SRM Valliammai Engineering College

An Autonomous Institution

SRM Nagar, Kattankulathur - 603203

Department of Electrical and Electronics Engineering

1905611 – Power system Simulation Laboratory Manual

LAB MANUAL

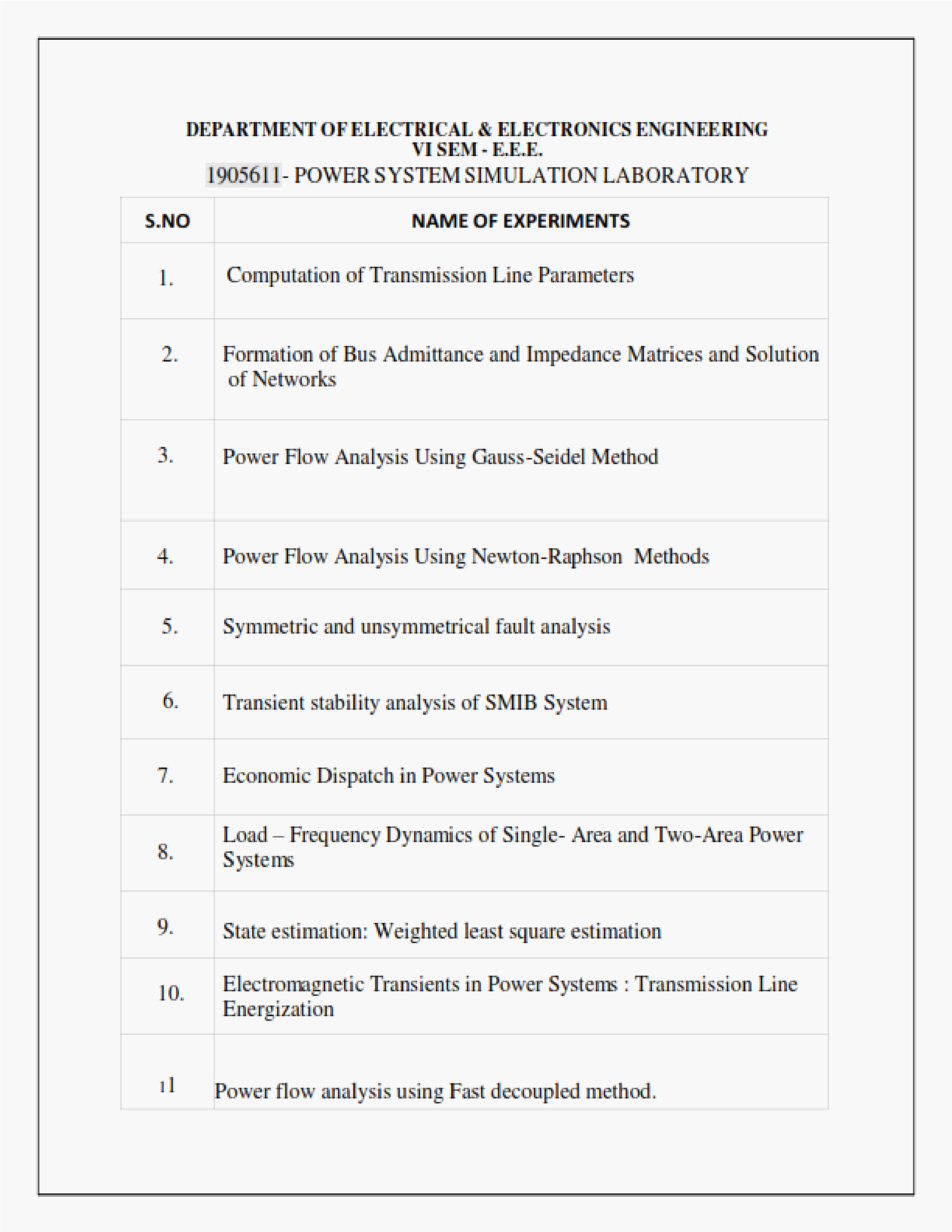
III Year- VI Semester - Electrical and Electronics Engineering

Academic Year 2021-2022

(2019 Regulation)

Prepared by,

Dr.T.Santhosh Kumar/A.P- (Sr.G)



DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING
VI SEM - E.E.E.

1905611- POWER SYSTEM SIMULATION LABORATORY

S.NO	NAME OF EXPERIMENTS
1.	Computation of Transmission Line Parameters
2.	Formation of Bus Admittance and Impedance Matrices and Solution of Networks
3.	Power Flow Analysis Using Gauss-Seidel Method
4.	Power Flow Analysis Using Newton-Raphson Methods
5.	Symmetric and unsymmetrical fault analysis
6.	Transient stability analysis of SMIB System
7.	Economic Dispatch in Power Systems
8.	Load – Frequency Dynamics of Single- Area and Two-Area Power Systems
9.	State estimation: Weighted least square estimation
10.	Electromagnetic Transients in Power Systems : Transmission Line Energization
11	Power flow analysis using Fast decoupled method.

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

VII SEM – E.E.E.

1905611- POWER SYSTEM SIMULATION LABORATORY

CYCLE I

1. Computation of Transmission Line Parameters.
2. Formation of Bus Admittance and Impedance Matrices and Solution of Networks.
3. Power Flow Analysis Using Gauss-Seidel Method.
4. Power Flow Analysis Using Newton-Raphson Method.
5. Symmetric and unsymmetrical fault analysis.

CYCLE II

6. Transient and Small Signal Stability Analysis: Single-Machine Infinite Bus System.
7. Economic Dispatch in Power Systems.
8. Load – Frequency Dynamics of Single- Area and Two-Area Power Systems.
9. State estimation: Weighted least square estimation.
10. Electromagnetic Transients in Power Systems : Transmission Line Energization.
11. Power flow analysis using Fast decoupled method.

ADDITIONAL EXPERIMENTS

1. Load flow analysis of a given power system with STATCOM
2. Transient analysis of single machine infinite bus system with STATCOM
3. Transient Stability Analysis of Multi machine Power Systems

EXP.NO:1 (A)
DATE:

COMPUTATION TRANSMISSION LINE OF PARAMETERS

AIM:

To determine the positive sequence line parameters L and C per phase per kilometer of a single phase, three phase single and double circuit transmission lines for different conductor arrangements.

SOFTWARE REQUIRED: MATLAB

THEORY:

Transmission line has four parameters – resistance, inductance, capacitance and conductance. The inductance and capacitance are due to the effect of magnetic and electric fields around the conductor. The resistance of the conductor is best determined from the manufactures data, the inductances and capacitances can be evaluated using the formula.

ARRANGEMENT	INDUCTANCE	CAPACITANCE
SINGLE PHASE SYSTEM	$L_c=2*10^{-7}\ln(D/r')$ $r'=0.7788r.$ $L_{loop}=2L_c$	$C_c=2\pi\xi_0/\ln(D/r)$ $C_{loop}=C_c/2.$
3PHASE SYMMETRICAL SYSTEM	$L_c=L_{ph}=2*10^{-7}\ln(D/r')$	$C_c=C_{ph}=2\pi\xi_0/\ln(D/r)$
3PHASE UNSYMMETRICAL TRANSPOSED SYSTEM	$L_c=L_{ph}=2*10^{-7}\ln(D_{eq}/r')$ $D_{eq}=(D_{AB}*D_{BC}*D_{CA})^{(1/3)}$	$C_c=C_{ph}=2\pi\xi_0/\ln(D_{eq}/r)$
3PHASE UNSYMMETRICAL UNTRANSPOSED SYSTEM	$L_a=2*10^{-7}[\ln\surd(D_{ab}*D_{ca})/r'$ $+j\surd3*\ln\surd(D_{ab}/D_{ca})$ $L_b=2*10^{-7}[\ln\surd(D_{bc}*D_{ab})/r'$ $+j\surd3*\ln\surd(D_{bc}/D_{ab})$ $L_c=2*10^{-7}[\ln\surd(D_{ca}*D_{bc})/r'$ $+j\surd3*\ln\surd(D_{ca}/D_{bc})$	------------

3PHASE SYMMETRICAL DOUBLECIRCUIT SYSTEM	$L_c=2*10^{-7}\ln(\sqrt{3}D/2r')$ $L_{ph}=L_c/2$	$C_c=2\pi\xi_0/\ln(\sqrt{3}D/2r)$ $C_{ph}=C_c*2.$
3PHASE UNSYMMETRICAL TRANSPOSED SYSTEM WITH VERTICAL PROFILE	$L_c=2*10^{-7}\ln[2^{(1/3)}(D/r')(m/n)^{(2/3)}]$ $L_{ph}=L_c/2$	$C_c=2\pi\xi_0/\ln(2^{(1/3)}(D/r)(m/n)^{(2/3)}]$ $C_{ph}=C_c*2.$
3PHASE UNSYMMETRICAL TRANSPOSED DOUBLE CIRCUIT	$L_c=2*10^{-7}\ln(D_m/D_s)$ $L_{ph}=L_c/2$	$C_c=2\pi\xi_0/\ln[i^2m^2jh/r^3n^3d]^{(1/3)}$ $C_{ph}=C_c*2.$
3PHASE LINE WITH BUNDELED CONDUCTORS	$L_c=2*10^{-7}\ln(D_m/D_s)$ $L_{ph}=L_c/2$ $D_m=(D_{AB}*D_{BC}*D_{CA})^{(1/3)}$ $D_s=(D_{SA}*D_{SB}*D_{SC})^{(1/3)}$	--------

PROCEDURE:

1. Enter the command window of the MATLAB.

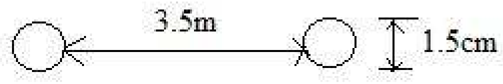
2. Create a new M – file by selecting File - New – M – File

3. Type and save the program in the editor window.

4. Execute the program by either pressing Tools – Run.

5. View the results.

1. (a) Calculate the loop inductance and capacitance of a 1 phase line with two parallel conductors spaced 3.5m apart. The diameter of each conductor is 1.5 cm.



Manual Calculation:

1(a) CALCULATION OF INDUCTANCE AND CAPACITANCE OF SINGLE PHASE LINE

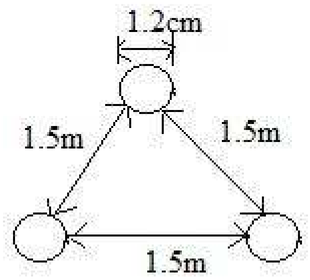
PROGRAM:

```
clc;
clear all;
disp('CALCULATION OF INDUCTANCE AND CAPACITANCE OF 1 PHASE LINE');
d=input('Enter diameter in cm:');
r=d/2;
rad=r*10^(-2);
D=input('Enter distance between conductors in m:');
r1=rad*0.7788;
 L=4*10^(-7)*log(D/r1);
C=(pi*8.854*10^(-12))/(log(D/rad));
 disp('INDUCTANCE(in H/m):');
 disp(L);
disp('CAPACITANCE(in F/m):');
disp(C);
```

OUTPUT:

```
CALCULATION OF INDUCTANCE AND CAPACITANCE OF 1 PHASE LINE

Enter diameter in cm: 1.5

Enter distance between conductors in m: 3.5

INDUCTANCE (in H/m):  2.5582e-006

CAPACITANCE (in F/m):  4.5261e-012
```

1. (b) Calculate the inductance and capacitance of a conductor of a 3 phase system shown which has 1.2 cm diameter and conductors at the edge of an equilateral triangle of side 1.5m.



Manual Calculation:

1b) CALCULATION OF INDUCTANCE AND CAPACITANCE OF THREE PHASE SYMMETRIC LINE

PROGRAM:

```
clc;
clear all;
disp('CALCULATION OF INDUCTANCE AND CAPACITANCE OF 3 PHASE SYMMETRIC
LINE');
d=input('Enter diameter in cm:');
r=d/2;
rad=r*10^(-2);
D=input('Enter distance between conductors in m:');
r1=rad*0.7788;
L=2*10^(-7)*log(D/r1);
C=(2*pi*8.854*10^(-12))/(log(D/rad));
disp('INDUCTANCE(in H/m):');
disp(L);
disp('CAPACITANCE(in F/m):');
disp(C);
```

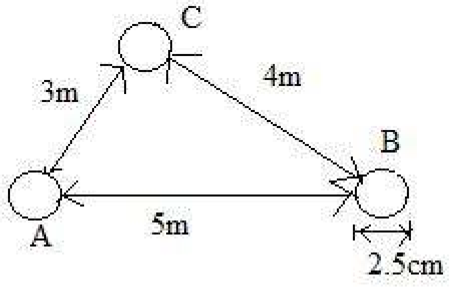
OUTPUT:

```
CALCULATION OF INDUCTANCE AND CAPACITANCE OF 3 PHASE SYMMETRIC LINE

Enter diameter in cm: 1.2

Enter distance between conductors in m: 1.5

INDUCTANCE (in H/m):  1.1543e-006

CAPACITANCE (in F/m):  1.0075e-011
```

1. (c) Calculate the inductance, capacitance and reactance of 3 phase 50 Hz overhead transmission line which has conductors of 2.5cm diameter. Distance between conductors are

5m between A & B
4m between B & C
3m between C & A

Assume conductors are transposed regularly



Manual Calculation:

1c) CALCULATION OF INDUCTANCE AND CAPACITANCE OF THREE PHASE UNSYMMETRIC TRANSPOSED LINE

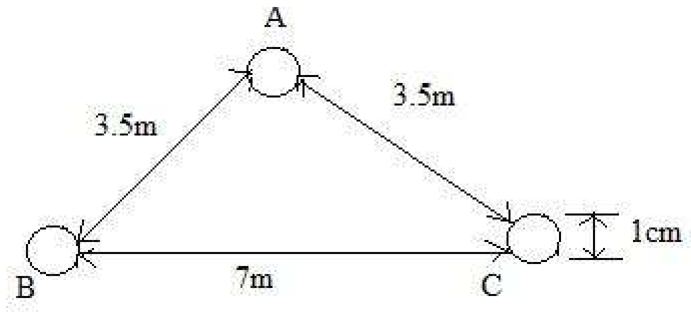
PROGRAM:

```
clc;
clear all;
disp('CALCULATION OF INDUCTANCE AND CAPACITANCE OF 3 PHASE UNSYMMETRIC
LINE - TRANSPOSED');
d=input('Enter diameter in cm:');
r=d/2;
rad=r*10^(-2);
Dab=input('Enter distance between conductors A & B in m:');
Dbc=input('Enter distance between conductors B & C in m:');
Dca=input('Enter distance between conductors C & A in m:');
f=input('Enter Frequency');
Deq=(Dab*Dbc*Dca)^(1/3);
r1=rad*0.7788;
L=2*10^(-7)*log(Deq/r1);
C=(2*pi*8.854*10^(-12))/(log(Deq/rad));
disp('INDUCTANCE(in H/m):');
disp(L);
disp('CAPACITANCE(in F/m):');
disp(C);
XL=2*pi*f*L;
XC=1/(2*pi*f*C);
disp('INDUCTIVER REACTANCE(in ohm/m):');
disp(XL);
disp('CAPACITIVE REACTANCE(in ohm/m):');
disp(XC);
```

OUTPUT:

```
CALCULATION OF INDUCTANCE AND CAPACITANCE OF 3 PHASE UNSYMMETRIC LINE
- TRANSPOSED

Enter diameter in cm: 2.5

Enter distance between conductors A & B in m: 5

Enter distance between conductors B & C in m: 4

Enter distance between conductors C & A in m: 3

Enter Frequency50

INDUCTANCE (in H/m):

  1.1994e-006


CAPACITANCE (in F/m):

  9.6804e-012


INDUCTIVER REACTANCE (in ohm/m):

  3.7679e-004


CAPACITIVE REACTANCE (in ohm/m):

  3.2882e+008
```

1. (d) Calculate the inductance and capacitance per phase of a 3 phase transmission line as shown in figure. Radius of conductor is 0.5 cm. Lines are untransposed.



Manual Calculation:

1d) CALCULATION OF INDUCTANCE AND CAPACITANCE OF THREE PHASE UNSYMMETRIC UNTRANSPOSED LINE

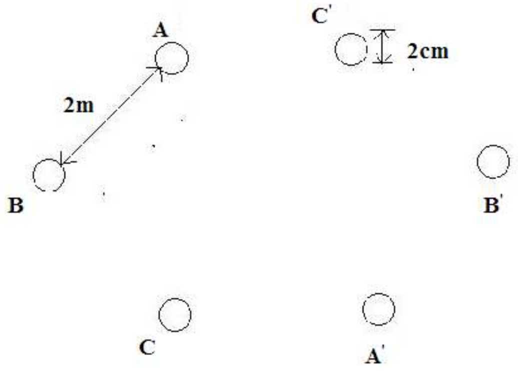
PROGRAM:

```
clc;
clear all;
disp('CALCULATION OF INDUCTANCE OF 3 PHASE UNSYMMETRIC LINE -
UNTRANSPOSED');
r=input('Enter radius in cm:');
rad=r*10^(-2);
Dab=input('Enter distance between conductors A & B in m:');
Dbc=input('Enter distance between conductors B & C in m:');
Dca=input('Enter distance between conductors C & A in m:');
r1=rad*0.7788;
La=2*10^(-7)*(log(1/r1)+log((Dab*Dca)^(1/2))+(3)^(1/2)*j*log((Dab/Dca)^(1/2)));
Lb=2*10^(-7)*(log(1/r1)+log((Dbc*Dab)^(1/2))+(3)^(1/2)*j*log((Dbc/Dab)^(1/2)));
Lc=2*10^(-7)*(log(1/r1)+log((Dca*Dbc)^(1/2))+(3)^(1/2)*j*log((Dca/Dbc)^(1/2)));

disp('INDUCTANCE(in H/m):');
disp(La);
disp(Lb);
disp(Lc);
```

OUTPUT:

```
CALCULATION OF INDUCTANCE OF 3 PHASE UNSYMMETRIC LINE - UNTRANSPOSED

Enter radius in cm: 0.5

Enter distance between conductors A & B in m: 3.5

Enter distance between conductors B & C in m: 3.5

Enter distance between conductors C & A in m: 7

INDUCTANCE (in H/m):

  1.4295e-006 -1.2006e-007i

  1.3602e-006

  1.4295e-006 +1.2006e-007i
```

1 (e) Calculate the inductance and capacitance of a 3 phase double circuit line as in the figure if the conductors are spaced 2m apart at the vertices of a hexagon and diameter of conductors is 2cm.



Manual Calculation:

1e) CALCULATION OF INDUCTANCE AND CAPACITANCE OF THREE PHASE SYMMETRIC DOUBLE CIRCUIT LINE

PROGRAM:

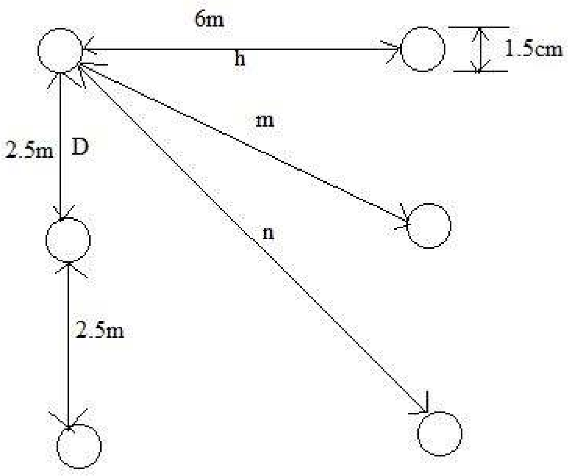
```
clc;
clear all;
disp('CALCULATION OF INDUCTANCE AND CAPACITANCE OF 3 PHASE DOUBLE
CIRCUIT - SYMMETRIC');
d=input('Enter diameter in cm:');
r=d/2;
rad=r*10^(-2);
D=input('Enter distance between conductors(side of hexagon) in m:');
r1=rad*0.7788;
L=10^(-7)*log((3)^(1/2)*D/(2*r1));
C=(4*pi*8.854*10^(-12))/(log((3)^(1/2)*D/(2*rad)));
disp('INDUCTANCE(in H/m):');
disp(L);
disp('CAPACITANCE(in F/m):');
disp(C);
```

OUTPUT:

```
CALCULATION OF INDUCTANCE AND CAPACITANCE OF 3 PHASE DOUBLE CIRCUIT –
SYMMETRIC
Enter diameter in cm: 2
Enter distance between conductors (side of hexagon) in m: 2
INDUCTANCE (in H/m):  5.4045e-007
CAPACITANCE (in F/m):  2.1586e-011
```

1 (f) Calculate the inductance and capacitance per phase of a 3phase double circuit as shown in the figure. Diameter of each conductor is 1.5cm.

Manual Calculation:



1f) CALCULATION OF INDUCTANCE AND CAPACITANCE OF THREE PHASE UNSYMMETRIC, TRANSPOSED DOUBLE CIRCUIT LINE WITH VERTICAL PROFILE

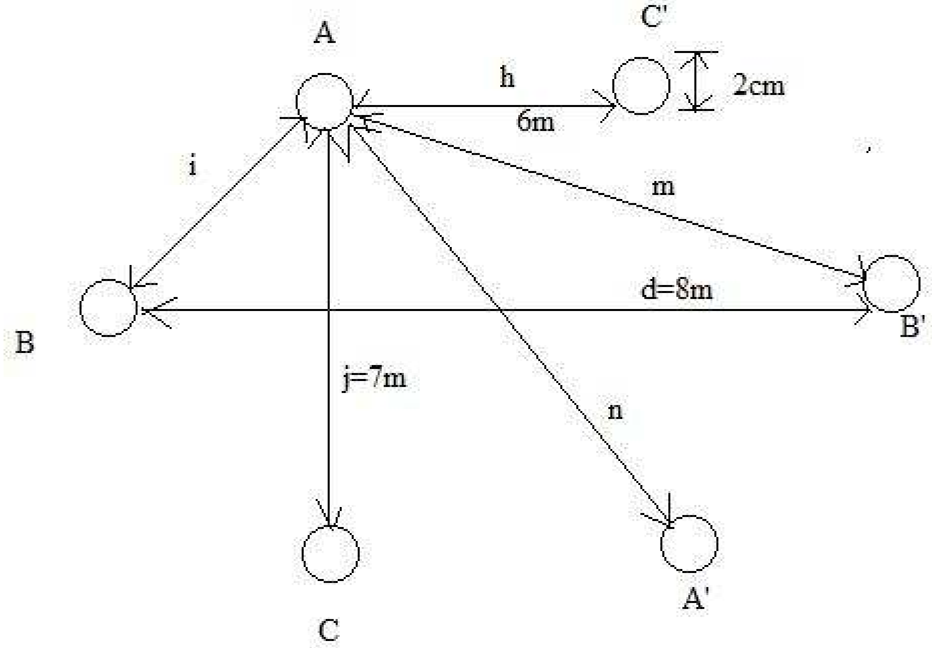
PROGRAM:

```
clc;
clear all;
disp('CALCULATION OF INDUCTANCE AND CAPACITANCE OF 3 PHASE DOUBLE
CIRCUIT - UNSYMMETRIC & TRANSPOSED');
epsilon=8.854*10^(-12);
dia=input('Enter diameter in cm:');
r=dia/2;
rad=r*10^(-2);
h=input('Enter distance h in m:');
D=input('Enter distance D in m:');
m=((D)^2+(h)^2)^(1/2);
n=((2*D)^2+(h)^2)^(1/2);
r1=rad*0.7788;
L=2*10^(-7)*log((2)^(1/6)*(D/r1)^(1/2)*(m/n)^(1/3));
C=4*pi*epsilon/(log(nthroot(2,3)*(D/rad)*(nthroot((m/n),(2/3)))));
disp('INDUCTANCE(in H/m):');
disp(L);
disp('CAPACITANCE(in F/m):');
disp(C);
```

OUTPUT:

```
CALCULATION OF INDUCTANCE AND CAPACITANCE OF 3 PHASE DOUBLE CIRCUIT -
UNSYMMETRIC & TRANSPOSED

Enter diameter in cm: 1.5

Enter distance h in m: 6

Enter distance D in m: 2.5

INDUCTANCE (in H/m):

  6.1678e-007
CAPACITANCE (in F/m):

  1.9301e-011
```

1 (g) Calculate inductance and capacitance per phase of a 3 phase double circuit as shown in the figure. Diameter of each conductor is 2cm. Line is transposed.



Manual Calculation:

1g) CALCULATION OF INDUCTANCE AND CAPACITANCE OF THREE PHASE UNSYMMETRIC, TRANSPOSED DOUBLE CIRCUIT LINE WITHOUT VERTICAL PROFILE

PROGRAM:

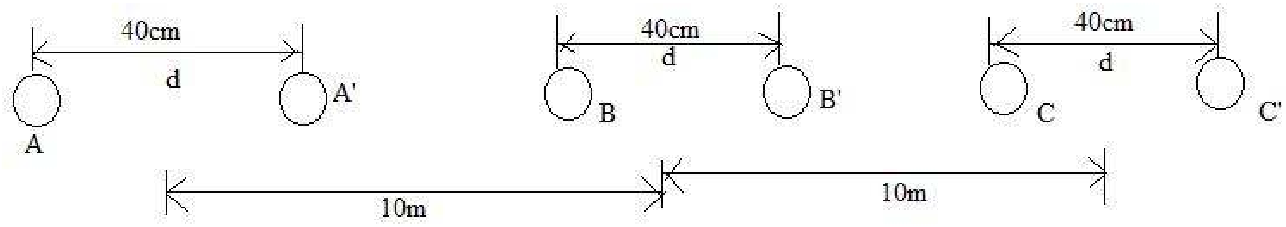
```
clc;
clear all;
disp('CALCULATION OF INDUCTANCE AND CAPACITANCE OF 3 PHASE DOUBLE
CIRCUIT –
UNSYMMETRIC & NON VERTICAL');
epsilon=8.854*10^(-12);
dia=input('Enter diameter in cm:');
r=dia/2;
rad=r*10^(-2);
h=input('Enter distance h in m:');
D=input('Enter distance D in m:');
offset=input('Enter offset distance in m:');
d=h+2*offset;
```

```
i=((offset)^2+(D)^2)^(1/2);
f=((h)^2+(2*D)^2)^(1/2);
g=((h+offset)^2+(D)^2)^(1/2);
j=D*2;
r1=rad*0.7788;
Dm=nthroot((i*i*g*g*h*h*j*j*i*i*g*g),12);
Ds=nthroot((r1*r1*f*f*r1*r1*d*d*r1*r1*f*f),12);
L=2*10^(-7)*log(Dm/Ds);
C=4*pi*epsilon/((1/3)*(log(((i)^2*(g)^2*j*h)/((rad^3)*(f^2)*d))));
disp('INDUCTANCE(in H/m):');
disp(L);
disp('CAPACITANCE(in F/m):');
disp(C);
```

OUTPUT:

```
CALCULATION OF INDUCTANCE AND CAPACITANCE OF 3 PHASE DOUBLE CIRCUIT –
UNSYMMETRIC & NON VERTICAL

Enter diameter in cm: 2

Enter distance h in m: 6

Enter distance D in m: 3.5

Enter offset distance in m: 1

INDUCTANCE (in H/m):

  6.1600e-007

CAPACITANCE (in F/m):

  1.8826e-011
```

1 (h) A 300 KV, 3 phase bundled conductor with sub conductors per phase has a horizontal configuration as in the figure. Find inductance per phase and capacitance if the radius of each subconductor is 1.2cm.



Manual Calculation:

1h) CALCULATION OF INDUCTANCE OF THREE PHASE LINE WITH BUNDLED CONDUCTORS

PROGRAM:

```
clc;
clear all;
disp('CALCULATION OF INDUCTANCE OF 3 PHASE BUNDLED CONDUCTORS');
epsilon=8.854*10^(-12);
r=input('Enter radius in cm:');
rad=r*10^(-2);
h=input('Enter distance between conductors in m:');
D=input('Enter distance between two phases in m:');
r1=rad*0.7788;
Dm=nthroot((D*(D+h)*D*(D-h)*D*(D+h)*D*(D-h)*(2*D)*((2*D)+h)*((2*D)-h)*2*D),12);
Ds=nthroot((r1*r1*h*h*r1*r1*h*h*r1*r1*h*h),12);
L=2*10^(-7)*log(Dm/Ds);
disp('INDUCTANCE(in H/m):');
disp(L);
```

OUTPUT:

```
CALCULATION OF INDUCTANCE OF 3 PHASE BUNDLED CONDUCTORS

Enter radius in cm: 1.2

Enter distance between conductors in m: 4

Enter distance between two phases in m: 10   INDUCTANCE (in H/m):  8.2889e-007
```

1(i) A three-phase transposed line composed of one ACSR, 1,43,000 cmil, 47/7 Bobolink conductor per phase with flat horizontal spacing of 11m between phases a and b and between phases b and c. The conductors have a diameter of 3.625 cm and a GMR of 1.439 cm. The line is to be replaced by a three-conductor bundle of ACSR 477,000-cmil, 26/7 Hawk conductors having the same cross sectional area of aluminum as the single-conductor line. The conductors have a diameter of 2.1793 cm and a GMR of 0.8839 cm. The new line will also have a flat horizontal configuration, but it is to be operated at a higher voltage and therefore the phase spacing is increased to 14m as measured from the center of the bundles. The spacing between the conductors in the bundle is 45 cm.

Determine the inductance and capacitance per phase per kilometer of the above two lines.

Manual Calculation:

PROGRAM:

```
clc;
clear all;
disp('CALCULATION OF INDUCTANCE AND CAPACITANCE');
D=input('Enter the diameter');
Dab=input('Dab=');
Dbc=input('Dbc=');
Dca=input('Dca=');
d=input('Enter the spacing');
r=d/2;
GMD=[Dab*Dbc*Dca]^(1/3);
disp(GMD);
GMR=(D*d^3)^(1/4);
GMR1=1.09*GMR;
disp(GMR1);
C=0.0556/log(GMD/GMR);
L=0.2*log(GMD/GMR);
disp('INDUCTANCE VALUE IN HENRY');
disp(L);
disp('CAPACILANCE VALUE IN FARAD');
disp(C);
```

OUTPUT:

```
CALCULATION OF INDUCTANCE AND CAPACITANCE
Enter the diameter.03625
Dab=11
Dbc=11
Dca=22
Enter the spacing0.045
   13.8591
    0.0465

  INDUCTANCE VALUE IN HENRY    1.1568    CAPACILANCE VALUE IN FARAD    0.0096
```

EXP.NO:1 (B)
DATE:

MODELLING OF TRANSMISSION LINES

AIM:

To understand modeling and performance of short, medium and long transmission lines.

SOFTWARE REQUIRED: MATLAB

FORM ULAE:

$$V_s=AV_R+BI_R$$

$$I_s=CV_R+DI_R$$

TYPE	METHOD	ABCD PARAMETERS
Short	-------------------	A=D=1; B=Z; C=0.

Medium	Nominal T Method	A=D=1+YZ/2;
		B=Z(1+YZ/4);
		C=Y;

	Nominal π Method	A=D=1+YZ/2;
		B=Z;
		C=Y(1+YZ/4);

Long	Rigorous Method	A=D=cos h(γℓ);
		B=Z_c sin h(γℓ);
		C=1/Z_c sin h(γℓ);
		γ=√(ZY);
		Z_c=√(Z/Y);

	Equivalent π Method	A=D=1+YZ/2;
		B=Z;
		C=Y(1+YZ/A);
		Z=Z sin h(γℓ)/ γℓ;
		Y=Y tan h(γℓ/2)/ (γℓ/2);
		γ=√(ZY);
		Z_c=√(Z/Y);

	Equivalent T Method	A=D=1+YZ/2;
		B=Z;
		C=Y(1+YZ/A);
		Z=Z tan h(γℓ/2)/ (γℓ/2);
		Y=Y sin h(γℓ)/ γℓ;
		γ=√(ZY);
		Z_c=√(Z/Y);

PROCEDURE:

1. Enter the command window of the MATLAB.

2. Create a new M – file by selecting File - New – M – File

3. Type and save the program in the editor window.

4. Execute the program by either pressing Tools – Run.

5. View the results.

1. An overhead 3 phase transmission line delivers 4000KW at 11 KV at 0.8 pf lagging. The resistance and reactance of each conductor are 1.5Ω and 4Ω per phase. Determine the line performance.

   Manual Calculation:

SHORT TRANSMISSION LINE

PROGRAM:

```
clc;
clear all;
R=input('Resistance                              :');
XL=input('Inductive Reactance                          :');
XC=input('Capacitive Reactance                         :');
G=input('Conductance                                 :');
length=input('Length of Transmission Line               :');
f=input('Frequency                                    :');
Z1=    (R+j*XL)*length;
Y1=   (G+j*XC)*length;
A = 1;
B = Z1;
C = 0;
D =1;
TM = [ A B; C D ];
VRL=input('ENTER RECEIVEING END VOLTAGE                :');
VRP=VRL/(sqrt(3));
PR = input('ENTER RECEIVING END LOAD IN MW             :');
Pf=input('ENTER THE RECEIVING END LOAD POWER FACTOR    :');
h=acos(Pf);
SR=PR/Pf;
SR=SR*(cos(h)+j*sin(h));
QR=imag(SR);
IR=conj(SR)/(3*conj(VRP));
SM=TM*[VRP;IR];
VS=SM(1,1);
IS=SM(2,1);
Pfs=cos(angle(VS)-angle(IS));
SS=3*VS*conj(IS);
VSA=angle(VS)*(180/pi);
ISA=angle(IS)*(180/pi);
VS=sqrt(3)*abs(VS);
IS=abs(IS)*1000;
VREG=((VS/(abs(TM(1,1)))-VRL)/VRL)*100;
PS=real(SS);
QS=imag(SS);
eff=PR/PS*100;
PL=PS-PR;
```

```
QL=QS-QR;
Z1
Y1
TM
fprintf('SENDING END LINE VOLTAGE %g at %g degrees \n',VS,VSA);
fprintf('SENDING END LINE CURRENT %g at %g degrees \n',IS,ISA);
fprintf('SENDING END POWER FACTOR %g\n',Pfs);
fprintf('SENDING END REAL POWER %g\n',PS);
fprintf('SENDING END REACTIVE POWER %g\n',QS);
fprintf('PERCENTAGE VOLTAGE REGULATION  %g\n',VREG);
fprintf('REAL POWER LOSS %g\n',PL);
fprintf('REACTIVE POWER LOSS %g\n',QL);
fprintf('EFFICIENCY %G', eff);
```

<u>OUTPUT:</u>

```
Resistance                              : 1.5
Inductive Reactance                         : 4
Capacitive Reactance                        : 0
Conductance                               : 0
Length of Transmission Line              : 1
Frequency                               : 50
ENTER RECEIVEING END VOLTAGE                : 11
ENTER RECEIVING END LOAD IN MW              : 4
ENTER THE RECEIVING END LOAD POWER FACTOR    : 0.8

Z1 =
   1.5000 + 4.0000i

Y1 =
     0
TM =

   1.0000             1.5000 + 4.0000i
        0          1.0000

SENDING END LINE VOLTAGE 12.6795 at 4.72953 degrees
SENDING END LINE CURRENT 262.432 at -36.8699 degrees
SENDING END POWER FACTOR 0.747805
SENDING END REAL POWER 4.30992
SENDING END REACTIVE POWER 3.82645
PERCENTAGE VOLTAGE REGULATION 15.2685
REAL POWER LOSS 0.309917
REACTIVE POWER LOSS 0.826446
EFFICIENCY 92.8092
```

2. A balanced 3 phase load of 30 MW is supplied at 132KV, 50Hz and 0.85 pf lag by means of a line. The series impedance is 20+j52Ω and total admittance is $315*10^{-6}$℧.Using Normal T method determine A,B,C,D parameters and regulation.

