MIDDLE EAST TECHNICAL UNIVERSITY

ELECTRICAL AND ELECTRONICS ENGINEERING DEPARTMENT

EE 374
ELECTRICAL EQUIPMENT AND APPLICATIONS
TERM PROJECT REPORT

Spring 2019-2020

Nicat Aliyev 1906619

Abstract

The power transmission lines are one of the most important areas of the XXI century. It plays a vital role in today's daily life of humans and economies of countries. Therefore it is important to know the details of the transmission line systems which can be obtained by finding the line parameters. The line parameters give the relation between the supply and receiving parts of transmission systems. Finding line transmission parameters is the main objective of our project. Therefore a MATLAB code that calculates the transmission line parameters is developed in this work.

Table of contents

1. Introduction

In this report an explanation of the project and the process duration is explained. In te first part the background knowledge needed for implementing in the project is given. Then the experimental procedure and test results for the different cases is explained. And finally in the conclusion part the knowledge that was obtained in project duration is written

2. Background knowledge

In the project procedure some throughout research was made about the transmission lines to obtain background knowledge. During this research it was learnt that there are different types of cables with their own properties. This properties shows the important parameters of cables like the DC and AC resistances for different temperatures, inductive/capacitive reactance, a cross sectional area of the conductors and also other parameters. It is important to know these parameters as different cables can withstand different weather conditions and maximum electrical power levels. As an example in the figure1 the change of power loss of an AC transmission line versus temperature can be observed. On this plot we can see that increasing the radius of the conductor even in several cm significantly decreases the power loss, as the resistance of the conductor is decreased due to high cross-sectional area. This example show how important is designing process of transmission lines and thus to decrease the costs.

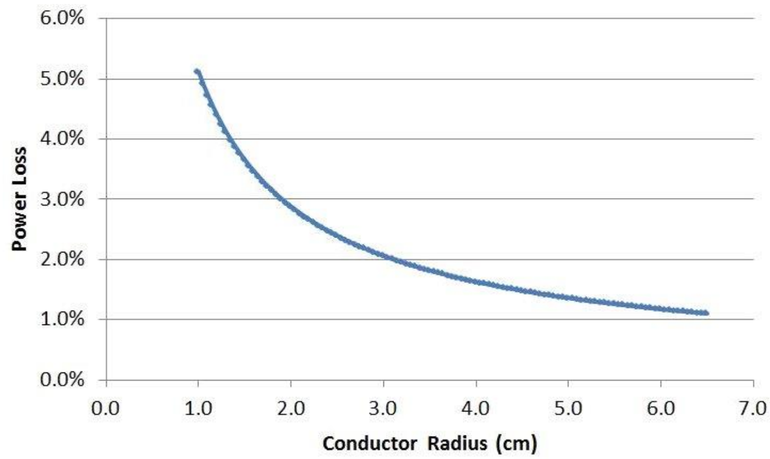

Figure 1. Relation of power loss percentage with the conductor radius in cm

Another information which is needed in the estimation of parameters is the tower configuration, meaning that one needs to know phase spacing, number of bundles in the conductor, bundle distance, and frequency.

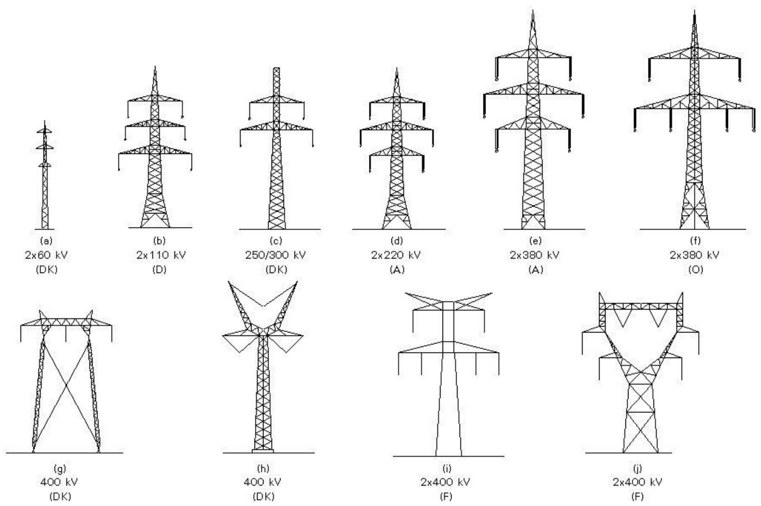

Figure 2. Different types of power towers

All these parameters are given for our project in excel and text document forms. After obtaining them the transmission line parameters can be calculated.

