REE 322
Advanced Circuits Design
Transmission Line Design Project

Course Coordinator: Dr. Mohamed Morgan
Teaching Assistant: Eng. Ahmed Galal

Prepared By:

Mohamed A. Ibrahim	202200445
Ziad Eissa	202201334

January 18, 2025

Contents

1 Abstract

In this project, the objective is to determine the most suitable conductor code for ACSR conductors to transmit a specified amount of power while adhering to critical design specifications such as voltage regulation percentage (VRL%), sending-end voltage and current, and power loss. The design integrates advanced circuit analysis knowledge with fundamental principles of medium transmission line modeling using the nominal π circuit and its specific ABCD parameters. MATLAB is utilized to simulate various scenarios, analyze outputs, and finalize an optimal transmission line design. The approach ensures compliance with electrical, mechanical, environmental, and economic constraints, resulting in an efficient and cost-effective transmission solution.

2 Introduction

Transmission lines play a pivotal role in the seamless and efficient transfer of electricity over vast distances, forming the backbone of power distribution networks. Their significance lies in the ability to bridge the gap between power generation sources and end-users, ensuring a reliable supply of electricity across diverse geographical regions. The modeling of transmission lines is essential for understanding and optimizing their behavior, and it varies based on their length—short, medium, or long. Each category presents distinct challenges and considerations. In this project, we focus on the modeling and design of a medium transmission line spanning 160 kilometers. Medium transmission lines, characterized by their moderate length, strike a balance between the complexities of long-distance transmission and the simplicity of shorter connections. This project endeavors to address the unique challenges posed by medium transmission lines, employing MATLAB and the π model to determine optimal design parameters for enhanced efficiency and minimized losses.

2.1 Medium Transmission Lines Model

Medium transmission lines, falling within the range of 80 to 250 kilometers at a frequency of 60 Hz, are characterized by a distinct modeling approach. In the case of these medium-length lines, a prevalent technique involves lumping the total shunt capacitance and distributing half at each end of the transmission line. This configuration, referred to as a nominal π circuit, simplifies the representation of the transmission line while maintaining accuracy in modeling its electrical characteristics. A schematic for the nominal π circuit is shown in Figure 1.

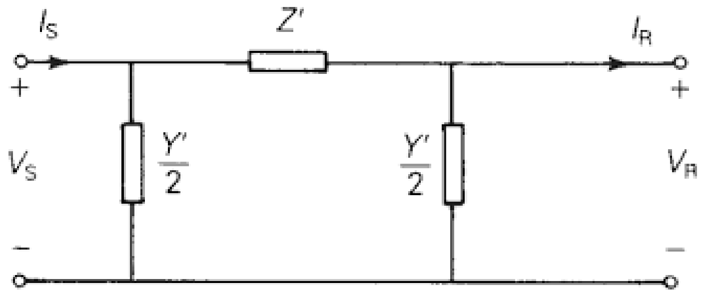


Figure 1: Schematic of the nominal π circuit for medium transmission lines.

The nominal π circuit allows for a practical and computationally efficient simulation of medium transmission line behavior. By strategically distributing shunt capacitance, this model captures the essential electrical dynamics, aiding in the analysis and design of transmission lines with lengths that fall within the medium range, such as the 160-kilometer line under consideration in this project.

Medium transmission line parameters can be summarized as follows:

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} \left(1 + \frac{YZ}{2}\right) & Z \\ Y\left(1 + \frac{YZ}{4}\right) & \left(1 + \frac{YZ}{2}\right) \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \tag{1}$$

where Y is the lumped admittance and Z is the impedance. The transmission line parameters A, B, C, and D are determined from Equation 1 as follows:

$$A = D = 1 + \frac{YZ}{2} \quad \text{(per unit)} \tag{2}$$

$$B = Z \quad \text{(in Ohms)} \tag{3}$$

$$C = Y\left(1 + \frac{YZ}{4}\right) \quad \text{(in Siemens)} \tag{4}$$

3 Methodology

3.1 Overview

This section outlines the step-by-step implementation of a MATLAB-based simulation for analyzing power transmission parameters. The simulation focuses on calculating transmission line currents, circuit requirements, line parameters, and efficiency for various materials and voltage levels.

3.2 Input Parameters

Power and Power Factor:

- **Power:** 800 MW
- **Power Factor:** 0.85

Transmission Parameters:

- **Route Length:** 160 km
- **Frequency:** 60 Hz
- **Maximum Power Angle:** 30 (converted to radians)
- **Voltage Regulation:** 10
- **Maximum Power Loss:** 100 kW

Voltage and Reactance Options:

- **Voltage Levels:** [345 kV, 500 kV, 765 kV]
- **Reactance:** [0.3, 0.32, 0.27] Ohm/km
- **Number of Bundles (k):** Ranges based on voltage level.