Manual Calculation:

MEDIUM TRANSMISSION LINE

PROGRAM:

```
clc;
clear all;
R=input('Resistance                         :');
XL=input('Inductive Reactance                 :');
XC=input('Capacitive Reactance                :');
G=input('Conductance                        :');
length=input('Length of Transmission Line         :');
f=input('Frequency                          :');
Z1= (R+j*XL)*length;
Y1= (G+j*XC)*length;
m=menu('ENTER THE TYPE OF NETWORK','NOMINAL T', 'NOMINAL PI');
switch m
   case {1}
      A = 1+(Z1*Y1/2);
      B=Z1*(1+(Z1*Y1/4));
      C=Y1;
      D=A;
   otherwise
      A = 1+(Z1*Y1/2);
      B=Z1;
      C=Y1*(1+(Z1*Y1/4));
      D=A;
end
TM = [ A B; C D ];
VRL=input('ENTER RECEIVEING END VOLTAGE           :');
VRP=VRL/(sqrt(3));
PR = input('ENTER RECEIVING END LOAD IN MW         :');
Pf=input('ENTER THE RECEIVING END LOAD POWER FACTOR    :');
h=acos(Pf);
SR=PR/Pf;
SR=SR*(cos(h)+j*sin(h));
QR=imag(SR);
IR=conj(SR)/(3*conj(VRP));
SM=TM*[VRP;IR];
VS=SM(1,1);
IS=SM(2,1);
Pfs=cos(angle(VS)-angle(IS));
SS=3*VS*conj(IS);
```

```
VSA=angle(VS)*(180/pi);
ISA=angle(IS)*(180/pi);
VS=sqrt(3)*abs(VS);
IS=abs(IS)*1000;
VREG=((VS/(abs(TM(1,1)))-VRL)/VRL)*100;
PS=real(SS);
QS=imag(SS);
eff=PR/PS*100;
PL=PS-PR;
QL=QS-QR;
Z1
Y1
TM
fprintf('SENDING END LINE VOLTAGE %g at %g degrees \n',VS,VSA);
fprintf('SENDING END LINE CURRENT %g at %g degrees \n',IS,ISA);
fprintf('SENDING END POWER FACTOR %g\n',Pfs);
fprintf('SENDING END REAL POWER %g\n',PS);
fprintf('SENDING END REACTIVE POWER %g\n',QS);
fprintf('PERCENTAGE VOLTAGE REGULATION  %g\n',VREG);
fprintf('REAL POWER LOSS %g\n',PL);
fprintf('REACTIVE POWER LOSS %g\n',QL);
fprintf('EFFICIENCY %G', eff);
```

NOMINAL T

OUTPUT:

```
Resistance                          : 20
Inductive Reactance                     : 52
Capacitive Reactance                    : 315*10^(-6)
Conductance                           : 0
Length of Transmission Line             : 1
Frequency                           : 50
ENTER RECEIVEING END VOLTAGE              : 132
ENTER RECEIVING END LOAD IN MW             : 30
ENTER THE RECEIVING END LOAD POWER FACTOR    : 0.85
Z1 =
  20.0000 +52.0000i
Y1 =
        0 +3.1500e-004i
TM =
   0.9918 + 0.0031i 19.8362 +51.8186i
        0 + 0.0003i   0.9918 + 0.0031i
SENDING END LINE VOLTAGE 143.035 at 3.76761 degrees
SENDING END LINE CURRENT 142.007 at -23.3284 degrees
SENDING END POWER FACTOR 0.890244
SENDING END REAL POWER 31.3199
SENDING END REACTIVE POWER 16.0245
PERCENTAGE VOLTAGE REGULATION 9.25407
REAL POWER LOSS 1.31989
REACTIVE POWER LOSS -2.56785
EFFICIENCY 95.7858
```

3. A 50Hz, 3 phase, 100 km transmission line has total impedance of 35Ω, reactance 0f 140Ω and shunt admittance of $930*10^{-6}$ Ʊ . It delivers 40 MW at 220KV, 0.9 pf lag .Using nominal π determine A,B ,C,D V_s, V_{SA},I_{SA} , pf, P_s Q_s,η.

Manual Calculation:

NOMINAL PI

OUTPUT:

```
Resistance                          : 20
Inductive Reactance                     : 52
Capacitive Reactance                    : 315*10^(-6)
Conductance                           : 0
Length of Transmission Line             : 1
Frequency                           : 50
ENTER RECEIVEING END VOLTAGE              : 132
ENTER RECEIVING END LOAD IN MW            : 30
ENTER THE RECEIVING END LOAD POWER FACTOR    : 0 .85

Z1 =
  20.0000 +52.0000i
Y1 =
        0 +3.1500e-004i
TM =

   0.9918 + 0.0031i 20.0000 +52.0000i
  -0.0000 + 0.0003i   0.9918 + 0.0031i

SENDING END LINE VOLTAGE 143.099 at 3.77321 degrees
SENDING END LINE CURRENT 142.011 at -23.3709 degrees
SENDING END POWER FACTOR 0.889862
SENDING END REAL POWER 31.3214
SENDING END REACTIVE POWER 16.0584
PERCENTAGE VOLTAGE REGULATION 9.30284
REAL POWER LOSS 1.32135
REACTIVE POWER LOSS -2.53394
EFFICIENCY 95.7813
```

1. A 3 phase 50 Hz,240KV line is 200m long. The line parameters are R=0.017Ω/ph/km; L=0.94mH/ph/km; C=0.0111µ F/ph/km.Calculate line performance when load is 500MW, 0.9 pf lag at 220KV.

Manual Calculation:

LONG TRANSMISSION LINE (ABCD CONSTANTS)

PROGRAM:

```
clc;
clear all;
R=input('Resistance                          :');
L=input('Inductance                          :');
C=input('Capacitance                         :');
G=input('Conductance                         :');
length=input('Length of Transmission Line         :');
f=input('frequency');
z= R+j*(2*pi*f*L*0.001);
y= G+j*(2*pi*f*C*0.000001);
gm=sqrt(z*y);
zc=sqrt(z/y);
m=menu('ENTER THE TYPE OF NETWORK','Equivalent PI', 'Equivalent T');
switch m
   case {1}
      Z1 = z*sinh(gm*length)/(gm*length);
      Y1 = y*tanh(gm*length/2)/(gm*length/2);
   otherwise
      Y1= y*sinh(gm*length)/(gm*length);
      Z1 = z*tanh(gm*length/2)/(gm*length/2);
end
A = 1+(Z1*Y1/2);
B=Z1;
C=Y1*(1+(Z1*Y1/4));
D=A;
TM = [ A B; C D ]; Z=B;
Y=(2*tanh(gm*length/2))/zc;
VRL=input('ENTER RECEIVEING END VOLTAGE              :');
VRP=VRL/(sqrt(3));
PR = input('ENTER RECEIVING END LOAD IN MW             :');
Pf=input('ENTER THE RECEIVING END LOAD POWER FACTOR   :');
h=acos(Pf);
SR=PR/Pf;
SR=SR*(cos(h)+j*sin(h));
QR=imag(SR);
```

```
IR=conj(SR)/(3*conj(VRP));
SM=TM*[VRP;IR];
VS=SM(1,1);
IS=SM(2,1);
Pfs=cos(angle(VS)-angle(IS));
SS=3*VS*conj(IS);
VSA=angle(VS)*(180/pi);
ISA=angle(IS)*(180/pi);
VS=sqrt(3)*abs(VS);
IS=abs(IS)*1000;
VREG=((VS/(abs(TM(1,1)))-VRL)/VRL)*100;
PS=real(SS);
QS=imag(SS);
eff=PR/PS*100;
PL=PS-PR;
QL=QS-QR;
z
y
zc
Z
Y
TM
fprintf('SENDING END VOLTAGE %g at %g degrees \n',VS,VSA);
fprintf('SENDING END CURRENT %g at %g degrees \n',IS,ISA);
fprintf('SENDING END POWER FACTOR %g\n',Pfs);
fprintf('SENDING END REAL POWER %g\n',PS);
fprintf('SENDING END REACTIVE POWER %g\n',QS);
fprintf('PERCENTAGE VOLTAGE REGULATION  %g\n',VREG);
fprintf('REAL POWER LOSS %g\n',PL);
fprintf('REACTIVE POWER LOSS %g\n',QL);
fprintf('EFFICIENCY %G', eff);
```

OUTPUT:

EQUIVALENT PI METHOD

```
Resistance                          : 0.0170
Inductance                          : 0.94
Capacitance                         : 0.0111
Conductance                         : 0
Length of Transmission Line         : 200
frequency50
ENTER RECEIVEING END VOLTAGE              : 220
ENTER RECEIVING END LOAD IN MW            : 500
ENTER THE RECEIVING END LOAD POWER FACTOR    : 0.9

z =
   0.0170 + 0.2953i
y =
        0 +3.4872e-006i
zc =
  2.9113e+002 -8.3727e+000i
Z =
   0.0168 + 0.2933i
Y =
  1.3896e-007 +6.9984e-004i
TM =

   1.0000 + 0.0000i   0.0168 + 0.2933i
   0.0000 + 0.0000i   1.0000 + 0.0000i

SENDING END VOLTAGE 220.362 at 0.168518 degrees
SENDING END CURRENT 1457.76 at -25.8262 degrees
SENDING END POWER FACTOR 0.898834
SENDING END REAL POWER 500.107
SENDING END REACTIVE POWER 243.861
PERCENTAGE VOLTAGE REGULATION 0.164499
REAL POWER LOSS 0.106943
REACTIVE POWER LOSS 1.70041
EFFICIENCY 99.9786
```

EQUIVALENT T METHOD

```
Resistance                           : 0.0170
Inductance                           : 0.94
Capacitance                          : 0.0111
Conductance                          : 0
Length of Transmission Line          : 200
frequency50
ENTER RECEIVEING END VOLTAGE                 : 220
ENTER RECEIVING END LOAD IN MW               : 500
ENTER THE RECEIVING END LOAD POWER FACTOR    : 0.9

z =
   0.0170 + 0.2953i
y =
        0 +3.4872e-006i
zc =
  2.9113e+002 -8.3727e+000i
Z =
   0.0171 + 0.2963i
Y =
  1.3896e-007 +6.9984e-004i
TM =

   1.0000 + 0.0000i   0.0171 + 0.2963i
  -0.0000 + 0.0000i   1.0000 + 0.0000i

SENDING END VOLTAGE 220.366 at 0.170206 degrees
SENDING END CURRENT 1457.76 at -25.8264 degrees
SENDING END POWER FACTOR 0.89882
SENDING END REAL POWER 500.109
SENDING END REACTIVE POWER 243.883
PERCENTAGE VOLTAGE REGULATION 0.166386
REAL POWER LOSS 0.109075
REACTIVE POWER LOSS 1.72148
EFFICIENCY 99.9782
```

5. The following data refers to a 3 phase overhead transmission line. The voltage is 220KV. Total series impedance /ph=200∟30’. Total shunt admittance/ph= 0.0013∟90’Ʊ. Load delivers is 100MW at 0.8pf lag. Using rigorous method determine line performance.

Manual Calculation:

LONG TRANSMISSION LINE (RIGOROUS METHOD)

PROGRAM:

```
clc;
clear all;
R=input('Resistance                              :');
XL=input('Inductive Reactance                     :');
XC=input('Capacitive Reactance                    :');
G=input('Conductance                              :');
length=input('Length of Transmission Line          :');
f=input('Frequency                                :');
z= (R+j*XL); y=
(G+j*XC); gm=sqrt(z*y);
zc=sqrt(z/y);
A=cosh(gm*length);
B=zc*sinh(gm*length);
C=1/zc *sinh(gm*length);
D=A;
TM=[A B;C D]; Z=B;
Y=2/zc*tanh(gm*length/2);
VRL=input('ENTER RECEIVEING END VOLTAGE               :');
VRP=VRL/(sqrt(3));
PR = input('ENTER RECEIVING END LOAD IN MW            :');
Pf=input('ENTER THE RECEIVING END LOAD POWER FACTOR    :');
h=acos(Pf);
SR=PR/Pf;
SR=SR*(cos(h)+j*sin(h));
QR=imag(SR);
IR=conj(SR)/(3*conj(VRP));
SM=TM*[VRP;IR];
VS=SM(1,1);
IS=SM(2,1);
Pfs=cos(angle(VS)-angle(IS));
SS=3*VS*conj(IS);
VSA=angle(VS)*(180/pi);
ISA=angle(IS)*(180/pi);
VS=sqrt(3)*abs(VS);
IS=abs(IS)*1000;
VREG=((VS/(abs(TM(1,1)))-VRL)/VRL)*100;
PS=real(SS);
QS=imag(SS);
eff=PR/PS*100;
PL=PS-PR;
```

```
QL=QS-QR;
z
y
zc
Z
Y
TM
fprintf('SENDING END VOLTAGE %g at %g degrees \n',VS,VSA);
fprintf('SENDING END CURRENT %g at %g degrees \n',IS,ISA);
fprintf('SENDING END POWER FACTOR %g\n',Pfs);
fprintf('SENDING END REAL POWER %g\n',PS);
fprintf('SENDING END REACTIVE POWER %g\n',QS);
fprintf('PERCENTAGE VOLTAGE REGULATION  %g\n',VREG);
fprintf('REAL POWER LOSS %g\n',PL);
fprintf('REACTIVE POWER LOSS %g\n',QL);
fprintf('EFFICIENCY %G', eff);
```

OUTPUT:

```
Resistance                        : 35
Inductive Reactance                   : 197
Capacitive Reactance                  : 0.0013
Conductance                         : 0
Length of Transmission Line           : 1
Frequency                         : 50
ENTER RECEIVEING END VOLTAGE              : 220
ENTER RECEIVING END LOAD IN MW            : 100
ENTER THE RECEIVING END LOAD POWER FACTOR    : 0.8
z =
  3.5000e+001 +1.9700e+002i
y =
       0 + 0.0013i
zc =
  3.9080e+002 -3.4446e+001i
Z =
  3.2069e+001 +1.8895e+002i
Y =
   0.0000 + 0.0013i
TM =
  1.0e+002 *
   0.0087 + 0.0002i   0.3207 + 1.8895i
  -0.0000 + 0.0000i   0.0087 + 0.0002i
SENDING END VOLTAGE 282.874 at 16.3752 degrees
SENDING END CURRENT 232.732 at -2.03335 degrees
SENDING END POWER FACTOR 0.948829
SENDING END REAL POWER 108.192
SENDING END REACTIVE POWER 36.0088
PERCENTAGE VOLTAGE REGULATION 46.9731
REAL POWER LOSS 8.19239
REACTIVE POWER LOSS -38.9912
EFFICIENCY 92.4279
```

RESULT:

Thus the line modeling of different types of transmission lines was done.

EXP NO: 2 (a)

DATE:

FORMATION OF BUS ADMITTANCE AND IMPEDANCE MATRICES AND SOLUTION OF NETWORKS

AIM:

To develop a program to obtain Y_{bus} matrix for the given networks by the method of inspection.

FORMATION OF Y-BUS MATRIX

Generalized [Y-Bus] = $\begin{bmatrix} Y_{ii} & Y_{ij} \\ Y_{ji} & Y_{jj} \end{bmatrix}$

Each admittance Y_{ii} (i =1,2,……n) is called the self admittance or driving point admittance of bus I and equals the sum of all admittances terminating on the particular bus.

Each off-diagonal term Y_{ij} (i,j = 1,2…n; j≠i) is the transfer admittance between buses I and j, n=total number of buses. Further, $Y_{ij} = Y_{ji}$

SIMULATION

In this exercise matrix, Z-Bus for the system is developed by first forming the Y_{bus} and then inverting it to get the Z-Bus matrix. The generator and transformer impedances are taken into account.

Y_{bus} is a sparse matrix, Z-Bus is a full matrix, i.e., zero elements of Y_{bus} become non-zero values in the corresponding Z-Bus elements. The bus impedance matrix is most useful for short circuit studies.

ALGORITHM

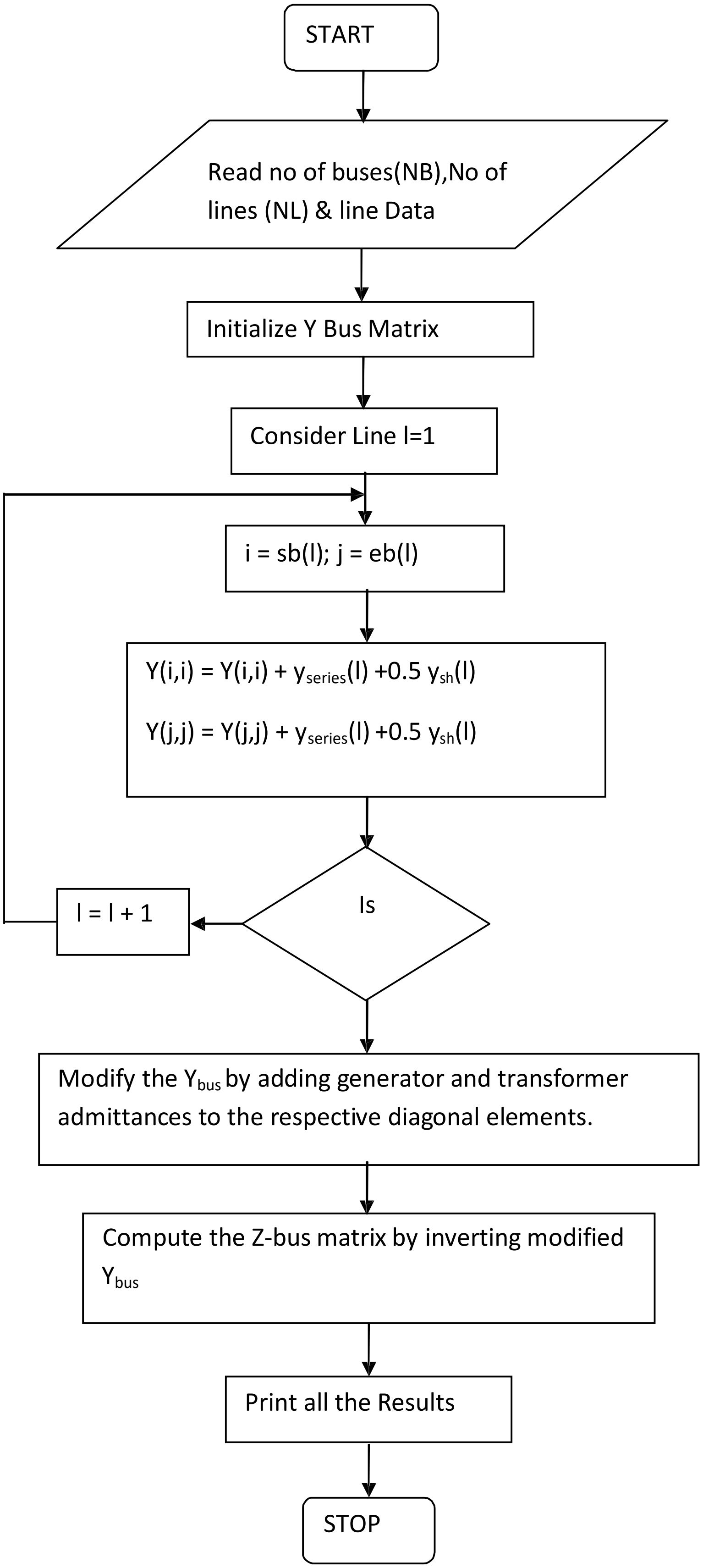
Step (1): Initialize [Y-Bus] matrix that is replace all entries by zero.

$Y_{ij} = Y_{ij}\text{-}y_{ij} = Y_{ji}$ = off diagonal element

Step (2): Compute $Y_{ii} = \sum_{j=1}^{n} y_{ij}$ = diagonal element.

Step (3) : Modify the Y_{bus} matrix by adding the transformer and the generator admittances to the respective diagonal elements of Y- bus matrix.

Step (4) : Compute the Z-Bus matrix by inverting the modified Y_{bus} matrix.


START
Read no of buses(NB),No of lines (NL) & line Data
Initialize Y Bus Matrix
Consider Line l=1
i = sb(l); j = eb(l)
$Y(i,i) = Y(i,i) + y_{series}(l) + 0.5\, y_{sh}(l)$
$Y(j,j) = Y(j,j) + y_{series}(l) + 0.5\, y_{sh}(l)$
Is
l = l + 1
Modify the Y_{bus} by adding generator and transformer admittances to the respective diagonal elements.
Compute the Z-bus matrix by inverting modified Y_{bus}
Print all the Results
STOP

1. The [Y-Bus] matrix is formed by inspection method for a four bus system. The line data and is given below.

LINE DATA

Line Number	SB	EB	Series Impedance (p.u)	Line charging Admittance (p.u)
1	1	2	0.10 + j0.40	j0.015
2	2	3	0.15 + j0.60	j0.020
3	3	4	0.18 + j0.55	j0.018
4	4	1	0.10 + j0.35	j0.012
5	4	2	0.25 + j0.20	j0.030

Manual Calculation:

FORMATION OF Y-BUS BY THE METHOD OF INSPECTION

PROGRAM:

```
clc;
clear all;
n=input('Enter number of buses');
l=input('Number of lines');
s=input('1.Impedance or 2:Admittance');
ybus=zeros(n,n);
lc=zeros(n,n);

for i=1:l
    a=input('Starting bus:');
    b=input('Ending bus:');
    t=input('Admittance or Impedance of line:');
    lca=input('Line charging admittance:');
    if(s==1)
        y(a,b)=1/t;
    else
        y(a,b)=t;
    end
    y(b,a)=y(a,b);
    lc(a,b)=lca;
    lc(b,a)=lc(a,b);
end

for i=1:n
    for j=1:n
        if i==j
            for k=1:n
                ybus(i,j)=ybus(i,j)+y(i,k)+lc(i,k)/2;
            end
            else
            ybus(i,j)=-y(i,j);
        end
        ybus(j,i)=ybus(i,j);
    end
end
ybus
```

OUTPUT:

```
Enter number of buses4
Number of lines5
1. Impedance or 2:Admittance1
Starting bus: 1
Ending bus: 2
Admittance or Impedance of line: 0.1+0.4i
Line charging admittance: 0.015i
Starting bus: 2
Ending bus: 3
Admittance or Impedance of line: 0.15+0.6i
Line charging admittance: 0.02i
Starting bus: 3
Ending bus: 4
Admittance or Impedance of line: 0.18+0.55i
Line charging admittance: 0.018i
Starting bus: 4
Ending bus: 1
Admittance or Impedance of line: 0.1+0.35i
Line charging admittance: 0.012i
Starting bus: 4
Ending bus: 2
Admittance or Impedance of line: 0.25+0.2i
Line charging admittance: 0.03i
```

```
ybus =


  1.3430 - 4.9810i      -0.5882 + 2.3529i       0                   -0.7547 + 2.6415i

 -0.5882 + 2.3529i       3.4194 - 5.8403i     -0.3922 + 1.5686i   -2.4390 + 1.9512i

       0                -0.3922 + 1.5686i      0.9296 - 3.1919i    -0.5375 + 1.6423i

 -0.7547 + 2.6415i     -2.4390 + 1.9512i      -0.5375 + 1.6423i    3.7312 - 6.2050i
```

RESULT:

Thus the program for the Y_{bus} formation by the method of inspection was executed and the output is verified with the manual calculation.

EXP NO: 2 (b)

DATE:

FORMATION OF BUS IMPEDANCE MATRICES

AIM:

To develop a program to obtain the Z bus matrix for the given network by the method of bus building algorithm.

FORMATION OF Z-BUS MATRIX

Z-Bus matrix is an important matrix used in different kinds of power system studies such as short circuit study, load flow study, etc.

In short circuit analysis, the generator and transformer impedances must be taken into account. In contingency analysis, the shunt elements are neglected while forming the Z-Bus matrix, which is used to compute the outage distribution factors.

This can be easily obtained by inverting the Y_{bus} formed by inspection method or by analytical method.Taking inverse of the Y_{bus} for large systems is time consuming; Moreover, modification in the system requires the whole process to be repeated to reflect the changes in the system. In such cases, the Z-Bus is computed by Z-Bus building algorithm.

ALGORITHM:

Step 1: Start the program.

Step 2: Read the number of buses, starting bus and ending bus.

Step 3: Initialize the Z Bus matrix.

Step 4: Form the Z – Bus matrix as follows

Case 1:

When a new bus of impedance Z_b is connected to reference bus $Z_{bus,new} = [Z_b]$

Case 2: Adding new bus p to existing bus q

$$Z_{bus,new} = \begin{bmatrix} Z_{orig} & & & Z_{1q} \\ & & & Z_{2q} \\ & & & .. \\ Z_{q1} & Z_{q2} & \ldots\ldots & Z_{qq} + Z_b \end{bmatrix}$$

Case 3: Adding impedance from an existing bus to reference bus.

$$Z_{jk,act} = Z_{jk} - \frac{Z_{j(n+1)} * Z_{(n+1)k}}{Z_{(n+1)(n+1)}}$$

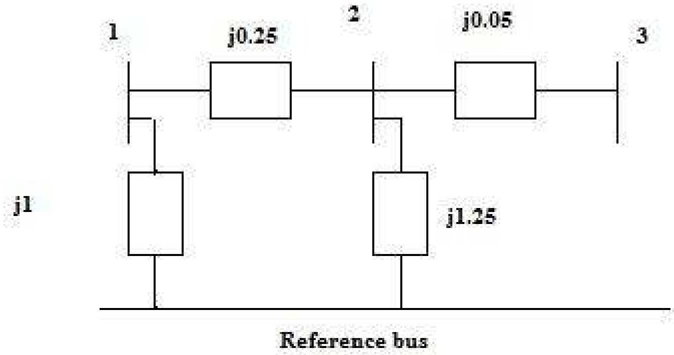
Case 4: Adding Z_b between two existing buses h and q

$$Z_{bus,new} = \begin{bmatrix} Z_{orig} & & Z_{1h} - Z_{1q} \\ & & Z_{2h} - Z_{2q} \\ Z_{h1} - Z_{q1} & Z_{h2} - Z_{q2} \ldots\ldots & Z_{(n+1)(n+1)} \end{bmatrix}$$

Case 5: Print the Z – bus matrix.

Problem:

Find the bus impedance matrix for the given network



FORMATION OF Z BUS BY BUS BUILDING ALGORITHM

PROGRAM

```
clc;
clear all;
nbus=input('Enter the total no. of buses excluding reference bus');
zbus=zeros(nbus,nbus);
t=1;
while t==1
        s=menu('Specify the case number','Connect a new bus to reference bus','connect an old bus to new bus','Connect two old buses','Connect an old bus to reference bus','Quit');
        switch(s);
            case{1}
                zb=input('Enter the Zbus value');
                zbus=zb;
            case{2}
                n=input('Enter the 2nd bus no.:');
                zb=input('Enter the reactance value:');
                for i=1:n
                    if i==n
                        zbus(n,n)=zbus(n-1,n-1)+zb;
                    else
                        zbus(i,n)=zbus(i,n-1);
                        zbus(n,i)=zbus(n-1,i);
                    end
                end
            case{3}
                l=input('Enter the existing bus number 1:');
                n=input('Enter the existing bus number 2:');
                zb=input('Enter the reactance value:');
                n=n+1;
                for i=1:n
                    if i==n
                        zbus(n,n)=zbus(l,l)+zb+zbus(n-1,n-1)-(2*zbus(1,n-1));
                        zbus(n,n);
                    else
                        zbus(i,n)=zbus(l,i)-zbus(i,n-1);
                        zbus(i,n);
                        zbus(n,i)=zbus(i,n);
                    end
                end
                for i=1:nbus
                    for j=1:nbus
                        if i==j
                            zbus(i,j)=zbus(i,j)-(zbus(i,n)*zbus(n,j)/zbus(n,n));
                            zbus(j,i)=zbus(i,j);
                            zbus(i,j)=zbus(j,i);
                        else
                            zbus(i,j)=zbus(i,j)-(zbus(i,n)*zbus(n,j)/zbus(n,n));
                        end
                    end
                end
                zbus(i,n)=0;
                zbus(n,i)=0;
        case{4}
            n=input('Enter the bus number:');
            zb=input('Enter reactance value:');
            n=n+1;
            for i=1:n
                if i==n
                    zbus(n,n)=zbus(n-1,n-1)+zb;
                else
                    zbus(i,n)=zbus(i,n-1);
                    zbus(n,i)=zbus(n-1,i);
                end
            end
            for i=1:n
                for j=1:n
                    if i==j
                        zbus(i,j)=zbus(i,j)-(zbus(i,n)*zbus(n,j)/zbus(n,n));
                        zbus(j,i)=zbus(i,j);
                        zbus(i,j)=zbus(j,i);
                    else
                        zbus(i,j)=zbus(i,j)-(zbus(i,n)*zbus(n,j)/zbus(n,n));
                    end
                end
            end
            case{5}
                disp('End the program');
                choice=menu('Would you like to print program or end program','print','end');
                if choice==1
                    zbus
                else
                    t=0;
                end
end
end
```

OUTPUT:

```
Enter the total no. of buses excluding reference bus3

Enter the Zbus value1i

Enter the 2nd bus no.:2

Enter the reactance value: .25i

Enter the bus number: 2

Enter reactance value: 1.25i

Enter the 2nd bus no.:3

Enter the reactance value: 0.05i

End the program


zbus =


   0 + 0.6000i     0 + 0.5000i     0 + 0.5000i

   0 + 0.5000i     0 + 0.6250i     0 + 0.6250i

   0 + 0.5000i     0 + 0.6250i     0 + 0.6750i


End the program
```

RESULT:

Thus the program for the Z bus formation by the method of inspection was executed and the output is verified with the manual calculation.

EXP NO: 3

DATE:

LOAD FLOW ANALYSIS BY GAUSS – SEIDAL METHOD

AIM:

To carryout load flow analysis of the given power system by Gauss – Seidal method.

ALGORITHM:

Step 1: Assume a flat voltage profile of 1+j0 for all buses except the slack bus. The voltage of slack bus is the specified voltage and it is not modified in any iteration.

Step 2: Assume a suitable value of ε called convergence criterion. Here ε is a specified change in bus voltage that is used to compare the actual change in bus voltage k^{th} and $(k+1)^{th}$ iteration.

Step 3: Set iteration count, k=0 and assumed voltage profile of the buses is denoted as $V_0^1, V_0^2, V_0^3, \ldots. V_0^n$ except slack bus.

Step 4: Set bus count, p=1

Step 5: Check for slack bus. If it is a slack bus then go to step-12,otherwise go to next step.

Step 6: Check for generator bus. If it is a generator bus go to next step, otherwise (i.e., if it is load bus) go to step=9.

Step 7: Temporarily set $\left|V_p^k\right| = \left|V_p\right|_{spec}$ and phase of ${}^{k}_{p}$ as the k^{th} iteration value if the bus-p is a generator bus where $\left|V_p\right|_{spec}$ is the specified magnitude of voltage for bus – p. Then calculate the reactive power of the generator bus using the following equation.

$$V_{p,temp}^{k+1} = \frac{1}{Y_{pp}}\left\{\frac{P_p - jQ_p}{\left(V_p^k\right)^*} - \sum_{q=1}^{p-1} Y_{pq}V_q^{k+1} - \sum_{q=p+1}^{n} Y_{pq}V_q^k\right\}$$

$$\delta_p^{k+1} = \tan^{-1}\left[\frac{\mathrm{Im}\left(V_{p,temp}^{k+1}\right)}{\mathrm{Re}al\left(V_{p,temp}^{k+1}\right)}\right]$$

The calculated reactive power may be within specified limits or it may violate the limits.

If the calculated reactive power is within the specified limits then consider this bus as generator bus and set $Q_p = Q_{p,cal}^{k+1}$ for this iteration and go to step-8.

If the calculated reactive power violates the specified limit for reactive power then treat this bus as a load bus.The magnitude of the reactive power at this bus will correspond to the limit it has violated.

i.e., if $Q_{p,cal}^{k+1} < Q_{p,\min}$ then $Q_p = Q_{p,\min}$

(or) $Q_{p,cal}^{k+1} > Q_{p,\min}$ then $Q_p = Q_{p,\max}$

Since the bus is treated as load bus, take actual value of V_p^k for $(k+1)^{th}$ iteration.i.e. V_p^k Need not be replaced by $|V_p|_{spec}$ when the generator bus is treated as load bus. Go to step9.

Step 8: For generator bus the magnitude of voltage does not change and so for all iteration the magnitude of bus voltage is the specified value. The phase of the bus voltage can be as shown below.

$$V_{p,temp}^{k+1} = \frac{1}{Y_{pp}} \left\{ \frac{P_p - jQ_p}{\left(V_p^k\right)^*} - \sum_{q=1}^{p-1} Y_{pq} V_q^{k+1} - \sum_{q=p+1}^{n} Y_{pq} V_q^k \right\}$$

$$\delta_p^{k+1} = \tan^{-1}\left[\frac{\mathrm{Im}\left(V_{p,temp}^{k+1}\right)}{\mathrm{Re}al\left(\quad_{p,temp}\right)} \right]$$

Now that $(k+1)^{th}$ iteration voltage of the generator bus is given by

$$V_p^{k+1} = |V_p|_{spec} \angle \delta_p^{k+1}$$

After calculating V_p^{k+1} for generator buses, go to step -11.

