

Scilab Manual for
Electrical Power System II
by Prof Mehulsinh Ghanshyamsinh Jadeja
Electrical Engineering
Vishwakarma Government Engineering
College, Chandkheda¹

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

June 7, 2026

¹Funded 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 Scilab Solutions	4
1 A Scilab Program Obtain Voltage Regulation And Efficiency Of A Short Transmission Line.	6
2 A Scilab Program To Evaluate Performance Of Short Transmission Line Using Graphical Method	9
3 Finding Voltage Regulation And Efficiency Of A Medium Transmission Line Using Nominal T Method Through Scilab	12
4 Finding Voltage Regulation And Efficiency Of A Medium Transmission Line Using Nominal Pi Method Through Scilab	15
5 Finding Voltage Regulation And Efficiency Of A Medium Transmission Line Using End Condenser Method Through Scilab	20
6 A Scilab Program To Evaluate Performance Of Medium Transmission Line Using Graphical Method	25
7 Finding voltage regulation and efficiency of a long transmission line using Rigorous solution method through SCILAB	28
8 A Scilab Program To Evaluate Performance Of Long Transmission Line Using Graphical Method	32
9 A Scilab Program To Find Permissible Frequency And Length For Limiting Ferranti Effect In Long Transmission Lines.	35

10 A Scilab Program For Computing The Abcd Parameters Of Short, Medium And Long Transmission Lines	38
11 A Scilab Program For Conversion Of Unsymmetrical Ac Phasors Into Symmetrical Components	46
12 A Scilab Program For Conversion Of Symmetrical Components Into Ac Phasors	51
13 Modification Of Zbus Matrix For Short Circuit Study Using Scilab Programming	54
14 Symmetrical Fault Analysis Using Scilab Programming	64

List of Experiments

Solution 1.1	exp1	6
Solution 2.2	exp2	9
Solution 3.3	exp3	12
Solution 4.4	exp4	15
Solution 5.5	exp5	20
Solution 6.6	exp6	25
Solution 7.7	exp7	28
Solution 8.8	exp8	32
Solution 9.9	exp9	35
Solution 10.10	exp10	38
Solution 11.11	exp11	46
Solution 12.12	exp12	51
Solution 13.13	exp13	54
Solution 14.14	exp14	64

List of Figures

1.1	exp1	8
2.1	exp2	11
3.1	exp3	14
4.1	exp4	19
5.1	exp5	24
6.1	exp6	27
7.1	exp7	31
8.1	exp8	34
9.1	exp9	37
10.1	exp10	44
10.2	exp10	45
11.1	exp11	50
11.2	exp11	50
12.1	exp12	53
13.1	exp13	62
13.2	exp13	63
14.1	exp14	68
14.2	exp14	69

Experiment: 1

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

Scilab code Solution 1.1 expl

```
1 //Experiment-1
2 // windows Vista - 32-Bit
3 //Scilab - 5.5.2
4
5
6 //Aim : A SCILAB program obtain voltage regulation
      and efficiency of a short transmission line.
7
8 clc;
9 clear all;
10 VRLL=220; VR = VRLL/sqrt(3);
11 Z = (0.15+%i*2*%pi*60*1.3263e-3)*40; //change
      frequency or length
12
13 disp('(a)')
```

```

14 SR=304.8+%i*228.6; // lagging power factor
15 IR = conj(SR)/(3*conj(VR)) ;
16 IS = IR;
17 VS = VR + Z*IR;
18 VSLL = sqrt(3)*abs(VS)
19 SS = 3*VS*conj(IS)
20 REG = (VSLL - VRLL)/VRLL*100
21 Eff = real(SR)/real(SS)*100
22 mprintf('Sending end line voltage = %f kV \n',VSLL)
23 mprintf('Sending Apprent power = %f MVA \n',SS)
24 mprintf('Regulation = %f %% \n',REG)
25 mprintf('Efficiency = %f %% \n',Eff)
26
27 disp('(b)')
28 SR=304.8-%i*228.6; // leading power factor
29 IR = conj(SR)/(3*conj(VR)); IS = IR;
30 VS = VR + Z*IR;
31 VSLL = sqrt(3)*abs(VS)
32 SS = 3*VS*conj(IS)
33 REG = (VSLL - VRLL)/VRLL*100
34 Eff = real(SR)/real(SS)*100
35 mprintf('Sending end line voltage = %f kV \n',VSLL)
36 mprintf('Sending Apprent power = %f MVA \n',SS)
37 mprintf('Regulation = %f %% \n',REG)
38 mprintf('Efficiency = %f %% \n',Eff)
39
40 //output:-
41 //(a)
42 //Sending end line voltage = 250.018650 kV
43 //Sending Apprent power = 322.795165 MVA
44 //Regulation = 13.644841 %
45 //Efficiency = 94.425206 %
46 //
47 //(b)
48 //Sending end line voltage = 210.288383 kV
49 //Sending Apprent power = 322.795165 MVA
50 //Regulation = -4.414372 %
51 //Efficiency = 94.425206 %

```

(a)
Sending end line voltage = 250.018650 kV
Sending Apparent power = 322.795165 MVA
Regulation = 13.644841 %
Efficiency = 94.425206 %

(b)
Sending end line voltage = 210.288383 kV
Sending Apparent 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

```
1 //Experiment-2
2 // windows Vista - 32-Bit
3 //Scilab - 5.5.2
4
5
6 //Aim : A SCILAB program to evaluate performance of
   short transmission line using graphical method
7
8
9 clear all;
10 clc;
11 // This program calculates the percentage regulation
   of a short transmission line
12 // It is assumed that the receiving end voltage Vr
   and magnitude of current Ir are constant
```

```

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

21         Irang = acos(pf);
22         Ir = Irmag * exp(-%i*Irang); // For
    lagging power factor
23         Vs = (Vr/sqrt(3))+ Ir*Z; // sqrt(3)
    is used here as we need to do a per
    phase analysis
24         VrNL = abs(Vs);
25 VrFL = abs(Vr/sqrt(3));
26 per_vol_regu= ((VrNL-VrFL)/VrFL)*100;
27 plot(pf,per_vol_regu,'-');
28 xlabel('power factor')
29 ylabel('voltage regulation')
30 title('Graph of power factor v/s voltage regulation'
    )
31
32 // For leading power factor range, angle should be
    positive So,
33
34 Ir = Irmag * exp(%i*Irang);
35
36 Vs = (Vr/sqrt(3))+ Ir*Z;
37
38 VrNL = abs(Vs); // absolute value of Vs
39
40 VrFL = abs(Vr/sqrt(3)); //abs value of Vr per phase
41
42 per_vol_regu= ((VrNL-VrFL)/VrFL)*100;