Potential Material Properties:

- **Materials:** Linnet, Wedgion, Merlin, Piper, Waxwing
- **Resistance and GMR values:** Used to calculate inductance and capacitance.

3.3 Step 1: Transmission Data Setup

A table is created to display voltage options and their corresponding reactance values. These values form the basis of subsequent calculations.

3.4 Step 2: Loop Through Voltage Levels and Bundle Configurations

- **Current Calculation:** Line current is calculated based on conjugate power and voltage.
- **Minimum Circuits:** The minimum number of circuits required is determined using the line length, reactance, and power angle.
- **Bundle Current:** The current per bundle is computed for each configuration.

Results are stored in a table for each voltage and bundle configuration.

3.5 Step 3: Line Parameters Calculation

For each material:

- **Inductance and Capacitance:**
  - Calculated using GMR, GMD, and material properties.
- **ABCD Parameters:**
  - Medium Π model used to compute transmission line parameters (A, B, C, D).

3.6 Step 4: Power Transmission Analysis

- **Voltage and Current at Sending End:**
  - Derived using ABCD parameters and line current.
- **Power Factor and Power Loss:**
  - Calculated from voltage and current phasors.

- **Voltage Regulation and Efficiency:**
  - Voltage regulation is determined using no-load and full-load voltages.
  - Efficiency is calculated by accounting for power loss.

Results are aggregated into a table.

3.7 Step 5: Filtering and Optimization

- Filter designs based on:
  - Voltage Regulation (<10
  - Efficiency (>90
- Sort filtered results by voltage level and material.
- Select the top three designs with the best performance.

4 Matlab Code

Listing 1: MATLAB Code

```matlab
clear;

% Input parameters
Power = 800 * 1e3;                % Power in Watt
S = 800 - 495.7851j;              % Complex power
S_Conjugate = 800 + 495.7851j;    % Complex power Conjugate
PF = 0.85;                        % Power Factor
L = 160;                          % Route Length in Km
f = 60;                           % Frequency in Hz
Power_angle_max = 30 * pi / 180;  % Power angle in Rad
Conductor_Spacing = 8;            % Conductor Spacing in meters
Bundle_Spacing = 0.45;            % Bundle Spacing in meters
VR_Maximum = 0.1;                 % Voltage Regulation Percentage
Power_Loss_Max = 100 * 1e3;       % Maximum Power Loss in Watt

% Properties Table
Voltage_options = [345; 500; 765];       % Transmission Voltage (kV)
Reactance = [0.3; 0.32; 0.27];           % Reactance per km (Ohms/km)
k_range = {[2]; [2; 3; 4]; [3; 4]};      % Range of k values as arrays

% Display voltage and reactance data
disp('Transmission Data:');
disp(table(Voltage_options, Reactance, 'VariableNames', {'Voltage_kV', 'Reactance_per_km_Ohm'}));

