ELECTRICAL SYSTEMS SIMULATION LAB MANUAL

IV B.TECH – I SEM (R16)

<u>Prepared BY</u>:

G.KUMARASWAMY

ASSISTANT PROFESSOR

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING
PRIYADARSINI INSTITUTE OF SCIENCE & TECHNOLOGY FOR WOMEN

LIST OF EXPERIMENTS

Prerequisite: Electrical and Electronic circuits, Power System Analysis & Power Electronics

Course Objectives:

- To Simulate and analyse electrical and electronic systems.
- To evaluate the performance of transmission lines.
- To Analyze various Faults in power systems
- To Model, simulate and analyze the performance of DC Machines and Induction Motors.
- To Analyze performance of feedback and load frequency control of the systems

Course Outcomes: After going through this lab the student will be able to

- Design and Analyze electrical systems in time and frequency domain
- Analyze various transmission lines and perform fault analysis
- Model Load frequency control of Power Systems
- Design various Power Electronic Converters and Drives.

Any ten of the following experiments are required to be conducted using suitable software

1. Design of first and second order circuits in time and frequency domain
2. Performance evaluation of medium and long transmission lines
3. Symmetrical component analysis
4. Transmission Line Fault Analysis
5. LG, LL and 3-Φ fault analysis of Transformer
6. Short Circuit Analysis of Power system models
7. Speed Control of DC Motor
8. Speed Control of Induction motor
9. Design and analysis of feedback control system
10. Transient analysis of open ended line and short circuited line
11. Load frequency control of single area and two area power system
12. Economic Dispatch of Thermal Units
13. Design of Single Phase and Three Phase Inverters
14. Design of Single Phase and Three Phase Full Converters

INDEX

1. DESIGN OF FIRST AND SECOND ORDER CIRCUITS IN TIME AND FREQUENCY DOMAIN

AIM: To find the I) Time response for step input
II) Frequency response for sinusoidal input.

Software Required: MATLAB software, R2009a
Windows XP operating system.

Apparatus Required: Personal Computer (PC).

Procedure:

1. Open the MATLAB software
2. Open the M-file
3. Type the program in editor window
4. Save in current directory as "filename"
5. Compile and run the program

I) Time response for step input: For the closed loop system defined by

$$\frac{C(S)}{R(S)} = \frac{100}{S^2 + 12S + 100}$$

Plot the unit step response curve and time domain specifications.

Program:

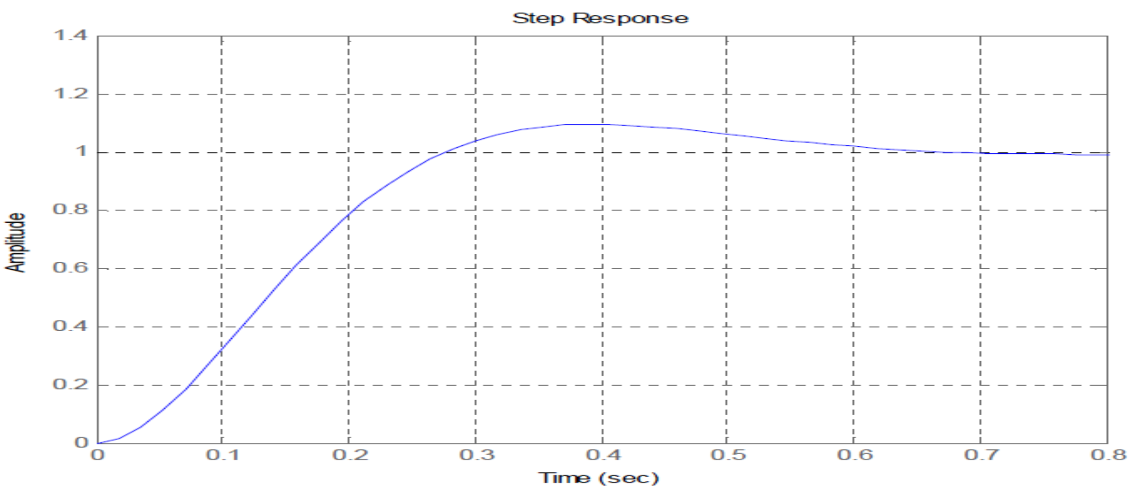
```
clc;
clear all;
close all;
num=input('enter the numerator coefficients-----');
den=input('enter the denominator coefficients----');
system=tf(num,den);
system
step(system)
grid on;
wn=sqrt(den(1,3));
zeta= den(1,2)/(2*wn);
wd=wn*sqrt(1-zeta^2);
disp('Delay time in seconds is')
td=(1+0.7*zeta)/wd
disp('Rise time in seconds is')
theta=atan(sqrt(1-zeta^2)/zeta);
tr=(pi-theta)/wd
disp('Peak time in seconds');
tp=pi/wd
```

```
disp('Peak overshoot is');
mp=exp(-zeta*pi/sqrt(1-zeta^2))*100
disp('settling time in seconds is');
ts=4/(zeta*wn)
```

Output:

```
enter the numerator coefficients---->100
enter the denominator coefficients---->[1 12 100]
Transfer function:
100
----------------
s^2 + 12 s + 100
Delay time in seconds is
td =
0.1775
Rise time in seconds is
tr =
0.2768
Peak time in seconds
tp =
0.3927
Peak overshoot is
mp =
9.4780
settling time in seconds is
ts =
0.6667
```

Simulation waveform:



II) Frequency response for sinusoidal input:

Program:

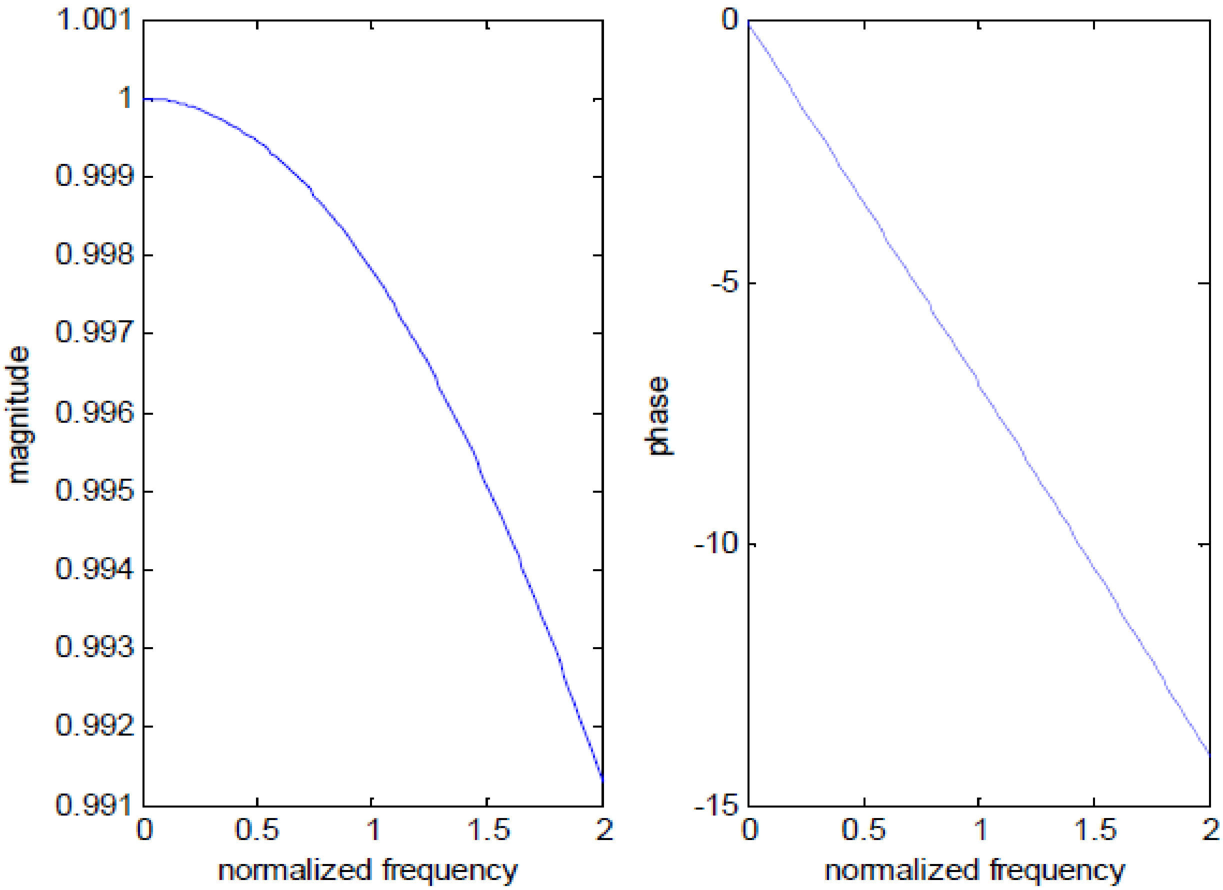
```
%Frequency Response of second order system
clc;
clear all;
close all;
num=input('enter the numerator coefficients---->');
den=input('enter the denominator coefficients---->');
%Transfer function
sys=tf(num,den);
wn=sqrt(den(1,3));
zeta= den(1,2)/(2*wn);
w=linspace(0,2);
u=w/wn;
len=length(u);
for k=1:len
m(k)=1/(sqrt((1-u(k)^2)+(2*zeta*u(k))^2));
phi(k)=-atan((2*zeta*u(k))/(1-u(k)^2))*180/pi;
end
subplot(1,2,1)
plot(w,m)
xlabel('normalized frequency')
ylabel('magnitude')
subplot(1,2,2)
plot(w,phi)
xlabel('normalized frequency')
ylabel('phase')
disp('resonant peak is');
mr=1/(2*zeta*sqrt(1-zeta^2))
disp('resonant frequency in rad/sec is');
wr=wn*sqrt(1-2*zeta^2)
disp('bandwidth in rad/sec is');
wb=wn*sqrt(1-2*zeta^2+sqrt(2-4*zeta^2+4*zeta^4))
disp('phase margin in degrees is')
  pm=180+(atan(2*zeta/sqrt(-2*zeta^2+sqrt(4*zeta^4 +1))))*180/pi
```

Output:

```
enter the numerator coefficients---->100
enter the denominator coefficients---->[1 12 100]
resonant peak is
mr =
1.0417
resonant frequency in rad/sec is
wr =
5.2915
bandwidth in rad/sec is
wb =
11.4824
phase margin in degrees is
```

```
pm =
239.1873
```

Simulation waveform:



Result:

2. PERFORMANCE EVALUATION OF MEDIUM AND LONG TRANSMISSION LINES

AIM: Determination of ABCD parameters for medium and long transmission lines with a given condition and hence studies the performance of the line regulation and efficiency.

Software Required: MATLAB software, R2009a

Windows XP operating system.

Apparatus Required: Personal Computer (PC).

