

ASSIGNMENT 3

LAB01: LINE PERFORMANCE OF TRANSMISSION LINE ON MATLAB

Lecturer: Sean Piseth

Student:

CHANN KANNHAPHANSA
CHEA KIMLEANG
CHEANG HOR PHENG
CHEAT CHETTRA

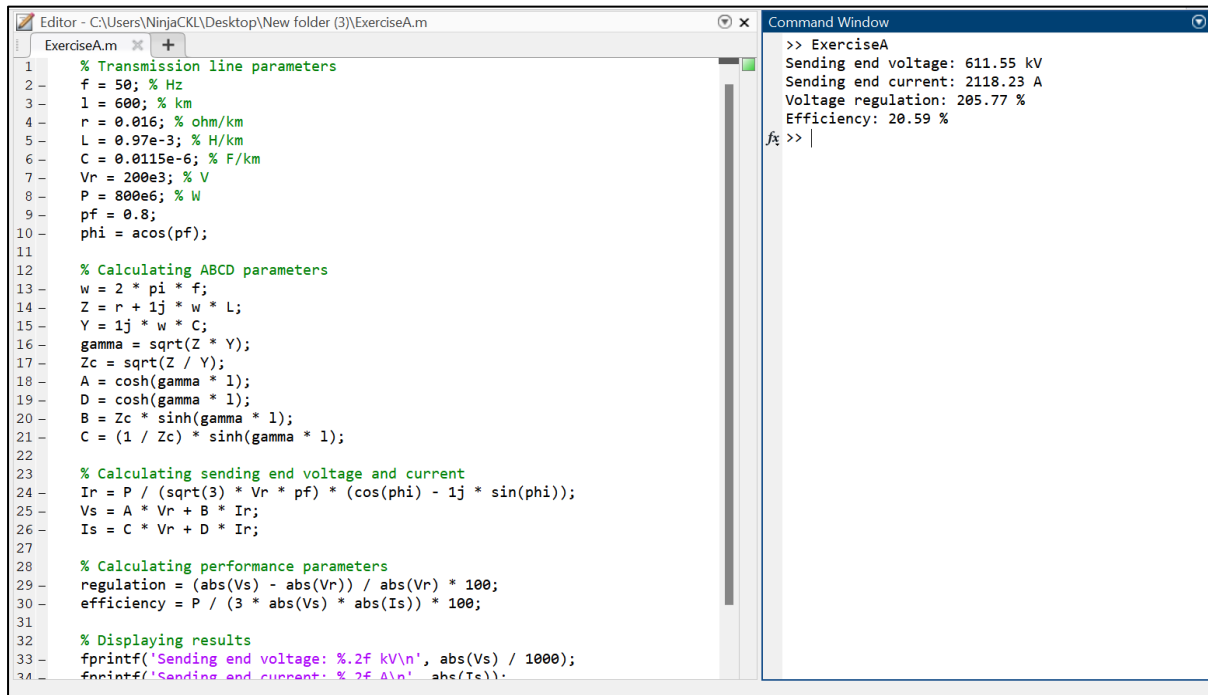
e20190062
e20190075
e20190085
e20190087

2023

Assignment TP3 | Line Performance on Transmission Line

A three phase 50Hz, 220kV transmission line having length of 600km. The line parameters per phase per unit length are found to be $r = 0.0162\Omega/\text{km}$, $L = 0.97\text{mH}/\text{km}$, $C = 0.0115\mu\text{F}/\text{km}$

- A. Determine the line performance when load at the receiving end is 800 MW 0.8 power factor lagging 200kV.



