

**SEMBODAI RUKMANI VARATHARAJAN ENGINEERING COLLEGE,
SEMBODAI, NAGAPATTINAM.
DEPARTMENT OF ELECTRICAL AND ELECTRONICS**



VII- Semester – B.E (EEE)

EE2404 POWER SYSTEM SIMULATION LAB

Prepared by,

Mr.R.Dhineshkumar M.E., AP/EEE

Date: 18.06.2015

EE2404 – POWER SYSTEM SIMULATION LABORATORY

1. Computation of line parameters and Modeling of Transmission Lines using MATLAB
2. Formation of Bus Admittance and Impedance Matrices and Solution of Networks using MATLAB
3. Load Flow Analysis I - Solution of Load Flow and Related Problems Using Gauss-Seidel Method using MATLAB
4. Load Flow Analysis II - Solution of Load Flow and Related Problems Using Newton-Raphson and Fast-Decoupled Methods using MATLAB
5. Fault Analysis of AC Power System using MATLAB
6. Transient and Small Signal Stability Analysis: Single-Machine Infinite Bus System using MATLAB SIMULINK
7. Transient Stability Analysis of Multi-machine Power Systems using MATLAB
8. Electromagnetic Transients in Power Systems using MATLAB
9. Load-Frequency Dynamics of Single-Area and Two-Area Power Systems using MATLAB SIMULINK
10. Economic Dispatch in Power Systems using MATLAB

GENERAL INSTRUCTIONS

1. Lab users should maintain professional and courteous communication. Electronic devices should be used on a professional level. No obnoxious or belligerent behavior will be tolerated.
2. Activities in the lab(s) that are considered by the lab aides to be abusive to the software, hardware, and/or personnel may result in expulsion from the lab(s) and denial of future use of the lab(s).
3. Software may be installed by Computer Labs staff only. Do not install any software on your own. Files not put on by Computer Labs staff will be routinely removed.
4. Do not modify any software or files. Do not overwrite the operating system, modify the autoexec.bat or config.sys or any other system parameters.
5. Respect the equipment. Do not remove or disconnect parts, cables, or labels.
6. Personal Internet use for chat rooms, instant messaging (IM), or email is strictly prohibited.
7. Do not personalize the computer settings. (This includes desktop, screen saver, etc.)
8. Chewing gum, food, or drinks are not allowed in the computer lab or anywhere near a computer.
9. Students must not disturb or distract others by conducting themselves in an unacceptable manner.
10. Log-off — leave the computer ready for the next person to use. Pick-up your materials and push in the chair.

CONTENTS

S.No	Date	Name of the Experiment	Page No	Marks	Remarks
1.					
2.					
3.					
4.					
5.					
6.					
7.					
8.					
9.					
10.					

Average marks / Grade :

Signature of Lab In-charge :

EX.NO:1 COMPUTATION OF PARAMETERS AND MODELING OF TRANSMISSION LINES

AIM:

To develop a program in MATLAB for determining the transmission line parameters and verify using MATLAB simulation.

SOFTWARE REQUIRED:

MATLAB software package

FORMULAE:

Single phase Inductance = $10^{-7}[1+4\log(d/r)]$

Capacitance = $3.14*8.854*10^{-12}/\log(d/r)$

Three phase Inductance = $10^{-7}[0.5+2\log(d/r)]$

Capacitance = $2*3.14*8.854*10^{-12}/\log(d/r)$

$D_{\text{equivalent}} = [d1*d2*d3]^{1/3}$

where

d = spacing of conductors

r = radius of conductors

ALGORITHM:

- STEP 1: Find that the given transmission line is single phase or three phase
- STEP 2: If it is single phase, get the value of distance between the conductors.
- STEP 3: Get the radius of the conductor
- STEP 4 : Using the appropriate formula, find inductance and capacitance.
- STEP 5: If the given system is three phase, classify whether it is symmetrical or unsymmetrical
- STEP 6: If symmetrical, get the distance between the conductors and radius of the

conductor

STEP 7: Using the appropriate formula, find inductance and capacitance

STEP 8: If unsymmetrical, get the distance between the conductors and radius of the conductor. Using the appropriate formula, find inductance and capacitance.

PROBLEM:

Determine the sending end voltage, current, power & power factor for a **160km** section of 3phase line delivering **50MW** at **132kV** and P.F **0.8** lagging. Also find the efficiency and regulation of the line. Resistance per line **0.1557ohm** per km, spacing **3.7m, 6.475m, 7.4m** transposed. Evaluate the A, B, C, D parameters also. Diameter **1.956cm**.

Write and execute a MATLAB program and also verify the output with the manual calculation results.

PROGRAM :

```
clc;
clear all;
ab=input('value of ab');
bc=input('value of bc');
ca=input('value of ca');
pr=input('receiving end power in MW');
vr=input('receiving end voltage in kv');
pfr=input('receiving end power factor');
l=input('length of the line in km');
r=input('resistance/ph/km');
f=input('frequency');
D=input('diameter in m');
rad=D/2;
newrad=(0.7788*rad);
deq=(ab*bc*ca)^(1/3);
L=2*10^(-7)*log(deq/rad);
C=(2*pi*8.854*10^-12)/log(deq/rad);
XL=2*pi*f*L*l*1000;
rnew=r*l;
Z=rnew+1i*(XL);
Y=1i*(2*pi*f*C*l*1000);
a=1+((Z*Y)/2);
d=a;
b=Z;
c=Y*(1+((Z*Y)/4));
vrph=(vr*10^3)/1.732;
```



```

irold=(pr*10^6)/(1.732*vr*10^3*0.8);
k=sin(acos(pfr));
ir=irold*(pfr-(1j*k));
vs=(a*vrph)+(b*ir);
is=(c*vrph)+(d*ir);
angle(vs);
angle(is);
e=angle(vs);
u=angle(is);
PFS=cos(e-u);
eff=((pr*10^6)/(3*abs(vs)*abs(is)*PFS))*100;
reg=((abs(vs)/abs(a))-abs(vrph))/abs(vrph))*100;
L
C
XL
rnew
a
b
c
d
Vs=abs(vs)
Is=abs(is)
voltage_angle=angle(vs)*180/pi
current_angle=angle(is)*180/pi
PFS
eff
reg

```

EXECUTION:







RESULT:

Thus the computation of line parameters of a given power transmission system is done by using MATLAB and the output is verified.

EX.NO:2A FORMATION OF BUS ADMITTANCE MATRIX

AIM:

To form the bus admittance matrix (Y_{bus}) for a given power system using MATLAB program.

SOFTWARE REQUIRED:

MATLAB software package

ALGORITHM:

STEP1: Read all the data namely R and X for the system.

STEP2: Calculate the mutual or transfer reactance for the reactance between i and j and $i=j=1, 2, 3, 4, \dots$

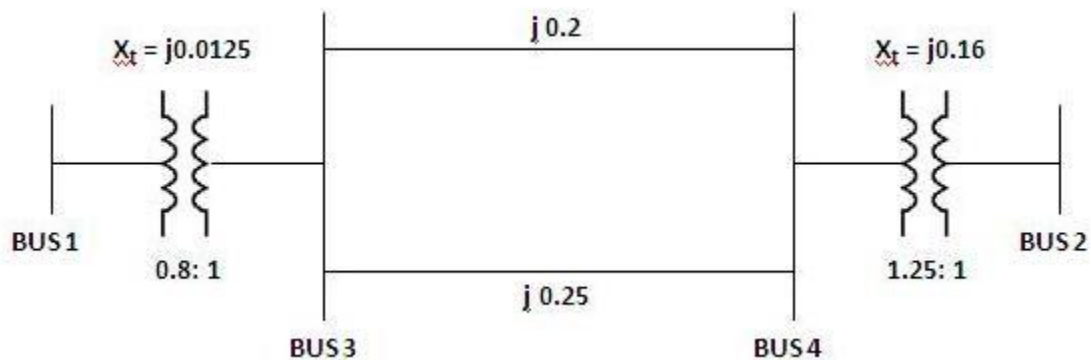
STEP3: Calculate the self- admittance or point admittance bus $i=1, 2, 3, 4, \dots$

STEP4: Output the Y-bus matrix.

STEP5: Print the result

PROBLEM

Form the Ybus matrix for the given power system. Write and execute a MATLAB program and also verify the output with the manual calculation results.



