Bapuji Educational Association (Regd)

Bapuji Institute of Engineering and Technology

Davanagere-577004

Department of Electrical and Electronics Engineering

Manual for

Power System Simulation Lab

[Semester: 7th ; Subject Code: 15EEL76]

15EEL76 Power System Simulation Laboratory

Subject Code: 15EEL76 **IA Marks: 20**
No. of Practical Hrs. / Week: 03 **Exam Hours: 03**
Total No. of Practical Hrs: 42 **Exam Marks: 80**

Power system simulation using MATLAB/ C or C ++ /Sci lab /octave

1. a) Formation for symmetric π configuration for Verification of $AD-BC$=1, Determination of Efficiency and regulation.
   b) Formation for symmetric T configuration for Verification of $AD-BC$=1, Determination of Efficiency and regulation.
2. Determination of Power Angle Diagrams, Reluctance Power, Excitation, Emf and Regulation for Salient and Non-Salient Pole Synchronous Machines
3. To obtain Swing Curve and to Determine Critical Clearing Time, Regulation, Inertia Constant/Line Parameters /Fault Location/Clearing Time/Pre-Fault Electrical Output for a Single Machine connected to Infinite Bus through a Pair of identical Transmission Lines Under 3-Phase Fault On One of the two Lines.
4. Y Bus Formation for Power Systems with and without Mutual Coupling, by Singular Transformation and Inspection Method
5. Formation of Z Bus(without mutual coupling) using Z-Bus Building Algorithm.
6. Determination of Bus Currents, Bus Power and Line Flow for a Specified System Voltage (Bus) Profile.
7. Formation of Jacobian for a System not Exceeding 4 Buses (No PV Buses) in Polar Coordinates.
8. Load Flow Analysis using Gauss Siedel Method, NR Method and Fast Decoupled Method for Both PQ and PV Buses.
9. To Determine Fault Currents and Voltages in a Single Transmission Line System With Star-Delta Transformers at a Specified Location for LG and LLG faults by simulation
10. Optimal Generation Scheduling for Thermal power plants.

Note: Questions 1-6: Simulation Experiments using MATLAB/C or C++/Scilab/Octave
Questions 7-10: Use suitable standard software package.

****

Program 1 a): ABCD of equivalent PI Network

```
clc
clear all
z=0.2+0.408i; y=0+3.14e-6i;
k1=input('\n Enter 1-for short line 2-for medium line 3-for long line : ');
switch k1
    case 1,
        length=40;
        Z=z*length; Y=y*length;
        A=1; B=Z; C=0; D=1;
    case 2,
        length=140;
        Z=z*length; Y=y*length;
        A=1+Y*Z/2;
        B=Z;
        C=Y*(1+Y*Z/4);
        D=A;
    case 3,
        length=300;
        zc=sqrt(z/y);
        gam=sqrt(z*y)*length;
        A=cosh(gam);
        D=A;
        B=zc*sinh(gam);
        C=1/zc*sinh(gam);
        fprintf('\n the equvalent PI circuit constants:');
        zeq=z*length*sinh(gam)/gam;
        yeq=y*length/2*tanh(gam/2)/(gam/2);
        fprintf('\n Zeq=%10.4f %+15.4fi', real(zeq), imag(zeq));
        fprintf('\n Yeq=%10.4f %+15.4fi', real(yeq), imag(yeq));
    otherwise
        disp('wrong choice of tr.line');
end
fprintf('\n A,B,C and D constants:\n');
fprintf('-----------------------');
fprintf('\nA=%10.4f %+10.4fi',real(A), imag(A));
fprintf('\nA=%10.4f %+10.4fi',real(B), imag(B));
fprintf('\nA=%10.4f %+10.4fi',real(C), imag(C));
fprintf('\nA=%10.4f %+10.4fi',real(D), imag(D));
fprintf('\n the product AD-BC=%f', A*D-B*C);
k2=input('\n Enter 1-To read Vr, Ir and compute Vs, Is\n    2-To read Vs, Is and compute Vr, Ir :’ );
switch k2,
```

```
    case 1,
    %vr=input('enter Vr/phase');
     %ir=input('enter Ir/phase');
     vrl=132+0.0i;
     ir=174.96-131.22i;
     vr=vrl*1e3/sqrt(3);
     vs=(A*vr+B*ir);
     is=C*vr+D*ir;
     fprintf('\n sending end voltage/ph=%f %+fi KV', real(vs)/1e3, imag(vs)/1e3);
     fprintf('\n sending end current/ph=%f %+fi AMP', real(is), imag(is));
     vsl=vs*sqrt(3.0);
    case 2,
     %vs=input('enter Vs/phase');
     %is=input('enter Is/phase');
     vsl=132+0.0i;
     is=174.96-131.22i;
     vs=vsl*1e3/sqrt(3);
     vr=(D*vs-B*is);
     ir=-C*vs+D*is;
     fprintf('\n Receiving end voltage/ph=%f %+fi KV', real(vr)/1e3, imag(vr)/1e3);
     fprintf('\n sending end current/ph=%f %+fi AMP', real(ir), imag(ir));
     vrl=vr*sqrt(3.0);
    otherwise
        disp('wrong choice');
end
rec_pow=3*real(vr*conj(ir))/1e6;
send_pow=3*real(vs*conj(is))/1e6;
eff=rec_pow/send_pow*100;
reg=(abs(vs)/abs(A)-abs(vr))/abs(vr)*100;
fprintf('\n Receiving end power=%.2f MW', rec_pow);
fprintf('\n Sending end power=%.2f MW', send_pow);
fprintf('\n Efficiency=%.2f %%',eff);
fprintf('\n Voltage Regulation=%.2f%%', reg);
```

O/P 1:-

```
Enter 1-for short line 2-for medium line 3-for long line:1

  A,B,C and D constants:
----------------
A=      1.0000    +0.0000i
B=      8.0000   +16.3200i
C=      0.0000    +0.0000i
D=      1.0000    +0.0000i
  the product AD-BC=1.000000
  Enter 1-To read Vr,Ir and compute Vs,Is
  2-To read Vs,Is and compute Vr,Ir:1

  sending end voltage/ph=79.751426+1.805587i KV
  sending end current/ph=174.960000-131.220000i AMP
  Receiving end power=40.00 MW
  Sending end power=41.15 MW
  Effiency=97.21 %
  Voltage regulation=4.67 %>>
```

O/P 2:-

```
Enter 1-for short line 2-for medium line 3-for long line:1

  A,B,C and D constants:
----------------
A=      1.0000    +0.0000i
B=      8.0000   +16.3200i
C=      0.0000    +0.0000i
D=      1.0000    +0.0000i
  the product AD-BC=1.000000
  Enter 1-To read Vr,Ir and compute Vs,Is
  2-To read Vs,Is and compute Vr,Ir:1

  sending end voltage/ph=79.751426+1.805587i KV
  sending end current/ph=174.960000-131.220000i AMP
  Receiving end power=40.00 MW
  Sending end power=41.15 MW
  Effiency=97.21 %
  Voltage regulation=4.67 %>>
```

O/P 3:-

```
Enter 1-for short line 2-for medium line 3-for long line:2

  A,B,C and D constants:
----------------
A=      0.9874     +0.0062i
B=     28.0000    +57.1200i
C=     -0.0000     +0.0004i
D=      0.9874     +0.0062i
  the product AD-BC=1.000000
  Enter 1-To read Vr,Ir and compute Vs,Is
  2-To read Vs,Is and compute Vr,Ir:1

  sending end voltage/ph=87.647584+6.788583i KV
  sending end current/ph=173.467869-95.204051i AMP
  Receiving end power=40.00 MW
  Sending end power=43.67 MW
  Effiency=91.59 %
  Voltage regulation=16.82 %>>
```