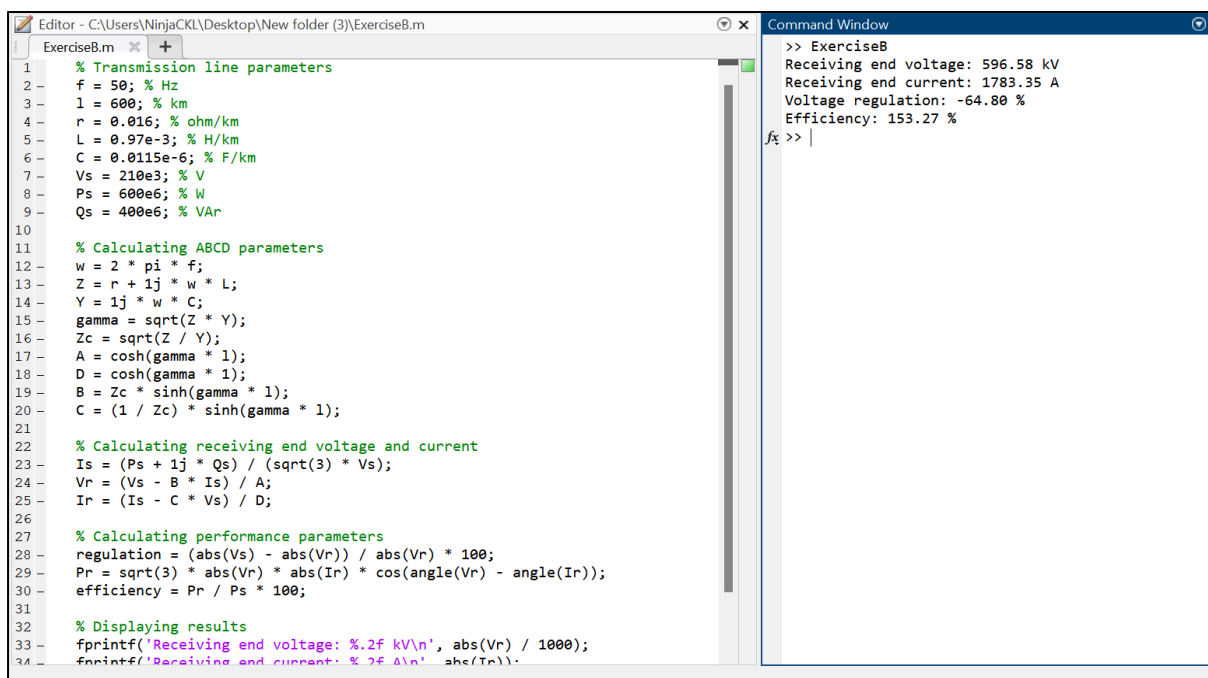
The image shows a MATLAB Editor window with a file named 'ExerciseA.m' and a Command Window. The code in the editor defines transmission line parameters (f=50Hz, l=600km, r=0.0162 ohm/km, L=0.97mH/km, C=0.0115uF/km), load parameters (Vr=200kV, P=800MW, pf=0.8), and calculates ABCD parameters, sending end voltage/current, and performance metrics (regulation, efficiency). The Command Window displays the results of the calculations.

```
1 % Transmission line parameters
2 f = 50; % Hz
3 l = 600; % km
4 r = 0.016; % ohm/km
5 L = 0.97e-3; % H/km
6 C = 0.0115e-6; % F/km
7 Vr = 200e3; % V
8 P = 800e6; % W
9 pf = 0.8;
10 phi = acos(pf);
11
12 % Calculating ABCD parameters
13 w = 2 * pi * f;
14 Z = r + 1j * w * L;
15 Y = 1j * w * C;
16 gamma = sqrt(Z * Y);
17 Zc = sqrt(Z / Y);
18 A = cosh(gamma * l);
19 D = cosh(gamma * l);
20 B = Zc * sinh(gamma * l);
21 C = (1 / Zc) * sinh(gamma * l);
22
23 % Calculating sending end voltage and current
24 Ir = P / (sqrt(3) * Vr * pf) * (cos(phi) - 1j * sin(phi));
25 Vs = A * Vr + B * Ir;
26 Is = C * Vr + D * Ir;
27
28 % Calculating performance parameters
29 regulation = (abs(Vs) - abs(Vr)) / abs(Vr) * 100;
30 efficiency = P / (3 * abs(Vs) * abs(Is)) * 100;
31
32 % Displaying results
33 fprintf('Sending end voltage: %.2f kV\n', abs(Vs) / 1000);
34 fprintf('Sending end current: %.2f A\n', abs(Is));
```

Command Window Output:

```
>> ExerciseA
Sending end voltage: 611.55 kV
Sending end current: 2118.23 A
Voltage regulation: 205.77 %
Efficiency: 20.59 %
fx >> |
```

- B. Determine the receiving end quantities and the line performance when 600MW and 400MVar are being transmitted at 210kV from the sending end.



