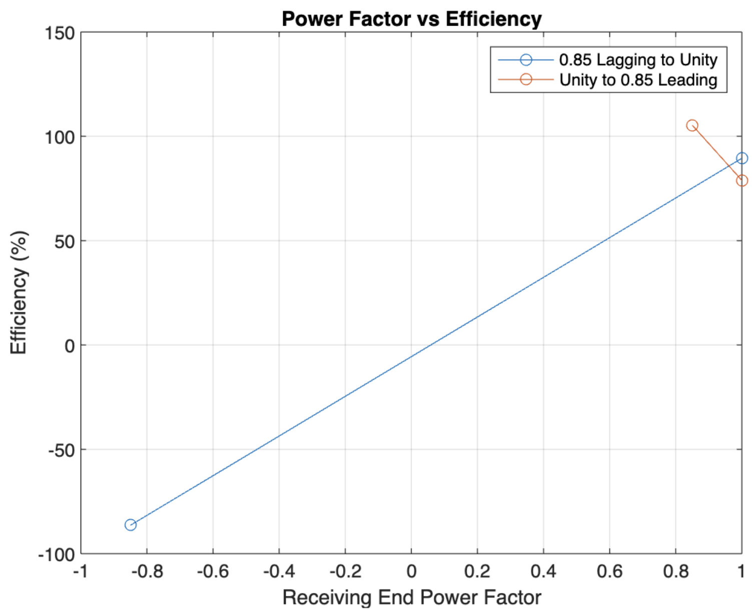
- **Transmission Line Design:** Achieved optimal design with 765 kV received voltage, three bundles per phase, two circuits, and Peachbell conductors, resulting in 7.0998% voltage regulation and 9.6933% power losses.

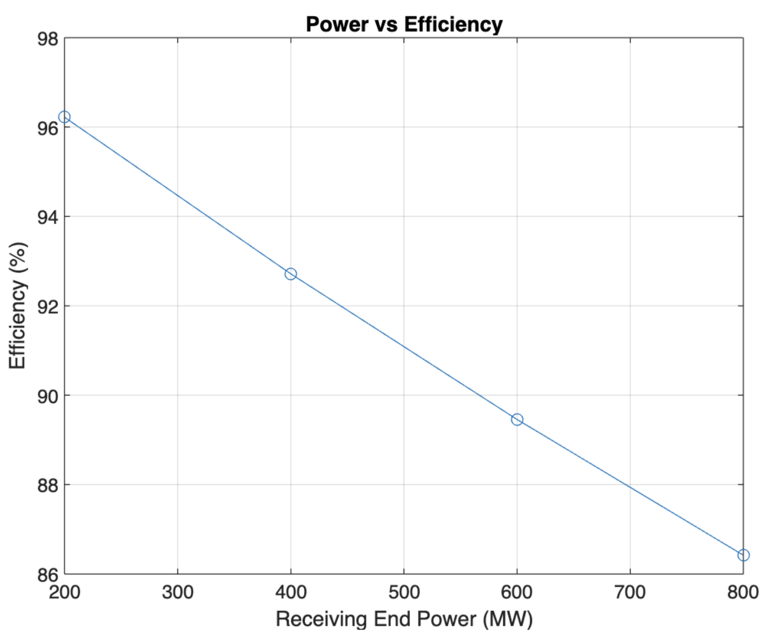
- **Efficiency Optimization:** Efficiency increases with improved power factor, reaching maximum at unity power factor.