O/P 4:-

```
  Enter 1-for short line 2-for medium line 3-for long line:2

  A,B,C and D constants:
----------------
A=      0.9874     +0.0062i
B=     28.0000    +57.1200i
C=     -0.0000     +0.0004i
D=      0.9874     +0.0062i
  the product AD-BC=1.000000
  Enter 1-To read Vr,Ir and compute Vs,Is
  2-To read Vs,Is and compute Vr,Ir:2

  receiving end voltage/ph=62.859251-5.850527i KV
  sending end current/ph=173.674054-161.787473i AMP
  Receiving end power=35.59 MW
  Sending end power=40.00 MW
  Effiency=88.97 %
  Voltage regulation=22.25 %>>
```

O/P 5:-

```
  Enter 1-for short line 2-for medium line 3-for long line:3
  the equivalent PI circuit constants:
  Zeq=      57.7123            +120.6169i
  Yeq=        0.0000                 +0.0005i
  A,B,C and D constants:
----------------
A=        0.9428      +0.0277i
B=      57.7123 +120.6169i
C=      -0.0000      +0.0009i
D=        0.9428      +0.0277i
  the product AD-BC=1.000000
  Enter 1-To read Vr,Ir and compute Vs,Is
  2-To read Vs,Is and compute Vr,Ir:1

  sending end voltage/ph=97.773395+15.642655i KV
  sending end current/ph=167.915908-48.443896i AMP
  Receiving end power=40.00 MW
  Sending end power=46.98 MW
  Effiency=85.15 %
  Voltage regulation=37.75 %>>
```

O/P6:-

```
  Enter 1-for short line 2-for medium line 3-for long line:3

  the equivalent PI circuit constants:
  Zeq=      57.7123            +120.6169i
  Yeq=        0.0000                 +0.0005i
  A,B,C and D constants:
----------------
A=        0.9428      +0.0277i
B=      57.7123 +120.6169i
C=      -0.0000      +0.0009i
D=        0.9428      +0.0277i
  the product AD-BC=1.000000
  Enter 1-To read Vr,Ir and compute Vs,Is
  2-To read Vs,Is and compute Vr,Ir:2

  receiving end voltage/ph=45.924027-11.417589i KV
  sending end current/ph=169.252896-189.276902i AMP
  Receiving end power=29.80 MW
  Sending end power=40.00 MW
  Effiency=74.50 %
  Voltage regulation=70.75 %>
```

Program 1 b): ABCD of equivalent T Network

```
clc
clear all
z=0.2+0.408i; y=0+3.14e-6i;
k1=input('\n Enter 1-for short line 2-for medium line 3-for long line : ');
switch k1
    case 1,
        length=40;
        Z=z*length; Y=y*length;
        A=1; B=Z; C=0; D=1;
    case 2,
        length=140;
        Z=z*length; Y=y*length;
        A=1+Y*Z/2;
        B=Z*(1+Y*Z/4);
        C=Y;
        D=1+Y*Z/2;
    case 3,
        length=300;
        zc=sqrt(z/y);
        gam=sqrt(z*y)*length;
        A=cosh(gam);
        D=A;
        B=zc*sinh(gam);
        C=1/zc*sinh(gam);
        fprintf('\n the equvalent T circuit constants:');
        zeq=z*length/2*tanh(gam/2)/(gam/2);
        yeq=y*length*sinh(gam)/gam;
        fprintf('\n Zeq=%10.4f %+15.4fi', real(zeq), imag(zeq));
        fprintf('\n Yeq=%10.4f %+15.4fi', real(yeq), imag(yeq));
    otherwise
        disp('wrong choice of tr.line');
end
fprintf('\n A,B,C and D constants:\n');
fprintf('-----------------------');
fprintf('\nA=%10.4f %+10.4fi',real(A), imag(A));
fprintf('\nA=%10.4f %+10.4fi',real(B), imag(B));
fprintf('\nA=%10.4f %+10.4fi',real(C), imag(C));
fprintf('\nA=%10.4f %+10.4fi',real(D), imag(D));
fprintf('\n the product AD-BC=%f', A*D-B*C);
k2=input('\n Enter 1-To read Vr, Ir and compute Vs, Is\n    2-To read Vs, Is and compute Vr,
Ir :');
switch k2,
```

```
        case 1,
        %vr=input('enter Vr/phase');
          %ir=input('enter Ir/phase');
          vrl=132+0.0i;
          ir=174.96-131.22i;
          vr=vrl*1e3/sqrt(3);
          vs=(A*vr+B*ir);
          is=(C*vr+D*ir);
          fprintf('\n sending end voltage/ph=%f %+fi KV', real(vs)/1e3, imag(vs)/1e3);
          fprintf('\n sending end current/ph=%f %+fi AMP', real(is), imag(is));
          vsl=vs*sqrt(3.0);
        case 2,
          %vs=input('enter Vs/phase');
          %is=input('enter Is/phase');
          vsl=132+0.0i;
          is=174.96-131.22i;
          vs=vsl*1e3/sqrt(3);
          vr=(D*vs-B*is);
          ir=(-C*vs+D*is);
          fprintf('\n Receiving end voltage/ph=%f %+fi KV', real(vr)/1e3, imag(vr)/1e3);
          fprintf('\n sending end current/ph=%f %+fi AMP', real(ir), imag(ir));
          vrl=vr*sqrt(3.0);
        otherwise
                disp('wrong choice');
end
rec_pow=3*real(vr*conj(ir))/1e6;
send_pow=3*real(vs*conj(is))/1e6;
eff=rec_pow/send_pow*100;
reg=(abs(vs)/abs(A)-abs(vr))/abs(vr)*100;
fprintf('\n Receiving end power=%.2f MW', rec_pow);
fprintf('\n Sending end power=%.2f MW', send_pow);
fprintf('\n Efficiency=%.2f %%',eff);
fprintf('\n Voltage Regulation=%.2f%%', reg);
```

O/P 1:-

```
  enter 1-for short line 2-medium line 3-for long line:1

  A,B,C and D constants:
-----
  A=      1.0000     +0.0000i
  B=      8.0000    +16.3200i
  C=      0.0000     +0.0000i
  D=      1.0000     +0.0000i
  the product AD-BC=1.000000
  enter 1 to read Vr,Ir and compute Vs,Is
  2 to read Vs,Is and compute Vr,Ir:1

  sending end voltage/ph=79.751426+1.805587iKV
  sending end current/ph=174.960000-131.220000iAMP
  receiving end power=40.00MW
  sending end power=41.15MW
  efficiency=97.21%
  voltage regulation= 4.67%>>
```

O/P2:-

```
enter 1-for short line 2-medium line 3-for long line:1

  A,B,C and D constants:
-----
  A=      1.0000     +0.0000i
  B=      8.0000    +16.3200i
  C=      0.0000     +0.0000i
  D=      1.0000     +0.0000i
  the product AD-BC=1.000000
  enter 1 to read Vr,Ir and compute Vs,Is
  2 to read Vs,Is and compute Vr,Ir:2

  receiving end voltage/ph=72.669045-1.805587iKV
  sending end current/ph=174.960000-131.220000iAMP
  receiving end power=38.85MW
  sending end power=40.00MW
  efficiency=97.13%
  voltage regulation= 4.84%>>
```

O/P 3:-

```
enter 1-for short line 2-medium line 3-for long line:2

  A,B,C and D constants:
-----
  A=       0.9874      +0.0062i
  B=     27.6485    +56.8476i
  C=       0.0000      +0.0004i
  D=       0.9874      +0.0062i
  the product AD-BC=1.000000
  enter 1 to read Vr,Ir and compute Vs,Is
  2 to read Vs,Is and compute Vr,Ir:1

  sending end voltage/ph=87.550333+6.787052iKV
  sending end current/ph=173.570962-94.993743iAMP
  receiving end power=40.00MW
  sending end power=43.65MW
  efficiency=91.63%
  voltage regulation= 16.69%>>
```