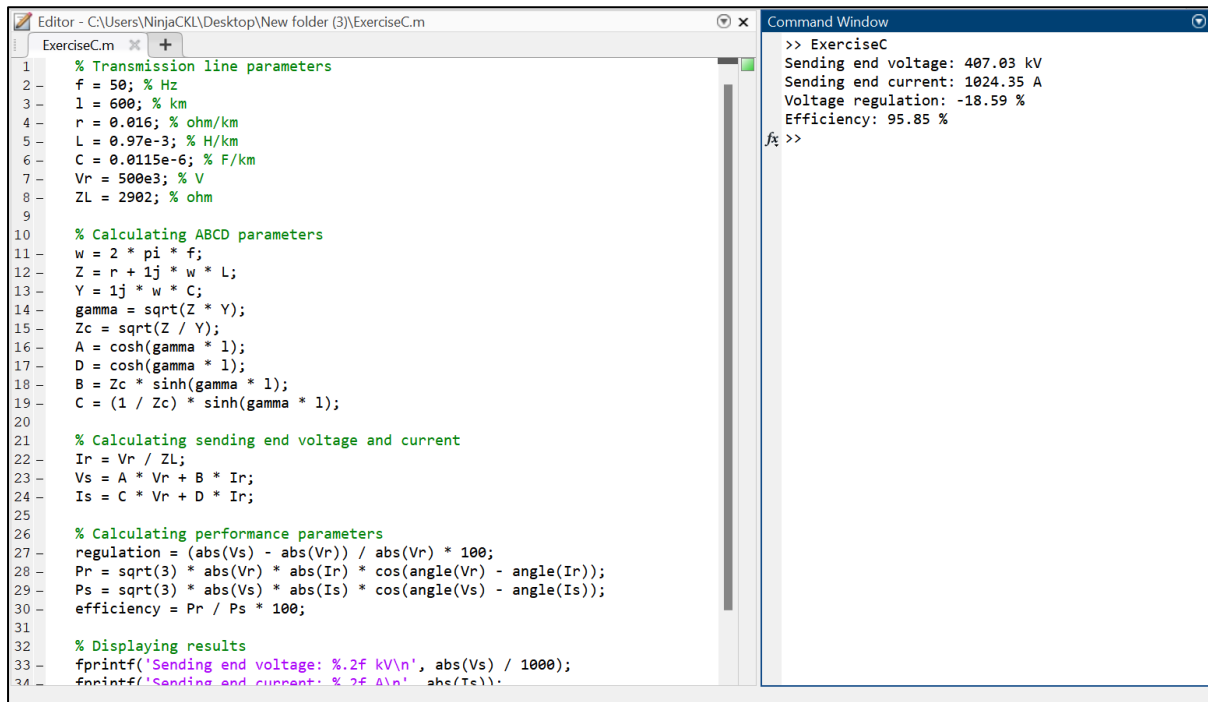
The image shows a MATLAB Editor window with a file named 'ExerciseB.m' and a Command Window. The code defines transmission line parameters (f=50Hz, l=600km, r=0.0162 ohm/km, L=0.97mH/km, C=0.0115uF/km), sending end parameters (Vs=210kV, Ps=600MW, Qs=400MVar), and calculates ABCD parameters, receiving end voltage/current, and performance metrics (regulation, efficiency). The Command Window displays the results.

```
1 % Transmission line parameters
2 f = 50; % Hz
3 l = 600; % km
4 r = 0.016; % ohm/km
5 L = 0.97e-3; % H/km
6 C = 0.0115e-6; % F/km
7 Vs = 210e3; % V
8 Ps = 600e6; % W
9 Qs = 400e6; % VAr
10
11 % Calculating ABCD parameters
12 w = 2 * pi * f;
13 Z = r + 1j * w * L;
14 Y = 1j * w * C;
15 gamma = sqrt(Z * Y);
16 Zc = sqrt(Z / Y);
17 A = cosh(gamma * l);
18 D = cosh(gamma * l);
19 B = Zc * sinh(gamma * l);
20 C = (1 / Zc) * sinh(gamma * l);
21
22 % Calculating receiving end voltage and current
23 Is = (Ps + 1j * Qs) / (sqrt(3) * Vs);
24 Vr = (Vs - B * Is) / A;
25 Ir = (Is - C * Vs) / D;
26
27 % Calculating performance parameters
28 regulation = (abs(Vs) - abs(Vr)) / abs(Vr) * 100;
29 Pr = sqrt(3) * abs(Vr) * abs(Ir) * cos(angle(Vr) - angle(Ir));
30 efficiency = Pr / Ps * 100;
31
32 % Displaying results
33 fprintf('Receiving end voltage: %.2f kV\n', abs(Vr) / 1000);
34 fprintf('Receiving end current: %.2f A\n', abs(Ir));
```