Procedure:

1. Open the MATLAB software
2. Open the M-file
3. Type the program in editor window
4. Save in current directory as “filename”
5. Compile and run the program

I) Medium transmission line:

Statement: A 3-phase 50Hz overhead transmission line delivers 10 MW at 0.8pf lagging at 66 kV. The resistance, inductive reactance and capacitive susceptance 10 ohm, 20 ohm and 4*10^-4 siemen. Determine 1)Sending end current 2)Sending end voltage 3)Sending end power factor 4)ABCD parameter 5) Regulation 6) Transmission effeciency using nominal T method.

Program:

```
%medium transmission line
 clear;
clc;
R=input('resistance of the line in ohm=');
X=input('reactance of the line in ohm=');
B=input('susceptance of shunt line in mho=');
 VR3ph=input('voltage at receiving end in KV=');
 PR=input('real load at receiving end in MW=');
QR=input('reactive load at receiving end in MVAR=');
 Z=R+j*X;
 Y=0+j*B;
 type=input('type(P-Pie/T-tmethod)=','s');
 switch type
 case'P'
ABCD=[1+Z*Y/2 Z;Y*(1+Z*Y/4) 1+Z*Y/2];
case'T'
```

```
 ABCD=[1+Z*Y/2 Z*(1+Z*Y/4);
Y 1+Z*Y/2];
Otherwise
 Error('Invaid type choosen!!!')
 end
VR=VR3ph/sqrt(3)+j*0;
SR=PR+j*QR;
 IR=conj(SR)/(3*conj(VR));
VSIS=ABCD*[VR;IR];
 VS=VSIS(1);
 VS3ph=sqrt(3)*abs(VS);
 IS=VSIS(2);
ISm=1000*abs(IS);
 Pfs=cos(angle(VS)-angle(IS));
 SS=3*VS*conj(IS);
 Reg=(VS3ph-VR3ph)/VR3ph*100;
 Eff=PR/real(SS)*100;
 fprintf('\n IS=%g A',ISm);
fprintf('\n Pfs=%g',Pfs);
 fprintf('\n VS=%g L-L KV',VS3ph);
fprintf('\n PS=%g MW',real(SS));
 fprintf('\n QS=%g MVAR',imag(SS));
fprintf('\n percentage voltage regulation=%g',Reg);
 fprintf('\n percentage transmission line efficiency=%g',Eff);
 fprintf('\n ABCD parameters of transmission line\n');
disp(ABCD);
```

Inputs:

Resistance of the line in ohm=10

Reactance of the line in ohm=20

Voltage at receiving end in KV=66

Real load at receiving end in MW=10

Reactive load at receiving end in MVAR=7.5

Susceptance of shunt line in mho=4e-4

Type (P-Pie/T-tmethod)=T.

Outputs:

```
IS=100.533 A
 Pfs=0.853122
VS=69.5439 L-L KV
 PS=10.331 MW
QS=6.31771 MVAR
 Percentage voltage regulation=5.36958
 Percentage transmission line efficiency=96.7965
ABCD parameters of transmission line
0.9960 + 0.0020i      9.9600 +19.9700i
 0    + 0.0004i           0.9960 + 0.0020i
```

II) Long transmission line:

Statement: Determine the efficiency and regulation of 3-phase, 50Hz, 120km long transmission line delivering 40MW at 132kV at 0.8 lagging pf with following details. Resistance/km/phase=0.2 ohm Inductive reactance/km/phase=1.3mH Capactive susceptance/km/phase=0.01 micro farad

Program:

```
%long transmission line
clear;
 clc;
R=input('resistance of the line in ohm=');
 X=input('reactance of the line in ohm=');
 B=input('susceptance of shunt line in mho=');
VR3ph=input('voltage at receiving end in KV=');
 PR=input('real load at receiving end in MW=');
 QR=input('reactive load at receiving end in MVAR=');
 Z=R+j*X;
 Y=0+j*B;
gamma=sqrt(Z*Y);
 Zc=sqrt(Z/Y);
A=cosh(gamma);
 B=Zc*sinh(gamma);
 C=1/Zc*sinh(gamma);
 D=A;
ABCD=[A B;C D];
VR=VR3ph/sqrt(3)+j*0;
 SR=PR+j*QR;
IR=conj(SR)/(3*conj(VR));
 VSIS=ABCD*[VR;IR];
 VS=VSIS(1);
```

```
VS3ph=sqrt(3)*abs(VS);
 IS=VSIS(2);
 ISm=1000*abs(IS);
 Pfs=cos(angle(VS)-angle(IS));
SS=3*VS*conj(IS);
 Reg=(VS3ph-VR3ph)/VR3ph*100;
 Eff=PR/real(SS)*100;
Val=(A*D)-(B*C);
 fprintf('\n IS=%g A',ISm);
fprintf('\n Pfs=%g',Pfs);
fprintf('\n VS=%g L-L KV',VS3ph);
 fprintf('\n PS=%g MW',real(SS));
 fprintf('\n Qs=%g MVAR',imag(SS));
fprintf('\n percentage voltage regulation=%g',Reg);
 fprintf('\n Efficiency=%g',Eff);
 fprintf('\n A*D-B*C=%g',Val);
fprintf('\n ABCd parameters of transmission line\n');
 disp(ABCD);
```

Inputs:

Resistance of the line in ohm=24
Reactance of the line in ohm=49.0088
Susceptance of shunt line in mho=3.76e-4
Voltage at receiving end in KV=132
Real load at receiving end in MW=40
Reactive load at receiving end in MVAR=30

Outputs:

```
IS=200.922 A
 Pfs=0.830052
VS=149.46 L-L KV
PS=43.1736 MW
Qs=29.007 MVAR
 percentage voltage regulation=13.2272
Efficiency=92.6493
A*D-B*C=1
ABCD parameters of transmission line
 0.9908 + 0.0045i    23.8528 +48.8944i
-0.0000 + 0.0004i    0.9908 + 0.0045i
```

Result:

3. SYMMETRICAL COMPONENT ANALYSIS

AIM: To Analyze symmetrical fault.

Software Required: MATLAB software, R2009a
Windows XP operating system.

Procedure:

1. Open Matlab-->Simulink--> File ---> New---> Model

2. Open Simulink Library and browse the components

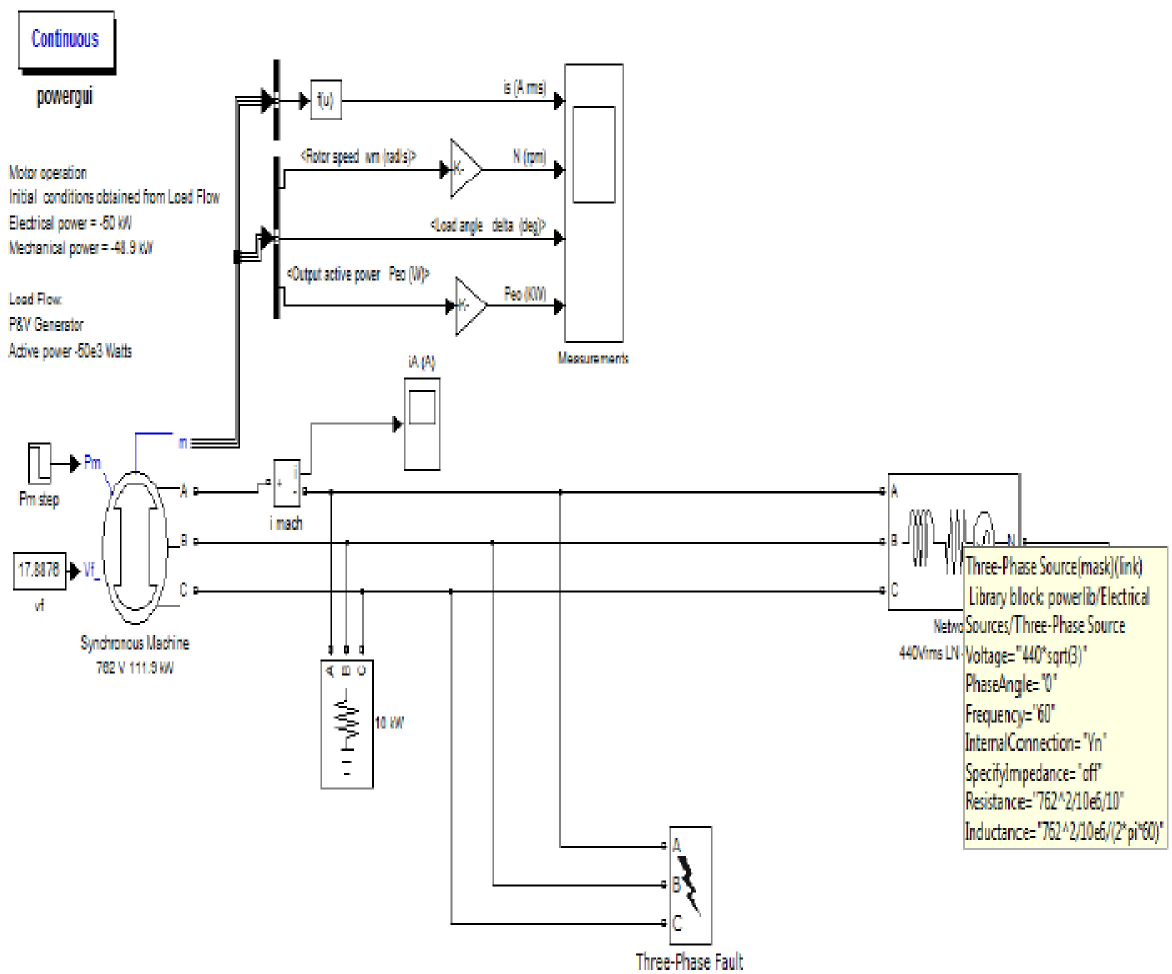
3. Connect the components as per circuit diagram