3. Work Procedure

The project process was divided into three phases. In the first phase of the project the goal was extracting the input data from an input text file which contains the number of circuits, number of bundle conductors, bundle distance, length of the line, name of the ACSR conductor, and location of the phases with respect to the origin. After obtaining these parameters it is required to get the conductor parameters from the library. The code reads the input text line by line and assigns the data on each data line to a corresponding variable according to the text data line. It also checks if the name of the conductor is correct and in case if the name is misspelled or not listed in the table it asks for another name input and continues the processing according to the new name.

Then in the second phase we were required to calculate the electrical parameters of the line, i.e., series resistance & reactance (Ω/km), and shunt susceptance (℧/km). First with the basic geometrical calculations the distance between phases and consequently GMR/GMD values for all cases are found. These cases include the number of circuits (one or two circuits) and number of bundles. The code allows calculating the GMR values for both capacitor and inductor up to 8 bundles in a conductor. Additionally, earth effect capacitance was also found to get better results. Using these parameters inductance and capacitance were calculated and DC resistance was extracted from the library. Finally using the following code series resistance & reactance, and shunt susceptance were calculated.

```
L=2*10^-4*log(GMD/GMRL);
C=2*pi*eps/(log(GMD/GMRC)-log(Cearth));
if k==2
    C=2*pi*eps/(log(GMD/GMRC));
end
Rf=lbr{Index,7}/((1.60934)*n*k);
Xf=2*pi*f*L;
Bf=2*pi*10^3*f*C;
```

Figure 3. The code to calculate the second phase parameters

Finally in the third phase ABCD parameters of the line for medium and long length line assumptions were found. The line parameter calculation is different for long and medium lines. In real designing processes the line is considered to be long when it is longer than 150 miles or 241 km. During the tests it was observed that if the length is less than 240 km the line parameters are almost the same and they even get closer when we decrease the length. After 240 km the line is considered to be long and longer lengths results in bigger differences. Therefore according to the length of the line either the first four Am, Bm, Cm, Dm for the medium length or last four Al, Bl, Cl, Dl values for the long length should be used.

4. Experimental Procedure and Results

Finally, after finishing the Matlab code it was tested by giving different input examples. These inputs included different number of bundles, conductor types, and conductor lengths with one or two circuits. Test results for the input_file_example1.txt and input_file_example.txt, where first has 2 number of circuits and the second – 1 circuit.

Al	0.9944 + 0.0004i
Am	0.9944 + 0.0004i
ans	0
Bl	0.7715 + 9.6319i
Bm	0.0008 + 0.0096i
Cl	-0.0000 + 0.0012i
Cm	-2.6033e-16 + 1.1596e-06i
Dl	0.9944 + 0.0004i
Dm	0.9944 + 0.0004i
fileID	18
length	100000
library_path	'library.csv'
text_path	'Input_file_example1.txt'

Al	0.9913 + 0.0004i
Am	0.9913 + 0.0004i
ans	0
Bl	0.8285 + 18.9490i
Bm	0.0008 + 0.0190i
Cl	-1.1651e-07 + 9.1406e-04i
Cm	-1.7507e-16 + 9.1672e-07i
Dl	0.9913 + 0.0004i
Dm	0.9913 + 0.0004i
fileID	18
length	125000
library_path	'library.csv'
text_path	'Input_file_example2.txt'

As mentioned before the tests results showed that the line parameters give different results as we increase the length of the line. In the below test results the line parameters for 100 km, 200 km, 241 km, and 500 km are shown (test results are calculated for the input_file_example1.txt)

Al	0.9944 + 0.0004i
Am	0.9944 + 0.0004i
ans	0
Bl	0.7715 + 9.6319i
Bm	0.0008 + 0.0096i
Cl	-0.0000 + 0.0012i
Cm	-2.6033e-16 + 1.1596e-06i
Dl	0.9944 + 0.0004i
Dm	0.9944 + 0.0004i
fileID	18
length	100000
library_path	'library.csv'
text_path	'Input_file_example1.txt'

Al	0.9777 + 0.0018i
Am	0.9776 + 0.0018i
ans	0
Bl	1.5257 + 19.1567i
Bm	0.0015 + 0.0193i
Cl	-0.0000 + 0.0023i
Cm	-2.0826e-15 + 2.3192e-06i
Dl	0.9777 + 0.0018i
Dm	0.9776 + 0.0018i
fileID	18
length	200000
library_path	'library.csv'
text_path	'Input_file_example1.txt'