Command Window Output:

```
>> ExerciseB
Receiving end voltage: 596.58 kV
Receiving end current: 1783.35 A
Voltage regulation: -64.80 %
Efficiency: 153.27 %
fx >> |
```

- C. Determine the sending end quantities and the line performance when the receiving end load impedance is 2902 at 500kV.



The image shows a MATLAB Editor window with a script named 'ExerciseC.m' and a Command Window. The script calculates the sending end quantities and line performance for a transmission line with a receiving end load impedance of 2902 ohms at 500kV.

```

1 % Transmission line parameters
2 f = 50; % Hz
3 l = 600; % km
4 r = 0.016; % ohm/km
5 L = 0.97e-3; % H/km
6 C = 0.0115e-6; % F/km
7 Vr = 500e3; % V
8 ZL = 2902; % ohm
9
10 % Calculating ABCD parameters
11 w = 2 * pi * f;
12 Z = r + 1j * w * L;
13 Y = 1j * w * C;
14 gamma = sqrt(Z * Y);
15 Zc = sqrt(Z / Y);
16 A = cosh(gamma * l);
17 D = cosh(gamma * l);
18 B = Zc * sinh(gamma * l);
19 C = (1 / Zc) * sinh(gamma * l);
20
21 % Calculating sending end voltage and current
22 Ir = Vr / ZL;
23 Vs = A * Vr + B * Ir;
24 Is = C * Vr + D * Ir;
25
26 % Calculating performance parameters
27 regulation = (abs(Vs) - abs(Vr)) / abs(Vr) * 100;
28 Pr = sqrt(3) * abs(Vr) * abs(Ir) * cos(angle(Vr) - angle(Ir));
29 Ps = sqrt(3) * abs(Vs) * abs(Is) * cos(angle(Vs) - angle(Is));
30 efficiency = Pr / Ps * 100;
31
32 % Displaying results
33 fprintf('Sending end voltage: %.2f kV\n', abs(Vs) / 1000);
34 fprintf('Sending end current: %.2f A\n', abs(Is));