O/P 4:-

```
  enter 1-for short line 2-medium line 3-for long line:2

  A,B,C and D constants:
-----
  A=       0.9874      +0.0062i
  B=     27.6485    +56.8476i
  C=       0.0000      +0.0004i
  D=       0.9874      +0.0062i
  the product AD-BC=1.000000
  enter 1 to read Vr,Ir and compute Vs,Is
  2 to read Vs,Is and compute Vr,Ir:2

  receiving end voltage/ph=62.956502-5.848995iKV
  sending end current/ph=173.570962-161.997782iAMP
  receiving end power=35.62MW
  sending end power=40.00MW
  efficiency=89.06%
  voltage regulation= 22.06%>
```

O/P 5:-

```
enter 1-for short line 2-medium line 3-for long line:3

  the equivalent T circuit constants:
  Zeq=    122.3432           +246.5944i
  Yeq=      -0.0000                +0.0009i
  A,B,C and D constants:
-----
  A=        0.9428      +0.0277i
  B=      57.7123 +120.6169i
  C=      -0.0000      +0.0009i
  D=        0.9428      +0.0277i
  the product AD-BC=1.000000
  enter 1 to read Vr,Ir and compute Vs,Is
  2 to read Vs,Is and compute Vr,Ir:1

  sending end voltage/ph=97.773395+15.642655iKV
  sending end current/ph=167.915908-48.443896iAMP
  receiving end power=40.00MW
  sending end power=46.98MW
  efficiency=85.15%
  voltage regulation= 37.75%>>
```

O/P 6:-

```
  enter 1-for short line 2-medium line 3-for long line:3
  the equivalent T circuit constants:
  Zeq=    122.3432           +246.5944i
  Yeq=      -0.0000                +0.0009i
  A,B,C and D constants:
-----
  A=        0.9428      +0.0277i
  B=      57.7123 +120.6169i
  C=      -0.0000      +0.0009i
  D=        0.9428      +0.0277i
  the product AD-BC=1.000000
  enter 1 to read Vr,Ir and compute Vs,Is
  2 to read Vs,Is and compute Vr,Ir:2

  receiving end voltage/ph=45.924027-11.417589iKV
  sending end current/ph=169.252896-189.276902iAMP
  receiving end power=29.80MW
  sending end power=40.00MW
  efficiency=74.50%
  voltage regulation= 70.75%>>
```

PROGRAM 2 a): Power angle characteristics of Salient pole rotor syn. m/c

```
clc
clear all
%Xd=1.2;xq=0.8;ra=0.025; v=1.0; p=0.6; pf=0.8;
xd=input('Enter direct axis reactance per phase in pu\n');
xq=input('Enter quadrature    axis reactance per phase in pu\n');
ra=input('Enter the armature resistance per phase in pu\n');
v=input('Enter the terminal voltage of the generator in pu\n');
p=input('Enter the power output of the generator    in pu\n');
pf=input('Enter power factor\n');
pfsign=input('Enter -1 for lagging power factor & 1 for leading power factor');
iamag=p/(v*pf);
theta=acos(pf)*pfsign;
realia=iamag*cos(theta);
imagia=iamag*sin(theta);
ia=complex(realia, imagia);
efdash=v+(ia*ra)+(ia*j*xq);
delta=angle(efdash);
si=delta-theta;
id=iamag*sin(si);
iq=iamag*cos(si);
ef=abs(efdash)+id*(xd-xq);
p1=ef*abs(v)/xd;
p2=abs(v)^2*(xd-xq)/(2*xd*xq);
del=0:0.01:pi;
pext=p1*sin(del);
prlc=p2*sin(2*del);
pe=pext+prlc;
[pmax,k]=max(pe);
plot(del*180/pi,pext,'r');
hold on
plot(del*180/pi,prlc,'m');
plot(del*180/pi,pe,'b');
legend('excitation power','reluctance power', 'salient power');
title('power angle characteristics salient pole syn. m/c');
```

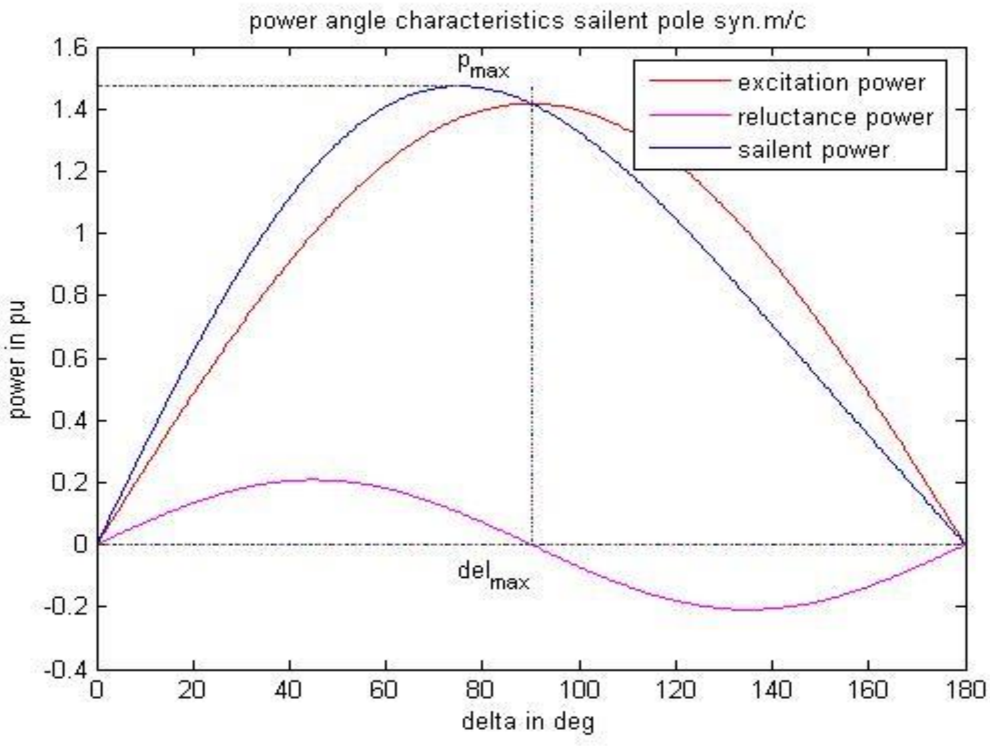
```
xlabel('delta in deg');
ylabel('power in pu');
% line(del*180/pi,0); del=0:0.01:del(k); line(del*180/pi,pmax);
%y=0:0.01:pmax; line(pi/2*(180/pi),y); text(75,1.05*pmax,'p_{max}');
% text(del(k)*180/pi,-0.1,'del_{max}');
reg=(abs(ef)-abs(v))/abs(v)*100;
fprintf('\n the excitation voltage=%f p.u',abs(ef));
fprintf('\n the maximum power=%f p.u and the corresponding angle=%f degrees',pmax,del(k)*180/pi);
fprintf('\n the voltage regulation=%f%%',reg);
fprintf('\n reluctance power=%f pu',p2);
```

O/P 1:-

Enter direct axis reactance per phase in pu
1.2
Enter quadrature axis reactance per phase in pu
0.8
Enter the armature resistance per phase in pu
0.025
Enter the terminal voltage of the generator in pu
1
Enter the power output of the generator in pu
0.6
Enter power factor
0.8
Enter -1 for lagging power factor& 1 for leading power factor-1

the excitation voltage=1.700518 pu
the max power =1.472984 pu and the corresponding angle=75.057471 degrees
the voltage regulation=70.051849%
reluctance power=0.208333pu>>
ibus =