```

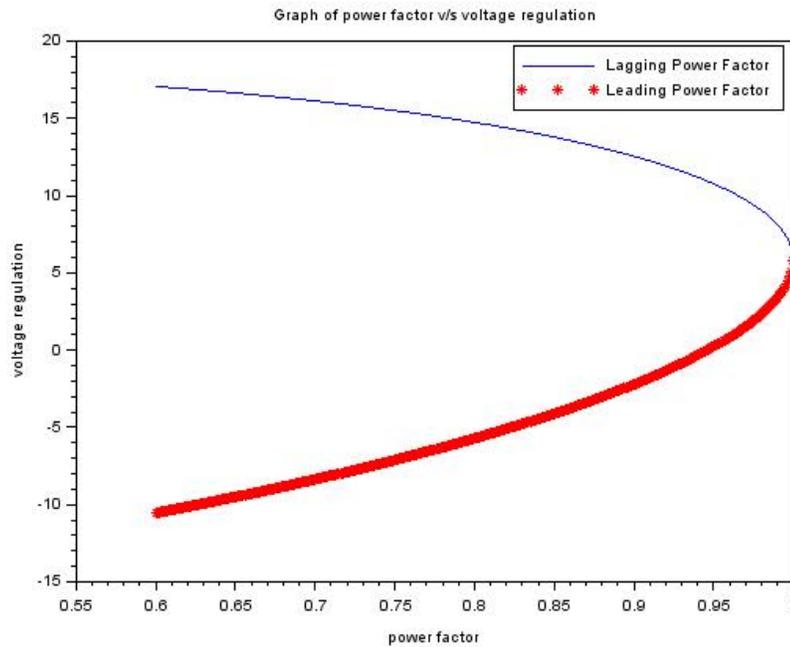


Figure 2.1: exp2

```

43 plot(pf,per_vol_regu,'r*');
44 legend('Lagging Power Factor','Leading Power Factor'
45 )
46 //gtext('Vreg vs lagging pf')
47 //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

```
1 //Experiment-3
2 // windows Vista - 32-Bit
3 //Scilab - 5.5.2
4
5
6 // Aim ; Finding voltage regulation and efficiency
   of a medium transmission line using nominal T
   method through SCILAB
7
8 clc;
9 clear all;
10 R=10;
11 XL=20;
```

```

12 Y= 4e-4;
13 VRLL=66e3;
14 PR=10000e3;
15 pf=0.8; // change power factor of load
16 VR=66e3/sqrt(3);
17 Z=R+%i*XL;
18 IR=PR/(sqrt(3)*VRLL*pf);
19 IR_COMPLEX=IR*(cos(acos(0.8))-%i*sin(acos(0.8)));
20 VM=VR+IR_COMPLEX*(Z/2);
21 IC=%i*Y*VM;
22 IS=IR_COMPLEX+IC;
23 VS=VM+IS*(Z/2);
24 [VS_abs,Phase_VS]= polar(VS);
25 [IS_abs,Phase_IS]= polar(IS);
26 VS_abs;
27 IS_abs
28 VSLL=VS*sqrt(3);
29 VSLL_abs=abs(VSLL)/1000
30 PHASE_DIFF=Phase_VS-Phase_IS;
31 SendingEnd_PF=cos(PHASE_DIFF)
32 PS=3*abs(VS)*abs(IS)*cos(PHASE_DIFF);
33 EFF=(abs(PR)/abs(PS))*100
34 REG = 100*((abs(VSLL) - abs(VRLL))/ abs(VRLL))
35 mprintf('Sendingend Current = %f A \n',IS_abs)
36 mprintf('Sendingend Line Voltage = %f kV \n',
    VSLL_abs)
37 mprintf('Sendingend Power factor = %f \n',
    SendingEnd_PF)
38 mprintf('Efficiency = %f %% \n', EFF)
39 mprintf('Regulation = %f %% \n', REG)
40
41
42 //Output:-
43 //
44 //Sendingend Current = 100.533214 A
45 //Sendingend Line Voltage = 69.543925 kV
46 //Sendingend Power factor = 0.853122
47 //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

```
48 //Regulation = 5.369583 %
49 //
```

Experiment: 4

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

Scilab code Solution 4.4 exp4

```
1 //Experiment-4
2 // windows Vista - 32-Bit
3 //Scilab - 5.5.2
4
5
6 // Aim : Finding voltage regulation and efficiency
   of a medium transmission line using nominal PI
   method through SCILAB
7
8 mode(0);
9
10 clc
11 clear all
```

```

12 disp('Nominal PI Method for medium transmission line
      ')
13 R = input('Enter the Total line Resistance in Ohms :
      '); // Total Line Resistance in Ohms like 10
      Ohms
14 XL = input('Enter the Total line Inductive Reactance
      in Ohms : '); // Total line Inductive Reactance
      in Ohms like 50 Ohms
15 Y = input('Enter the Total line Suseptance in Seimen
      : '); // Total line Suseptance in Seimen like 10
      e-4 Seimen
16 f = input('Enter the Frequency in Hz: '); //
      Frequency in Hz like 50 Hz
17 PR= input('Enter the Recieving End Power in Watt : '
      ); // Recieving End Power in Volt like 20e6 W
18 VRLL= input('Enter the Recieving End Line Voltage in
      Volt :'); // Recieving End Line Voltage in Volt
      like 66e3 V
19 pf= input('Enter Recievinf End Lagging Power Factor
      : '); // Recievinf End Lagging Power Factor like
      0.9
20 Z=R+%i*XL;
21 VR = VRLL/sqrt(3);
22
23 IR=PR/(sqrt(3)*VRLL*pf);
24 IR_COMPLEX=IR*(cos(acos(pf))-%i*sin(acos(pf)));
25
26 IC1=%i*Y/2*VR;
27
28 IL = IR_COMPLEX + IC1;
29
30 VS = VR + IL*Z;
31
32 IC2 = %i*Y/2*VS;
33
34 IS=IL + IC2
35
36 [VS_abs ,Phase_VS]= polar(VS);

```

```

37 [IS_abs,Phase_IS]= polar(IS);
38 VS_abs;
39 IS_abs;
40 VLL=VS*sqrt(3);
41 VLL_abs=abs(VLL)/1000
42 PHASE_DIFF=Phase_VS-Phase_IS;
43 SendingEnd_PF=cos(PHASE_DIFF);
44
45 PS=3*abs(VS)*abs(IS)*cos(PHASE_DIFF)
46 EFF=(abs(PR)/abs(PS))*100
47 REG = 100*((abs(VLL) - abs(VRLL))/ abs(VRLL))
48
49 fprintf('Sendingend Current = %f A \n',IS_abs)
50 fprintf('Sendingend Line Voltage = %f kV \n',
    VSLL_abs)
51 fprintf('Sendingend Power factor = %f \n',
    SendingEnd_PF)
52 fprintf('Efficiency = %f %% \n', EFF)
53 fprintf('Regulation = %f %% \n', REG)
54
55
56 //output:-
57 //
58 //Nominal PI Method for medium transmission line
59 //Enter the Total line Resistance in Ohms : 10
60 //Enter the Total line Inductive Reactance in Ohms :
    50
61 //Enter the Total line Suseptance in Seimen : 10e-4
62 //Enter the Frequency in Hz: 50
63 //Enter the Recieving End Power in Watt : 20e6
64 //Enter the Recieving End Line Voltage in Volt :66e3
65 //Enter Recievinf End Lagging Power Factor : 0.9
66 // IS =
67 //
68 //      170.90917 - 44.112456 i
69 // VSLL_abs =
70 //
71 //      76.021348