```

```
% Results storage
Results = [];

% Loop through each voltage level and its corresponding k_range
for i = 1:length(Voltage_options)
    Voltage_V = Voltage_options(i) * 1e3;  % Convert to volts
    k_values = k_range{i};                % Extract k values for this voltage level

    for k = k_values' % Loop through each k value for the current voltage level
        % Calculate Line Current
        LineCurrent = S_Conjugate * 1000 / (sqrt(3) * Voltage_V);

        % Calculate Minimum Number of Circuits
        Num_Circuits_Minimum = 1 + ceil((L * Reactance(i) * Power) / ...
                                    (Voltage_V^2 * sin(Power_angle_max)));

        % Calculate Current per Bundle
        I_b = LineCurrent / (Num_Circuits_Minimum * k);

        % Store results in a row
        Results = [Results; Voltage_options(i), k, LineCurrent, Num_Circuits_Minimum, I_b];
    end
end

% Create results table
ResultsTable = array2table(Results, ...
    'VariableNames', {'Voltage_kV', 'k', 'Line_Current', 'Num_Circuits_Minimum', 'Bundle_Current'});
disp('Results:');
disp(ResultsTable);

% Additional parameters from the provided table
Materials = {'Linnet', 'Wedgion', 'Merlin', 'Piper', 'Waxwing'};
Resistance_AC = [178.8, 173.7, 173.0, 195.0, 218.1] * 1e-3; % Resistance in Ohms/km
GMR_values = [7.41, 7.05, 6.74, 7.05, 6.00] * 1e-3;       % GMR in meters
D1 = 8;
D2 = 16;
GMD = (D1 * D2 * D1)^(1/3); % Geometric Mean Distance

% Constants
mu_0 = 4 * pi * 1e-7; % Permeability of free space (H/m)
epsilon_0 = 8.854e-12; % Permittivity of free space (F/m)

% Calculate Inductance, Capacitance, and ABCD parameters
```

```
Inductance = zeros(length(Materials), 1);
Capacitance = zeros(length(Materials), 1);
A = zeros(length(Materials), 1);
B = zeros(length(Materials), 1);
C = zeros(length(Materials), 1);
D = zeros(length(Materials), 1);

for i = 1:length(Materials)
    % Extract resistance and GMR
    R = Resistance_AC(i);
    GMR = GMR_values(i);

    % Inductance per unit length (H/m)
    Inductance(i) = (mu_0 / (2 * pi)) * log(GMD / GMR);

    % Capacitance per unit length (F/m)
    Capacitance(i) = (2 * pi * epsilon_0) / log(GMD / GMR);

    % Inductive and Capacitive Reactances
    XL = 2 * pi * f * Inductance(i) * L; % Ohm
    XC = 1 / (2 * pi * f * Capacitance(i) * L); % Ohm

    % ABCD Parameters (Medium Pi Model)
    z = R + 1j * XL;                        % Total impedance
    y = 1j * 2 * pi * f * Capacitance(i);   % Total admittance
    Z = z * L;                              % Total impedance
       over the line
    Y = y * L;                              % Total admittance
       over the line

    A(i) = 1 + (Z * Y) / 2;
    B(i) = Z;
    C(i) = Y * (1 + (Z * Y) / 4);
    D(i) = A(i);
end

% Display Results
disp('Inductance (H/m):');
disp(Inductance);

disp('Capacitance (F/m):');
disp(Capacitance);

ABCD_Table = table(Materials', A, B, C, D, ...
    'VariableNames', {'Material', 'A', 'B', 'C', 'D'});
disp('ABCD Parameters:');
disp(ABCD_Table);

% Extract columns from ABCD_Table
Materials_Array = ABCD_Table.Material;  % Materials as a cell
   array
```

```
A_Array = ABCD_Table.A;                    % A column as an array
B_Array = ABCD_Table.B;                    % B column as an array
C_Array = ABCD_Table.C;                    % C column as an array
D_Array = ABCD_Table.D;                    % D column as an array

% Extract data from ResultsTable
Voltage_Array = ResultsTable.Voltage_kV * 1000; % Convert kV to V
LineCurrentArray = ResultsTable.Line_Current;   % Line current
    array
% Define transmitted power (800 MW assumed)
power_trans = 800 * 1e6; % Transmitted Power in Watts (800 MW)

% Initialize the results table
Table_Results = table();

% Loop through each voltage and material
for i = 1:length(Voltage_Array)
    for j = 1:length(Materials_Array)
        % Calculate V_Sending and I_Sending using
           LineCurrentArray
        V_Sending = (A_Array(j) * Voltage_Array(i) / sqrt(3)) +
           ...
                    (B_Array(j) * LineCurrentArray(i));
        I_Sending = (C_Array(j) * Voltage_Array(i)) + ...
                    (D_Array(j) * LineCurrentArray(i) * (10^3));

        % Calculate Power Factor (pf_sending)
        pf_sending = cosd(rad2deg(angle(I_Sending)) - rad2deg(
           angle(V_Sending)));

        % Calculate Power (Ps) in Watts
        Ps = 3 * abs(I_Sending) * abs(V_Sending) * pf_sending; %
           in Watts
        P_loss = (Ps - power_trans)/ 1e3; % Power loss in Watts
        Ps_MW = (800 - P_loss) ;  % Convert to MW after
           accounting for power loss

        % Calculate No-load Voltage (V_no_load)
        V_no_load = (abs(V_Sending) * sqrt(3)) / abs(A_Array(j));

        % Calculate Voltage Regulation (VRegulation)
        V_load = Voltage_Array(i);  % Full load voltage
        VRegulation = 1000 * (V_no_load - V_load) / V_load;

        % Efficiency calculation
        efficiency = Ps_MW/800 * 100; % Efficiency in percentage

        % Add to the results table
        new_row = table(string(Materials_Array{j}), Voltage_Array
           (i), Ps_MW, VRegulation, efficiency, P_loss , ...
```