Step 9: For the load bus the $(k+1)^{th}$ iteration value of load bus –p voltage, V_p^{k+1} can be calculated using the following equation.

$$Y_{pp} \left\{ \quad \left(V_p^k\right)^* \quad q=1 \qquad q=p+1 \qquad \right|$$

Step 10: An acceleration factor, α can be used for faster convergence. If the acceleration factor is specified then modify the $(k+1)^{th}$ iteration value of bus-p voltage using the following equation.

$$V_{p,acc}^{k+1} = V_p^k + \alpha(V_p^{k+1} - V_p^k)$$

Then set , $V_p^{k+1} = V_{p,acc}^{k+1}$

Step 11: Calculate the change in the bus-p voltage, using the relation,

$$\Delta V_p^{k+1} = V_p^{k+1} - V_p^k$$

Step 12: Repeat the steps 5 to 11 until all the bus voltages have been calculated. For this increment the bus count by1 and go to step-5, until the bus count is n.

Step 13: Find out the largest of the absolute value of the change in voltage.

i.e., Find the largest among $\left|\Delta V_1^{k+1}\right|$,$\left|\Delta V_2^{k+1}\right|$, …..,$\left|\Delta V_n^{k+1}\right|$ Let this be the largest change $\left|\Delta V_{\max}\right|$.Check whether this largest change $\left|\Delta V_{\max}\right|$ is less than the prescribed tolerance ε .If $\left|\Delta V_{\max}\right|$ is less than ε then move to the next step. Otherwise increment the iteration count and go to step-4.

Step 14: Calculate the line flows and slack bus power using the bus voltages.

FLOW CHART

PROBLEM:

The system data for a load flow solution are given below. Determine the voltages by Gauss – Seidal method

Line admittances

Bus code	Admittance
1 – 2	2 – j8
1 – 3	1 – j4
2 – 3	0.666 – j2.664
2 – 4	1 – j4
3 – 4	2 – j8

Bus Specifications

Bus code	P	Q	V	Remarks
1	-	-	1.06	Slack
2	0.5	0.2	-	PQ
3	0.4	0.3	-	PQ
4	0.3	0.1	-	PQ

Manual Calculation:

PROGRAM

```
clc;
clear all;
n=input ('no of buses');
l=input ('no of lines');
s=input ('impedance 1 or admittance 2');
for i=1:l
    a=input ('starting bus');
    b=input ('ending bus');
    t=input ('admittance or impedance value');
    if s==1
        y(a,b)=1/t;
    else
        y(a,b)=t;
    end
    y(b,a)=y(a,b);
end
ybus=zeros(n,n);
for i=1:n
    for j=1:n
    if i==j
        for k=1:n
            ybus(i,j)=ybus(i,j)+y(i,k);
        end
    else
        ybus(i,j)=-y(i,j);
    end
    ybus(j,i)=ybus(i,j);
end
end

ybus
p=zeros(1,n);
q=zeros(1,n);
v=zeros(1,n);
pv=input ('no of pv buses');
pq=input('no of pq buses');
s=input ('slack bus number');
v(s)=input ('slack bus voltage');
acc=input ('accleration factor');
```

```
for i=1:pv
    b(i)=input('pv bus number');
    p(b(i))=input('real power');
    v(b(i))=input ('voltage value');
    qmin(b(i))=input ('min value of q');
    qmax(b(i))=input ('max value of q');
end
for i=1:pq
    c(i)=input('pq bus number');
    p(c(i))=input('real power');
    p(c(i))=-p(c(i));
    q(c(i))=input ('reactive power');
    q(c(i))=-q(c(i));
    v(c(i))=1+0i;
end
e=v;
e
enew(s)=v(s);
it=0;
yy=zeros(1,n);
for ii=1:n
    ypq(ii)=0;
    if ii~=s
        flag=0;
        gen=0;
        for j=1:pv
            if ii==b(j)
                flag=1;
            end
        end
        if flag==1
            for k=1:n
                yy(ii)=yy(ii)+ybus(ii,k)*v(k);
            end
            qcal(ii)=-imag(conj(v(ii))*yy(ii));
            if qcal(ii)<qmin(ii)
qcal(ii)=qmin(ii);
elseif qcal(ii)>qmax(ii)
    qcal(ii)=qmax(ii);
    else
        qcal(ii)=qcal(ii);
        gen=1;
        end
        else
            qcal(ii)=q(ii);
            end
```

```
qcal(ii)=qcal(ii)*sqrt(-1);
for k=1:n
    if k~=ii
        ypq(ii)=ypq(ii)+ybus(ii,k)*e(k);
    end
end
enew(ii)=(((p(ii)-qcal(ii))/conj(e(ii)))-ypq(ii))/ybus(ii,ii);
dele(ii)=enew(ii)-e(ii);
enew(ii)=e(ii)+acc*dele(ii);

if gen==1 ang=angle(enew(ii));
    enew(ii)=v(ii)*cos(ang)+v(ii)*sin(ang)*sqrt(-1);
end
   e(ii)=enew(ii);
   end
   end
   disp('voltages');
   enew
```

OUTPUT:

```
no of buses4
no of lines5
impedance 1 or admittance 22
starting bus1
ending bus2
admittance or impedance value2-8i
starting bus1
ending bus3
admittance or impedance value1-4i
starting bus2
ending bus3
admittance or impedance value0.666-2.664i
starting bus2
ending bus4
admittance or impedance value1-4i
starting bus3
ending bus4
admittance or impedance value2-8i

ybus =

   3.0000 -12.0000i  -2.0000 + 8.0000i  -1.0000 + 4.0000i        0
  -2.0000 + 8.0000i   3.6660 -14.6640i  -0.6660 + 2.6640i  -1.0000 + 4.0000i
  -1.0000 + 4.0000i  -0.6660 + 2.6640i   3.6660 -14.6640i  -2.0000 + 8.0000i
        0            -1.0000 + 4.0000i  -2.0000 + 8.0000i   3.0000 -12.0000i

no of pv buses0 no
of pq buses3 slack
bus number1
slack bus voltage1.06
accleration factor1
pq bus number2
real power0.5
reactive power0.2
pq bus number3
real power0.4
reactive power0.3
pq bus number4
real power0.3
```

```
reactive power0.1

e =

    1.0600    1.0000    1.0000    1.0000

voltages

enew =

   1.0600             1.0119 - 0.0289i   0.9929 - 0.0261i   0.9855 - 0.0486i
```

RESULT:

Thus load flow analysis by Gauss – Seidal method was done for the given power system.

EXP NO: 4

DATE:

LOAD FLOW ANALYSIS BY NEWTON - RAPHSON METHOD

AIM:

To carryout load flow analysis of the given power system by Newton raphson method.

ALGORITHM:

Step-1: Assume a flat voltage profile 1 + j0 for all buses (nodes) except the slack bus. The voltage of the slack bus is the specified voltage and it is not modified in any iteration.

Step-2: Assume a suitable value of ε called convergence criterion. Hence ε is a specified change in the residue that is used to compare the critical residues (Δ P and Δ Q or Δ V) at the end of each iteration.

Step-3: Set iteration count k = 0, and assumed voltage profile of the buses are denoted as V10, V20 …Vn0 except slack bus.

Step-4: Set bus count p = 1.

Step-5: Check for slack bus. If it is a slack bus then go to Step 13, otherwise go to next step.

Step-6: Calculate the real and reactive power of bus-p using the following equation.

$$P_i = \sum_{k=1}^{n} |V_i||V_k||Y_{ik}|\cos(\theta_{ik} - \delta_i + \delta_k)$$

$$Q_i = -\sum_{k=1}^{n} |V_i||V_k||Y_{ik}|\sin(\theta_{ik} - \delta_i + \delta_k)$$

Step-7: Calculate the change in real power, change in real power, $\Delta P_k = Pp_{,spec} - P_{pk}$; where $Pp_{,spec}$ = Specified real power for bus-p.

Step-8: Check for Generator bus. If it is a Generator bus go to next step, otherwise go to Step 12.

Step-9: Check for reactive power limit violation of Generator buses. For this compare the calculated reactive power Qpk with specified limits. If the limit is violated go to Step 11, otherwise go to next step.

Step-10: If the calculated reactive power is within the specified limits then consider this bus as Generator bus. Now calculate the voltage residue (change in voltage) using the following equation.

$|\Delta V_{pk}|^2 = |V_p|_{spec}^2 - |V_{pk}|^2$ where $|V_p|_{spec}$ = specified voltage.

Step-11: If the reactive power limit is violated then treat this bus as a load bus. Now the specified reactive power for this bus will correspond to the limit violated.

i.e., if $Q_{pk} < Q_p$, min then Qp, $_{spec}$ = Qp$_{, min}$

(Or) if $Q_{pk} > Q_p$, min then Qp, $_{spec}$ = Qp, $_{max}$

Step-12: Calculate the change in reactive power for load bus (or for the Generator bus treated as load bus). Change in reactive power, Δ Qp_k = |Qp, $_{spec}$| - Qp_k

Step-13: Repeat steps 5 to12 until all residues (change in P and Q or V) are calculated. For this increment the bus count by 1 and go to Step 5, until the bus count is n.

Step-14: Determine the largest of the absolute value of the residue (i.e., find the largest among Δ P_k,

Δ Q_k or |Δ Vp_k|^2. Let this largest change be Δ E.

Step-15: Compare Δ E and ε. If Δ E < ε then to Step 20, If Δ E > ε go to next step.

Step-16: Determine the elements of Jacobian matrix (**J**) by partially differentiating the load flow equations and evaluating the equation using Kth iteration values.

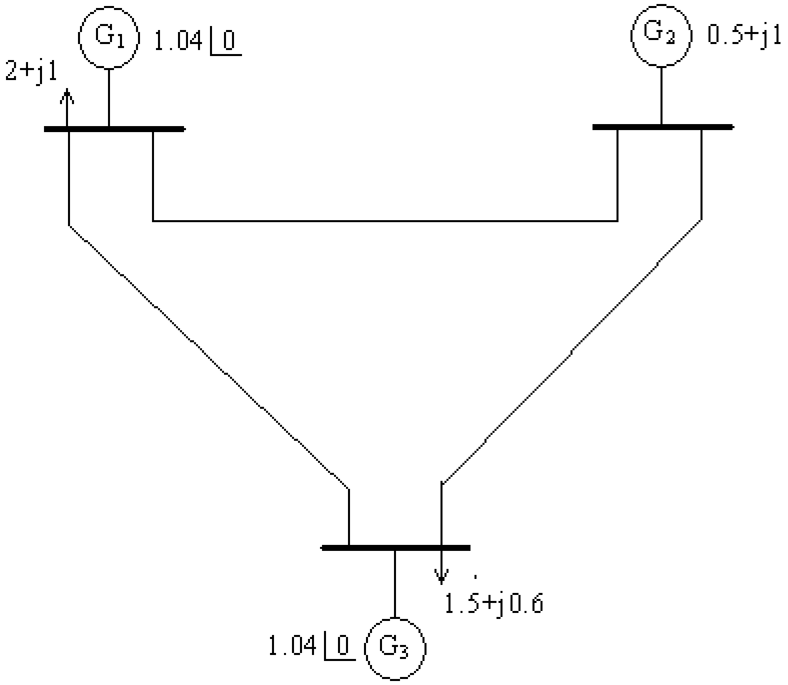
Step-17: Calculate the increments in real and reactive part of voltages.

Step-18: Calculate the new bus voltage.

Step-19: Advance the iteration count, i.e., k = k + 1 and go to Step 4.

Step-20: Calculate the line flows.

PROBLEM:



Consider the 3 bus system each of the 3 line bus a series impedance of 0.02 + j0.08 p.u and a total shunt admittance of j0.02 pu.The specified quantities at the buses are given below : Find the voltages in each bus for the given system using Newton-Raphson Method

Bus	Real load demand, P_D	Reactive Load demand, Q_D	Real power generation,P_G	Reactive Power Generation, Q_G	Voltage Specified
1	2	1	-	-	V_1=1.04
2	0	0	0.5	1	Unspecified
3	1.5	0.6	0	Q_{G3} = ?	V_3 = 1.04

Manual Calculation:

PROGRAM:

```
clc;
basemva=100;
accuracy=0.001;
accel=1.8;
maxiter=100;
busdata=[1  1   1.04   0    0      0    0    0    0   0    0
            2  0   1        0    0.5    1    0    0    0   0    0
            3  2   1.04   0    0      0    1.5 0.6 0    0    0   ];
linedata=[ 1    2   0.02    0.08    0.01    1
              1    3   0.02    0.08    0.01    1
              2    3   0.02    0.08    0.01    1];
lfybus
lfnewton
busout
lineflow
```

lfybus

```
j=sqrt(-1); i = sqrt(-1);
nl = linedata(:,1); nr = linedata(:,2); R = linedata(:,3);
X = linedata(:,4); Bc = j*linedata(:,5); a = linedata(:, 6);
nbr=length(linedata(:,1)); nbus = max(max(nl), max(nr));
Z = R + j*X; y= ones(nbr,1)./Z;        %branch admittance
for n = 1:nbr
if a(n) <= 0  a(n) = 1; else end
Ybus=zeros(nbus,nbus);     % initialize Ybus to zero
               % formation of the off diagonal elements
for k=1:nbr;
       Ybus(nl(k),nr(k))=Ybus(nl(k),nr(k))-y(k)/a(k);
       Ybus(nr(k),nl(k))=Ybus(nl(k),nr(k));
    end
end
              % formation of the diagonal elements
for  n=1:nbus
     for k=1:nbr
         if nl(k)==n
         Ybus(n,n) = Ybus(n,n)+y(k)/(a(k)^2) + Bc(k);
         elseif nr(k)==n
         Ybus(n,n) = Ybus(n,n)+y(k) +Bc(k);
         else, end
     end
end
clear Pgg
```

lfnewton

```
ns=0; ng=0; Vm=0; delta=0; yload=0; deltad=0;
nbus = length(busdata(:,1));
for k=1:nbus
n=busdata(k,1);
kb(n)=busdata(k,2); Vm(n)=busdata(k,3); delta(n)=busdata(k, 4);
Pd(n)=busdata(k,5); Qd(n)=busdata(k,6); Pg(n)=busdata(k,7); Qg(n) = busdata(k,8);
Qmin(n)=busdata(k, 9); Qmax(n)=busdata(k, 10);
Qsh(n)=busdata(k, 11);
    if Vm(n) <= 0  Vm(n) = 1.0; V(n) = 1 + j*0;
    else delta(n) = pi/180*delta(n);
         V(n) = Vm(n)*(cos(delta(n)) + j*sin(delta(n)));
         P(n)=(Pg(n)-Pd(n))/basemva;
         Q(n)=(Qg(n)-Qd(n)+ Qsh(n))/basemva;
         S(n) = P(n) + j*Q(n);
    end
end
for k=1:nbus
if kb(k) == 1, ns = ns+1; else, end
if kb(k) == 2 ng = ng+1; else, end
ngs(k) = ng;
nss(k) = ns;
end
Ym=abs(Ybus); t = angle(Ybus);
m=2*nbus-ng-2*ns;
maxerror = 1; converge=1;
iter = 0;
% Start of iterations
clear A  DC   J  DX
while maxerror >= accuracy & iter <= maxiter % Test for max. power mismatch
for i=1:m
for k=1:m
   A(i,k)=0;      %Initializing Jacobian matrix
end, end
iter = iter+1;
for n=1:nbus
nn=n-nss(n);
lm=nbus+n-ngs(n)-nss(n)-ns;
J11=0; J22=0; J33=0; J44=0;
   for i=1:nbr
     if nl(i) == n | nr(i) == n
        if nl(i) == n,  l = nr(i); end
        if nr(i) == n,  l = nl(i); end
        J11=J11+ Vm(n)*Vm(l)*Ym(n,l)*sin(t(n,l)- delta(n) + delta(l));
        J33=J33+ Vm(n)*Vm(l)*Ym(n,l)*cos(t(n,l)- delta(n) + delta(l));
        if kb(n)~=1
        J22=J22+ Vm(l)*Ym(n,l)*cos(t(n,l)- delta(n) + delta(l));
        J44=J44+ Vm(l)*Ym(n,l)*sin(t(n,l)- delta(n) + delta(l));
        else, end
        if kb(n) ~= 1  & kb(l) ~=1
        lk = nbus+l-ngs(l)-nss(l)-ns;
        ll = l -nss(l);
      % off diagonalelements of J1
        A(nn, ll) =-Vm(n)*Vm(l)*Ym(n,l)*sin(t(n,l)- delta(n) + delta(l));
              if kb(l) == 0  % off diagonal elements of J2
              A(nn, lk) =Vm(n)*Ym(n,l)*cos(t(n,l)- delta(n) + delta(l));end
              if kb(n) == 0  % off diagonal elements of J3
              A(lm, ll) =-Vm(n)*Vm(l)*Ym(n,l)*cos(t(n,l)- delta(n)+delta(l)); end
              if kb(n) == 0 & kb(l) == 0  % off diagonal elements of  J4
              A(lm, lk) =-Vm(n)*Ym(n,l)*sin(t(n,l)- delta(n) + delta(l));end
        else end
     else , end
   end
   Pk = Vm(n)^2*Ym(n,n)*cos(t(n,n))+J33;
   Qk = -Vm(n)^2*Ym(n,n)*sin(t(n,n))-J11;
   if kb(n) == 1 P(n)=Pk; Q(n) = Qk; end   % Swing bus P
     if kb(n) == 2  Q(n)=Qk;
         if Qmax(n) ~= 0
           Qgc = Q(n)*basemva + Qd(n) - Qsh(n);
           if iter <= 7                  % Between the 2th & 6th iterations
              if iter > 2                % the Mvar of generator buses are
                if Qgc  < Qmin(n),       % tested. If not within limits Vm(n)
                Vm(n) = Vm(n) + 0.01;    % is changed in steps of 0.01 pu to
                elseif Qgc  > Qmax(n),   % bring the generator Mvar within
                Vm(n) = Vm(n) - 0.01;end % the specified limits.
              else, end
           else,end
         else,end
     end
   if kb(n) ~= 1
     A(nn,nn) = J11;  %diagonal elements of J1
     DC(nn) = P(n)-Pk;
   end
   if kb(n) == 0
     A(nn,lm) = 2*Vm(n)*Ym(n,n)*cos(t(n,n))+J22; %diagonal elements of J2
     A(lm,nn)= J33;        %diagonal elements of J3
     A(lm,lm) =-2*Vm(n)*Ym(n,n)*sin(t(n,n))-J44; %diagonal of elements of J4
     DC(lm) = Q(n)-Qk;
   end
end
DX=A\DC';
for n=1:nbus
  nn=n-nss(n);
  lm=nbus+n-ngs(n)-nss(n)-ns;
    if kb(n) ~= 1
    delta(n) = delta(n)+DX(nn); end
    if kb(n) == 0
    Vm(n)=Vm(n)+DX(lm); end
 end
  maxerror=max(abs(DC));
     if iter == maxiter & maxerror > accuracy
   fprintf('\nWARNING: Iterative solution did not converged after ')
   fprintf('%g', iter), fprintf(' iterations.\n\n')
   fprintf('Press Enter to terminate the iterations and print the results \n')
   converge = 0; pause, else, end

end

if converge ~= 1
   tech= ('                      ITERATIVE SOLUTION DID NOT CONVERGE'); else,
   tech=('                   Power Flow Solution by Newton-Raphson Method');
```

```
end
V = Vm.*cos(delta)+j*Vm.*sin(delta);
deltad=180/pi*delta;
i=sqrt(-1);
k=0;
for n = 1:nbus
     if kb(n) == 1
     k=k+1;
     S(n)= P(n)+j*Q(n);
     Pg(n)  = P(n)*basemva + Pd(n);
     Qg(n)  = Q(n)*basemva + Qd(n) - Qsh(n);
     Pgg(k)=Pg(n);
     Qgg(k)=Qg(n);
     elseif kb(n) ==2
     k=k+1;
     S(n)=P(n)+j*Q(n);
     Qg(n)  = Q(n)*basemva + Qd(n) - Qsh(n);
     Pgg(k)=Pg(n);
     Qgg(k)=Qg(n);
  end
yload(n) = (Pd(n)- j*Qd(n)+j*Qsh(n))/(basemva*Vm(n)^2);
end
busdata(:,3)=Vm'; busdata(:,4)=deltad';
Pgt = sum(Pg);  Qgt = sum(Qg); Pdt = sum(Pd); Qdt = sum(Qd); Qsht = sum(Qsh);
```

busout

```
disp(tech)
fprintf('                      Maximum Power Mismatch = %g \n', maxerror)
fprintf('                             No. of Iterations = %g \n\n', iter)
head =['    Bus  Voltage  Angle    ------Load------    ---Generation---
Injected'
       '    No.  Mag.     Degree     MW       Mvar       MW       Mvar       Mvar
'
       '
'];
disp(head)
for n=1:nbus
     fprintf(' %5g', n), fprintf(' %7.3f', Vm(n)),
     fprintf(' %8.3f', deltad(n)), fprintf(' %9.3f', Pd(n)),
     fprintf(' %9.3f', Qd(n)),  fprintf(' %9.3f', Pg(n)),
     fprintf(' %9.3f ', Qg(n)), fprintf(' %8.3f\n', Qsh(n))
end
    fprintf('      \n'), fprintf('    Total              ')
    fprintf(' %9.3f', Pdt), fprintf(' %9.3f', Qdt),
    fprintf(' %9.3f', Pgt), fprintf(' %9.3f', Qgt), fprintf(' %9.3f\n\n', Qsht)
```

lineflow

```
SLT = 0;
fprintf('\n')
fprintf('                            Line Flow and Losses \n\n')
fprintf('     --Line--  Power at bus & line flow    --Line loss--  Transformer\n')
fprintf('     from  to    MW      Mvar     MVA       MW      Mvar      tap\n')

for n = 1:nbus
busprt = 0;
   for L = 1:nbr;
       if busprt == 0
       fprintf('   \n'), fprintf('%6g', n), fprintf('      %9.3f', P(n)*basemva)
       fprintf('%9.3f', Q(n)*basemva), fprintf('%9.3f\n', abs(S(n)*basemva))

       busprt = 1;
       else, end
       if nl(L)==n      k = nr(L);
       In = (V(n) - a(L)*V(k))*y(L)/a(L)^2 + Bc(L)/a(L)^2*V(n);
       Ik = (V(k) - V(n)/a(L))*y(L) + Bc(L)*V(k);
       Snk  =  V(n)*conj(In)*basemva;
       Skn  =  V(k)*conj(Ik)*basemva;
       SL   = Snk + Skn;
       SLT = SLT + SL;
       elseif nr(L)==n  k = nl(L);
       In = (V(n) - V(k)/a(L))*y(L) + Bc(L)*V(n);
       Ik = (V(k) - a(L)*V(n))*y(L)/a(L)^2 + Bc(L)/a(L)^2*V(k);
       Snk  = V(n)*conj(In)*basemva;
       Skn  = V(k)*conj(Ik)*basemva;
       SL   = Snk + Skn;
       SLT = SLT + SL;
       else, end
         if nl(L)==n | nr(L)==n
         fprintf('%12g', k),
         fprintf('%9.3f', real(Snk)), fprintf('%9.3f', imag(Snk))
         fprintf('%9.3f', abs(Snk)),
         fprintf('%9.3f', real(SL)),
             if nl(L) ==n & a(L) ~= 1
             fprintf('%9.3f', imag(SL)), fprintf('%9.3f\n', a(L))
             else, fprintf('%9.3f\n', imag(SL))
             end
         else, end
  end
end
SLT = SLT/2;
fprintf('   \n'), fprintf('    Total loss                         ')
fprintf('%9.3f', real(SLT)), fprintf('%9.3f\n', imag(SLT))
clear Ik In SL SLT Skn Snk
```

OUTPUT:

```
                Power Flow Solution by Newton-Raphson Method
                  Maximum Power Mismatch = 6.82452e-005
                        No. of Iterations = 3

 Bus  Voltage  Angle    ------Load------    ---Generation---   Injected
 No.  Mag.     Degree    MW       Mvar       MW       Mvar       Mvar

  1   1.040    0.000    0.000    0.000    -1.002    -2.436     0.000
  2   1.040    0.002    0.500    1.000     0.000     0.000     0.000
  3   1.040    0.038    0.000    0.000     1.500    -3.061     0.000

 Total                  0.500    1.000     0.498    -5.497     0.000


                          Line Flow and Losses

  --Line--  Power at bus & line flow    --Line loss--  Transformer
  from  to    MW      Mvar     MVA       MW      Mvar       tap

  1        -1.002   -2.436    2.634
       2   -0.167   -1.560    1.568    0.000   -2.164
       3   -0.833   -0.873    1.207    0.000   -2.163

  2        -0.500   -1.000    1.118
       1    0.167   -0.604    0.627    0.000   -2.164
       3   -0.667   -0.396    0.775    0.000   -2.163

  3         1.500   -3.061    3.409
       1    0.833   -1.290    1.535    0.000   -2.163
       2    0.667   -1.768    1.889    0.000   -2.163

 Total loss                            0.000   -6.490
```

RESULT:

Parameter	Calculated Value	Simulated Value
Bus voltage after the first iteration using Newton Raphson method		

Thus the load flow analysis of the given power system by Newton – Raphson method was performed for the given problem using Matlab-Power Tool Software.

EXP NO: 5

DATE:

S□MMETRI□ AN□ □NS□MMERTI□AL □A□LT ANAL□SIS

AIM:

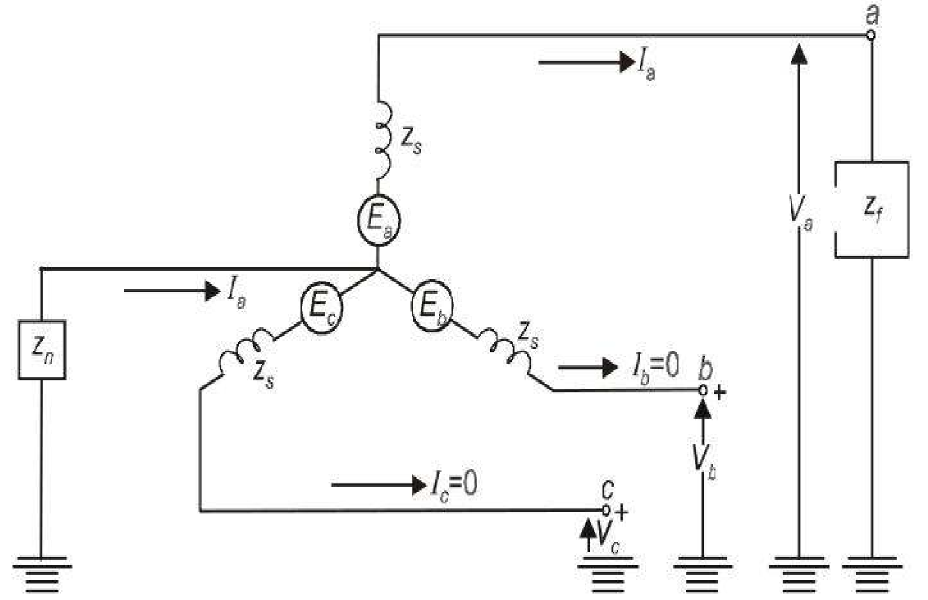
To become familiar with modeling and analysis of power systems under faulted condition and to compare the fault level, post-fault voltages and currents for different types of faults, both symmetric and unsymmetric.

OBJECTIVES

To conduct fault analysis on a given system using software available and obtain fault analysis report with fault level and current at the faulted point and post-fault voltages and currents in the network for the following faults.

1. Line-to-Ground
2. Line-to-Line
3. Double Line-to-Ground

SINGLE LINE-TO-GROUND FAULT

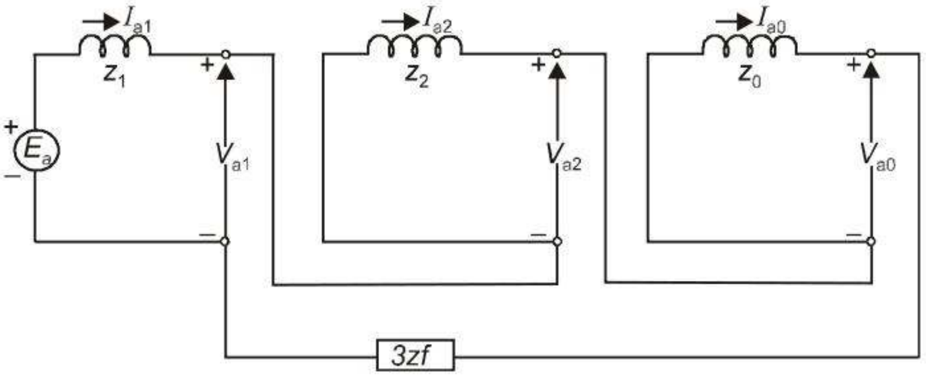


$I_b = 0$

$I_c = 0$

$V_a = Z_f\, I_a$

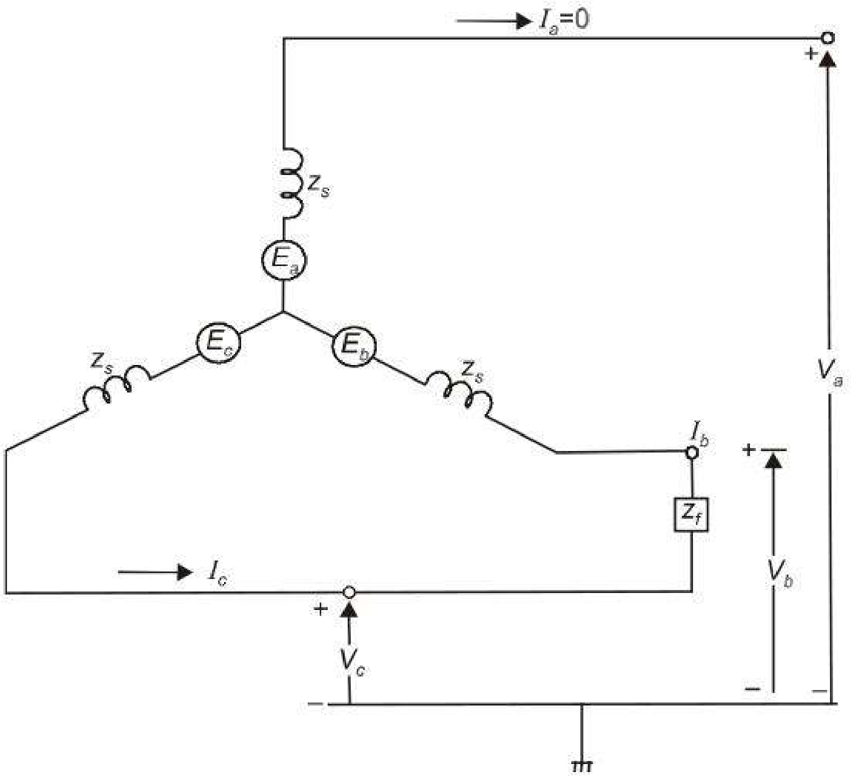
Sequence Network of Single line-to-ground-fault



$$I_a = 3I_{a1} = \frac{3E_a}{(Z_1 + Z_2 + Z_0) + 3Z_f}$$

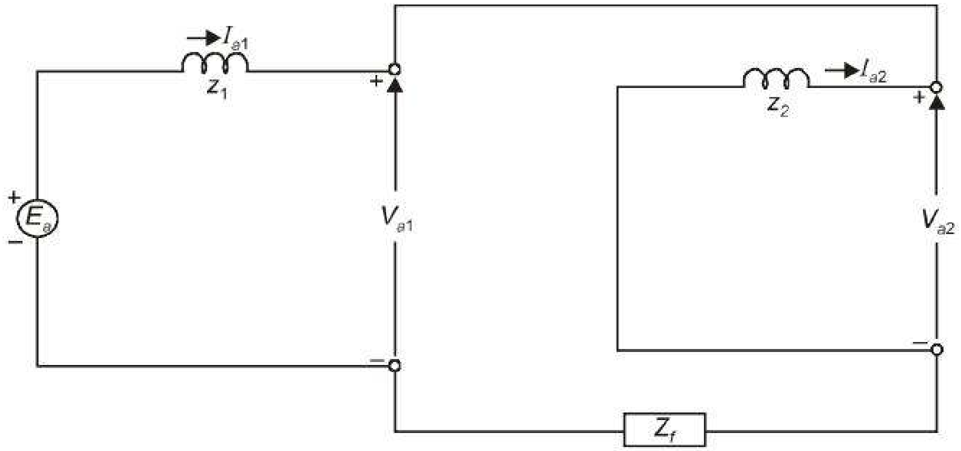
Fault Current

LINE-TO-LINE FAULT



$$V_b - V_c = Z_f \, .I_b$$
$$I_b + I_c = 0$$
$$I_a = 0$$

Sequence Network of Line-to-Line Fault



$$I_{a1} = \frac{E_a}{(Z_1 + Z_2 + Z_f)}$$

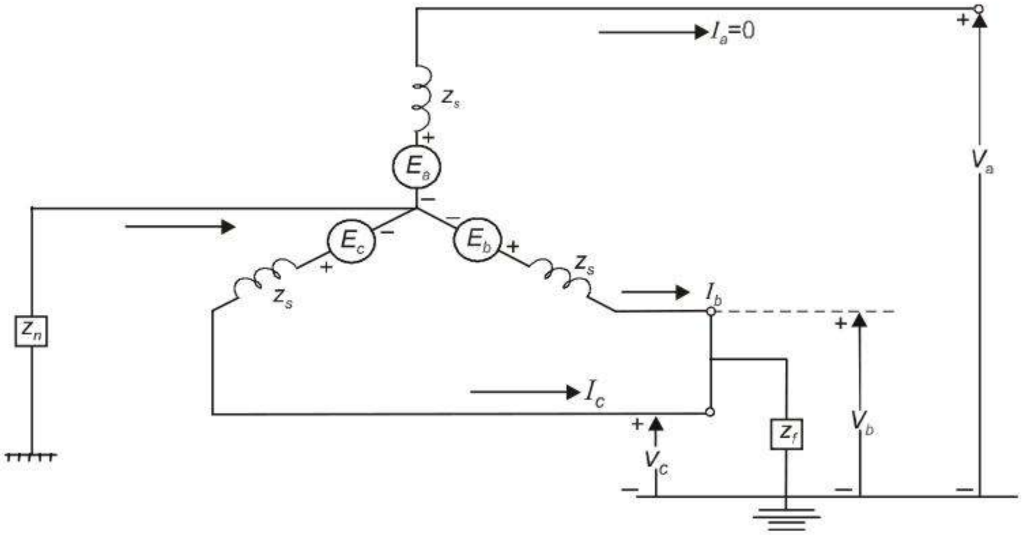
$$I_b = -I_c = \frac{-j\sqrt{3}\,E_a}{(Z_1 + Z_2 + Z_f)}$$