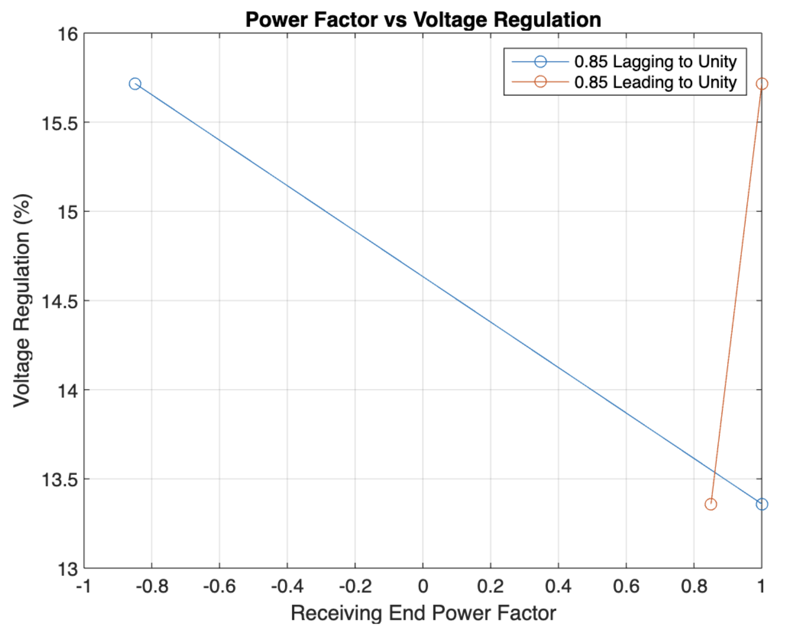
- **Voltage Regulation Impact:** Voltage regulation improves with better power factor, achieving optimal results at unity power factor.

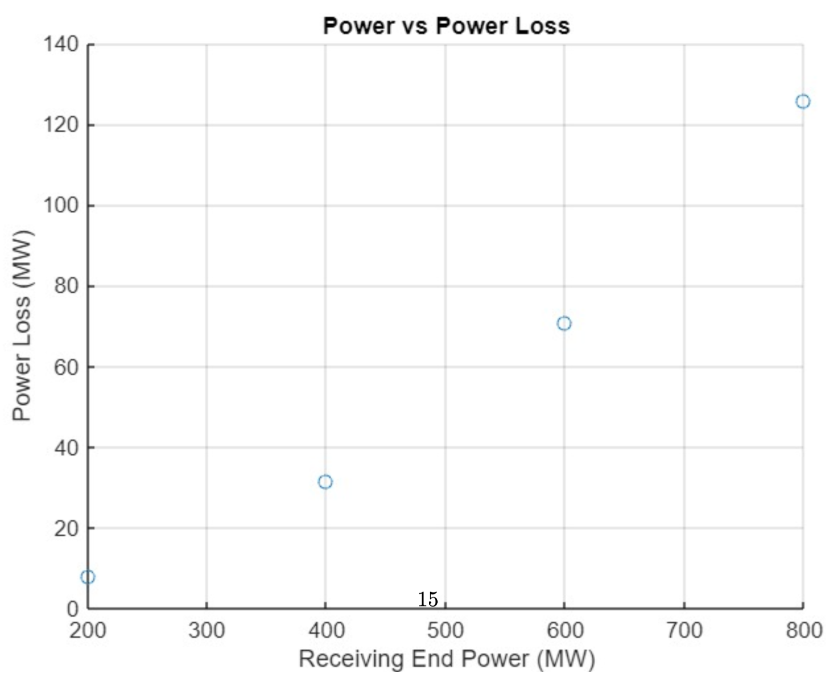
- **Efficiency vs Power Relationship:** Efficiency decreases as receiving end power increases.

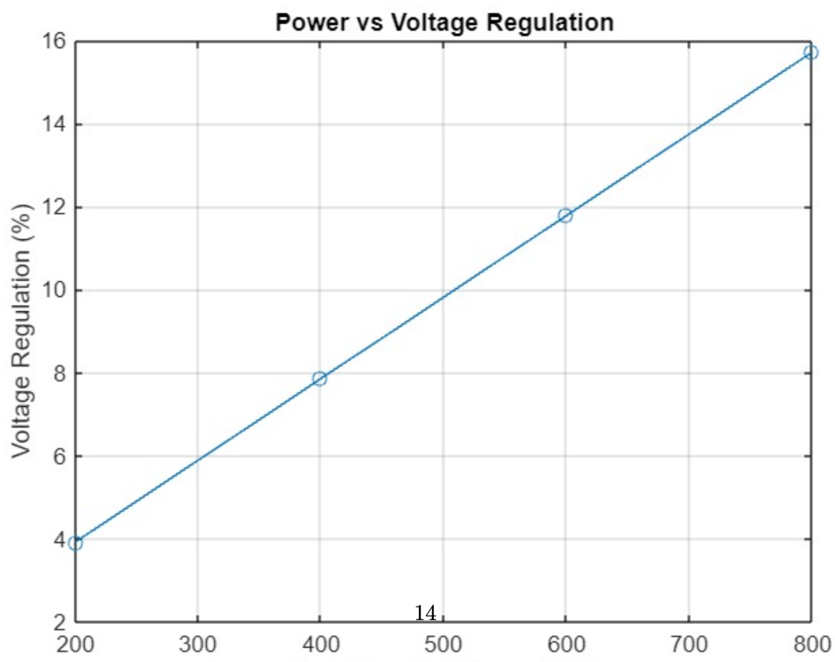
- **Voltage Regulation vs Power Relationship:** Voltage regulation increases as receiving end power increases.