Al	0.9501 + 0.0040i
Am	0.9496 + 0.0040i
ans	0
Bl	2.2458 + 28.4688i
Bm	0.0023 + 0.0289i
Cl	-0.0000 + 0.0034i
Cm	-7.0288e-15 + 3.4788e-06i
Dl	0.9501 + 0.0040i
Dm	0.9496 + 0.0040i
fileID	18
length	300000
library_path	'library.csv'
text_path	'Input_file_example1.txt'

Al	0.9118 + 0.0070i
Am	0.9105 + 0.0072i
ans	0
Bl	2.9151 + 37.4647i
Bm	0.0031 + 0.0386i
Cl	-0.0000 + 0.0045i
Cm	-1.6661e-14 + 4.6384e-06i
Dl	0.9118 + 0.0070i
Dm	0.9105 + 0.0072i
fileID	18
length	400000
library_path	'library.csv'
text_path	'Input_file_example1.txt'

Test results with the input_file_example1.txt with 4 (upper-left), 5 (upper-right), 6(down-left), 7(down-right) bundles. We observe that although A and D parameters don't change the other two parameters change

Al	0.9944 + 0.0004i
Am	0.9944 + 0.0004i
ans	0
Bl	1.5430 + 20.0446i
Bm	0.0015 + 0.0201i
Cl	-8.0225e-08 + 5.5676e-04i
Cm	-1.2047e-16 + 5.5780e-07i
Dl	0.9944 + 0.0004i
Dm	0.9944 + 0.0004i
fileID	18
length	100000
library_path	'library.csv'
text_path	'Input_file_example1.txt'

Al	0.9944 + 0.0004i
Am	0.9944 + 0.0004i
ans	0
Bl	0.6172 + 8.7471i
Bm	0.0006 + 0.0088i
Cl	-0.0000 + 0.0013i
Cm	-2.5138e-16 + 1.2740e-06i
Dl	0.9944 + 0.0004i
Dm	0.9944 + 0.0004i
fileID	18
length	100000
library_path	'library.csv'
text_path	'Input_file_example1.txt'

Al	0.9944 + 0.0004i
Am	0.9944 + 0.0004i
ans	0
Bl	0.5143 + 8.0457i
Bm	0.0005 + 0.0081i
Cl	-0.0000 + 0.0014i
Cm	-2.4683e-16 + 1.3829e-06i
Dl	0.9944 + 0.0004i
Dm	0.9944 + 0.0004i
fileID	18
length	100000
library_path	'library.csv'
text_path	'Input_file_example1.txt'

Al	0.9944 + 0.0003i
Am	0.9944 + 0.0003i
ans	0
Bl	0.4409 + 7.4659i
Bm	0.0004 + 0.0075i
Cl	-0.0000 + 0.0015i
Cm	-2.4514e-16 + 1.4886e-06i
Dl	0.9944 + 0.0003i
Dm	0.9944 + 0.0003i
fileID	18
length	100000
library_path	'library.csv'
text_path	'Input_file_example1.txt'

5. Conclusion

In this project we learned the importance of the transmission lines in our world. We learned how important is to make the design of the transmission system by considering the safety and costs also the fact that temperature and also other weather conditions can affect the design . This research gave us a great knowledge on parameters of transmission line systems and calculating them. Also using Matlab was a very good practice to improve our skills on this programming platform.