DOUBLE LINE-TO-GROUND FAULT

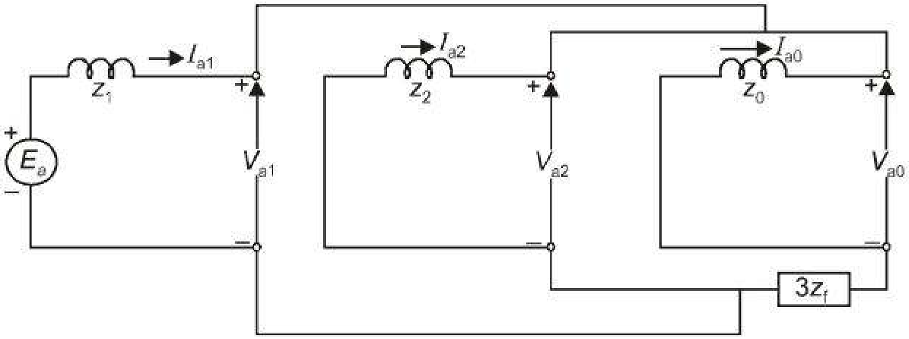
$$I_a = 0$$

$$I_{a1} + I_{a2} + I_{a0} = 0$$

$$V_b = V_c = (I_b + I_c)\ Z_f = 3Z_f\,I_{a0}$$



Sequence of Double line-to-ground fault



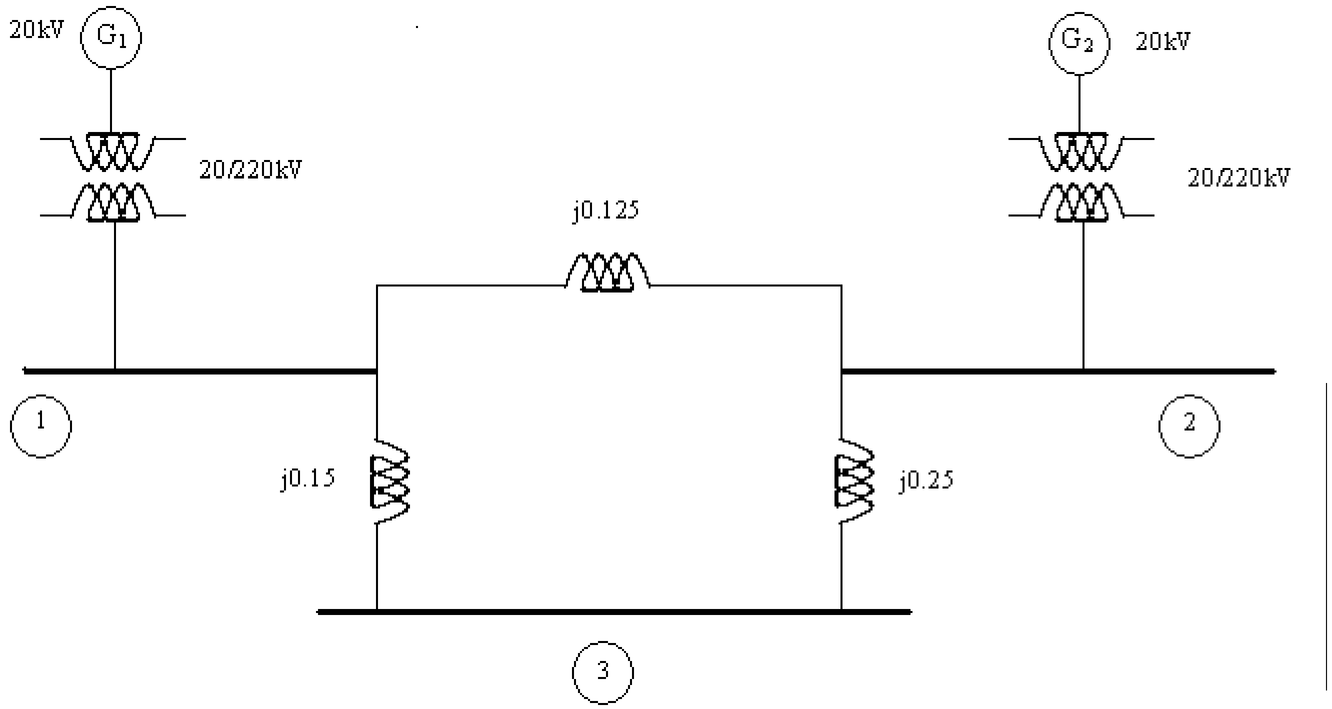
$$I_{a0} = \frac{-(E_a - Z_1 I_{a1})}{(Z_0 + 3Z_f)}$$

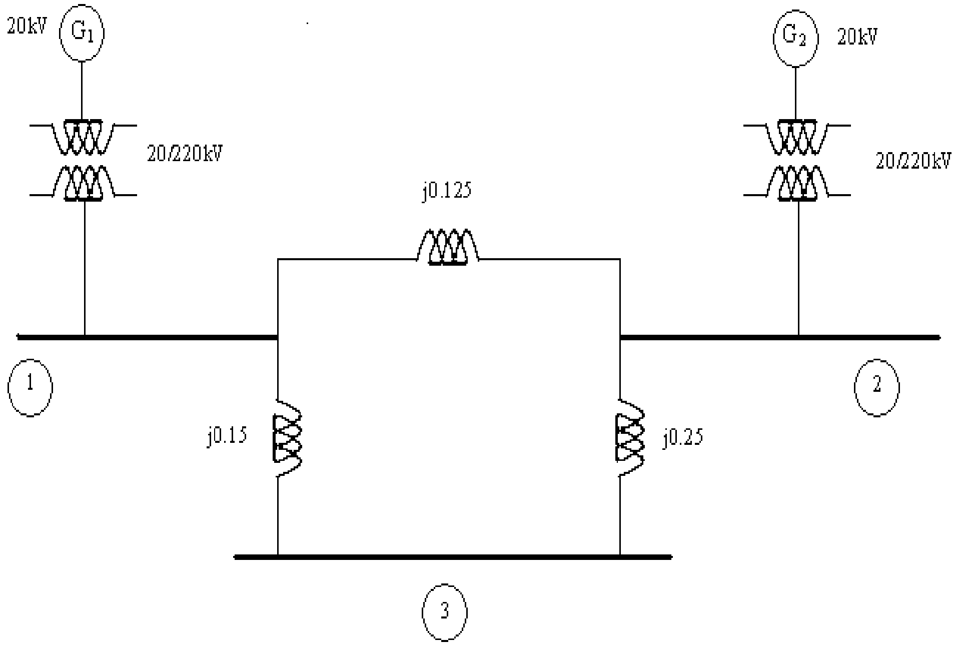
$$I_{a1} = \frac{E_a}{Z_1 + \dfrac{Z_2(Z_0 + 3Z_f)}{(Z_2 + Z_0 + 3Z_f)}}$$

$$I_{a2} = \frac{-(E_a - Z_1 I_{a1})}{Z_2}$$

PROBLEM:

The one line diagram of a simple power system is shown in the figure. The neutral of each generator is grounded through a current limiting reactor of 0.25/3 per unit on a 100MVA base. The system data expressed in per unit on a common 100MVA base is tabulated below. The generators are running on no load at their rated voltage and rated frequency with their emf's in phase.





PROGRAM:

```
zdata1 = [0 1  0  0.25
          0 2  0  0.25
          1 2  0  0.125
          1 3  0  0.15
          2 3  0  0.25];

zdata0 = [0 1  0  0.40
          0 2  0  0.10
          1 2  0  0.30
          1 3  0  0.35
          2 3  0  0.7125];
zdata2 = zdata1;
Zbus1  = zbuild(zdata1)
Zbus0  = zbuild(zdata0)
Zbus2  = Zbus1;
symfault(zdata1,Zbus1)
lgfault(zdata0, Zbus0, zdata1, Zbus1, zdata2, Zbus2)
llfault(zdata1, Zbus1, zdata2, Zbus2)
dlgfault(zdata0, Zbus0, zdata1, Zbus1, zdata2, Zbus2)
```

symfault

```
function symfaul(zdata, Zbus, V)

nl = zdata(:,1); nr = zdata(:,2); R = zdata(:,3);
X = zdata(:,4);
nc = length(zdata(1,:));
  if nc > 4
  BC = zdata(:,5);
  elseif nc ==4, BC = zeros(length(zdata(:,1)), 1);
  end
ZB = R + j*X;
nbr=length(zdata(:,1)); nbus = max(max(nl), max(nr));
if exist('V') == 1
    if length(V) == nbus
    V0 = V;
    else, end
else,  V0 = ones(nbus, 1) + j*zeros(nbus, 1);
end
fprintf('\Three-phase balanced fault analysis \n')
ff = 999;
while ff > 0
nf = input('Enter Faulted Bus No. -> ');
     while nf <= 0 | nf > nbus
     fprintf('Faulted bus No. must be between 1 & %g \n', nbus)
     nf = input('Enter Faulted Bus No. -> ');
     end
fprintf('\nEnter Fault Impedance Zf = R + j*X in ')
Zf = input('complex form (for bolted fault enter 0). Zf = ');
fprintf('  \n')
fprintf('Balanced three-phase fault at bus No. %g\n', nf)

If = V0(nf)/(Zf + Zbus(nf, nf));
Ifm = abs(If); Ifmang=angle(If)*180/pi;
fprintf('Total fault current = %8.4f per unit \n\n', Ifm)
%fprintf(' p.u. \n\n', Ifm)
fprintf('Bus Voltages during fault in per unit \n\n')
fprintf('     Bus     Voltage       Angle\n')
fprintf('     No.     Magnitude     degrees\n')

for n = 1:nbus
    if n==nf
    Vf(nf) = V0(nf)*Zf/(Zf + Zbus(nf,nf)); Vfm = abs(Vf(nf));
angv=angle(Vf(nf))*180/pi;
    else, Vf(n) = V0(n) - V0(n)*Zbus(n,nf)/(Zf + Zbus(nf,nf));
    Vfm = abs(Vf(n)); angv=angle(Vf(n))*180/pi;
    end
    fprintf('   %4g',  n), fprintf('%13.4f', Vfm),fprintf('%13.4f\n', angv)

end

fprintf('  \n')

fprintf('Line currents for fault at bus No.  %g\n\n', nf)
fprintf('     From      To     Current     Angle\n')
fprintf('     Bus       Bus    Magnitude   degrees\n')

for n= 1:nbus
   %Ign=0;
    for I = 1:nbr
      if nl(I) == n | nr(I) == n
         if nl(I) ==n       k = nr(I);
         elseif nr(I) == n  k = nl(I);
         end
           if k==0
            Ink = (V0(n) - Vf(n))/ZB(I);
            Inkm = abs(Ink); th=angle(Ink);
               %if th <= 0
                if real(Ink) > 0
                fprintf('      G   '), fprintf('%7g',n), fprintf('%12.4f', Inkm)
                fprintf('%12.4f\n', th*180/pi)
                elseif real(Ink) ==0 & imag(Ink) < 0
                fprintf('      G   '), fprintf('%7g',n), fprintf('%12.4f', Inkm)
                fprintf('%12.4f\n', th*180/pi)
               else, end
           Ign=Ink;
           elseif k ~= 0
            Ink = (Vf(n) - Vf(k))/ZB(I)+BC(I)*Vf(n);
            %Ink = (Vf(n) - Vf(k))/ZB(I);
            Inkm = abs(Ink); th=angle(Ink);
            %Ign=Ign+Ink;
            %if th <= 0
             if real(Ink) > 0
                  fprintf('%7g', n), fprintf('%10g', k),
                  fprintf('%12.4f', Inkm), fprintf('%12.4f\n', th*180/pi)
                  elseif real(Ink) ==0 & imag(Ink) < 0
                  fprintf('%7g', n), fprintf('%10g', k),
                  fprintf('%12.4f', Inkm), fprintf('%12.4f\n', th*180/pi)
                  else, end
              else, end
      else, end
   end

 if n==nf
 fprintf('%7g',n), fprintf('         F'),  fprintf('%12.4f', Ifm)
 fprintf('%12.4f\n', Ifmang)
 else, end
end
   resp=0;
   while strcmp(resp, 'n')~=1 & strcmp(resp, 'N')~=1 & strcmp(resp, 'y')~=1 &
strcmp(resp, 'Y')~=1
   resp = input('Another fault location? Enter ''y'' or ''n'' within single quote
-> ');
   if strcmp(resp, 'n')~=1 & strcmp(resp, 'N')~=1 & strcmp(resp, 'y')~=1 &
strcmp(resp, 'Y')~=1
   fprintf('\n Incorrect reply, try again \n\n'), end
   end
   if resp == 'y' | resp == 'Y'
   nf = 999;
   else ff = 0; end
end   % end for while
```

lgfault

```
function lgfault(zdata0, Zbus0, zdata1, Zbus1, zdata2, Zbus2, V)
if exist('zdata2') ~= 1
zdata2=zdata1;
else, end
if exist('Zbus2') ~= 1
Zbus2=Zbus1;
else, end
nl = zdata1(:,1); nr = zdata1(:,2); nl0 = zdata0(:,1);
nr0 = zdata0(:,2); nbr=length(zdata1(:,1)); nbus =
max(max(nl), max(nr)); nbr0=length(zdata0(:,1));
```

```
R0  = zdata0(:,3); X0  = zdata0(:,4);
R1  = zdata1(:,3); X1  = zdata1(:,4);
R2 = zdata1(:,3); X2 = zdata1(:,4);

for k=1:nbr0
    if R0(k)==inf | X0(k) ==inf
    R0(k) = 99999999;  X0(k) = 99999999;
    else, end
end
ZB1 = R1 + j*X1;  ZB0 = R0 + j*X0;
ZB2 = R2 + j*X2;

if exist('V') == 1
    if length(V) == nbus
    V0 = V;
    else, end
else,  V0 = ones(nbus, 1) + j*zeros(nbus, 1);
end
fprintf('\nLine-to-ground fault analysis \n')
ff = 999;
while ff > 0
nf = input('Enter Faulted Bus No. -> ');
while nf <= 0 | nf > nbus
     fprintf('Faulted bus No. must be between 1 & %g \n', nbus)
     nf = input('Enter Faulted Bus No. -> ');
     end
fprintf('\nEnter Fault Impedance Zf = R + j*X in ')
Zf = input('complex form (for bolted fault enter 0). Zf = ');
fprintf('  \n')
fprintf('Single line to-ground fault at bus No. %g\n', nf)
a =cos(2*pi/3)+j*sin(2*pi/3);
sctm = [1   1   1; 1 a^2  a; 1 a  a^2];
Ia0 = V0(nf)/(Zbus1(nf,nf)+Zbus2(nf, nf)+ Zbus0(nf, nf)+3*Zf); Ia1=Ia0; Ia2=Ia0;
I012=[Ia0; Ia1; Ia2];
Ifabc = sctm*I012;
Ifabcm = abs(Ifabc);
fprintf('Total fault current = %9.4f per unit\n\n', Ifabcm(1))
fprintf('Bus Voltages during the fault in per unit \n\n')
fprintf('     Bus    -------Voltage Magnitude-------  \n')
fprintf('     No.    Phase a     Phase b     Phase c  \n')

for n = 1:nbus
Vf0(n)= 0 - Zbus0(n, nf)*Ia0;
Vf1(n)= V0(n) - Zbus1(n, nf)*Ia1;
Vf2(n)= 0 - Zbus2(n, nf)*Ia2;
Vabc = sctm*[Vf0(n); Vf1(n); Vf2(n)];
Va(n)=Vabc(1); Vb(n)=Vabc(2); Vc(n)=Vabc(3);
fprintf(' %5g',n)
fprintf(' %11.4f', abs(Va(n))),fprintf(' %11.4f', abs(Vb(n)))
fprintf(' %11.4f\n', abs(Vc(n)))
end
fprintf('  \n')
fprintf('Line currents for fault at bus No.  %g\n\n', nf)
fprintf('     From      To      -----Line Current Magnitude----  \n')
fprintf('     Bus       Bus     Phase a     Phase b     Phase c  \n')
for n= 1:nbus
```

```
   for I = 1:nbr
      if nl(I) == n | nr(I) == n
         if nl(I) ==n       k = nr(I);
         elseif nr(I) == n  k = nl(I);
         end
            if k ~= 0
            Ink1(n, k) = (Vf1(n) - Vf1(k))/ZB1(I);
            Ink2(n, k) =  (Vf2(n) - Vf2(k))/ZB2(I);
            else, end
      else, end
   end
   for I = 1:nbr0
      if nl0(I) == n | nr0(I) == n
         if nl0(I) ==n       k = nr0(I);
         elseif nr0(I) == n  k = nl0(I);
         end
            if k ~= 0
            Ink0(n, k) =  (Vf0(n) - Vf0(k))/ZB0(I);
            else, end
      else, end
   end
   for I = 1:nbr
      if nl(I) == n | nr(I) == n
         if nl(I) ==n       k = nr(I);
         elseif nr(I) == n  k = nl(I);
         end
            if k ~= 0
            Inkabc = sctm*[Ink0(n, k); Ink1(n, k); Ink2(n, k)];
            Inkabcm = abs(Inkabc); th=angle(Inkabc);
                if real(Inkabc(1)) > 0
                fprintf('%7g', n), fprintf('%10g', k),
                fprintf(' %11.4f', abs(Inkabc(1))),fprintf(' %11.4f',
abs(Inkabc(2)))
                fprintf(' %11.4f\n', abs(Inkabc(3)))
                elseif real(Inkabc(1)) ==0 & imag(Inkabc(1)) < 0
                fprintf('%7g', n), fprintf('%10g', k),
                fprintf(' %11.4f', abs(Inkabc(1))),fprintf(' %11.4f',
abs(Inkabc(2)))
                fprintf(' %11.4f\n', abs(Inkabc(3)))
                else, end
             else, end
      else, end
   end
   if n==nf
   fprintf('%7g',n), fprintf('         F'),
   fprintf(' %11.4f', Ifabcm(1)),fprintf(' %11.4f', Ifabcm(2))
   fprintf(' %11.4f\n', Ifabcm(3))
   else, end
end
resp=0;
   while strcmp(resp, 'n')~=1 & strcmp(resp, 'N')~=1 & strcmp(resp, 'y')~=1 &
strcmp(resp, 'Y')~=1
   resp = input('Another fault location? Enter ''y'' or ''n'' within single quote
-> ');
   if strcmp(resp, 'n')~=1 & strcmp(resp, 'N')~=1 & strcmp(resp, 'y')~=1 &
strcmp(resp, 'Y')~=1
   fprintf('\n Incorrect reply, try again \n\n'), end
   end
   if resp == 'y' | resp == 'Y'
   nf = 999;
   else ff = 0; end
end   % end for while
```

llfault

```
function llfault(zdata1, Zbus1, zdata2, Zbus2, V)
if exist('zdata2') ~= 1
zdata2=zdata1;
else, end
if exist('Zbus2') ~= 1
Zbus2=Zbus1;
else, end

nl  = zdata1(:,1); nr  = zdata1(:,2);
R1  = zdata1(:,3); X1  = zdata1(:,4);
R2  = zdata2(:,3); X2  = zdata2(:,4);
ZB1 = R1 + j*X1;  ZB2 = R2 + j*X2;
nbr=length(zdata1(:,1)); nbus = max(max(nl), max(nr));
if exist('V') == 1
    if length(V) == nbus
    V0 = V;
    else, end
else,  V0 = ones(nbus, 1) + j*zeros(nbus, 1);
end
fprintf('\nLine-to-line fault analysis \n')
ff = 999;
while ff > 0
nf = input('Enter Faulted Bus No. -> ');
while nf <= 0 | nf > nbus
     fprintf('Faulted bus No. must be between 1 & %g \n', nbus)
     nf = input('Enter Faulted Bus No. -> ');
     end
fprintf('\nEnter Fault Impedance Zf = R + j*X in ')
Zf = input('complex form (for bolted fault enter 0). Zf = ');
fprintf('  \n')
fprintf('Line-to-line fault at bus No. %g\n', nf)
a =cos(2*pi/3)+j*sin(2*pi/3);
sctm = [1   1   1; 1 a^2  a; 1 a  a^2];
Ia0=0;
Ia1 = V0(nf)/(Zbus1(nf,nf)+Zbus2(nf, nf)+Zf); Ia2=-Ia1;
I012=[Ia0; Ia1; Ia2];
Ifabc = sctm*I012;
Ifabcm = abs(Ifabc);
fprintf('Total fault current = %9.4f per unit\n\n', Ifabcm(2))
fprintf('Bus Voltages during the fault in per unit \n\n')
fprintf('     Bus    -------Voltage Magnitude-------  \n')
fprintf('     No.    Phase a     Phase b     Phase c  \n')

for n = 1:nbus
Vf0(n)= 0;
Vf1(n)= V0(n) - Zbus1(n, nf)*Ia1;
Vf2(n)= 0 - Zbus2(n, nf)*Ia2;
```

```
Vabc = sctm*[Vf0(n); Vf1(n); Vf2(n)];
Va(n)=Vabc(1); Vb(n)=Vabc(2); Vc(n)=Vabc(3);
fprintf(' %5g',n)
fprintf(' %11.4f', abs(Va(n))),fprintf(' %11.4f', abs(Vb(n)))
fprintf(' %11.4f\n', abs(Vc(n)))
end
fprintf('  \n')
fprintf('Line currents for fault at bus No.  %g\n\n', nf)
fprintf('     From      To      -----Line Current Magnitude----  \n')
fprintf('     Bus       Bus     Phase a     Phase b     Phase c  \n')

for n= 1:nbus
   for I = 1:nbr
      if nl(I) == n | nr(I) == n
         if nl(I) ==n       k = nr(I);
         elseif nr(I) == n  k = nl(I);
         end
            if k ~= 0
            Ink0(n, k) = 0;
            Ink1(n, k) = (Vf1(n) - Vf1(k))/ZB1(I);
            Ink2(n, k) =  (Vf2(n) - Vf2(k))/ZB2(I);

            Inkabc = sctm*[Ink0(n, k); Ink1(n, k); Ink2(n, k)];
            Inkabcm = abs(Inkabc); th=angle(Inkabc);
                 if real(Inkabc(2)) < 0

               fprintf('%7g', n), fprintf('%10g', k),
                fprintf(' %11.4f', abs(Inkabc(1))),fprintf(' %11.4f',
abs(Inkabc(2)))
                fprintf(' %11.4f\n', abs(Inkabc(3)))
                elseif real(Inkabc(2)) ==0 & imag(Inkabc(2)) > 0
                fprintf('%7g', n), fprintf('%10g', k),
                fprintf(' %11.4f', abs(Inkabc(1))),fprintf(' %11.4f',
abs(Inkabc(2)))
                fprintf(' %11.4f\n', abs(Inkabc(3)))
                else, end
             else, end
      else, end
   end
   if n==nf
   fprintf('%7g',n), fprintf('        F'),
   fprintf(' %11.4f', Ifabcm(1)),fprintf(' %11.4f', Ifabcm(2))
   fprintf(' %11.4f\n', Ifabcm(3))
   else, end
end
resp=0;
   while strcmp(resp, 'n')~=1 & strcmp(resp, 'N')~=1 & strcmp(resp, 'y')~=1 &
strcmp(resp, 'Y')~=1
   resp = input('Another fault location? Enter ''y'' or ''n'' within single quote
-> ');
   if strcmp(resp, 'n')~=1 & strcmp(resp, 'N')~=1 & strcmp(resp, 'y')~=1 &
strcmp(resp, 'Y')~=1
   fprintf('\n Incorrect reply, try again \n\n'), end
   end
   if resp == 'y' | resp == 'Y'
   nf = 999;
   else ff = 0; end
end   % end for while
```

dlgfault

```
function dlgfault(zdata0, Zbus0, zdata1, Zbus1, zdata2, Zbus2, V)

if exist('zdata2') ~= 1
zdata2=zdata1;
else, end
if exist('Zbus2') ~= 1
Zbus2=Zbus1;
else, end

nl = zdata1(:,1); nr = zdata1(:,2); nl0 = zdata0(:,1);
nr0 = zdata0(:,2); nbr=length(zdata1(:,1)); nbus =
max(max(nl), max(nr)); nbr0=length(zdata0(:,1));
R0  = zdata0(:,3); X0  = zdata0(:,4);
R1  = zdata1(:,3); X1  = zdata1(:,4);
R2 = zdata2(:,3); X2 = zdata2(:,4);

for k = 1:nbr0
    if R0(k) == inf | X0(k) == inf
    R0(k) = 99999999; X0(k) = 999999999;
    else, end
end
ZB1 = R1 + j*X1;  ZB0 = R0 + j*X0;
ZB2 = R2 + j*X2;

if exist('V') == 1
    if length(V) == nbus
    V0 = V;
    else, end
else,  V0 = ones(nbus, 1) + j*zeros(nbus, 1);
end

fprintf('\nDouble line-to-ground fault analysis \n')
ff = 999;
while ff > 0
nf = input('Enter Faulted Bus No. -> ');
while nf <= 0 | nf > nbus
     fprintf('Faulted bus No. must be between 1 & %g \n', nbus)
     nf = input('Enter Faulted Bus No. -> ');
     end
fprintf('\nEnter Fault Impedance Zf = R + j*X in ')
Zf = input('complex form (for bolted fault enter 0). Zf = ');
fprintf('  \n')
fprintf('Double line-to-ground fault at bus No. %g\n', nf)
a =cos(2*pi/3)+j*sin(2*pi/3);
sctm = [1   1   1; 1 a^2  a; 1 a  a^2];

Z11 = Zbus2(nf, nf)*(Zbus0(nf, nf)+ 3*Zf)/(Zbus2(nf, nf)+Zbus0(nf, nf)+3*Zf);
Ia1 = V0(nf)/(Zbus1(nf,nf)+Z11);
Ia2 =-(V0(nf) - Zbus1(nf, nf)*Ia1)/Zbus2(nf,nf);
```

```
Ia0 =-(V0(nf) - Zbus1(nf, nf)*Ia1)/(Zbus0(nf,nf)+3*Zf);
I012=[Ia0; Ia1; Ia2];
Ifabc = sctm*I012; Ifabcm=abs(Ifabc);
Ift = Ifabc(2)+Ifabc(3);
Iftm = abs(Ift);

fprintf('Total fault current = %9.4f per unit\n\n', Iftm)
fprintf('Bus Voltages during the fault in per unit \n\n')
fprintf('     Bus    -------Voltage Magnitude-------  \n')
fprintf('     No.    Phase a     Phase b     Phase c  \n')

for n = 1:nbus
Vf0(n)= 0 - Zbus0(n, nf)*Ia0;
Vf1(n)= V0(n) - Zbus1(n, nf)*Ia1;
Vf2(n)= 0 - Zbus2(n, nf)*Ia2;
Vabc = sctm*[Vf0(n); Vf1(n); Vf2(n)];
Va(n)=Vabc(1); Vb(n)=Vabc(2); Vc(n)=Vabc(3);
fprintf(' %5g',n)
fprintf(' %11.4f', abs(Va(n))),fprintf(' %11.4f', abs(Vb(n)))
fprintf(' %11.4f\n', abs(Vc(n)))
end
fprintf('  \n')
fprintf('Line currents for fault at bus No.  %g\n\n', nf)
fprintf('     From      To      -----Line Current Magnitude----  \n')
fprintf('     Bus       Bus     Phase a     Phase b     Phase c  \n')
for n= 1:nbus
   for I = 1:nbr
      if nl(I) == n | nr(I) == n
         if nl(I) ==n       k = nr(I);
         elseif nr(I) == n  k = nl(I);
         end
            if k ~= 0
            Ink1(n, k) = (Vf1(n) - Vf1(k))/ZB1(I);
            Ink2(n, k) =  (Vf2(n) - Vf2(k))/ZB2(I);
            else, end
      else, end
   end
   for I = 1:nbr0
      if nl0(I) == n | nr0(I) == n
         if nl0(I) ==n       k = nr0(I);
         elseif nr0(I) == n  k = nl0(I);
         end
            if k ~= 0
            Ink0(n, k) =  (Vf0(n) - Vf0(k))/ZB0(I);
            else, end
      else, end
   end
   for I = 1:nbr
      if nl(I) == n | nr(I) == n
         if nl(I) ==n       k = nr(I);
         elseif nr(I) == n  k = nl(I);
         end
            if k ~= 0
            Inkabc = sctm*[Ink0(n, k); Ink1(n, k); Ink2(n, k)];
            Inkabcm = abs(Inkabc); th=angle(Inkabc);
                if real(Inkabc(2)) < 0
                fprintf('%7g', n), fprintf('%10g', k),
                fprintf(' %11.4f', abs(Inkabc(1))),fprintf(' %11.4f', 
abs(Inkabc(2)))
                fprintf(' %11.4f\n', abs(Inkabc(3)))
                elseif real(Inkabc(2)) ==0 & imag(Inkabc(2)) > 0
                fprintf('%7g', n), fprintf('%10g', k),
                fprintf(' %11.4f', abs(Inkabc(1))),fprintf(' %11.4f', 
abs(Inkabc(2)))
                fprintf(' %11.4f\n', abs(Inkabc(3)))
                else, end
             else, end
      else, end
   end
   if n==nf
   fprintf('%7g',n), fprintf('         F'),
   fprintf(' %11.4f', Ifabcm(1)),fprintf(' %11.4f', Ifabcm(2))
   fprintf(' %11.4f\n', Ifabcm(3))
   else, end
end
resp=0;
   while strcmp(resp, 'n')~=1 & strcmp(resp, 'N')~=1 & strcmp(resp, 'y')~=1 & 
strcmp(resp, 'Y')~=1
   resp = input('Another fault location? Enter ''y'' or ''n'' within single quote 
-> ');
   if strcmp(resp, 'n')~=1 & strcmp(resp, 'N')~=1 & strcmp(resp, 'y')~=1 & 
strcmp(resp, 'Y')~=1
   fprintf('\n Incorrect reply, try again \n\n'), end
   end
   if resp == 'y' | resp == 'Y'
   nf = 999;
   else ff = 0; end
end   % end for while
```

OUTPUT:

```
Enter Faulted Bus No. -> 3

Enter Fault Impedance Zf = R + j*X in complex form (for bolted fault enter 0). Zf = j*0.1

Balanced three-phase fault at bus No. 3

Total fault current =   3.1250 per unit

Bus Voltages during fault in per unit

   Bus     Voltage      Angle

   No.    Magnitude    degrees

    1      0.5938      0.0000

    2      0.6250      0.0000
   3     0.3125     0.0000

Line currents for fault at bus No.  3

   From     To     Current     Angle

   Bus      Bus    Magnitude   degrees

   G        1      1.6250    -90.0000

   1        3      1.8750    -90.0000

   G        2      1.5000    -90.0000

   2        1      0.2500    -90.0000

   2        3      1.2500    -90.0000

   3        F      3.1250    -90.0000

Another fault location? Enter 'y' or 'n' within single quote -> 'n'

Line-to-ground fault analysis

Enter Faulted Bus No. -> 3

Enter Fault Impedance Zf = R + j*X in complex form (for bolted fault enter 0). Zf = j*0.1

Single line to-ground fault at bus No. 3

Total fault current =    2.7523 per unit

Bus Voltages during the fault in per unit

   Bus     -------Voltage Magnitude-------

   No.   Phase a    Phase b    Phase c

   1     0.6330     1.0046     1.0046

   2     0.7202     0.9757     0.9757

   3     0.2752     1.0647     1.0647

Line currents for fault at bus No.  3

   From     To     -----Line Current Magnitude----

   Bus      Bus    Phase a    Phase b    Phase c
```

```
     1      3     1.6514     0.0000     0.0000

     2      1     0.3761     0.1560     0.1560

     2      3     1.1009     0.0000     0.0000

     3      F     2.7523     0.0000     0.0000

Another fault location? Enter 'y' or 'n' within single quote -> 'n'

Line-to-line fault analysis

Enter Faulted Bus No. -> 3

Enter Fault Impedance Zf = R + j*X in complex form (for bolted fault enter 0). Zf = j*0.1

Line-to-line fault at bus No. 3

Total fault current =    3.2075 per unit

Bus Voltages during the fault in per unit

   Bus     -------Voltage Magnitude-------

   No.   Phase a    Phase b    Phase c

   1     1.0000     0.6720     0.6720

   2     1.0000     0.6939     0.6939

   3     1.0000     0.5251     0.5251

Line currents for fault at bus No.  3

   From     To     -----Line Current Magnitude----

   Bus      Bus    Phase a    Phase b    Phase c

     1      3     0.0000     1.9245     1.9245

     2      1     0.0000     0.2566     0.2566

     2      3     0.0000     1.2830     1.2830

     3      F     0.0000     3.2075     3.2075

Another fault location? Enter 'y' or 'n' within single quote -> 'n'

Double line-to-ground fault analysis
```

```
Enter Faulted Bus No. -> 3

Enter Fault Impedance Zf = R + j*X in complex form (for bolted fault enter 0). Zf = j*0.1

Double line-to-ground fault at bus No. 3

Total fault current =    1.9737 per unit

Bus Voltages during the fault in per unit

     Bus      -------Voltage Magnitude-------

     No.    Phase a     Phase b     Phase c

     1      1.0066      0.5088      0.5088

     2      0.9638      0.5740      0.5740

     3      1.0855      0.1974      0.1974

Line currents for fault at bus No.  3

     From      To      -----Line Current Magnitude----

     Bus       Bus     Phase a     Phase b     Phase c

      1         3      0.0000      2.4350      2.4350

      2         1      0.1118      0.3682      0.3682

      2         3      0.0000      1.6233      1.6233

      3         F      0.0000      4.0583      4.0583

Another fault location? Enter 'y' or 'n' within single quote -> 'n'
```

RESULT:

Quantity	Calculated Value	Simulated Value
FAULT CURRENT FOR 1. THREE PHASE FAULT 2. L-G FAULT 3. L-L FAULT 4. DOUBLE LINE FAULT		

Thus the modeling and analysis of power system under faulted condition was made familiar and the fault level, post fault voltage and currents for different types of fault both symmetric and unsymmetrical was computed.

EXP NO: 6

DATE:

TRANSIENT AND SMALL SIGNAL STABILITY ANALYSIS – SINGLE MACHINE INFINITE BUS SYSTEM

AIM:

To become familiar with various aspects of the transient and small signal stability analysis of Single-Machine-Infinite Bus (SMIB) system.

OBJECTIVES

- To understand modeling and analysis of transient and small signal stability of a SMIB power system.
- To examine the transient stability of a SMIB and determine the critical clearing time of the system, through stimulation by trial and error method and by direct method.
- To assess the transient stability of a multi- machine power system when subjected to a common disturbance sequence: fault application on a transmission line followed by fault removal and line opening.
- To determine the critical clearing time.