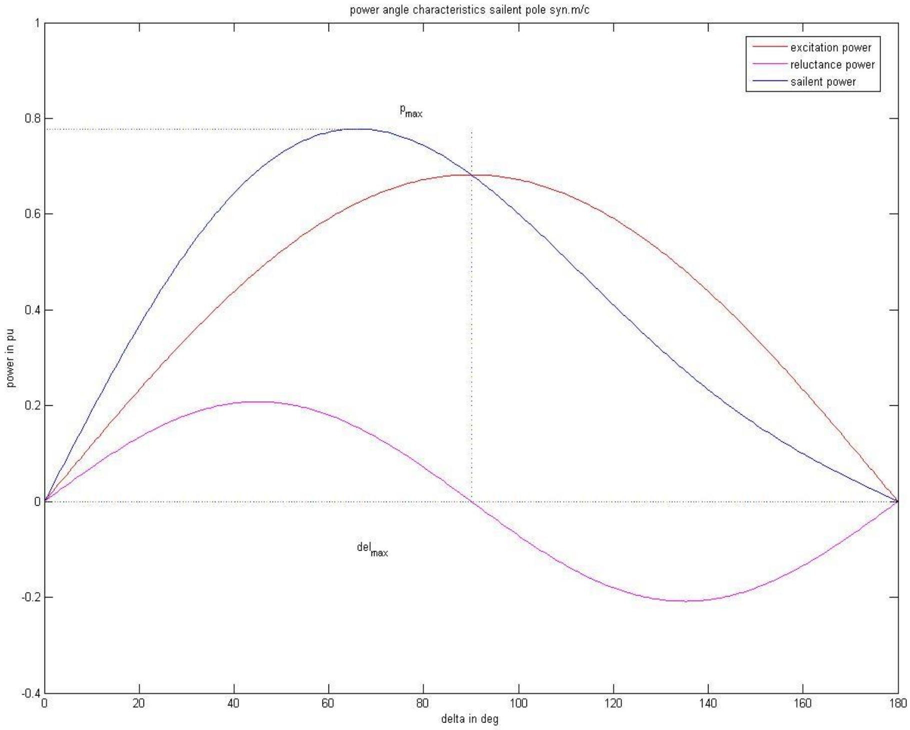
3.9000 - 1.8000i 1.2600 - 0.3200i 1.3600 - 0.5200i



O/P 2:-

Enter direct axis reactance per phase in pu
1.2
Enter quadrature axis reactance per phase in pu
0.8
Enter the armature resistance per phase in pu
0.025
Enter the terminal voltage of the generator in pu
1
Enter the power output of the generator in pu
0.6
Enter power factor
0.8
Enter -1 for lagging power factor& 1 for leading power factor1

the excitation voltage=0.818750 pu
the max power =0.778126 pu and the corresponding angle=65.890146 degrees
the voltage regulation=-18.125000%
reluctance power=0.208333pu>>

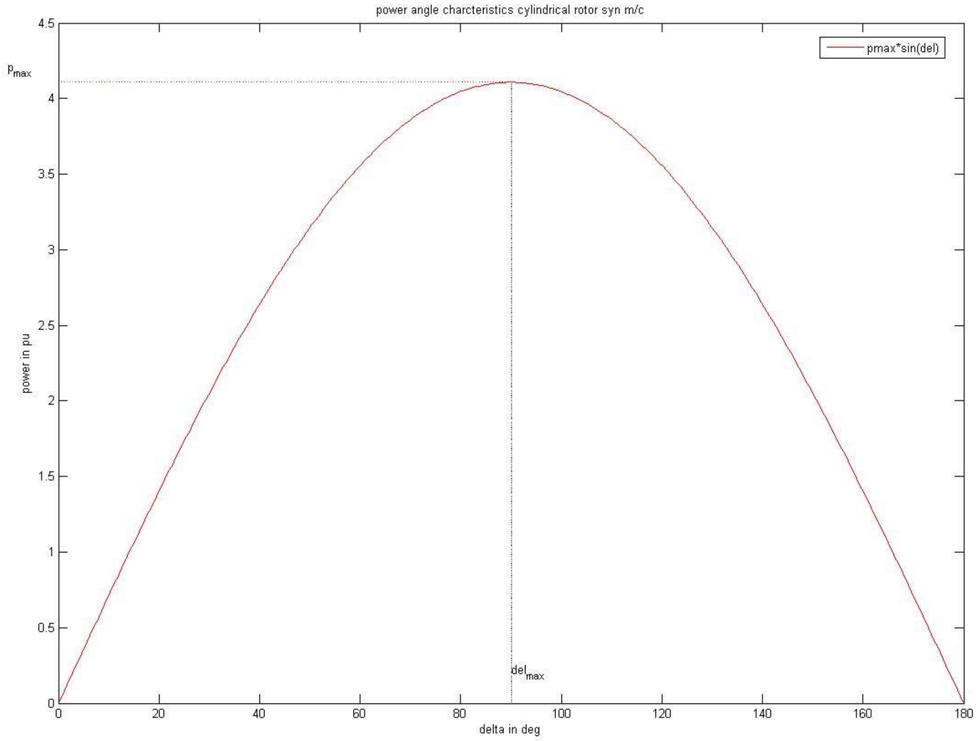


PROGRAM 2 b): Power angle characteristics of Non salient pole rotor synchronous machine

```
clc
clear all
%Xdd=0.3;ra=0.0; v=1.0; p=0.9; pf=0.8;
fprintf('generator is modelled by a voltage source behind a reactance\n');
xdd=input('Enter generator reactance per phase in pu\n');
ra=input('Enter the armature resistance per phase in pu\n');
v=input('Enter the terminal voltage of the generator in pu\n');
p=input('Enter the power output of the generator    in pu\n');
pf=input('Enter power factor\n');
pfsign=input('Enter -1 for lagging power factor & 1 for leading power factor');

iamag=p/(v*pf);
theta=acos(pf)*pfsign;
realia=iamag*cos(theta);
imagia=iamag*sin(theta);
ia=complex(realia, imagia);
ef=v+ia*(ra+j*xdd);
pmax=abs(ef)*abs(v)/xdd;
del=0:0.01:pi;
plot(del*180/pi,pmax*sin(del),'r');

title('power angle characteristics cylindrical rotor syn m/c');

xlabel('Delta in deg');
ylabel('Power in pu');
del=0:0.01:pi/2;
  legend('pmax*sin(del)');
%line(del*180/pi,pmax);
% y=0:0.01:pmax; line(pi/2*(180/pi),y);
% text(-10,1.02*pmax,'p_{max}');
% text(pi/2*(180/pi),0.2,'del_{max}');
reg=(abs(ef)-abs(v))/abs(v)*100;
fprintf('\n the voltage regulation=%f %%\n',reg);
fprintf('the maximum power=%f pu',pmax);
```

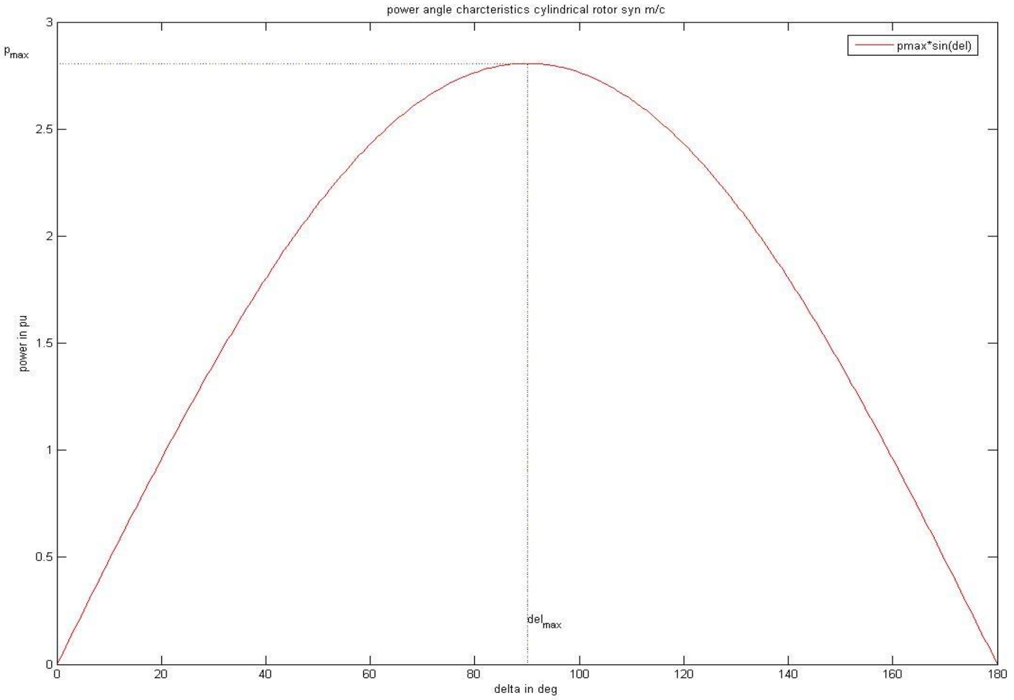
O/P 1:-

generator is modelled by a voltage source behind a reactance
enter generator reactance per phase in pu
0.3
enter the armature resistance per phase in pu
0.0
enter the tewrminal voltage of the generator per phase in pu
1.0
enter the power output of the generator in pu
0.9
enter the power factor
0.8
enter -1 for lagging power factor & 1 for leading power factor-1the maximum power=2.806552 pu
the voltage regulation=23.243915%
the maximum power=4.108130 pu>>