```

```
72 // PS =
73 //
74 //      21047697. + 1.094D-09i
75 // EFF =
76 //
77 //      95.022274
78 // REG =
79 //
80 //      15.183861
81 //Sendingend Current = 176.510208 A
82 //Sendingend Line Voltage = 76.021348 kV
83 //Sendingend Power factor = 0.905604
84 //Efficiency = 95.022274 %
85 //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
VLL_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

```
1 //Experiment-5
2 // windows Vista - 32-Bit
3 //Scilab - 5.5.2
4
5
6 // Aim : Finding voltage regulation and efficiency
   of a medium transmission line using End
   Condenser method through SCILAB
7
8 mode(0);
9
10 clc
11 clear all
```

```

12 disp('End Condensor Method for medium transmission
      line')
13 R =25 // ('Enter the line Resistance in Ohms : ')
14 X =80 // ('Enter the line Inductive Reactance in
      Ohms : ')
15 Y =14e-4 // ('Enter the line Suseptance in Seimen :
      ')
16 f =50 // ('Enter the frequency : ')
17 PR=15e6 // ('Enter the Recieving End Power in Watt :
      ')
18 VRLine=66e3 // ('Enter the Recieving End Line
      Voltage in Volt :')
19 Pf=0.8 // ('Enter Recievinf End Lagging Power Factor
      : ')
20 // X = 2*%pi*f*L
21 Z = R + X*%i;
22 //Y = %i* 2 *%pi*f*C
23
24 IR = PR/(VRLine*Pf);
25
26 Pfang = acos(Pf);
27
28 IR_Complex= IR*(Pf - %i*sin(Pfang));
29
30 IC_Complex = %i*Y*VRLine;
31
32 IS = IR_Complex + IC_Complex
33
34 Voltage_drop = IS*Z;
35
36 VS = VRLine + IS*Z
37
38 [VS_abs,Phase_VS]= polar(VS);
39
40 [IS_abs,Phase_IS]= polar(IS);
41
42 VS_abs;
43 IS_abs;

```

```

44
45 PHASE_DIFF=Phase_VS-Phase_IS;
46 SendingEnd_PF=cos(PHASE_DIFF);
47 Sending_End_PF = abs(SendingEnd_PF)
48 PS=abs(VS)*abs(IS)*cos(PHASE_DIFF);
49 EFF=(abs(PR)/abs(PS))*100
50 REG = 100*((abs(VS) - abs(VRLine))/ abs(VRLine))
51
52 //
53 //output:-
54 //
55 //End Condensor Method for medium transmission line
56 // R =
57 //
58 // 25.
59 // X =
60 //
61 // 80.
62 // Y =
63 //
64 // 0.0014
65 // f =
66 //
67 // 50.
68 // PR =
69 //
70 // 15000000.
71 // VRLine =
72 //
73 // 66000.
74 // Pf =
75 //
76 // 0.8
77 // IS =
78 //
79 // 227.27273 - 78.054545 i
80 // VS =
81 //

```

```
82 //      77926.182 + 16230.455 i
83 // Sending_End_PF  =
84 //
85 //      0.8596751
86 // EFF  =
87 //
88 //      91.220706
89 // REG  =
90 //
91 //      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

Experiment: 6

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

Scilab code Solution 6.6 exp6

```
1 //Experiment-6
2 // windows Vista - 32-Bit
3 //Scilab - 5.5.2
4
5
6 // Aim : A SCILAB program to evaluate performance of
   medium transmission line using graphical method
7
8 clear all;
9 clc;
10 // Program to calculate voltage regulation of
   nominal PI method of medium length transmission
   line
11 // Line parameters are taken from Stevenson as
   before. One may use any other data using the
```

```

    %commented input functions
12 Vr = 220e3;    // input('Enter the value of Vr = ') ;
13 R = 33.5;     // input('Enter the value of R = ') ;
14 X = 138;     // input('Enter the value of X = ') ;
15 pf = 0.9;    // input('Enter the value of pf = ') ;
16 Pr = 50e6;   // input('Enter the value of pr = ') ;
17 Y = %i*0.3e-3; // input('Enter the value of Y = ') ;
18 Z = R + X * %i;
19 Irmag = Pr/(sqrt(3)*Vr*pf);
20 pf = 0.5:0.001:1;
21 Irang = acos (pf); // Angle in radian
22 Ir =Irmag * exp (-%i*Irang);
23 // 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

24
25 Ir1 =Irmag * exp (%i*Irang);    // For leading power
    factor
26 // Calculations for lagging power factor for medium
    line length
27 Vs = Vr/ sqrt (3) + (Ir+(Y/2)*Vr/ sqrt (3))*Z; //
    all terms are of voltage and are phase values
28 VrNLm =abs( Vs / (1+((Y*Z)/2)));
29 VrFLm= abs (Vr / sqrt (3));
30 per_vol_regulag = ( (VrNLm -VrFLm) / VrFLm) * 100;
31 // Calculations for leading power factor for medium
    line length
32 Vs1m = Vr/ sqrt (3) + (Ir1+(Y/2)*Vr/ sqrt (3))*Z;
33 // all terms are of voltage and are in phase values
34 VrNL1m =abs( Vs1m / (1+((Y*Z)/2)));
35 per_vol_regulead = ( (VrNL1m -VrFLm) / VrFLm) * 100;
36 // plot the two waveforms on the same graph
37 plot(pf,per_vol_regulag , 'r' ,pf,per_vol_regulead , '*')
    ;
38

```

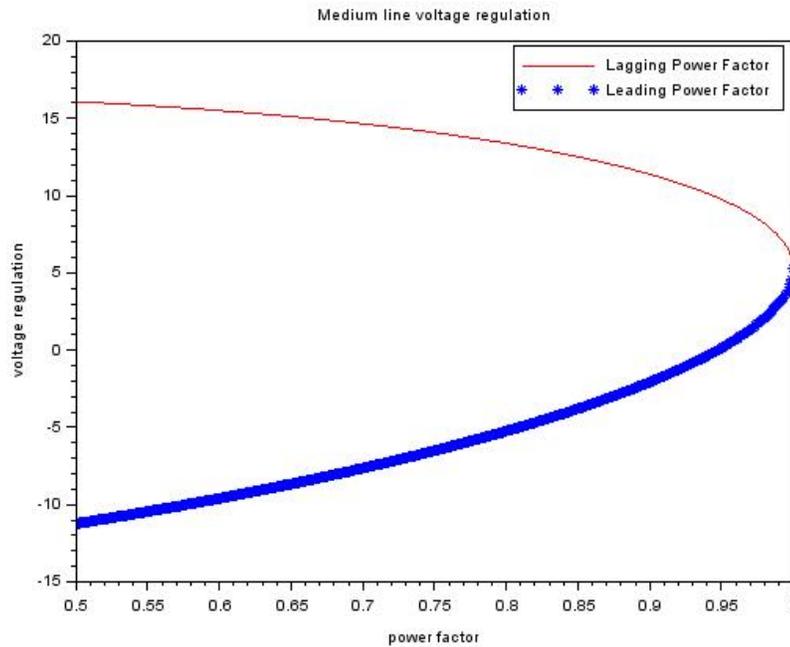


Figure 6.1: exp6

```

39 xlabel('power factor')
40 ylabel('voltage regulation')
41 title('Medium line voltage regulation')
42 legend('Lagging Power Factor','Leading Power Factor'
43 )
44 // gtext('Vreg vs lagging pf')
45 // 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

