Scilab Manual for Electrical Power System II by Prof Mehulsinh Ghanshyamsinh Jadeja Electrical Engineering Vishwakarma Government Engineering College, Chandkheda[1]

Solutions provided by
Prof Mehulsinh Ghanshyamsinh Jadeja
Electrical Engineering
Vishwakarma Government Engineering College, Chandkheda

May 11, 2025

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Scilab Manual and Scilab codes written in it can be downloaded from the "Migrated Labs" section at the website http://scilab.in

Contents

List of Experiments

List of Figures

Experiment: 1

A Scilab Program Obtain Voltage Regulation And Efficiency Of A Short Transmission Line.

Scilab code Solution 1.1 exp1

```
//Experiment-1
// windows Vista - 32-Bit
//Scilab - 5.5.2


//Aim : A SCILAB program obtain voltage regulation
    and efficiency of a short transmission line.

clc;
clear all;
VRLL=220; VR = VRLL/sqrt(3);
Z = (0.15+%i*2*%pi*60*1.3263e-3)*40; //change
    frequency or length

disp('(a)')
```

```
SR=304.8+%i*228.6; // legging power factor
IR = conj(SR)/(3*conj(VR)) ;
IS = IR;
VS = VR + Z*IR;
VSLL = sqrt(3)*abs(VS)
SS = 3*VS*conj(IS)
REG = (VSLL - VRLL)/VRLL*100
Eff = real(SR)/real(SS)*100
mprintf('Sendingend line voltage = %f kV \n',VSLL)
mprintf('Sending Apprent power = %f MVA \n',SS)
mprintf('Regulation = %f %% \n',REG)
mprintf('Efficiency = %f %% \n',Eff)

disp('(b)')
SR=304.8-%i*228.6; // leading power factor
IR = conj(SR)/(3*conj(VR)); IS = IR;
VS = VR + Z*IR;
VSLL = sqrt(3)*abs(VS)
SS = 3*VS*conj(IS)
REG = (VSLL - VRLL)/VRLL*100
Eff = real(SR)/real(SS)*100
mprintf('Sendingend line voltage = %f kV \n',VSLL)
mprintf('Sending Apprent power = %f MVA \n',SS)
mprintf('Regulation = %f %% \n',REG)
mprintf('Efficiency = %f %% \n',Eff)

//output:-
//(a)
//Sendingend line voltage = 250.018650 kV
//Sending Apprent power = 322.795165 MVA
//Regulation = 13.644841 %
//Efficiency = 94.425206 %
//
// (b)
//Sendingend line voltage = 210.288383 kV
//Sending Apprent power = 322.795165 MVA
//Regulation = -4.414372 %
//Efficiency = 94.425206 %
```

```
 (a)
Sendingend line voltage = 250.018650 kV
Sending Apprent power = 322.795165 MVA
Regulation = 13.644841 %
Efficiency = 94.425206 %

 (b)
Sendingend line voltage = 210.288383 kV
Sending Apprent power = 322.795165 MVA
Regulation = -4.414372 %
Efficiency = 94.425206 %
```

Figure 1.1: exp1

```
//
```

Experiment: 2

A Scilab Program To Evaluate Performance Of Short Transmission Line Using Graphical Method

Scilab code Solution 2.2 exp2

```
//Experiment-2
// windows Vista - 32-Bit
//Scilab - 5.5.2


//Aim : A SCILAB program to evaluate performance of
   short transmission line using graphical method


clear all;
clc;
// This program calculates the percentage regulation
    of a short transmission line
// It is assumed that the receiving end voltage Vr
   and magnitude of current Ir are constant
```

```
Vr = 220e3; // receiving end voltage
R = 33.5;   // input line resistance
X = 138;    // line reactance
Z = R + X*%i;   // line impedance
Irmag = 164.02; // receiving end current magnitude
// This Irmag is assumed to be constant. This will change the angle of
// phasor of Ir in the vector diagram over the range of pf set in following line
            pf = 1:-0.001:0.5;   // power factor

            Irang = acos(pf);
            Ir = Irmag * exp(-%i*Irang);    // For lagging power factor
            Vs = (Vr/sqrt(3))+ Ir*Z;    // sqrt(3) is used here as we need to do a per phase analysis
            VrNL = abs(Vs);
VrFL = abs(Vr/sqrt(3));
per_vol_regu= ((VrNL-VrFL)/VrFL)*100;
plot(pf,per_vol_regu,'-');
xlabel('power factor')
ylabel('voltage regulation')
title('Graph of power factor v/s voltage regulation')

// For leading power factor range, angle should be positive So,

Ir = Irmag * exp(%i*Irang);

Vs = (Vr/sqrt(3))+ Ir*Z;

VrNL = abs(Vs); // absolute value of Vs

VrFL = abs(Vr/sqrt(3)); //abs value of Vr per phase

per_vol_regu= ((VrNL-VrFL)/VrFL)*100;
```

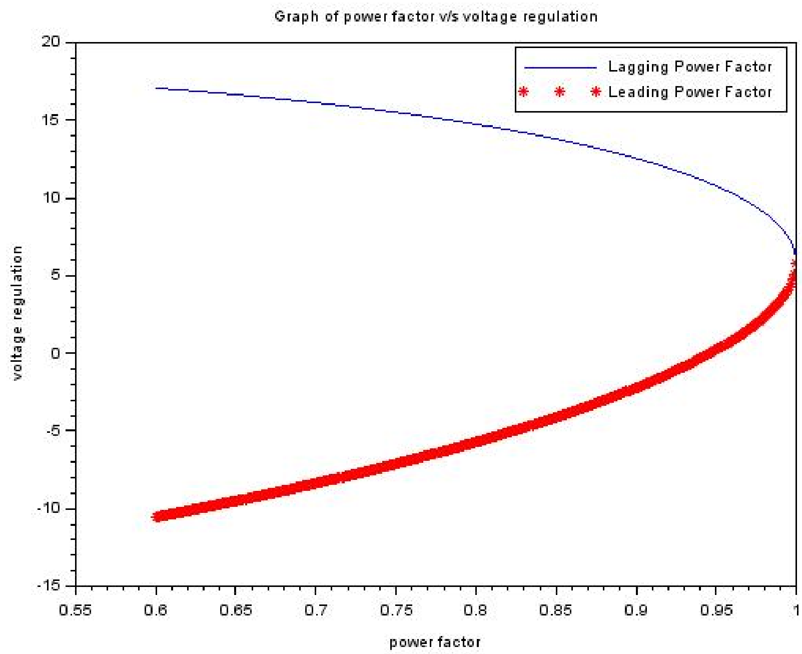


Figure 2.1: exp2

```
plot(pf,per_vol_regu,'r*');
legend('Lagging Power Factor','Leading Power Factor'
    )
//gtext('Vreg vs lagging pf')
//gtext('Vreg vs leading pf')
```

Experiment: 3

Finding Voltage Regulation And Efficiency Of A Medium Transmission Line Using Nominal T Method Through Scilab

Scilab code Solution 3.3 exp3

```
//Experiment-3
// windows Vista - 32-Bit
//Scilab - 5.5.2


// Aim ; Finding voltage regulation and efficiency
   of a medium transmission line using nominal T
   method through SCILAB

clc;
clear all;
R=10;
XL=20;
```

```
Y= 4e-4;
VRLL=66e3;
PR=10000e3;
pf=0.8; // change power factor of load
VR=66e3/sqrt(3);
Z=R+%i*XL;
IR=PR/(sqrt(3)*VRLL*pf);
IR_COMPLEX=IR*(cos(acos(0.8))-%i*sin(acos(0.8)));
VM=VR+IR_COMPLEX*(Z/2);
IC=%i*Y*VM;
IS=IR_COMPLEX+IC;
VS=VM+IS*(Z/2);
[VS_abs,Phase_VS]= polar(VS);
[IS_abs,Phase_IS]= polar(IS);
VS_abs;
IS_abs
VSLL=VS*sqrt(3);
VSLL_abs=abs(VSLL)/1000
PHASE_DIFF=Phase_VS-Phase_IS;
SendingEnd_PF=cos(PHASE_DIFF)
PS=3*abs(VS)*abs(IS)*cos(PHASE_DIFF);
EFF=(abs(PR)/abs(PS))*100
REG = 100*((abs (VSLL) - abs (VRLL)))/ abs (VRLL)
mprintf('Sendingend Current = %f A \n',IS_abs)
mprintf('Sendingend Line Voltage = %f kV \n',
   VSLL_abs)
mprintf('Sendingend Power factor = %f \n',
   SendingEnd_PF)
mprintf('Efficiency = %f %% \n', EFF)
mprintf('Regulation = %f %% \n', REG)


//Output:-
//
//Sendingend Current = 100.533214 A
//Sendingend Line Voltage = 69.543925 kV
//Sendingend Power factor = 0.853122
//Efficiency = 96.796480 %
```