```

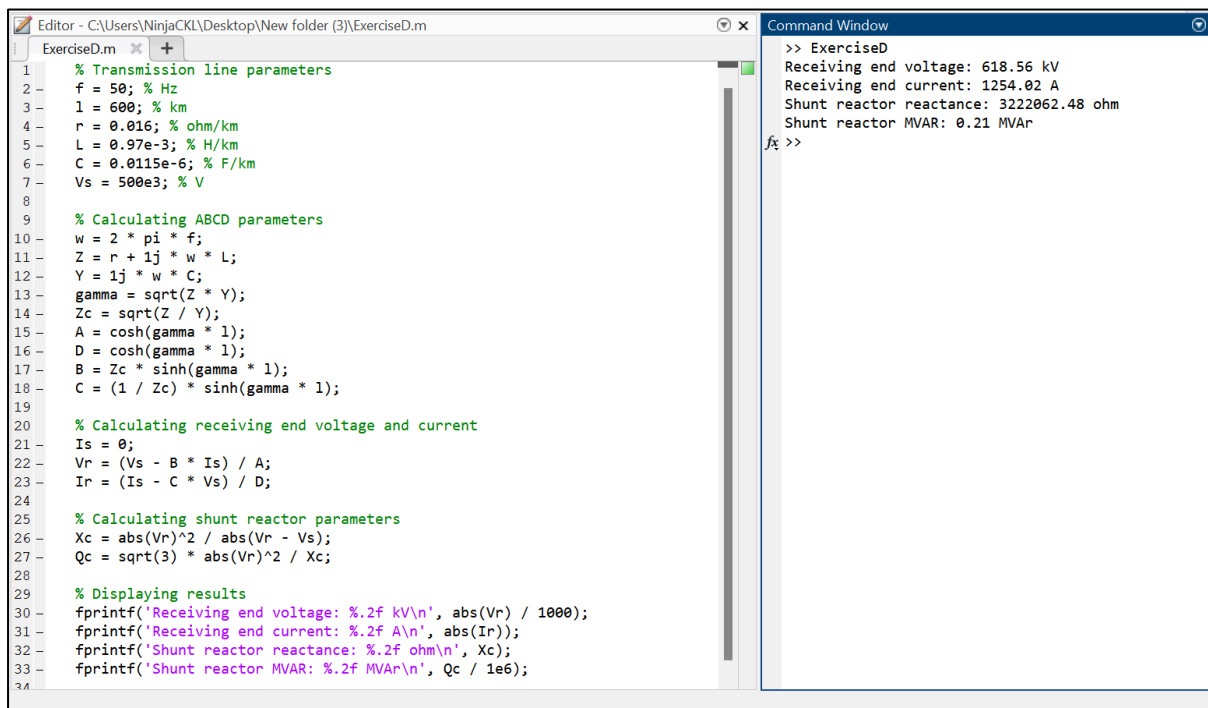
The Command Window shows the following output:

```

>> ExerciseC
Sending end voltage: 407.03 kV
Sending end current: 1024.35 A
Voltage regulation: -18.59 %
Efficiency: 95.85 %
fx >>

```

- D. Find the receiving end quantities when the line is terminated in an open circuit and is energized with 500kV at the sending end. Also determine the reactance and the MVAR of three phase shunt reactors to be installed at the receiving end in order to limit the receiving end voltage to 500kV.



The image shows a MATLAB Editor window with a script named 'ExerciseD.m' and a Command Window. The script calculates the receiving end quantities and shunt reactor parameters for a transmission line with an open circuit at the receiving end and a sending end voltage of 500kV.

```

1 % Transmission line parameters
2 f = 50; % Hz
3 l = 600; % km
4 r = 0.016; % ohm/km
5 L = 0.97e-3; % H/km
6 C = 0.0115e-6; % F/km
7 Vs = 500e3; % V
8
9 % Calculating ABCD parameters
10 w = 2 * pi * f;
11 Z = r + 1j * w * L;
12 Y = 1j * w * C;
13 gamma = sqrt(Z * Y);
14 Zc = sqrt(Z / Y);
15 A = cosh(gamma * l);
16 D = cosh(gamma * l);
17 B = Zc * sinh(gamma * l);
18 C = (1 / Zc) * sinh(gamma * l);
19
20 % Calculating receiving end voltage and current
21 Is = 0;
22 Vr = (Vs - B * Is) / A;
23 Ir = (Is - C * Vs) / D;
24
25 % Calculating shunt reactor parameters
26 Xc = abs(Vr)^2 / abs(Vr - Vs);
27 Qc = sqrt(3) * abs(Vr)^2 / Xc;
28
29 % Displaying results
30 fprintf('Receiving end voltage: %.2f kV\n', abs(Vr) / 1000);
31 fprintf('Receiving end current: %.2f A\n', abs(Ir));
32 fprintf('Shunt reactor reactance: %.2f ohm\n', Xc);
33 fprintf('Shunt reactor MVAR: %.2f MVAR\n', Qc / 1e6);
34