4. Set the desired voltage and required frequency

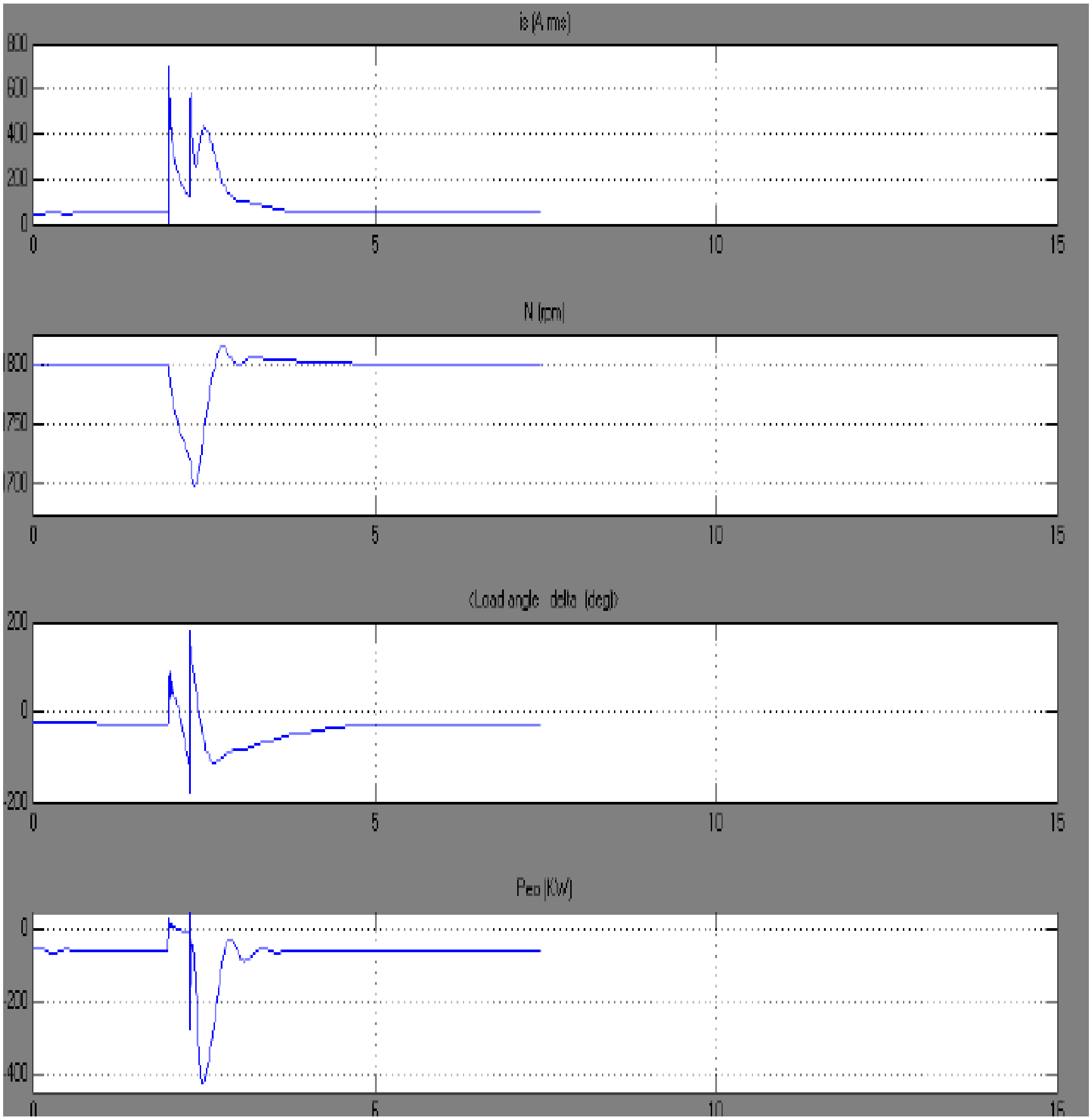
5. Simulate the circuit using MATLAB

6. Plot the waveforms

Circuit Diagram:



Output waveform:



Result:

4. TRANSMISSION LINE FAULT ANALYSIS

AIM: To find the fault current in a given power system where there is

a) Balanced 3-ϕ fault. (LLL/LLLG).

b) Single line to ground fault(LG).

c) Line to line fault(LL).

d) Double line to ground fault(LLG).

Software Required: MATLAB software, R2009a

Windows XP operating system.

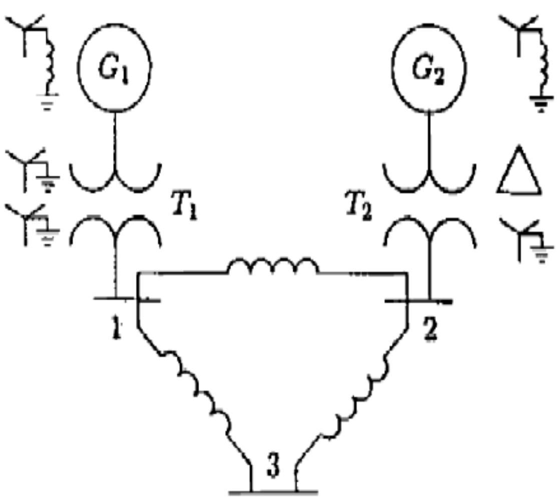
Apparatus Required: Personal Computer (PC).

Procedure:

1. Open the MATLAB software
2. Open the M-file
3. Type the program in editor window
4. Save in current directory as "filename"
5. Compile and run the program

STATEMENT:

For the given power systems shown in fig, the neutral of each generator is grounded through a current limiting reactor of 0.25/3 p.u. on a 100 MVA base. The system data expressed in p.u. on a 100 MVA base is tabulated below. The generators are running on no load at their related voltage and rated frequency with their emfs in phase.



Item	Base MVA	Voltage Rating	X^1	X^2	X^0
G_1	100	20 kV	0.15	0.15	0.05
G_2	100	20 kV	0.15	0.15	0.05
T_1	100	20/220 kV	0.10	0.10	0.10
T_2	100	20/220 kV	0.10	0.10	0.10
L_{12}	100	220 kV	0.125	0.125	0.30
L_{13}	100	220 kV	0.15	0.15	0.35
L_{23}	100	220 kV	0.25	0.25	0.7125

Determine the fault current for the following details of faults.

a) A balanced 3-f fault at bus 3 through a fault impedance Zf = 0.1pu.

b) A Single line to ground fault at bus 3 through a fault impedance Zf = 0.1pu.

c) A line to line fault at bus 3, fault impedance Zf = 0.1pu.

d) A double line to ground fault at bus 3 through a fault impedance Zf = 0.1pu

Program:

```
%program to find fault analysis%
clc;
clear all;
close all;
%positive sequence reactance data%
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];
%zero sequence impedence data%
zdata0=[0 1 0 0.4
0 2 0 0.1
1 2 0 0.3
1 3 0 0.35
2 3 0 0.7125];
%negative sequence reactance=positive reactance%
% zdata2=[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];
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)
```

Outputs:

a) **Balanced three-phase fault(LLL/LLLG)**

```
Enter Faulted Bus No. -> 3
Enter Fault Impedance Zf = R + j*X in complex form (for bolted
faultenter 0). Zf = 0+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
```

b) **Single line to-ground fault (LG)**

```
Enter Faulted Bus No. -> 3

Enter Fault Impedance Zf = R + j*X in

complex form (for bolted fault enter 0). Zf = 0+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
```

c) Line-to-line fault analysis (LL)

```
Enter Faulted Bus No. -> 3

Enter Fault Impedance Zf = R + j*X in

complex form (for bolted fault enter 0). Zf = 0+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
```

d) **Double line-to-ground fault analysis(LLG)**

```
Enter Faulted Bus No. -> 3

Enter Fault Impedance Zf = R + j*X in

complex form (for bolted fault enter 0). Zf = 0+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
```

Results:

\

5. SHORT CIRCUIT ANALYSIS OF POWER SYSTEM MODELS

AIM: Determination of short circuit analysis of power system models.

Software Required: MATLAB software, R2009a

Windows XP operating system.

Apparatus Required: Personal Computer (PC).

Procedure:

1. Open the MATLAB software
2. Open the M-file
3. Type the program in editor window
4. Save in current directory as "filename"
5. Compile and run the program

PROGRAM:

```
%Short circuit fault analysis
clc
clear all
disp('Short circuit fault analysis');

disp('STEP 1');
z1 = input('Enter the Z1 value:');
zbus1 = [z1]
disp('STEP 2');
z2 = input('Enter the z2 value: '); z3 = z1+z2;
zbus = [z1 z1;
           z1 z3]
disp('STEP 3');
z4 = input('Enter the z3 value:');
z5 = z3+z4;
zbus = [z1 z1 z1;
        z1 z3 z3;
        z1 z3 z5]
disp('STEP 4');
z6 = input('Enter the z4 value: ');
z7 = z5+z6;
zbus = [z1 z1 z1 z1;
        z1 z3 z3 z3;
      z1 z3 z5 z5;
      z1 z3 z5 z7]
n=4;
for i=1:4
for j=1:4
zbus(i,j) = zbus(i,j)- ((zbus(i,n)*zbus(n,j))/zbus(n,n));
end
end
```

```
disp('After Elimination:'); zbus(:,4)=[];zbus(4,:)=[]

disp('STEP 5');
z8 = input('Enter the next value: ');
z44 = z8 + zbus(1,1)+zbus(3,3)-
2*zbus(1,3); z41 = zbus(1,1)-zbus(1,3);
z42 = zbus(1,2)-zbus(2,3);
z43 = zbus(1,3)-zbus(3,3); z14 = z41;z24 = z42; z34 = z43;
zbus = [zbus(1,1) zbus(1,2) zbus(1,3) z14;
        zbus(2,1) zbus(2,2) zbus(2,3) z24;
        zbus(3,1) zbus(3,2) zbus(3,3) z34;
         z41        z42       z43     z44]
n=4;
for i=1:4
     for j=1:4
zbus(i,j) = zbus(i,j)- ((zbus(i,n)*zbus(n,j))/zbus(n,n));
end

end
zbus(:,4)=[];
disp('Final Result:');
zbus(4,:)=[]

v = input('Enter the voltage value: ');

disp('Fault at Bus 2');
disp('Fault current:');
 If = v/zbus(2,2)

disp('Post fault Bus voltages:');
v1f = v-(zbus(1,2)*If)
v2f = v-(zbus(2,2)*If)
v3f = v-(zbus(3,2)*If)

disp('Line Flows:');
I13f = (v1f-v3f)/abs(z8)

disp('Fault at Bus 3');
disp('Fault Current:');
If = v/zbus(3,3)

disp('Post fault Voltages:');
v1f = v-(zbus(1,3)*If)
v2f = v-(zbus(2,3)*If)
v3f = v-(zbus(3,3)*If)

disp('Line Flows:');
I13f = (v1f-v3f)/abs(z8)
```

OUTPUT:

```
Short circuit fault analysis
STEP 1
Enter the Z1 value: 1.5j
          zbus1 =
                 0 + 1.5000i
STEP 2
Enter the z2 value: 0.2j
zbus =
  0 + 1.5000i        0 + 1.5000i
  0 + 1.5000i        0 + 1.7000i
STEP 3
Enter the z3 value: 0.15j
zbus =
0 + 1.5000i      0 + 1.5000i  0 + 1.5000i
0 + 1.5000i      0 + 1.7000i  0 + 1.7000i
0 + 1.5000i      0 + 1.7000i  0 + 1.8500i
STEP 4
Enter the z4 value: 1.5j
zbus =
0 + 1.5000i    0 + 1.5000i    0 + 1.5000i    0 + 1.5000i
0 + 1.5000i    0 + 1.7000i    0 + 1.7000i    0 + 1.7000i
0 + 1.5000i    0 + 1.7000i    0 + 1.8500i    0 + 1.8500i
0 + 1.5000i    0 + 1.7000i    0 + 1.8500i    0 + 3.3500i
After Elimination:

 zbus =
         0 + 0.8284i        0 + 0.7388i        0 +  0.6716i
         0 + 0.7388i        0 + 0.8373i        0 +  0.7612i
         0 + 0.6716i        0 + 0.7612i        0 +  0.8284i

STEP 5
Enter the next value: 0.3j
   0 + 0.8284i     0 + 0.7388i     0 + 0.6716i     0 + 0.1567i
   0 + 0.7388i     0 + 0.8373i     0 + 0.7612i     0 - 0.0224i
   0 + 0.6716i     0 + 0.7612i     0 + 0.8284i     0 - 0.1567i
   0 + 0.1567i     0 - 0.0224i     0 - 0.1567i     0 + 0.6134i
Final Result:
```

```
zbus =
 0 + 0.7883i        0 + 0.7445i        0 + 0.7117i
 0 + 0.7445i        0 + 0.8365i        0 + 0.7555i
 0 + 0.7117i        0 + 0.7555i        0 + 0.7883i

Enter the voltage value: 1 Fault at Bus 2
Fault current:
If =                0 - 1.1955i

Post fault Bus voltages:

v1f = 0.1099
v2f = 0
v3f = 0.0969

 Line Flows:
 I13f =   0.0436

Fault at Bus 3 Fault Current:
If =  0 - 1.2685i

Post fault
Voltages: v1f =     0.0972
v2f =               0.0417
v3f =               0

Line Flows:

I13f =  0.3241
```

Result:

6. SPEED CONTROL OF DC MOTOR

AIM: To construct three speed control dc motor using BJT-H bridge simulation using MATLAB.