PROGRAM:

```
clear all;
clc;
n= input('Enter the number of buses');
fprintf('Enter your choice');
p= input ('1. impedance, 2. admittance');
if (p==1)
for q= 1:n
for r=q+1:n
fprintf('Enter the impedance value between %d-%d',q,r);
z(q,r)=input(':');
if (z(q,r)==0)
y(q,r)=0;
else
y(q,r)=inv(z(q,r));
end
y(r,q)= y(q,r);
fprintf('enter the half line charging admittance');
x(q,r)= input(':');
x(r,q)=x(q,r);
end
end
elseif (p==2)
for a= 1:n
for b=a+1:n
fprintf('Enter the admittance value between %d-%d',a,b);
y(a,b)=input(':');
y(b,a)= y(a,b);
fprintf('enter the half line charging admittance');
x(a,b)= input(':');
x(b,a)=x(a,b);
end
end
else
fprintf('enter the correct choice');
end
tr=zeros(n,n);
fprintf('Off-Nominal Tap ratio exists in the system? Enter 1.Yes
2.No');
o= input(':');
if(o==1)
for k=1:n
for l=k+1:n
fprintf('Transformer tap for %d-%d exists? 1.Yes 2.No',k,l)
g=input(':');
if(g==1)
fprintf('Enter the choice 1.1:a 2.a:1  ?')
d=input(':');
```

```

fprintf('Enter the tap ratio value(a) between %d-%d',k,1);
t(k,1)=input(':');
if (t(k,1)==0)
tr(k,k)=0;
tr(k,1)=0;
else
yse=y(k,1)/t(k,1);
ysh1=((1-t(k,1))/(t(k,1)^2))*y(k,1);
ysh2=((t(k,1)-1)/t(k,1))*y(k,1);
end
if (d == 1)
tr(k,k)=yse+ysh2;
tr(1,1)=yse+ysh1;
tr(k,1)=yse;
tr(1,k)=tr(k,1);
else
tr(k,k)=yse+ysh1;
tr(1,1)=yse+ysh2;
tr(k,1)=yse;
tr(1,k)=tr(k,1);
end
y(k,1)=0;
y(1,k)=y(k,1);
end
end
end
end
for s=1:n
fprintf('enter the self admittance of the bus %d',s);
u(s)=input(':');
end
ybus=zeros(n,n);
for a = 1:n
for b=1:n
if (a==b)
for c = 1:n
ybus(a,a)= ybus(a,a)+ y(a,c)+x(a,c);
end
else
ybus(a,b)=-y(b,a);
end
end
ybus(a,a)=ybus(a,a)+u(a);
end
for r=1:n
for h = 1:n
if(r==h)
ybus(r,r)=ybus(r,r)+tr(r,r);
else

```



```
ybus(r,h)=- (y(r,h)+tr(r,h)) ;  
end  
end  
end  
ybus
```

EXECUTION:



**RESULT:**

Thus, the formation of bus admittance matrix is done by using MATLAB and the output is verified for the given power system.

EX.NO:2B**FORMATION OF BUS IMPEDANCE MATRIX****AIM:**

To form the bus impedance matrix (Z_{bus}) for a given power system using MATLAB program.

SOFTWARE REQUIRED:

MATLAB software package

ALGORITHM:

It is the step by step programmable technique which places branch by branch. It is the advantage that modification of the network does not require complete rebuilding of Z_{bus} .

STEP1: Start the new document, give a impedance value

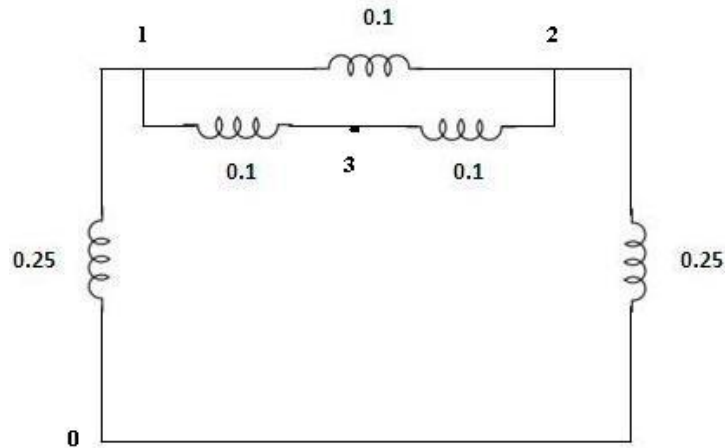
STEP2: Choose the type of bus in a network

- 1- Start the new network.
- 2- Adding a new bus to reference bus
- 3- Adding a new bus to already existing bus
- 4- Adding a new element between already existing bus and reference bus
- 5- Adding a element between two existing bus

STEP3: Build the Z_{bus} for the given network & print the Z_{bus} matrix

PROBLEM:

Form the Z_{bus} matrix for the given power system. Write and execute a MATLAB program and also verify the output with the manual calculation results.



PROGRAM:

```

clc;
clear all;
e = input('enter the no of elements:');
disp('1.Btwn a new node and reference');
disp('2.Btwn a new node and existing node');
disp('3.Btwn an existing node and reference');
disp('4.Btwn two existing nodes');
zb=complex(0,0);
zt=complex(0,0);
for i=1:e
    if(i==1)
        a=0;
    else
        [a,c]=size(zb);
    end
    b=a+1;
    disp(['elements no:',num2str(i)]);
    ty(i)=input('enter the type of element:');
    z(i)=input('enter the impedance value:');
    s(i)=input('enter the starting node:');
    t(i)=input('enter the ending node:');
    if(ty(i)==1)
        zb(b,b)=z(i);
        for j=1:a
            zb(j,b)=0;
            zb(b,j)=0;
        end
        disp(zb);
    elseif(ty(i)==2)
        zb(b,b)=z(i)+zb(t(i),t(i));
        for j=1:a
            zb(b,j)=zb(t(i),j);

```

```

zb(j,b)=zb(j,t(i));
end
disp(zb);
elseif(ty(i)==3)
zb(b,b)=z(i)+zb(s(i),s(i));
for j=1:a
zb(b,j)=zb(s(i),j);
zb(j,b)=zb(j,s(i));
end
w=zb(b,b);
for j=1:a
for k=1:a
zt(j,k)=complex(0,0);
zt(j,k)=zb(j,k)-(zb(j,b)*zb(b,k))/w;
end
end
zb=zt;
disp(zb);
elseif(ty(i)==4)
zb(b,b)=z(i)+zb(s(i),s(i))+zb(t(i),t(i))-2*zb(s(i),t(i));
for j=1:a
zb(j,b)=zb(j,s(i))-zb(j,t(i));
zb(b,j)=zb(s(i),j)-zb(t(i),j);
end
w=zb(b,b);
for j=1:a
for k=1:a
zt(j,k)=complex(0,0);
zt(j,k)=zb(j,k)-(zb(j,b)*zb(b,k))/w;
end
end
zb=zt;
else
disp('wrong data');
end
end
disp('bus impedance matrix');
disp(zb);

```

EXECUTION:



**RESULT:**

Thus, the formation of bus impedance matrix is done by using MATLAB and the output is verified for the given power system.

EX.NO:3 LOAD FLOW ANALYSIS – I: SOLUTION OF LOAD FLOW AND RELATED PROBLEMS USING GAUSS-SEIDEL METHOD

AIM:

To find load flow analysis using Gauss-Seidal method in MATLAB

SOFTWARE REQUIRED:

MATLAB software package

ALGORITHM:

STEP1: The slack bus voltage magnitude and angle are measured usually $V_1 = 1$ p.u. with the load profile known at each bus, we allocate P_i and Q_i to all generating Stations with this step, bus injections ($P_i + Q_i$) are known at all buses other than the slack bus.

STEP2: Assembly of bus admittance matrix: with the line and shunt admittance data stored in the computer. Y_{bus} is assembled by using the algorithm developed earlier. Alternatively bus is assembled using $Y_{bus} = AY$ where the input is in the form of primitive admittance matrix Y and singular connection bus incidence matrix A .

STEP3: Iterative computation of bus voltages (V_i , $i=1,2,\dots,n$) to start iteration a set of initial values is assumed, since in a power system the voltage spread is not too wide, it is normal practice to use a flat voltage start, i.e. initially all voltages are set equal to $(1+j0)$ except slack bus voltage which is fixed. this reduced the n equations in complex number which are to solved iteratively for finding complex voltages V_2, V_3, \dots, V_n . If complex no options are not available in a computer, the equation is real unknown js

$$V_i = e_i + jf_i = V_i \sum_j Z_{ij}$$

We also define,
 $A_i = P_i - jQ_i / Y_{ii} = 2, \dots, n$

$$B_{ik} = Y_{ik} \quad i=2, \dots, k \neq i$$

Now for (r+1) the iteration the voltage becomes,

$$V_i(r+1) = [A_i / (V_{ir}) * -\sum_{i=1}^{n-1} (B_{ik} V_k(r+1) - \sum_{bik} V_k r)]$$

The iterative process is continued till the change in magnitude of bus voltage $|\Delta V_i(r+1)|$

Between two consecutive is less than a certain for all bus voltages i.e.,

$$|\Delta V_i(r+1)| = |V_i(r+1) - V_i(r)| \leq \epsilon_i, i=2, \dots, n$$

Also we see if

$$|\Delta V_i|_{\min} \leq |\Delta V_i|_{\max}, i=2, \dots, n$$

If not, we fix $|\Delta V_i|$ at one of extreme values i.e.