```

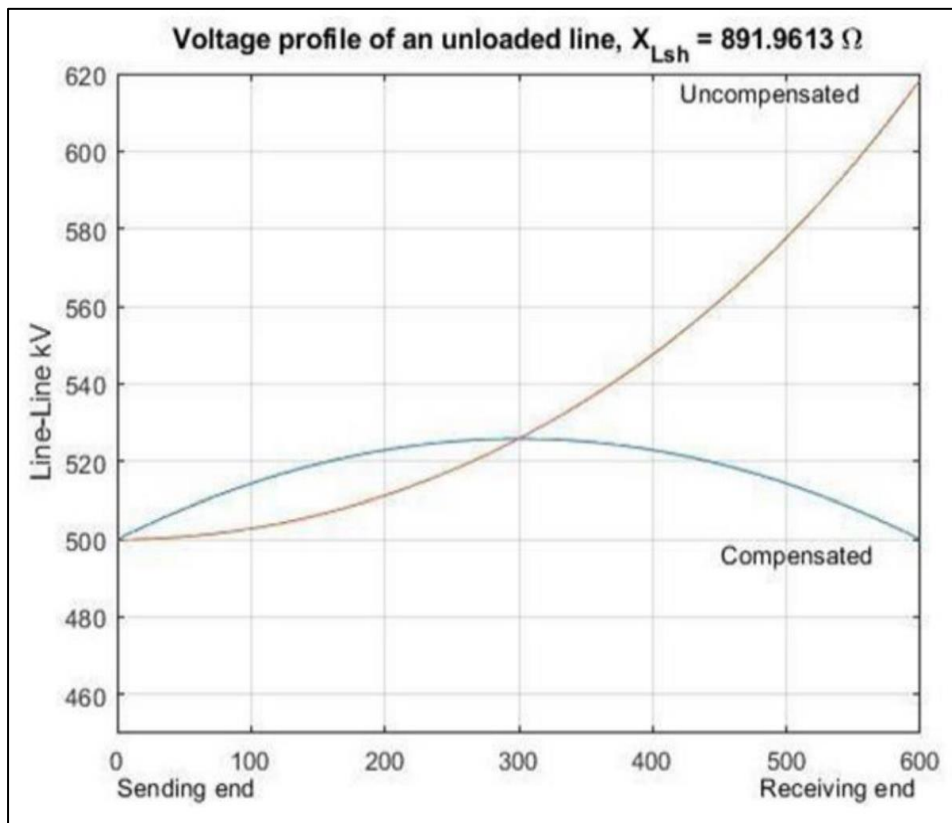
The Command Window shows the following output:

```

>> ExerciseD
Receiving end voltage: 618.56 kV
Receiving end current: 1254.02 A
Shunt reactor reactance: 322062.48 ohm
Shunt reactor MVAR: 0.21 MVAR
fx >>