```
1 //Experiment-7
2 // windows Vista - 32-Bit
3 //Scilab - 5.5.2
4
5
6 // Aim : Finding voltage regulation and efficiency
      of a long transmission line using Rigorous
      solution method through SCILAB
7
8 mode(0);
9
10 clc
11 clear all
12 disp('Rigorous Method for Long transmission line')
13 R = input('Enter the Total line Resistance in Ohms :')
```

```

    '); // Total line Resistance in Ohms like 32
    Ohms
14 XL = input('Enter the Total line Inductive Reactance
    in Ohms : '); // Total line Inductive Reactance
    in Ohms like 50 Ohms
15 y = input('Enter the Total line Suseptance in Seimen
    : '); // Total line Suseptance in Seimen like
    0.0003
16 f = input('Enter the frequency in Hz : '); //
    frequency in Hz like 50 Hz
17 PR= input('Enter the Recieving End Power in Watt : '
    );// Recieving End Power in Watt like 20e6
18 VRLL= input('Enter the Recieving End Line Voltage in
    Volt :'); // Recieving End Line Voltage in Volt
    like 110e3
19 pf= input('Enter Recievinf End Lagging Power Factor
    : '); // Recievinf End Lagging Power Factor like
    0.8
20 Z=R+%i*XL;
21 Y= %i*y;
22 VR = VRLL/sqrt(3);
23 IR=PR/(sqrt(3)*VRLL*pf);
24
25 A = 1+ (Z*Y)/2 + (Z*Z*Y*Y)/24; //A= cosh(sqrt(Y*Z
    ))
26
27 B = sqrt(Y*Z) + ((Y*Z)^(3/2))/6; //B = sinh(sqrt(
    Y*Z))
28
29 VS = VR*A + IR*sqrt(Z/Y)*B; //VS = VR*cosh(sqrt(Y
    *Z)) + IR*sqrt(Z/Y)*sinh(sqrt(Y*Z))
30
31 IS = VR*sqrt(Y/Z)*B + IR*A; //IS = VR*sqrt(Y/Z)*
    sinh(sqrt(Y*Z)) + IR*cosh(sqrt(Y*Z))
32
33 [VS_abs,Phase_VS]= polar(VS);
34 [IS_abs,Phase_IS]= polar(IS);
35 VS_abs;

```

```

36 IS_abs;
37 VLL=VS*sqrt(3);
38 VLL_abs=abs(VLL)/1000;
39 Phase_VS;
40 Phase_IS;
41
42 PHASE_DIFF=Phase_VS-Phase_IS;
43
44 SendingEnd_PF=cos(PHASE_DIFF);
45
46 PS=3*VS_abs*IS_abs*cos(PHASE_DIFF);
47 EFF=(abs(PR)/abs(PS))*100;
48 REG = 100*((abs(VLL) - abs(VLLR))/ abs(VLLR));
49
50 mprintf('\n\nSendingend Current = %f A \n',IS_abs)
51 mprintf('Sendingend Line Voltage = %f kV \n',
        VLL_abs)
52 mprintf('Sendingend Power factor = %f \n',
        SendingEnd_PF)
53 mprintf('Efficiency = %f %% \n', EFF)
54 mprintf('Regulation = %f %% \n', REG)
55
56
57 //output:-
58 //
59 // Rigorous Method for Long transmission line
60 //Enter the Total line Resistance in Ohms : 32
61 //Enter the Total line Inductive Reactance in Ohms :
        50
62 //Enter the Total line Suseptance in Seimen : 0.0003
63 //Enter the frequency in Hz : 50
64 //Enter the Recieving End Power in Watt : 20e6
65 //Enter the Recieving End Line Voltage in Volt :110
        e3
66 //Enter Recievinf End Lagging Power Factor : 0.8
67 //
68 //
69 //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

```

70 //Sendingend Line Voltage = 117.015932 kV
71 //Sendingend Power factor = 0.998847
72 //Efficiency = 75.028310 %
73 //Regulation = 6.378120 %

```

Experiment: 8

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

Scilab code Solution 8.8 exp8

```
1 //Experiment-8
2 // windows Vista - 32-Bit
3 //Scilab - 5.5.2
4
5
6 // Aim : A SCILAB program to evaluate performance of
      long transmission line using graphical method
7
8 clc
9 clear all
10 disp('Rigorous Method for Long transmission line
      using graphical method')
11 R =32 // ('Enter the Total line Resistance in Ohms :
      ')
12 XL =50 // ('Enter the Total line Inductive Reactance
```

```

        in Ohms : ')
13 y =0.003 // ('Enter the Total line Suseptance in
    Seimen : ')
14 f =50 // ('Enter the frequency : ')
15 PR= 20e6 //('Enter the Recieving End Power in Watt :
    ')
16 VRLL=110e3 // ('Enter the Recieving End Line Voltage
    in Volt :')
17 pf=0.8,
18 Z=R+%i*XL;
19 Y= %i*y;
20 VR = VRLL/sqrt(3);
21 IR=PR/(sqrt(3)*VRLL*pf);
22 A = 1+ (Z*Y)/2 + (Z*Z*Y*Y)/24; //A= cosh(sqrt(Y*Z
    ))
23
24 B = sqrt(Y*Z) + ((Y*Z)^(3/2))/6; //B = sinh(sqrt(
    Y*Z))
25
26 pf = 1:-0.01:0.5
27
28
29 Irang = acos (pf); // Angle in radian
30
31 Ir =IR * exp (-%i*Irang);
32 Ir1 =IR * exp (%i*Irang);
33
34
35 VS = VR*A + Ir*sqrt(Z/Y)*B; //VS = VR*cosh(sqrt(Y
    *Z)) + IR*sqrt(Z/Y)*sinh(sqrt(Y*Z))
36
37 VS1 = VR*A + Ir1*sqrt(Z/Y)*B ; //VS = VR*cosh(sqrt
    (Y*Z)) + IR*sqrt(Z/Y)*sinh(sqrt(Y*Z))
38
39
40 [VS_abs,Phase_VS]= polar(VS);
41 [VS1_abs,Phase_VS1]= polar(VS1);
42