$$|\Delta V_i|_{\min} \text{ if } |\Delta V_i| \leq |\Delta V_i|_{\min} \text{ or } |\Delta V_i|_{\max} \text{ if } |\Delta V_i| \geq |\Delta V_i|_{\max}$$

STEP4: Computation of slack bus power; substitution of all bus voltages computed in step3 with V_i and $I=1$ yield real and reactive power at slack bus i.e., $S_1 = P_1 + jQ_1$

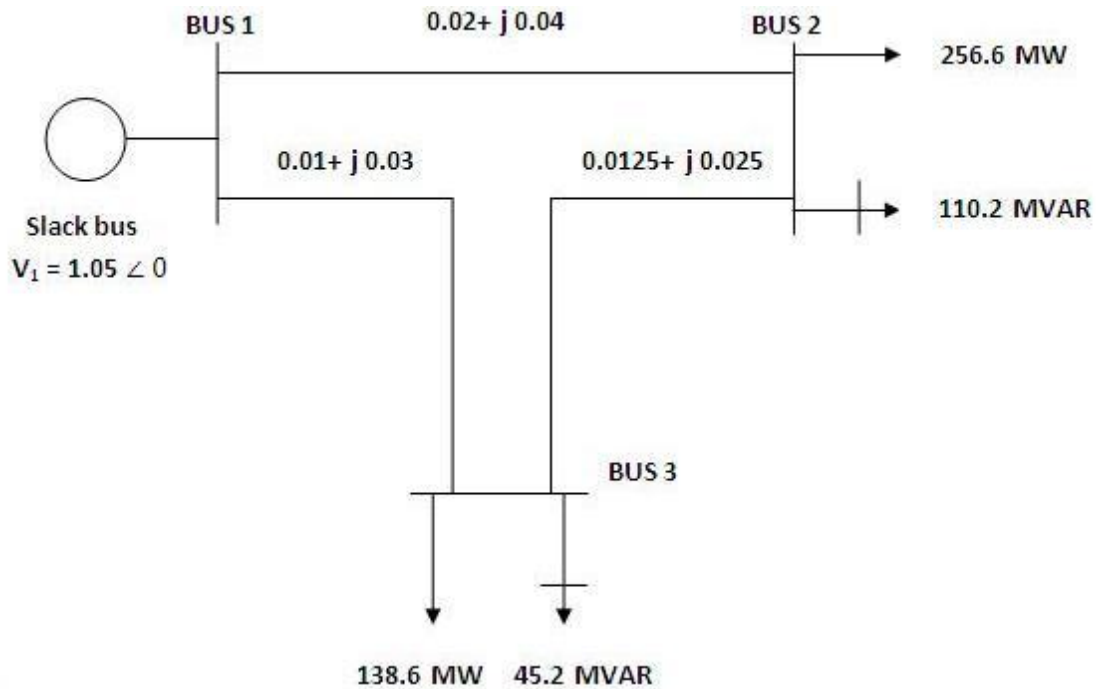
STEP5: Computation of line flows; this is the last step in the load flow analysis where in the power flows on the various lines of the network are computed. This also enables us to check whether any line overloaded. Consider the line connecting buses I and k. The line and transformer at each end can be represented by a circuit with series admittance Y_{ik} and to shunt admittances Y_{i0} and Y_{k0} . As the current fed by bus I in to the line can be expressed as

$$I_{ik} = I_{ik1} + I_{ik0} = (V_i - V_k)Y_{ik} + V_i Y_{i0}$$

PROBLEM:

Figure shows the one-line diagram of a three-bus power system with generation at bus1. The magnitude of voltage at bus1 is adjusted to **1.05** per unit. The scheduled loads at buses 2 & 3 are as marked on the diagram. Line impedances are marked in per unit on a **100MVA** base and the line charging susceptances are neglected.

- i). Using Gauss-Seidal method, determine the phasor values of the voltage at the load buses 2 & 3 (PQ buses) accurate to 4 decimal places.
- ii). Write and execute a MATLAB program and also verify the output with the manual calculation results.



PROGRAM:

```
%----- FORMATION OF YBUS -----%
clear all;
clc;
n= input('Enter the number of buses ');
fprintf('Enter your choice');
p= input ('1. impedance, 2. admittance');
if (p==1)
for q= 1:n
for r=q+1:n
fprintf('Enter the impedance value between %d-%d',q,r);
z(q,r)=input(':');
if (z(q,r)==0)
y(q,r)=0;
else
y(q,r)=inv(z(q,r));
end
y(r,q)= y(q,r);
end
end
```

```

end
end
elseif (p==2)
for a= 1:n
for b=a+1:n
fprintf('Enter the admittance value between %d-%d',a,b);
y(a,b)=input(':');
y(b,a)= y(a,b);
end
end
else
fprintf('enter the correct choice');
end
ybus=zeros(n,n);
for a = 1:n
for b=1:n
if (a==b)
for c = 1:n
ybus(a,a)= ybus(a,a)+ y(a,c);
end
else
ybus(a,b)=-y(b,a);
end
end
end
ybus
%----- BUSDATA -----%
busdata =input(' Bus number | 1.Slack 2.PQ 3.PV | V| angle|Pg | Qg | PL
| QL | Qmin | Qmax ');
bus = busdata(:,1);
type = busdata(:,2);
V = busdata(:,3);
th = busdata(:,4);
GenMW = busdata(:,5);
GenMVAR = busdata(:,6);
LoadMW = busdata(:,7);
LoadMVAR = busdata(:,8);
Qmin = busdata(:,9);
Qmax = busdata(:,10);
nbus = max(bus);
P = GenMW - LoadMW;
Q = GenMVAR - LoadMVAR;
Vprev = V;
toler = 1;
iteration = 1;
disp(' Bus number | 1.Slack 2.PQ 3.PV | V| angle|Pg | Qg | PL | QL |
Qmin | Qmax ');
busdata
ybus

```

```

%----- VOLTAGE CALCULATION -----%
while (toler> 0.00001)
for i = 2:nbus
sumyv = 0;
for k = 1:nbus
if i ~= k
sumyv = sumyv + ybus(i,k)* V(k);
end
end
if type(i) == 2
Q(i) = -imag(conj(V(i))*(sumyv + ybus(i,i)*V(i)));
if (Q(i) >Qmax(i)) || (Q(i) <Qmin(i))
if Q(i) <Qmin(i)
Q(i) = Qmin(i);
else
Q(i) = Qmax(i);
end
type(i) = 3;
end
end
V(i) = (1/ybus(i,i))*((P(i)-j*Q(i))/conj(V(i)) - sumyv);
if type(i) == 2
V(i) = pol2rect(abs(Vprev(i)), angle(V(i)));
end
end
iteration = iteration + 1;
toler = max(abs(abs(V) - abs(Vprev)));
Vprev = V;
end
iteration
V
Vmag = abs(V)
Ang = 180/pi*angle(V)
sum=0;

%----- REAL AND REACTIVE POWER CALCULATION -----%
for i=1:nbus
if i==1
for f=1:nbus
sum=sum+(ybus(i,f)*V(f));
real_power(i)=(real(V(i)*sum))*100;
reactive_power(i)=-(imag(V(i)*sum))*100;
end
else
end
end
real_power
reactive_power

```



EXECUTION:





RESULT:

Thus the load flow analysis using is performed by Gauss-Seidal method and a program is developed using MATLAB to find the solution of load flow for given power system and the output is verified.

EX.NO:4 **LOAD FLOW ANALYSIS – II: SOLUTION OF LOAD FLOW USING NEWTON – RAPHSON METHOD**

AIM:

To develop in MATLAB to find the solution of power flows using Newton Raphson method.

SOFTWARE REQUIRED:

MATLAB software package

ALGORITHM:

- STEP 1: Assume a flat profile $1+j0$ for all buses except the slack bus in 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 actual residue that is used to compare the actual residue at the end of each iteration.
- STEP 3: Set the iteration count $K=0$ and assumed voltage profile of the buses are denoted as $V_1^0, V_2^0, \dots, V_n^0$.
- STEP 4: Set the 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 & reactive power of bus p using the following equation,
- $$P_{pk} = \sum_{q=1}^n \{ e_p^k (e_q^k G_{pq} + f_q^k B_{pq}) + f_p^k (f_q^k G_{pq} - e_q^k B_{pq}) \}$$
- $$Q_{pk} = \sum_{q=1}^n \{ f_p^k (e_q^k + G_p^k + f_q^k B_{pq}) + e_p^k (f_q^k G_{pq} - e_q^k B_{pq}) \}$$
- STEP 7: Calculate the change in real power,
- $$\Delta P^k = P_{p \text{ spec}} - P_p^k$$
- Where,
- $P_{p \text{ spec}}$ = specified real power of bus p.
- STEP 8: Check for generator bus. If it is a generator bus gob to next step otherwise go to step 12.
- STEP 9: Check for generator bus. If it is a generator reactive power limit Violation of generator buses. For this compare the calculated

reactive power Q with specified limits. If the limits are violated go to step 11. otherwise go to next step.