THEORY :

Stability : Stability problem is concerned with the behaviour of power system when it is subjected to disturbance and is classified into small signal stability problem if the disturbances are small and transient stability problem when the disturbances are large.

Transient stability: When a power system is under steady state, the load plus transmission loss equals to the generation in the system. The generating units run a synchronous speed and system frequency, voltage, current and power flows are steady. When a large disturbance such as three phase fault, loss of load, loss of generation etc., occurs the power balance is upset and the generating units rotors experience either acceleration or deceleration. The system may come back to a steady state condition maintaining synchronism or it may break into subsystems or one or more machines may pull out of synchronism. In the former case the system is said to be stable and in the later case it is said to be unstable.

Small signal stability: When a power system is under steady state, normal operating condition, the system may be subjected to small disturbances such as variation in load and generation, change in field voltage, change in mechanical toque etc., The nature of system response to small disturbance depends on the operating conditions, the transmission system strength, types of controllers etc. Instability that may result from small disturbance may be of two forms,

(i) Steady increase in rotor angle due to lack of synchronising torque.

(ii) Rotor oscillations of increasing magnitude due to lack of sufficient damping torque.

FORMULA :

Reactive power $Q_e = \sin(\cos^{-1}(p.f))$

$$\text{Stator Current } I_t = \frac{S^*}{E_t^*}$$

$$= \frac{P_e - jQ_e}{E_t^*}$$

Voltage behind transient condition

$$E^1 = E_t + j X_d{}^1 I_t$$

Voltage of infinite bus

$$E_B = E_t - j(X_3 + X_{tr})I_t$$

$$\text{where,} \quad X_3 = \frac{X_1 X_2}{X_1 + X_2}$$

Angular separation between E^1 and E_B

$$\delta o = \angle E^1 - \angle E_B$$

Prefault Operation:

$$X = j\,X_d^1 + jX_{tr} + \frac{X_1\,X_2}{X_1 + X_2}$$

$$\text{Power } P_e = \frac{E^1 \text{ x } E_B}{X} \sin\delta_o$$

$$\delta_o = \sin^{-1} \frac{\left(P_e * \; X\right)}{E^1 * E_B}$$

During Fault Condition:

$P_e = P_{Eii} = 0$

Find out X from the equivalent circuit during fault condition

Post fault Condition:

Find out X from the equivalent circuit during post fault condition

$$\text{Power } P_e = \left\{\frac{E^1 \text{ x } E_B}{X}\right\} \sin\delta_o$$

$$\delta_{max} = \pi - \delta_o$$

$$P_e = \frac{P_m}{\mathbf{sin}\boldsymbol{\delta}_{\mathbf{max}}}$$

Critical Clearing Angle:

$$\cos\delta_{cr} = \frac{P_m(\delta_{max} - \delta_o) + P_{3max}\cos\delta_{max} - P_{2max}\cos\delta_o}{P_{3max} - P_{2max}}$$

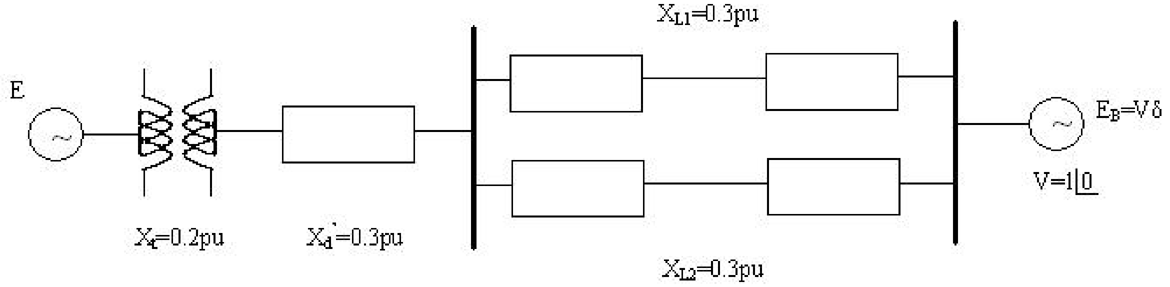
Critical Clearing Time:

$$t_{cr} = \sqrt{\frac{2H(\delta_{cr} - \delta_o)}{\pi f_o P_m}} \quad \text{Secs}$$

PROBLEM:

Transient stability of SMIB system

A 60Hz synchronous generator having inertia constant H = 5 MJ/MVA and a direct axis transient reactance X_d^1 = 0.3 per unit is connected to an infinite bus through a purely reactive circuit as shown in figure. Reactances are marked on the diagram on a common system base. The generator is delivering real power P_e = 0.8 per unit and Q = 0.074 per unit to the infinite bus at a voltage of V = 1 per unit.



- **a)** A temporary three-phase fault occurs at the sending end of the line at point F.When the fault is cleared, both lines are intact. Determine the critical clearing angle and the critical fault clearing time.
- b) A three phase fault occurs at the middle of one of the lines, the fault is cleared and teh faulted line is isolated. Determine the critical clearing angle.

Manual Calculation:

PROGRAM :

a) Pm = 0.8; E = 1.17; V = 1.0;

X1 = 0.65; X2 = inf; X3 = 0.65;

eacfault(Pm, E, V, X1, X2, X3)

b) Pm = 0.8; E = 1.17; V = 1.0;

X1 = 0.65; X2 = 1.8; X3 = 0.8;

eacfault(Pm, E, V, X1, X2, X3)

eacfault

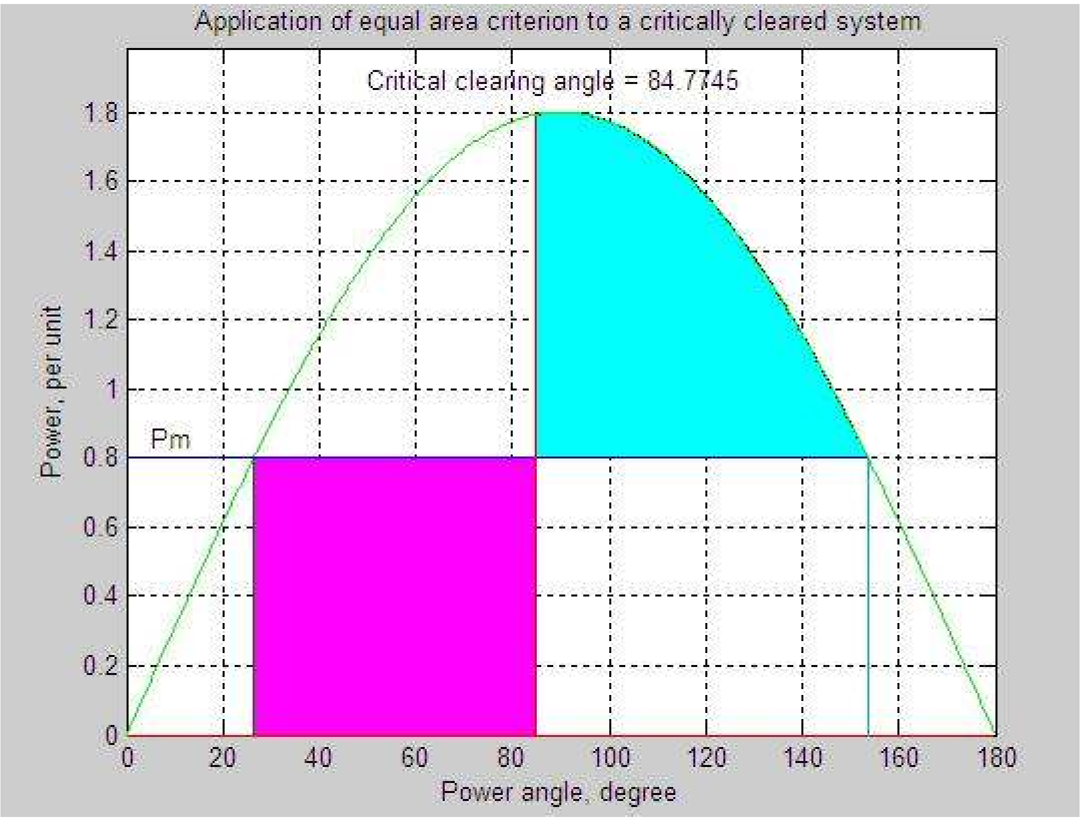
```
function eacfault(Pm, E, V, X1, X2, X3)
if exist('Pm')~=1
Pm = input('Generator output power in p.u. Pm = '); else, end
if exist('E')~=1
E = input('Generator e.m.f. in p.u. E = '); else, end
if exist('V')~=1
V = input('Infinite bus-bar voltage in p.u. V = '); else, end
if exist('X1')~=1
X1 = input('Reactance before Fault in p.u. X1 = '); else, end
if exist('X2')~=1
X2 = input('Reactance during Fault in p.u. X2 = '); else, end
if exist('X3')~=1
X3 = input('Reactance aftere Fault in p.u. X3 = '); else, end
Pe1max = E*V/X1; Pe2max=E*V/X2; Pe3max=E*V/X3;
delta = 0:.01:pi;
Pe1 = Pe1max*sin(delta); Pe2 = Pe2max*sin(delta); Pe3 = Pe3max*sin(delta);
d0 =asin(Pm/Pe1max); dmax = pi-asin(Pm/Pe3max);
cosdc = (Pm*(dmax-d0)+Pe3max*cos(dmax)-Pe2max*cos(d0))/(Pe3max-Pe2max);
  if abs(cosdc) > 1
  fprintf('No critical clearing angle could be found.\n')
  fprintf('system can remain stable during this disturbance.\n\n')
  return
  else, end
dc=acos(cosdc);
if dc > dmax
fprintf('No critical clearing angle could be found.\n')
  fprintf('System can remain stable during this disturbance.\n\n')
  return
  else, end
```

```
Pmx=[0  pi-d0]*180/pi; Pmy=[Pm Pm];
x0=[d0 d0]*180/pi; y0=[0 Pm]; xc=[dc dc]*180/pi; yc=[0 Pe3max*sin(dc)];
xm=[dmax dmax]*180/pi; ym=[0 Pe3max*sin(dmax)];
d0=d0*180/pi; dmax=dmax*180/pi; dc=dc*180/pi;
x=(d0:.1:dc);
y=Pe2max*sin(x*pi/180);
y1=Pe2max*sin(d0*pi/180);
y2=Pe2max*sin(dc*pi/180);
x=[d0 x dc];
y=[Pm y Pm];
xx=dc:.1:dmax;
h=Pe3max*sin(xx*pi/180);
xx=[dc xx  dmax];
hh=[Pm h  Pm];
delta=delta*180/pi;
if X2 == inf
fprintf('\nFor this case tc can be found from analytical formula. \n')
H=input('To find tc enter Inertia Constant H, (or 0 to skip) H = ');
   if H ~= 0
   d0r=d0*pi/180; dcr=dc*pi/180;
   tc = sqrt(2*H*(dcr-d0r)/(pi*60*Pm));
   else, end
else, end
%clc
fprintf('\nInitial power angle      = %7.3f \n', d0)
fprintf('Maximum angle swing      = %7.3f \n', dmax)
fprintf('Critical clearing angle = %7.3f \n\n', dc)
if X2==inf & H~=0
fprintf('Critical clearing time  = %7.3f sec. \n\n', tc)
else, end
h = figure; figure(h);
fill(x,y,'m')
hold;
fill(xx,hh,'c')
plot(delta, Pe1,'-', delta, Pe2,'r-',  delta, Pe3,'g-', Pmx, Pmy,'b-', x0,y0,
xc,yc, xm,ym), grid
Title('Application of equal area criterion to a critically cleared system')
xlabel('Power angle, degree'), ylabel(' Power, per unit')
text(5, 1.07*Pm, 'Pm')
text(50, 1.05*Pe1max,['Critical clearing angle = ',num2str(dc)])
axis([0 180  0 1.1*Pe1max])
hold off;
```

OUTPUT:

a) To find tc enter Inertia Constant H, (or 0 to skip) H = 5

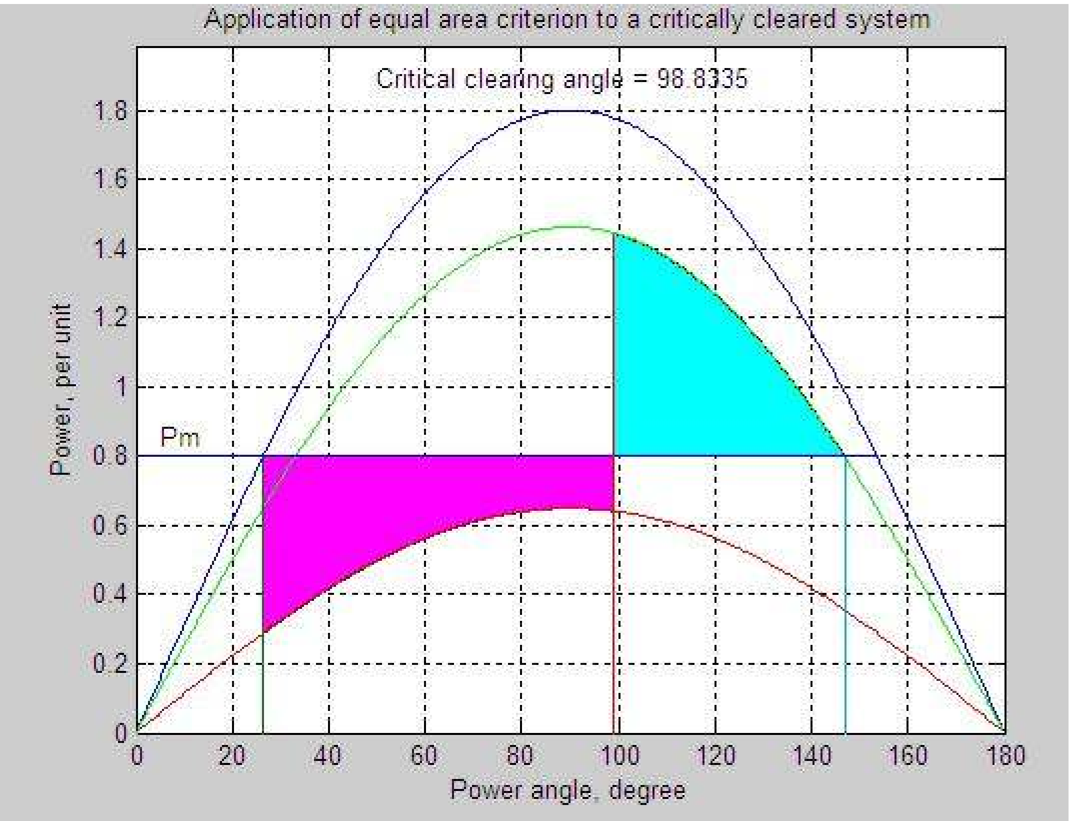
```
Initial power angle    =  26.388

Maximum angle swing    = 153.612

Critical clearing angle =  84.775

Critical clearing time  =   0.260 sec.
```



b) Initial power angle = 26.388

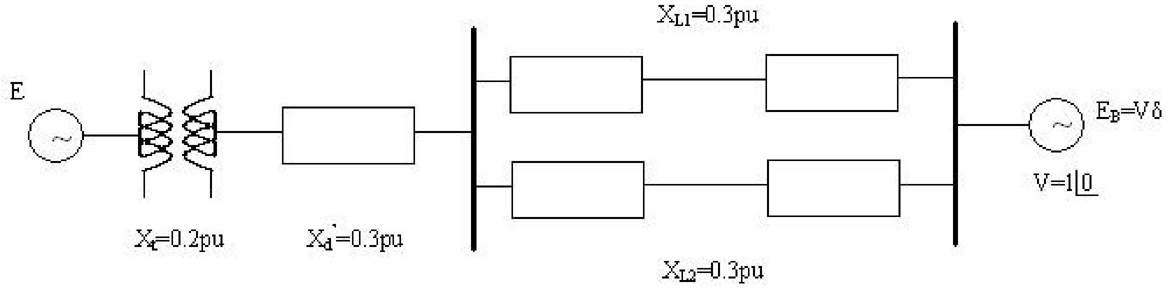
Maximum angle swing = 146.838

Critical clearing angle = 98.834


Application of equal area criterion to a critically cleared system
Critical clearing angle = 98.8335
Pm
Power, per unit
Power angle, degree
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
0
20
40
60
80
100
120
140
160
180

Small signal stability of SMIB system

2) A 60Hz synchronous generator having inertia constant H = 9.94 MJ/MVA and a direct axis transient reactance $X_d^1 = 0.3$ per unit is connected to an infinite bus through a purely reactive circuit as shown in figure. Reactances are marked on the diagram on a common system base. The generator is delivering real power $P_e = 0.6$ per unit and 0.8 power factor lagging to the infinite bus at a voltage of V = 1 per unit.



Assume the per unit damping power co-efficient is D=0.138. Consider a small disturbance of $\Delta\delta$=10°=0.1745 radian. Obtain equations describing the motion of the rotor angle and the generator frequency.

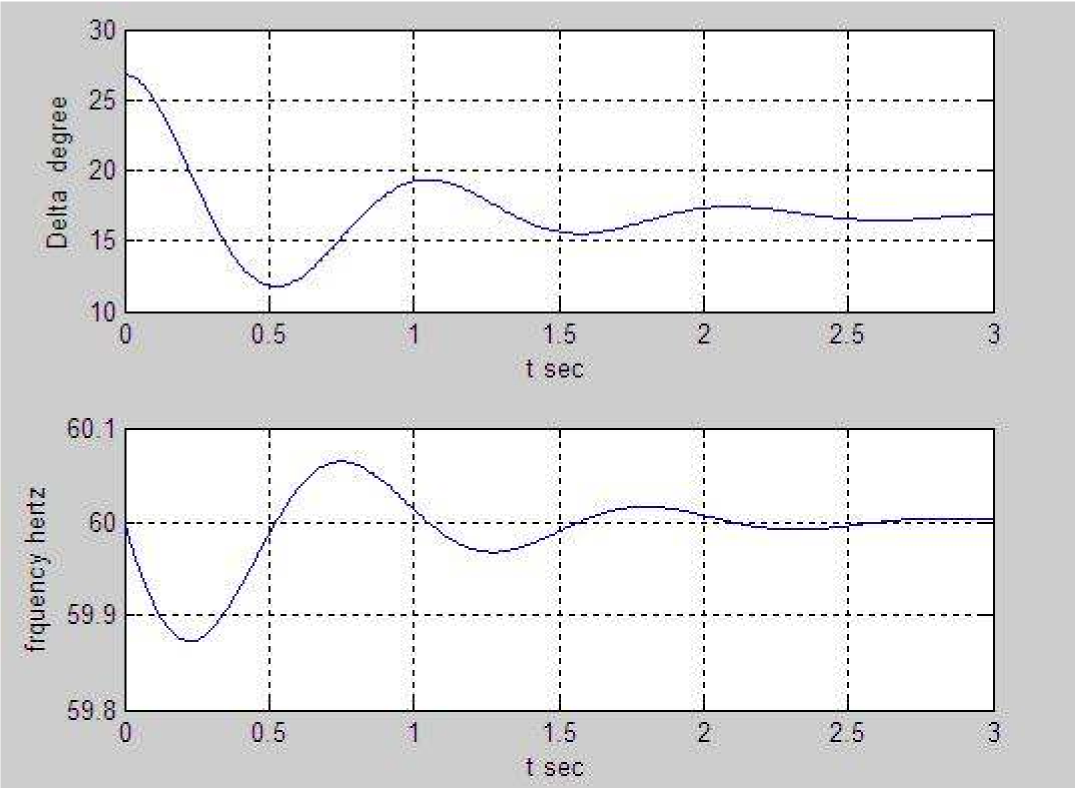
Manual Calculation:

PROGRAM:

```
E=1.35; V=1.0; H=9.94; X=0.65; Pm=0.6; D=0.138; f0=60;
Pmax=E*V/X, d0=asin(Pm/Pmax)
Ps=Pmax*cos(d0)
wn=sqrt(pi*60/H*Ps)
z=D/2*sqrt(pi*60/(H*Ps))
wd=wn*sqrt(1-z^2),fd=wd/(2*pi)
tau=1/(z*wn)
th=acos(z)
Dd0=10*pi/180;
t=0:.01:3;
Dd=Dd0/sqrt(1-z^2)*exp(-z*wn*t).*sin(wd*t+th);
d=(d0+Dd)*180/pi;
Dw=-wn*Dd0/sqrt(1-z^2)*exp(-z*wn*t).*sin(wd*t);
f=f0+Dw/(2*pi);
subplot(2,1,1),plot(t,d),grid
xlabel('t sec'),ylabel('Delta  degree')
subplot(2,1,2),plot(t,f),grid
xlabel('t sec'),ylabel('frquency hertz')
subplot(111)
```

OUTPUT:

```
Pmax =    2.0769

d0 =    0.2931

Ps =    1.9884

wn =    6.1405

z =    0.2131

wd =    5.9995

fd =    0.9549

tau =    0.7643

th =    1.3561
```



RESULT :

Thus the various aspects of transient and small signal stability analysis of single machine infinite bus system were made familiar.

EXP NO: 7

DATE:

ECONOMIC DISPATCH USING LAMBDA ITERATION METHOD

AIM:

To understand the basics of the Economic Dispatch by optimally adjusting the generation schedules of thermal generating units so as to meet the system demand which are required for unit commitment and economic operation of power systems.

SOFTWARE REQUIRED:

MATLAB Software

ALGORITHM:

Step 1:Choose appropriate value of Lagrangian multiplier λ

Step 2:Start iteration iter=0

Step 3:Iteration iter=iter+1

Step 4:Solve for power generated by ith unit using equation

$P_i^{(k)}=(\lambda^{(k)} - b_i)/2(a_i+ \lambda^{(k)} B_{ii})$

Step 5:Check if any P_i is beyond or below the inequality constant

If $P_i < P_{i,min}$, fix $P_i= P_{i,min}$

If $P_i>P_{i,max}$, fix $P_i= P_{i,max}$

Step 6:Calculate the power loss using the equation $P_L=\sum B_{ii} P_i^2$

Step 7:Calculate power mismatch using the formula, $\Delta P^{(k)}=P_D+P_L^{(k)}-\sum P_i^{(k)}$

Step 8:If $\Delta P^{(k)}>0$, then increment λ, $\lambda_{new}=\lambda+0.001$ and go to step 3

else go to step 9

Step 9:If $\Delta P^{(k)}<0$, then decrement λ, $\lambda_{new}=\lambda-0.001$ and go to step 3

else go to step 10

Step 10:If ΔP is less than tolerance value , print the values of generated power and losses

Step 11:Stop

Program:

```
clc;
clear all;
% a        b      c     fc   max     min
data=       [0.00142   7.20    510   1.1  600     150
              0.00194   7.85    310    1   400     100
              0.00482   7.97     78     1   200     050];
ng=length(data(:,1));
a=data(:,1);
b=data(:,2);
c=data(:,3);
fc=data(:,4);
pmax=data(:,5);
pmin=data(:,6);
% loss=[0.00003 0.00009 0.00012];
 loss=[ 0  0   0];
C=fc.*c; B=fc.*b; A=fc.*a;
la=1; pd=850; acc=0.2;
diff=1;
```

```
while acc<(abs(diff));
for i=1:ng;
    p(i)= (la-B(i))/(2*(la*loss(i)+A(i)));
if p(i)<pmin(i);
      p(i)=pmin(i);
end;
if p(i)>pmax(i);
        p(i)=pmax(i);
end;
end;
    LS=sum(((p.*p).*loss));
    diff=(pd+LS-sum(p));
if diff>0
      la=la+0.001;
else la=la-0.001;
end;
end;
PowerShared=p
Lambda=la
Loss=LS
```

Output:

a). When loss = [0.00003 0.00009 0.00012]

```
Power Shared =     435.1026  299.9085  130.6311

Lambda =      9.5290

Loss =    15.8222
```

b). When loss = 0

```
Power Shared =    393.0858  334.5361  122.1992

Lambda =    9.1490

Loss  =     0
```

Results:

Using Mat Lab program Optimal scheduling of generators is done for the given three generator system by Lamda iteration method with and without loss and the calculated values of this system are verified with the output result.

EXP NO: □

DATE:

LOAD – FREQUENCY DYNAMICS OF SINGLE- AREA AND TWO-AREA POWER SYSTEMS

AIM:

To become familiar with the modeling and analysis of load frequency and tie line flow dynamics of a power systems with load frequency controller (LFC) under different control modes and to design improved controllers to obtain the best system response.

OBJECTIVES:

i. To study the time response(both steady state and transient) of area frequency deviation and transient power output change of regulating generator following a small load change in a single-area power system with the regulating generator under "free governor action" for different operating conditions and different system parameters.

ii. To study the time response (both steady state and transient) of area frequency deviation and the turbine output change of regulating generator following a small load change in a single area system provided with an integral frequency controller, to study the effect of changing the gain of the controller and to select the best gain for the controller to obtain the best response.

iii. To analyze the time response of area frequency deviation and net interchange deviation following a small load change in one of the areas in an inter connected two area power system under different control modes, to study the effect of changes in controller parameters on the response and to select the optimal set of parameters for the controllers to obtain the best response under different operating conditions.

LOAD FREQUENCY CONTROL:

Primary control:

The speed change from the synchronous speed initiates the governor control action resulting in all the participating generator-turbine units taking up the change in load, and stabilizes the system frequency.

Secondary control:

It adjusts the load reference set points of selected turbine-generator units so as to give nominal value of frequency. The frequency control is a matter of speed control of the machines in the generating stations. The frequency of a power system is dependent entirely upon the speed in which the generators are rotated by their prime movers. All prime movers, whether they are steam or hydraulic turbines, are equipped with speed governors which are purely mechanical speed sensitive devices, to adjust the gate or control valve opening for the constant speed.

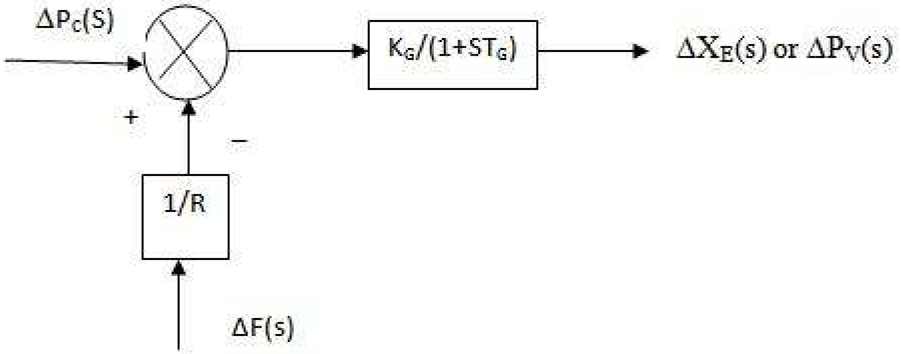
$$N = 120\,f / P$$

Therefore $N \,\alpha\, f$

where, N = Speed in rpm

f = Frequency in Hz

P = Number of poles.



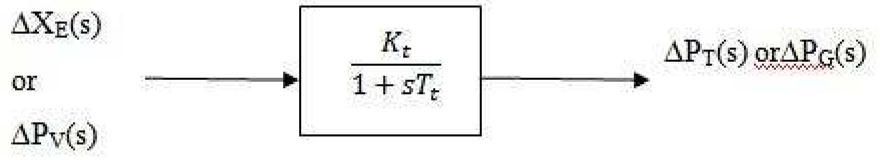
$\Delta X_E(s) = [\Delta Pc\,(s) - (1/R)\,\Delta F(s)\,]\ x\ \frac{k_G}{1+sT_G}$

where $R = \frac{k_1 k_c}{k_2}$ = speed regulation of the governor in Hz/MW

$k_G = \frac{k_2 k_1 k_c}{k_4}$ = Gain of speed governor

$T_G = \frac{1}{k_4 k_3}$ = Time constant of speed governor

Turbine model:

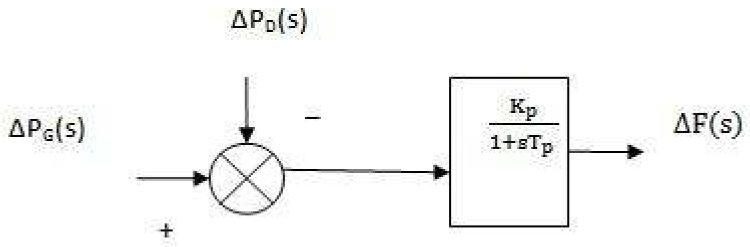


T_t = Time constant of turbine

k $_t$ = Gain constant

$\Delta P_V(s)$ = per unit change in valve position from nominal value

Generator Load model:

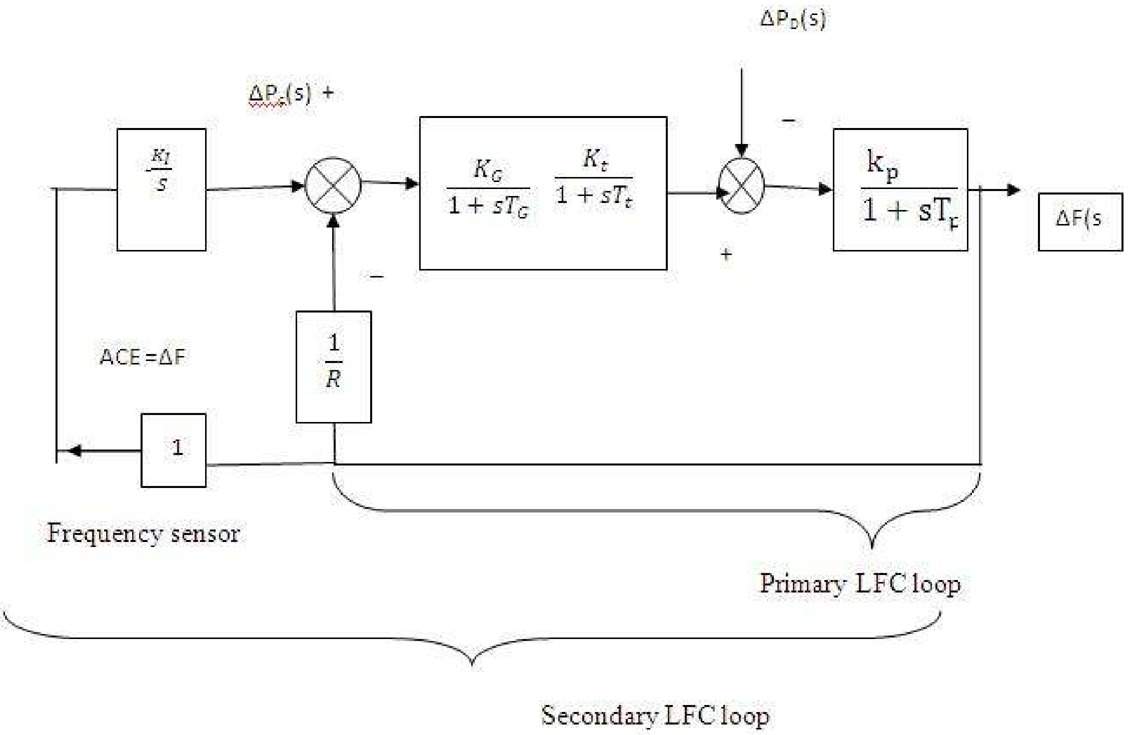


$\Delta P_D(s)$ = Real load change

k $_p$ = 1/B = Power system gain

T_p = 2H/Bf_o = Power system time constant

Model of load frequency control with integral control of single area system:



PROBLEM:

1. The load dynamics of a single area system are P_r=2000 MW; N_{OL}=1000MW;H=5s;f=50Hz; R=4%; T_G=0.08s;T_T=0.3s; Assume linear characteristics . The area has governor but not frequency control. It is subjected to an increase of 20MW. Construct simulink diagram and hence i) determine steady state frequency. ii) If speed governor loop was open, what would be the frequency drop? iii) Prove frequency is zero if secondary controller is included.