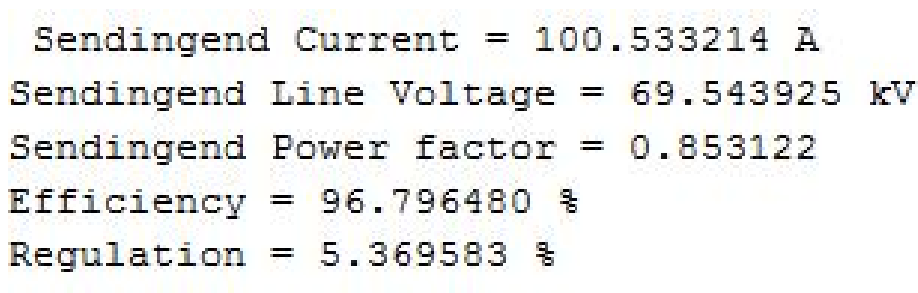

Sendingend Current = 100.533214 A
Sendingend Line Voltage = 69.543925 kV
Sendingend Power factor = 0.853122
Efficiency = 96.796480 %
Regulation = 5.369583 %

Figure 3.1: exp3

```
//Regulation = 5.369583 %
//
```

Experiment: 4

Finding Voltage Regulation And Efficiency Of A Medium Transmission Line Using Nominal Pi Method Through Scilab

Scilab code Solution 4.4 exp4

```
//Experiment-4
// windows Vista - 32-Bit
//Scilab - 5.5.2


// Aim : Finding voltage regulation and efficiency
   of a medium  transmission line using nominal PI
   method through SCILAB

mode(0);

clc
clear all
```

```
disp('Nominal PI Method for medium transmission line')
R = input('Enter the Total line Resistance in Ohms : '); // Total Line Resistance in Ohms like 10 Ohms
XL = input('Enter the Total line Inductive Reactance in Ohms : '); // Total line Inductive Reactance in Ohms like 50 Ohms
Y = input('Enter the Total line Suseptance in Seimen : '); // Total line Suseptance in Seimen like 10e-4 Seimen
f = input('Enter the Frequency in Hz: '); // Frequency in Hz like 50 Hz
PR= input('Enter the Recieving End Power in Watt : '); // Recieving End Power in Volt like 20e6 W
VRLL= input('Enter the Recieving End Line Voltage in Volt :'); // Recieving End Line Voltage in Volt like 66e3 V
pf= input('Enter Recievinf End Lagging Power Factor : '); // Recievinf End Lagging Power Factor like 0.9
Z=R+%i*XL;
VR = VRLL/sqrt(3);

IR=PR/(sqrt(3)*VRLL*pf);
IR_COMPLEX=IR*(cos(acos(pf))-%i*sin(acos(pf)));

IC1=%i*Y/2*VR;

IL = IR_COMPLEX + IC1;

VS = VR + IL*Z;

IC2 = %i*Y/2*VS;

IS=IL + IC2

[VS_abs,Phase_VS]= polar(VS);
```

```
[IS_abs,Phase_IS]= polar(IS);
VS_abs;
IS_abs;
VSLL=VS*sqrt(3);
VSLL_abs=abs(VSLL)/1000
PHASE_DIFF=Phase_VS-Phase_IS;
SendingEnd_PF=cos(PHASE_DIFF);

PS=3*abs(VS)*abs(IS)*cos(PHASE_DIFF)
EFF=(abs(PR)/abs(PS))*100
REG = 100*((abs (VSLL) - abs (VRLL)))/ abs (VRLL)

mprintf('Sendingend Current = %f A \n',IS_abs)
mprintf('Sendingend Line Voltage = %f kV \n', VSLL_abs)
mprintf('Sendingend Power factor = %f \n', SendingEnd_PF)
mprintf('Efficiency = %f %% \n', EFF)
mprintf('Regulation = %f %% \n', REG)


//output:-
//
//Nominal PI Method for medium transmission line
//Enter the Total line Resistance in Ohms : 10
//Enter the Total line Inductive Reactance in Ohms : 50
//Enter the Total line Suseptance in Seimen : 10e-4
//Enter the Frequency in Hz: 50
//Enter the Recieving End Power in Watt : 20e6
//Enter the Recieving End Line Voltage in Volt :66e3
//Enter Recievinf End Lagging Power Factor : 0.9
// IS =
//
//    170.90917 - 44.112456i
// VSLL_abs =
//
//    76.021348
```

```
// PS  =
//
//     21047697. + 1.094D-09i
// EFF  =
//
//     95.022274
// REG  =
//
//     15.183861
//Sendingend Current = 176.510208 A
//Sendingend Line Voltage = 76.021348 kV
//Sendingend Power factor = 0.905604
//Efficiency = 95.022274 %
//Regulation = 15.183861 %
```

```
 Nominal PI Method for medium transmission line
Enter the Total line Resistance in Ohms : 10
Enter the Total line Inductive Reactance in Ohms : 50
Enter the Total line Suseptance in Seimen : 10e-4
Enter the Frequency in Hz: 50
Enter the Recieving End Power in Watt : 20e6
Enter the Recieving End Line Voltage in Volt :66e3
Enter Recievinf End Lagging Power Factor : 0.9
 IS  =

    170.90917 - 44.112456i
 VSLL_abs  =

    76.021348
 PS  =

    21047697. + 1.094D-09i
 EFF  =

    95.022274
 REG  =

    15.183861
Sendingend Current = 176.510208 A
Sendingend Line Voltage = 76.021348 kV
Sendingend Power factor = 0.905604
Efficiency = 95.022274 %
Regulation = 15.183861 %
```

Figure 4.1: exp4

Experiment: 5

Finding Voltage Regulation And Efficiency Of A Medium Transmission Line Using End Condenser Method Through Scilab

Scilab code Solution 5.5 exp5

```
//Experiment-5
// windows Vista - 32-Bit
//Scilab - 5.5.2


// Aim : Finding voltage regulation and efficiency
   of a medium  transmission line using End
   Condenser method through SCILAB

mode(0);

clc
clear all
```

```
disp('End Condensor Method for medium transmission line')
R =25 // ('Enter the line Resistance in Ohms : ')
X =80 // ('Enter the line Inductive Reactance in Ohms : ')
Y =14e-4 // ('Enter the line Suseptance in Seimen : ')
f =50 // ('Enter the frequency : ')
PR=15e6 // ('Enter the Recieving End Power in Watt : ')
VRLine=66e3 // ('Enter the Recieving End Line Voltage in Volt :')
Pf=0.8 // ('Enter Recievinf End Lagging Power Factor : ')
// X = 2*%pi*f*L
Z = R + X*%i;
//Y = %i* 2 *%pi*f*C

IR = PR/(VRLine*Pf);

Pfang = acos(Pf);

IR_Complex= IR*(Pf - %i*sin(Pfang));

IC_Complex = %i*Y*VRLine;

IS = IR_Complex + IC_Complex

Voltage_drop = IS*Z;

VS = VRLine + IS*Z

[VS_abs,Phase_VS]= polar(VS);

[IS_abs,Phase_IS]= polar(IS);

VS_abs;
IS_abs;
```

```
PHASE_DIFF=Phase_VS-Phase_IS;
SendingEnd_PF=cos(PHASE_DIFF);
Sending_End_PF = abs(SendingEnd_PF)
PS=abs(VS)*abs(IS)*cos(PHASE_DIFF);
EFF=(abs(PR)/abs(PS))*100
REG = 100*((abs (VS) - abs (VRLine)))/ abs (VRLine)

//
//output:-
//
//End Condensor Method for medium transmission line
// R =
//
//     25.
// X =
//
//     80.
// Y =
//
//     0.0014
// f =
//
//     50.
// PR =
//
//     15000000.
// VRLine =
//
//     66000.
// Pf =
//
//     0.8
// IS =
//
//     227.27273 - 78.054545i
// VS =
//
```

```
//      77926.182 + 16230.455i
// Sending_End_PF  =
//
//      0.8596751
// EFF  =
//
//      91.220706
// REG  =
//
//      20.603752
```

```
End Condensor Method for medium transmission line
R  =

   25.
X  =

   80.
Y  =

   0.0014
f  =

   50.
PR  =

   15000000.
VRLine  =

   66000.
Pf  =

   0.8
IS  =

   227.27273 - 78.054545i
VS  =

   77926.182 + 16230.455i
Sending_End_PF  =

   0.8596751
EFF  =

   91.220706
REG  =

   20.603752
```