STEP 10: If the calculated reactive power is within the specified limit then consider this bus as generator bus. Now calculate the voltage residue using the equation
$$|V_p^k|^2 = |V_{p\text{ spec}}|^2 - |V_p^k|^2$$

where $|V_{p\text{ spec}}|$ = specified voltage magnitude for generation bus.
Then go to step 13.

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

STEP 12: Calculate the change in reactive power for load bus change in reactive power,
$$\Delta Q_k = |\Delta Q_{p\text{ spec}}| - Q_p^k$$

STEP 13: Repeat the step 5 to 12 until all residues are calculated for increment the bus count n by 1 to 5 steps until the bus count is n .

STEP 14: Determine the largest of the absolute value of the residue (i.e.) Find the largest value among ΔP_p^k , ΔQ_p^k or $|\Delta V_p^k|^2$

STEP 15: Compare ΔE and E , if $\Delta E < E$ then go to step 20. If $\Delta E > E$ go to next Step.

STEP 16: Determine the element, the load flow equation using k^{th} iteration Value.

STEP 17: Calculate the increment in real and reactive part of voltage Δe_p^k and Δf_p^k by solving the matrix equation $B=JC$.

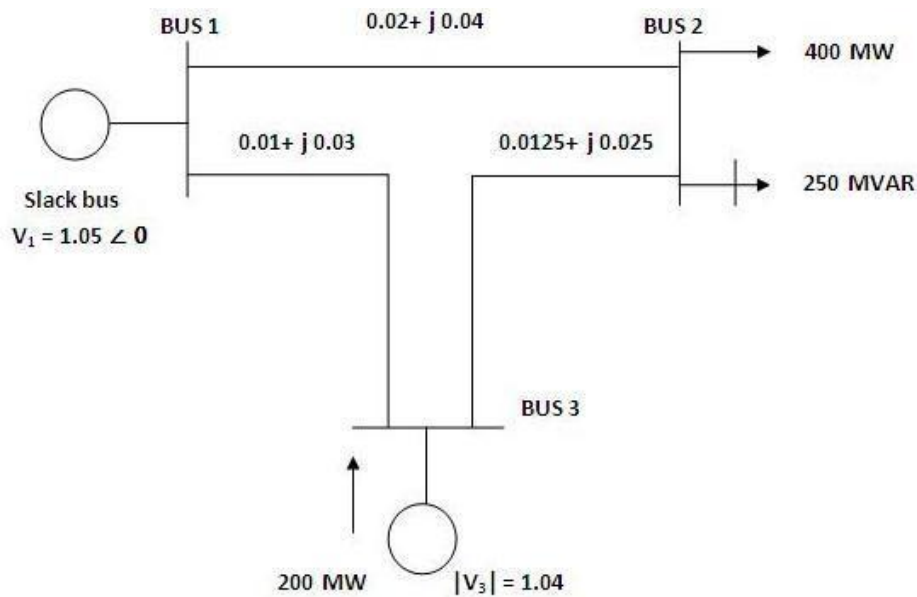
STEP 18: Calculate the new bus voltage.

STEP 19: Advance the iteration count $k=k+1$ and go to step 4.

STEP 20: Calculate the line flows.

PROBLEM:

Figure shows the one-line diagram of a three-bus power system with generators at buses 1 & 3. The magnitude of voltage at bus1 is adjusted to **1.05** per unit. The magnitude of voltage at bus 3 is fixed at **1.04pu** with a real power generation of **200 MW**. A load consists of **400 MW** and **250 MVAR** is taken from bus2. Line impedances are marked in per unit on a **100MVA** base and the line charging susceptances are neglected.



- Obtain the load flow solution by Newton-Raphson method.
- Write and execute a MATLAB program and also verify the output with the manual calculation results.

PROGRAM:

```
clear all;
clc;
v=[1.05;1.0;1.04];
d=[0;0;0];
ps=[-4;2.0];
qs=-2.5;
n= input('Enter the number of buses ');
fprintf('Enter your choice');
p= input ('1. impedance, 2. admittance');
if (p==1)
for q= 1:n
```

```

for r=q+1:n
fprintf('Enter the impedance value between %d-%d',q,r);
z(q,r)=input(':');
if (z(q,r)==0)
y(q,r)=0;
else
y(q,r)=inv(z(q,r));
end
y(r,q)= y(q,r);
end
end
elseif (p==2)
for a= 1:n
for b=a+1:n
fprintf('Enter the admittance value between %d-%d',a,b);
y(a,b)=input(':');
y(b,a)= y(a,b);
end
end
else
fprintf('enter the correct choice');
end
ybus=zeros(n,n);
for a = 1:n
for b=1:n
if (a==b)
for c = 1:n
ybus(a,a)= ybus(a,a)+ y(a,c);
end
else
ybus(a,b)=-y(b,a);
end
end
end
ybus
y=abs(ybus); t=angle(ybus);
iter=0;
pwracur=0.00025;% Power accuracy
dc=10;% Set the maximum power residual to a high value
while max(abs(dc))>pwracur
iter=iter+1
p=[v(2)*v(1)*y(2,1)*cos(t(2,1)-
d(2)+d(1))+v(2)^2*y(2,2)*cos(t(2,2))+v(2)*v(3)*y(2,3)*cos(t(2,3)-
d(2)+d(3));
v(3)*v(1)*y(3,1)*cos(t(3,1)-
d(3)+d(1))+v(3)^2*y(3,3)*cos(t(3,3))+v(3)*v(2)*y(3,2)*cos(t(3,2)-
d(3)+d(2))];
q=-v(2)*v(1)*y(2,1)*sin(t(2,1)-d(2)+d(1))-v(2)^2*y(2,2)*sin(t(2,2))-
v(2)*v(3)*y(2,3)*sin(t(2,3)-d(2)+d(3));

```

```

j(1,1)=v(2)*v(1)*y(2,1)*sin(t(2,1)-
d(2)+d(1))+v(2)*v(3)*y(2,3)*sin(t(2,3)-d(2)+d(3));
j(1,2)=-v(2)*v(3)*y(2,3)*sin(t(2,3)-d(2)+d(3));
j(1,3)=v(1)*y(2,1)*cos(t(2,1)-
d(2)+d(1))+2*v(2)*y(2,2)*cos(t(2,2))+v(3)*y(2,3)*cos(t(2,3)-d(2)+d(3));
j(2,1)=-v(3)*v(2)*y(3,2)*sin(t(3,2)-d(3)+d(2));
j(2,2)=v(3)*v(1)*y(3,1)*sin(t(3,2)-
d(3)+d(1))+v(3)*v(2)*y(3,2)*sin(t(3,2)-d(3)+d(2));
j(2,3)=v(3)*y(2,3)*cos(t(3,2)-d(3)+d(2));
j(3,1)=v(2)*v(1)*y(2,1)*cos(t(2,1)-
d(2)+d(1))+v(2)*v(3)*y(2,3)*cos(t(2,3)-d(2)+d(3));
j(3,2)=-v(2)*v(3)*y(2,3)*cos(t(3,2)-d(2)+d(3));
j(3,3)=-v(1)*y(2,1)*sin(t(2,1)-d(2)+d(1))-2*v(2)*y(2,2)*sin(t(2,2))-
v(3)*y(2,3)*sin(t(2,3)-d(2)+d(3));
dp=ps-p;
dq=qs-q;
dc=[dp;dq]
j
dx=j\dc
d(2)=d(2)+dx(1);
d(3)=d(3)+dx(2);
v(2)=v(2)+dx(3);
v,d,delta=180/pi*d;
end
p1=v(1)^2*y(1,1)*cos(t(1,1))+v(1)*v(2)*y(1,2)*cos(t(1,2)-
d(1)+d(2))+v(1)*v(3)*y(1,3)*cos(t(1,3)-d(1)+d(3));
q1=-v(1)^2*y(1,1)*sin(t(1,1))-v(1)*v(2)*y(1,2)*sin(t(1,2)-d(1)+d(2))-
v(1)*v(3)*y(1,3)*sin(t(1,3)-d(1)+d(3));
q3=-v(3)*v(1)*y(3,1)*sin(t(3,1)-d(3)+d(1))-v(3)*v(2)*y(3,2)*sin(t(3,2)-
d(3)+d(2))-v(3)^2*y(3,3)*sin(t(3,3));

```

EXECUTION:







RESULT:

Thus the load flow analysis using is performed by Newton-Raphson method and a program is developed using MATLAB to find the solution of load flow for given power system and the output is verified.

