Experiment No. 06

6.1 Experiment Name

Generate an algorithm and write a program on load flow study of a given power system using Newton Raphson method

6.2 Objectives

- To become acquainted with the load flow study of a given power system
- To learn how to generate a MATLAB code for numerical analysis using Newton Raphson method
- To get familiar with the procedure of designing and analyzing a power system in MATLAB

6.3 Theory

Power system is a very large interconnected electrical network. So different techniques have been developed to analyze power systems. Node voltage method is one of those techniques.

The equations in the nodal admittance form result in a simultaneous complex algebraic equation in terms of load currents. Solving these equations gives the voltages and currents of the buses.

The biggest challenge in solving these equations is finding $\boldsymbol{n}$ unknown quantities from $\boldsymbol{n}$ nonlinear equations. Iterative methods are suitable for solving nonlinear systems of equations.

For this reason, iterative methods are suitable for solving node voltage equations. Net injected bus current has the relation. For a power system with $\boldsymbol{n}$ nodes the network equation where the current injected into the $\boldsymbol{i^{th}}$ node can be obtained as $\boldsymbol{I_i = \sum_{k=1}^{n} Y_{ik} V_{ik}}$

The power injected into the $\boldsymbol{i^{th}}$ node can be written as

$$\boldsymbol{S_i = P_i + jQ_i = \sum_{K=1}^{n} |V_i|\,|V_k||V_{ik}|\angle\delta_i - \theta_{ik} - \delta_k}$$

6.4 Required apparatus

- MATLAB

6.5 Block diagram

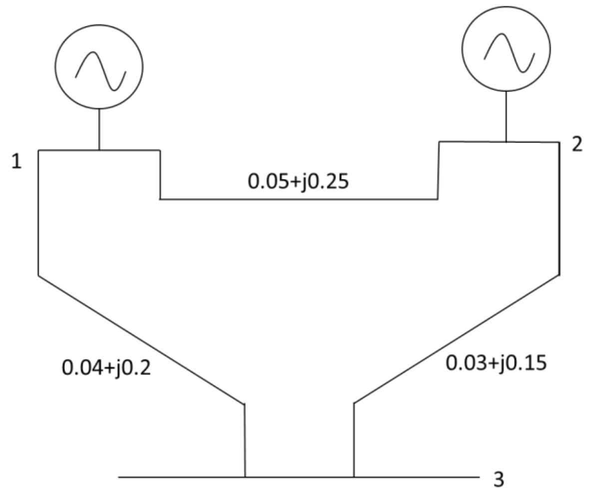

Fig. 6.1: Diagram of a three- bus system

6.6 Data table

Bus	|v|	Pg	Qg	Pl	Ql	Angle
1	1.03	0	0	0	0	0
2	1.02	0.8	0	0.4	0.3	0
3	1	0	0	1	0.8	0

Fig. 6.1 Excel file of the Load data

Bus	Bus	R	X
1	1	0	1
1	2	0.05	0.25
1	3	0.04	0.02
2	2	0	1
2	3	0.03	0.15

Fig. 6.2 Excel file of the Impedance data

6.7 Algorithm

1 Start

2 Read an excel file that has bus numbers in the first two columns and resistance and admittance value in the third and fourth columns which represent impedance between the buses

3 Construct a matrix $\boldsymbol{M_{imp}}$ whose element $\boldsymbol{a_{i,j}}$ denotes the impedance between i and j buses

4 Perform element wise inversion of $\boldsymbol{M_{imp}}$ matrix and keep it in y matrix

5 Use the formulas to construct the $\boldsymbol{Y_{bus}}$ matrix $\boldsymbol{Y_{i,j}|_{i=j} = \sum_0^j \frac{1}{y_{(i,j)}}}$ and $\boldsymbol{y_{i,j}|_{i\neq j} = \frac{-1}{y_{(i,j)}}}$

6 Read the bus parameters from an excel file where P and V are defined for generator buses and P and Q are defined for load buses and P and δ are defined

7 Assume V = 1 and δ = 0° for nodes if not defined

8 Calculate the complex power as follows:

9 $\boldsymbol{S_i^j = (P_{g_i} - P_{l_i}) + i \times Im(V_i^{j-1} \times (\sum_{k=1}^n Y_{i,k} \times V_k^{mostrecent})^*)}$

10 Calculate the jacobian $\boldsymbol{J = \begin{bmatrix} H & L \\ M & N \end{bmatrix}}$ as follows:

$\boldsymbol{H = \frac{\partial P}{\partial \delta}}$; $\boldsymbol{L = \frac{\partial P}{\partial V}}$; $\boldsymbol{M = \frac{\partial Q}{\partial \delta}}$; $\boldsymbol{H = \frac{\partial Q}{\partial V}}$