Figure 5.1: exp5

Experiment: 6

A Scilab Program To Evaluate Performance Of Medium Transmission Line Using Graphical Method

Scilab code Solution 6.6 exp6

```
//Experiment-6
// windows Vista - 32-Bit
//Scilab - 5.5.2


// Aim : A SCILAB program to evaluate performance of
    medium transmission line using graphical method

clear all;
clc;
// Program to calculate voltage regulation of
   nominal PI method of medium length transmission
   line
// Line parameters are taken from Stevenson as
   before. One may use any other data using the
```

```
%commented input functions
Vr = 220e3;   // input('Enter the value of Vr = ') ;
R =  33.5;   // input('Enter the value of R = ') ;
X = 138;   // input('Enter the value of X = ') ;
pf = 0.9;   // input('Enter the value of pf = ') ;
Pr = 50e6;   // input('Enter the value of pr = ') ;
Y = %i*0.3e-3; // input('Enter the value of Y = ') ;
Z = R + X * %i;
Irmag = Pr/(sqrt(3)*Vr*pf);
pf = 0.5:0.001:1;
Irang = acos (pf);  // Angle in radian
Ir =Irmag * exp (-%i*Irang);
// Irmag is kept constant. Only Ir phasor rotates. -ve sign is used because of lagging power factor %and to indicate that with Vr as reference phasor, angle of Ir should be negative.Irang should be %in radian only. It wil give wrong result, if Irang is in degrees

Ir1 =Irmag * exp (%i*Irang);    // For leading power factor
// Calculations for lagging power factor for medium line length
Vs = Vr/ sqrt (3) + (Ir+(Y/2)*Vr/ sqrt (3))*Z; // all terms are of voltage and are phase values
VrNLm =abs( Vs / (1+((Y*Z)/2)));
VrFLm= abs (Vr / sqrt (3));
per_vol_regulag = ( (VrNLm -VrFLm) / VrFLm) * 100;
// Calculations for leading power factor for medium line length
Vs1m = Vr/ sqrt (3) + (Ir1+(Y/2)*Vr/ sqrt (3))*Z;
// all terms are of voltage and are in phase values
VrNL1m =abs( Vs1m / (1+((Y*Z)/2)));
per_vol_regulead = ( (VrNL1m -VrFLm) / VrFLm) * 100;
// plot the two waveforms on the same graph
plot(pf,per_vol_regulag,'r',pf,per_vol_regulead,'*');

```

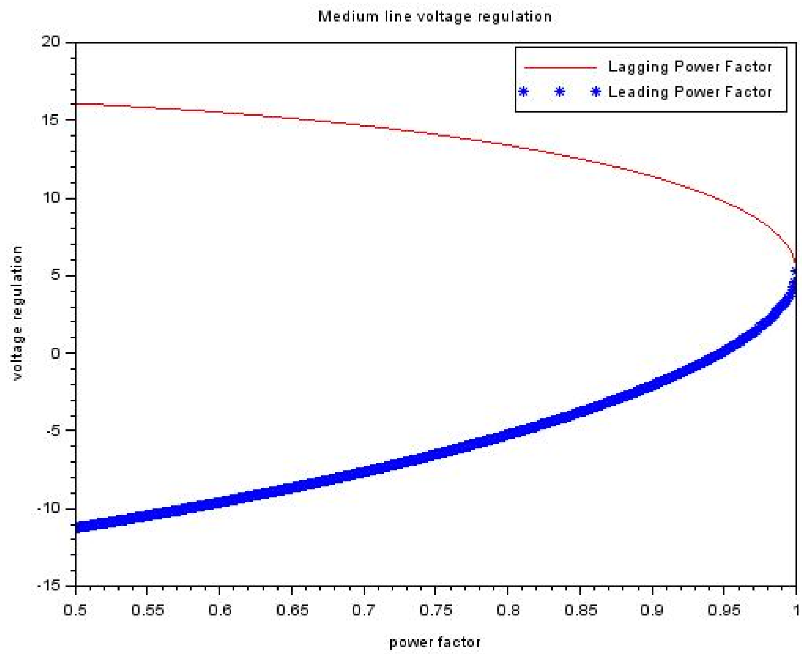


Figure 6.1: exp6

```
xlabel('power factor')
ylabel('voltage regulation')
title('Medium line voltage regulation')
legend('Lagging Power Factor','Leading Power Factor'
    )
// gtext('Vreg vs lagging pf')
// gtext('Vreg vs leading pf')
```

Experiment: 7

Finding voltage regulation and efficiency of a long transmission line using Rigorous solution method through SCILAB

Scilab code Solution 7.7 exp7

```
//Experiment-7
// windows Vista - 32-Bit
//Scilab - 5.5.2


// Aim : Finding voltage regulation and efficiency
   of a long transmission line using Rigorous
   solution method through SCILAB

mode(0);

clc
clear all
disp('Rigorous Method for Long transmission line')
R = input('Enter the Total line Resistance in Ohms :
```

```
'); // Total line Resistance in Ohms like 32 Ohms
XL = input('Enter the Total line Inductive Reactance in Ohms : '); // Total line Inductive Reactance in Ohms like 50 Ohms
y = input('Enter the Total line Suseptance in Seimen : '); // Total line Suseptance in Seimen like 0.0003
f = input('Enter the frequency in Hz : '); // frequency in Hz like 50 Hz
PR= input('Enter the Recieving End Power in Watt : ');// Recieving End Power in Watt like 20e6
VRLL= input('Enter the Recieving End Line Voltage in Volt :'); // Recieving End Line Voltage in Volt like 110e3
pf= input('Enter Recievinf End Lagging Power Factor : '); // Recievinf End Lagging Power Factor like 0.8
Z=R+%i*XL;
Y= %i*y;
VR = VRLL/sqrt(3);
IR=PR/(sqrt(3)*VRLL*pf);

A = 1+ (Z*Y)/2 + (Z*Z*Y*Y)/24;     //A= cosh(sqrt(Y*Z))

B = sqrt(Y*Z) + ((Y*Z)^(3/2))/6;     //B = sinh(sqrt(Y*Z))

VS = VR*A + IR*sqrt(Z/Y)*B;     //VS = VR*cosh(sqrt(Y*Z)) + IR*sqrt(Z/Y)*sinh(sqrt(Y*Z))

IS = VR*sqrt(Y/Z)*B + IR*A;    //IS = VR*sqrt(Y/Z)*sinh(sqrt(Y*Z)) + IR*cosh(sqrt(Y*Z))

[VS_abs,Phase_VS]= polar(VS);
[IS_abs,Phase_IS]= polar(IS);
VS_abs;
```

```
IS_abs;
VSLL=VS*sqrt(3);
VSLL_abs=abs(VSLL)/1000;
Phase_VS;
Phase_IS;

PHASE_DIFF=Phase_VS-Phase_IS;

SendingEnd_PF=cos(PHASE_DIFF);

PS=3*VS_abs*IS_abs*cos(PHASE_DIFF);
EFF=(abs(PR)/abs(PS))*100;
REG = 100*((abs (VSLL) - abs (VRLL)))/ abs (VRLL);

mprintf('\n\nSendingend Current = %f A \n',IS_abs)
mprintf('Sendingend Line Voltage = %f kV \n',VSLL_abs)
mprintf('Sendingend Power factor = %f \n',SendingEnd_PF)
mprintf('Efficiency = %f %% \n', EFF)
mprintf('Regulation = %f %% \n', REG)


//output:-
//
// Rigorous Method for Long transmission line
//Enter the Total line Resistance in Ohms : 32
//Enter the Total line Inductive Reactance in Ohms : 50
//Enter the Total line Suseptance in Seimen : 0.0003
//Enter the frequency in Hz : 50
//Enter the Recieving End Power in Watt : 20e6
//Enter the Recieving End Line Voltage in Volt :110e3
//Enter Recievinf End Lagging Power Factor : 0.8
//
//
//Sendingend Current = 131.674018 A
```

```
 Rigorous Method for Long transmission line
Enter the Total line Resistance in Ohms : 32
Enter the Total line Inductive Reactance in Ohms : 50
Enter the Total line Suseptance in Seimen : 0.0003
Enter the frequency in Hz : 50
Enter the Recieving End Power in Watt : 20e6
Enter the Recieving End Line Voltage in Volt :110e3
Enter Recievinf End Lagging Power Factor : 0.8


Sendingend Current = 131.674018 A
Sendingend Line Voltage = 117.015932 kV
Sendingend Power factor = 0.998847
Efficiency = 75.028310 %
Regulation = 6.378120 %
```