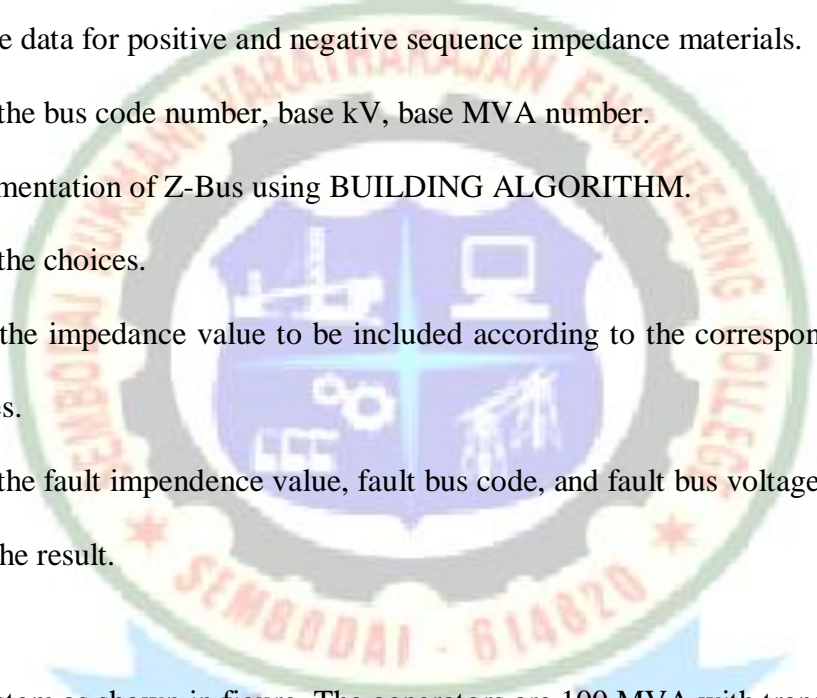
AIM:

To develop a MATLAB program to perform the fault analysis for a given power system.

SOFTWARE REQUIRED:

MATLAB software package

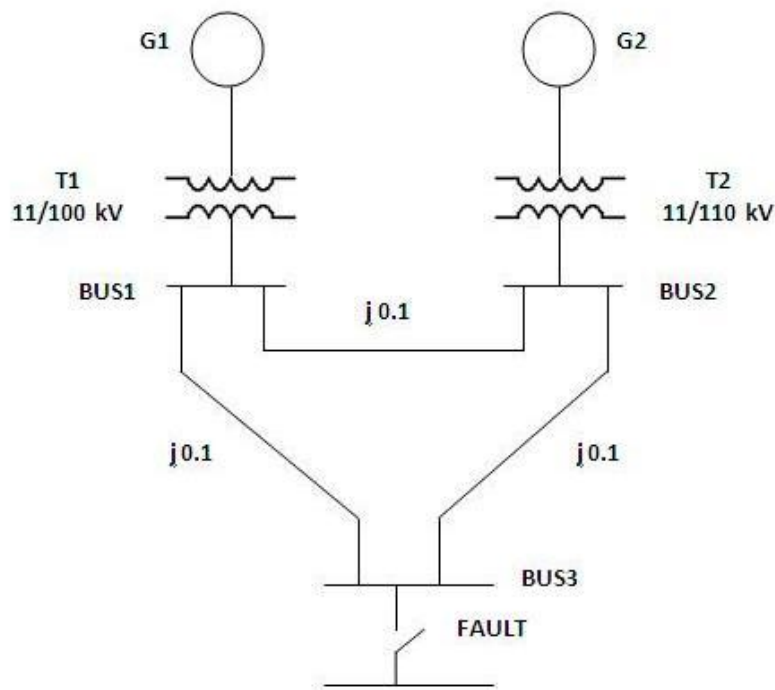
ALGORITHM:

- 
- STEP 1: Get the data for positive and negative sequence impedance materials.
- STEP 2: Enter the bus code number, base kV, base MVA number.
- STEP 3: Implementation of Z-Bus using BUILDING ALGORITHM.
- STEP 4: Enter the choices.
- STEP 5: Enter the impedance value to be included according to the corresponding choices.
- STEP 6: Enter the fault impedance value, fault bus code, and fault bus voltage.
- STEP 7: Print the result.

PROBLEM:

Consider the 3bus system as shown in figure. The generators are 100 MVA with transient reactance 10% each. Both the transformers are 100 MVA with a leakage reactance of 5%. The reactance of each of the lines to a base of 100 MVA, 110kV is 10%. Obtain the short circuit calculation for a 3phase solid short circuit on bus 3.

Assume pre-fault voltages to be 1pu and pre-fault currents to be zero.



Write and execute a MATLAB program and also verify the output with the manual calculation results.

PROGRAM:

```
clear all;
close all;
clc;
kv=input('enter the base kv');
mva=input('enter the base mva');
bc=[(mva*1000)/(sqrt(3)*kv)];
e = input('enter the no of elements:');
disp('1.Btwn a new node and reference');
disp('2.Btwn a new node and existing node');
disp('3.Btwn an existing node and reference');
disp('4.Btwn two existing nodes');
zb=complex(0,0);
zt=complex(0,0);
for i=1:e
if(i==1)
a=0;
else
[a,c]=size(zb);
end
b=a+1;
```

```

disp(['elements no:',num2str(i)]);
ty(i)=input('enter the type of element:');
z(i)=input('enter the impedance value:');
s(i)=input('enter the starting node:');
t(i)=input('enter the ending node:');
if(ty(i)==1)
zb(b,b)=z(i);
for j=1:a
zb(j,b)=0;
zb(b,j)=0;
end
disp(zb);
elseif(ty(i)==2)
zb(b,b)=z(i)+zb(t(i),t(i));
for j=1:a
zb(b,j)=zb(t(i),j);
zb(j,b)=zb(j,t(i));
end
disp(zb);
elseif(ty(i)==3)
zb(b,b)=z(i)+zb(s(i),s(i));
for j=1:a
zb(b,j)=zb(s(i),j);
zb(j,b)=zb(j,s(i));
end
w=zb(b,b);
for j=1:a
for k=1:a
zt(j,k)=complex(0,0);
zt(j,k)=zb(j,k)-(zb(j,b)*zb(b,k))/w;
end
end
zb=zt;
disp(zb);
elseif(ty(i)==4)
zb(b,b)=z(i)+zb(s(i),s(i))+zb(t(i),t(i))-2*zb(s(i),t(i));
for j=1:a
zb(j,b)=zb(j,s(i))-zb(j,t(i));
zb(b,j)=zb(s(i),j)-zb(t(i),j);
end
w=zb(b,b);
for j=1:a
for k=1:a
zt(j,k)=complex(0,0);
zt(j,k)=zb(j,k)-(zb(j,b)*zb(b,k))/w;
end
end
zb=zt;
else

```



```

disp('wrong data');
end
end
disp('bus impedance matrix');
disp(zb);
bn=input('enter fault bus number');
zf=input('enter fault impedance');
vf=complex(1,0);
fc=vf/(zb(bn,bn)+zf);
fca=fc*bc;
disp(['fault current in pu=',num2str(fc),'pu']);
disp('post fault voltages:');
for i=1:a
vpf(i)=[vf-(zb(i,bn)*fc)];
disp(['post fault voltage',num2str(i),':',num2str(vpf(i))]);
end
for i=1:e
if(s(i)~=0&&t(i)~=0)
pfc(i)=(vpf(s(i))-vpf(t(i)))/z(i);
disp(['current flow I',num2str(s(i)),'and I',num2str(t(i))]);
disp(pfc(i));
end
end

```

EXECUTION:







RESULT:

Thus a program is developed using MATLAB to perform the fault analysis for a given power system network and the output is verified.

**EX.NO:6 TRANSIENT STABILITY ANALYSIS OF A SINGLE MACHINE
INFINITE BUS SYSTEM**

AIM:

To analyze the transient stability of a single machine infinite bus system by point by point method using MATLAB.

SOFTWARE REQUIRED:

MATLAB software package

ALGORITHM:

STEP 1: Evaluate the accelerating power P_a .

STEP 2: From the swing equation

$$d^2\delta/dt^2 = \omega(0) = P_a(Q_t)/M$$

Where δ is the acceleration factor. Evaluation δ .

STEP 3: The change in angular velocity for the first interval is calculated.

STEP 4: The change in rotor angle for the first interval is also calculated.

STEP 5: If the discontinuity occurs due to removal of the fault or due to switching operation there are three possibilities.

- a. The discontinuity occurs at the beginning of i^{th} interval.
- b. The discontinuity occurs at the middle of the i^{th} interval.

STEP 6: To evaluate P_a when under first situation, one should use the value

corresponding to average value of Q-accelerating power i.e., power before and after clearing the fault.

STEP 7: To evaluate P_a under the situation, a weighted average value of P_a before and after the discontinuity may be used.

STEP 8: Thus equating for n^{th} interval can be written as

$$P_a(n-1) = P_s - P_e(n-1)$$

$$\delta_n = \delta_{n-1} + \Delta\delta_n$$

These are used for plotting the curve.

STEP 9: To evaluate P_a under second situation no procedure is required. It is taken as the value at the beginning of the interval.

PROBLEM:

Consider a system which consists of generator having a rating of **50 MVA** & **H=2.7 MJ/MVA** at rated speed. **E=1.05**, **V=1**, **X_{d'}=0.2**, **X₁=X₂=0.4 pu**. The generator supplies **50 MW** to the infinite bus when a 3phase fault occurs at middle of line 2.