Manual Calculation:

PROGRAM:

```
clear all;
clc;
rac=input('Enter the value of Related Area
capcity:'); nol=input('Enter the value of nominal
operating load:'); f=input('Enter the value of
frequency:'); cil=input('Enter the value of change
in load:');

dpd=input('Enter the value of deviation in load in percentage:');
df=input('Enter the value of deviation in frequency in percentage:');
r=input('Enter the value of regulation in Hz/pu MW:');
H=input('Enter the value of Inertia Time constant:');
D=((dpd*(nol))/(df*f));
Dpu=(D/rac);
Kp=(1/Dpu);
Tp=((2*H)/(f*Dpu));
delpd=cil/rac;
if r~= inf
    B=(Dpu+(1/r));
else
    B=(Dpu);
end
Fs=-(delpd/B);
disp('The static gain of the power system in  Hz/pu MW' );
Kp
disp('The Time constant of the power system in  seconds');
Tp
disp('Steady state frequency deviation in Hz');
Fs
```

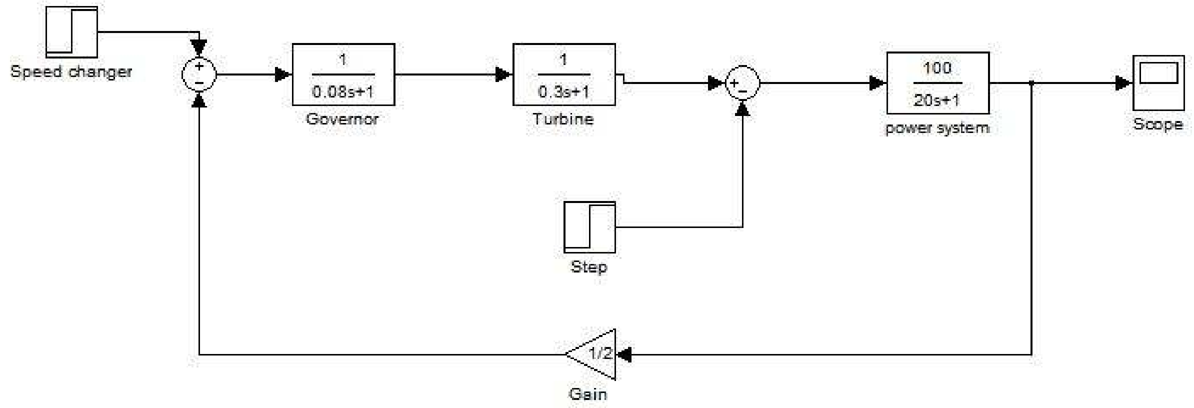
OUTPUT:

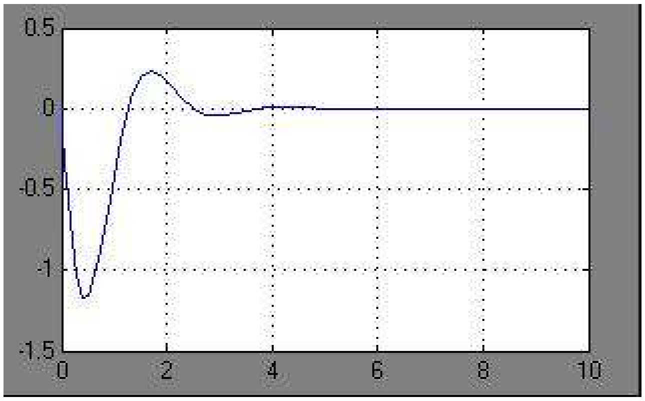
```
Enter the value of Related Area capcity:2000
Enter the value of nominal operating load:1000
Enter the value of frequency:50
Enter the value of change in load:20
Enter the value of deviation in load in percentage:1
Enter the value of deviation in frequency in percentage:1
Enter the value of regulation in Hz/pu MW:2
Enter the value of Inertia Time constant:5
The static gain of the power system in  Hz/pu MW
Kp =   100
The Time constant of teh power system in  seconds
Tp =    20
Steady state frequency deviation in Hz
Fs =   -0.0196
```

OUTPUT:

```
Enter the value of Related Area capcity:2000
Enter the value of nominal operating load:1000
Enter the value of frequency:50
Enter the value of change in load:20
Enter the value of deviation in load in percentage:1
Enter the value of deviation in frequency in percentage:1
Enter the value of regulation in Hz/pu MW:  inf
Enter the value of Inertia Time constant:5
The static gain of the power system in  Hz/pu MW
Kp =   100
The Time constant of teh power system in  seconds
Tp =    20
Steady state frequency deviation in Hz
Fs =    -1
```

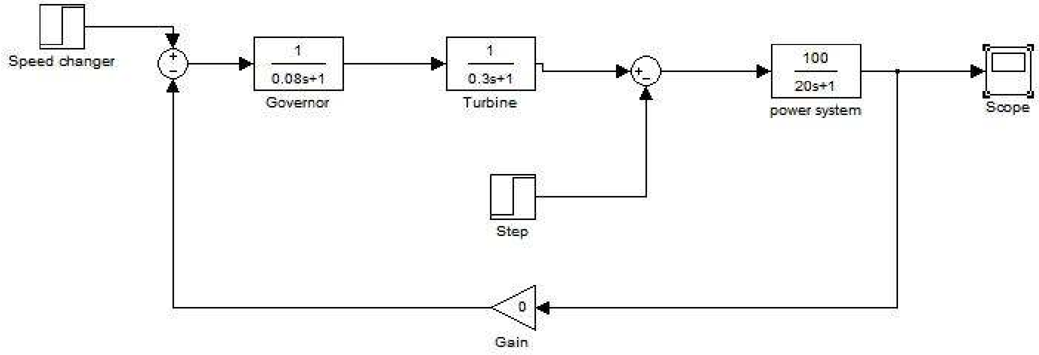
WITH GAIN

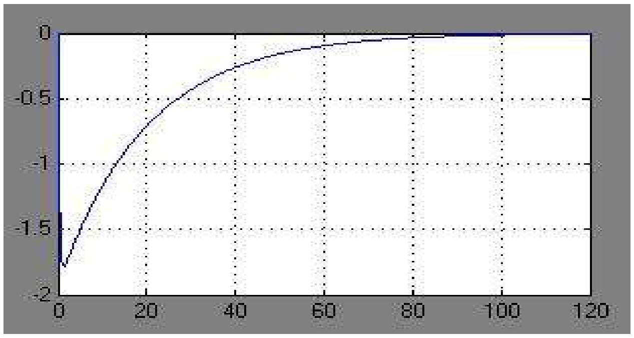




Frequency response

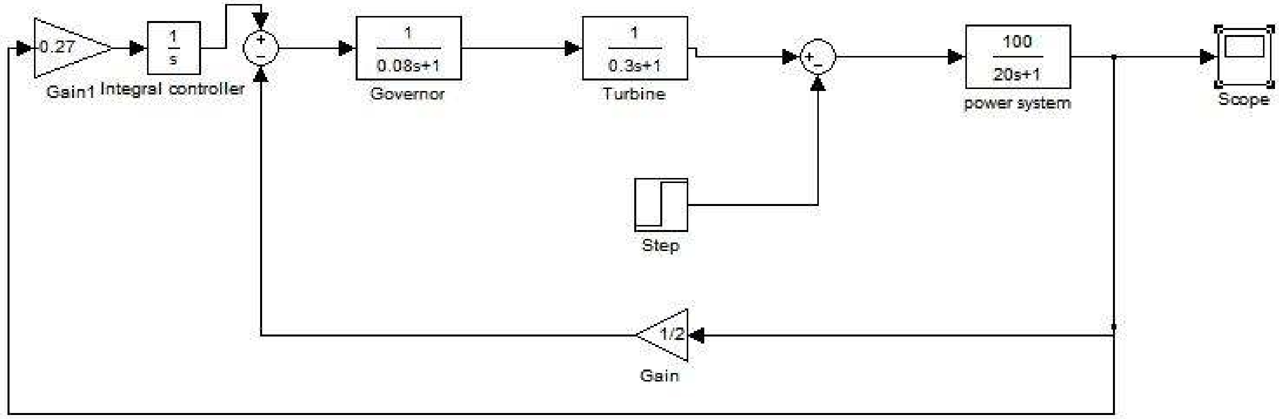
WITHOUT GAIN

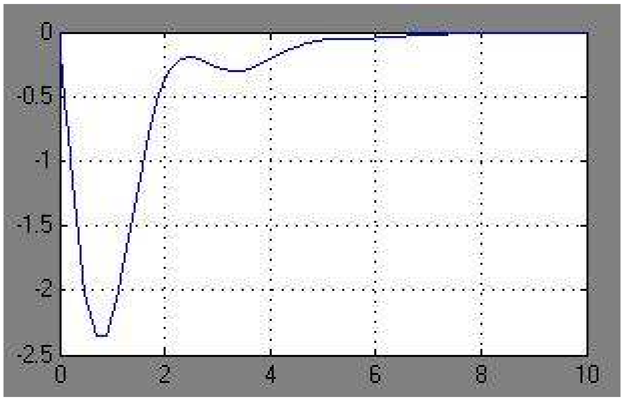




Frequency response

WITH INTEGRAL CONTROLLER





Frequency response

FOR TWO AREA SYSTEMS

1. A two area system connected by a tie line has the following parameters on a 1000 MVA base. R_1=0.05pu, R_2=0.0625pu,D_1=0.6, D_2=0.9,H_1=5,H_2=4; Base $power_1$=Base $power_2$=1000MVA, T_{G1}=0.2s, T_{G2}=0.3s, T_{T1}=0.5s,T_{T2}=0.6s. The units are operating in parallel at the nominal frequency of 50Hz. The synchronizing power coefficient is 2pu. A load change of 200MW occurs in area1. Find the new steady state frequency and change in the tie line flow. Construct simulink block diagram and find deviation in frequency response for the condition mentioned.

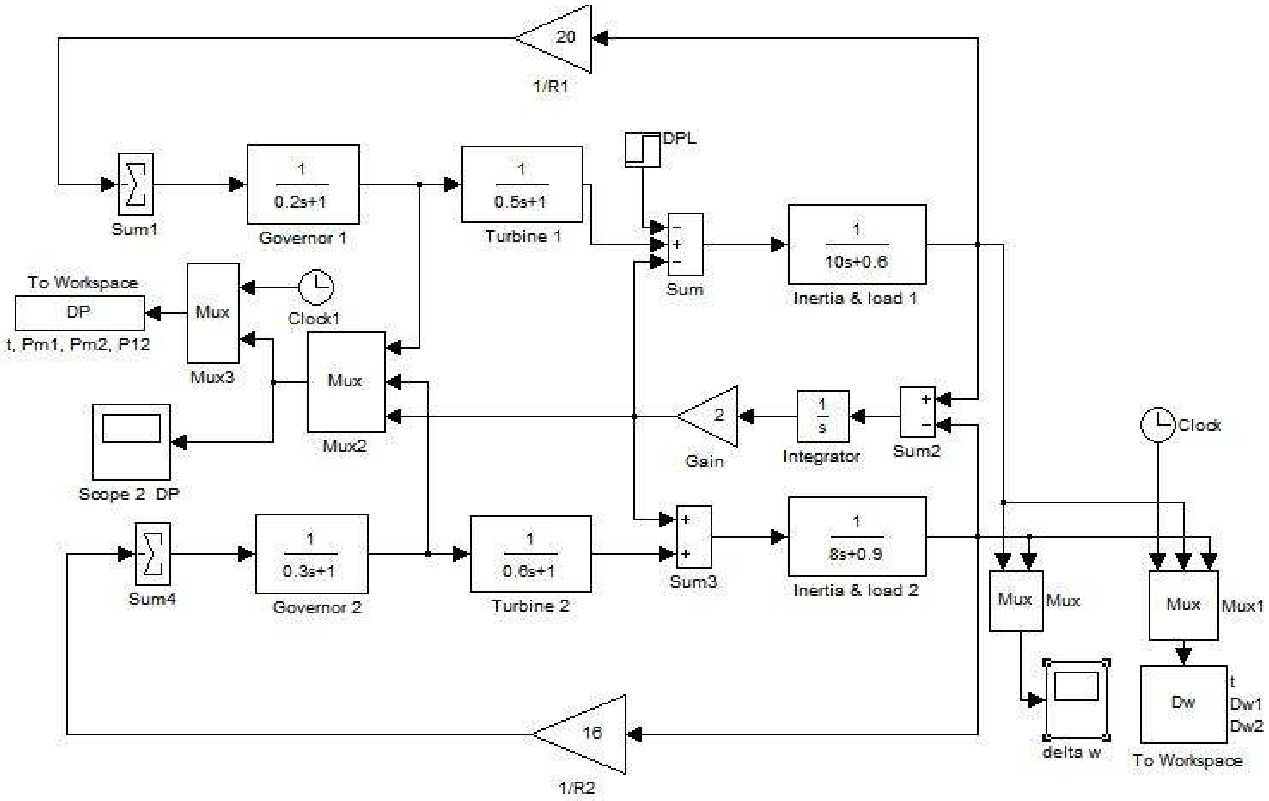
   Manual Calculation:

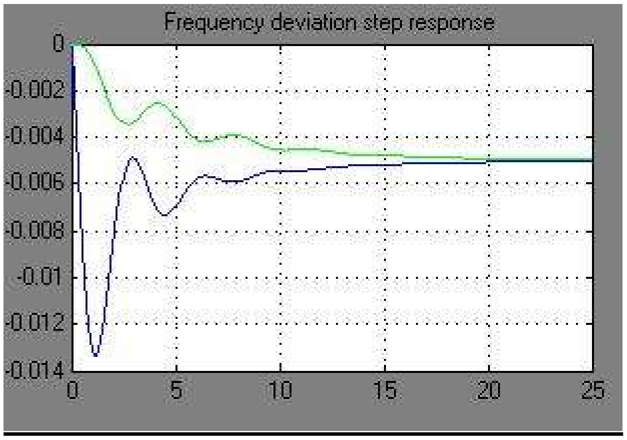
PROGRAM:

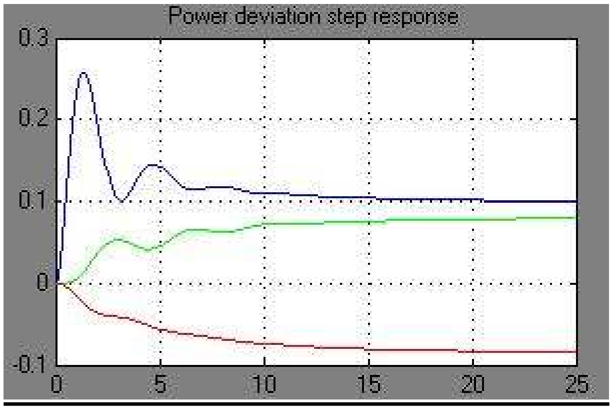
```
clear all;
clc;
rac1=input('Enter the value of Related area capacity-1 in MW:');
cil1=input('Enter the value of change in load-1 in MW:');
D1=input('Enter the value of damping coefficient-1 in pu:');
r1=input('Enter teh value of regulation-1 in pu:');
rac2=input('Enter the value of Related area capacity-2 in MW:');
cil2=input('Enter the value of change in load-2 in MW:');
D2=input('Enter the value of damping coefficient-2 in pu:');
r2=input('Enter teh value of regulation-2 in pu:');
f0=input('Enter the value of frequency:');
B1=(D1+(1/r1));
B1=B1/f0;
B2=(D1+(1/r2));
B2=B2/f0;
delpd1=cil1/rac1;
delpd2=cil2/rac2;
a12=-(rac1/rac2);
delf=((a12*delpd1)-delpd2)/(B2-(a12*B1));
f=f0+delf;
delptie=((B1*delpd2)-(B2*delpd1))/(B2-(a12*B1));
delptie=delptie*rac1;
fprintf('Steady state frequency deviation is              : %g    Hz\n',delf);
fprintf('System frequency                     :%g   Hz\n',f);
fprintf('Change in tie line flow from area 1 to area 2         :%g       MW\n',
delptie);
```

OUTPUT:

```
Enter the value of Related area capacity-1 in MW:1000
Enter the value of change in load-1 in MW:200
Enter the value of damping coefficient-1 in pu:0.6
Enter the value of regulation-1 in pu:0.05
Enter the value of Related area capacity-2 in MW:1000
Enter the value of change in load-2 in MW:0
Enter the value of damping coefficient-2 in pu:0.9
Enter teh value of regulation-2 in pu:0.0625
Enter the value of frequency:50
Steady state frequency deviation is           : -0.268817    Hz
System frequency                    :49.7312   Hz
Change in tie line flow from area 1 to area 2       :-89.2473      MW
```

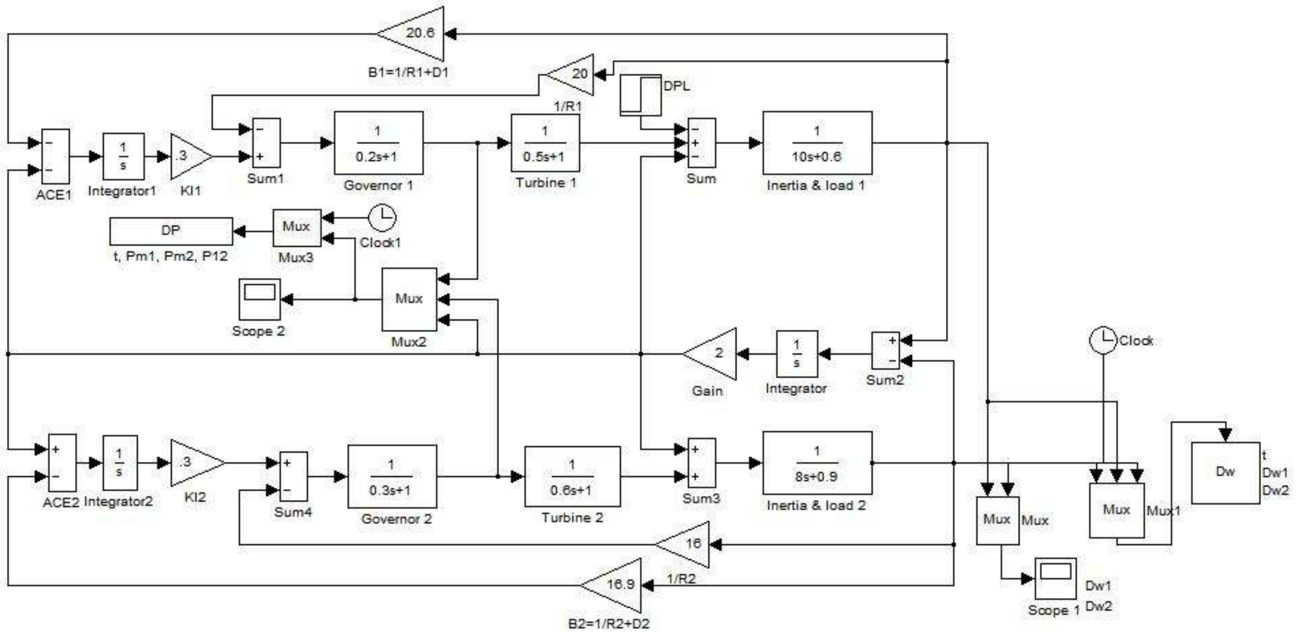




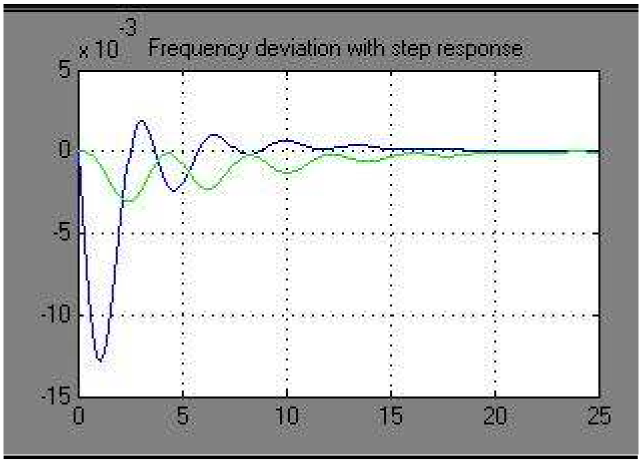


2. A two area system connected by a tie line has the following parameters on a 1000 MVA base. R_1=0.05pu, R_2=0.0625pu, D_1=0.6, D_2=0.9, H_1=5, H_2=4; Base power$_1$=Base power$_2$=1000MVA, T_{G1}=0.2s, T_{G2}=0.3s, T_{T1}=0.5s, T_{T2}=0.6s. The units are operating in parallel at the nominal frequency of 50Hz. The synchronizing power coefficient is 2pu. A load change of 200MW occurs in area1. Find the new steady state frequency and change in the tie line flow. Construct simulink block diagram **with the inclusion of the ACE's** and find deviation in frequency response for the condition mentioned.

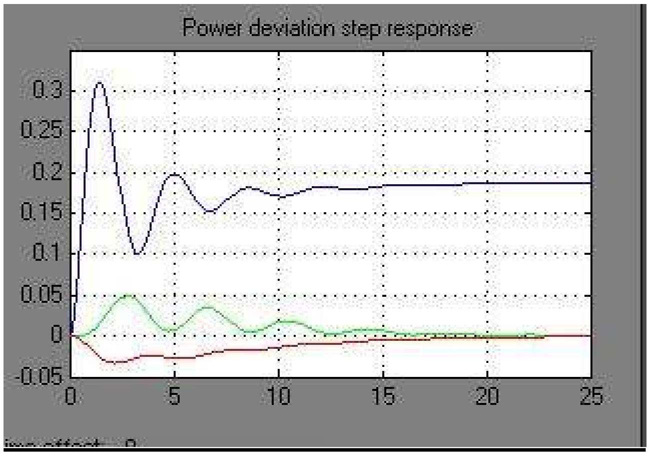
   Manual Calculation:



20.6
B1=1/R1+D1
20
1/R1
DPL
ACE1
1/s
Integrator1
.3
KI1
Sum1
1/(0.2s+1)
Governor 1
1/(0.5s+1)
Turbine 1
Sum
1/(10s+0.6)
Inertia & load 1
DP
t, Pm1, Pm2, P12
Mux
Mux3
Clock1
Scope 2
Mux
Mux2
2
Gain
1/s
Integrator
Sum2
Clock
ACE2
1/s
Integrator2
.3
KI2
Sum4
1/(0.3s+1)
Governor 2
1/(0.6s+1)
Turbine 2
Sum3
1/(8s+0.9)
Inertia & load 2
Mux
Mux
Mux
Mux1
Dw
t
Dw1
Dw2
16
16.9
1/R2
B2=1/R2+D2
Scope 1
Dw1
Dw2



Frequency deviation with step response
x 10^-3
5
0
-5
-10
-15
0
5
10
15
20
25



Power deviation step response
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
0
5
10
15
20
25
ime offset: 0

RESULT:

Thus the modeling and analysis of load frequency and tie line flow dynamics of a power systems with load frequency controller (LFC) under different control modes and to design improved controllers to obtain the best system response was done using Matlab simulink.

Exp.No: 9
Date:

State Estimation

Aim:

To obtain the bests possible estimate of state of the power system for the given set of measurement by weighted least squares method.

State Estimation BY WLSE method

State estimation plays a very important role in the monitoring and control of modern power system. The main aim of this is to obtain the voltages and bus angles by processing the available system data.

State estimation is defined as the data processing algorithm for converting redundant meter reading and other available information into as estimate of the state of electrical power system.

Real time measurement are collected in power system through SCADA system. Typical data includes real and reactive line flows and real and reactive bus injections and bus voltage magnitude. This telemetered data may contain errors. Theseerrors render the outputuseless. It is for this reason that, power system state estimation techniques have been developed.

A commonly used criterion is that of minimizing the sum of the squares of the differences between estimated measurement quantities and actual measurement. This is known as “weighted least squares” criterion. The mathematical model of state estimation is based on the relation between the measurement variable and the state variable.

Let

[Z] = Set of measurements
[X] = The vector of state variables
[f(X] = The equation relating measurement variables to the state variable
[e] = The measurement error vector

We have

[Z] = [f(x)+[e] → (1)

The errors [e1,e2,….em]T are assumed to be independent random variable with Gaussian distribution whose mean is zero. The variationmeasurement errorσiprovides an indication of the certainity about the particular measurements. A large variance indicates that the corresponding measurement accurate.

The objective function to be minimsed

$$J(X) = \sum_{i=1}^{m} \frac{(Zi-fi)^2}{\sigma i} \rightarrow \qquad (2)$$

Here m is the number of measurements

Minimise

J(x)={[f(x)-[Z]}T [W}[f(X/]-[Z]}

Linearsing equation (1) and simultaneously minimizing the objective function (3), We get state correction vector as

[ΔX]={[H][W][H]-1[H][W]{[Z]-F9X)]}} (4)

Where

$$[W] = \begin{bmatrix} w1\ 0\ 0\ 0\ 0\ 0 \\ 0\ W2\ 0\ 0\ 0\ 0 \\ \cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdot \\ 0\ 0\ 0\ 0\ 0\ 0\ Wm \end{bmatrix} = \begin{bmatrix} \frac{1}{\sigma1^2}\ 0\ 0\ 0\ 0 \\ 0\ \frac{1}{\sigma2^2}\ 0\ 0\ 0 \\ \cdots\cdots\cdots\cdots \\ 0\ 0\ 0\ 0\ 0\ \frac{1}{\sigma m^2} \end{bmatrix}$$

[W]=Diagonal matrix which contain weightage value of each measurement
[H]=Matrix of partial differential derivatives of measurement functions with respect to state variables

$$Hij = \frac{\partial fi}{\partial Xi}$$

The correction vector [ΔX] should be computed using the latest available system state must be checked for convergence.

Algorithm:
Read all the relevant data
Initialize the state vector
Compute measurement function [f(x)] and Jocobian matrix [H] using latest known system state variable
Check weather all the elements of [ΔX] are within the tolerance value, if so latest [X] is the present system state or else go to next step.
Update the state vector
[X]=[X0]+[ΔX] and go to step 3

Line data

Line number	Start Bus	End bus	Reactance in p.u
1	1	2	0.2
2	2	3	0.4
3	1	3	0.25

Measurement Data:

S.No	Measurement Quanities	Values (p.u)	Weightages
1	P1	0.72	1.0
2	P1,2	0.68	1.0
3	P3,2	-0.405	0.1
4	P1,3	-0.04	0.2

Program:

```
num = 3;
ybus = ybusppg(num);
zdata = zdatas(num);
bpq = bbusppg(num);
nbus = max(max(zdata(:,4)),max(zdata(:,5)));
type = zdata(:,2);
z = zdata(:,3);
fbus = zdata(:,4);
tbus = zdata(:,5);
Ri = diag(zdata(:,6));
V = ones(nbus,1);
del = zeros(nbus,1);
E = [del(2:end); V];
G = real(ybus);
B = imag(ybus);
```

```
vi = find(type == 1);
ppi = find(type == 2);
qi = find(type == 3);
pf = find(type == 4);
qf = find(type == 5);
nvi = length(vi);
npi = length(ppi);
nqi = length(qi);
npf = length(pf);
nqf = length(qf);
iter = 1;
tol = 5;

while(tol> 1e-4)

    h1 = V(fbus(vi),1);
    h2 = zeros(npi,1);
    h3 = zeros(nqi,1);
    h4 = zeros(npf,1);
    h5 = zeros(nqf,1);

fori = 1:npi
        m = fbus(ppi(i));
for k = 1:nbus
            h2(i) = h2(i) + V(m)*V(k)*(G(m,k)*cos(del(m)-del(k)) + B(m,k)*sin(del(m)-del(k)));
end
end

fori = 1:nqi
        m = fbus(qi(i));
for k = 1:nbus
            h3(i) = h3(i) + V(m)*V(k)*(G(m,k)*sin(del(m)-del(k)) - B(m,k)*cos(del(m)-del(k)));
end
end

fori = 1:npf
        m = fbus(pf(i));
        n = tbus(pf(i));
        h4(i) = -V(m)^2*G(m,n) - V(m)*V(n)*(-G(m,n)*cos(del(m)-del(n)) - B(m,n)*sin(del(m)-
del(n)));
end

fori = 1:nqf
        m = fbus(qf(i));
        n = tbus(qf(i));
        h5(i) = -V(m)^2*(-B(m,n)+bpq(m,n)) - V(m)*V(n)*(-G(m,n)*sin(del(m)-del(n)) +
B(m,n)*cos(del(m)-del(n)));
end

    h = [h1; h2; h3; h4; h5];

        r = z - h;


    H11 = zeros(nvi,nbus-1);

    H12 = zeros(nvi,nbus);
```

```
for k = 1:nvi
for n = 1:nbus
if n == k
H12(k,n) = 1;
end
end
end

        H21 = zeros(npi,nbus-1);
fori = 1:npi
        m = fbus(ppi(i));
for k = 1:(nbus-1)
if k+1 == m
for n = 1:nbus
                    H21(i,k) = H21(i,k) + V(m)* V(n)*(-G(m,n)*sin(del(m)-del(n)) + B(m,n)*cos(del(m)-
del(n)));
end
H21(i,k) = H21(i,k) - V(m)^2*B(m,m);
else
                H21(i,k) = V(m)* V(k+1)*(G(m,k+1)*sin(del(m)-del(k+1)) - B(m,k+1)*cos(del(m)-
del(k+1)));
end
end
end


    H22 = zeros(npi,nbus);
fori = 1:npi
        m = fbus(ppi(i));
for k = 1:(nbus)
if k == m
for n = 1:nbus
                    H22(i,k) = H22(i,k) + V(n)*(G(m,n)*cos(del(m)-del(n)) + B(m,n)*sin(del(m)-del(n)));
end
H22(i,k) = H22(i,k) + V(m)*G(m,m);
else
                H22(i,k) = V(m)*(G(m,k)*cos(del(m)-del(k)) + B(m,k)*sin(del(m)-del(k)));
end
end
end

% H31 - Derivative of Reactive Power Injections with Angles..
    H31 = zeros(nqi,nbus-1);
fori = 1:nqi
        m = fbus(qi(i));
for k = 1:(nbus-1)
if k+1 == m
for n = 1:nbus
                    H31(i,k) = H31(i,k) + V(m)* V(n)*(G(m,n)*cos(del(m)-del(n)) + B(m,n)*sin(del(m)-
del(n)));
end
H31(i,k) = H31(i,k) - V(m)^2*G(m,m);
else
                H31(i,k) = V(m)* V(k+1)*(-G(m,k+1)*cos(del(m)-del(k+1)) - B(m,k+1)*sin(del(m)-
del(k+1)));
end
end
end
```

```
    H32 = zeros(nqi,nbus);
fori = 1:nqi
        m = fbus(qi(i));
for k = 1:(nbus)
if k == m
for n = 1:nbus
                    H32(i,k) = H32(i,k) + V(n)*(G(m,n)*sin(del(m)-del(n)) - B(m,n)*cos(del(m)-del(n)));
end
H32(i,k) = H32(i,k) - V(m)*B(m,m);
else
                H32(i,k) = V(m)*(G(m,k)*sin(del(m)-del(k)) - B(m,k)*cos(del(m)-del(k)));
end
end
end


    H41 = zeros(npf,nbus-1);
fori = 1:npf
        m = fbus(pf(i));
        n = tbus(pf(i));
for k = 1:(nbus-1)
if k+1 == m
                H41(i,k) = V(m)* V(n)*(-G(m,n)*sin(del(m)-del(n)) + B(m,n)*cos(del(m)-del(n)));
elseif k+1 == n
                H41(i,k) = -V(m)* V(n)*(-G(m,n)*sin(del(m)-del(n)) + B(m,n)*cos(del(m)-del(n)));
else
H41(i,k) = 0;
end
end
end
end

% H42 - Derivative of Real Power Flows with V..
    H42 = zeros(npf,nbus);
fori = 1:npf
        m = fbus(pf(i));
        n = tbus(pf(i));
for k = 1:nbus
if k == m
                H42(i,k) = -V(n)*(-G(m,n)*cos(del(m)-del(n)) - B(m,n)*sin(del(m)-del(n))) -
2*G(m,n)*V(m);
elseif k == n
                H42(i,k) = -V(m)*(-G(m,n)*cos(del(m)-del(n)) - B(m,n)*sin(del(m)-del(n)));
else
H42(i,k) = 0;
end
end
end
end

    H51 = zeros(nqf,nbus-1);
fori = 1:nqf
        m = fbus(qf(i));
        n = tbus(qf(i));
for k = 1:(nbus-1)
if k+1 == m
                H51(i,k) = -V(m)* V(n)*(-G(m,n)*cos(del(m)-del(n)) - B(m,n)*sin(del(m)-del(n)));
elseif k+1 == n
                H51(i,k) = V(m)* V(n)*(-G(m,n)*cos(del(m)-del(n)) - B(m,n)*sin(del(m)-del(n)));
```

```
else
H51(i,k) = 0;
end
end
end
end


    H52 = zeros(nqf,nbus);
fori = 1:nqf
        m = fbus(qf(i));
        n = tbus(qf(i));
for k = 1:nbus
if k == m
                H52(i,k) = -V(n)*(-G(m,n)*sin(del(m)-del(n)) + B(m,n)*cos(del(m)-del(n))) -
2*V(m)*(-B(m,n)+ bpq(m,n));
elseif k == n
                H52(i,k) = -V(m)*(-G(m,n)*sin(del(m)-del(n)) + B(m,n)*cos(del(m)-del(n)));
else
H52(i,k) = 0;
end
end
end
end


    H = [H11 H12; H21 H22; H31 H32; H41 H42; H51 H52];

    Gm = H'*inv(Ri)*H;
    J = sum(inv(Ri)*r.^2);

dE = inv(Gm)*(H'*inv(Ri)*r);
    E = E + dE;
del(2:end) = E(1:nbus-1);
    V = E(nbus:end);
iter = iter + 1;
tol = max(abs(dE));
end

CvE = diag(inv(H'*inv(Ri)*H));
Del = 180/pi*del;
E2 = [V Del];
disp('-------- State Estimation ------------------');
disp('--------------------------');
disp('| Bus |    V   |  Angle  | ');
disp('| No  |   pu   |  Degree | ');
disp('--------------------------');
for m = 1:n
fprintf('%4g', m); fprintf('  %8.4f', V(m)); fprintf('   %8.4f', Del(m)); fprintf('\n');
end
disp('--------------------------------------------');
```

```
functionbbus = bbusppg(num)
linedata = linedatas(num);
fb = linedata(:,1);
tb = linedata(:,2);
b = linedata(:,5);
nbus = max(max(fb),max(tb));
nbranch = length(fb);
bbus = zeros(nbus,nbus);

for k=1:nbranch
bbus(fb(k),tb(k)) = b(k);
bbus(tb(k),fb(k)) = bbus(fb(k),tb(k));
end
functionybus = ybusppg(num)  % Returns ybus

linedata = linedatas(num);
fb = linedata(:,1);
tb = linedata(:,2);
r = linedata(:,3);
x = linedata(:,4);
b = linedata(:,5);
a = linedata(:,6);
z = r + i*x;
y = 1./z;
b = i*b;

nbus = max(max(fb),max(tb));
nbranch = length(fb);
ybus = zeros(nbus,nbus);

for k=1:nbranch
ybus(fb(k),tb(k)) = ybus(fb(k),tb(k))-y(k)/a(k);
ybus(tb(k),fb(k)) = ybus(fb(k),tb(k));
end

for m =1:nbus
for n =1:nbranch
if fb(n) == m
ybus(m,m) = ybus(m,m) + y(n)/(a(n)^2) + b(n);
elseiftb(n) == m
ybus(m,m) = ybus(m,m) + y(n) + b(n);
end
end
end


function [rho theta] = rect2pol(x)
rho = sqrt(real(x).^2 + imag(x).^2);
theta = atan(imag(x)./real(x));

functionrect = pol2rect(rho,theta)
rect = rho.*cos(theta) + j*rho.*sin(theta);
```

RESULT:

The state of the given system has estimated using weighted least square method and the results are found to be correct

EXP NO: 10

DATE:

ELECTROMAGNETIC TRANSIENTS IN POWER SYSTEMS

AIM:

(i) To study and understand the electromagnetic transient phenomena in power systems caused due to switching and fault by PSCAD software.

ii) To become proficient in the usage of PSCAD software to address problems in the areas of overvoltage protection and mitigation and insulation coordination of EHV systems.

OBJECTIVES:

a)To study the transients due to energization of a single-phase and three-phase load from a non-ideal source with line represented by π model.

b)To study the transients due to energization of a single-phase and three-phase load from a non-ideal source and line represented by distributed parameters.

c)To study the transient over voltages due to faults for a SLG fault at far end of a line.

d) To study the Transient Recovery Voltage (TRV) associated with a breaker for a

Solution Method for Electromagnetic Transients Analysis

Intentional and inadvertent switching operations in EHV systems initiate over voltages, which might attain dangerous values resulting in destruction of apparatus. Accurate computation of these over voltages is essential for proper sizing, coordination of insulation of various equipments and specification of protective devices. Meaningful design of EHV systems is dependent on modeling philosophy built into a computer program. The models of equipment's must be detailed enough to reproduce actual conditions successfully – an important aspect where a general purpose digital computer program scores over transient network analyzers. The program employs a direct integration time-domain technique evolved by Dommel. The essence of this method is discretization of differential equations associated with network elements using trapezoidal rule of integration and solution of the resulting difference equations for the unknown voltages. Any network which consists of interconnections of resistances, inductances, capacitances, single and multiphase π circuits, distributed parameter lines, and certain other elements can be solved. To keep the explanations simple, however single phase network elements will be used rather than the more complex multiphase network elements.