```

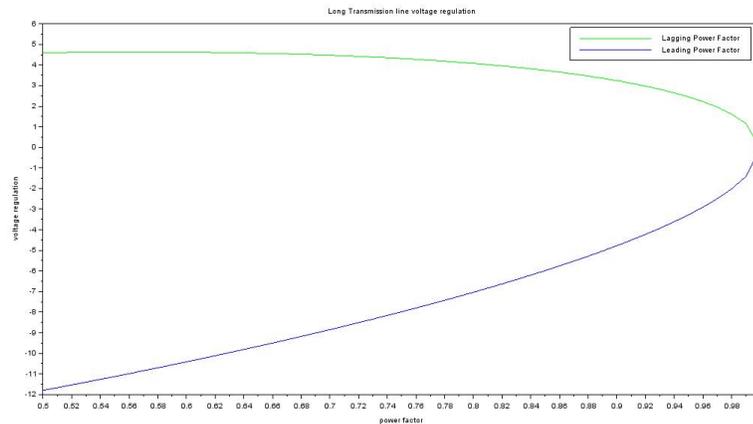


Figure 8.1: exp8

```

43 VSLL=VS;
44 VSLL1=VS1;
45 REGllag = 100*((abs (VSLL) - abs (VR)))/ abs (VR)
46 REGllead = 100*((abs (VSLL1) - abs (VR)))/ abs (VR)
47
48
49 plot(pf,REGllag, 'g-');
50 plot(pf,REGllead, 'b');
51
52 xlabel('power factor')
53 ylabel('voltage regulation')
54 title('Long Transmission line voltage regulation')
55 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

```
1 //Experiment-9
2 // windows Vista - 32-Bit
3 //Scilab - 5.5.2
4
5
6 // Aim : A SCILAB Program to find permissible
    frequency and length for limiting Ferranti effect
    in long transmission lines.
7
8
9 clc;
10 clear all; // Clear All Variables & Command Window
11 Standard_f=50;
```

```

12 l=400;
13 VSLL=220e3; //Defining Given DATA
14 r=0.125; //Defining Given DATA
15 x=0.4; //Defining Given DATA
16 y=%i*2.8e-6; //Defining Given DATA
17 R=r*l;
18 X=x*l;
19 Y=y*l;
20 z=r+%i*x;
21 Z=R+%i*X;
22 A=1+Y*Z/2; // Defining ABCD Parameters
23 B=Z*(1+Y*Z/6); // Defining ABCD
Parameters
24 C=Y*(1+Y*Z/6); // Defining ABCD Parameters
25 D=A; // Defining ABCD Parameters
26
27 No_load_VRLL=VSLL/A;
28 NoLoad_VRLL=abs(No_load_VRLL)/1000 //Display
No Load Voltage at Receiving
29 NoLoad_VR=NoLoad_VRLL/sqrt(3); // Calculate
Voltage per Phase
30 NoLoad_IS=abs(C*NoLoad_VR) // Calculate
Sending end Current
31
32 VRLL_MAX=235e3; // Define limit
, the receiving end Volt
33 new_A=VSLL/VRLL_MAX; // Evaluate
Parameter A for VRLL_MAX
34 l=sqrt(real((2*new_A-2)/y/z)); // Equation for
maximum Length
35 Permissible_Lenth=abs(1) // Find the
Magnitude out of Complex
36
37 VRLL_MAX2=250e3; // Similar to
above
38 new_A2=VSLL/VRLL_MAX2;
39 Max_Permissible_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

```
40
41 mprintf('No Load Voltage at Receiving end = %f kV \n
    \n', NoLoad_VRLL)
42 mprintf('No load Sending end Current = %f A \n\n',
    NoLoad_IS)
43 mprintf('Permissible length = %f Km \n\n',
    Permissible_Lenth)
44 mprintf('Max. Permissible Frequency = %f Hz \n\n',
    Max_Permissible_f)
45
46
47 //output:-
48 //
49 // No Load Voltage at Receiving end = 241.537811 kV
50 //
51 //No load Sending end Current = 0.151528 A
52 //
53 //Permissible length = 322.243754 Km
54 //
55 //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

```
1 //Experiment-10
2 // windows Vista - 32-Bit
3 //Scilab - 5.5.2
4
5
6 // Aim : A SCILAB program for computing the ABCD
      parameters of short, medium and long transmission
      lines
7
8 clc
9 clear all
10 disp('Enter your choice:')
11 disp('1 : for short transmission line')
12 disp('2 : for medium transmission line')
13 disp('3 : for long transmission line')
```

```

14 check = input('Enter your choice : ')
15 R = input('Enter the line Resistance in Ohms : ')
16 L = input('Enter the line Inductance in Henry : ')
17 C = input('Enter the line Capacitance in Farad : ')
18 f = input('Enter the frequency : ')
19 X = 2*%pi*f*L
20 Z = R + %i*X;
21 Y = %i*2*%pi*f*C
22 if(check==1)
23 Adata = 1; // ABCD in cartesian form
24 Bdata = Z; // ABCD in cartesian form
25 Cdata = 0; // ABCD in cartesian form
26 Ddata = 1; // ABCD in cartesian form
27 Result = [Adata, Bdata, Cdata, Ddata]
28 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)
29 end
30
31 if(check==2)
32 disp(' Enter 1 for Nominal T-method')
33 disp('Enter 2 for Nominal PI-method')
34 s = input('Enter your choice : ')
35 if(s==1)
36 Adata = 1 + (Y*Z)/2;
37 Bdata = Z*(1 + (Y*Z)/4);
38 Cdata = Y;
39 Ddata = (1 + (Y*Z)/2);
40
41 Result = [Adata, Bdata, Cdata, Ddata]
42 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))
43 else

```

```

44 Adata = 1 + (Y*Z)/2;
45 Bdata = Z
46 Cdata = Y * (1 + (Y*Z)/4);
47 Ddata = (1 + (Y*Z)/2);
48 Result = [Adata, Bdata, Cdata, Ddata]
49 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))
50 end end
51
52 if(check==3)
53 Adata = cosh(sqrt(Y*Z));
54 Bdata = (sqrt(Z/Y) * sinh(sqrt(Y*Z)));
55 Cdata = (1/sqrt(Z/Y) * sinh(sqrt(Y*Z)));
56 Ddata = cosh(sqrt(Y*Z));
57 Result = [Adata,Bdata,Cdata,Ddata];
58 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))
59 end
60
61 //
62 //output:-
63 //
64 //For Short Transmission line
65 //
66 //Enter your choice:
67 //
68 // 1 : for short transmission line
69 //
70 // 2 : for medium transmission line
71 //

```

```

72 // 3 : for long transmission line
73 //Enter your choice : 1
74 //Enter the line Resistance in Ohms : 20
75 //Enter the line Inductance in Henry : 50
76 //Enter the line Capacitance in Farad : 0.0003
77 //Enter the frequency : 50
78 //ABCD constant for Short Transmission line
79 // A constant = 1.000000
80 // B constant= 20.000000 + i 15707.963268 ohm
81 // C constant=0.000000 mho
82 // D constant = 1.000000
83 //
84 //
85 //For Nominal T method of medium Transmission line
86 //
87 //Enter your choice:
88 //
89 // 1 : for short transmission line
90 //
91 // 2 : for medium transmission line
92 //
93 // 3 : for long transmission line
94 //Enter your choice : 2
95 //Enter the line Resistance in Ohms : 20
96 //Enter the line Inductance in Henry : 50
97 //Enter the line Capacitance in Farad : 0.0003
98 //Enter the frequency : 50
99 //
100 // Enter 1 for Nominal T-method
101 //
102 // Enter 2 for Nominal PI-method
103 //Enter your choice : 1
104 //ABCD constant for Medium Transmission line Nominal
    T-method
105 // A constant = -739.220330 + i 0.942478
106 // B constant= -14784.406602 + i -5797959.489510 ohm
107 // C constant=0.000000 + i 0.094248 mho
108 // D constant = -739.220330 + i0.942478

```