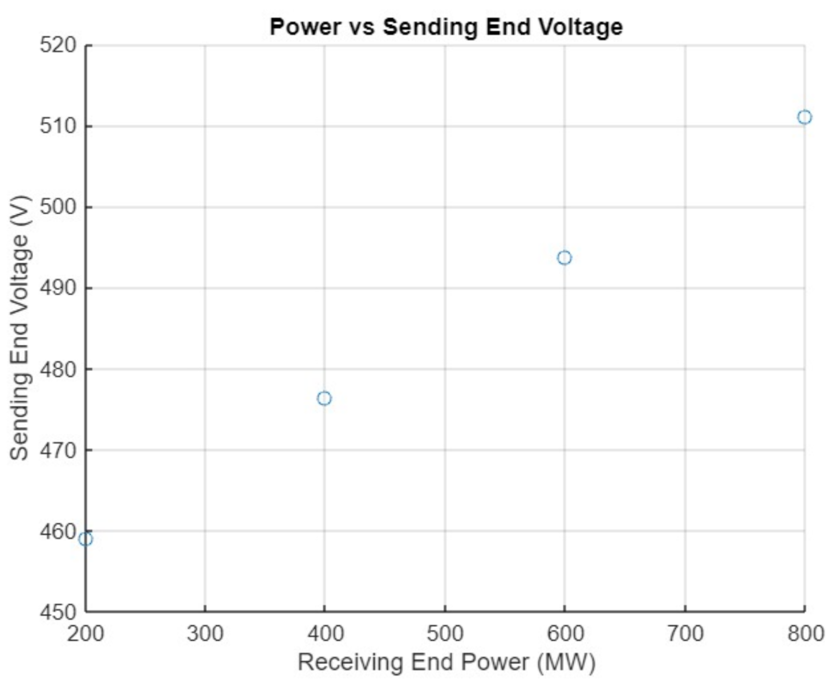

Power Factor vs Efficiency
0.85 Lagging to Unity
Unity to 0.85 Leading
Efficiency (%)
Receiving End Power Factor
150
100
50
0
-50
-100
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


Power vs Efficiency
Efficiency (%)
Receiving End Power (MW)
98
96
94
92
90
88
86
200
300
400
500
600
700
800


Power Factor vs Voltage Regulation
0.85 Lagging to Unity
0.85 Leading to Unity
Voltage Regulation (%)
Receiving End Power Factor
16
15.5
15
14.5
14
13.5
13
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


Power vs Power Loss
Power Loss (MW)
Receiving End Power (MW)
140
120
100
80
60
40
20
0
200
300
400
500
600
700
800
15


Power vs Voltage Regulation
Voltage Regulation (%)
16
14
12
10
8
6
4
2
200
300
400
500
600
700
800


Power vs Sending End Voltage
Sending End Voltage (V)
520
510
500
490
480
470
460
450
200
300
400
500
600
700
800
Receiving End Power (MW)

- **Voltage vs Power Relationship:** Sending end voltage increases as receiving end power increases.

5 conclusion

After analyzing the relationships between key parameters like voltage levels, power losses, voltage regulation, and conductor characteristics, MATLAB reveals that the choice of transmission voltage significantly impacts the line's performance and efficiency. Higher voltage levels, such as 765 kV, reduce power losses and improve voltage regulation, making them more efficient for long-distance power transmission. However, they require higher initial costs. The conductor's geometric mean radius (GMR) and resistance determine the line's reactance and losses. The ACSR "peachbell" conductor, with its favorable electrical and mechanical properties, supports efficient power delivery across different voltage levels. MATLAB identifies the 765 kV configuration as the optimal design for this project, minimizing power losses and maintaining stable voltage levels while balancing long-term efficiency with cost considerations.