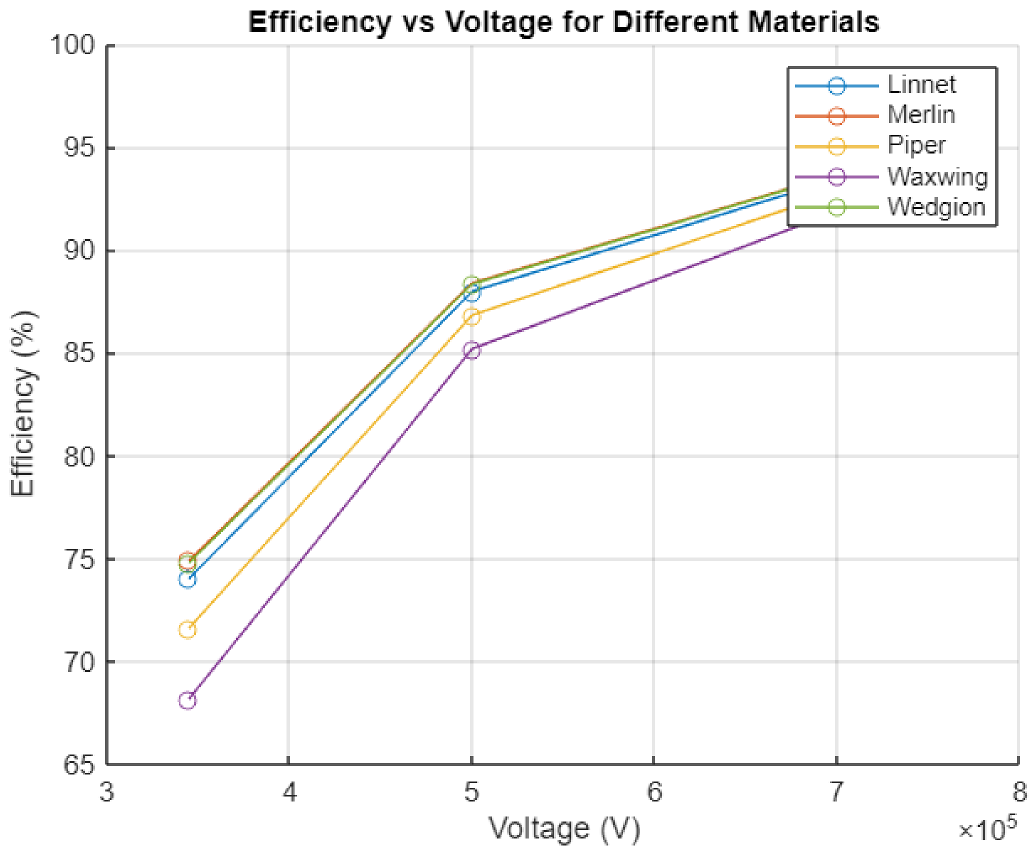
```
        'VariableNames', {'Material', 'Voltage', 'Power_MW', 'Voltage_Regulation', 'Efficiency', 'Power_Loss_kW'});
        Table_Results = [Table_Results; new_row];
    end
end

% Display the table
disp('Results Table:');
disp(Table_Results);

% Filter designs based on Power Factor and Efficiency
Filtered_Results = Table_Results(Table_Results.Voltage_Regulation < 10 & ...
                                 Table_Results.Efficiency > 90, :);

% Sort filtered results by Voltage and Material (ascending order)
Sorted_Results = sortrows(Filtered_Results, {'Voltage', 'Material'});

% Select the best three designs
Best_Three_Designs = Sorted_Results(1:min(3, height(Sorted_Results)), :);

% Display the top three designs
disp('Best Three Designs:');
disp(Best_Three_Designs);

% Extract unique materials
materials = unique(Table_Results.Material);

figure;
hold on;
for i = 1:length(materials)
    % Filter data for each material
    material_data = Table_Results(strcmp(Table_Results.Material, materials{i}), :);

    % Plot Efficiency vs Voltage
    plot(material_data.Voltage, material_data.Efficiency, '-o', 'DisplayName', materials{i});
end
hold off;
grid on;
title('Efficiency vs Voltage for Different Materials');
xlabel('Voltage (V)');
ylabel('Efficiency (%)');
legend('show');
```

5 Results

This section includes the results and the best three transmission line designs based on the specified constraints as shown in figure 2. At a minimum Number of circuits = 2, and 3,4,3 Number of bundles respectively:

Best Three Designs:

Material	Voltage	Power_MW	Voltage_Regulation	Efficiency	Power_Loss_MW
"Linnet"	7.65e+05	761.86	0.02731	95.232	38.142
"Linnet"	7.65e+05	761.86	0.02731	95.232	38.142
"Merlin"	7.65e+05	763.26	0.025886	95.408	36.737



Results

6 Conclusion

This research successfully used MATLAB to construct and analyze a medium transmission line model. The built MATLAB algorithm analyzed critical performance factors, including efficiency and voltage regulation.

Providing useful insights into the system's behavior. The data helped guide material selection and optimize the transmission line. The graphs show how the model performs under different settings, providing a clear description of its behavior. This research not only provided a comprehensive understanding of medium transmission line dynamics, but also showcased the efficiency of MATLAB for system design and analysis.

7 References

[1] Grainger, J. J., Stevenson, W. D. (1994). Power System Analysis. McGraw Hill.
[2] Glover, J. D., Sarma, M. S., Overbye, T. J. (2008). Power System Analysis and Design. Thomson.