O/P 2:-

generator is modelled by a voltage source behind a reactance
enter generator reactance per phase in pu
0.3
enter the armature resistance per phase in pu
0.0
enter the tewrminal voltage of the generator per phase in pu
1.0
enter the power output of the generator in pu
0.9
enter the power factor
0.8
enter -1 for lagging power factor & 1 for leading power factor1
the voltage regulation=-15.803429%
the maximum power=2.806552 pu>>



PROGRAM 3(a): To determine i) swing curve ii) critical clearing time for a single machine connected to infinite bus through a pair of identical transmission lines, for a 3ϕ sustained fault on one of the lines.

```
% ps=mechanical power input.
%x2=reactance after fault
clc
clear all
ps=0.9;e=1.1;v=1;m=0.00028;xe=0.35;xl=0.2;
% m in electrical degrees
x1=xe+xl/2;
ch=input('enter 1 for fault at the beginning \n    2-for fault at the middle: ');
switch(ch)
        case 1,
                x2=inf;
        case 2,
                x2=(xe*xl+xe*xl/2+xl*xl/2)/(xl/2);
        otherwise
                disp('wrong input');
end
dt=0.05;
it=1;      % it is the iteration count
t(it)=0;
deld=0;
pm1=e*v/x1;
del(it)=asin(ps/pm1);
pm2=e*v/x2;
pm=pm2;
pe=pm2*sin(del(it));
pa=(ps-pe)/2;
fprintf('\n sustained fault');
fprintf('\n----------------------------');
fprintf('\n time            pmax                delta');
fprintf('\n----------------------------');
while(t(it)<=1.0)
        deld=deld+(pa*(dt^2/m));
        it=it+1;
        t(it)=t(it-1)+dt;
```

```
del(it)=del(it-1)+deld*(pi/180); % deld is converted to radian and added.
    fprintf('\n %5.3f     %5.2f          %5.2f',t(it),pm,del(it)*(180/pi));
     pm=pm2;
    pe=pm*sin(del(it));
    pa=ps-pe;
end


plot(t,del*(180/pi),'r')
title('Swing curve')
xlabel('time in seconds')
ylabel('\delta in degrees')
x3=xe+xl; % x3=0.55
pm3=e*v/x3;
delm=pi-del(1);
cdc=(ps*(delm-del(1))+pm3*cos(delm)-pm2*cos(del(1)))/(pm3-pm2);
delc=acos(cdc);
D=delc*(180/pi);
fprintf('\n critical clearing angle %f',D);
it=1;
while(t(it)<1)
    if(del(it)>=delc)
        break;
    end
    it=it+1;
end
fprintf('critical clearing time=%f sec\n',t(it));
```

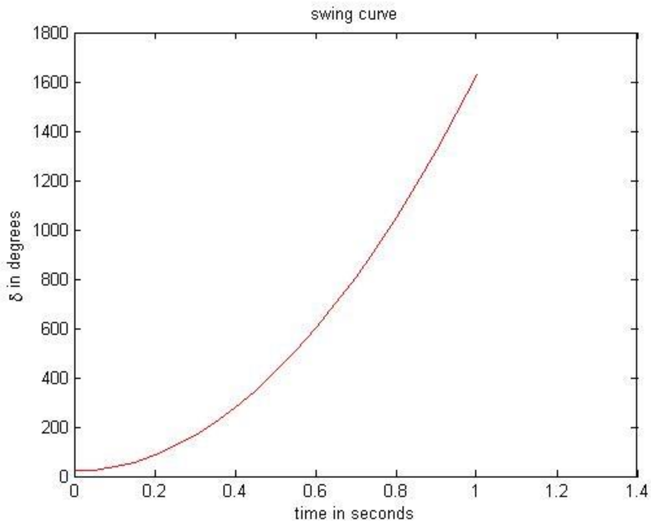
O/P 1:-

enter 1 for fault at the beginning
 2-for fault at the middle:1

 sustained fault
----
 time pmax delta

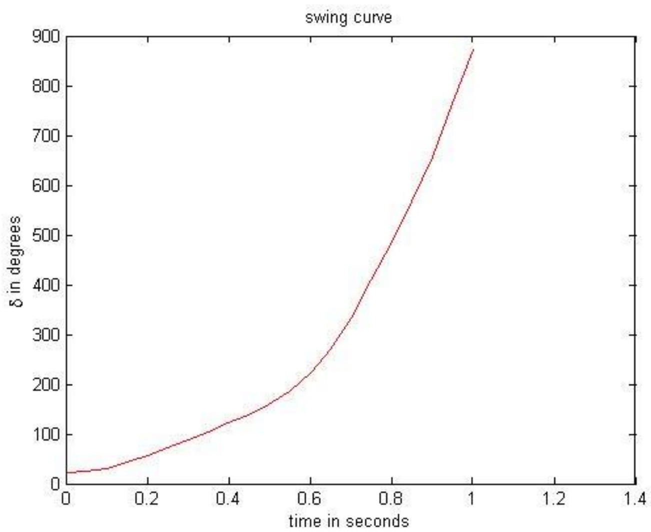
0.050	0.00	25.62
0.100	0.00	37.67
0.150	0.00	57.76
0.200	0.00	85.89
0.250	0.00	122.05
0.300	0.00	166.25
0.350	0.00	218.48
0.400	0.00	278.75
0.450	0.00	347.05
0.500	0.00	423.39
0.550	0.00	507.76
0.600	0.00	600.17
0.650	0.00	700.62
0.700	0.00	809.10
0.750	0.00	925.62
0.800	0.00	1050.17
0.850	0.00	1182.76
0.900	0.00	1323.39
0.950	0.00	1472.05
1.000	0.00	1628.75

 critical clearing angle 81.684989critical clearing time=0.200000 sec
>>



O/P 2:-

```
enter 1 for fault at the beginning
  2-for fault at the middle:2

  sustained fault
----------
  time        pmax       delta
  -------------
0.050     0.88     24.17
0.100     0.88     31.56
0.150     0.88     42.88
0.200     0.88     56.88
0.250     0.88     72.34
0.300     0.88     88.34
0.350     0.88     104.53
0.400     0.88     121.15
0.450     0.88     139.08
0.500     0.88     159.89
0.550     0.88     186.05
0.600     0.88     221.06
0.650     0.88     269.28
0.700     0.88     333.38
0.750     0.88     409.04
0.800     0.88     486.81
0.850     0.88     566.32
0.900     0.88     657.34
0.950     0.88     763.39
1.000     0.88     872.07
  critical clearing angle 118.182332critical clearing time=0.400000 sec
>>
```