11 where bus real power, $\boldsymbol{P_i = \sum_{j=1}^n |V_i||Y_{i,j}||V_j|\cos(\theta_i - \delta_i + \delta_j)}$ and bus reactive power, $\boldsymbol{Q_i = -\sum_{j=1}^n |V_i||Y_{i,j}||V_j|\sin(\theta_i - \delta_i + \delta_j)}$

12 Calculate the power mismatches as follows: $\boldsymbol{\Delta P = P_g - P_l - \mathrm{Re}(S)}$ and $\boldsymbol{\Delta Q = -Q_l - \mathrm{Im}(S)}$

13 Calculate the phase and amplitude differences of buses using $\boldsymbol{\Delta(\delta, V) = J^{-1} \times \Delta(P, Q)}$

14 Correct the bus voltages using $\boldsymbol{V^{i+1} = V^i + \Delta V}$ and $\boldsymbol{\delta^{i+1} = \delta^i + \Delta\delta}$

15 Calculate error as: - $\boldsymbol{\varepsilon = |V_i^{j-1} - V_i^j|}$

16 If ε < tolerance value goes to next step otherwise go to step 8

17 End

6.8 MATLAB Code & Output

```
clc;  %Clears previous data from command window
clear all; %Removes all variables from the current workspace
cd('F:\Study material\Lab\3-2\Power System I'); %change the file directory
A = xlsread('EXp02p02'); %Read the excel file
```

```
n = length(A); %Determine the length of the excel file
% Applying symmetric condition
for w=1:n
    Z(A(w,1),A(w,2)) = A(w,3)+i*A(w,4);
    Z(A(w,2),A(w,1)) = A(w,3)+i*A(w,4);
end
m = length(Z); %Determine the length of the new matrix
for j=1:m
    for k=1:m
        if Z(j,k) == 0
            Z(j,k) = inf;
        end
    end
end
fprintf(' Z matrix is \n') %Display the text
disp(Z) %Display the output
y = 1./Z %Taking inverse impedance matrix
p = sum(y,2) %Taking symmetric summation
%Apply looping condition to determine value of the matrix element
for u=1:m
    for x=1:m
        if u~=x
            Y(u,x)= -y(u,x); %For diagonal element
        else
            Y(u,x)= p(x); %For non-diagonal element
        end
    end
end
fprintf(' Y- bus matrix is \n') %Display the text
disp(Y) %Display the output
cd('F:\Study material\Lab\3-2\Power System I'); %change the file directory
LFS = xlsread('Exp05'); %Read the excel file
busnum=LFS(:,1); bustype=LFS(:,1); nbus=length(busnum); voltage=LFS(:,2);
Angle=LFS(:,7); Pg=LFS(:,3); Pl=LFS(:,5); Qg=LFS(:,4); Ql=LFS(:,6);
P=LFS(:,3)-LFS(:,5); Q=LFS(:,3)-LFS(:,5); Psp=P; Qsp=P;
for j=1:nbus
    v(j)=voltage(j)*exp(1i*Angle(j));
end
for i=1:nbus
    bustype(1)=busnum(1);
    if Pg(i)~=0&&Qg(i)==0&&v(i)~=1
        bustype(i)=2;
    else if P(i)~=0&&Q(i)~=0&&v(i)==1
            bustype(i)=3;
        end
    end
end
w=real(v); u=imag(v); [del V]=cart2pol(w,u); vnew=V; w=real(Y); u=imag(Y);
[theta Y]=cart2pol(w,u); pv=find(bustype==2); pq=find(bustype==3);
npv=length(pv); npq=length(pq); delv=1; count=0; tol=0.1
v1=0;
v2=0;
v3=0;
V=vnew;
P=zeros(nbus,1);
Q=zeros(nbus,1);
for i=1:nbus
        for k=1:nbus
            P(i)=P(i)+(V(i)*Y(i,k)*V(k)*cos(theta(i,k)-del(i)+del(k)));
            Q(i)=Q(i)-(V(i)*Y(i,k)*V(k)*sin(theta(i,k)-del(i)+del(k)));
        end
```

```
    end
    dPp=Psp-P;
    dQp=Qsp-Q;
    k=1;
    dQ=zeros(npq,1);
    for i=1:npq
        if bustype(i)==2
            dQ(k)=dQp(i);
        end
    end
    dP=dPp(2:nbus);
    M=[dP;dQ];
    J1=zeros(nbus-1,nbus-1);
    for i=1:nbus-1
        m=i+1;
        for k=1:nbus-1
            n=k+1;
            if m==n
                for n=1:nbus
                    J1(i,k)=J1(i,k)+(V(m)*Y(m,n)*V(n)*sin(theta(m,n)-
del(m)+del(n)));
                end
                J1(i,k)=J1(i,k)-(V(m)*Y(m,n)*V(m)*sin(theta(m,m)));  %hh
            else
            J1(i,k)=-(V(m)*Y(m,n)*V(n)*sin(theta(m,n)-del(m)+del(n)));
            end
        end
    end
    J2=zeros(nbus-1,npq);
    for i=1:nbus-1
        m=i+1;
         for k=1:npq
            n=pq(k);
            if m==n
                for n=1:nbus
                    J2(i,k)=J2(i,k)+(V(m)*Y(m,n)*V(n)*cos(theta(m,n)-
del(m)+del(n)));
                end
                J2(i,k)=J2(i,k)+(2*V(m)*Y(m,n)*cos(theta(m,m)));
            else
            J2(i,k)=(V(m)*Y(m,n)*sin(theta(m,n)-del(m)+del(n)));
            end
        end
    end
         J3=zeros(npq,nbus-1);
     for i=1:npq
        m=pq(i);
         for k=1:nbus-1
            n=k+1;
            if m==n
                for n=1:nbus
                    J3(i,k)=J3(i,k)+(V(m)*Y(m,n)*V(n)*cos(theta(m,n)-
del(m)+del(n)));
                end
                J3(i,k)=J3(i,k)-(V(m)*Y(m,m)*V(m)*cos(theta(m,m)));  %h
            else
            J3(i,k)=-(V(m)*Y(m,n)*V(n)*cos(theta(m,n)-del(m)+del(n)));
            end
        end
     end
          J4=zeros(npq,npq);
```

```
     for i=1:npq
        m=pq(i);
         for k=1:npq
            n=pq(k);
            if m==n
                for n=1:nbus
                    J4(i,k)=J4(i,k)-(V(n)*Y(m,n)*sin(theta(m,n)-
del(m)+del(n)));
                end
                J4(i,k)=J4(i,k)-(2*V(m)*Y(m,m)*sin(theta(m,m)));  %h
            else
            J4(i,k)=-(V(m)*Y(m,n)*sin(theta(m,n)-del(m)+del(n)));
            end
        end
     end
J=[J1 J2;J3 J4]
X=inv(J)*M;
dth=X(1:nbus-npq);
dV=X(npv+1:nbus);
for i=2:nbus
    del(i)=del(i)+dth(i-1);
end
for i=1:nbus
    k=1;
    if(bustype(i)==3)
        vnew(i)=V(i)+dV(k);
    end
    k=k+1;
end
for i=1:nbus
   I=[vnew(i) del(i)];
end
%disp
count=count+1; delv=abs(vnew-V); I=[count vnew del delv];
disp('Iteration Magnitude Angle Error'); disp(I); Bus=busnum;
Magnitude=vnew'; ang=radtodeg(del); Angle=ang';
T=table(Bus,Magnitude,Angle);
disp(T);
```