```

109 //
110 //For Nominal PI method of medium Transmission line
111 //
112 //Enter your choice:
113 //
114 // 1 : for short transmission line
115 //
116 // 2 : for medium transmission line
117 //
118 // 3 : for long transmission line
119 //Enter your choice : 2
120 //Enter the line Resistance in Ohms : 20
121 //Enter the line Inductance in Henry : 50
122 //Enter the line Capacitance in Farad : 0.0003
123 //Enter the frequency : 50
124 //
125 // Enter 1 for Nominal T-method
126 //
127 // Enter 2 for Nominal PI-method
128 //Enter your choice : 2
129 //ABCD constant for Medium Transmission line Nominal
    PI-method
130 // A constant =  $-739.220330 + i 0.942478$ 
131 // B constant=  $20.000000 + i15707.963268$  ohm
132 // C constant= $-0.044413 + i -34.787813$  mho
133 // D constant =  $-739.220330 + i0.942478$ 
134 //
135 //
136 //For long Transmission line
137 //
138 //Enter your choice:
139 //
140 // 1 : for short transmission line
141 //
142 // 2 : for medium transmission line
143 //
144 // 3 : for long transmission line
145 //Enter your choice : 3

```

```
146 //Enter the line Resistance in Ohms : 20
147 //Enter the line Inductance in Henry : 50
148 //Enter the line Capacitance in Farad : 0.0003
149 //Enter the frequency : 50
150 //ABCD constant for Long Transmission line
151 // A constant = 0.712960 + i 0.017183
152 // B constant= 7.310527 + i286.435826 ohm
153 // C constant=0.000042 + i 0.001719 mho
154 // 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 + i0.942478

```

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

```

Experiment: 11

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

Scilab code Solution 11.11 exp11

```
1 //Experiment-11
2 // windows Vista - 32-Bit
3 //Scilab - 5.5.2
4
5
6 // Aim : A SCILAB program for conversion of
   // unsymmetrical AC phasors into symmetrical
   // components
7
8
9
10 clc
11
12 disp('Enter your choice:');
13
```

```

14 disp('1 for phasor to symmetrical; 2 for
    symmetrical to phasor');
15 choice=input('Enter your transformation type=');
16
17 alpha = -0.5+%i*0.866;
18 if(choice==1)
19
20 Va=input('Enter the value of complex phase voltage
    VA=') //complex phase voltage VA=200 - %i*100
21 Vb=input('Enter the value of complex phase voltage
    VB=') //complex phase voltage VB=-36.59 - %i
    *209.804
22 Vc=input('Enter the value of complex phase voltage
    VC=') //Enter the value of complex phase voltage
    VC=136.59 +%i*309.80
23 disp('Positive sequence component =');
24 Va1=(Va+(alpha*Vb)+(alpha)^2*Vc)/3
25 mprintf('Va1 = %f + i%f \n',real(Va1),imag(Va1))
26
27 disp('Negative sequence component =');
28 Va2=[Va+(alpha^2)*Vb+(alpha*Vc)]/3
29 mprintf('Va2 = %f + i%f \n',real(Va2),imag(Va2))
30
31 disp('Zero sequence component =');
32 Va0=[Va+Vb+Vc]/3
33 mprintf('Va0 = %f + i%f \n',real(Va0),imag(Va0))
34 end
35
36 if(choice==2)
37
38 Va1=input('Enter the value of positive sequence
    voltage Va1='); // positive sequence voltage Va1
    =200 - %i*100
39 Va2=input('Enter the value of negative sequence
    voltage Va2='); //negative sequence voltage Va2
    =-100
40 Va0=input('Enter the value of zero sequence voltage
    Va0='); //zero sequence voltage Va0=100

```

```

41 disp('Complex voltage Va =');
42
43 Va=[Va1+Va2+Va0]
44 mprintf('Va = %f + i%f \n',real(Va),imag(Va))
45
46 disp('Complex Voltage Vb=');
47 Vb=[(alpha^2*Va1)+(alpha*Va2)+Va0]
48 mprintf('Vb = %f + i%f \n',real(Vb),imag(Vb))
49
50 disp('Complex Voltage Vc=');
51 Vc=[(alpha*Va1)+(alpha^2*Va2)+Va0]
52 mprintf('Vc = %f + i%f \n',real(Vc),imag(Vc))
53 end
54
55 //
56 //output:-
57 //
58 //for phasor to symmetrical :
59 //
60 //Enter your choice:
61 //
62 // 1 for phasor to symmetrical; 2 for symmetrical
   to phasor
63 //Enter your transformation type=1
64 //Enter the value of complex phase voltage VA=200 -
   %i*100
65 //Enter the value of complex phase voltage VB=-36.59
   - %i*209.804
66 //Enter the value of complex phase voltage VC=136.59
   +%i*309.80
67 //
68 // Positive sequence component =
69 //Va1 = 199.994358 + i-99.986083
70 //
71 // Negative sequence component =
72 //Va2 = -99.992891 + i-0.011117
73 //
74 // Zero sequence component =

```

```
75 //Va0 = 100.000000 + i-0.001333
76 //
77 //
78 //
79 //for symmetrical to phasor :
80 //
81 //Enter your choice:
82 //
83 // 1 for phasor to symmetrical; 2 for symmetrical
    to phasor
84 //Enter your transformation type=2
85 //Enter the value of positive sequence voltage Va1
    =200 - %i*100
86 //Enter the value of negative sequence voltage Va2
    =-100
87 //Enter the value of zero sequence voltage Va0=100
88 //
89 // Complex voltage Va =
90 //Va = 200.000000 + i-100.000000
91 //
92 // Complex Voltage Vb=
93 //Vb = -36.591200 + i-209.804400
94 //
95 // Complex Voltage Vc=
96 //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

```
1 //Experiment-12
2 // windows Vista - 32-Bit
3 //Scilab - 5.5.2
4
5
6 // Aim : A SCILAB program for conversion of
   symmetrical components into AC phasors
7
8
9 // To find Symmetrical Components for 3-phase
   Unbalanced System
10
11
12 clc;
13 clear all;
14
15 Q=[3 310
```