PROGRAM 3 (b): Program to obtain swing curve when the fault is cleared using MATLAB.

```
% check the output for case 1- time=0.05 & 0.125secs and
  for case 2-time=0.5secs and 0.2secs
clear all
clc
ps=0.9;e=1.1; v=1.0; m=0.00028; xe=0.35;xl=0.2;
x1=xe+xl/2;
ch=input('enter 1 for fault at the beginning: \n 2 for fault at the middle:');
switch(ch)
case 1,
      x2=inf;
case 2,
      x2=(xe*xl+xe*xl/2+xl*xl/2)/(xl/2);
otherwise
      disp('wrong input');
end
x3=xe+xl;
dt=0.05;
ct=input('\n enter clearing time in secs:');
k=ct/dt;
r=ct-floor(k)*dt;
if (r==0)
      fprintf('fault is cleared at the beginning of an interval');
else
      fprintf('fault is cleared at the middle of an interval');
end
deld=0;
it=1;
t(it)=0;
pmax1=e*v/x1;
del(it)=asin(ps/pmax1);
pmax2=e*v/x2;
pmax3=e*v/x3;
pm=pmax2;
pe=pm*sin(del(it));
pa=(ps-pe)/2;
fprintf('\n
while(t(it)<=1.05)
      deld=deld+(pa*(dt^2/m));    % deld is calculated in degrees.
      fprintf('\n %5.2f,       %5.2f,          %5.2f ',t(it),pm,del(it)*(180/pi));
      it=it+1;
      t(it)=t(it-1)+dt;
```

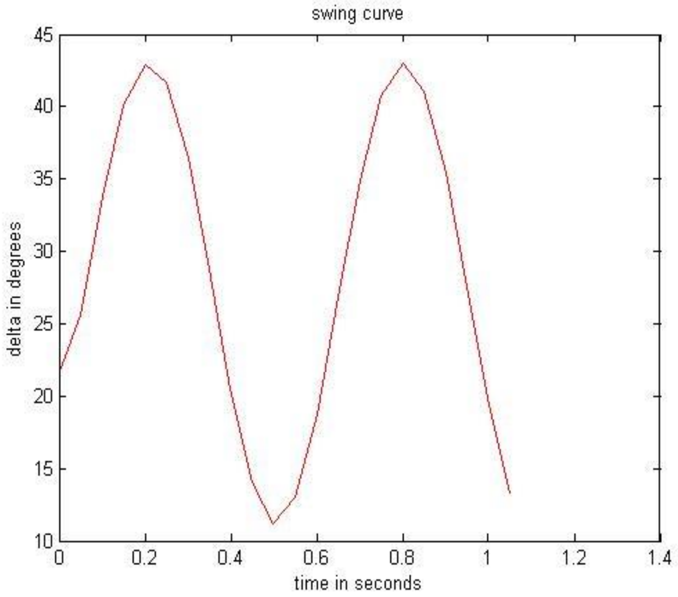
```
        del(it)=del(it-1)+deld*(pi/180);    % deld is converted into radians and then added.
if(r==0)
        if(t(it)<ct)

                pm=pmax2;
        else
            if (t(it)==ct)
            pm=(pmax2+pmax3)/2;
         else
                pm=pmax3;
         end
end
if (r~=0)
        if    (t(it)<ct)
                pm=pmax2;
        else
                pm=pmax3;
        end
end
pe=pm*sin(del(it));
pa=ps-pe;
end
plot(t,del*(180/pi),'r');
title('Swing curve');
xlabel('time in seconds');
ylabel('delta in degrees');
pr=0;
for k=2:it
        if (del(k)<del(k-1))
                        pr=1;
                break
        end
end


if (pr)
        text(0.5,1.0,'system is stable');
else
                text(0.5,1.0, 'system is unstable');
        end
end
```

O/P 1:-

enter 1 for fault at the beginning
 2-for fault at the middle:1
enter clearing time in sec:0.05
fault is cleared at the beginning of an interval

TIME	PMAX	DELTA
0.00,	0.00,	21.60
0.05,	1.00,	25.62
0.10,	2.00,	33.81
0.15,	2.00,	40.10
0.20,	2.00,	42.93
0.25,	2.00,	41.62
0.30,	2.00,	36.50
0.35,	2.00,	28.78
0.40,	2.00,	20.51
0.45,	2.00,	14.01
0.50,	2.00,	11.23
0.55,	2.00,	13.00
0.60,	2.00,	18.80
0.65,	2.00,	26.87
0.70,	2.00,	34.91
0.75,	2.00,	40.77
0.80,	2.00,	43.00
0.85,	2.00,	41.09
0.90,	2.00,	35.47
0.95,	2.00,	27.54
1.00,	2.00,	19.38>>

critical clearing angle 81.684989critical clearing time=0.200000 sec

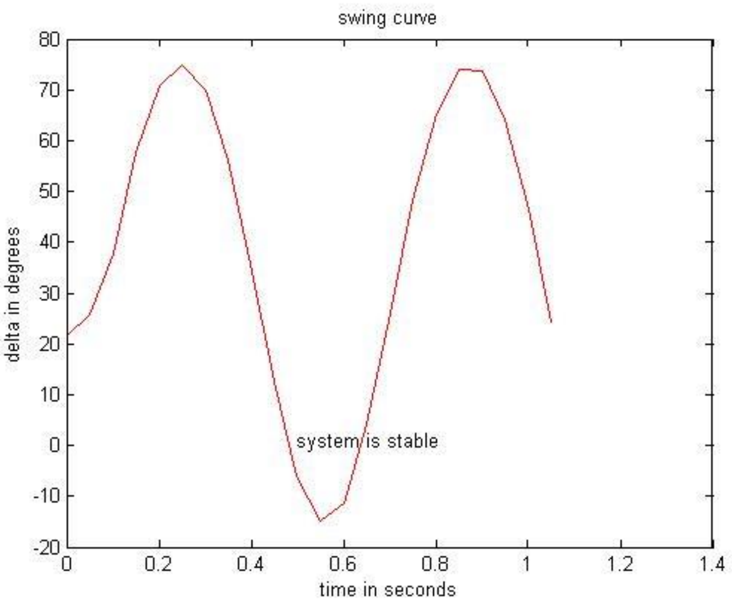


O/P 2:-

enter 1 for fault at the beginning
2-for fault at the middle:1

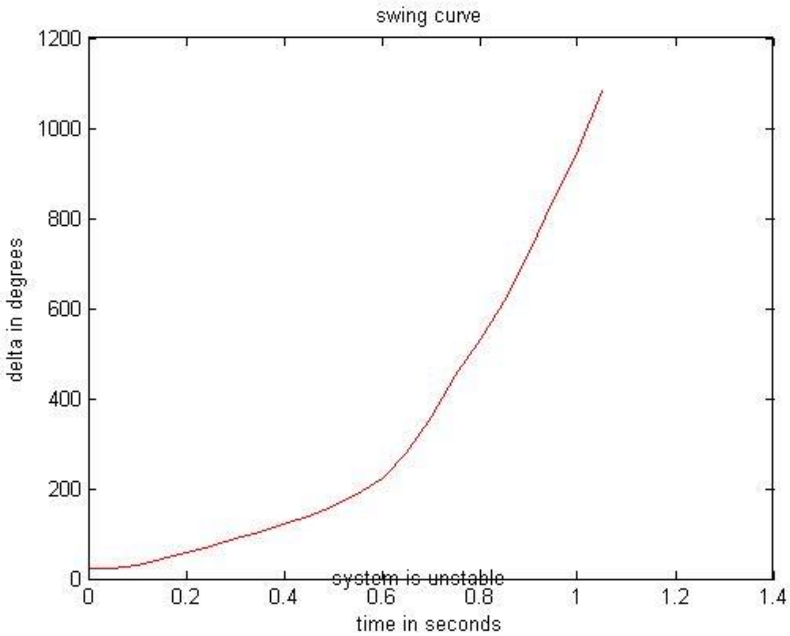
enter clearing time in sec:0.125
fault is cleared at the middlw of an interval

TIME	PMAX	DELTA
0.00,	0.00,	21.60
0.05,	0.00,	25.62
0.10,	0.00,	37.67
0.15,	2.00,	57.76
0.20,	2.00,	70.78
0.25,	2.00,	74.98
0.30,	2.00,	69.96
0.35,	2.00,	56.20
0.40,	2.00,	35.64
0.45,	2.00,	12.71
0.50,	2.00,	-6.11
0.55,	2.00,	-15.00
0.60,	2.00,	-11.23
0.65,	2.00,	4.05
0.70,	2.00,	26.11
0.75,	2.00,	48.35
0.80,	2.00,	65.27
0.85,	2.00,	74.02
0.90,	2.00,	73.63
0.95,	2.00,	64.14
1.00,	2.00,	46.62>>