- Plot swing curve for a sustained fault up to **0.5 sec**.
- Plot the swing curve if the fault is cleared in **0.1 sec** by simultaneous opening of breakers at both ends of line 2.
- Find the critical clearing angle & clearing time.
- Write and execute a MATLAB program and also verify the output with the manual calculation results.

PROGRAM:

```
clear all;
clc;
t=0;
tf=0;
f=input('Enter the input frequency');
s= input('Enter the machine rating');
ang(1)=input('Enter the initial angle');
h=input('Enter the moment of inertia constant');
tfinal=input('Enter the end time');
tstep=input('Enter the change in time');
tc=input('enter the clearing time');
pm=input('enter the power transfer');
pmaxbf=input('enter the prefault power');
pmaxdf=input('enter the power during fault');
pmaxaf=input('enter the postfault power');
m= (s*h)/(180*f);
delta=ang(1)*pi/180;
i=2;
```

```

ddelta=0;
time(1)=0;
while t<tfinal
if t==tf
paminus=pm-pmaxbf*sin(delta);
paplus=pm-pmaxdf*sin(delta);
paav=(paminus+paplus)/2;
pa=paav;
end
if(t==tc)
paminus=pm-pmaxdf*sin(delta);
paplus=pm-pmaxaf*sin(delta);
paav=(paminus+paplus)/2;
pa=paav;
end
if(t>tf&t<tc)
pa=pm-pmaxdf*sin(delta);
end
if(t>tc)
pa=pm-pmaxaf*sin(delta);
end
t,pa
ddelta=ddelta+(tstep*tstep*pa/m);
delta=((delta*180/pi)+ddelta)*(pi/180);
deltadeg=delta*180/pi;
t=t+tstep;
pause
time(i)=t;
ang(i)=deltadeg;
i=i+1;
end
axis([0 0.6 0 180])
plot(time,ang,'ko-')

```

EXECUTION:









RESULT:

Thus the transient stability analysis for single machine connected to infinite bus and fault clearance in different time was studied.

**EX.NO:7 SMALL STABILITY ANALYSIS OF A SINGLE MACHINE
INFINITE BUS SYSTEM**

AIM:

To analyze the small signal stability of the given power system with the help of MATLAB program.

SOFTWARE REQUIRED:

MATLAB software package

ALGORITHM:

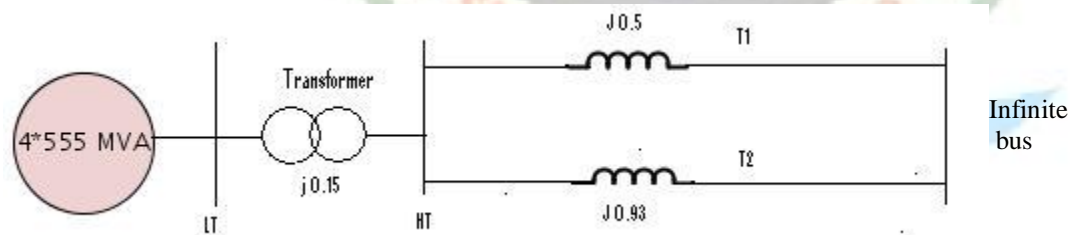
STEP: 1 Click the PSCAD icon.

STEP: 2 Open the untiled window and create the model for of transient stability analysis of multimachine power system from the various library functions available in the PSCAD browser.

STEP: 3 Save the file and simulate the model.

STEP: 4 Double click the scope and we can get the waveform.

PROBLEM:



The shown fig shows the system representation applicable to a thermal generating station consisting of four 555 MVA, 24kV, 60 Hz units. The network reactances are in per unit on 220 MVA, 24kV base (referred to the LT side of the step up transformer). Resistances are assumed to be negligible. The objective of this problem analyzes the small signal stability characteristics of the system about the steady state operating condition following the loss of ckt1. The post fault systems condition is per unit on the 220MVA, 24kV base as follows; $P=0.9$, $Q=0.3$ (over excited), $E_t=1.0 \angle 36^\circ$, $E_B=0.995 \angle 0^\circ$.

The generators are to be modeled as a single equivalent generator represented by the classical model by the following parameters expressed in pu on 2220MVA, 24 kV, base $X_2' = 0.2$, $H = 3.5$.

- a). Write the linearized state equations of the system. Determine the eigen values (λ_1, λ_2) damped frequency of oscillation in HZ, damping ratio and undamped natural frequency for each of the following parameters expresses values of damping coefficients. i). $K_D = 0$, . ii). $K_D = 10$. iii). $K_D = -10$.

PROGRAM:

```
clc;
clear all;
p=0.9;q=0.3;f=60;xd=0.3*i;xtr=0.15*i;x1=0.5*i;x2=0.93*i;Et=1;Eb=0.995;
w0=2*pi*f;
s=p+q*i;
It=s'/Et';
FL=input('ENTER THE FAULTED LINE NO:');
if FL==1
xt=xd+xtr+x2;
else
xt=xd+xtr+x1;
end
E=Et+It*xd;
D=angle(E)-angle(Eb);
ks=(abs(E)*abs(Eb)*cos(D))/abs(xt);
H=input('ENTER THE VALUE OF INERTIA CONSTANT , H:');
kd=input('ENTER THE Kd VALUE :');
A=[(-kd/(2*H)) (-ks/(2*H));w0 0];
lambda=eig(A)
w=ks*w0/(2*H);
wn=sqrt(w)/(2*pi)
z=0.5*kd/sqrt(2*ks*H*w0)
wd=wn*sqrt(1-z*z)
[VD]=eig(A);
g=[(-0.0019+0.0168*i) (-0.0019-0.0168*i); 1 1];
L=inv(g);
[P]=[L(1,1)*g(1,1) L(1,2)*g(2,1);L(1,2)*g(2,1) L(2,2)*g(2,2)]
theta=acos(z)
D0=5*pi/180;
t=0:.01:3;
Dd=D0/sqrt(1-z^2)*exp(-z*wn*t).*sin(wd*t+theta);
d=(D+Dd)*180/pi;
plot(t,d),grid
xlabel('t sec'),ylabel('delta degree')
```

EXECUTION:





RESULT:

Thus the small signal stability of the given power system is analyzed & the output is verified using MATLAB program.

EX: NO: 8

Electromagnetic Transients in Power Systems

AIM:

To plot the electromagnetic transients in power systems using MATLAB.

SOFTWARE REQUIRED:

MATLAB software package

FORMULAE USED:

Case (i): Line terminated by inductance

$$i_t(s) = \frac{2E_f / L}{s(s + z_c / L)}$$

$$e_t(s) = \frac{2E_f}{(s + z_c / L)}$$

$$i_t(s) = \frac{2E(1 - e^{-Z_c t / L})}{z_c}$$

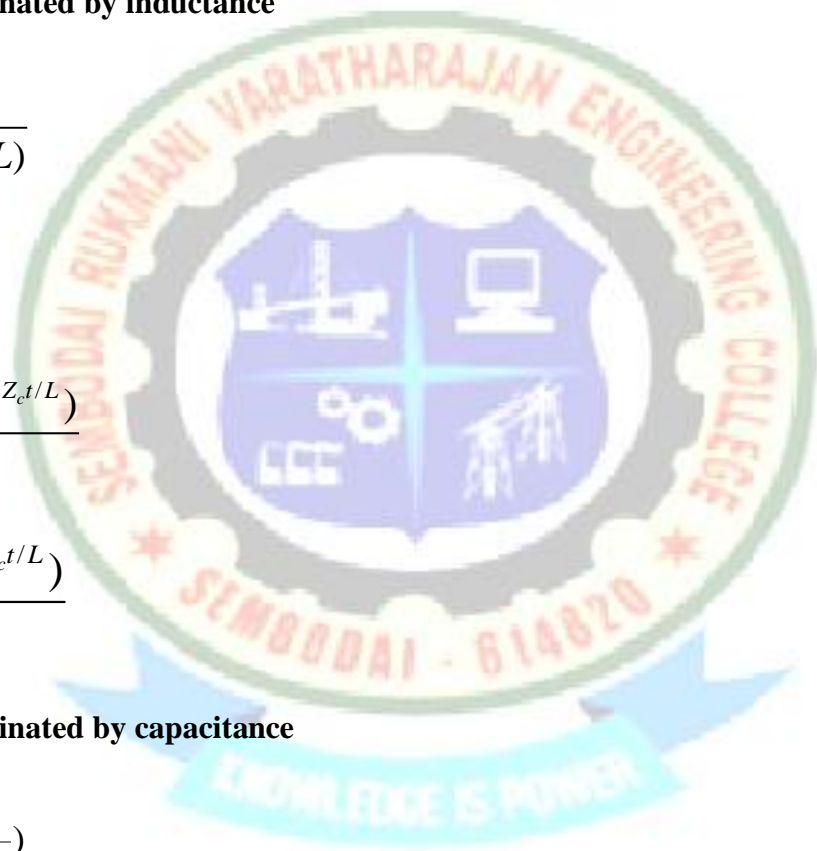
$$i_r = \frac{E(1 - 2e^{-Z_c t / L})}{z_c}$$