Figure 7.1: exp7

```
//Sendingend Line Voltage = 117.015932 kV
//Sendingend Power factor = 0.998847
//Efficiency = 75.028310 %
//Regulation = 6.378120 %
```

Experiment: 8

A Scilab Program To Evaluate Performance Of Long Transmission Line Using Graphical Method

Scilab code Solution 8.8 exp8

```
//Experiment-8
// windows Vista - 32-Bit
//Scilab - 5.5.2


// Aim : A SCILAB program to evaluate performance of
    long transmission line using graphical method

clc
clear all
disp('Rigorous Method for Long transmission line
   using graphical method')
R =32 // ('Enter the Total line Resistance in Ohms :
    ')
XL =50 // ('Enter the Total line Inductive Reactance
```

```
in Ohms : ')
y =0.003 // ('Enter the Total line Suseptance in Seimen : ')
f =50 // ('Enter the frequency : ')
PR= 20e6 //('Enter the Recieving End Power in Watt : ')
VRLL=110e3 // ('Enter the Recieving End Line Voltage in Volt :')
pf=0.8,
Z=R+%i*XL;
Y= %i*y;
VR = VRLL/sqrt(3);
IR=PR/(sqrt(3)*VRLL*pf);
A = 1+ (Z*Y)/2 + (Z*Z*Y*Y)/24;     //A= cosh(sqrt(Y*Z))

B = sqrt(Y*Z) + ((Y*Z)^(3/2))/6;     //B = sinh(sqrt(Y*Z))

pf = 1:-0.01:0.5


Irang = acos (pf);  // Angle in radian

Ir =IR * exp (-%i*Irang);
Ir1 =IR * exp (%i*Irang);


VS = VR*A + Ir*sqrt(Z/Y)*B;     //VS = VR*cosh(sqrt(Y*Z)) + IR*sqrt(Z/Y)*sinh(sqrt(Y*Z))

VS1 = VR*A + Ir1*sqrt(Z/Y)*B ;    //VS = VR*cosh(sqrt(Y*Z)) + IR*sqrt(Z/Y)*sinh(sqrt(Y*Z))


[VS_abs,Phase_VS]= polar(VS);
[VS1_abs,Phase_VS1]= polar(VS1);

```

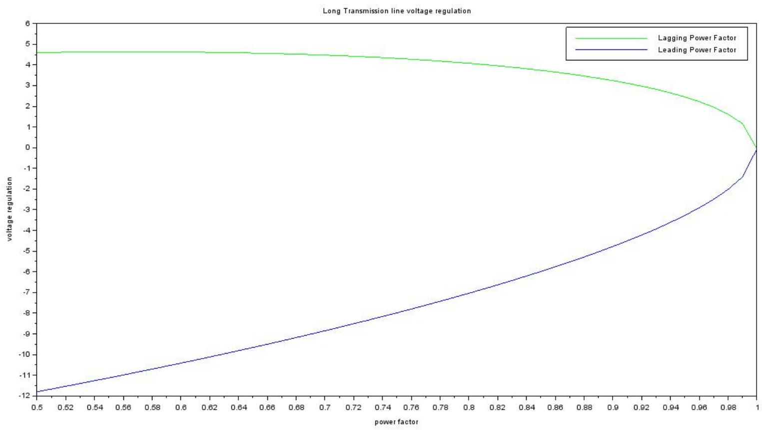


Figure 8.1: exp8

```
VSLL=VS;
VSLL1=VS1;
REGlag = 100*((abs (VSLL) - abs (VR)))/ abs (VR)
REGlead = 100*((abs (VSLL1) - abs (VR)))/ abs (VR)


plot(pf,REGlag,'g--');
plot(pf,REGlead,'b');

xlabel('power factor')
ylabel('voltage regulation')
title('Long Transmission line voltage regulation')
legend('Lagging Power Factor','Leading Power Factor')
```

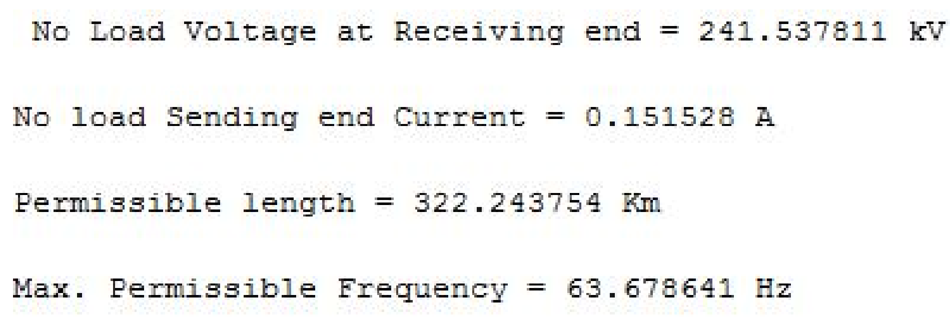
Experiment: 9

A Scilab Program To Find Permissible Frequency And Length For Limiting Ferranti Effect In Long Transmission Lines.

Scilab code Solution 9.9 exp9

```
//Experiment-9
// windows Vista - 32-Bit
//Scilab - 5.5.2


// Aim : A SCILAB Program to find permissible
    frequency and length for limiting Ferranti effect
     in long transmission lines.


clc;
clear all; // Clear All Variables & Command Window
Standard_f=50;
```

```
l=400;
VSLL=220e3;     //Defining Given DATA
r=0.125; //Defining Given DATA
x=0.4;  //Defining Given DATA
y=%i*2.8e-6;          //Defining Given DATA
R=r*l;
X=x*l;
Y=y*l;
z=r+%i*x;
Z=R+%i*X;
A=1+Y*Z/2; // Defining ABCD Parameters
B=Z*(1+Y*Z/6);             // Defining ABCD Parameters
C=Y*(1+Y*Z/6);  // Defining ABCD Parameters
D=A;                       // Defining ABCD Parameters

No_load_VRLL=VSLL/A;
NoLoad_VRLL=abs(No_load_VRLL)/1000        //Display No Load Voltage at Receiving
NoLoad_VR=NoLoad_VRLL/sqrt(3);        // Calculate Voltage per Phase
NoLoad_IS=abs(C*NoLoad_VR)            // Calculate Sending end Current

VRLL_MAX=235e3;                       // Define limit , the receiving end Volt
new_A=VSLL/VRLL_MAX;                  // Evaluate Parameter   A   for VRLL_MAX
l=sqrt(real((2*new_A-2)/y/z));        // Equation for maximum Length
Permisible_Lenth=abs(l)               // Find the Magnitude out of Complex

VRLL_MAX2=250e3;                      //  Similar to above
new_A2=VSLL/VRLL_MAX2;
Max_Permisible_f=abs((2500*sqrt(R^2*Y^2/10000+X*Y*(new_A2-1)/1250)-25*R*Y)/X/Y)
```