SOLUTION METHOD FOR ELECTROMAGNETIC TRANSIENTS ANALYSIS:

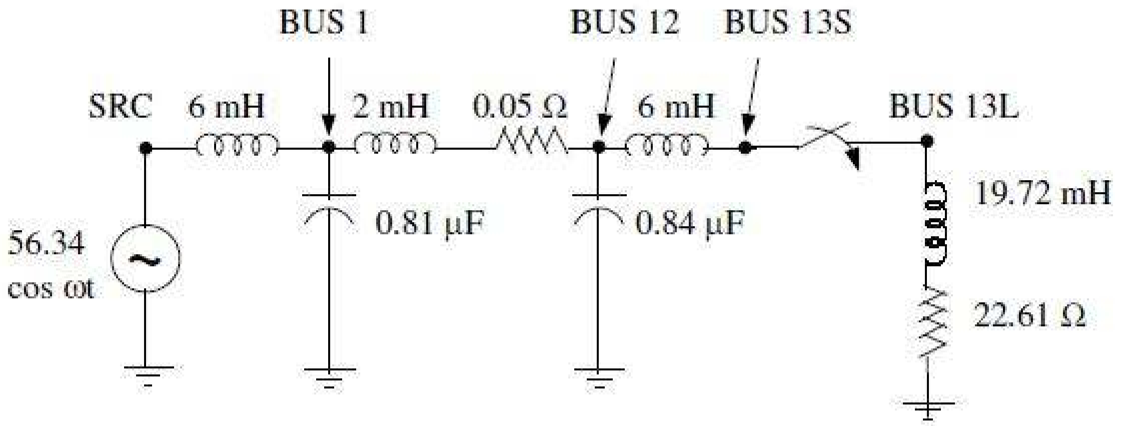
Intentional and inadvertent switching operations in EHV systems initiate over voltages, which might attain dangerous values resulting in destruction of apparatus. Accurate computation of these voltages is essential for proper sizing, co-ordination of insulation of various equipments and specification of protective devices. Meaningful use of EHV is dependent on modeling philosophy built into a computer program. The models of equipments must be detailed enough to reproduce actual conditions successfully –an important aspect where a general purpose digital computer program scores over transient network analyzers.

The program employs a direct integration time-domain technique evolved by Dommel. The essence of this method is discretization of differential equations associated with network elements using trapezoidal rule of integration and solution of the resulting equations for the unknown voltages. Any network which consists of interconnections of resistances, inductances, capacitances, single and multiphase pi circuits, distributed parameter lines, and certain other elements can be solved. To keep the explanations simple, however, single phase network elements will be used, rather than the complex multiphase network elements.

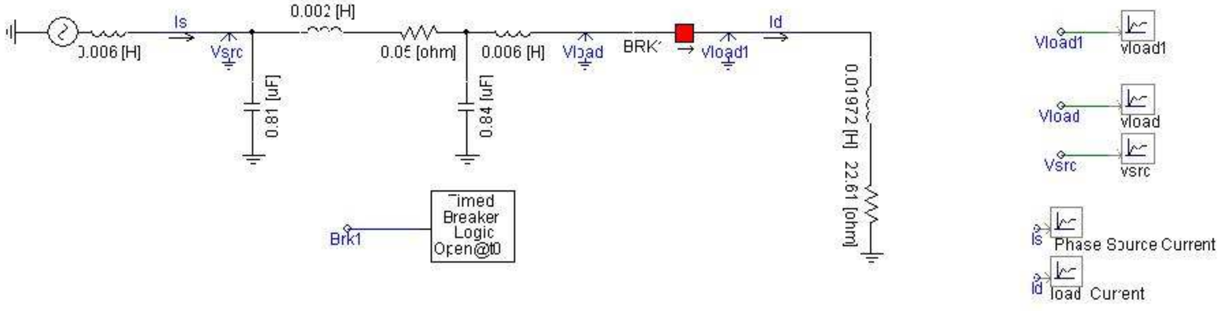
PROBLEM

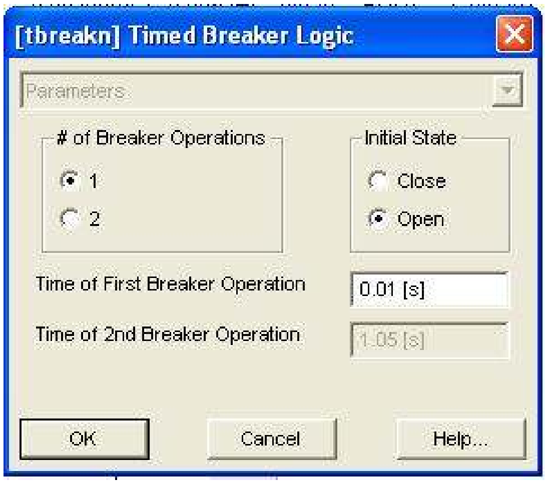
1. Prepare the data for the network given in the given figure and run PSCAD software .Obtain the plots of source voltage, load bus voltage and load current following the energization of a single-phase load. Comment on the results. Double the source inductance and obtain the plots of the variables mentioned earlier. Comment on the effect of doubling the source inductance. Energization of a single phase 0.95 pf load from a non ideal source and a more realistic line representation (lumped R,L,C) using PSCAD software.

(i) Circuit Diagram



PSCAD MODEL





[tbreakn] Timed Breaker Logic

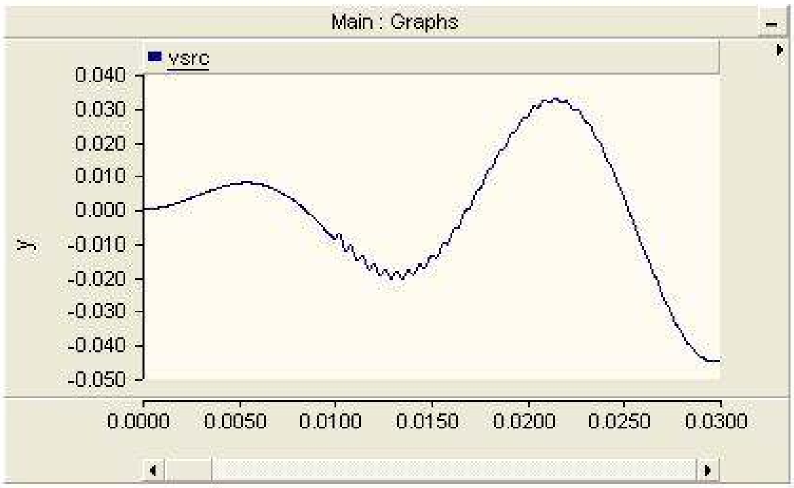
Parameters

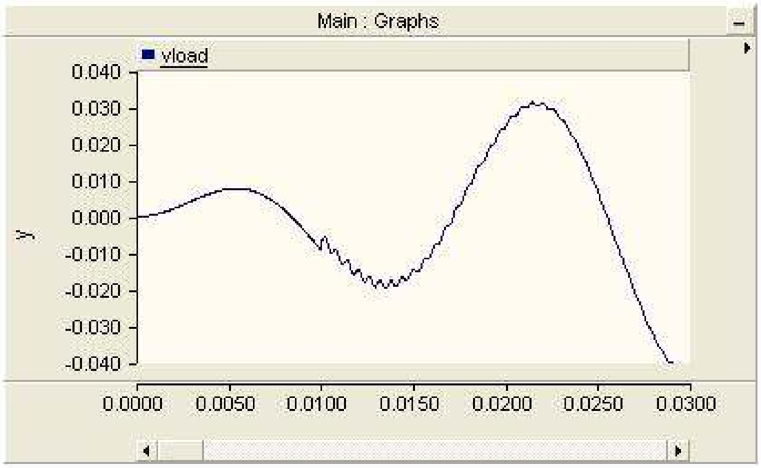
# of Breaker Operations: ● 1 ○ 2

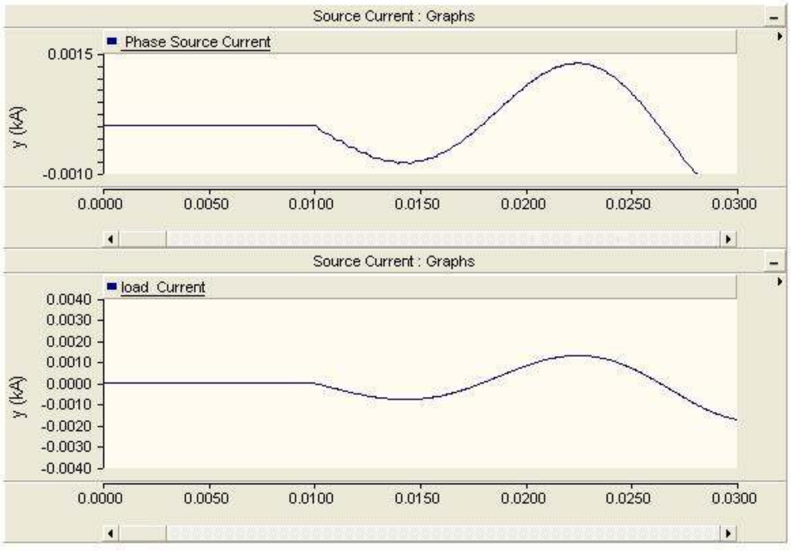
Initial State: ○ Close ● Open

Time of First Breaker Operation: 0.01 [s]

Time of 2nd Breaker Operation: 1.05 [s]







RESULT:

Thus the electromagnetic transient phenomena in power systems caused due to switching and fault by using PSCAD software are analyzed and results were obtained.

ADITIONAL EXP NO:1
DATE:

LOAD FLOW ANALYSIS OF A GIVEN POWER SYSTEM WITH STATCOM

AIM:

To calculate the compensated voltage and angle in the given system using STATCOM as the compensator

SOFTWARE REQUIRED:

Power system module of MATLAB

THEORY:

The STATCOM (or SSC) is a shunt-connected reactive-power compensation device that is capable of generating and /or absorbing reactive power and in which the output can be varied to control the specific parameters of an electric power system. It is in general a solid-state switching converter capable of generating or absorbing independently controllable real and reactive power at its output terminals when it is fed from an energy source or energy-storage device at its input terminals. Specifically, the STATCOM considered in this chapter is a voltage-source converter that, from a given input of dc voltage, produces a set of 3-phase ac-output voltages, each in phase with and coupled to the corresponding ac system voltage through a relatively small reactance (which is provided by either an interface reactor or the leakage inductance of a coupling transformer). The dc voltage is provided by an energy-storage capacitor. A STATCOM can improve power-system performance in such areas as the following:

1. The dynamic voltage control in transmission and distribution systems;

2. The power-oscillation damping in power-transmission systems;

3. The transient stability;

4. The voltage flicker control; and

5. The control of not only reactive power but also (if needed) active power in the connected line, requiring a dc energy source.

Furthermore, a STATCOM does the following:

1. It occupies a small footprint, for it replaces passive banks of circuit elements by compact electronic converters;

2. It offers modular, factory-built equipment, thereby reducing site work and commissioning time; and

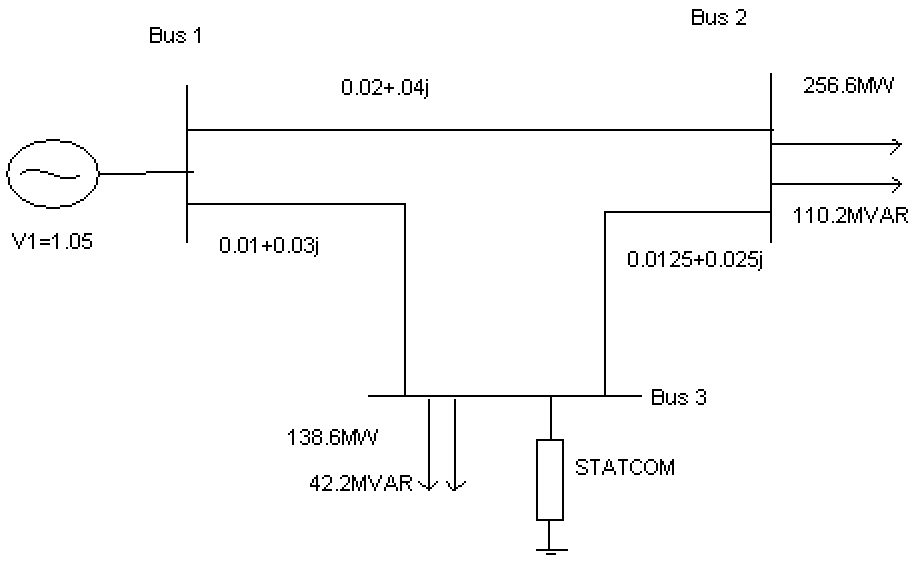
3. It uses encapsulated electronic converters, thereby minimizing its environmental impact.

A STATCOM is analogous to an ideal synchronous machine, which generates a balanced set of three sinusoidal voltages at the fundamental frequency with controllable amplitude and phase angle. This ideal machine has no inertia, is practically instantaneous, does not significantly alter the existing system impedance, and can internally generate reactive (both capacitive and inductive) power.

The Tennessee Valley Authority (TVA) installed the first 100-MVA STATCOM in 1995 at its Sullivan substation. The application of this STATCOM is expected to reduce the TVA's need for load tap changers, thereby achieving savings by minimizing the potential for transformer failure. This STATCOM aids in resolving the off-peak dilemma of over voltages in the Sullivan substation area while avoiding the more labor- and space-intensive installation of an additional transformer bank. Also, this STATCOM provides instantaneous control and therefore increased capacity of transmission voltage, providing the TVA with greater flexibility in bulk-power transactions, and it also increases the system reliability by damping grids of major oscillations in this grid.

EXERCISE:

Rs=0.01; Xs=0.1;Rp=200;K+=0.9



BUS DATA OF 3 BUS SYSTEM:

Bus no.	Bus code	V p.u.	angle	load		gen	
				MW	MVAR	MW	MVAR
1	1	1	0	0	0	0	0
2	0	1	0	256.6	110.2	0	0
3	0	1	0	138.6	45.2	0	0

LINE DATA:

Bus		R pu	X pu	B pu
From	To			
1	2	0.02	0.04	0.05

1	3	0.01	0.03	0.03
2	3	0.0125	0.025	0.06

PROGRAM:

```
clear all;
clc;
n=3;
pd=[0 2.562 1.102];
qd=[0 1.386 0.452];
qg=[0 0 0];
pg=[0 0 0];
vs=[1.05 1 1];
theta=[0 0 0];
con=0.1;
yb=[20-50j -10+20j -10+30j;
-10+20j 26-52j -16+32j;
-10+30j -16+32j 26-62j];
zs=0.01+0.1j;
beta=angle(zs);
rp=200;
k=0.9;
m=1;
cont=.1;
vdc=1;
alpha=0;
b=imag(yb);
g=real(yb);
```

```
an=angle(yb);
my=abs(yb);
iter=1;
while(cont>0.01 && iter<4)
iter
vc=(k*k)*(m)*(vdc);
pg(3)=((vc*vs(3)*cos(theta(3)-alpha+beta))-(vs(3)*vs(3)*cos(beta)))/abs(zs);
qg(3)=((vc*vs(3)*sin(theta(3)-alpha+beta))-(vs(3)*vs(3)*sin(beta)))/abs(zs);
pac=((vc^2*cos(beta))-(vs(3)*vc*cos(beta+alpha-theta(3))))/abs(zs);
p=pg-pd;
q=qg-qd;
for o=1:n
pp(o)=0;
qq(o)=0;
for l=1:n
pe(o)=vs(o)*vs(l)*my(o,l)*cos(an(o,l)-theta(o)+theta(l))+pp(o);
pp(o)=pe(o);
qe(o)=-vs(o)*vs(l)*my(o,l)*sin(an(o,l)-theta(o)+theta(l))+qq(o);
qq(o)=qe(o);
end
end
pp;
qq;
pchang(1:2)= p(2:3)-pp(2:3);
qchang(1:2)=q(2:3)-qq(2:3);
pdc=(vdc^2)/rp;
pext=pac-pdc;
```

```
del=[0 0 0 0 0];

del=[pchang qchang pext];

%calculation of jacobian

for k=2:n

for l=1:n

if k~=l

H(k,l)=-vs(k)*vs(l)*my(k,l)*sin(an(k,l)+theta(l)-theta(k));

N(k,l)=vs(k)*vs(l)*my(k,l)*cos(an(k,l)+theta(l)-theta(k));

J(k,l)=-vs(k)*vs(l)*my(k,l)*cos(an(k,l)+theta(l)-theta(k));

L(k,l)=-vs(k)*vs(l)*my(k,l)*sin(an(k,l)+theta(l)-theta(k));

else

H(k,l)=-qq(k)-vs(k)*vs(k)*b(k,k);

N(k,l)=pp(k)+vs(k)*vs(k)*g(k,k);

J(k,l)=pp(k)-vs(k)*vs(k)*g(k,k);

L(k,l)=qq(k)-vs(k)*vs(k)*b(k,k);

end

end

end

H11(1:2,1:2)=H(2:3,2:3);

N12(1:2,1:2)=N(2:3,2:3);

J21(1:2,1:2)=J(2:3,2:3);

L22(1:2,1:2)=L(2:3,2:3);

jac=zeros(5,5);

jac=[H11 N12;J21 L22];

jac(1,4)=0;

jac(2,2)=(vs(3)*vc*sin(theta(3)-alpha+beta)/abs(zs))-(qq(k)-vs(k)*vs(k)*b(k,k));

jac(2,4)=-(k*vdc*cos(theta(3)-alpha+beta))/abs(zs);
jac(2,5)=-(k*m*vdc*sin(theta(3)-alpha+beta))/abs(zs);

jac(3,4)=0;

jac(4,2)=-((vs(3)*vdc*cos(theta(3)-alpha+beta))/abs(zs))-(pp(k)-vs(k)*vs(k)*g(k,k));

jac(4,4)=(k*vdc*vs(3)*sin(theta(3)-alpha+beta))/abs(zs);

jac(4,5)=(-k*m*vdc*vs(3)*cos(theta(3)-alpha+beta))/abs(zs);

jac(5,2)=-(k*vdc*m*vs(3)*sin(alpha-theta(3)+beta))/abs(zs);

jac(5,4)=(k*m*vs(3)*vdc*sin(alpha-theta(3)+beta))/abs(zs);

jac(5,5)=-((vdc*k*vs(3)*cos(alpha-theta(3)+beta))-(2*k*k*m*vdc^2*cos(beta)))/abs(zs);

jac;

delta=(jac)\del';

dtheta(2:3)=delta(1:2);

theta=theta+dtheta

dv=[0 0 0];

dv(2:2)=delta(3:3);

vs=vs+dv

con=max(abs(dv));

iter=iter+1;

m=m+delta(4);

alpha=alpha+delta(5);

end
```

OUTPUT:

```
iter =     1

theta =     0   -0.0262    0.0033

vs =    1.0500    0.9778    1.0000
```

```
iter =     2

theta =     0   -0.1781   -0.2471

vs =    1.0500    0.9770    1.0000

iter =     3

theta =       0   -0.5447   -0.8483

vs =  1.0500    0.9305    1.0000
```

RESULT:

Thus the compensated voltage and angle in the given system is calculated by using STATCOM as the compensator

ADITIONAL EXP NO:2 **DATE:**

TRANSIENT ANALYSIS OF SINGLE MACHINE INFINITE BUS (SMIB) SYSTEM WITH STATCOM

AIM:

To analyse the transient performance of Single Machine Infinite Bus (SMIB) system with STATCOM using MATLAB.

THEORY:

A STATCOM is a voltage- sourced converter (VSC) –based shunt FACTS device and is capable of injecting controllable reactive current into the system. Consider that a STATCOM is placed at bus m in the SMIB system as shown in Fig.(1).The equivalent circuit of the system is shown in Fig.(2) where the STATCOM is represented by a shunt reactive current source I_s.

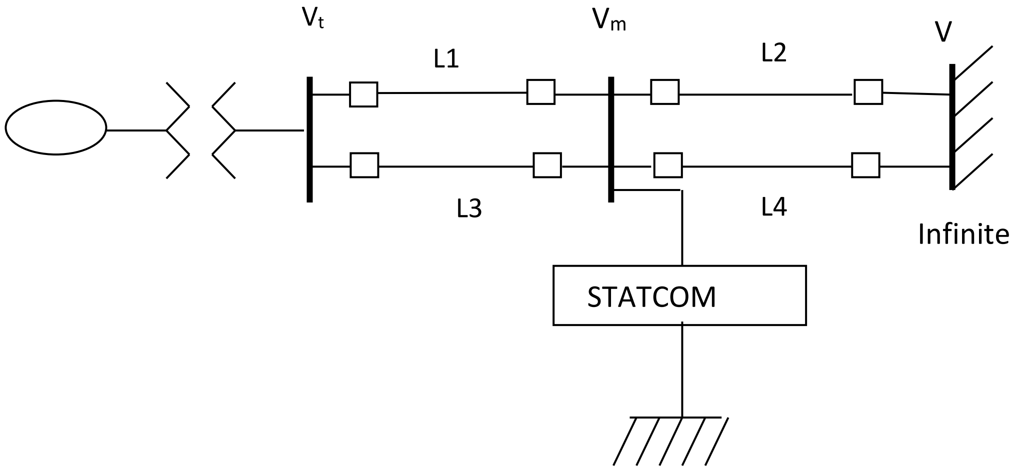


Fig 1 Schematic diagram of SMIB system with STATCOM

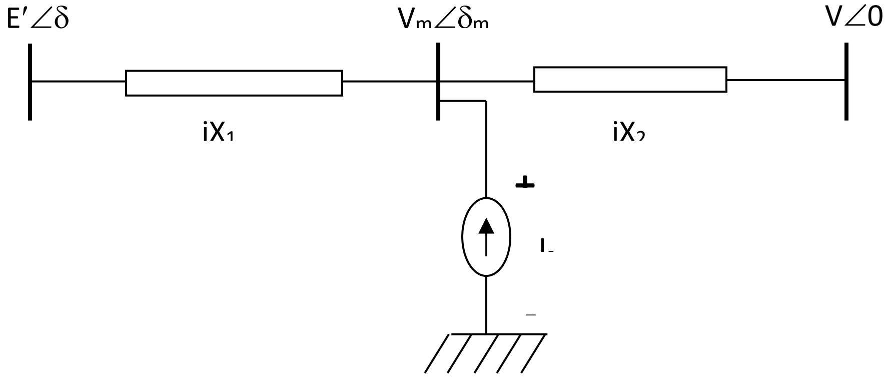


Fig 2 Equivalent Circuit of SMIB system with STATCOM

STATCOM can be represented by a shunt reactive current source I_s

$$I_s = I_s e^{j(\delta_m \pm \pi/2)} \qquad (1)$$

Here δ_m is the angle of voltage at bus m and is given by

$$\delta_m = \tan^{-1}\left(\frac{E'X_2 \sin\delta}{VX_1 + E'X_2\cos\delta}\right) \qquad (2)$$

$$V_m = \frac{E'X_2\cos(\delta-\delta_m) + VX_1\cos\delta_m}{X_1+X_2} + \frac{X_1X_2}{X_1+X_2} I_s \qquad (3)$$

Where, δ is the angle of the machine, X1 represents the equivalent reactance between the machine internal bus and the intermediate bus m, X2 represents the equivalent reactance between bus m and the infinite bus, E' is the machine internal voltage and V is the infinite bus voltage.

The above equations Eq.(2) and Eq.(3) indicates that the angle δ_m is independent of I_s but the bus voltage V_m depends upon I_s. The electrical power P_e of the machine can be written as

$$P_e = \frac{E'V_m}{X_1}\sin(\delta-\delta_m) \qquad (4)$$

Substituting the value of V_m and δ_m in Eq.(4)

$$P_e = \frac{E'}{X_1}\left[\frac{E'X_2\cos(\delta-\delta_m)+VX_1\cos\delta_m}{X_1+X_2}\right]\sin(\delta-\delta_m) + \frac{E'}{X_1}\left[\frac{X_1X_2}{X_1+X_2}\right]I_s\sin(\delta-\delta_m) \qquad (5)$$

Applying some basic circuit equations P_e written as

$P_e = P_{max}\sin\delta + f_1(\delta)I_s$ (6)

Where,

$$P_{max} = \frac{E'V}{X_1+X_2} \qquad (7)$$

$$f_1(\delta) = \frac{E'X_2}{X_1+X_2}\sin(\delta-\delta_m) \qquad (8)$$

Note that f_1 (δ) is positive when δ oscillates between zero and π Eq.(6) suggests that P_e can be modulated by controlling the STATCOM current I_s .it may be mentioned here that I_s in eq.(6) is positive (negative) for capacitive (inductive) operation of the STATCOM.

The speed of the machine ω is an appropriate control signal that can be used to enhancement of the power system damping. With the above control I_s can be expressed as

$I_s = K_1\omega$, $I_s^{min} \leq I_s \leq I_s^{max}$ (9)

Here, K1 is a positive gain and it depends upon the maximum current rating (I_s^{max}) of the STATCOM. From the mathematical model (Eqns.6, 7, 8 & 9).The simulation block diagram of the SMIB system with STATCOM is shown in the Fig.(3).

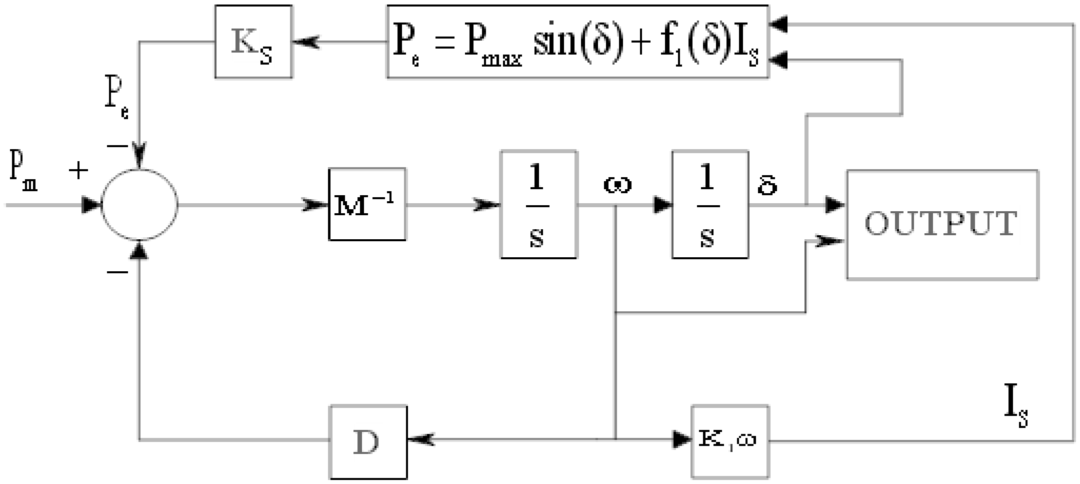


Fig 3 Simulation block diagram of SMIB system with STATCOM Controller

Using Eq.(6) and Eq.(9) -E can be written as

$$-E\ (\delta, \omega) = [D+K_1 f_1(\delta)]\ \omega^2 \qquad (10)$$

Here –E is considered as the rate of dissipation of transient energy. The first term within the square bracket of Eq.(10) is the natural damping coefficient D and the second term can be considered as additional damping coefficient (D_{STAT}) provided by the STATCOM.

$$D_{STAT} = K_1 f_1(\delta) = K_1 \frac{E' X_2}{X_1 + X_2} \sin(\delta - \delta_m) \qquad (11)$$

The above equations indicates that the value of D_{STAT} depends on the reactances (X_1,X_2) ,and hence the location of STATCOM. When the STATCOM is placed near the infinite bus ($X_2 \rightarrow 0$), D_{STAT} of Eq.(11) approaches to zero..On the other hand, when X_1 tends to zero, δ_m of Eq.(2) becomes almost the same as δ, and hence D_{STAT} is also approaches to zero. For a given δ, it can be shown that the maximum value of D_{STAT} can be obtained when the reactance ratio a_x=(X_1/X_2) becomes the same as the voltage ratio a_v= (E'/V).For such case, δ_m of Eq.(2) becomes δ/2, and thus the maximum damping coefficient provided by the STATCOM can be expressed as:

$$D_{STAT}^{max} = K_1 \frac{E'}{a+1} \sin(\frac{\delta}{2}) \qquad (12)$$

Here, $a=a_x=a_v$.

Data for the SMIB system:

Generator: X`=0.3 pu, f=60 Hz, D=0, H=5s (M=H/πf).

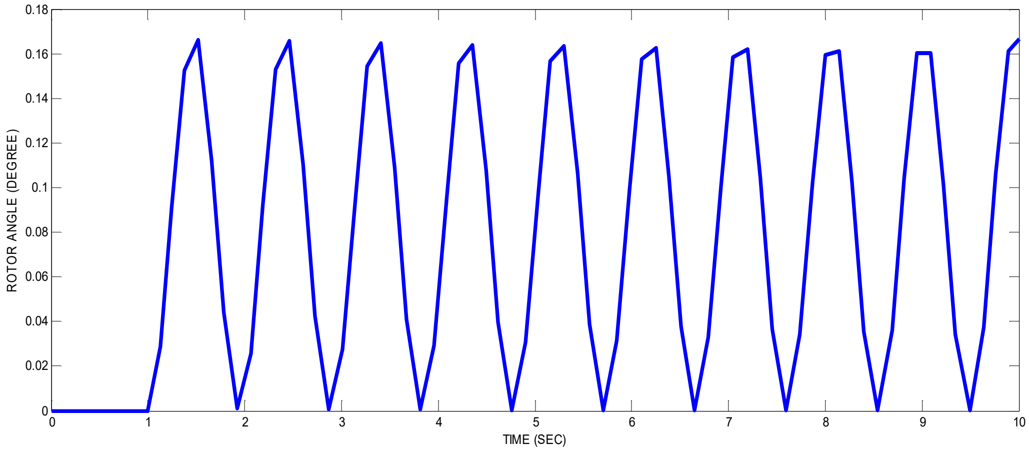
Transformer: X=0.1 pu.

Transmission lines: X=0.4 pu of each line.

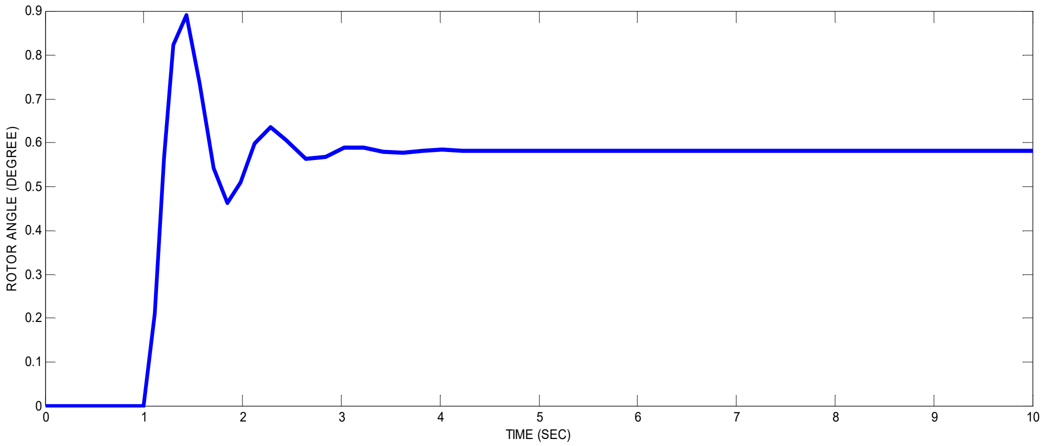
The generator initially delivers a power of 1.0 pu at a terminal voltage of 1.05 pu and the infinite bus voltage of 1.0 pu. The generator internal voltage E` for the above operating condition is found as 1.2356∠40.35° pu.

OUTPUT:

ROTOR ANGLE RESPONSE WITHOUT STATCOM:



ROTOR ANGLE RESPONSE WITH STATCOM:



INFERENCE:

From the output response we can understand that with the inclusion of STATCOM the SMIB system regains its rotor angle stability whereas without STATCOM the oscillations continued and rotor angle stability could not be regained.

RESULT:

The transient performance of the Single Machine Infinite Bus (SMIB) system with STATCOM was analysed using MATLAB.

ADITIONAL EXP NO:

DATE:

TRANSIENT STABILITY ANALYSIS OF MULTIMACHINE POWER SYSTEMS

AIM:

(i) To become familiar with modeling aspects of synchronous machines and network for transient stability analysis of multi-machine power systems.

(ii) To become familiar with the state-of-the-art algorithm for simplified transient stability simulation involving only classical machine models for synchronous machines.

(iii) To understand system behavior when subjected to large disturbances in the presence of synchronous machine controllers.

(iv) To become proficient in the usage of the software to tackle real life problems encountered in the areas of power system planning and operation.

OBJECTIVES

(i) To assess the transient stability of a multi machine power system when subjected to a common disturbance sequence: fault application on a transmission line followed by fault removal and line opening.

(ii) To determine the critical clearing time for the above sequence.

(iii)To observe system response and understand its behavior during a full load rejection at a substation with and without controllers.

(iv)To observe system response and understand its behavior during loss of a major generating station.

(v) To understand machine and system behavior during loss of excitation.

(vi)To study the effect of load relief provided by under frequency load shedding scheme.

THEORY:

The classical transient stability study is based on the application of a three – phase fault. A solid three – phase fault at bus k in the network results in $V_k = 0$. This is simulated by removing the kth row and column from the pre-fault bus admittance matrix. The new bus admittance matrix is reduced by eliminating all nodes except the internal generator nodes. The generator excitation voltages during the fault and post – fault modes are assumed to remain constant.