Case (ii): Line terminated by capacitance

$$i_t(s) = \frac{2E(\frac{1}{s} + \frac{1}{Z_c c})}{Z_c}$$

Case (iii): Line terminated by a resistance equal to surge impedance

$$i_t(s) = \frac{2E}{Z_c s}$$



Case (iv):

Open circuited line

$$e_t = 2e_f$$

$$e_r = e_f$$

$$i_r = -i_f$$

Case (iv):

Short circuited line

$$i_t = 2i_f$$

$$e_r = -e_f$$

$$i_r = i_f$$



PROGRAM:

LINE TERMINATED BY INDUCTOR FOR VOLTAGE:

```
Ef=10000;  
L=0.004;  
Zc=400;  
n1=[2*Ef 0];  
d1=[1 Zc/L];  
t=0:0.00001:0.0001;  
Et=step(n1,d1,t);  
plot(t,Et,'r');  
Er=Et-Ef;  
hold on;  
plot(t,Er,'b')
```

LINE TERMINATED BY CAPACITOR FOR VOLTAGE:

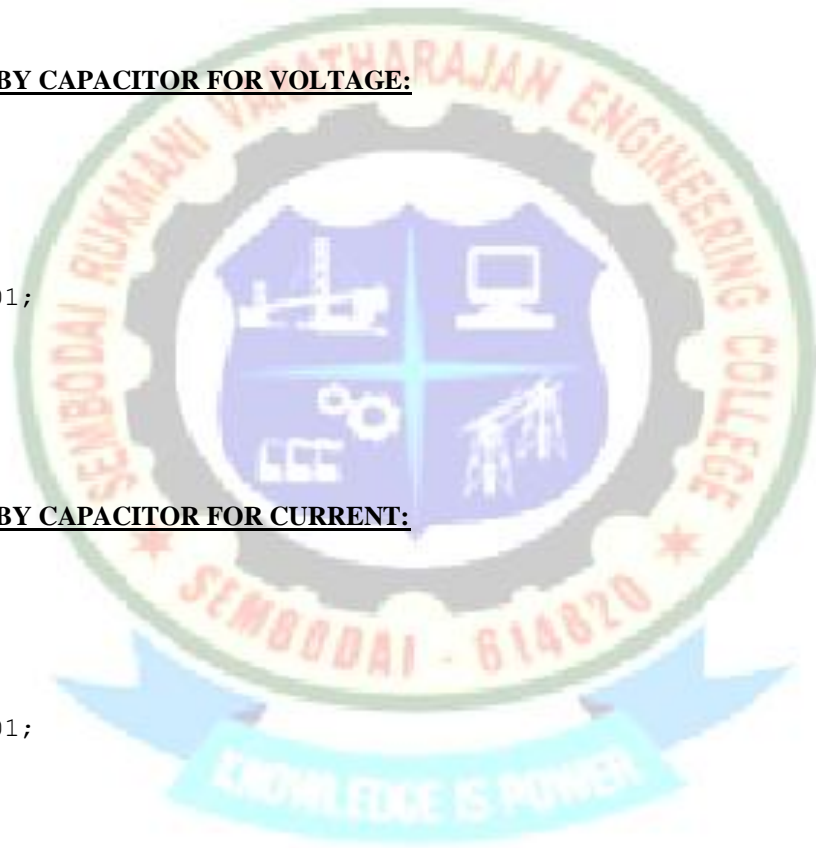
```
Ef=10000;  
C=0.000000009;  
Zc=400;  
n1=[2*Ef/(Zc*C)];  
d1=[1 1/(Zc*C)];  
t=0:0.00001:0.0001;  
Et=step(n1,d1,t);  
plot(t,Et,'r');  
Er=Et-Ef;  
hold on;  
plot(t,Er,'b')
```

LINE TERMINATED BY CAPACITOR FOR CURRENT:

```
Ef=10000;  
C=0.000000009;  
Zc=400;  
n1=[2*Ef/Zc 0];  
d1=[1 1/(Zc*C)];  
t=0:0.00001:0.0001;  
It=step(n1,d1,t);  
plot(t,It,'r');  
hold on;  
If=Ef/Zc;  
Ir=It-If;  
plot(t,Ir,'b')
```

LINE TERMINATED BY INDUCTOR FOR CURRENT:

```
Ef=10000;  
L=0.004;  
Zc=400;  
n1=[2*Ef 0];  
d1=[1 Zc/L];  
tf(n1,d1)  
t=0:0.00001:0.0001;
```




```
n2=[2*Ef/L];  
d2=[1 Zc/L];  
It=step(n2,d2,t);  
plot(t,It,'r');  
If=Ef/Zc;  
Ir=It-If;  
hold on;  
plot(t,Ir,'b');
```

EXECUTION





RESULT:

Thus MATLAB program for electromagnetic transients is executed and its performance is analyzed.

EX: NO: 9

**LOAD FREQUENCY DYNAMICS OF SINGLE AND TWO
AREA SYSTEMS**

AIM:

To become familiar with modeling and analysis of the frequency and tie-line flow dynamics of a power system without and with load frequency controllers (LFC) and to design better controllers for getting better responses using MATLAB simulink.

SOFTWARE REQUIRED:

MATLAB software package

ALGORITHM:

STEP: 1 Click the simulink icon.

STEP: 2 Open the untitled window and create the model for load frequency control of single area system using various library functions available in the simulink browser.

STEP: 3 Save the file and simulate the model.

STEP: 4 Double click the scope and we can get the waveform as change in frequency with respect to time.

STEP: 5 Simulate the model for uncontrolled case and observe the waveform.

STEP: 6 Simulate the model for controlled case (i.e.), introduce one integral control in the model and observe the waveform.

STEP: 7 Finally compare the results obtained from simulation and comment the result.

PROBLEM:

For Single area system:

An isolated power station has the following parameters

Turbine time constant = **0.5** sec

Governor time constant = **0.2** sec

Generator inertia constant = **5** sec

Governor speed regulation = **R** per unit

The load varies by **0.8** percent for a **1** percent change in frequency i.e., **D=0.8**. The governor speed regulation is set to **R=0.05 pu**. The turbine rated output is **250 MW** at nominal frequency of **60Hz**. A sudden load change of **50 MW ($\Delta P_D=0.2$ pu)** occurs.

i). Find the steady state frequency deviation in Hz.

ii). Draw a MATLAB simulink model and also verify the output with the manual calculation results.

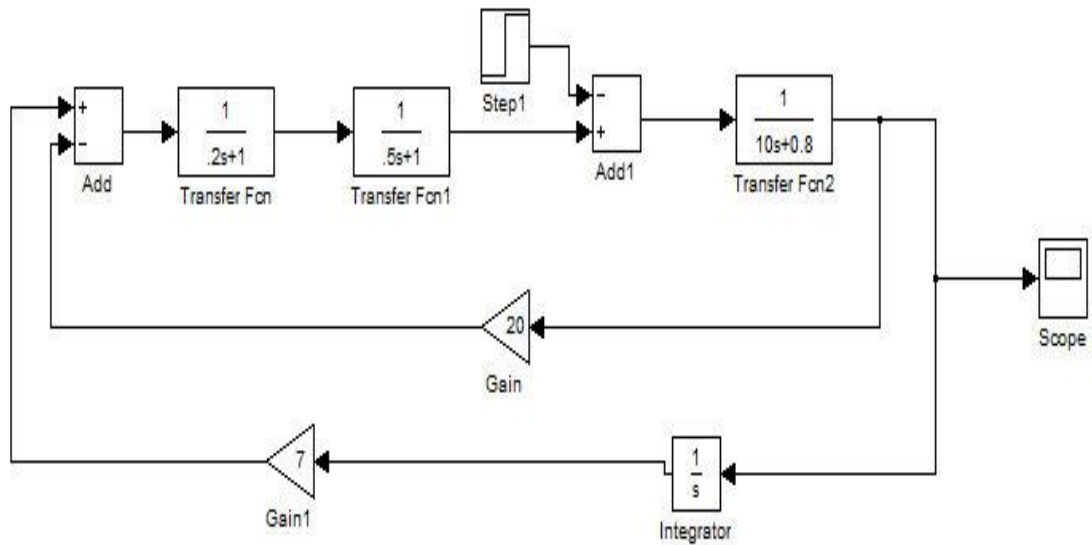
For Two area system:

A two area system connected by a tie line has the following parameters on a 1000 MVA common base