No Load Voltage at Receiving end = 241.537811 kV

No load Sending end Current = 0.151528 A

Permissible length = 322.243754 Km

Max. Permissible Frequency = 63.678641 Hz

Figure 9.1: exp9

```
mprintf('No Load Voltage at Receiving end = %f kV \n
   \n', NoLoad_VRLL)
mprintf('No load Sending end Current = %f A \n\n',
   NoLoad_IS)
mprintf('Permissible length = %f Km \n\n',
   Permisible_Lenth)
mprintf('Max. Permissible Frequency = %f Hz \n\n',
   Max_Permisible_f)


//output:-
//
// No Load Voltage at Receiving end = 241.537811 kV
//
//No load Sending end Current = 0.151528 A
//
//Permissible length = 322.243754 Km
//
//Max. Permissible Frequency = 63.678641 Hz
```

Experiment: 10

A Scilab Program For Computing The Abcd Parameters Of Short, Medium And Long Transmission Lines

Scilab code Solution 10.10 exp10

```
//Experiment-10
// windows Vista - 32-Bit
//Scilab - 5.5.2


// Aim : A SCILAB program for computing the ABCD
   parameters of short, medium and long transmission
    lines

clc
clear all
disp('Enter your choice:')
disp('1 : for short transmission line')
disp('2 : for medium transmission line')
disp('3 : for long transmission line')
```

```
check = input('Enter your choice : ')
R = input('Enter the line Resistance in Ohms : ')
L = input('Enter the line Inductance in Henry : ')
C = input('Enter the line Capacitance in Farad : ')
f = input('Enter the frequency : ')
X = 2*%pi*f*L
Z = R + %i*X;
Y = %i*2*%pi*f*C
if(check==1)
Adata = 1; // ABCD in cartesian form
Bdata = Z; // ABCD in cartesian form
Cdata = 0; // ABCD in cartesian form
Ddata = 1; // ABCD in cartesian form
Result = [Adata, Bdata, Cdata, Ddata]
mprintf('ABCD constant for Short Transmission line\n
     A constant  = %f \n  B constant= %f + i %f  ohm
    \n C constant=%f  mho \n D constant = %f \n',
   Adata,real(Bdata),imag(Bdata),Cdata,Ddata)
end

if(check==2)
disp(' Enter 1 for Nominal T-method')
disp('Enter 2 for Nominal PI-method')
s = input('Enter your choice : ')
if(s==1)
Adata = 1 + (Y*Z)/2;
Bdata = Z*(1 + (Y*Z)/4);
Cdata = Y;
Ddata = (1 + (Y*Z)/2);

Result = [Adata, Bdata, Cdata, Ddata]
mprintf('ABCD constant for Medium Transmission line
   Nominal T-method \n  A constant  = %f + i %f \n
   B constant= %f + i%f ohm \n C constant=%f + i %f
    mho \n D constant = %f  + i%f \n',real(Adata),
   imag(Adata),real(Bdata),imag(Bdata),real(Cdata),
   imag(Cdata),real(Ddata),imag(Ddata))
else
```

```
Adata = 1 + (Y*Z)/2;
Bdata = Z
Cdata = Y * (1 + (Y*Z)/4);
Ddata = (1 + (Y*Z)/2);
Result = [Adata, Bdata, Cdata, Ddata]
mprintf('ABCD constant for Medium Transmission line Nominal PI-method\n A constant = %f + i %f \n B constant= %f + i%f ohm \n C constant=%f + i %f mho \n D constant = %f + i%f \n',real(Adata),imag(Adata),real(Bdata),imag(Bdata),real(Cdata),imag(Cdata),real(Ddata),imag(Ddata))
end end

if(check==3)
Adata = cosh(sqrt(Y*Z));
Bdata = (sqrt(Z/Y) * sinh(sqrt(Y*Z)));
Cdata = (1/sqrt(Z/Y) * sinh(sqrt(Y*Z)));
Ddata = cosh(sqrt(Y*Z));
Result = [Adata,Bdata,Cdata,Ddata];
mprintf('ABCD constant for Long Transmission line \n A constant = %f + i %f \n B constant= %f + i%f ohm \n C constant=%f + i %f mho \n D constant = %f + i%f \n',real(Adata),imag(Adata),real(Bdata),imag(Bdata),real(Cdata),imag(Cdata),real(Ddata),imag(Ddata))
end

//
//output:-
//
//For Short Transmission line
//
//Enter your choice:
//
// 1 : for short transmission line
//
// 2 : for medium transmission line
//
```

```
// 3 : for long transmission line
//Enter your choice : 1
//Enter the line Resistance in Ohms : 20
//Enter the line Inductance in Henry : 50
//Enter the line Capacitance in Farad : 0.0003
//Enter the frequency : 50
//ABCD constant for Short Transmission line
//  A constant  = 1.000000
//  B constant= 20.000000 + i 15707.963268  ohm
// C constant=0.000000  mho
// D constant = 1.000000
//
//
//For Nominal T method of medium Transmission line
//
//Enter your choice:
//
// 1 : for short transmission line
//
// 2 : for medium transmission line
//
// 3 : for long transmission line
//Enter your choice : 2
//Enter the line Resistance in Ohms : 20
//Enter the line Inductance in Henry : 50
//Enter the line Capacitance in Farad : 0.0003
//Enter the frequency : 50
//
//  Enter 1 for Nominal T-method
//
// Enter 2 for Nominal PI-method
//Enter your choice : 1
//ABCD constant for Medium Transmission line Nominal
    T-method
//  A constant  = -739.220330 + i 0.942478
//  B constant= -14784.406602 + i-5797959.489510 ohm
// C constant=0.000000 + i 0.094248  mho
// D constant = -739.220330  + i0.942478
```

```
//
//For Nominal PI method of medium Transmission line
//
//Enter your choice:
//
// 1 : for short transmission line
//
// 2 : for medium transmission line
//
// 3 : for long transmission line
//Enter your choice : 2
//Enter the line Resistance in Ohms : 20
//Enter the line Inductance in Henry : 50
//Enter the line Capacitance in Farad : 0.0003
//Enter the frequency : 50
//
//  Enter 1 for Nominal T-method
//
// Enter 2 for Nominal PI-method
//Enter your choice : 2
//ABCD constant for Medium Transmission line Nominal
    PI-method
//  A constant  = -739.220330 + i 0.942478
//  B constant= 20.000000 + i15707.963268 ohm
// C constant=-0.044413 + i -34.787813  mho
// D constant = -739.220330  + i0.942478
//
//
//For long Transmission line
//
//Enter your choice:
//
// 1 : for short transmission line
//
// 2 : for medium transmission line
//
// 3 : for long transmission line
//Enter your choice : 3
```

```
//Enter the line Resistance in Ohms : 20
//Enter the line Inductance in Henry : 50
//Enter the line Capacitance in Farad : 0.0003
//Enter the frequency : 50
//ABCD constant for Long Transmission line
// A constant  = 0.712960 + i 0.017183
//  B constant= 7.310527 + i286.435826 ohm
// C constant=0.000042 + i 0.001719  mho
// D constant = 0.712960  + i0.017183
```

```
 Enter your choice:

 1 : for short transmission line

 2 : for medium transmission line

 3 : for long transmission line
Enter your choice : 1
Enter the line Resistance in Ohms : 20
Enter the line Inductance in Henry : 50
Enter the line Capacitance in Farad : 0.0003
Enter the frequency : 50
ABCD constant for Short Transmission line
  A constant  = 1.000000
  B constant= 20.000000 + i 15707.963268  ohm
 C constant=0.000000  mho
 D constant = 1.000000
```

```
 Enter your choice:

 1 : for short transmission line

 2 : for medium transmission line

 3 : for long transmission line
Enter your choice : 2
Enter the line Resistance in Ohms : 20
Enter the line Inductance in Henry : 50
Enter the line Capacitance in Farad : 0.0003
Enter the frequency : 50

  Enter 1 for Nominal T-method

 Enter 2 for Nominal PI-method
Enter your choice : 1
ABCD constant for Medium Transmission line Nominal T-method
  A constant  = -739.220330 + i 0.942478
  B constant= -14784.406602 + i-5797959.489510 ohm
 C constant=0.000000 + i 0.094248  mho
 D constant = -739.220330  + 10.942478
```