APPENDIX

Matlab code

```
function [am, bm, cm, dm, al, bl, cl, dl, l] = e190661_aliyev(txt_path, lbr_path)
Xb=-1;
Bb=-1;
  %% Open the file and get the first line
  fileID = fopen(txt_path,'r');

  tline = fgetl(fileID);
```

```
%% assign d2XX values to -1
d2AB=-1;
d2AC=-1;
d2BC=-1;
%% get the library
lbr = table2cell(readtable(lbr_path, 'PreserveVariableNames',true));
%% some constants and variables
eps=8.854*10^-12 % permitivity of free space
f=50; % frequency of the system,(Hz)

%% exporting variables from the file
while ~strcmp(tline,'-999')
  if (strcmp(tline,'Number of circuits'))
    k=str2double(fgetl(fileID));    % Number of circuits
  end

  if (strcmp(tline,'Number of bundle conductors per phase'))
    n=str2double(fgetl(fileID));    % Number of bundle conductors per phase
  end

  if (strcmp(tline,'Bundle distance (m)'))
    d=str2double(fgetl(fileID));    % Bundle distance (m)
  end

  if (strcmp(tline,'Length of the line (km)'))
    l=str2double(fgetl(fileID));    % Length of the line (km)
  end

  if (strcmp(tline,'ACSR conductor name'))
    name=fgetl(fileID);             % ACSR conductor name

    while (~any(strcmp(lbr(:,1),name)))  % give warning in case of wrong conductor name
      warning('WARNING!!! Conductor name is not listed in the library. Please try another conductor
name.');
      name=input('Enter anoter name: ', 's');
    end
    %% get the parameters of the conductor
    names=lbr(:,1); % names of the conductors in double format
    IndexC = strfind(names,name); % find row of the conductor
    Index = find(not(cellfun('isempty',IndexC))); % row of the conductor in the library
    sAL=lbr{Index,2}*5.067e-10;   % Aliminium area (m^2)
    stranding=lbr{Index,3};      % Stranding
    layers=lbr{Index,4};         % Layers of Aliminium
    Do=lbr{Index,5}*0.0254;      % Outside diameter (m)
    ro = Do/2;                   % Outside radius (m)

    Rac=lbr{Index,7}/1609.344;    % AC 50 Hz Resistance 20?C (ohm/m)
    GMR=lbr{Index,8}*0.3048;      % GMR (m)
  end

  %% get the coordinates of each phase and distances between centers of each bundles
  if (strcmp(tline,'C1 Phase A (centre)')) %% C1 Phase A (centre)
    x1=str2double(fgetl(fileID));
    y1=str2double(fgetl(fileID));
  end

  if (strcmp(tline,'C1 Phase B (centre)')) %% C1 Phase B (centre)
    x2=str2double(fgetl(fileID));
    y2=str2double(fgetl(fileID));
```

```
    end

    if (strcmp(tline,'C1 Phase C (centre)')) %% C1 Phase C (centre)
      x3=str2double(fgetl(fileID));
      y3=str2double(fgetl(fileID));
    end

    if (strcmp(tline,'C2 Phase A (centre)')) %% C2 Phase A (centre)
      x4=str2double(fgetl(fileID));
      y4=str2double(fgetl(fileID));
    end

    if (strcmp(tline,'C2 Phase B (centre)')) %% C2 Phase B (centre)
      x5=str2double(fgetl(fileID));
      y5=str2double(fgetl(fileID));
    end

    if (strcmp(tline,'C2 Phase C (centre)')) %% C2 Phase C (centre)
      x6=str2double(fgetl(fileID));
      y6=str2double(fgetl(fileID));
    end
    tline = fgetl(fileID); %% get the next line (corresponds to text lines)
  end
  fclose('all');


  %% Phase II
  % distances between phases
  d1AB=sqrt((x2-x1)^2+(y2-y1)^2);
  d1AC=sqrt((x3-x1)^2+(y3-y1)^2);
  d1BC=sqrt((x2-x3)^2+(y2-y3)^2);
  GMD = nthroot(d1AB*d1AC*d1BC,3);
  if k==2
    d2AB=sqrt((x5-x4)^2+(y5-y4)^2);
    d2AC=sqrt((x6-x4)^2+(y6-y4)^2);