The electrical power of the ith generator in terms of the new reduced bus admittance matrices are obtained from

$$P_{ei} = \sum_{j=1}^{m} |E_i'| \, |E_j'| \, |Y_{ij}| \cos(\theta_{ij} - \delta_i + \delta_j) \quad \text{---------------}(1)$$

The swing equation with damping neglected, for machine i becomes, is given by

$$\frac{H_i}{\pi f_0} \frac{d^2\delta_i}{dt^2} = P_{mi} - \sum_{j=1}^{m} |E_i'| \, |E_j'| \, |Y_{ij}| \cos(\theta_{ij} - \delta_i + \delta_j) \quad \text{---------------}(2)$$

where Y_{ij} are the elements of the faulted reduced bus admittance matrix, and H_i is the inertia constant of machine i expressed on the common MVA base S_B. If H_{Gi} is the inertia constant of machine i expressed on the machine rated MVA S_{Gi}, then H_i is given by

$$H_i = (S_{Gi} / S_B)\, H_{Gi} \quad \text{-------------}(3)$$

Showing the electrical power of the ith generator by P_e^f and transforming equation (2) into state variable model yields

$$\frac{d\delta_i}{dt} = \Delta\omega_i, \quad i = 1,2, \ldots\ldots\ldots m \quad \text{----------------}(4)$$

$$\frac{d\Delta\omega_i}{dt} = (\pi f_0 / H_i)(P_m - P_e^f) \quad \text{------------}(5)$$

For multi-machine transient stability analysis of an interconnected power system, it is necessary to solve two state equations for each generator, with initial power angles δ_0 and $\Delta\omega_{0i} = 0$.

When the fault is cleared, which may involve the removal of the faulty line, the bus admittance matrix is recomputed to reflect the change in the network. Next the post- fault reduced bus admittance matrix is evaluated and the post- fault electrical power of the ith generator shown by P_i^{pf} is readily determined from (1).

Using the post-fault power P_i^{pf}, the simulation is continued to determine the system stability, until the plots reveal a definite trend as to stability or instability.

Usually the slack generator is selected as the reference, and the phase angle difference of all other generators with respect to the reference machine are plotted.

Usually, the solution is carried out for two swings to show that the second swing is not greater than the first one. If the angle differences do not increase, the system is stable. If any of the angle differences increase indefinitely, the system is unstable.

PROBLEM:

For bus 1, the voltage is given as V_1=1.06∟0 and it is taken as slack bus. The base value is 100MVA.

LOAD DATA		
BUS NO	LOAD	
	MW	Mvar
1	0	0
2	0	0
3	0	0
4	100	70
5	90	30
6	160	110

GENERATION SCHEDULE				
BUS NO	VOLTAGE MAG	GENERATION MW	Mvar LIMITS	
			Min	Max
1	1.06	------	-----	------
2	1.04	150	0	140
3	1.03	100	0	90

LINE DATA				
LINE NO (START)	LINE NO(END)	R(PU)	X(PU)	1/2B(PU)
1	4	0.035	0.225	0.0065
1	5	0.025	0.105	0.0045
1	6	0.040	0.215	0.0055
2	4	0.000	0.035	0.0000
3	5	0.000	0.042	0.0000
4	6	0.028	0.125	0.0035
5	6	0.026	0.175	0.0300

MACHINE DATA			
GEN	Ra	Xd’	H
1	0	0.20	20
2	0	0.15	4
3	0	0.25	5

A three phase occurs on line 5-6 near bus 6 and is cleared by the simultaneous opening of breakers at both ends of the line. Perform the transient stability analysis and determine the system stability for a) when the fault is cleared in 0.4 second b) when the fault is cleared in 0.5 second c) Repeat the simulation to determine the critical clearing angle

Manual Calculation:

PROGRAM:

```
basemva = 100;  accuracy = 0.0001;  maxiter = 10;
busdata=[1  1  1.06   0.0  00.00    00.00    0.00    00.00     0     0   0
         2  2  1.04   0.0  00.00    00.00  150.00    00.00     0   140   0
         3  2  1.03   0.0  00.00    00.00  100.00    00.00     0    90   0
         4  0  1.0    0.0 100.00    70.00   00.00    00.00     0     0   0
         5  0  1.0    0.0  90.00    30.00   00.00    00.00     0     0   0
         6  0  1.0    0.0 160.00   110.00   00.00    00.00     0     0   0];

linedata=[1  4  0.035   0.225   0.0065   1.0
          1  5  0.025   0.105   0.0045   1.0
          1  6  0.040   0.215   0.0055   1.0
          2  4  0.000   0.035   0.0000   1.0
          3  5  0.000   0.042   0.0000   1.0
          4  6  0.028   0.125   0.0035   1.0
          5  6  0.026   0.175   0.0300   1.0];

lfybus                    % form the bus admittance matrix
lfnewton      % Power flow solution by Newton-Raphson method
busout       % Prints the power flow solution on the screen

%        Gen.  Ra   Xd'   H
gendata=[ 1     0   0.20   20
          2     0   0.15    4
          3     0   0.25    5];
trstab
```

trstab

```
global Pm f H E  Y th ngg
f=60;
%zdd=gendata(:,2)+j*gendata(:,3);
ngr=gendata(:,1);
```

```
%H=gendata(:,4);
ngg=length(gendata(:,1));
%%
for k=1:ngg
zdd(ngr(k))=gendata(k, 2)+j*gendata(k,3);
%H(ngr(k))=gendata(k, 4);
H(k)=gendata(k,4);   % new
end
%%
for k=1:ngg
I=conj(S(ngr(k)))/conj(V(ngr(k)));
%Ep(ngr(k)) = V(ngr(k))+zdd(ngr(k))*I;
%Pm(ngr(k))=real(S(ngr(k)));
Ep(k) = V(ngr(k))+zdd(ngr(k))*I;  % new
Pm(k)=real(S(ngr(k)));            % new

end
E=abs(Ep); d0=angle(Ep);
for k=1:ngg
nl(nbr+k) = nbus+k;

nr(nbr+k) = gendata(k, 1);

%R(nbr+k)  = gendata(k, 2);
%X(nbr+k)  = gendata(k, 3);

R(nbr+k)  = real(zdd(ngr(k)));
X(nbr+k)  = imag(zdd(ngr(k)));

Bc(nbr+k)  = 0;
a(nbr+k) = 1.0;
yload(nbus+k)=0;
end
nbr1=nbr; nbus1=nbus;
nbrt=nbr+ngg;
nbust=nbus+ngg;
linedata=[nl, nr, R, X, -j*Bc, a];
[Ybus, Ybf]=ybusbf(linedata, yload, nbus1,nbust);
fprintf('\nPrefault reduced bus admittance matrix \n')
Ybf
Y=abs(Ybf); th=angle(Ybf);
Pm=zeros(1, ngg);
disp(['      G(i)    E''(i)     d0(i)      Pm(i)'])
for ii = 1:ngg
for jj = 1:ngg
Pm(ii) = Pm(ii) + E(ii)*E(jj)*Y(ii, jj)*cos(th(ii, jj)-d0(ii)+d0(jj));

end,
fprintf('       %g', ngr(ii)), fprintf('   %8.4f',E(ii)), fprintf('   %8.4f',
180/pi*d0(ii))
fprintf('  %8.4f \n',Pm(ii))
end
respfl='y';
while respfl =='y' | respfl=='Y' nf=input('Enter
faulted bus No. -> '); fprintf('\nFaulted reduced
bus admittance matrix\n')
```

```
Ydf=ybusdf(Ybus, nbus1, nbust, nf)
%Fault cleared
[Yaf]=ybusaf(linedata, yload, nbus1,nbust, nbrt);
fprintf('\nPostfault reduced bus admittance matrix\n')
Yaf
resptc='y';
while resptc =='y' | resptc=='Y'
tc=input('Enter clearing time of fault in sec. tc = ');
tf=input('Enter final simulation time in sec.  tf = ');
clear t  x  del
t0 = 0;
w0=zeros(1, length(d0));
x0 = [d0,  w0];
tol=0.0001;
Y=abs(Ydf); th=angle(Ydf);
%[t1, xf] =ode23('dfpek', t0, tc, x0, tol);  % Solution during fault (use with
MATLAB 4)
tspan=[t0, tc];
[t1, xf] =ode23('dfpek', tspan, x0);  % Solution during fault (use with MATLAB 5)
x0c =xf(length(xf), :);
Y=abs(Yaf); th=angle(Yaf);
%[t2,xc] =ode23('afpek', tc, tf, x0c, tol); % Postfault solution (use with MATLAB
4)
tspan = [tc, tf];                           % use with MATLAB 5
[t2,xc] =ode23('afpek', tspan, x0c);        % Postfault solution (use with MATLAB
5)
t =[t1; t2]; x = [xf; xc];
fprintf('\nFault is cleared at %4.3f Sec. \n', tc)
for k=1:nbus
    if kb(k)==1
    ms=k; else, end
end
fprintf('\nPhase angle difference of each machine \n')
fprintf('with respect to the slack in degree.\n')
fprintf('   t - sec')
kk=0;
for k=1:ngg
    if k~=ms
    kk=kk+1;
    del(:,kk)=180/pi*(x(:,k)-x(:,ms));
    fprintf('    d(%g,',ngr(k)), fprintf('%g)', ngr(ms))
    else, end
end
fprintf(' \n')
disp([t, del])
h=figure; figure(h)
plot(t, del)
title(['Phase angle difference (fault cleared at ', num2str(tc),'s)'])
xlabel('t, sec'), ylabel('Delta, degree'), grid
   resp=0;
   while strcmp(resp, 'n')~=1 & strcmp(resp, 'N')~=1 & strcmp(resp, 'y')~=1 &
strcmp(resp, 'Y')~=1
   resp=input('Another clearing time of fault? Enter ''y'' or ''n'' within quotes
-> ');

   if strcmp(resp, 'n')~=1 & strcmp(resp, 'N')~=1 & strcmp(resp, 'y')~=1 &
strcmp(resp, 'Y')~=1
```

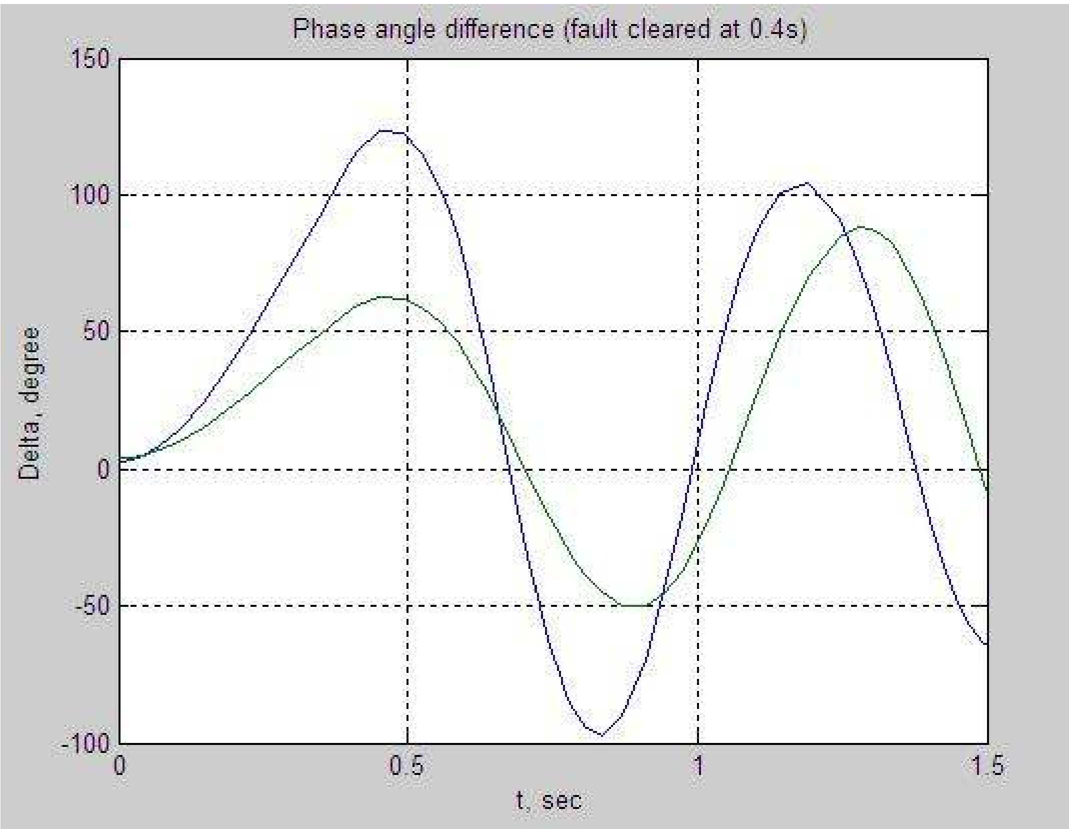
```
   fprintf('\n Incorrect reply, try again \n\n'), end
   end
resptc=resp;

end
    resp2=0;
    while strcmp(resp2, 'n')~=1 & strcmp(resp2, 'N')~=1 & strcmp(resp2, 'y')~=1 &
strcmp(resp2, 'Y')~=1
    resp2=input('Another fault location: Enter ''y'' or ''n'' within quotes -> ');
    if strcmp(resp2, 'n')~=1 & strcmp(resp2, 'N')~=1 & strcmp(resp2, 'y')~=1 &
strcmp(resp2, 'Y')~=1
    fprintf('\n Incorrect reply, try again \n\n'), end
    respf1=resp2;
    end
    if respf1=='n' | respf1=='N', return, else, end
end
```

OUTPUT:

```
                  Power Flow Solution by Newton-Raphson Method

                     Maximum Power Mismatch = 1.80187e-007

                            No. of Iterations = 4

    Bus  Voltage  Angle    ------Load------    ---Generation---   Injected

    No.  Mag.     Degree     MW       Mvar       MW       Mvar       Mvar

     1   1.060    0.000     0.000     0.000   105.287   107.335     0.000

     2   1.040    1.470     0.000     0.000   150.000    99.771     0.000

     3   1.030    0.800     0.000     0.000   100.000    35.670     0.000

     4   1.008   -1.401   100.000    70.000     0.000     0.000     0.000

     5   1.016   -1.499    90.000    30.000     0.000     0.000     0.000

     6   0.941   -5.607   160.000   110.000     0.000     0.000     0.000

    Total                 350.000   210.000   355.287   242.776     0.000

            Prefault reduced bus admittance matrix
```

```
        Ybf =

          0.3517 - 2.8875i   0.2542 + 1.1491i   0.1925 + 0.9856i
          0.2542 + 1.1491i   0.5435 - 2.8639i   0.1847 + 0.6904i
          0.1925 + 0.9856i   0.1847 + 0.6904i   0.2617 - 2.2835i

            G(i)    E'(i)     d0(i)      Pm(i)
             1     1.2781     8.9421    1.0529
             2     1.2035    11.8260    1.5000
             3     1.1427    13.0644    1.0000

Enter faulted bus No. -> 6

Faulted reduced bus admittance matrix

Ydf =

   0.1913 - 3.5849i   0.0605 + 0.3644i   0.0523 + 0.4821i

   0.0605 + 0.3644i   0.3105 - 3.7467i   0.0173 + 0.1243i

   0.0523 + 0.4821i   0.0173 + 0.1243i   0.1427 - 2.6463i Fault

is cleared by opening a line. The bus to bus nos. of the line to

be removed must be entered within brackets, e.g. [5, 7] Enter

the bus to bus Nos. of line to be removed -> [5,6] Postfault

reduced bus admittance matrix

Yaf =

   0.3392 - 2.8879i   0.2622 + 1.1127i   0.1637 + 1.0251i

   0.2622 + 1.1127i   0.6020 - 2.7813i   0.1267 + 0.5401i

   0.1637 + 1.0251i   0.1267 + 0.5401i   0.2859 - 2.0544i

Enter clearing time of fault in sec. tc = 0.4

Enter final simulation time in sec.  tf = 1.5

        Fault is cleared at 0.400 Sec.
```

```
Phase angle difference of each machine
with respect to the slack in degree.
  t - sec    d(2,1)    d(3,1)
        0    2.8839    4.1224
   0.0000    2.8839    4.1224
   0.0000    2.8839    4.1224
   0.0001    2.8839    4.1224
   0.0003    2.8840    4.1224
   0.0015    2.8862    4.1235
```



```
   0.0073    2.9399    4.1509
   0.0224    3.4099    4.3908
```

0.0444	4.9406	5.1720
0.0723	8.2866	6.8808
0.1060	14.2907	9.9504
0.1460	23.8359	14.8373
0.1860	35.4934	20.8099
0.2260	48.6576	27.5407
0.2660	62.7474	34.6877
0.3060	77.2873	41.9274
0.3460	91.9675	48.9846
0.3860	106.6754	55.6506
0.4000	111.8343	57.8619
0.4000	111.8343	57.8619
0.4117	115.7711	59.5250
0.4514	123.3461	62.7084
0.4910	122.4353	62.1741
0.5246	114.9564	58.8080
0.5571	101.5237	53.0062
0.5707	93.9706	49.8307
0.5843	85.2511	46.2219
0.5992	74.3703	41.7856
0.6151	61.2586	36.5099
0.6315	46.3109	30.5474
0.6484	29.6213	23.8982
0.6665	10.9104	16.3713
0.6874	-11.0636	7.2886

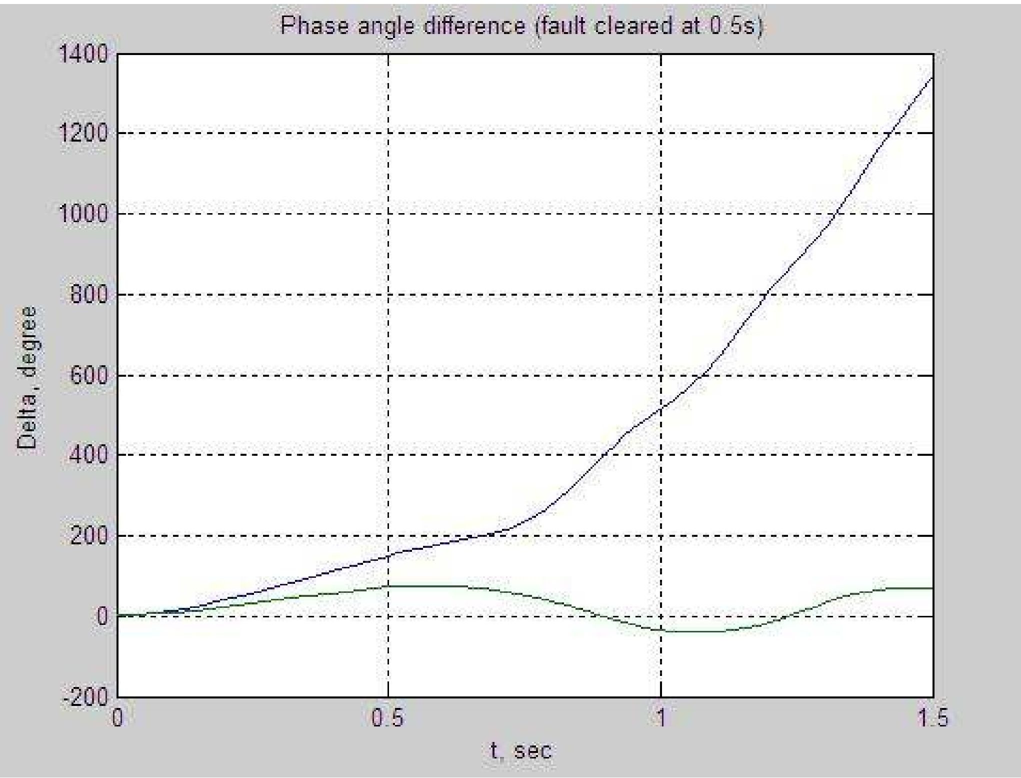
0.7076	-31.4452	-1.5650
0.7277	-50.0888	-10.2773
0.7453	-64.3270	-17.5845
0.7605	-74.7647	-23.5451
0.7728	-81.8172	-28.0628
0.7869	-88.2825	-32.8413
0.8064	-94.3483	-38.6949
0.8333	-97.0931	-45.0843
0.8705	-90.1891	-50.2025
0.9099	-70.0025	-50.2533
0.9431	-44.2898	-45.5321
0.9756	-14.0114	-36.4903
1.0106	19.9645	-22.2399
1.0438	49.4313	-5.4158
1.0726	70.6234	10.6518
1.1035	87.7813	28.1804
1.1415	100.5913	48.4869
1.1896	103.9511	69.7127
1.2444	90.9792	84.8240
1.2773	75.0434	88.1302
1.2974	62.6822	87.7623
1.3174	48.6144	85.5064
1.3345	35.5322	82.0560
1.3498	23.3208	77.8011
1.3651	10.8942	72.4688

```
    1.3850   -5.1516   63.9678

    1.4059  -21.0700   53.3371

    1.4268  -35.3195   41.1849

    1.4478  -47.3380   27.7068

    1.4706  -57.1182   12.1591

    1.4960  -63.5779   -5.6903

    1.5000  -64.1454   -8.4713


Another clearing time of fault? Enter 'y' or 'n' within quotes -> 'y'

Enter clearing time of fault in sec. tc = 0.5

Enter final simulation time in sec.  tf = 1.5

Fault is cleared at 0.500 Sec.

Phase angle difference of each machine

with respect to the slack in degree.

   t - sec    d(2,1)    d(3,1)

  1.0e+003 *

         0    0.0029    0.0041

    0.0000    0.0029    0.0041

    0.0000    0.0029    0.0041

    0.0000    0.0029    0.0041

    0.0000    0.0029    0.0041

    0.0000    0.0029    0.0041

    0.0000    0.0029    0.0042

    0.0000    0.0034    0.0044

    0.0000    0.0049    0.0052
```

0.0001	0.0083	0.0069
0.0001	0.0143	0.0100
0.0001	0.0240	0.0149
0.0002	0.0387	0.0225
0.0002	0.0557	0.0311
0.0003	0.0737	0.0402
0.0003	0.0920	0.0490
0.0004	0.1104	0.0573
0.0004	0.1291	0.0647
0.0005	0.1503	0.0716
0.0005	0.1503	0.0716
0.0005	0.1572	0.0734
0.0006	0.1738	0.0754
0.0006	0.1861	0.0730
0.0007	0.1997	0.0671
0.0007	0.2163	0.0593
0.0008	0.2389	0.0500
0.0008	0.2674	0.0400
0.0008	0.2889	0.0331
0.0008	0.3137	0.0254
0.0008	0.3362	0.0185
0.0009	0.3570	0.0120
0.0009	0.3789	0.0052
0.0009	0.4034	-0.0027
0.0009	0.4265	-0.0101

0.0009	0.4457	-0.0162
0.0009	0.4645	-0.0219
0.0010	0.4843	-0.0274
0.0010	0.5072	-0.0327
0.0010	0.5374	-0.0374
0.0011	0.5729	-0.0397
0.0011	0.6061	-0.0399
0.0011	0.6451	-0.0385
0.0011	0.6853	-0.0355
0.0012	0.7332	-0.0298
0.0012	0.7731	-0.0231
0.0012	0.8105	-0.0148
0.0012	0.8456	-0.0052
0.0012	0.8834	0.0067
0.0013	0.9326	0.0226
0.0013	0.9720	0.0340
0.0013	1.0159	0.0446
0.0014	1.0606	0.0532
0.0014	1.1117	0.0606
0.0014	1.1624	0.0657
0.0014	1.2046	0.0684
0.0014	1.2445	0.0698
0.0015	1.2883	0.0704
0.0015	1.3322	0.0702
0.0015	1.3394	0.0701

```
Another clearing time of fault? Enter 'y' or 'n' within quotes -> 'n'

Another fault location: Enter 'y' or 'n' within quotes -> 'n'
```



RESULT:

Thus the multi-machine transient stability analysis is simulated on a given power system network.

Ex.No-11 **POWER FLOW USING FAST DECOUPLED LOAD FLOW METHOD**

Date:

AIM

To compute the program to find the load flow equations and solutions using FDLF method.

SOFTWARE REQUIRED

MATLAB/ETAP/Power world simulator

PREREQUISITE QUESTIONS

1. Recall the control loops available in a synchronous generator.
2. State the relationship between reactive power delivered and terminal voltage of an alternator.
3. Differentiate generator bus and slack bus

ALGORITHM

Step 1: Read the system data.

Step 2: Initialize all the bus voltages.

Step 3: Form admittance matrix.

Step 4: Form B’, B" and then invert them.

Step 5: Compute $\Delta P = P^{spec} - P^{calc}$

Step 6: Check for convergence. If converged, set $P_{conv} = 1$, and go to step 10.

Step 7: Form Jacobian matrix H.

Step 8: If not converged, find $[\Delta\delta]$.

Step 9: Update δ value.

Step 10: Compute ΔQ.

Step 11: Check for convergence. If converged set $Q_{conv} = 1$, and go to step 15.

Step 12: Form [L].

Step 13: If not converged, find $[\Delta V]$.

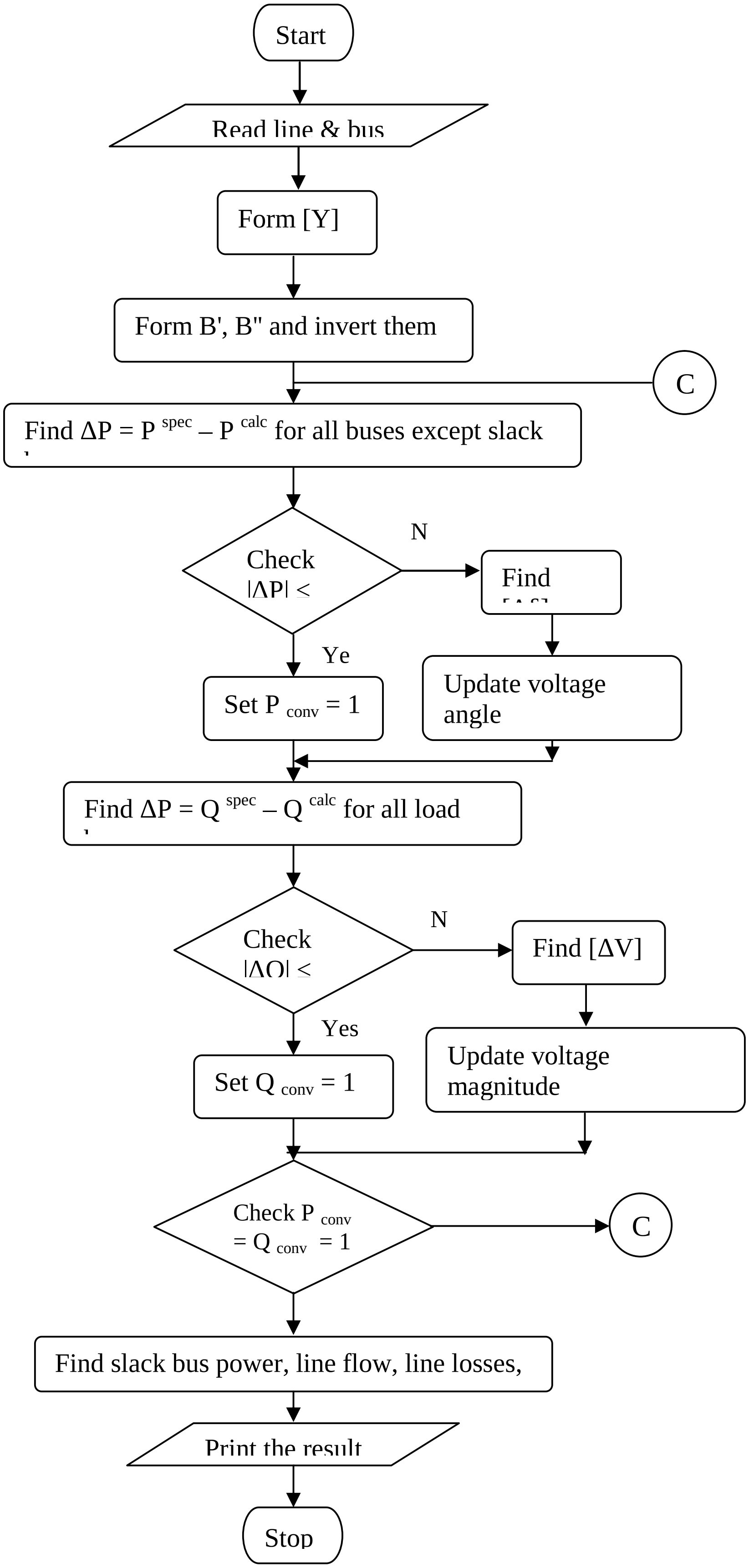
Step 14: Update $|V|$ value.

Step 15: If $P_{conv} = 1$ and $Q_{conv} = 1$, then solution is obtained and print the result.

Step 16: Else go to step 5.

Step 17: Stop.

FLOWCHART


Start
Read line & bus
Form [Y]
Form B', B" and invert them
C
Find ΔP = P spec – P calc for all buses except slack
N
Check
|ΔP| <
Find
Ye
Set P conv = 1
Update voltage angle
Find ΔP = Q spec – Q calc for all load
N
Check
|ΔQ| <
Find [ΔV]
Yes
Set Q conv = 1
Update voltage magnitude
Check P conv = Q conv = 1
C
Find slack bus power, line flow, line losses,
Print the result
Stop

PROBLEM STATEMENT

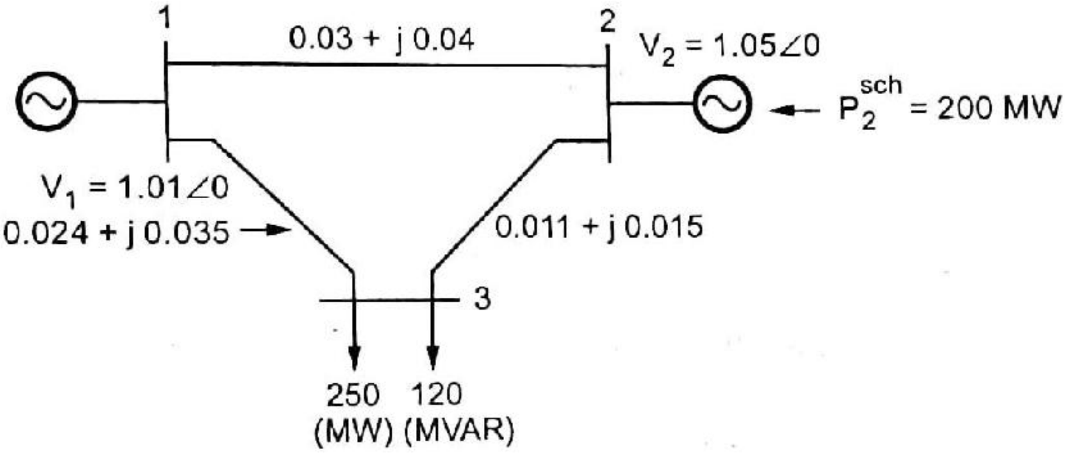
Bus data in p.u.

Bus Number	Type	Generator (MW)		Load (MW)		Voltage	Angle	Reactive Power Limit	
		P_G	Q_G	P_D	Q_D			Q_{min}	Q_{max}
1	Slack	0	0	0	0	1.06	0	0	0
2	PQ	0	0	270	130	1.0	0	0	0
3	PV	230	0	0	0	1.04	0	0	230

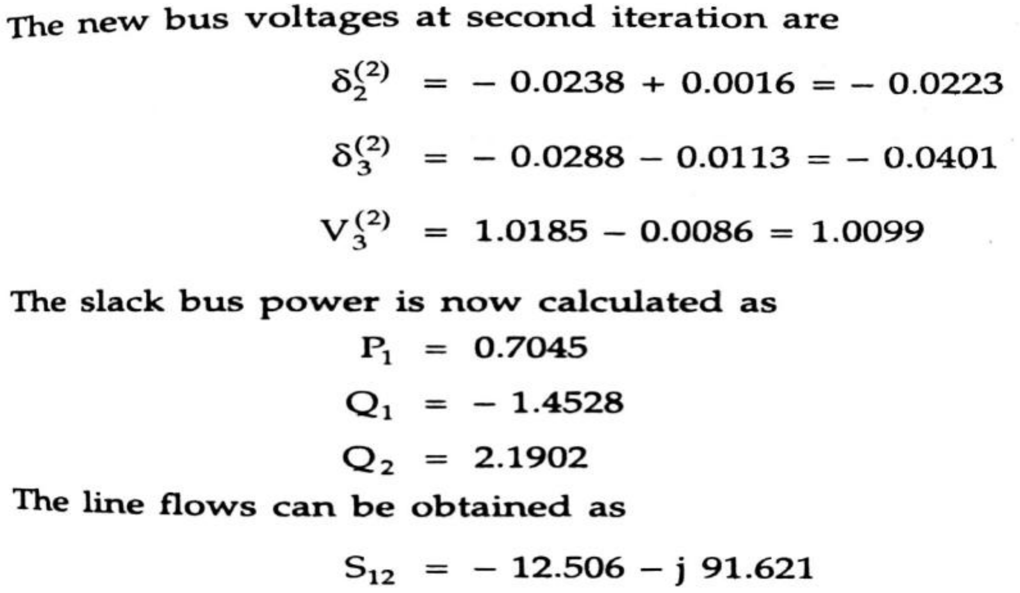
Line data in p.u.

From	To	R p.u.	X p.u.	½ B1 p.u.	Tap Position
1	2	0.03	0.04	0	1
2	3	0.011	0.015	0	1
1	3	0.024	0.035	0	1

Base MVA: 100MVA



SOLUTION:



The new bus voltages at second iteration are

$$\delta_2^{(2)} = -0.0238 + 0.0016 = -0.0223$$

$$\delta_3^{(2)} = -0.0288 - 0.0113 = -0.0401$$

$$V_3^{(2)} = 1.0185 - 0.0086 = 1.0099$$

The slack bus power is now calculated as

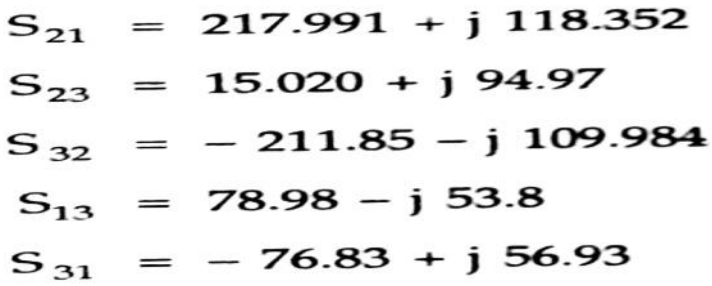
$$P_1 = 0.7045$$

$$Q_1 = -1.4528$$

$$Q_2 = 2.1902$$

The line flows can be obtained as

$$S_{12} = -12.506 - j\,91.621$$



$$S_{21} = 217.991 + j\,118.352$$

$$S_{23} = 15.020 + j\,94.97$$

$$S_{32} = -211.85 - j\,109.984$$

$$S_{13} = 78.98 - j\,53.8$$

$$S_{31} = -76.83 + j\,56.93$$

VIVA QUESTIONS:

1. Indicate the Jacobian J2 and J3 of a network 1 slack bus and 1 generator bus and 2 load buses using FDLF method
2. Indicate the reason for Jacobian sparsity in FDLF method.

RESULT