```

16     5 105
17     4 12
18         ]
19 i=sqrt(-1);
20 n=3; // Number of Phases
21 a=1.0*(cos((360/n)*%pi/180)+%i*sin((360/n)*%pi/180))
    ;
22 a2=a^2;
23 // Conversion from Polar to Cartesian Form of Matrix
    I
24 Ia=Q(1)*(cos(Q(4)*%pi/180)+%i*sin(Q(4)*%pi/180));
25 Ib=Q(2)*(cos(Q(5)*%pi/180)+%i*sin(Q(5)*%pi/180));
26 Ic=Q(3)*(cos(Q(6)*%pi/180)+%i*sin(Q(6)*%pi/180));
27 // Define symmetrical components transformation
    matrix
28 A= [1     1     1
29     1     a2    a
30     1     a     a2];
31 A012=inv(A)*[Ia;Ib;Ic];
32 // To check the results
33 Add0=A012(1)+A012(2)+A012(3); //This is Ia
34 Add1=A012(1)+a2*A012(2)+a*A012(3); //This is Ib
35 Add2=A012(1)+a*A012(2)+a2*A012(3); // This is Ic
36 // Symmetrical Components in Cartesian Form
37 Symm3= [A012(1) A012(2) A012(3)
38         A012(1) a2*A012(2) a*A012(3)
39         A012(1) a*A012(2) a2*A012(3)];
40 // Symmetrical Components in Polar Form
41 // Sym3Abs=abs(Symm3)
42 [sym3abs, Sym3Angle]=polar(Symm3)
43 sym3angle= Sym3Angle*180/%pi
44 disp(' [Magnitude , Angle] ')
45 mprintf('\n')
46 for i=1:3
47     for j=1:3
48         mprintf(' [%0.3f,%0.3f]      ',sym3abs(i,j),
49                 sym3angle(i,j))

```

```

[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

```

50     mprintf( '\n' )
51 end
52
53 // End of the Program
54
55
56 //output:-
57 //
58 // [Magnitude , Angle]
59 //
60 // [3.817 ,35.558]    [-0.276 , -19.726]
61 // [-0.276 , -19.726]    [3.817 ,60.635]
62 // [-0.276 ,13.835]    [-0.276 ,53.517]
63 // [3.817 , -84.773]

```

Experiment: 13

Modification Of Zbus Matrix For Short Circuit Study Using Scilab Programming

Scilab code Solution 13.13 exp13

```
1 //Experiment-13
2 // windows Vista - 32-Bit
3 //Scilab - 5.5.2
4
5
6 // Aim : Modification of Zbus matrix for short
   circuit study using SCILAB Programming
7
8
9 // Display mode
10 mode(0);
11
12 clc;
13 clear;
14 s=input('Enter Size of Z bus Matrix :')// Size of Z
   bus Matrix like 3 or 4
15 mprintf('Enter %g X %g Matrix ',s,s)
```

```

16
17 Zorg = input('')// Zorg like [1 2 3;4 5 6;7 8 9]
18 disp('Enter 1 for Type 1 2 for Type 2 3 for Type 3
      and 4 for Type 4 Zbus modification')
19 check = input('Enter Your Choice :')
20
21 if check==1 then
22
23 Zb = input ('Enter value of Impidance between New
      Bus and Reference:')// value of Impidance between
      New Bus and Reference like 10 or 20 etc.
24 l = max(size(Zorg));
25 Znew= zeros(l+1,l+1);
26 for i = 1:l+1
27     for j = 1:l+1
28         if i<=(l) & j<=(l) then
29             Znew(i,j)=Zorg(i,j);
30         elseif i==l+1 & j==l+1 then
31             Znew(i,j) = Zb;
32         else
33             Znew(i,j) = Znew(i,j);
34         end;
35     end;
36 end;
37 Znew
38 end;
39
40
41
42 if check==2 then
43 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
44 Zb = input ('Enter value of Impidance between New
      Bus and existing Bus: ')//Impidance between New
      Bus and existing Bus like 10 or 20
45 l=max(size(Zorg));

```

```

46 row =Zorg(bc,:);
47 column =Zorg(:,bc);
48 for i = 1:l + 1
49     for j = 1:l + 1
50         if i<=l & j<=l
51             Znew(i,j)=Zorg(i,j);
52         elseif i==l+1
53             for p=1:l
54                 Znew(i,p)=row(p);
55             end
56         elseif j ==l + 1
57             for q=1:l
58                 Znew(q,j)=column(q);
59             end
60         end
61         if i ==l + 1 & j ==l + 1
62             Znew(i,j)=Zb+Zorg(bc, bc);
63         end
64     end
65 end
66 Znew
67 end
68
69 if check==4 then
70     aa = input('Enter the From Bus number where
        Impedance is Connected :')//From Bus number
        where Impedance is Connected in between
        define matrix like 2
71     bb = input('Enter the To Bus number where
        Impedance is Connected :')// To Bus number
        where Impedance is Connected in between
        define matrix like 3
72 Zb = input ('Enter value of Impidance between
        existing Buses : ')// Impidance between existing
        Buses like 10 or 20
73     l = max(size(Zorg));
74
75     Zb_row = Zorg(aa,:)-Zorg(bb,:);

```

```

76             Zb_col = Zorg(:,aa)-Zorg(:,bb);
77     xy=((Zb_col*Zb_row)/(Zorg(aa,aa)+Zorg(bb,bb)+Zb
           -2*Zorg(aa,bb)));
78
79     Znew = Zorg-xy
80
81
82 end
83     if check==3 then
84         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
85         Zb = input ('Enter value of Impidance
                    between Existing Bus and Reference: ')
            // value of Impidance between Existing
            // Bus and Reference like 10 or 20
86         l = max(size(Zorg));
87
88             Zb_row = Zorg(aa,:);
89             Zb_col = Zorg(:,aa);
90     xy=((Zb_col*Zb_row)/(Zorg(aa,aa)+Zb));
91
92     Znew = Zorg-xy
93 end
94
95 //
96 //output:-
97 //
98 //For Type-1 Zbus Modification
99 //
100 //
101 //Enter Size of Z bus Matrix :3
102 // s =
103 //
104 //     3.
105 //Enter 3 X 3 Matrix
106 //[1 2 3;4 5 6;7 8 9]

```

```

107 // Zorg =
108 //
109 //      1.      2.      3.
110 //      4.      5.      6.
111 //      7.      8.      9.
112 //
113 // Enter 1 for Type 1 2 for Type 2 3 for Type 3 and
      4 for Type 4 Zbus m
114 //      odification
115 //Enter Your Choice :1
116 // check =
117 //
118 //      1.
119 //Enter value of Impidance between New Bus and
      Reference:10
120 // Zb =
121 //
122 //      10.
123 // Znew =
124 //
125 //      1.      2.      3.      0.
126 //      4.      5.      6.      0.
127 //      7.      8.      9.      0.
128 //      0.      0.      0.      10.
129 //
130 //
131 //
132 //For Type-2 Zbus Modification
133 //
134 //
135 //Enter Size of Z bus Matrix :3
136 // s =
137 //
138 //      3.
139 //Enter 3 X 3 Matrix
140 //[1 2 3;4 5 6;7 8 9]
141 // Zorg =
142 //

```

```

143 //      1.      2.      3.
144 //      4.      5.      6.
145 //      7.      8.      9.
146 //
147 // Enter 1 for Type 1 2 for Type 2 3 for Type 3 and
      4 for Type 4 Zbus m
148 //      odification
149 //Enter Your Choice :2
150 // check =
151 //
152 //      2.
153 //Enter the Bus number where Impedance is Connected
      : 2
154 // bc =
155 //
156 //      2.
157 //Enter value of Impidance between New Bus and
      existing Bus: 15
158 // Zb =
159 //
160 //      15.
161 // Znew =
162 //
163 //      1.      2.      3.      2.
164 //      4.      5.      6.      5.
165 //      7.      8.      9.      8.
166 //      4.      5.      6.      20.
167 //
168 //For Type-3 Zbus Modification
169 //
170 //
171 //Enter Size of Z bus Matrix :3
172 // s =
173 //
174 //      3.
175 //Enter 3 X 3 Matrix
176 //[1 2 3;4 5 6;7 8 9]
177 // Zorg =