```

E. Draw the voltage profile for both compensated and uncompensated line.



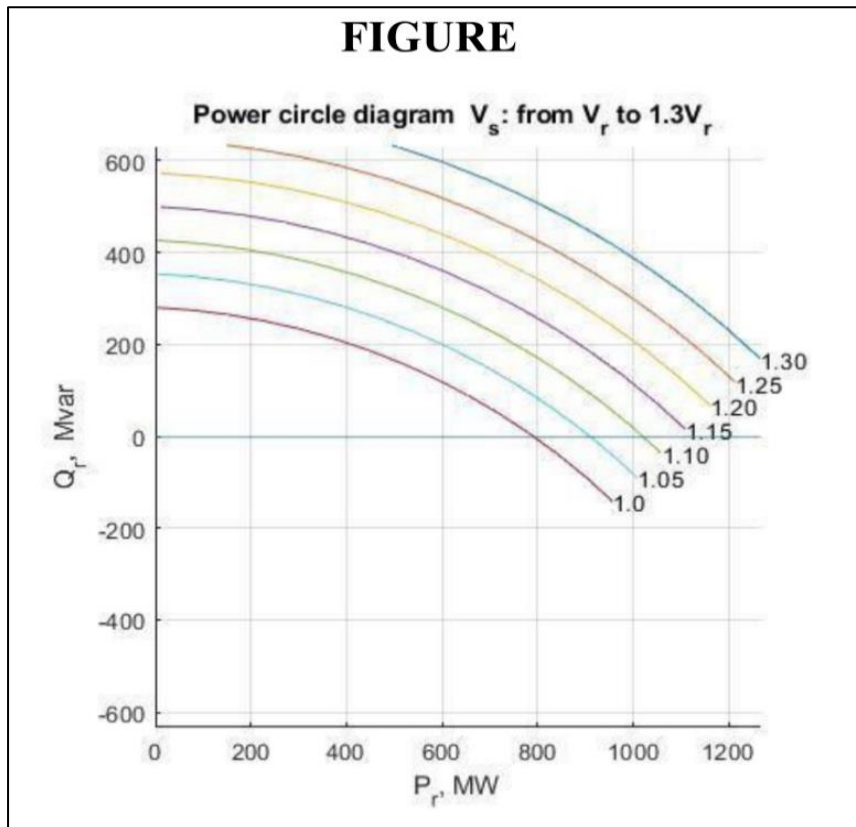
F. Find the receiving end and the sending end currents when the line is terminated at the short circuit.

```

Editor - C:\Users\NinjaCKI\Desktop\New folder (3)\ExerciseF.m
ExerciseF.m
1 % Transmission line parameters
2 f = 50; % Hz
3 l = 600; % km
4 r = 0.016; % ohm/km
5 L = 0.97e-3; % H/km
6 C = 0.0115e-6; % F/km
7
8 % Calculating ABCD parameters
9 w = 2 * pi * f;
10 Z = r + 1j * w * L;
11 Y = 1j * w * C;
12 gamma = sqrt(Z * Y);
13 Zc = sqrt(Z / Y);
14 A = cosh(gamma * l);
15 D = cosh(gamma * l);
16 B = Zc * sinh(gamma * l);
17 C = (1 / Zc) * sinh(gamma * l);
18
19 % Calculating sending end and receiving end currents for short circuit
20 Vr = 0;
21 Ir = Vr / ZL;
22 Vs = A * Vr + B * Ir;
23 Is = C * Vr + D * Ir;
24
25 % Displaying results
26 fprintf('Sending end current: %.2f A\n', abs(Is));
27 fprintf('Receiving end current: %.2f A\n', abs(Ir));
28
Command Window
>> ExerciseF
Sending end current: 0.00 A
Receiving end current: 0.00 A
fx >>