Figure 10.1: exp10

```
 Enter your choice:

 1 : for short transmission line

 2 : for medium transmission line

 3 : for long transmission line
Enter your choice : 2
Enter the line Resistance in Ohms : 20
Enter the line Inductance in Henry : 50
Enter the line Capacitance in Farad : 0.0003
Enter the frequency : 50

  Enter 1 for Nominal T-method

 Enter 2 for Nominal PI-method
Enter your choice : 2
ABCD constant for Medium Transmission line Nominal PI-method
  A constant  = -739.220330 + i 0.942478
  B constant= 20.000000 + i15707.963268 ohm
 C constant=-0.044413 + i -34.787813  mho
 D constant = -739.220330  + i0.942478
```

```
 Enter your choice:

 1 : for short transmission line

 2 : for medium transmission line

 3 : for long transmission line
Enter your choice : 3
Enter the line Resistance in Ohms : 20
Enter the line Inductance in Henry : 50
Enter the line Capacitance in Farad : 0.0003
Enter the frequency : 50
ABCD constant for Long Transmission line
 A constant  = 0.712960 + i 0.017183
  B constant= 7.310527 + i286.435826 ohm
 C constant=0.000042 + i 0.001719  mho
 D constant = 0.712960  + i0.017183
```


Figure 10.2: exp10

Experiment: 11

A Scilab Program For Conversion Of Unsymmetrical Ac Phasors Into Symmetrical Components

Scilab code Solution 11.11 exp11

```
//Experiment-11
// windows Vista - 32-Bit
//Scilab - 5.5.2


// Aim : A SCILAB program for conversion of
   unsymmetrical AC phasors into symmetrical
   components



clc

disp('Enter your choice:');

```

```
disp('1 for phasor to symmetrical;  2 for symmetrical to phasor');
choice=input('Enter your transformation type=');

alpha = -0.5+%i*0.866;
if(choice==1)

Va=input('Enter the value of complex phase voltage VA=') //complex phase voltage VA=200 - %i*100
Vb=input('Enter the value of complex phase voltage VB=') //complex phase voltage VB=-36.59 - %i*209.804
Vc=input('Enter the value of complex phase voltage VC=') //Enter the value of complex phase voltage VC=136.59 +%i*309.80
disp('Positive sequence component =');
Va1=(Va+(alpha*Vb)+(alpha)^2*Vc)/3
mprintf('Va1 = %f + i%f \n',real(Va1),imag(Va1))

disp('Negative sequence component =');
Va2=[Va+(alpha^2)*Vb+(alpha*Vc)]/3
mprintf('Va2 = %f + i%f \n',real(Va2),imag(Va2))

disp('Zero sequence component =');
Va0=[Va+Vb+Vc]/3
mprintf('Va0 = %f + i%f \n',real(Va0),imag(Va0))
end

if(choice==2)

Va1=input('Enter the value of positive sequence voltage Va1='); // positive sequence voltage Va1=200 - %i*100
Va2=input('Enter the value of negative sequence voltage Va2='); //negative sequence voltage Va2=-100
Va0=input('Enter the value of zero sequence voltage Va0='); //zero sequence voltage Va0=100
```

```
disp('Complex voltage Va =');

Va=[Va1+Va2+Va0]
mprintf('Va = %f + i%f \n',real(Va),imag(Va))

disp('Complex Voltage Vb=');
Vb=[(alpha^2*Va1)+(alpha*Va2)+Va0]
mprintf('Vb = %f + i%f \n',real(Vb),imag(Vb))

disp('Complex Voltage Vc=');
Vc=[(alpha*Va1)+(alpha^2*Va2)+Va0]
mprintf('Vc = %f + i%f \n',real(Vc),imag(Vc))
end

//
//output:-
//
//for phasor to symmetrical :
//
//Enter your choice:
//
// 1 for phasor to symmetrical;  2 for symmetrical
    to phasor
//Enter your transformation type=1
//Enter the value of complex phase voltage VA=200 -
    %i*100
//Enter the value of complex phase voltage VB=-36.59
     - %i*209.804
//Enter the value of complex phase voltage VC=136.59
     +%i*309.80
//
// Positive sequence component =
//Va1 = 199.994358 + i-99.986083
//
// Negative sequence component =
//Va2 = -99.992891 + i-0.011117
//
// Zero sequence component =
```

```
//Va0 = 100.000000 + i -0.001333
//
//
//
//for symmetrical to phasor :
//
//Enter your choice:
//
// 1 for phasor to symmetrical;  2 for symmetrical
    to phasor
//Enter your transformation type=2
//Enter the value of positive sequence voltage Va1
    =200 - %i*100
//Enter the value of negative sequence voltage Va2
    =-100
//Enter the value of zero sequence voltage Va0=100
//
// Complex voltage Va =
//Va = 200.000000 + i -100.000000
//
// Complex Voltage Vb=
//Vb = -36.591200 + i -209.804400
//
// Complex Voltage Vc=
//Vc = 136.595600 + i309.800000
```

```
 Enter your choice:

 1 for phasor to symmetrical;  2 for symmetrical to phasor
Enter your transformation type=1
Enter the value of complex phase voltage VA=200 - %i*100
Enter the value of complex phase voltage VB=-36.59 - %i*209.804
Enter the value of complex phase voltage VC=136.59 +%i*309.80

 Positive sequence component =
Va1 = 199.994358 + i-99.986083

 Negative sequence component =
Va2 = -99.992891 + i-0.011117

 Zero sequence component =
Va0 = 100.000000 + i-0.001333
```

Figure 11.1: exp11

```
 Enter your choice:

 1 for phasor to symmetrical;  2 for symmetrical to phasor
Enter your transformation type=2
Enter the value of positive sequence voltage Va1=200 - %i*100
Enter the value of negative sequence voltage Va2=-100
Enter the value of zero sequence voltage Va0=100

 Complex voltage Va =
Va = 200.000000 + i-100.000000

 Complex Voltage Vb=
Vb = -36.591200 + i-209.804400

 Complex Voltage Vc=
Vc = 136.595600 + i309.800000
```

Figure 11.2: exp11

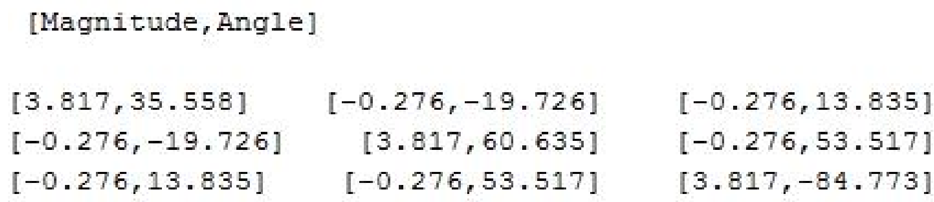
Experiment: 12

A Scilab Program For Conversion Of Symmetrical Components Into Ac Phasors

Scilab code Solution 12.12 exp12

```
//Experiment-12
// windows Vista - 32-Bit
//Scilab - 5.5.2


// Aim : A SCILAB program for conversion of
   symmetrical components into AC phasors


// To find Symmetrical Components for 3-phase
   Unbalanced System


clc;
clear all;

Q=[3 310
```

```
    5  105
    4  12
         ]
i=sqrt(-1);
n=3; // Number of Phases
a=1.0*(cos((360/n)*%pi/180)+%i*sin((360/n)*%pi/180));
a2=a^2;
// Conversion from Polar to Cartesian Form of Matrix I
Ia=Q(1)*(cos(Q(4)*%pi/180)+%i*sin(Q(4)*%pi/180));
Ib=Q(2)*(cos(Q(5)*%pi/180)+%i*sin(Q(5)*%pi/180));
Ic=Q(3)*(cos(Q(6)*%pi/180)+%i*sin(Q(6)*%pi/180));
// Define symmetrical components transformation matrix
A=  [1    1     1
      1    a2    a
      1    a     a2];
A012=inv(A)*[Ia;Ib;Ic];
// To check the results
Add0=A012(1)+A012(2)+A012(3); //This is Ia
Add1=A012(1)+a2*A012(2)+a*A012(3); //This is Ib
Add2=A012(1)+a*A012(2)+a2*A012(3); // This is Ic
// Symmetrical Components in Cartesian Form
Symm3= [A012(1) A012(2) A012(3)
        A012(1) a2*A012(2) a*A012(3)
        A012(1) a*A012(2) a2*A012(3)];
// Symmetrical Components in Polar Form
// Sym3Abs=abs(Symm3)
[sym3abs,Sym3Angle]=polar(Symm3)
sym3angle= Sym3Angle*180/%pi
disp('[Magnitude,Angle]')
mprintf('\n')
for i=1:3
    for j=1:3
        mprintf('[%.3f,%.3f]     ',sym3abs(i,j),sym3angle(i,j))
    end
```



```
[Magnitude,Angle]

[3.817,35.558]    [-0.276,-19.726]    [-0.276,13.835]
[-0.276,-19.726]    [3.817,60.635]    [-0.276,53.517]
[-0.276,13.835]    [-0.276,53.517]    [3.817,-84.773]
```