Output

Z matrix is

0.0000 + 1.0000i   0.0500 + 0.2500i   0.0400 + 0.0200i

0.0500 + 0.2500i   0.0000 + 1.0000i   0.0300 + 0.1500i

0.0400 + 0.0200i   0.0300 + 0.1500i   Inf + 0.0000i

y =

0.0000 - 1.0000i   0.7692 - 3.8462i 20.0000 -10.0000i

0.7692 - 3.8462i   0.0000 - 1.0000i   1.2821 - 6.4103i

20.0000 -10.0000i   1.2821 - 6.4103i   0.0000 + 0.0000i

p =

20.7692 -14.8462i

2.0513 -11.2564i

21.2821 -16.4103i

Y- bus matrix is

20.7692 -14.8462i -0.7692 + 3.8462i -20.0000 +10.0000i

-0.7692 + 3.8462i   2.0513 -11.2564i -1.2821 + 6.4103i

-20.0000 +10.0000i -1.2821 + 6.4103i 21.2821 -16.4103i

J =

5.5592   -6.5385   6.5385

-6.5385   16.8385   41.9385

1.3077 -21.9077   32.3923

Iteration Magnitude Angle Error

1.0000   1.0300   1.0200   1.0052   0   0.0742   0.0052   0   0   0.0052

Bus	**Magnitude**	**Angle**
1	**1.03**	**0**
2	**1.02**	**4.2497**
3	**1.0052**	**0.29937**

6.9 Discussion & Conclusion

In this experiment, we designed an algorithm, flow chart, and programmed a generalized code for load flow study of a given power system. The sinusoidal steady state of the entire system is provided by the load flow. The code was generated to read the bus parameters from another excel file, solve the bus voltages, and then calculate the admittance matrix using impedances in an excel file.

The only adjustment to the code we may need is changing the directory of the file to work with and the given data saved inside the file. The bus numbers and the resistance and reactance values must also be given in the order defined for the code to work and give accurate result.