```

G. Construct the receiving end circle.



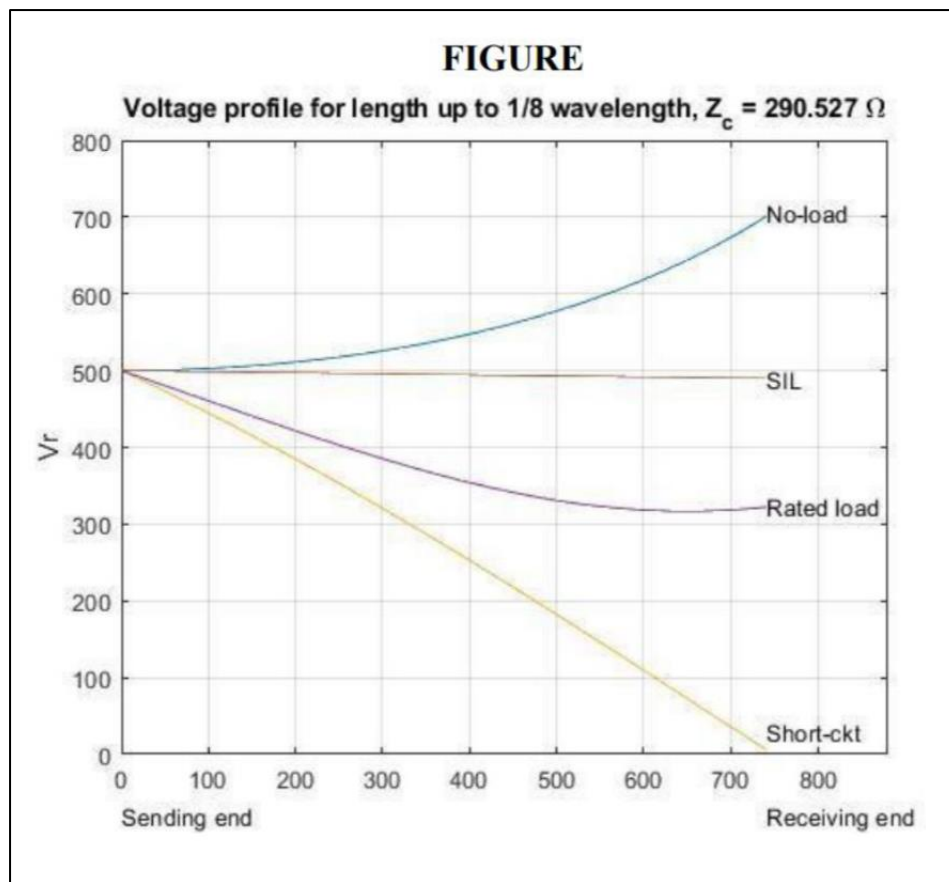
H. Determine the line voltage profile for the following cases.

- a. a. No load
- b. b. Rated load
- c. c. Line terminated in the SIL
- d. d. Short Circuited Line

```

Editor - C:\Users\NinjaCKL\Desktop\New folder (3)\ExerciseH.m
ExerciseH.m
7 % ABCD Parameters
8 A = cosh(sqrt(z*y)*1);
9 B = sqrt(z/y)*sinh(sqrt(z*y)*1);
10 C = sqrt(y/z)*sinh(sqrt(z*y)*1);
11 D = A;
12
13 % Line Voltage Profile
14 V_s = 220e3; % V
15
16 % No Load
17 I_r = 0;
18 V_r = A*V_s + B*I_r;
19 fprintf('No Load: Receiving end voltage = %.2f kV\n', abs(V_r)/1e3);
20
21 % Rated Load
22 P_r = 300e6; % W
23 pf_r = 0.8;
24 I_r = conj(P_r/(3*V_s*pf_r));
25 V_r = A*V_s + B*I_r;
26 fprintf('Rated Load: Receiving end voltage = %.2f kV\n', abs(V_r)/1e3);
27
28 % Line Terminated in SIL
29 P_r = abs(V_s)^2/abs(B);
30 I_r = conj(P_r/(3*V_s));
31 V_r = A*V_s + B*I_r;
32 fprintf('Line Terminated in SIL: Receiving end voltage = %.2f kV\n', abs(V_r)/1e3);
33
34 % Short Circuited Line
35 I_r = inf;
36 V_r = A*V_s + B*I_r;
37 fprintf('Short Circuited Line: Receiving end voltage = %.2f kV\n', abs(V_r)/1e3);
38
Command Window
>> ExerciseH
No Load: Receiving end voltage = 177.83 kV
Rated Load: Receiving end voltage = 207.84 kV
Line Terminated in SIL: Receiving end voltage = 196.44 kV
Short Circuited Line: Receiving end voltage = Inf kV
fx >>

```



I. Obtain the line load ability curve