Figure 12.1: exp12

```
    mprintf('\n')
end

// End of the Program


//output:-
//
//[Magnitude,Angle]
//
//[3.817,35.558]      [-0.276,-19.726]      [-0.276,13.835]
//[-0.276,-19.726]      [3.817,60.635]      [-0.276,53.517]
//[-0.276,13.835]      [-0.276,53.517]      [3.817,-84.773]
//
```

Experiment: 13

Modification Of Zbus Matrix For Short Circuit Study Using Scilab Programming

Scilab code Solution 13.13 exp13

```
//Experiment−13
// windows Vista − 32−Bit
//Scilab − 5.5.2


// Aim : Modification of Zbus matrix for short
   circuit study using SCILAB Programming


// Display mode
mode(0);

clc;
clear;
s=input('Enter Size of Z bus Matrix :')// Size of Z
   bus Matrix like 3 or 4
mprintf('Enter %g X %g Matrix',s,s)
```

```
Zorg = input('')// Zorg like [1 2 3;4 5 6;7 8 9]
disp('Enter 1 for Type 1  2 for Type 2 3 for Type 3 and 4 for Type 4 Zbus modification')
check = input('Enter Your Choice :')
if check==1 then
Zb = input ('Enter value of Impidance between New Bus and Reference:')// value of Impidance between New Bus and Reference like 10 or 20 etc.
l = max(size(Zorg));
Znew= zeros(l+1,l+1);
for i = 1:l+1
  for j = 1:l+1
    if i<=(l) & j<=(l) then
      Znew(i,j)=Zorg(i,j);
    elseif i==l+1 & j==l+1 then
      Znew(i,j) = Zb;
    else
      Znew(i,j) = Znew(i,j);
    end;
  end;
end;
Znew
end;
if check==2 then
bc=input('Enter the Bus number where Impedance is Connected : ')// Bus number where Impedance is Connected in between define matrix like 1 or 2 or 3
Zb = input ('Enter value of Impidance between New Bus and existing Bus: ')//Impidance between New Bus and existing Bus like 10 or 20
l=max(size(Zorg));
```

```
row =Zorg(bc,:);
column =Zorg(:,bc);
for i = 1:l + 1
    for j = 1:l + 1
        if i<=l & j<=l
            Znew(i,j)=Zorg(i,j);
        elseif i==l+1
for p=1:l
Znew(i,p)=row(p);
end
elseif j ==l + 1
for q=1:l
Znew(q,j)=column(q);
end
end
if i ==l + 1 & j ==l + 1
Znew(i,j)=Zb+Zorg(bc,bc);
end
end
end
Znew
end

if check==4 then
    aa = input('Enter the From Bus number where Impedance is Connected :')//From Bus number where Impedance is Connected in between define matrix like 2
     bb = input('Enter the To Bus number where Impedance is Connected :')// To Bus number where Impedance is Connected in between define matrix like 3
Zb = input ('Enter value of Impidance between existing Buses : ')// Impidance between existing Buses like 10 or 20
     l = max(size(Zorg));

             Zb_row = Zorg(aa,:)-Zorg(bb,:);
```

```
            Zb_col = Zorg(:,aa)-Zorg(:,bb);
    xy=((Zb_col*Zb_row)/(Zorg(aa,aa)+Zorg(bb,bb)+Zb-2*Zorg(aa,bb)));

     Znew = Zorg-xy


end
     if check==3 then
          aa = input('Enter the Bus number where Impedance is Connected to Reference :') // Bus number where Impedance is Connected to Reference like 1 or 2 or 3
          Zb = input ('Enter value of Impidance between Existing Bus and Reference: ') // value of Impidance between Existing Bus and Reference like 10 or 20
     l = max(size(Zorg));

            Zb_row = Zorg(aa,:);
            Zb_col = Zorg(:,aa);
    xy=((Zb_col*Zb_row)/(Zorg(aa,aa)+Zb));

     Znew = Zorg-xy
    end

//
//output:-
//
//For Type-1 Zbus Modification
//
//
//Enter Size of Z bus Matrix :3
// s =
//
//     3.
//Enter 3 X 3 Matrix
//[1 2 3;4 5 6;7 8 9]
```

```
// Zorg =
//
//     1.     2.     3.
//     4.     5.     6.
//     7.     8.     9.
//
// Enter 1 for Type 1  2 for Type 2 3 for Type 3 and
    4 for Type 4 Zbus m
//       odification
//Enter Your Choice :1
// check =
//
//     1.
//Enter value of Impidance between New Bus and
   Reference:10
// Zb =
//
//     10.
// Znew =
//
//     1.     2.     3.     0.
//     4.     5.     6.     0.
//     7.     8.     9.     0.
//     0.     0.     0.     10.
//
//
//
//For Type-2 Zbus Modification
//
//
//Enter Size of Z bus Matrix :3
// s =
//
//     3.
//Enter 3 X 3 Matrix
//[1 2 3;4 5 6;7 8 9]
// Zorg =
//
```

```
//     1.     2.     3.
//     4.     5.     6.
//     7.     8.     9.
//
// Enter 1 for Type 1  2 for Type 2 3 for Type 3 and
    4 for Type 4 Zbus m
//       odification
//Enter Your Choice :2
// check  =
//
//     2.
//Enter the Bus number where Impedance is Connected
   : 2
// bc  =
//
//     2.
//Enter value of Impidance between New Bus and
   existing Bus: 15
// Zb  =
//
//     15.
// Znew  =
//
//     1.     2.     3.     2.
//     4.     5.     6.     5.
//     7.     8.     9.     8.
//     4.     5.     6.     20.
//
//For Type-3 Zbus Modification
//
//
//Enter Size of Z bus Matrix :3
// s  =
//
//     3.
//Enter 3 X 3 Matrix
//[1 2 3;4 5 6;7 8 9]
// Zorg  =
```

```
//
//     1.     2.     3.
//     4.     5.     6.
//     7.     8.     9.
//
// Enter 1 for Type 1  2 for Type 2 3 for Type 3 and
    4 for Type 4 Zbus m
//       odification
//Enter Your Choice :3
// check  =
//
//     3.
//Enter the Bus number where Impedance is Connected
   to Reference :3
// aa  =
//
//     3.
//Enter value of Impidance between Existing Bus and
   Reference: 20
// Zb  =
//
//     20.
// Znew  =
//
//     0.2758621    1.1724138    2.0689655
//     2.5517241    3.3448276    4.137931
//     4.8275862    5.5172414    6.2068966
//
//For Type-4 Zbus Modification
//
//
//Enter Size of Z bus Matrix :3
// s  =
//
//     3.
//Enter 3 X 3 Matrix
//[1 2 3;4 5 6;7 8 9]
// Zorg  =
```

```
//
//     1.      2.      3.
//     4.      5.      6.
//     7.      8.      9.
//
// Enter 1 for Type 1  2 for Type 2 3 for Type 3 and
     4 for Type 4 Zbus m
//        odification
//Enter Your Choice :4
// check  =
//
//     4.
//Enter the From Bus number where Impedance is
    Connected :2
// aa  =
//
//     2.
//Enter the To Bus number where Impedance is
    Connected :3
// bb  =
//
//     3.
//Enter value of Impidance between existing Buses :
    30
// Zb  =
//
//     30.
// Znew  =
//
//     0.90625     1.90625     2.90625
//     3.90625     4.90625     5.90625
//     6.90625     7.90625     8.90625
```

```
Enter Size of Z bus Matrix :3
 s  =

    3.
Enter 3 X 3 Matrix
[1 2 3;4 5 6;7 8 9]
 Zorg  =

    1.    2.    3.
    4.    5.    6.
    7.    8.    9.

 Enter 1 for Type 1  2 for Type 2 3 for Type 3 and 4 for Type 4 Zbus m
      odification
Enter Your Choice :1
 check  =

    1.
Enter value of Impidance between New Bus and Reference:10
 Zb  =

    10.
 Znew  =

    1.    2.    3.    0.
    4.    5.    6.    0.
    7.    8.    9.    0.
    0.    0.    0.    10.
```

```
Enter Size of Z bus Matrix :3
 s  =

    3.
Enter 3 X 3 Matrix
[1 2 3;4 5 6;7 8 9]
 Zorg  =

    1.    2.    3.
    4.    5.    6.
    7.    8.    9.

 Enter 1 for Type 1  2 for Type 2 3 for Type 3 and 4 for Type 4 Zbus m
      odification
Enter Your Choice :2
 check  =

    2.
Enter the Bus number where Impedance is Connected : 2
 bc  =

    2.
Enter value of Impidance between New Bus and existing Bus: 15
 Zb  =

    15.
 Znew  =

    1.    2.    3.    2.
    4.    5.    6.    5.
    7.    8.    9.    8.
    4.    5.    6.    20.
```