```

```

178 //
179 //      1.      2.      3.
180 //      4.      5.      6.
181 //      7.      8.      9.
182 //
183 // Enter 1 for Type 1 2 for Type 2 3 for Type 3 and
      4 for Type 4 Zbus m
184 //      odification
185 //Enter Your Choice :3
186 // check =
187 //
188 //      3.
189 //Enter the Bus number where Impedance is Connected
      to Reference :3
190 // aa =
191 //
192 //      3.
193 //Enter value of Impidance between Existing Bus and
      Reference: 20
194 // Zb =
195 //
196 //      20.
197 // Znew =
198 //
199 //      0.2758621      1.1724138      2.0689655
200 //      2.5517241      3.3448276      4.137931
201 //      4.8275862      5.5172414      6.2068966
202 //
203 //For Type-4 Zbus Modification
204 //
205 //
206 //Enter Size of Z bus Matrix :3
207 // s =
208 //
209 //      3.
210 //Enter 3 X 3 Matrix
211 //[1 2 3;4 5 6;7 8 9]
212 // Zorg =

```

```

213 //
214 //      1.      2.      3.
215 //      4.      5.      6.
216 //      7.      8.      9.
217 //
218 // Enter 1 for Type 1 2 for Type 2 3 for Type 3 and
      4 for Type 4 Zbus m
219 //      odification
220 //Enter Your Choice :4
221 // check =
222 //
223 //      4.
224 //Enter the From Bus number where Impedance is
      Connected :2
225 // aa =
226 //
227 //      2.
228 //Enter the To Bus number where Impedance is
      Connected :3
229 // bb =
230 //
231 //      3.
232 //Enter value of Impidance between existing Buses :
      30
233 // Zb =
234 //
235 //      30.
236 // Znew =
237 //
238 //      0.90625      1.90625      2.90625
239 //      3.90625      4.90625      5.90625
240 //      6.90625      7.90625      8.90625

```

```

Enter Size of Z bus Matrix :3
a =
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
a =
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
n =
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
n =
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

```
1 //Experiment-14
2 // windows Vista - 32-Bit
3 //Scilab - 5.5.2
4
5
6 // Aim : Symmetrical Fault analysis using SCILAB
   programming
7
8 mode (0);
9
10 clc;
11 clear all;
12
13 SA = input('Enter Alternator Rating in kVA :');//
   Alternator Rating in kVA=10e3
14 XA = input('Enter Alternator % Reactance :');//
   Alternator % Reactance = 10%
15 ST = input ('Enter Transformer Rating in kVA :');//
   Transformer Rating in kVA =5e3
```

```

16 XT = input ('Enter Transformer % Reactance : '); //
    Transformer % Reactance=5%
17 VL = input ('Enter Transmission line Voltage in kV : '
    ); // Transmission line Voltage in kV=10
18 XL = input ('Enter Transmissin Line Reactance in Ohm
    : '); // Transmissin Line Reactance in Ohm=4
19 RL = input ('enter Transmission Line Resistance in
    Ohm : '); // Transmission Line Resistance in Ohm
    =1
20
21 Base_kVA = max(SA,ST);
22
23 PXA = (Base_kVA/SA)*XA; // Percentage Reactance of
    Alternator
24
25 PXT = (Base_kVA/ST)*XT; // Percentage Reactance of
    Transformer
26
27 PXL = (Base_kVA*XL)/(10*VL*VL); // Percentage
    Reactance of Transmission line
28
29 PRL = (Base_kVA*RL)/(10*VL*VL); // Percentage
    Resistance of Transmission line
30 disp('Fault at F1 means at source side and Fault at
    F2 means at load side')
31 disp('Enter 1 for Fault at F1 or 2 for Fault at F2')
32
33 check = input ('Enter Your Choice for Symmetrical
    Fault in Transmission line : ')
34
35 if check==2 then
36
37 TX = PXA+PXT+PXL; // Total Percentage Reactance
38
39 TR = PRL; // Total Percentage Resistance
40
41 PZ = sqrt(TX*TX + TR*TR); // Total Percentage
    Impidance

```

```

42
43 Short_Circuit_kVA = Base_kVA*(100/PZ);
44 Short_Circuit_Current= Short_Circuit_kVA/(sqrt(3)*VL
    );
45 mprintf('Short Circuit kVA for Fault at F2 : %f kVA
    \n',Short_Circuit_kVA)
46 mprintf('Short Circuit Current for Fault at F2 : %f
    A \n',Short_Circuit_Current)
47 end
48
49 if check==1 then
50     TX = PXA+PXT; // Total Percentage Reactance
51
52 Short_Circuit_kVA1 = Base_kVA*(100/TX);
53 Short_Circuit_Current1= Short_Circuit_kVA1/(sqrt(3)*
    VL);
54 mprintf('Short Circuit kVA for Fault at F1 : %f kVA
    \n',Short_Circuit_kVA1)
55 mprintf('Short Circuit Current for Fault at F1 : %f
    A \n',Short_Circuit_Current1)
56 end
57
58
59 //output:-
60 //
61 //Symmetrical Fault F1 at source side:
62 //
63 //Enter Alternator Rating in kVA :10e3
64 //Enter Alternator % Reactance :10
65 //Enter Transformer Rating in kVA :5e3
66 //Enter Transformer % Reactance :5
67 //Enter Transmission line Voltage in kV :10
68 //Enter Transmissin Line Reactance in Ohm :4
69 //enter Transmission Line Resistance in Ohm : 1
70 //
71 // Fault at F1 means at source side and Fault at F2
    means at load side
72 //

```

```

73 // Enter 1 for Fault at F1 or 2 for Fault at F2
74 //Enter Your Choice for Symmetrical Fault in
    Transmission line : 1
75 // check =
76 //
77 //      1.
78 //Short Circuit kVA for Fault at F1 : 50000.000000
    kVA
79 //Short Circuit Current for Fault at F1 :
    2886.751346 A
80 //
81 //Symmetrical Fault F2 at laod side:
82 //
83 //Enter Alternator Rating in kVA :10e3
84 //Enter Alternator % Reactance :10
85 //Enter Transformer Rating in kVA :5e3
86 //Enter Transformer % Reactance :5
87 //Enter Transmission line Voltage in kV :10
88 //Enter Transmissin Line Reactance in Ohm :4
89 //enter Transmission Line Resistance in Ohm : 1
90 //
91 // Fault at F1 means at source side and Fault at F2
    means at load side
92 //
93 // Enter 1 for Fault at F1 or 2 for Fault at F2
94 //Enter Your Choice for Symmetrical Fault in
    Transmission line : 2
95 // check =
96 //
97 //      2.
98 //Short Circuit kVA for Fault at F2 : 16439.898731
    kVA
99 //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 Symmetrical 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