O/P 3:-

```
enter 1 for fault at the beginning
  2-for fault at the middle:2

  enter clearing time in sec:0.5
fault is cleared at the beginning of an interval
  TIME        PMAX          DELTA
    0.00,       0.88,       21.60
    0.05,       0.88,       24.17
    0.10,       0.88,       31.56
    0.15,       0.88,       42.88
    0.20,       0.88,       56.88
    0.25,       0.88,       72.34
    0.30,       0.88,       88.34
    0.35,       0.88,       104.53
    0.40,       0.88,       121.15
    0.45,       0.88,       139.08
    0.50,       0.88,       159.89
    0.55,       2.00,       186.05
    0.60,       2.00,       222.12
    0.65,       2.00,       278.20
    0.70,       2.00,       359.99
    0.75,       2.00,       449.82
    0.80,       2.00,       529.83
    0.85,       2.00,       614.72
    0.90,       2.00,       724.88
    0.95,       2.00,       841.55
    1.00,       2.00,       951.04>>
```

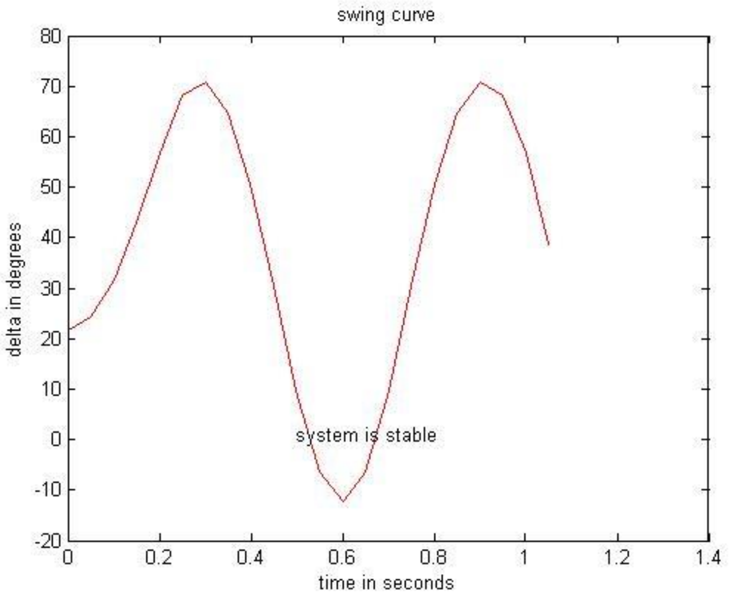


O/P 4:-

enter 1 for fault at the beginning
2-for fault at the middle:2

enter clearing time in sec:0.2
fault is cleared at the beginning of an interval

TIME	PMAX	DELTA
0.00,	0.88,	21.60
0.05,	0.88,	24.17
0.10,	0.88,	31.56
0.15,	0.88,	42.88
0.20,	1.44,	56.88
0.25,	2.00,	68.15
0.30,	2.00,	70.88
0.35,	2.00,	64.77
0.40,	2.00,	50.55
0.45,	2.00,	30.57
0.50,	2.00,	9.55
0.55,	2.00,	-6.40
0.60,	2.00,	-12.33
0.65,	2.00,	-6.40
0.70,	2.00,	9.55
0.75,	2.00,	30.57
0.80,	2.00,	50.55
0.85,	2.00,	64.77
0.90,	2.00,	70.88
0.95,	2.00,	68.15
1.00,	2.00,	56.88>>



PROGRAM 4 a): Formation of Y-bus without line charging-Inspection method

```
clc
data1=[ 1 2 0.3 0.4;                              % creating input file
        1 3 0.2 0.5;
        2 3 0.1 0.3];
nb=data1(1,1);                                     % no. of buses.
nl=data1(1,2);                                     % no. of lines
for i=1:nl
sb(i)=data1(i+1,1);                                % sending bus
eb(i)=data1(i+1,2);                                % end bus
serz(i)=(data1(i+1,3)+data1(i+1,4)*j);             % series impedance
end

for j=1:nb
    for k=1:nb
        ybus(j,k)=0;                              % initializes all elements of
ybus to zero.
    end
end

for i=1:nl
    j=sb(i);
    k=eb(i);
    ybus(j,j)=ybus(j,j)+1/serz(i);
    ybus(k,k)=ybus(k,k)+1/serz(i);
    ybus(j,k)=ybus(j,k)-1/serz(i);
    ybus(k,j)=ybus(j,k);
end
ybus                                              % displays ybus
```

O/P:-

```
ybus =

40.0000   -20.0000   -20.0000          0
-20.0000    46.6667   -10.0000   -16.6667
-20.0000   -10.0000    50.0000   -20.0000
0   -16.6667   -20.0000    36.6667
```

PROGRAM 4 b): Program for the formation of Ybus with line charging--Inspection Method

```
clear all
%          sb    eb       serr    serx    shty
linedata=[1      2      0.02    0.06 0.03;
            1      3      0.08    0.24 0.025;
            2      3      0.06    0.18 0.02;
            2      4      0.06    0.18 0.02;
            2      5      0.04    0.12 0.015;
            3      4      0.01    0.03    0.01;
            4      5      0.08    0.24 0.025];
sb=linedata(:,1);
eb=linedata(:,2);
nl=max(size(sb));
nb=max(max(sb,eb));
serz=(linedata(:,3)+j*linedata(:,4));
shty=(0+j*linedata(:,5));
ybus=zeros(nb,nb);
for i=1:nl
      j=sb(i);
      k=eb(i);
      ybus(j,j)=ybus(j,j)+1/serz(i)+shty(i);
      ybus(k,k)=ybus(k,k)+1/serz(i)+shty(i);
      ybus(j,k)=-1/serz(i);
      ybus(k,j)=ybus(j,k);
end
ybus
```

O/P:-

```
ybus =

    6.2500 -18.6950i    -5.0000 +15.0000i    -1.2500 + 3.7500i                0                          0
   -5.0000  +15.0000i    10.8333  -32.4150i    -1.6667  +  5.0000i    -1.6667  +  5.0000i    -2.5000  +
7.5000i
   -1.2500 + 3.7500i    -1.6667 + 5.0000i    12.9167 -38.6950i -10.0000 +30.0000i              0
             0                        -1.6667  +  5.0000i  -10.0000  +30.0000i    12.9167  -38.6950i    -1.2500  +
3.7500i
             0                        -2.5000  +  7.5000i              0                      -1.2500  +  3.7500i      3.7500
-11.2100i
```

Program 4 c): Formation of Y-bus using Singular Transformation Method with Mutual Coupling (line charging is neglected)

```
clc
clear all
data=[1 2 0.6i 0 0;
      1 3 0.5i 1 0.1i;
      3 4 0.5i 0 0;
      1 2 0.4i 1 0.2i;
      2 4 0.2i 0 0];
   fb=data(:,1);
   tb=data(:,2);
   z=data(:,3);
   mc=data(:,4);
   mz=data(:,5);
   nnode=max(max(fb),max(tb));
   nbus=nnode -1;
   nline=length(fb);
   zpr=zeros(nline,nline);
   for k=1:nline
        zpr(k,k)=z(k);
        if mc(k)~=0
             zpr(k,mc(k))=mz(k);
             zpr(mc(k),k)=mz(k); end
   end
   ypr=inv(zpr);
   Acap=zeros(nline,nnode);
   for k=1:nline
        Acap(k,fb(k))=1;
        Acap(k,tb(k))=-1;
   end
   A=Acap(:,2:nnode);
   ybus=A'*ypr*A;
   ybus
```

O/P:-

ybus =

```
   0 - 8.0208i     0 + 0.2083i     0 + 5.0000i
   0 + 0.2083i     0 - 4.0833i     0 + 2.0000i
   0 + 5.0000i     0 + 2.0000i     0 - 7.0000i
```

PROGRAM 4 d): Formation of Ybus using Singular Transformation Method without Mutual Coupling

```
clc
clear all
%           fb   tb   Z          hlcy(Admittance)
linedata=[1 2 0.02+0.06i    0.03i
          1 3 0.08+0.24i    0.025i
          2 3 0.06+0.18i    0.02i
          2 4 0.06+0.18i    0.02i
          2 5 0.04+0.12i    0.015i
          3 4 0.01+0.03i    0.01i
          4 5 0.08+0.24i    0.025i];
fb=linedata(:,1);
tb=linedata(:,2);
z=linedata(:,3);
hlcy=linedata(:,4);
y=1./z;
nbus=max(max(fb),max(tb));
ybus=zeros(nbus);
nline=length(fb);
nlb=nline+nbus;
A=zeros(nlb,nbus);

for k=1:nbus
    A(k,k)=1;
end

for k=1:nline
    A(nbus+k,fb(k))=1;
    A(nbus+k,tb(k))=-1;
end
sh=zeros(1,nbus);
for k=1:nline
    sh(fb(k))=sh(fb(k))+hlcy(k);
    sh(tb(k))=sh(tb(k))+hlcy(k);
end
```

```
ypr=zeros(nlb,nlb);
for k=1:nbus
    ypr(k,k)=sh(k);
end
for k=1:nline
    ypr(nbus+k,nbus+k)=y(k);
end
format short;
ybus=A'*ypr*A;
ybus
```

O/P:-

ybus =

```
   6.2500 -18.7500i  -5.0000 +15.0000i  -1.2500 + 3.7500i        0                  0
  -5.0000 +15.0000i  24.3333 -37.0000i  -1.6667 + 5.0000i  -1.6667 + 5.0000i -16.0000 +12.0000i
  -1.2500 + 3.7500i  -1.6667 + 5.0000i   2.9167 - 8.7500i        0                  0
        0            -1.6667 + 5.0000i        0             1.6667 - 5.0000i        0
        0           -16.0000 +12.0000i        0                  0            16.0000 -12.0000i
```

PROGRAM 5: FORMATION OF Z-BUS, USING Z-BUS BUILDING ALGORITHM WITHOUT MUTUAL COUPLING

```
clc;
clear;
disp('Zbus Building Algorithm');
zprimary = [1    1    0    0.25
                    2    2    1    0.1
                    3    3    1    0.1
                    4    2    0    0.25
                    5    2    3    0.1];
[elements,columns]=size(zprimary);
zbus=[];
currentbusno=0;
for count=1:elements
    [rows cols]=size(zbus);
    from=zprimary(count,2);
    to=zprimary(count,3);
    value=zprimary(count,4);
    newbus=max(from,to);
    ref=min(from,to);

%Type-1 Modification
%A new element is added from new bus to reference bus
if newbus>currentbusno & ref==0
      disp('Adding Z =');
      disp(value);
      disp('between buses:');
      disp(from);
      disp(to);
      disp('This impedance is added between a new bus and reference(Type1)');
    zbus=[zbus zeros(rows,1)
          zeros(1,cols) value]
      currentbusno=newbus;
      continue
end

%Type-2 Modification
%A element is added from new bus to old bus other than reference bus
if newbus>currentbusno & ref~=0
      disp('Adding Z =');
      disp(value);
      disp('between buses:');
      disp(from);
```

```
        disp(to);
        disp('This impedance is added between a new bus and an existing bus(Type2)');
      zbus=[zbus zbus(:,ref)
            zbus(ref,:) value+zbus(ref,ref)]
        currentbusno=newbus;
        continue
end

%Type-3 Modification
%A new element is added between an old bus and reference bus
if newbus<=currentbusno & ref==0
        disp('Adding Z =');
        disp(value);
        disp('between buses:');
        disp(from);
        disp(to);
        disp('This impedance is added between an existing bus and reference(Type3)');
      zbus=zbus-1/(zbus(newbus,newbus)+value)*zbus(:,newbus)*zbus(newbus,:)
      continue
end

%Type-4 Modification
%A new element is added between two old buses(bus-2 to 3)
  if newbus<=currentbusno & ref~=0
        disp('Adding Z =');
        disp(value);
        disp('between buses:');
        disp(from);
        disp(to);
        disp('This impedance is added between two existing buses(Type4)');

zbus=zbus-1/(value+zbus(from,from)+zbus(to,to)-2*zbus(from,to))*((zbus(:,from)-zbus(:,to)
)*((zbus(from,:)-zbus(to,:))))
      continue
  end
end
```

OUTPUT:

```
Zbus Building Algorithm
Adding Z =
    0.2500

between buses:
     1

     0

This impedance is added between a new bus and reference(Type1)

zbus =

    0.2500

Adding Z =
    0.1000

between buses:
     2

     1

This impedance is added between a new bus and an existing bus(Type2)

zbus =

    0.2500    0.2500
    0.2500    0.3500

Adding Z =
    0.1000

between buses:
     3

     1

This impedance is added between a new bus and an existing bus(Type2)
```

```
zbus =

    0.2500    0.2500    0.2500
    0.2500    0.3500    0.2500
    0.2500    0.2500    0.3500

Adding Z =
    0.2500

between buses:
     2

     0

This impedance is added between an existing bus and reference(Type3)

zbus =

    0.1458    0.1042    0.1458
    0.1042    0.1458    0.1042
    0.1458    0.1042    0.2458

Adding Z =
    0.1000

between buses:
     2

     3

This impedance is added between two existing buses(Type4)

zbus =

    0.1397    0.1103    0.1250
    0.1103    0.1397    0.1250
    0.1250    0.1250    0.1750
```

PROGRAM 6: Determination of bus current, bus power & line flows for a specified voltage bus

```
clc
clear all
%           fb    tb    z
linedata=[1      2    0.02+0.04i;
          1      3     0.01+0.03i;
          2      3     0.0125+0.025i];
  vb=[1.05+0.0i;
      0.98-0.06i;
      1.00-0.05i];
fb=linedata(:,1);
tb=linedata(:,2);
z=linedata(:,3);
nl=max(size(fb));
y=1./z;
for k=1:nl
    il(fb(k),tb(k))=y(k)*(vb(fb(k))-vb(tb(k)));
    il(tb(k),fb(k))=-il(fb(k),tb(k));
end
fprintf('the line currents:\n');
il
for k=1:nl
    lf(fb(k),tb(k))=vb(fb(k))*conj(il(fb(k),tb(k)));
    lf(tb(k),fb(k))=vb(tb(k))*conj(il(tb(k),fb(k)));
   ll(k)=lf(fb(k),tb(k))+lf(tb(k),fb(k));
end
fprintf('the line flows:\n');
lf
fprintf('the line losses:\n');
ll
for k=1:nl
    bp(k)=sum(lf(k,:));
    ibus(k)=conj(bp(k)/vb(k));
end
fprintf('the bus powers:\n');
bp
fprintf('the bus currents:\n');
ibus
```

O/P:-

the line currents:

il =

```
          0                  1.9000 - 0.8000i    2.0000 - 1.0000i
   1.9000 - 0.8000i          0                  -0.6400 + 0.4800i
   2.0000 - 1.0000i   -0.6400 + 0.4800i              0
```

the line flows:

lf =

```
          0                  1.9950 + 0.8400i    2.1000 + 1.0500i
   1.9100 + 0.6700i          0                  -0.6560 - 0.4320i
   2.0500 + 0.9000i   -0.6640 - 0.4480i              0
```

the line losses:

ll =

```
   3.9050 + 1.5100i    4.1500 + 1.9500i   -1.3200 - 0.8800i
```

the bus powers:

bp =

```
   4.0950 + 1.8900i    1.2540 + 0.2380i    1.3860 + 0.4520i
```

the bus currents:

ibus =

```
   3.9000 - 1.8000i    1.2600 - 0.3200i    1.3600 - 0.5200i
```