Software Required: MATLAB software, R2009a

Windows XP operating system.

Apparatus Required: Personal Computer (PC).

Procedure:

1. Open Matlab-->Simulink--> File ---> New---> Model

2. Open Simulink Library and browse the components

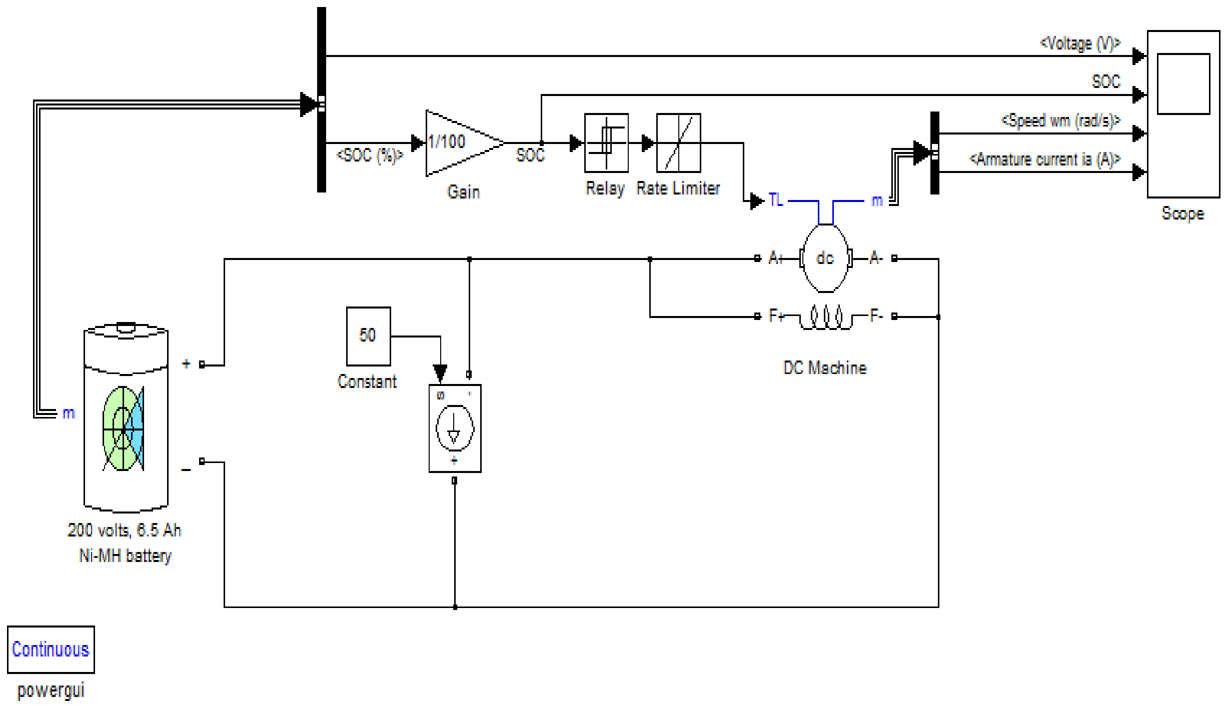
3. Connect the components as per circuit diagram

4. Set the desired voltage and required frequency

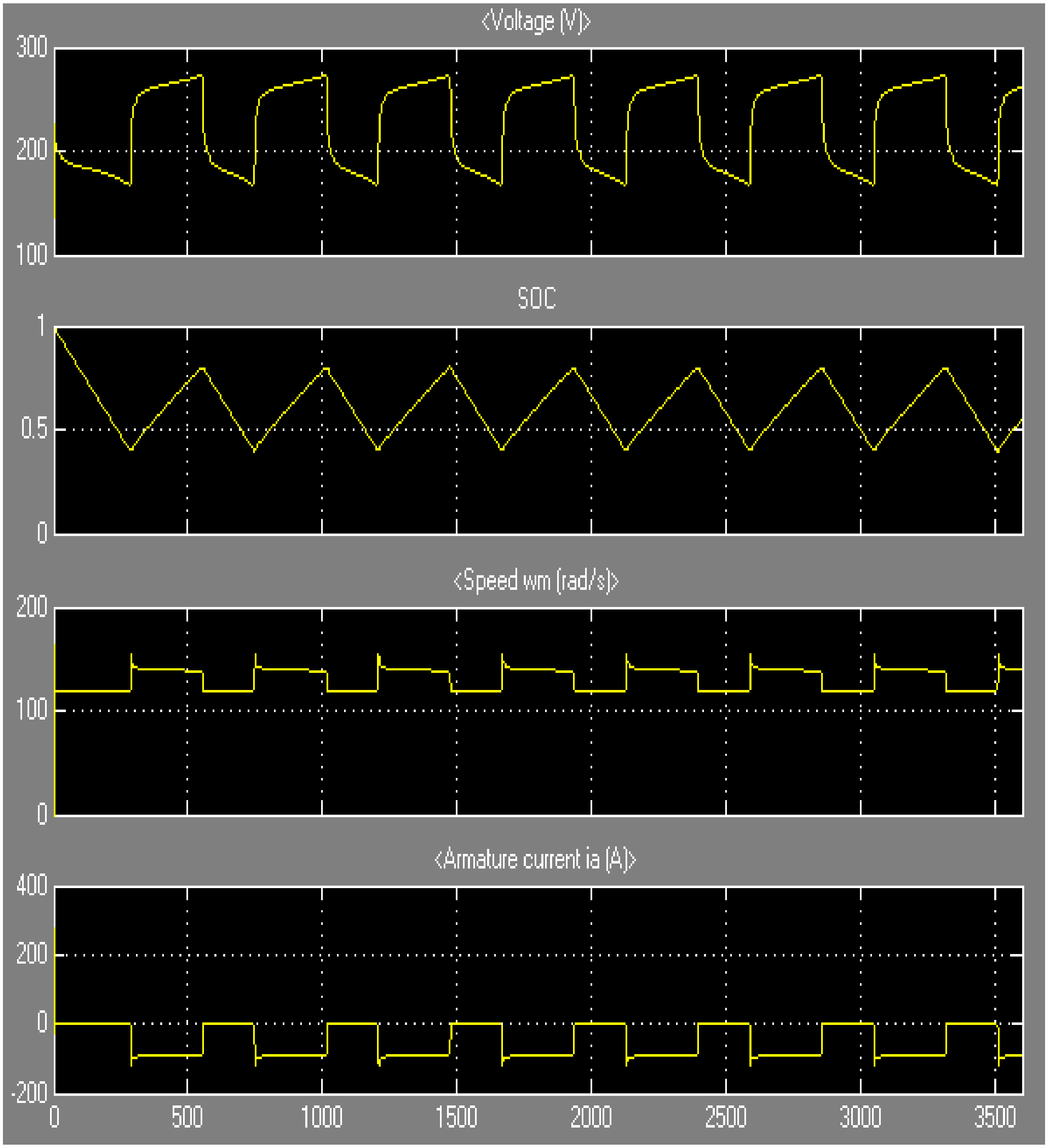
5. Simulate the circuit using MATLAB

6. Plot the waveforms

Circuit Diagram:



Output:



Result:

7. SPEED CONTROL OF INDUCTION MOTOR

AIM: Determination of speed control of induction motor.

Software Required: MATLAB software, R2009a

Windows XP operating system.

Apparatus Required: Personal Computer (PC).

Procedure:

1. Open Matlab-->Simulink--> File ---> New---> Model

2. Open Simulink Library and browse the components

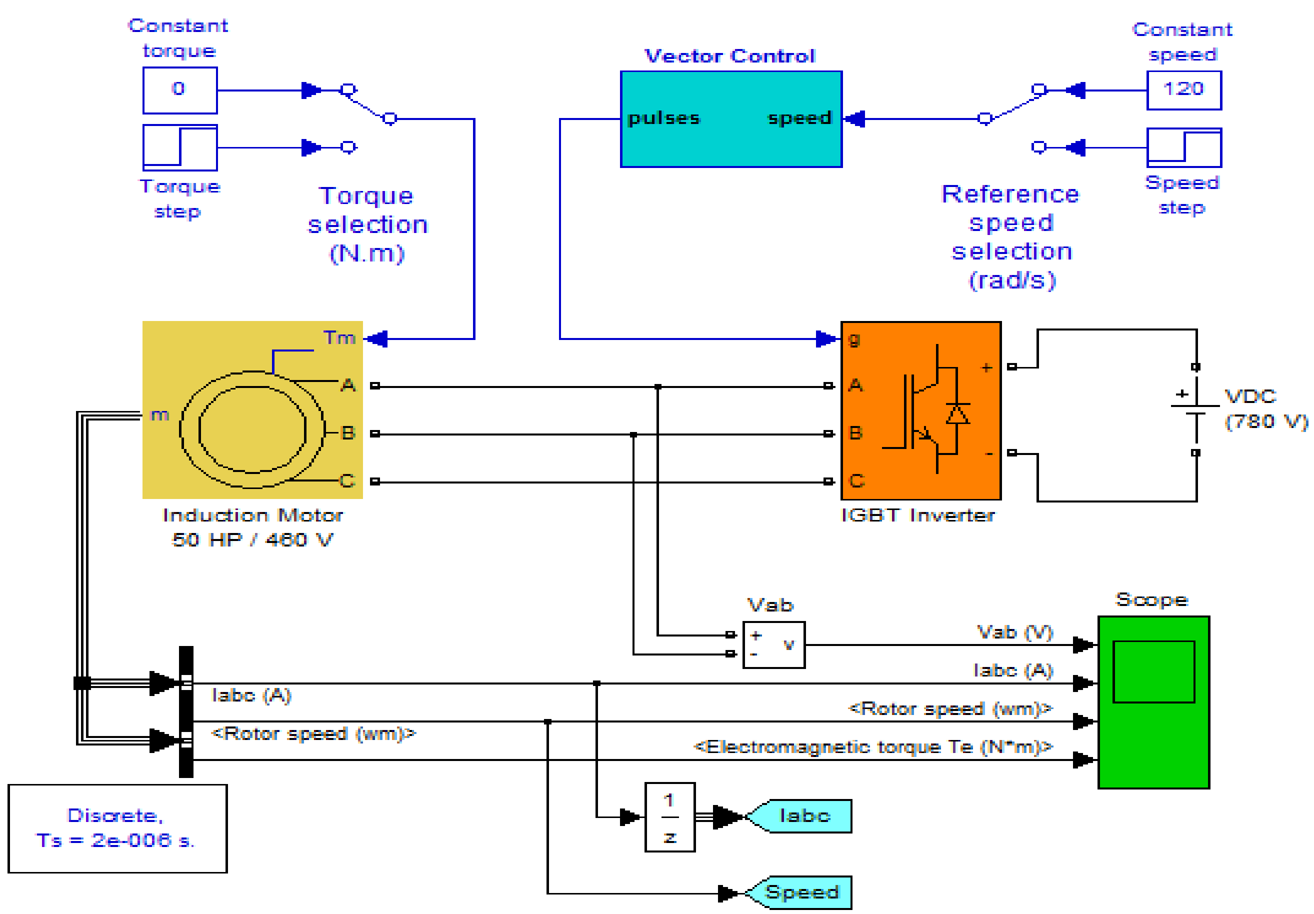
3. Connect the components as per circuit diagram

4. Set the desired voltage and required frequency

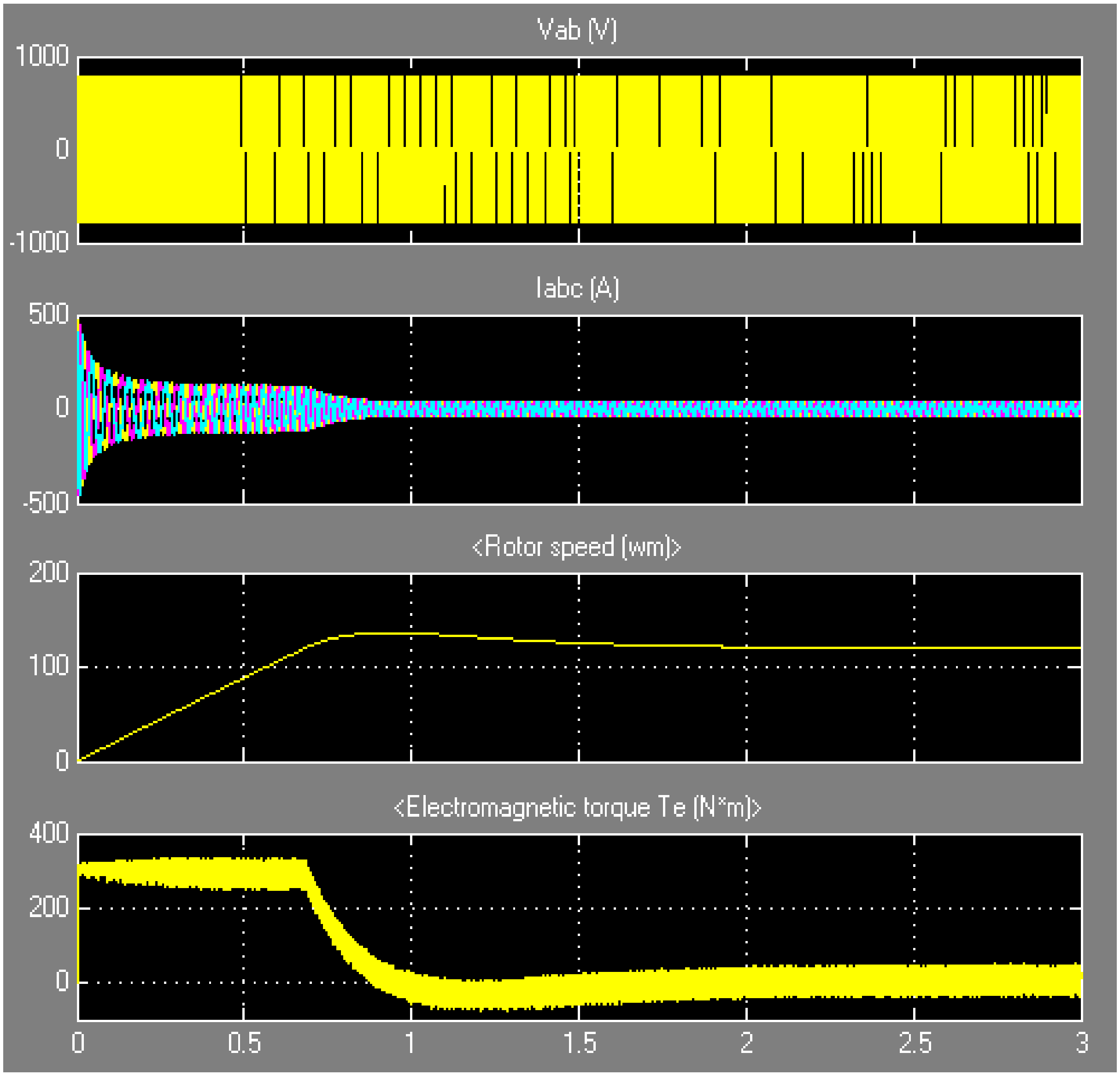
5. Simulate the circuit using MATLAB

6. Plot the waveforms

Circuit Diagram:



Outputs:



Result:

8. DESIGN AND ANALYSIS OF FEEDBACK CONTROL SYSTEM

AIM: Design and Analysis of unity feedback control system with forward path transfer function $G(s) = \frac{1}{s^2+10s+20}$, the effect of addition of a PI controller on the system performance.

Software Required: MATLAB software, R2009a
Windows XP operating system.

Apparatus Required: Personal Computer (PC).

Procedure:

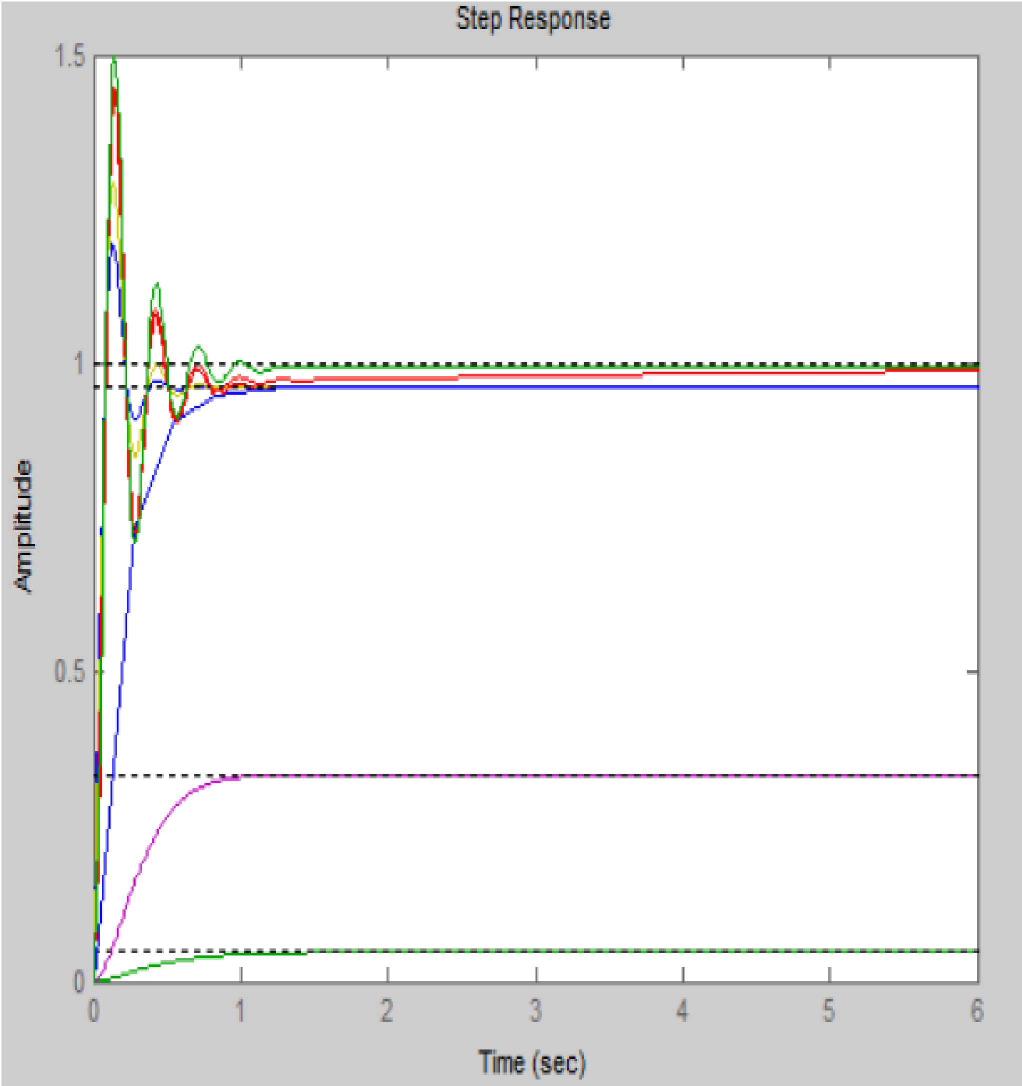
1. Open the MATLAB software
2. Open the M-file
3. Type the program in editor window
4. Save in current directory as "filename"
5. Compile and run the program

Program:

```
num=1;
den=[1 10 20];
g1=tf (num,den)
t1=feedback(g1,1)
step(t1,'g')
hold on
num1=10;
den1=[1 10 20];
g2=tf (num1,den1)
t2=feedback(g2,1)
step(t2,'m')
hold on
Kp=500;
Ki = 1
numc=[Kp Ki];
denc= [1 0]
```

```
numo=conv(numc,num)
deno=conv(den,denc)
g3=tf(numo,deno)
t3=feedback(g3,1)
step(t3,'b')
hold on
Kp=500;
Ki = 100
numc=[Kp Ki];
denc= [1 0]
numo=conv(numc,num)
deno=conv(den,denc)
g3=tf(numo,deno)
t4=feedback(g3,1)
step(t4,'r')
hold on
Kp=500;
Ki = 500
numc=[Kp Ki];
denc= [1 0]
numo=conv(numc,num)
deno=conv(den,denc)
g3=tf(numo,deno)
t5=feedback(g3,1)
step(t5,'g')
hold on
```

Output:



Result:

9. LOAD FREQUENCY CONTROL OF SINGLE AREA AND TWO AREA POWER SYSTEM

Aim: Design and analysis of load frequency control of single area and two area power system

Software Required: MATLAB software, R2009a
Windows XP operating system.