Figure 13.1: exp13

```
Enter Size of Z bus Matrix :3
 s  =

    3.
Enter 3 X 3 Matrix
[1 2 3;4 5 6;7 8 9]
 Zorg  =

    1.     2.     3.
    4.     5.     6.
    7.     8.     9.

 Enter 1 for Type 1  2 for Type 2 3 for Type 3 and 4 for Type 4 Zbus m
      odification
Enter Your Choice :3
 check  =

    3.
Enter the Bus number where Impedance is Connected to Reference :3
 aa  =

    3.
Enter value of Impidance between Existing Bus and Reference: 20
 Zb  =

    20.
 Znew  =

    0.2758621    1.1724138    2.0689655
    2.5517241    3.3448276    4.137931
    4.8275862    5.5172414    6.2068966
```

```
Enter Size of Z bus Matrix :3
 s  =

    3.
Enter 3 X 3 Matrix
[1 2 3;4 5 6;7 8 9]
 Zorg  =

    1.     2.     3.
    4.     5.     6.
    7.     8.     9.

 Enter 1 for Type 1  2 for Type 2 3 for Type 3 and 4 for Type 4 Zbus m
      odification
Enter Your Choice :4
 check  =

    4.
Enter the From Bus number where Impedance is Connected :2
 aa  =

    2.
Enter the To Bus number where Impedance is Connected :3
 bb  =

    3.
Enter value of Impidance between existing Buses : 30
 Zb  =

    30.
 Znew  =

    0.90625    1.90625    2.90625
    3.90625    4.90625    5.90625
    6.90625    7.90625    8.90625
```

Figure 13.2: exp13

Experiment: 14

Symmetrical Fault Analysis Using Scilab Programming

Scilab code Solution 14.14 exp14

```
//Experiment-14
// windows Vista - 32-Bit
//Scilab - 5.5.2


// Aim : Symmetrical Fault analysis using SCILAB
   programming

mode (0);

clc;
clear all;

SA = input('Enter Alternator Rating in kVA :');//
   Alternator Rating in kVA=10e3
XA = input('Enter Alternator % Reactance :');//
   Alternator % Reactance = 10%
ST = input ('Enter Transformer Rating in kVA :');//
   Transformer Rating in kVA =5e3
```

```
XT = input ('Enter Transformer % Reactance :');// Transformer % Reactance=5%
VL = input('Enter Transmission line Voltage in kV :');// Transmission line Voltage in kV=10
XL = input('Enter Transmissin Line Reactance in Ohm :');// Transmissin Line Reactance in Ohm=4
RL = input ('enter Transmission Line Resistance in Ohm : '); // Transmission Line Resistance in Ohm =1

Base_kVA = max(SA,ST);

PXA = (Base_kVA/SA)*XA; // Percentage Reactance of Alternator

PXT = (Base_kVA/ST)*XT; // Percentage Reactance of Transformer

PXL = (Base_kVA*XL)/(10*VL*VL); // Percentage Reactance of Transmission line

PRL = (Base_kVA*RL)/(10*VL*VL); // Percentage Resistance of Transmission line
disp('Fault at F1 means at source side and Fault at F2 means at load side')
disp('Enter 1 for Fault at F1 or 2 for Fault at F2')

check = input('Enter Your Choice for Symatrical Fault in Transmission line : ')

if check==2 then

TX = PXA+PXT+PXL; // Total Percentage Reactance

TR = PRL; // Total Percentage Resistance

PZ = sqrt(TX*TX + TR*TR); // Total Percentage Impidance
```

```
Short_Circuit_kVA = Base_kVA*(100/PZ);
Short_Circuit_Current= Short_Circuit_kVA/(sqrt(3)*VL);
mprintf('Short Circuit kVA for Fault at F2 : %f kVA \n',Short_Circuit_kVA)
mprintf('Short Circuit Current for Fault at F2 : %f A \n',Short_Circuit_Current)
end

if check==1 then
    TX = PXA+PXT; // Total Percentage Reactance

Short_Circuit_kVA1 = Base_kVA*(100/TX);
Short_Circuit_Current1= Short_Circuit_kVA1/(sqrt(3)*VL);
mprintf('Short Circuit kVA for Fault at F1 : %f kVA \n',Short_Circuit_kVA1)
mprintf('Short Circuit Current for Fault at F1 : %f A \n',Short_Circuit_Current1)
end


//output:-
//
//Symatrical Fault F1 at source side:
//
//Enter Alternator Rating in kVA :10e3
//Enter Alternator % Reactance :10
//Enter Transformer Rating in kVA :5e3
//Enter Transformer % Reactance :5
//Enter Transmission line Voltage in kV :10
//Enter Transmissin Line Reactance in Ohm :4
//enter Transmission Line Resistance in Ohm : 1
//
// Fault at F1 means at source side and Fault at F2 means at load side
//
```

```
// Enter 1 for Fault at F1 or 2 for Fault at F2
//Enter Your Choice for Symatrical Fault in
    Transmission line : 1
// check =
//
//    1.
//Short Circuit kVA for Fault at F1 : 50000.000000
    kVA
//Short Circuit Current for Fault at F1 :
    2886.751346 A
//
//Symatrical Fault F2 at laod side:
//
//Enter Alternator Rating in kVA :10e3
//Enter Alternator % Reactance :10
//Enter Transformer Rating in kVA :5e3
//Enter Transformer % Reactance :5
//Enter Transmission line Voltage in kV :10
//Enter Transmissin Line Reactance in Ohm :4
//enter Transmission Line Resistance in Ohm : 1
//
// Fault at F1 means at source side and Fault at F2
    means at load side
//
// Enter 1 for Fault at F1 or 2 for Fault at F2
//Enter Your Choice for Symatrical Fault in
    Transmission line : 2
// check =
//
//    2.
//Short Circuit kVA for Fault at F2 : 16439.898731
    kVA
//Short Circuit Current for Fault at F2 : 949.157996
    A
```

```
Enter Alternator Rating in kVA :10e3
Enter Alternator % Reactance :10
Enter Transformer Rating in kVA :5e3
Enter Transformer % Reactance :5
Enter Transmission line Voltage in kV :10
Enter Transmissin Line Reactance in Ohm :4
enter Transmission Line Resistance in Ohm : 1

 Fault at F1 means at source side and Fault at F2 means at load side

 Enter 1 for Fault at F1 or 2 for Fault at F2
Enter Your Choice for Symatrical Fault in Transmission line : 1
 check  =

    1.
Short Circuit kVA for Fault at F1 : 50000.000000 kVA
Short Circuit Current for Fault at F1 : 2886.751346 A
```

Figure 14.1: exp14

```
Enter Alternator Rating in kVA :10e3
Enter Alternator % Reactance :10
Enter Transformer Rating in kVA :5e3
Enter Transformer % Reactance :5
Enter Transmission line Voltage in kV :10
Enter Transmissin Line Reactance in Ohm :4
enter Transmission Line Resistance in Ohm : 1

 Fault at F1 means at source side and Fault at F2 means at load side

 Enter 1 for Fault at F1 or 2 for Fault at F2
Enter Your Choice for Symatrical Fault in Transmission line : 2
 check  =

    2.
Short Circuit kVA for Fault at F2 : 16439.898731 kVA
Short Circuit Current for Fault at F2 : 949.157996 A
```

Figure 14.2: exp14