    d2BC=sqrt((x6-x5)^2+(y6-y5)^2);
    dA1A2=sqrt((x6-x1)^2+(y6-y1)^2);
    dA1B2=sqrt((x5-x1)^2+(y5-y1)^2);
    dA1C2=sqrt((x4-x1)^2+(y4-y1)^2);
    dB1A2=sqrt((x6-x2)^2+(y6-y2)^2);
    dB1B2=sqrt((x5-x2)^2+(y5-y2)^2);
    dB1C2=sqrt((x4-x2)^2+(y4-y2)^2);
    dC1A2=sqrt((x6-x3)^2+(y6-y3)^2);
    dC1B2=sqrt((x5-x3)^2+(y5-y3)^2);
    dC1C2=sqrt((x4-x3)^2+(y4-y3)^2);
    DAC=nthroot(dA1C2*dC1A2*d1AC*d2AC,4);
    DAB=nthroot(d1AB*d2BC*dB1A2*dA1B2,4)
    DBC=nthroot(dB1C2*dC1B2*d1BC*d2AB,4)
    GMD=nthroot(DAB*DAC*DBC,3);
  end

%
  if n==1
    GMRL=GMR;
    GMRC=ro;
  elseif n==2
    GMRL=sqrt(GMR*d);
    GMRC=sqrt(ro*d);
  elseif n==3
    GMRL=nthroot(GMR*d^2,3);
```

```
        GMRC=nthroot(ro*d^2,3);
    elseif n==4
        GMRL=nthroot(GMR*d^3*sqrt(2),4);
        GMRC=nthroot(ro*d^3*sqrt(2),4);
    elseif n==5
        GMRL=nthroot(GMR*d^4*(sin(72*pi/180)/cos(54*pi/180))^2,5);
        GMRC=nthroot(ro*d^4*(sin(72*pi/180)/cos(54*pi/180))^2,5);
    elseif n==6
        GMRL = nthroot(GMR*2*d^5*3,6);
        GMRC= nthroot(ro*2*d^5*sqrt(3)^2,6);
    elseif n==7

            GMRL=nthroot(GMR*d^2*((2*sin((450/7)*pi/180)*d)^2)*((d*sin((540/7)*pi/180)/cos((450/7)*pi
            /180))^2),7);
            GMRC=nthroot(ro*d^2*((2*sin((450/7)*pi/180)*d)^2)*((d*sin((540/7)*pi/180)/cos((450/7)*pi/1
            80))^2),7);
        elseif n==8

            GMRL=nthroot(GMR*d^2*((2*dbundle*sin((135/2)*pi/180))^2)*((d*(sqrt(2)+1))^2)*(d/cos((135/
            2)*pi/180)),8);
            GMRC=nthroot(ro*d^2*((2*d*sin((135/2)*pi/180))^2)*((d*(sqrt(2)+1))^2)*(d/cos((135/2)*pi/18
            0)),8);
    end

    if k==2
        % GMR of each phase for inductor
        GMRa=sqrt(dA1A2*GMRL);
        GMRb=sqrt(dB1B2*GMRL);
        GMRc=sqrt(dC1C2*GMRL);
        %
        GMRCa=sqrt(dA1A2*GMRC);
        GMRCb=sqrt(dB1B2*GMRC);
        GMRCc=sqrt(dC1C2*GMRC);
        GMRL=nthroot(GMRa*GMRb*GMRc,3);
        GMRC=nthroot(GMRCa*GMRCb*GMRCc,3);
    end

%% Earth effect capacitance
if k == 1
    H1 = sqrt((x2 - x1)^2 + (y1 + y2)^2)*sqrt((x2 - x3)^2 + (y2 + y3)^2)*sqrt((x1 - x3)^2 + (y1 + y3)^2);
    Cearth = (H1/(2*y1*2*y2*2*y3))^(1/3);
elseif k == 2
    H2= (sqrt((x2-x1)^2 + (y1+y1)^2)*...
        sqrt((x4-x1)^2 + (y5+y1)^2)*...
        sqrt((x2-x6)^2 + (y1+y6)^2)*...
        sqrt((x6-x4)^2 + (y6+y5)^2)*...
        sqrt((x3-x2)^2 + (y3+y2)^2)*...
        sqrt((x4-x2)^2 + (y4+y2)^2)*...
        sqrt((x3-x4)^2 + (y3+y5)^2)*...
        sqrt((x4-x4)^2 + (y5+y4)^2)*...
        sqrt((x3-x1)^2 + (y3+y1)^2)*...
        sqrt((x4-x1)^2 + (y4+y1)^2)*...
        sqrt((x3-x6)^2 + (y3+y6)^2)*...
        sqrt((x4-x6)^2 + (y4+y6)^2))^(1/12);
    H3 = (64*y1*y3*y2*y4*y6*y5*...
          ((x6-x1)^2 + (y1+y6)^2)*...
          ((x3-x4)^2 + (y3+y4)^2)*...
          ((x2-x4)^2 + (y1+y5)^2))^(1/12);
    Cearth = (H2/H3);
end
```

```
    L=2*10^-4*log(GMD/GMRL);
    C=2*pi*eps/(log(GMD/GMRC)-log(Cearth));
    if k==2
      C=2*pi*eps/(log(GMD/GMRC));
    end
    Rf=lbr{Index,7}/((1.60934)*n*k);        % DC Resistance 20?C (ohm/km)
    Xf=2*pi*f*L;
    Bf=2*pi*10^3*f*C;




%% Phase III

R = Rf*l/1000;
X = Xf*l/1000;
Z = R+j*X;
Y = j*Bf*l/1000;
y = Y/l;
z = Z/l;
Zo = sqrt(z/y);
g = sqrt(z*y);
l = l*1000;
% if (l<241)
  am = (Z*Y/2)*10^6+1;
  bm = Z;
  cm = Y*(1+Z*Y/4);
  dm = am;

% else
  al = cosh(g*l);
  bl = Zo*sinh(g*l);
  cl = sinh(g*l)/Zo;
  dl = al;
% end
end
```