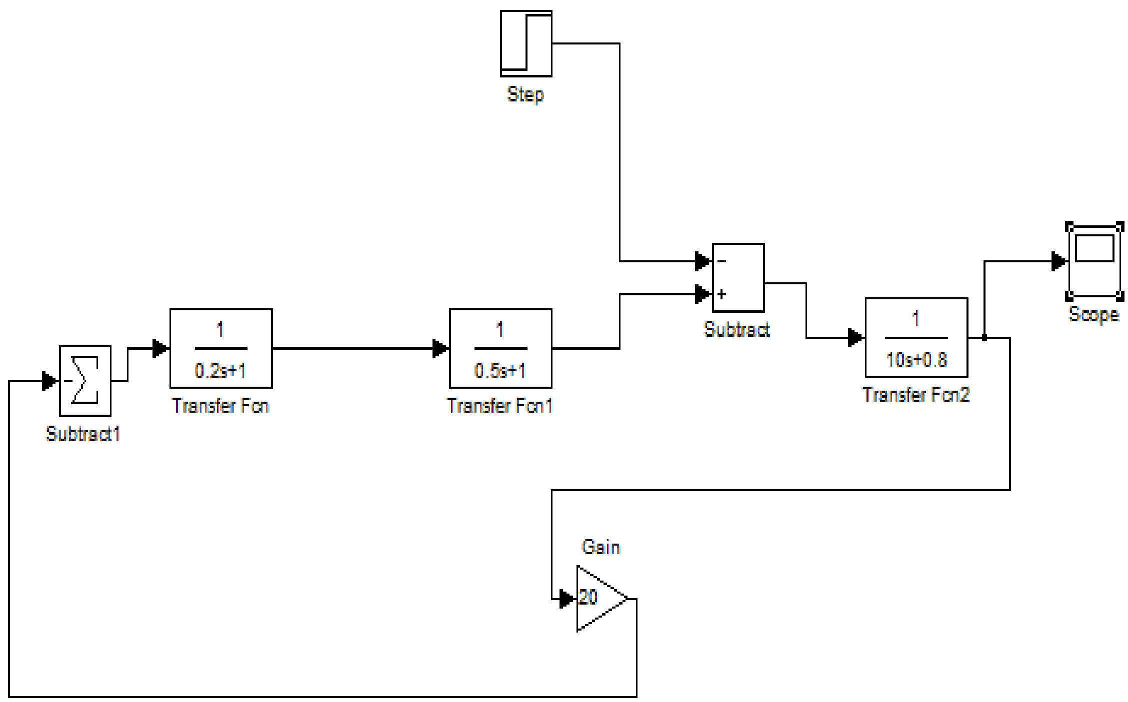
Apparatus Required: Personal Computer (PC).

Procedure:

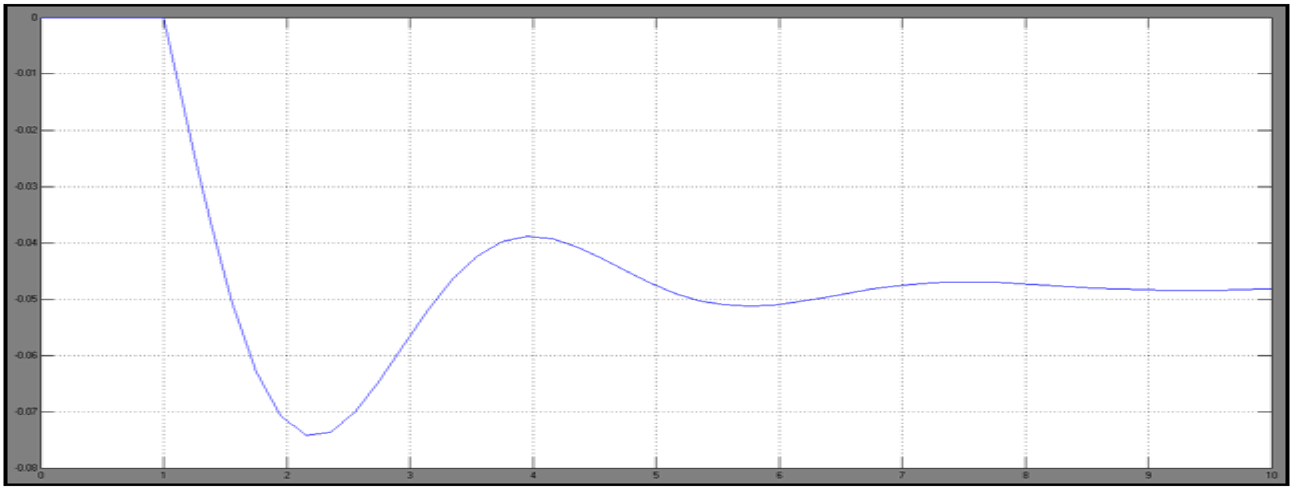
1. Open Matlab-->Simulink--> File ---> New---> Model
2. Open Simulink Library and browse the components
3. Connect the components as per circuit diagram
4. Set the desired voltage and required frequency
5. Simulate the circuit using MATLAB
6. Plot the waveforms

Circuit Diagram:

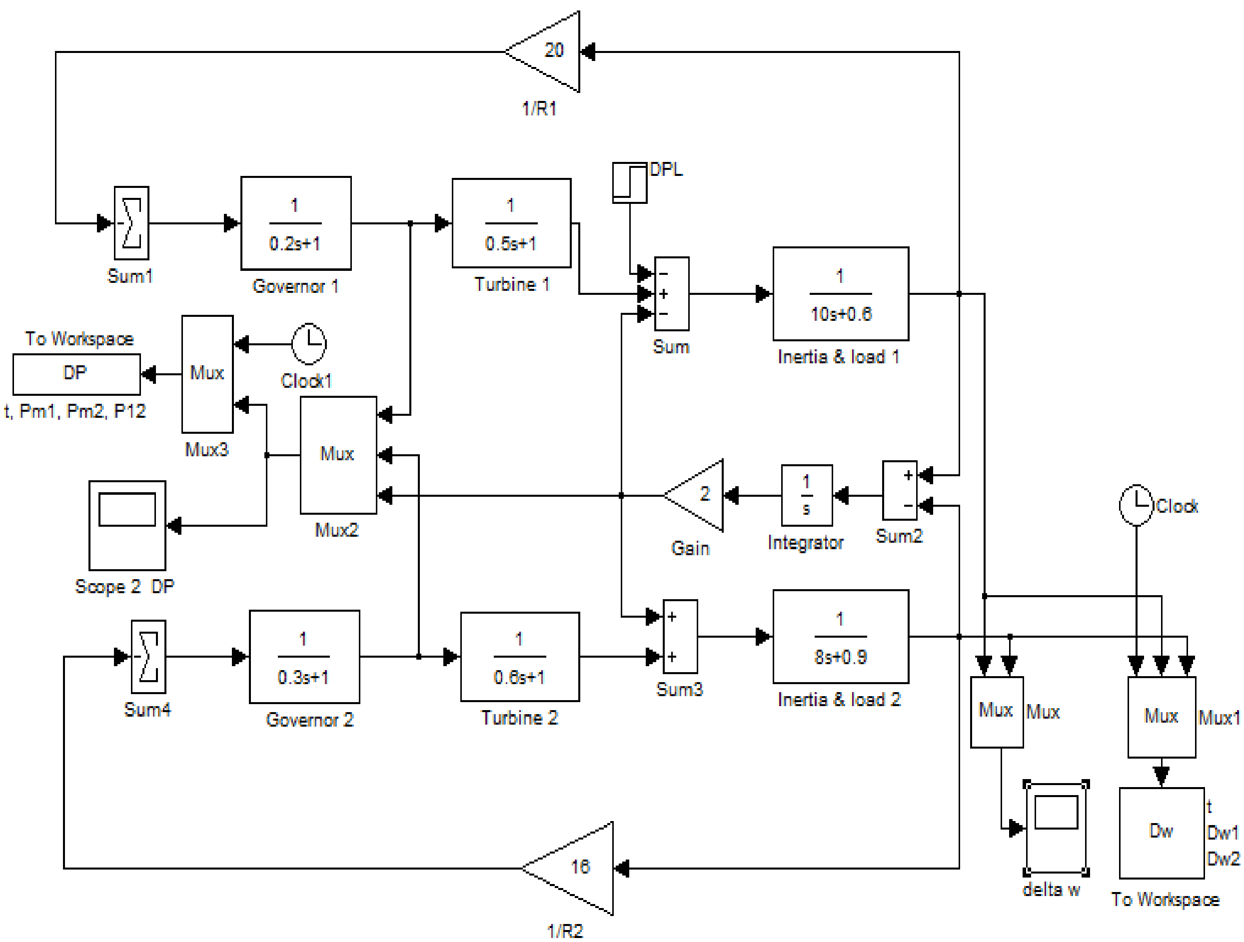
I) LOAD FREQUENCY CONTROL OF SINGLE AREA SYSTEM:



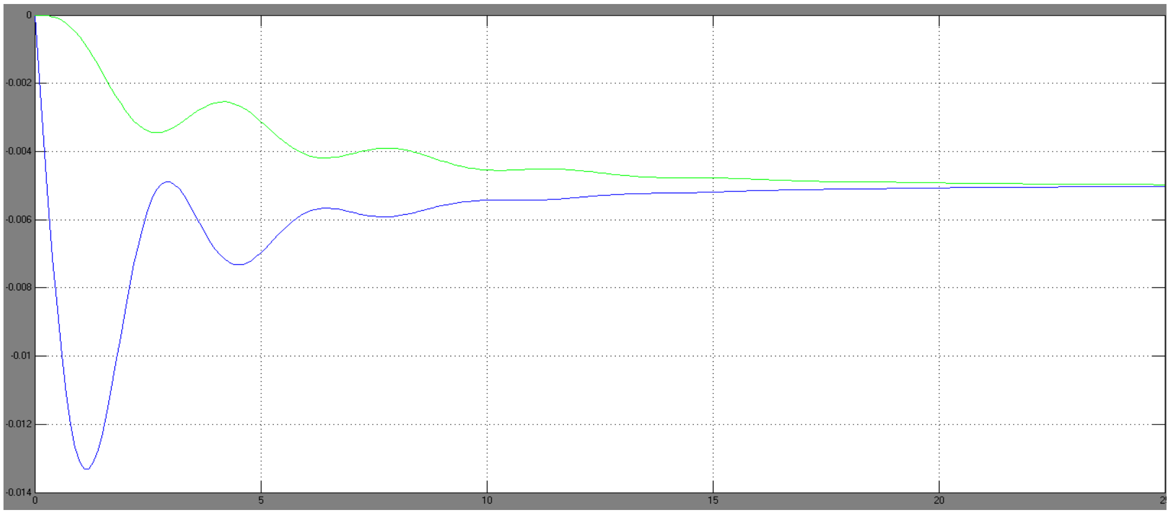
Output:

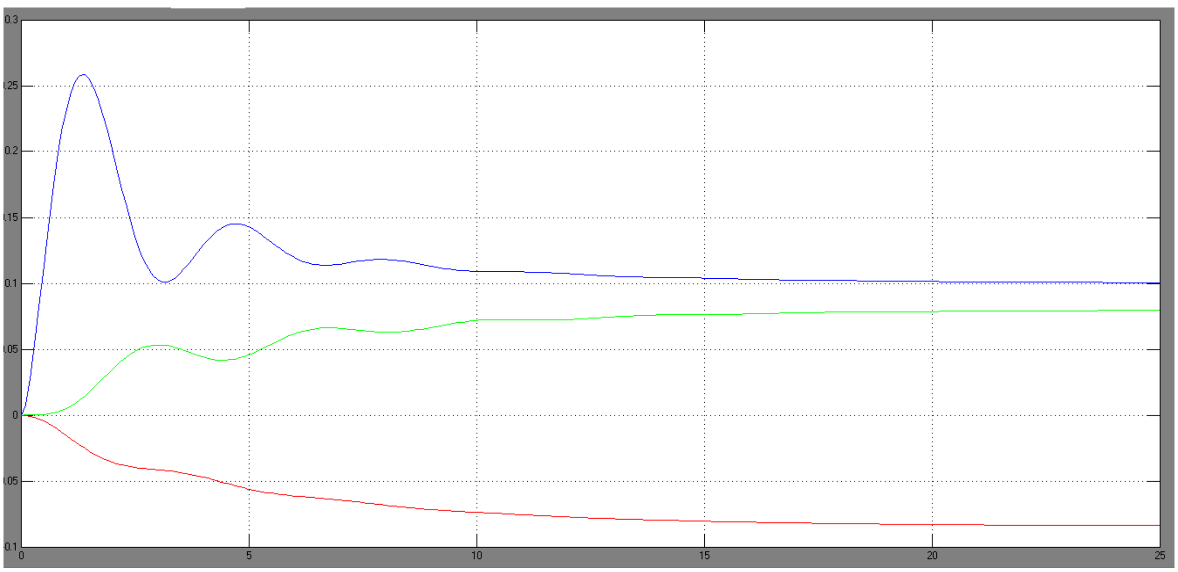


II) LOAD FREQUENCY CONTROL OF TWO AREA SYSTEM:



Output:





Result:

10.ECONOMIC DISPATCH OF THERMAL LOAD

AIM: Determination of economic dispatch of thermal load.

Software Required: MATLAB software, R2009a

Windows XP operating system.

Apparatus Required: Personal Computer (PC).

Procedure:

1. Open the MATLAB software
2. Open the M-file
3. Type the program in editor window
4. Save in current directory as "filename"
5. Compile and run the program