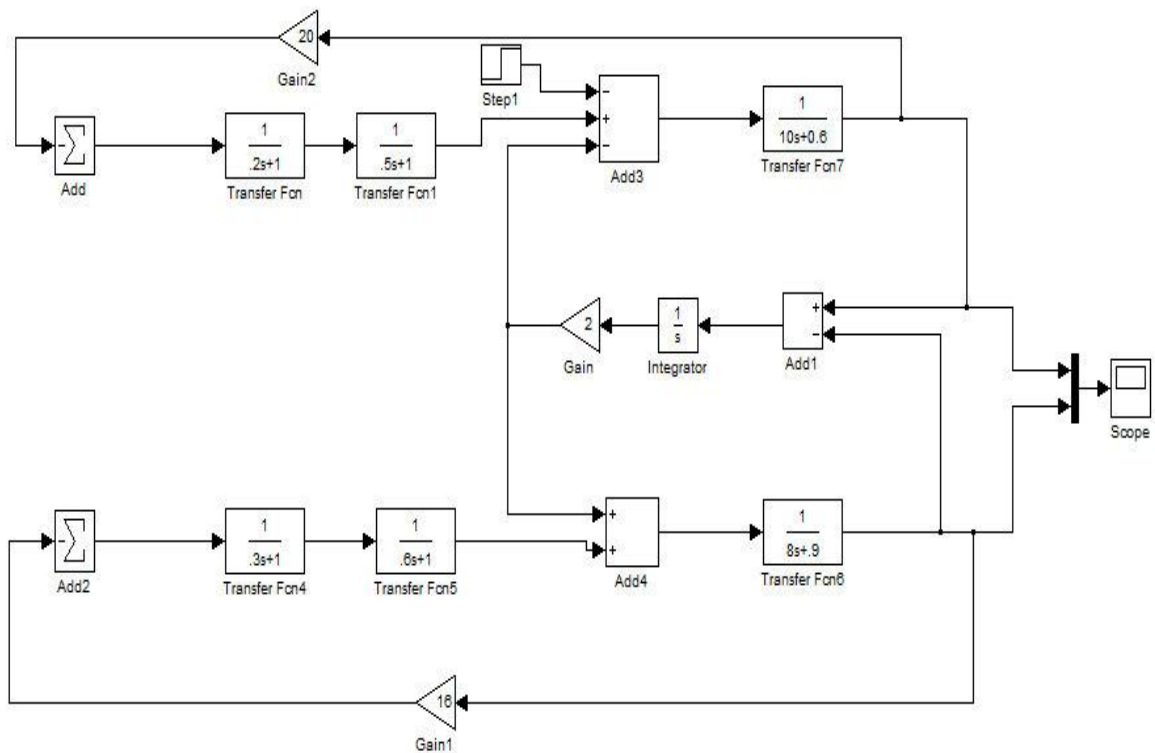
Area	1	2
Speed Regulation	$R_1=0.05$	$R_2=0.0625$
Frequency sensitive load coefficient	$D_1=0.6$	$D_2=0.9$
Inertia Constant	$H_1=5$	$H_2=4$
Base power	1000 MVA	1000 MVA
Governor time constant	$\tau_{g1}=0.2$ sec	$\tau_{g2}=0.3$ sec
Turbine time constant	$\tau_{T1}=0.5$ sec	$\tau_{T2}=0.6$ sec

The units are operating in parallel at the nominal frequency of 50Hz. The synchronizing power coefficient is computed from the initial operating condition and is given to be $P_s=2$ pu. A load change of 187.5MW occurs in area1. Determine the new steady state frequency and the change in tie-line flow.

SIMULINK Block Diagram for single area system:



SIMULINK Block Diagram for two area system:



EXECUTION:





RESULT:

Thus, the load frequency dynamics of single area and two area system are analyzed using MATLAB simulink and the outputs are verified.

AIM:

To obtain the solution for Economic dispatch by Lambda-iteration method with the help of MATLAB program.

SOFTWARE REQUIRED:

MATLAB software package

ALGORITHM:

STEP 1: Assign initial estimated value for λ .

STEP 2. Compute P_{gi} corresponding to (IC) $I = \lambda$ from the following equation

$$P_{gi} = a_i + b_i(I_c)_i + c_i(I_c)_i^2$$

STEP 3: Compute $\sum P_{gi}$: $n = \text{no. of units}$

STEP 4: Check the relationship

$$N$$

$$\sum P_{gi} = P_d$$

$$I = 1$$

(a) If the simulation is less than demand then assign a new value

$$\lambda_{\text{new}} = \lambda_{\text{old}} + \Delta \lambda$$

(b) If the simulation is greater than demand then assign a new value

$$\lambda_{\text{new}} = \lambda_{\text{old}} - \Delta \lambda$$

STEP 5: Print the corresponding units generation for $P_g = P_d$.

PROBLEM:

Find the optimal dispatch total cost for the plants.

$$C_1 = 500 + 5.3P_1 + 0.004P_1^2$$

$$C_2 = 400 + 5.5P_2 + 0.006P_2^2$$

$$C_3 = 200 + 5.8P_3 + 0.009P_3^2$$

When the total load is 975 MW with the following generator limits

$$200 \leq P_1 \leq 450;$$

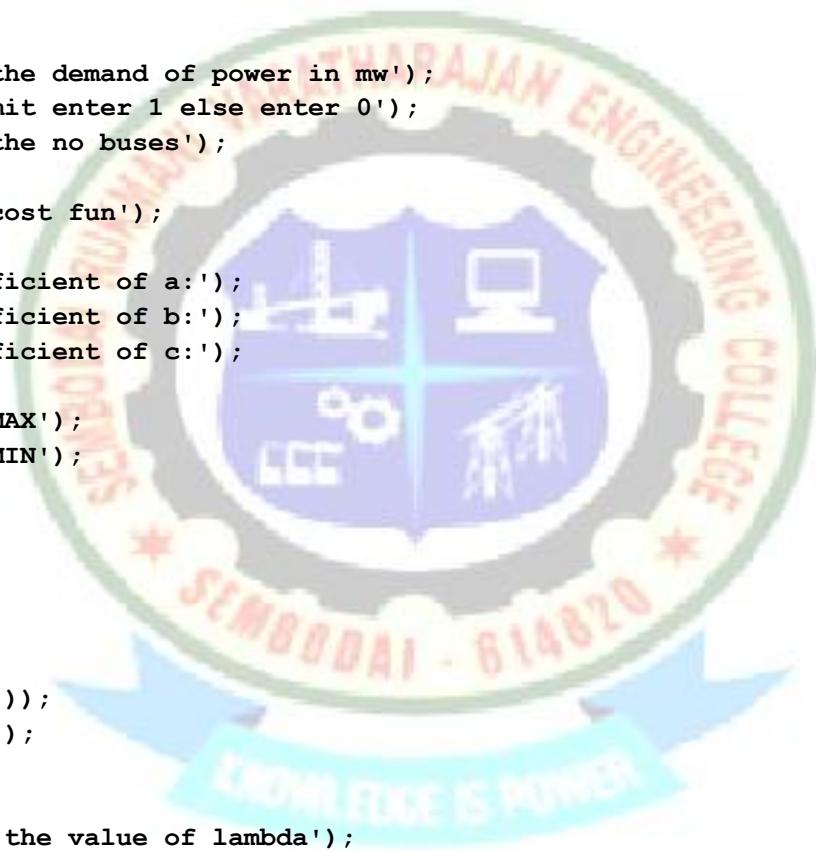
$$150 \leq P_2 \leq 350;$$

$$100 \leq P_3 \leq 225;$$

Write and execute a MATLAB program and also verify the output with the manual calculation results.

PROGRAM:

```
clc;
clear all;
pd=input('enter the demand of power in mw');
O=input('with limit enter 1 else enter 0');
n=input(' enter the no buses');
for i=1:n
disp('enter the cost fun');
disp(i);
a(i)=input('coefficient of a:');
b(i)=input('coefficient of b:');
c(i)=input('coefficient of c:');
if(O==1)
pmax(i)=input('PMAX');
pmin(i)=input('PMIN');
else
end
end
s=0;
s1=0;
for i=1:n
s=s+(b(i)/(2*c(i)));
s1=s1+(1/(2*c(i)));
end
if(O==1)
lam=input('enter the value of lambda');
else
lam=(pd+s)/(s1);
end
for k=1:50
for i=1:n
pg(i)=(lam-b(i))/(2*c(i));
if(O==1)
if(pg(i)>pmax(i))
pg(i)=pmax(i);
elseif(pg(i)<pmin(i))
pg(i)=pmin(i);
```



```

else
end
end
end
ptot=0;
for i=1:n
ptot=ptot+pg(i);
end
del_p=pd-ptot;
if(del_p==0)
break;
else
del_lambda=(del_p)/s1;
lam=lam+del_lambda;
end
%disp('pg in mwand lam');
totalcost=0;
%disp('the total cost is');
for i=1:n
F(i)=a(i)+(b(i)*pg(i))+(c(i)*pg(i)*pg(i));
F(i);
totalcost=totalcost+F(i);
end
end
pg
lam
totalcost

```

EXECUTION:







RESULT:

Thus a solution for economic load dispatch in power system is developed by using MATLAB Program.