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;
Power Shared=p
 Lambda=la
Loss=LS
```

Outputs:

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

Result:

11.DESIGN OF SINGLE AND THREE PHASE INVERERS

AIM: Determination of economic dispatch of thermal load.

Software Required: MATLAB software, R2009a

Windows XP operating system.

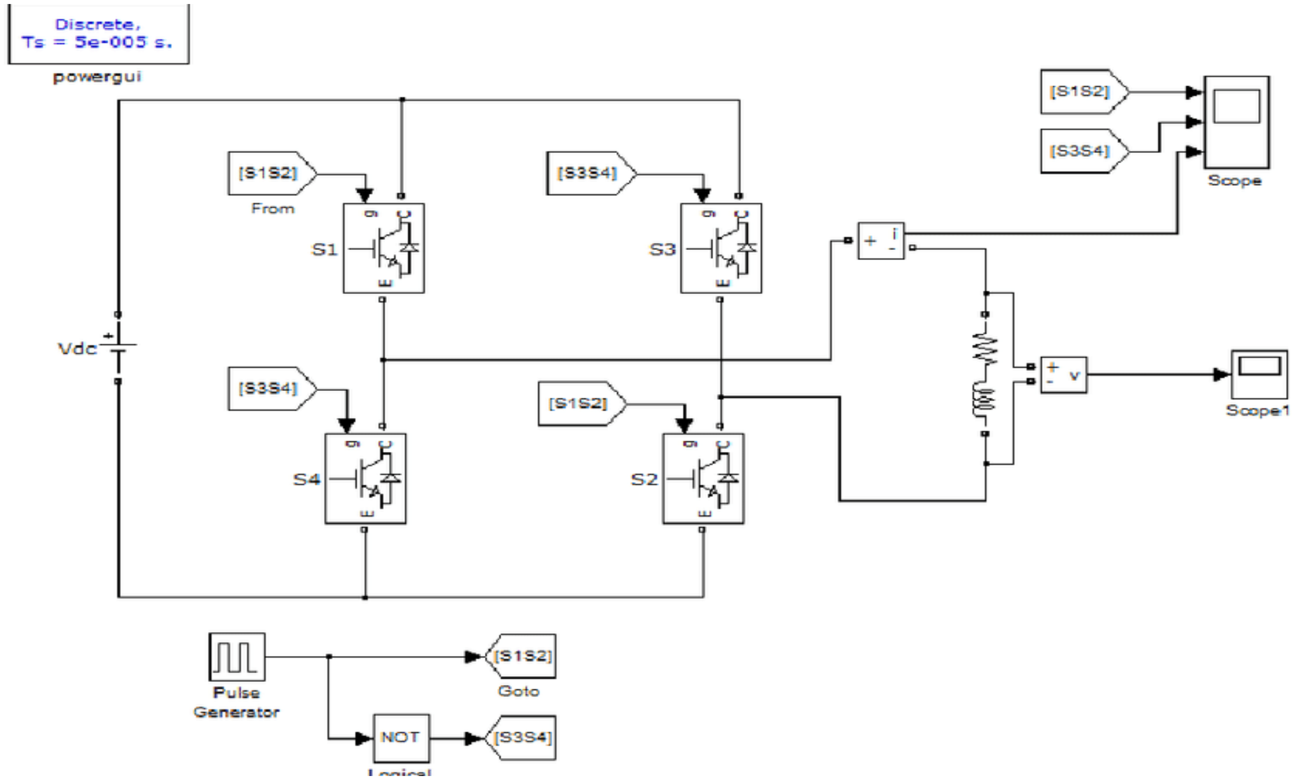
Apparatus Required: Personal Computer (PC).

Procedure:

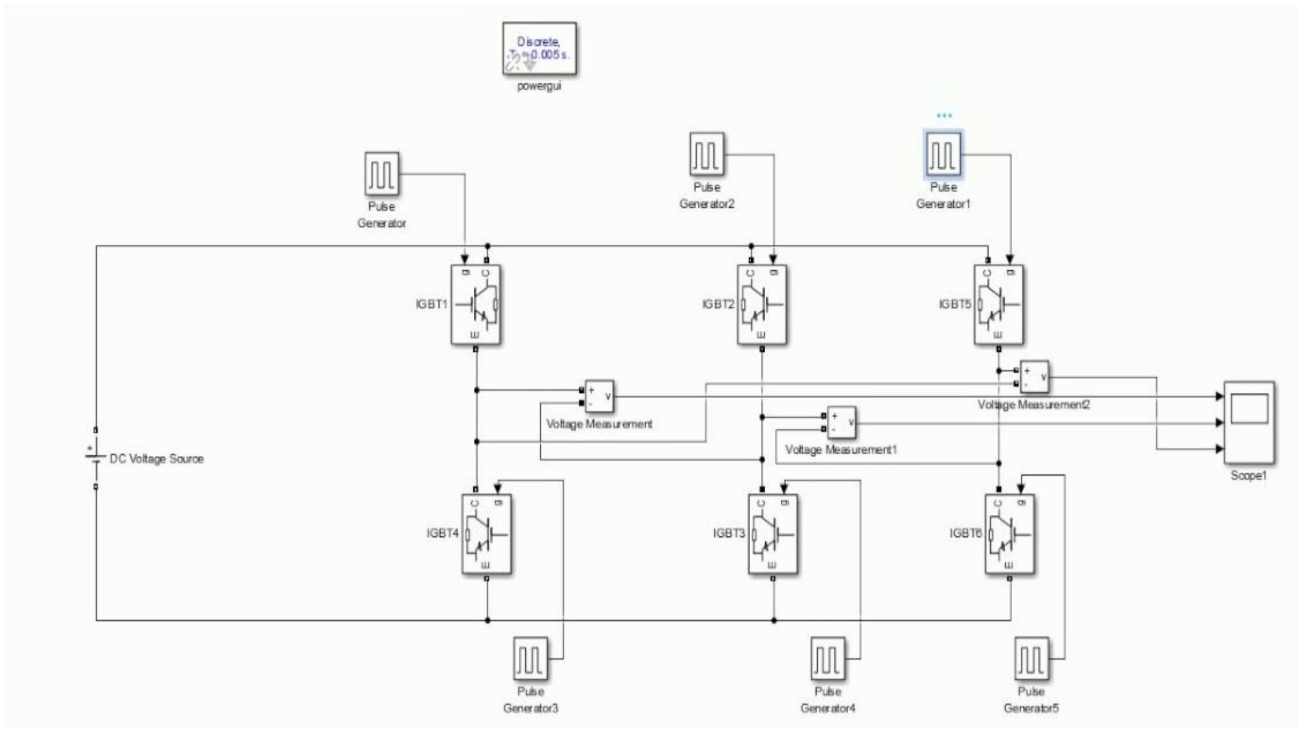
1. Open the MATLAB software
2. Open the NEW MODEL IN SIMULINK
3. Design the circuit as shown in circuit diagram
4. Save the model and give appropriate values
5. Compile and run the model observe the output results in the scope

Circuit diagram:

Single phase inverter:



Three phase inverter:



Outputs:

Result:

12.DESIGN OF SINGLE AND THREE PHASE FULL CONVERTERS

AIM: Determination of economic dispatch of thermal load.

Software Required: MATLAB software, R2009a

Windows XP operating system.

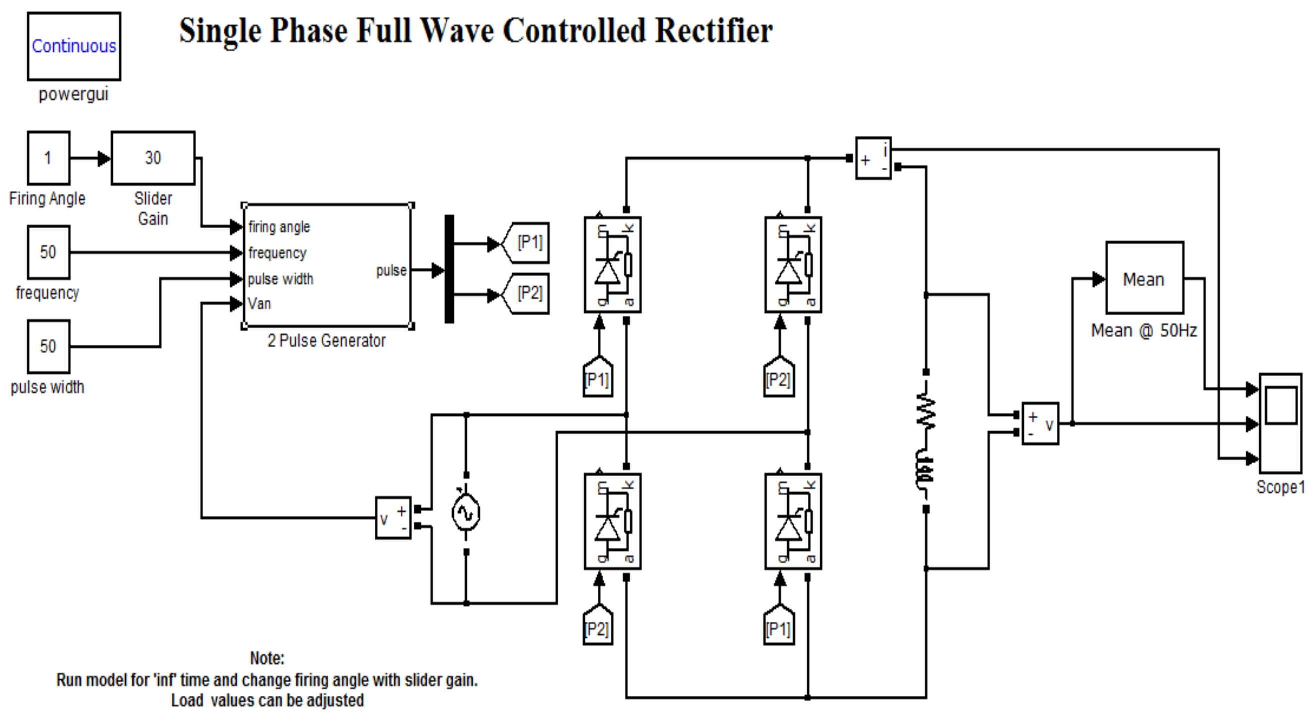
Apparatus Required: Personal Computer (PC).

Procedure:

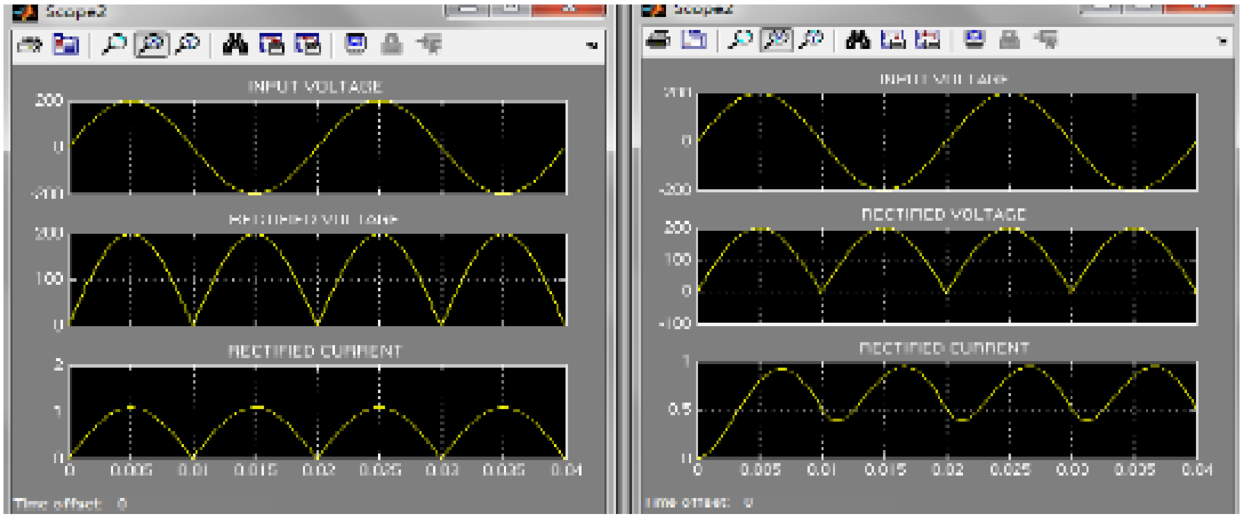
1. Open the MATLAB software
2. Open the NEW MODEL IN SIMULINK
3. Design the circuit as shown in circuit diagram
4. Save the model and give appropriate values
5. Compile and run the model observe the output results in the scope

Circuit diagram:

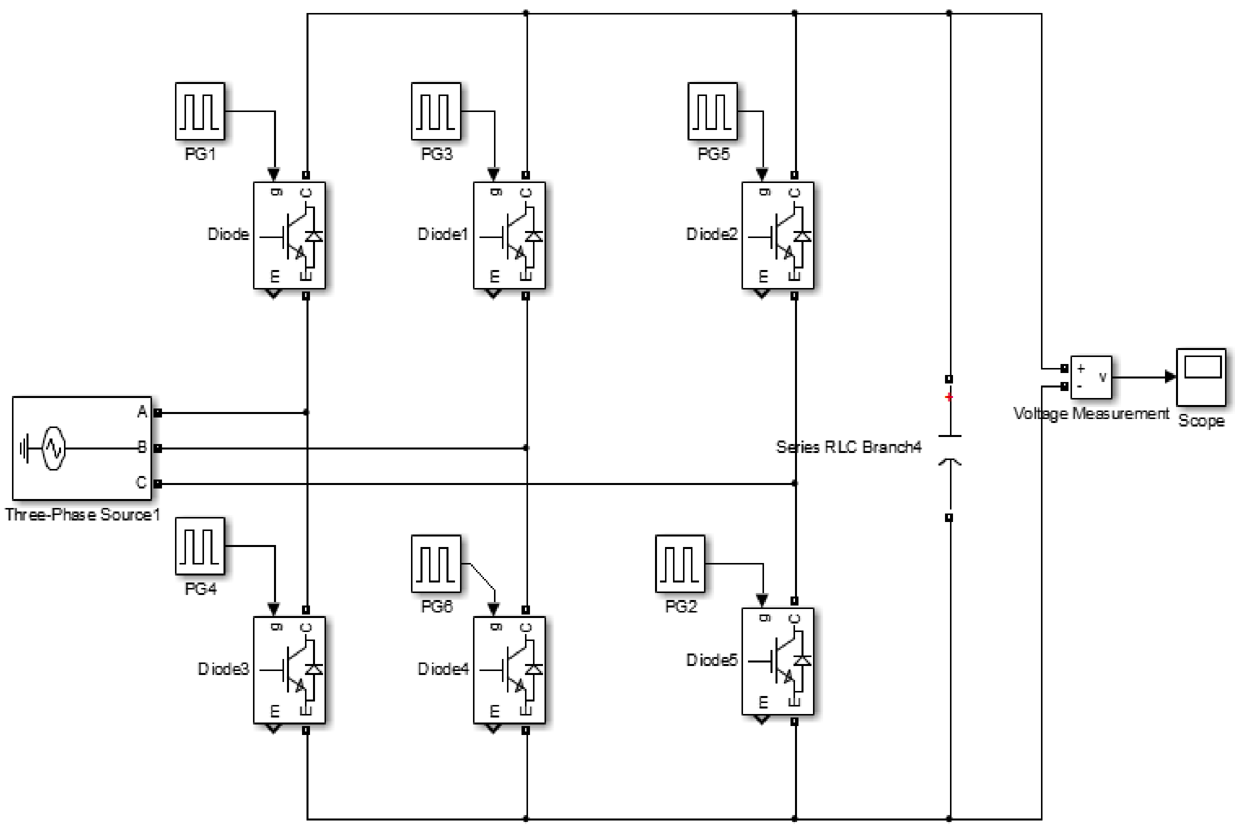
Single phase full converter:



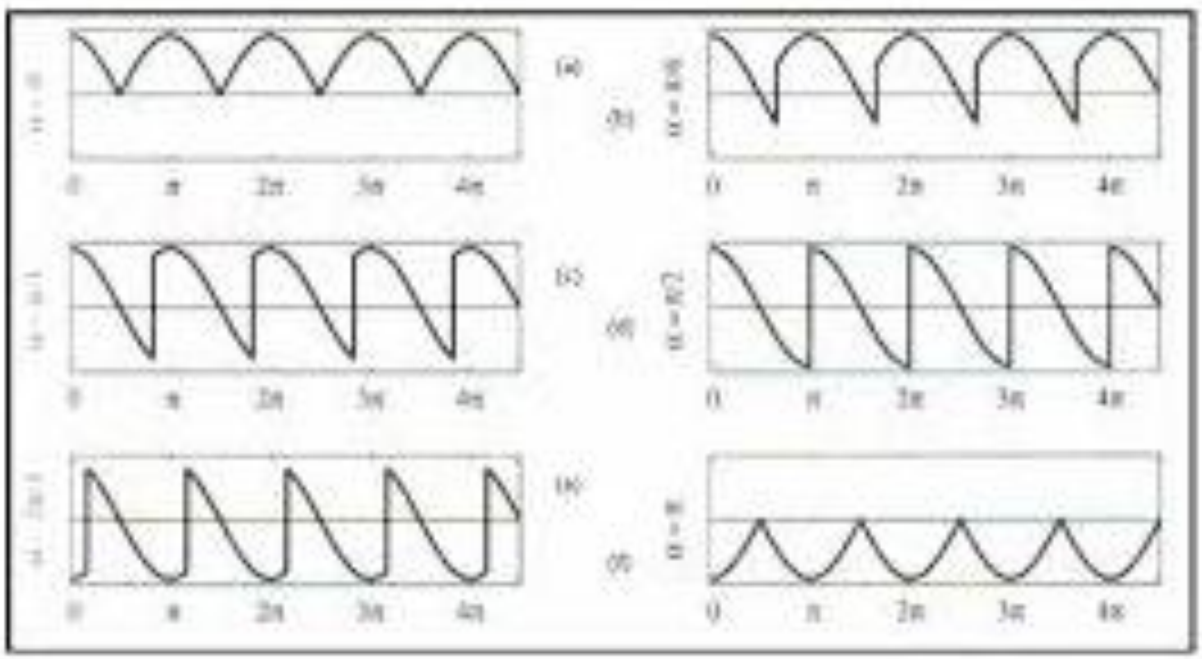
Outputs:



Three phase full converter:



Outputs:



Result: