MASTER OF TECHNOLOGY

EE 572

SEMESTER II

2023

POWER ENGINEERING LABORATORY

REPORT

UNDER THE GUIDANCE

OF

DR. SANJIB GANGULY

SUBMITTED

BY

SHIVRAJ VISHWAKARMA

(224102112)

ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT

POWER ENGINEERING LAB

ASSIGNMENT – 01 (18/01/23)

NAME: SHIVRAJ VISHWAKARMA **ROLL. NO.:** 224102112

OBJECTIVE:

For the standard IEEE 30 bus System given in the reference problem of NR Load Flow, Take the guess values for V & δ accordingly in a range and plot V vs δ by indicating (V, δ) by GREEN STAR (*) where NR converges & the (V, δ) by RED STAR (*) where NR not converges.

SOFTWARE: MATLAB R2022a

REFERECE PROBLEM: Obtain the power flow solution by the Newton-Raphson method for the system of given fig:

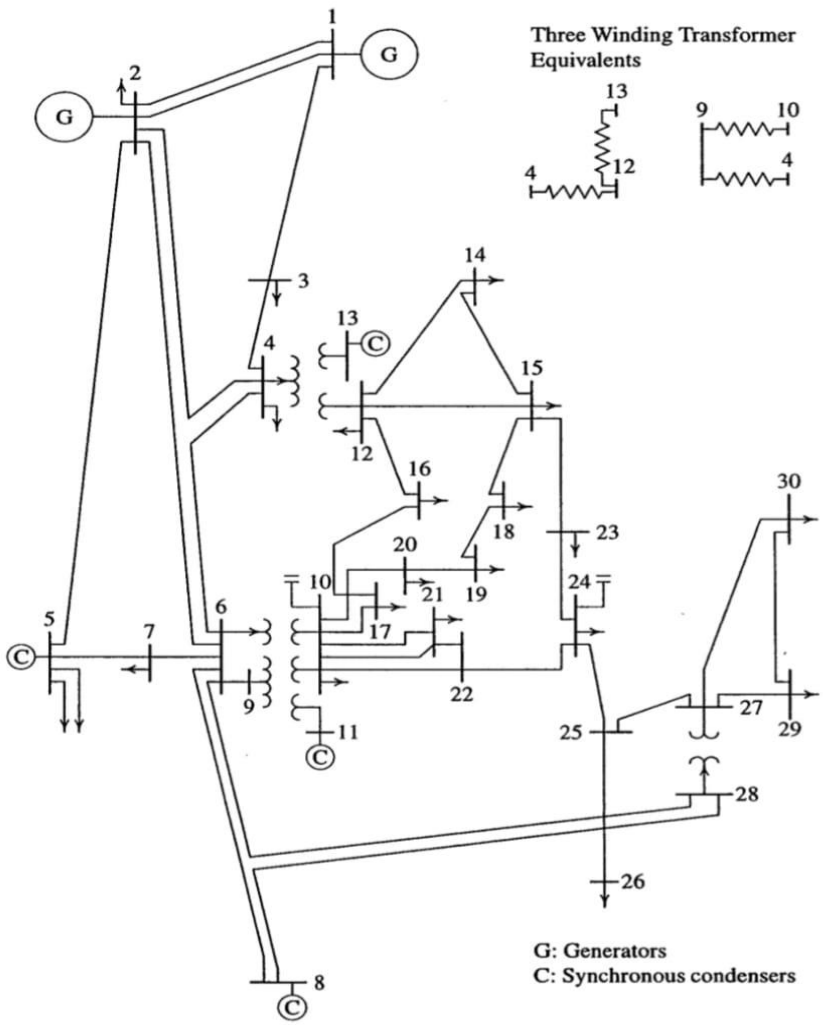


FIGURE 6.16
30-bus IEEE sample system.

The datafile for the above system prepared in MATLAB:

```
clear
basemva = 100;  accuracy = 0.00001; maxiter = 100;

%        IEEE 30-BUS TEST SYSTEM (American Electric Power)
%        Bus Bus  Voltage Angle   ---Load---- -------Generator----- Injected
%        No  code Mag.    Degree  MW    Mvar  MW  Mvar Qmin Qmax     Mvar
busdata=[1   1    1.06    0.0     0.0   0.0   0.0  0.0   0   0       0
         2   2    1.043   0.0    21.70 12.7   40.0  0.0 -40  50       0
         3   0    1.0     0.0     2.4   1.2   0.0  0.0   0   0       0
         4   0    1.06    0.0     7.6   1.6   0.0  0.0   0   0       0
         5   2    1.01    0.0    94.2  19.0   0.0  0.0 -40  40       0
         6   0    1.0     0.0     0.0   0.0   0.0  0.0   0   0       0
         7   0    1.0     0.0    22.8  10.9   0.0  0.0   0   0       0
         8   2    1.01    0.0    30.0  30.0   0.0  0.0 -10  60       0
         9   0    1.0     0.0     0.0   0.0   0.0  0.0   0   0       0
        10   0    1.0     0.0     5.8   2.0   0.0  0.0  -6  24      19
        11   2    1.082   0.0     0.0   0.0   0.0  0.0   0   0       0
        12   0    1.0     0       11.2  7.5   0    0     0   0       0
        13   2    1.071   0        0    0.0   0    0    -6  24       0
        14   0    1       0       6.2   1.6   0    0     0   0       0
        15   0    1       0       8.2   2.5   0    0     0   0       0
        16   0    1       0       3.5   1.8   0    0     0   0       0
        17   0    1       0       9.0   5.8   0    0     0   0       0
        18   0    1       0       3.2   0.9   0    0     0   0       0
        19   0    1       0       9.5   3.4   0    0     0   0       0
        20   0    1       0       2.2   0.7   0    0     0   0       0
        21   0    1       0      17.5  11.2   0    0     0   0       0
        22   0    1       0       0     0.0   0    0     0   0       0
        23   0    1       0       3.2   1.6   0    0     0   0       0
        24   0    1       0       8.7   6.7   0    0     0   0      4.3
        25   0    1       0       0     0.0   0    0     0   0       0
        26   0    1       0       3.5   2.3   0    0     0   0       0
        27   0    1       0       0     0.0   0    0     0   0       0
        28   0    1       0       0     0.0   0    0     0   0       0
        29   0    1       0       2.4   0.9   0    0     0   0       0
        30   0    1       0      10.6   1.9   0    0     0   0       0];
%                                          Line code
%         Bus bus   R      X     1/2 B   = 1 for lines
%         nl  nr  p.u.   p.u.   p.u.     > 1 or < 1 tr. tap at bus nl
linedata=[1   2   0.0192   0.0575   0.02640    1
          1   3   0.0452   0.1852   0.02040    1
          2   4   0.0570   0.1737   0.01840    1
          3   4   0.0132   0.0379   0.00420    1
          2   5   0.0472   0.1983   0.02090    1
          2   6   0.0581   0.1763   0.01870    1
          4   6   0.0119   0.0414   0.00450    1
          5   7   0.0460   0.1160   0.01020    1
          6   7   0.0267   0.0820   0.00850    1
          6   8   0.0120   0.0420   0.00450    1
          6   9   0.0      0.2080   0.0    0.978
          6  10   0         .5560   0      0.969
          9  11   0         .2080   0          1
          9  10   0         .1100   0          1
          4  12   0         .2560   0      0.932
         12  13   0         .1400   0          1
```

```
12  14    .1231    .2559   0          1
12  15    .0662    .1304   0          1
12  16    .0945    .1987   0          1
14  15    .2210    .1997   0          1
16  17    .0824    .1923   0          1
15  18    .1073    .2185   0          1
18  19    .0639    .1292   0          1
19  20    .0340    .0680   0          1
10  20    .0936    .2090   0          1
10  17    .0324    .0845   0          1
10  21    .0348    .0749   0          1
10  22    .0727    .1499   0          1
21  22    .0116    .0236   0          1
15  23    .1000    .2020   0          1
22  24    .1150    .1790   0          1
23  24    .1320    .2700   0          1
24  25    .1885    .3292   0          1
25  26    .2544    .3800   0          1
25  27    .1093    .2087   0          1
28  27     0       .3960   0      0.968
27  29    .2198    .4153   0          1
27  30    .3202    .6027   0          1
29  30    .2399    .4533   0          1
 8  28    .0636    .2000   0.0214     1
 6  28    .0169    .0599   0.065      1];
```

SIMULATION PROCEDURE:

1. First we have to update the guess values by our new guess values of V & δ. Which can be chosen as:

M = [V**:** δ], where

V = [0.1 0.2 0.3 0.4 ……. 3.0] ^{T} & δ = [-14 -13 -12 ……0……. +14 +15] ^{T}

So first we'll write the codes to update the bus data & then for each V & δ we'll run the below commands to get NR Load flow solution.

- **LFYBUS**
- **LFNEWTON**
  - (After **LFNEWTON**, we have to plot the V & δ with the conditions given in objective.)
- **LINEFLOW**
- **BUSOUT**

MATLAB CODES:

LFYBUS

```
% This program obtains the Bus Admittance Matrix for power flow
solution
```

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

LFNEWTON

```
%   Power flow solution by Newton-Raphson method
ns=0; ng=0; Vm=0; delta=0; yload=0; deltad=0;
nbus = length(busdata(:,1));
for k=1:nbus
n=busdata(k,1);
kb(n)=busdata(k,2); Vm(n)=busdata(k,3); delta(n)=busdata(k, 4);
Pd(n)=busdata(k,5); Qd(n)=busdata(k,6); Pg(n)=busdata(k,7); Qg(n) =
busdata(k,8);
Qmin(n)=busdata(k, 9); Qmax(n)=busdata(k, 10);
Qsh(n)=busdata(k, 11);
    if Vm(n) <= 0  Vm(n) = 1.0; V(n) = 1 + j*0;
    else delta(n) = pi/180*delta(n);
         V(n) = Vm(n)*(cos(delta(n)) + j*sin(delta(n)));
         P(n)=(Pg(n)-Pd(n))/basemva;
         Q(n)=(Qg(n)-Qd(n)+ Qsh(n))/basemva;
         S(n) = P(n) + j*Q(n);
    end
end
for k=1:nbus
if kb(k) == 1, ns = ns+1; else, end
if kb(k) == 2 ng = ng+1; else, end
ngs(k) = ng;
```

```
nss(k) = ns;
end
Ym=abs(Ybus); t = angle(Ybus);
m=2*nbus-ng-2*ns;
maxerror = 1; converge=1;
iter = 0;
% Start of iterations
clear A  DC   J  DX
while maxerror >= accuracy & iter <= maxiter % Test for max. power mismatch
for i=1:m
for k=1:m
   A(i,k)=0;      %Initializing Jacobian matrix
end, end
iter = iter+1;

for n=1:nbus
nn=n-nss(n);
lm=nbus+n-ngs(n)-nss(n)-ns;
J11=0; J22=0; J33=0; J44=0;
   for i=1:nbr
     if nl(i) == n | nr(i) == n
        if nl(i) == n,  l = nr(i); end
        if nr(i) == n,  l = nl(i); end
        J11=J11+ Vm(n)*Vm(l)*Ym(n,l)*sin(t(n,l)- delta(n) + delta(l));
        J33=J33+ Vm(n)*Vm(l)*Ym(n,l)*cos(t(n,l)- delta(n) + delta(l));
        if kb(n)~=1
        J22=J22+ Vm(l)*Ym(n,l)*cos(t(n,l)- delta(n) + delta(l));
        J44=J44+ Vm(l)*Ym(n,l)*sin(t(n,l)- delta(n) + delta(l));
        else, end
        if kb(n) ~= 1  & kb(l) ~=1
        lk = nbus+l-ngs(l)-nss(l)-ns;
        ll = l -nss(l);
      % off diagonalelements of J1
        A(nn, ll) =-Vm(n)*Vm(l)*Ym(n,l)*sin(t(n,l)- delta(n) + delta(l));
              if kb(l) == 0  % off diagonal elements of J2
              A(nn, lk) =Vm(n)*Ym(n,l)*cos(t(n,l)- delta(n) + delta(l));end
              if kb(n) == 0  % off diagonal elements of J3
              A(lm, ll) =-Vm(n)*Vm(l)*Ym(n,l)*cos(t(n,l)- delta(n)+delta(l)); end
              if kb(n) == 0 & kb(l) == 0  % off diagonal elements of  J4
              A(lm, lk) =-Vm(n)*Ym(n,l)*sin(t(n,l)- delta(n) + delta(l));end
        else end
     else , end
   end
   Pk = Vm(n)^2*Ym(n,n)*cos(t(n,n))+J33;
   Qk = -Vm(n)^2*Ym(n,n)*sin(t(n,n))-J11;
   if kb(n) == 1 P(n)=Pk; Q(n) = Qk; end   % Swing bus P
     if kb(n) == 2  Q(n)=Qk;
         if Qmax(n) ~= 0
           Qgc = Q(n)*basemva + Qd(n) - Qsh(n);
```

```
            if iter <= 7                  % Between the 2th & 6th iterations
               if iter > 2                % the Mvar of generator buses are
                 if Qgc  < Qmin(n),       % tested. If not within limits Vm(n)
                 Vm(n) = Vm(n) + 0.01;    % is changed in steps of 0.01 pu to
                 elseif Qgc  > Qmax(n),   % bring the generator Mvar within
                 Vm(n) = Vm(n) - 0.01;end % the specified limits.
               else, end
            else,end
          else,end
      end
    if kb(n) ~= 1
      A(nn,nn) = J11;  %diagonal elements of J1
      DC(nn) = P(n)-Pk;
    end
    if kb(n) == 0
      A(nn,lm) = 2*Vm(n)*Ym(n,n)*cos(t(n,n))+J22;  %diagonal elements of J2
      A(lm,nn)= J33;        %diagonal elements of J3
      A(lm,lm) =-2*Vm(n)*Ym(n,n)*sin(t(n,n))-J44;  %diagonal of elements of J4
      DC(lm) = Q(n)-Qk;
    end
end
DX=A\DC';
for n=1:nbus
  nn=n-nss(n);
  lm=nbus+n-ngs(n)-nss(n)-ns;
    if kb(n) ~= 1
    delta(n) = delta(n)+DX(nn); end
    if kb(n) == 0
    Vm(n)=Vm(n)+DX(lm); end
 end
        maxerror=max(abs(DC));

           if iter == maxiter & maxerror > accuracy
         fprintf('\nWARNING: Iterative solution did not converged after ')
         fprintf('%g', iter), fprintf(' iterations.\n\n')
         converge = 0;
      %    fprintf('Press Enter to terminate the iterations and print the results \n')
      %     pause,
           else, end
            %plot(iter,maxerror,'ro')
           %hold on
           %pause(1)
      end

      if converge ~= 1
```

```
        tech= ('                    ITERATIVE SOLUTION DID NOT CONVERGE'); else,
        tech=('                  Power Flow Solution by Newton-Raphson Method');
    end
    V = Vm.*cos(delta)+j*Vm.*sin(delta);
    deltad=180/pi*delta;
    i=sqrt(-1);
    k=0;
    for n = 1:nbus
         if kb(n) == 1
         k=k+1;
         S(n)= P(n)+j*Q(n);
         Pg(n) = P(n)*basemva + Pd(n);
         Qg(n) = Q(n)*basemva + Qd(n) - Qsh(n);
         Pgg(k)=Pg(n);
         Qgg(k)=Qg(n);     %june 97
         elseif  kb(n) ==2
         k=k+1;
         S(n)=P(n)+j*Q(n);
         Qg(n) = Q(n)*basemva + Qd(n) - Qsh(n);
         Pgg(k)=Pg(n);
         Qgg(k)=Qg(n);  % June 1997
       end
    yload(n) = (Pd(n)- j*Qd(n)+j*Qsh(n))/(basemva*Vm(n)^2);
    end
    busdata(:,3)=Vm'
    busdata(:,4)=deltad'
    Pgt = sum(Pg);  Qgt = sum(Qg); Pdt = sum(Pd); Qdt = sum(Qd); Qsht = sum(Qsh);

    %clear A DC DX  J11 J22 J33 J44 Qk delta lk ll lm
    %clear A DC DX  J11 J22 J33  Qk delta lk ll lm
```

LINEFLOW

```
% This program is used in conjunction with lfgauss or lf Newton
% for the computation of line flow and line losses.

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

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

```

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

BUSOUT

```
%  This program prints the power flow solution in a tabulated form
%  on the screen.
%
%  Copyright (C) 1998 by H. Saadat.

%clc
disp(tech)
fprintf('                      Maximum Power Mismatch = %g \n', maxerror)
fprintf('                              No. of Iterations = %g \n\n', iter)
head =['    Bus  Voltage  Angle    ------Load------    ---Generation--- Injected'
       '    No.  Mag.     Degree     MW       Mvar       MW       Mvar  Mvar '
       '
'];
disp(head)
```

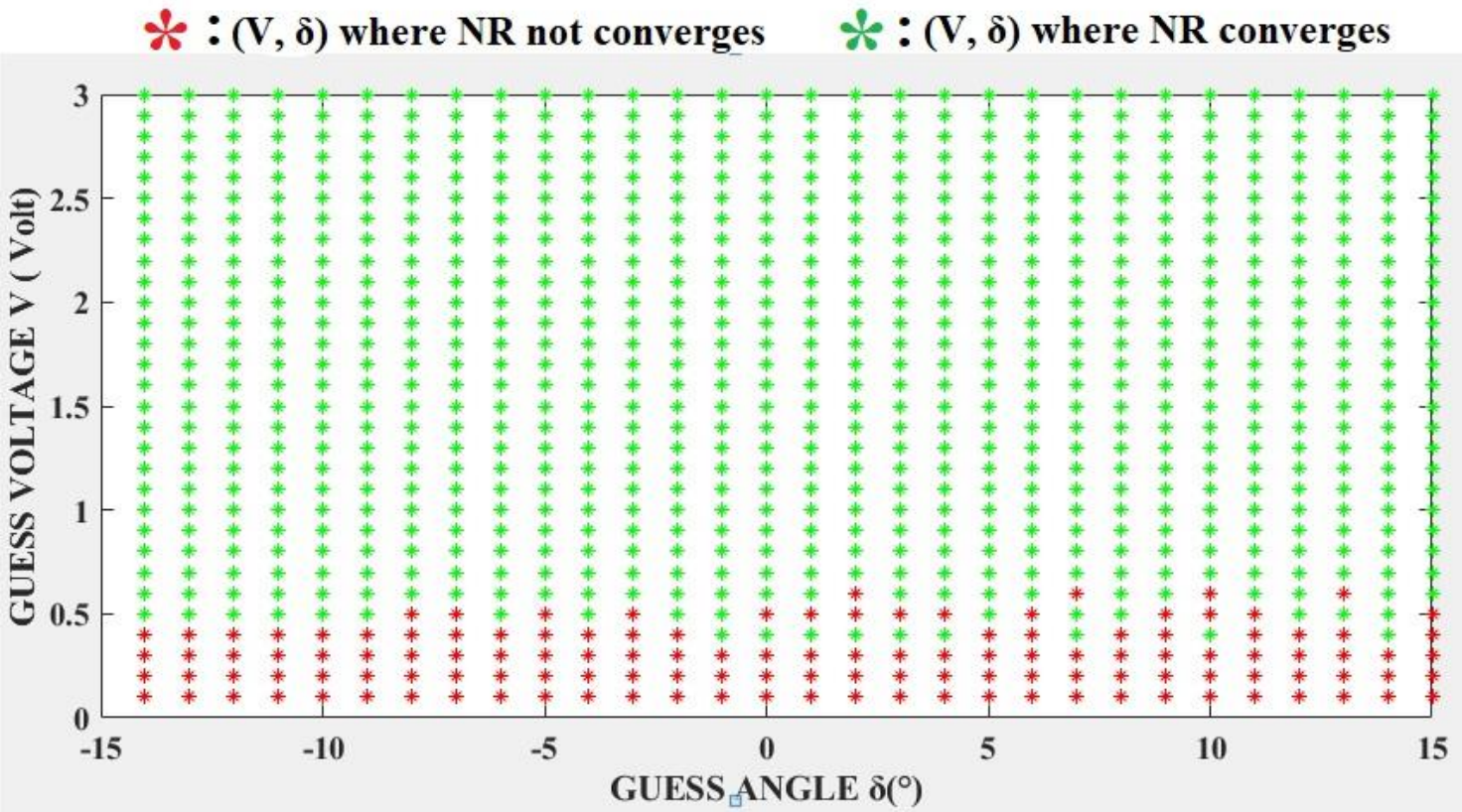
```
for n=1:nbus
     fprintf(' %5g', n), fprintf(' %7.3f', Vm(n)),
     fprintf(' %8.3f', deltad(n)), fprintf(' %9.3f', Pd(n)),
     fprintf(' %9.3f', Qd(n)),  fprintf(' %9.3f', Pg(n)),
     fprintf(' %9.3f ', Qg(n)), fprintf(' %8.3f\n', Qsh(n))
end
    fprintf('      \n'), fprintf('    Total              ')
    fprintf(' %9.3f', Pdt), fprintf(' %9.3f', Qdt),
    fprintf(' %9.3f', Pgt), fprintf(' %9.3f', Qgt), fprintf('
%9.3f\n\n', Qsht)
```

FINAL REQUIRED MATLAB CODE.

```
clear
clc
Datafile;
busdata;
nbus = length(busdata(:,1));
bc = busdata(:,2);
M=zeros(30,2);
M(:,1) = 0.1:0.1:3;
M(:,2) = -14:1:15;
Vlt=M(:,1);
delt=M(:,2);
for q=1:length(Vlt)
    for u=1:length(delt)
    for n = 1:nbus
    if bc(n) == 0
        busdata(n,3) = Vlt(q);
        busdata(n,4)= delt(u);
    else
        if bc(n) == 2
            busdata(n,4) = delt(u);
        end
    end
    end
busdata
LFYBUS
LFNEWTON
if converge ~= 1
    plot(delt(u),Vlt(q),'r*')
    hold on
else
    plot(delt(u), Vlt(q),'g*')
      hold on
end
LINEFLOW
BUSOUT
end
end
```

OBTAINED RESULTS:

After running the above codes we'll obtain the below V vs δ graph which represents the points where NR Load Flow solution Converges & where it si not converges.



⇒From the above fig we can see that at what values of V & δ, Newton Raphson load flow solution converges.

since at (V, δ) = (3 V, 15°) it converges so the NRLFS can be given as:

busdata =

1.0000	1.0000	1.0600	0	0	0	0	0	0	0	0
2.0000	2.0000	1.0530	15.0000	21.7000	12.7000	40.0000	0	-40.0000	50.0000	0
3.0000	0	3.0000	15.0000	2.4000	1.2000	0	0	0	0	0
4.0000	0	3.0000	15.0000	7.6000	1.6000	0	0	0	0	0
5.0000	2.0000	1.0300	15.0000	94.2000	19.0000	0	0	-40.0000	40.0000	0
6.0000	0	3.0000	15.0000	0	0	0	0	0	0	0
7.0000	0	3.0000	15.0000	22.8000	10.9000	0	0	0	0	0
8.0000	2.0000	1.0400	15.0000	30.0000	30.0000	0	0	-10.0000	60.0000	0
9.0000	0	3.0000	15.0000	0	0	0	0	0	0	0
10.0000	0	3.0000	15.0000	5.8000	2.0000	0	0	-6.0000	24.0000	19.0000
11.0000	2.0000	1.0820	15.0000	0	0	0	0	0	0	0
12.0000	0	3.0000	15.0000	11.2000	7.5000	0	0	0	0	0
13.0000	2.0000	1.1210	15.0000	0	0	0	0	-6.0000	24.0000	0
14.0000	0	3.0000	15.0000	6.2000	1.6000	0	0	0	0	0

15.0000	0	3.0000	15.0000	8.2000	2.5000	0	0	0	0	0
16.0000	0	3.0000	15.0000	3.5000	1.8000	0	0	0	0	0
17.0000	0	3.0000	15.0000	9.0000	5.8000	0	0	0	0	0
18.0000	0	3.0000	15.0000	3.2000	0.9000	0	0	0	0	0
19.0000	0	3.0000	15.0000	9.5000	3.4000	0	0	0	0	0
20.0000	0	3.0000	15.0000	2.2000	0.7000	0	0	0	0	0
21.0000	0	3.0000	15.0000	17.5000	11.2000	0	0	0	0	0
22.0000	0	3.0000	15.0000	0	0	0	0	0	0	0
23.0000	0	3.0000	15.0000	3.2000	1.6000	0	0	0	0	0
24.0000	0	3.0000	15.0000	8.7000	6.7000	0	0	0	0	4.3000
25.0000	0	3.0000	15.0000	0	0	0	0	0	0	0
26.0000	0	3.0000	15.0000	3.5000	2.3000	0	0	0	0	0
27.0000	0	3.0000	15.0000	0	0	0	0	0	0	0
28.0000	0	3.0000	15.0000	0	0	0	0	0	0	0
29.0000	0	3.0000	15.0000	2.4000	0.9000	0	0	0	0	0
30.0000	0	3.0000	15.0000	10.6000	1.9000	0	0	0	0	0

busdata =

1.0000	1.0000	1.0600	0	0	0	0	0	0	0	0
2.0000	2.0000	1.0530	15.0000	21.7000	12.7000	40.0000	0	-40.0000	50.0000	0
3.0000	0	1.0378	15.0000	2.4000	1.2000	0	0	0	0	0
4.0000	0	1.0326	15.0000	7.6000	1.6000	0	0	0	0	0
5.0000	2.0000	1.0300	15.0000	94.2000	19.0000	0	0	-40.0000	40.0000	0
6.0000	0	1.0353	15.0000	0	0	0	0	0	0	0
7.0000	0	1.0256	15.0000	22.8000	10.9000	0	0	0	0	0
8.0000	2.0000	1.0400	15.0000	30.0000	30.0000	0	0	-10.0000	60.0000	0
9.0000	0	1.0686	15.0000	0	0	0	0	0	0	0
10.0000	0	1.0679	15.0000	5.8000	2.0000	0	0	-6.0000	24.0000	19.0000
11.0000	2.0000	1.0820	15.0000	0	0	0	0	0	0	0
12.0000	0	1.0932	15.0000	11.2000	7.5000	0	0	0	0	0
13.0000	2.0000	1.1210	15.0000	0	0	0	0	-6.0000	24.0000	0
14.0000	0	1.0772	15.0000	6.2000	1.6000	0	0	0	0	0
15.0000	0	1.0712	15.0000	8.2000	2.5000	0	0	0	0	0
16.0000	0	1.0755	15.0000	3.5000	1.8000	0	0	0	0	0
17.0000	0	1.0650	15.0000	9.0000	5.8000	0	0	0	0	0
18.0000	0	1.0582	15.0000	3.2000	0.9000	0	0	0	0	0
19.0000	0	1.0536	15.0000	9.5000	3.4000	0	0	0	0	0
20.0000	0	1.0564	15.0000	2.2000	0.7000	0	0	0	0	0
21.0000	0	1.0565	15.0000	17.5000	11.2000	0	0	0	0	0
22.0000	0	1.0573	15.0000	0	0	0	0	0	0	0
23.0000	0	1.0581	15.0000	3.2000	1.6000	0	0	0	0	0
24.0000	0	1.0489	15.0000	8.7000	6.7000	0	0	0	0	4.3000
25.0000	0	1.0460	15.0000	0	0	0	0	0	0	0
26.0000	0	1.0288	15.0000	3.5000	2.3000	0	0	0	0	0

27.0000	0	1.0525	15.0000	0	0	0	0	0	0	0
28.0000	0	1.0358	15.0000	0	0	0	0	0	0	0
29.0000	0	1.0333	15.0000	2.4000	0.9000	0	0	0	0	0
30.0000	0	1.0221	15.0000	10.6000	1.9000	0	0	0	0	0

busdata =

1.0000	1.0000	1.0600	0	0	0	0	0	0	0	0
2.0000	2.0000	1.0530	-5.6069	21.7000	12.7000	40.0000	0	-40.0000	50.0000	0
3.0000	0	1.0378	-8.1448	2.4000	1.2000	0	0	0	0	0
4.0000	0	1.0326	-9.8196	7.6000	1.6000	0	0	0	0	0
5.0000	2.0000	1.0300	-14.3335	94.2000	19.0000	0	0	-40.0000	40.0000	0
6.0000	0	1.0353	-11.5211	0	0	0	0	0	0	0
7.0000	0	1.0256	-13.1817	22.8000	10.9000	0	0	0	0	0
8.0000	2.0000	1.0400	-12.3101	30.0000	30.0000	0	0	-10.0000	60.0000	0
9.0000	0	1.0686	-14.3559	0	0	0	0	0	0	0
10.0000	0	1.0679	-15.8415	5.8000	2.0000	0	0	-6.0000	24.0000	19.0000
11.0000	2.0000	1.0820	-14.3559	0	0	0	0	0	0	0
12.0000	0	1.0932	-15.3274	11.2000	7.5000	0	0	0	0	0
13.0000	2.0000	1.1210	-15.3274	0	0	0	0	-6.0000	24.0000	0
14.0000	0	1.0772	-16.1568	6.2000	1.6000	0	0	0	0	0
15.0000	0	1.0712	-16.2025	8.2000	2.5000	0	0	0	0	0
16.0000	0	1.0755	-15.7938	3.5000	1.8000	0	0	0	0	0
17.0000	0	1.0650	-16.0214	9.0000	5.8000	0	0	0	0	0
18.0000	0	1.0582	-16.7314	3.2000	0.9000	0	0	0	0	0
19.0000	0	1.0536	-16.8657	9.5000	3.4000	0	0	0	0	0
20.0000	0	1.0564	-16.6630	2.2000	0.7000	0	0	0	0	0
21.0000	0	1.0565	-16.2726	17.5000	11.2000	0	0	0	0	0
22.0000	0	1.0573	-16.2617	0	0	0	0	0	0	0
23.0000	0	1.0581	-16.5392	3.2000	1.6000	0	0	0	0	0
24.0000	0	1.0489	-16.6654	8.7000	6.7000	0	0	0	0	4.3000
25.0000	0	1.0460	-16.2932	0	0	0	0	0	0	0
26.0000	0	1.0288	-16.6898	3.5000	2.3000	0	0	0	0	0
27.0000	0	1.0525	-15.8163	0	0	0	0	0	0	0
28.0000	0	1.0358	-12.1735	0	0	0	0	0	0	0
29.0000	0	1.0333	-16.9775	2.4000	0.9000	0	0	0	0	0
30.0000	0	1.0221	-17.8096	10.6000	1.9000	0	0	0	0	0

Line Flow and Losses

--Line--		Power at bus & line flow			--Line loss--		Transformer
from	to	MW	Mvar	MVA	MW	Mvar	tap
1		261.031	-43.998	264.713			

	2	177.300	-39.977	181.751	5.606	10.894
	3	83.731	-4.021	83.827	2.822	7.071
2		18.300	35.979	40.366		
	1	-171.694	50.872	179.072	5.606	10.894
	4	45.684	-2.951	45.779	1.073	-0.731
	5	82.710	-3.459	82.783	2.913	7.702
	6	61.600	-8.482	62.182	2.010	2.021
3		-2.400	-1.200	2.683		
	1	-80.909	11.092	81.666	2.822	7.071
	4	78.509	-12.292	79.466	0.773	1.318
4		-7.600	-1.600	7.767		
	2	-44.610	2.220	44.665	1.073	-0.731
	3	-77.737	13.611	78.919	0.773	1.318
	6	69.337	-25.977	74.044	0.609	1.157
	12	45.410	8.546	46.207	-0.000	4.453	0.932
5		-94.200	19.920	96.283		
	2	-79.798	11.162	80.575	2.913	7.702
	7	-14.402	8.758	16.856	0.132	-1.822
6		0.000	0.000	0.000		
	2	-59.590	10.503	60.509	2.010	2.021
	4	-68.728	27.134	73.890	0.609	1.157
	7	37.688	-0.399	37.690	0.354	-0.718
	8	29.609	-20.402	35.957	0.143	-0.470
	9	26.896	-4.458	27.263	0.000	1.380	0.978
	10	15.459	0.675	15.473	0.000	1.166	0.969
	28	18.667	-13.052	22.778	0.061	-13.725
7		-22.800	-10.900	25.272		
	5	14.534	-10.581	17.978	0.132	-1.822
	6	-37.334	-0.319	37.336	0.354	-0.718
8		-30.000	19.977	36.043		
	6	-29.466	19.933	35.575	0.143	-0.470
	28	-0.534	0.044	0.536	0.003	-4.600
9		0.000	0.000	0.000		
	6	-26.896	5.838	27.522	0.000	1.380
	11	0.000	-6.880	6.880	-0.000	0.086
	10	26.896	1.042	26.916	0.000	0.698
10		-5.800	17.000	17.962		

	6	-15.459	0.492	15.466	0.000	1.166
	9	-26.896	-0.344	26.898	0.000	0.698
	20	8.655	2.046	8.894	0.065	0.145
	17	4.902	1.761	5.209	0.008	0.020
	21	15.542	9.061	17.991	0.099	0.213
	22	7.455	3.984	8.453	0.046	0.094
11		0.000	6.966	6.966		
	9	-0.000	6.966	6.966	-0.000	0.086
12		-11.200	-7.500	13.479		
	4	-45.410	-4.093	45.594	-0.000	4.453
	13	0.000	-21.707	21.707	0.000	0.552
	14	8.100	2.991	8.635	0.077	0.160
	15	18.403	9.234	20.590	0.235	0.463
	16	7.706	6.075	9.813	0.076	0.160
13		0.000	22.259	22.259		
	12	-0.000	22.259	22.259	0.000	0.552
14		-6.200	-1.600	6.403		
	12	-8.023	-2.831	8.508	0.077	0.160
	15	1.823	1.231	2.200	0.009	0.008
15		-8.200	-2.500	8.573		
	12	-18.168	-8.771	20.175	0.235	0.463
	14	-1.814	-1.223	2.188	0.009	0.008
	18	6.379	3.239	7.154	0.048	0.097
	23	5.404	4.255	6.878	0.041	0.083
16		-3.500	-1.800	3.936		
	12	-7.630	-5.915	9.654	0.076	0.160
	17	4.130	4.115	5.830	0.024	0.057
17		-9.000	-5.800	10.707		
	16	-4.106	-4.059	5.773	0.024	0.057
	10	-4.894	-1.741	5.195	0.008	0.020
18		-3.200	-0.900	3.324		
	15	-6.331	-3.142	7.068	0.048	0.097
	19	3.131	2.242	3.851	0.008	0.017
19		-9.500	-3.400	10.090		
	18	-3.123	-2.225	3.834	0.008	0.017
	20	-6.377	-1.175	6.485	0.013	0.026

20		-2.200	-0.700	2.309		
	19	6.390	1.201	6.502	0.013	0.026
	10	-8.590	-1.901	8.798	0.065	0.145
21		-17.500	-11.200	20.777		
	10	-15.444	-8.849	17.799	0.099	0.213
	22	-2.056	-2.351	3.124	0.001	0.002
22		0.000	0.000	0.000		
	10	-7.410	-3.890	8.369	0.046	0.094
	21	2.057	2.353	3.126	0.001	0.002
	24	5.352	1.537	5.568	0.032	0.050
23		-3.200	-1.600	3.578		
	15	-5.362	-4.172	6.794	0.041	0.083
	24	2.162	2.572	3.360	0.013	0.027
24		-8.700	-2.400	9.025		
	22	-5.320	-1.487	5.524	0.032	0.050
	23	-2.149	-2.545	3.331	0.013	0.027
	25	-1.231	1.632	2.044	0.007	0.012
25		0.000	0.000	0.000		
	24	1.238	-1.619	2.038	0.007	0.012
	26	3.542	2.363	4.258	0.042	0.063
	27	-4.780	-0.744	4.837	0.023	0.045
26		-3.500	-2.300	4.188		
	25	-3.500	-2.300	4.188	0.042	0.063
27		0.000	0.000	0.000		
	25	4.803	0.788	4.868	0.023	0.045
	28	-18.069	-4.089	18.526	0.000	1.227
	29	6.183	1.655	6.401	0.081	0.154
	30	7.083	1.646	7.272	0.153	0.288
28		0.000	0.000	0.000		
	27	18.069	5.316	18.835	0.000	1.227	0.968
	8	0.537	-4.644	4.675	0.003	-4.600
	6	-18.606	-0.672	18.618	0.061	-13.725
29		-2.400	-0.900	2.563		
	27	-6.102	-1.501	6.284	0.081	0.154
	30	3.702	0.601	3.750	0.032	0.060
30		-10.600	-1.900	10.769		

27	-6.930	-1.358	7.062	0.153	0.288
29	-3.670	-0.542	3.710	0.032	0.060

Total loss 17.631 19.903

Power Flow Solution by Newton-Raphson Method

Maximum Power Mismatch = 3.50329e-08

No. of Iterations = 10

Bus No.	Voltage Mag.	Angle Degree	------Load------ MW	Mvar	---Generation--- MW	Mvar	Injected Mvar
1	1.060	0.000	0.000	0.000	261.031	-43.998	0.000
2	1.053	-5.607	21.700	12.700	40.000	48.679	0.000
3	1.038	-8.145	2.400	1.200	0.000	0.000	0.000
4	1.033	-9.820	7.600	1.600	0.000	0.000	0.000
5	1.030	-14.333	94.200	19.000	0.000	38.920	0.000
6	1.035	-11.521	0.000	0.000	0.000	0.000	0.000
7	1.026	-13.182	22.800	10.900	0.000	0.000	0.000
8	1.040	-12.310	30.000	30.000	0.000	49.977	0.000
9	1.069	-14.356	0.000	0.000	0.000	0.000	0.000
10	1.068	-15.842	5.800	2.000	0.000	0.000	19.000
11	1.082	-14.356	0.000	0.000	0.000	6.966	0.000
12	1.093	-15.327	11.200	7.500	0.000	0.000	0.000
13	1.121	-15.327	0.000	0.000	0.000	22.259	0.000
14	1.077	-16.157	6.200	1.600	0.000	0.000	0.000
15	1.071	-16.203	8.200	2.500	0.000	0.000	0.000
16	1.076	-15.794	3.500	1.800	0.000	0.000	0.000
17	1.065	-16.021	9.000	5.800	0.000	0.000	0.000
18	1.058	-16.731	3.200	0.900	0.000	0.000	0.000
19	1.054	-16.866	9.500	3.400	0.000	0.000	0.000
20	1.056	-16.663	2.200	0.700	0.000	0.000	0.000
21	1.057	-16.273	17.500	11.200	0.000	0.000	0.000
22	1.057	-16.262	0.000	0.000	0.000	0.000	0.000
23	1.058	-16.539	3.200	1.600	0.000	0.000	0.000
24	1.049	-16.665	8.700	6.700	0.000	0.000	4.300
25	1.046	-16.293	0.000	0.000	0.000	0.000	0.000
26	1.029	-16.690	3.500	2.300	0.000	0.000	0.000
27	1.052	-15.816	0.000	0.000	0.000	0.000	0.000
28	1.036	-12.174	0.000	0.000	0.000	0.000	0.000
29	1.033	-16.978	2.400	0.900	0.000	0.000	0.000
30	1.022	-17.810	10.600	1.900	0.000	0.000	0.000
Total			283.400	126.200	301.031	122.803	23.300

ASSIGHNMENT NO. # 02

(25/01/23 & 01/02/2023)

NAME: SHIVRAJ VISHWAKARMA **ROLL. NO.:** 224102112

OBJECTIVE:

To determine the line loadability and voltage stability using MATLAB.

- **SOFTWARE:** MATLAB R2022a
- **REFERECE PROBLEM:** Obtain the power flow solution by the Newton-Raphson method for the system of given fig:

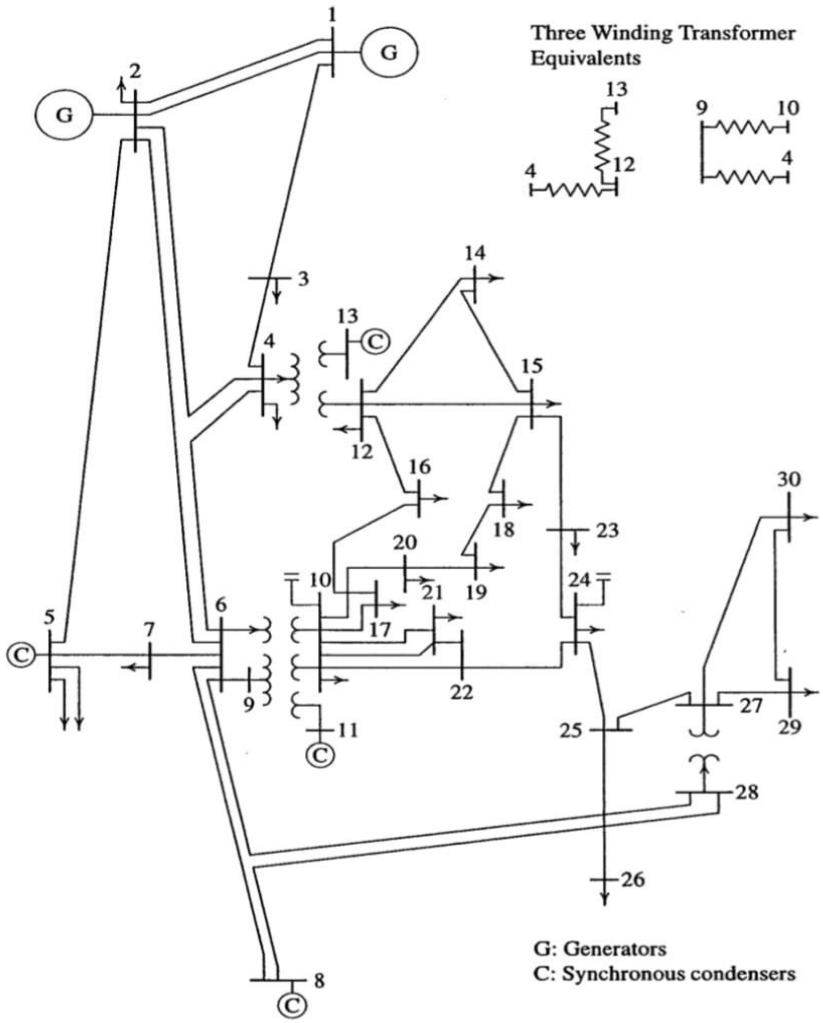


FIGURE 6.16
30-bus IEEE sample system.

➢ DATAFILE:

The datafile for the above system prepared in MATLAB:

```
clear
basemva = 100;  accuracy = 0.00001; maxiter = 100;

%        IEEE 30-BUS TEST SYSTEM (American Electric Power)
%        Bus Bus  Voltage Angle   ---Load---- -------Generator----- Injected
%        No  code Mag.    Degree  MW    Mvar  MW  Mvar Qmin Qmax     Mvar
busdata=[1   1    1.06    0.0     0.0   0.0    0.0  0.0   0   0       0
         2   2    1.043   0.0   21.70  12.7   40.0  0.0 -40  50       0
         3   0    1.0     0.0     2.4   1.2    0.0  0.0   0   0       0
         4   0    1.06    0.0     7.6   1.6    0.0  0.0   0   0       0
         5   2    1.01    0.0    94.2  19.0    0.0  0.0 -40  40       0
         6   0    1.0     0.0     0.0   0.0    0.0  0.0   0   0       0
         7   0    1.0     0.0    22.8  10.9    0.0  0.0   0   0       0
         8   2    1.01    0.0    30.0  30.0    0.0  0.0 -10  60       0
         9   0    1.0     0.0     0.0   0.0    0.0  0.0   0   0       0
        10   0    1.0     0.0     5.8   2.0    0.0  0.0  -6  24      19
        11   2    1.082   0.0     0.0   0.0    0.0  0.0   0   0       0
        12   0    1.0     0       11.2  7.5    0    0     0   0       0
        13   2    1.071   0        0    0.0    0    0    -6  24       0
        14   0    1       0       6.2   1.6    0    0     0   0       0
        15   0    1       0       8.2   2.5    0    0     0   0       0
        16   0    1       0       3.5   1.8    0    0     0   0       0
        17   0    1       0       9.0   5.8    0    0     0   0       0
        18   0    1       0       3.2   0.9    0    0     0   0       0
        19   0    1       0       9.5   3.4    0    0     0   0       0
        20   0    1       0       2.2   0.7    0    0     0   0       0
        21   0    1       0      17.5  11.2    0    0     0   0       0
        22   0    1       0       0     0.0    0    0     0   0       0
        23   0    1       0       3.2   1.6    0    0     0   0       0
        24   0    1       0       8.7   6.7    0    0     0   0      4.3
        25   0    1       0       0     0.0    0    0     0   0       0
        26   0    1       0       3.5   2.3    0    0     0   0       0
        27   0    1       0       0     0.0    0    0     0   0       0
        28   0    1       0       0     0.0    0    0     0   0       0
        29   0    1       0       2.4   0.9    0    0     0   0       0
        30   0    1       0      10.6   1.9    0    0     0   0       0];
%                                          Line code
%         Bus bus   R      X     1/2 B    = 1 for lines
%         nl  nr  p.u.   p.u.   p.u.      > 1 or < 1 tr. tap at bus nl
linedata=[1   2   0.0192   0.0575   0.02640    1
          1   3   0.0452   0.1852   0.02040    1
          2   4   0.0570   0.1737   0.01840    1
          3   4   0.0132   0.0379   0.00420    1
          2   5   0.0472   0.1983   0.02090    1
          2   6   0.0581   0.1763   0.01870    1
          4   6   0.0119   0.0414   0.00450    1
          5   7   0.0460   0.1160   0.01020    1
          6   7   0.0267   0.0820   0.00850    1
          6   8   0.0120   0.0420   0.00450    1
          6   9   0.0      0.2080   0.0    0.978
          6  10   0         .5560   0      0.969
```

```
 9  11   0         .2080   0          1
 9  10   0         .1100   0          1
 4  12   0         .2560   0      0.932
12  13   0         .1400   0          1
12  14    .1231    .2559   0          1
12  15    .0662    .1304   0          1
12  16    .0945    .1987   0          1
14  15    .2210    .1997   0          1
16  17    .0824    .1923   0          1
15  18    .1073    .2185   0          1
18  19    .0639    .1292   0          1
19  20    .0340    .0680   0          1
10  20    .0936    .2090   0          1
10  17    .0324    .0845   0          1
10  21    .0348    .0749   0          1
10  22    .0727    .1499   0          1
21  22    .0116    .0236   0          1
15  23    .1000    .2020   0          1
22  24    .1150    .1790   0          1
23  24    .1320    .2700   0          1
24  25    .1885    .3292   0          1
25  26    .2544    .3800   0          1
25  27    .1093    .2087   0          1
28  27     0       .3960   0      0.968
27  29    .2198    .4153   0          1
27  30    .3202    .6027   0          1
29  30    .2399    .4533   0          1
 8  28    .0636    .2000   0.0214     1
 6  28    .0169    .0599   0.065      1];
```

➢ THEORETICAL ANALYSIS

LINE LOADABILITY AND VOLTAGE STABILITY:

Loadability of line is defined as the extent of load which can flow through the line without exceeding the limitations. Line loadability is expressed in percentage of Surge Impedance Loading of line. The limiting factor for line loading are:

- ✓ thermal limit,
- ✓ voltage drop limit and
- ✓ steady state stability.

Fig. 1 shows the loadability of transmission line in terms of their Surge Impedance Loading (SIL).

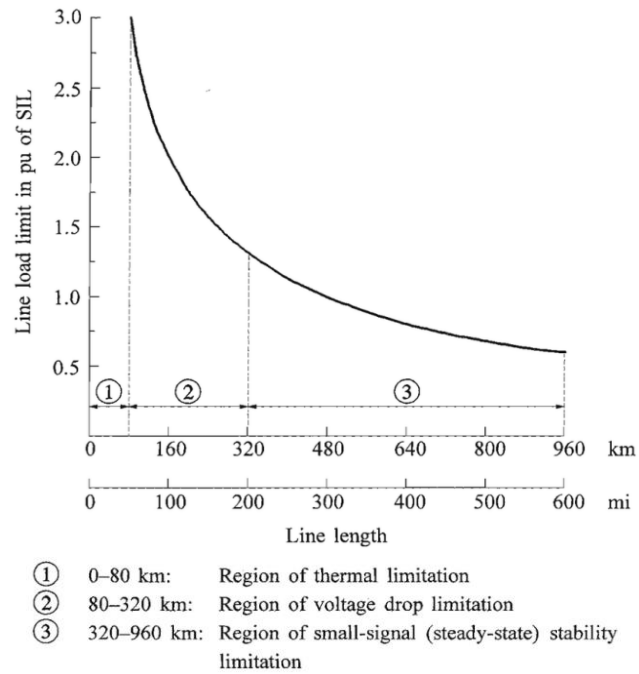


Fig. 1. Transmission line loadability curve

- **Thermal limit** corresponds to heating of transmission line and interposes a limit on current which can flow. Though of overhead transmission line, thermal limit is quite high but this cannot be neglected while deciding loading.
- **Voltage drop limit** corresponds to drop in receiving end voltage. In case the loading of line is very high compared to SIL, there will be a huge drop in receiving end voltage. A voltage drop of 5% is normally permissible.

To illustrate the voltage stability limit, a simple system is shown in Fig. 2.

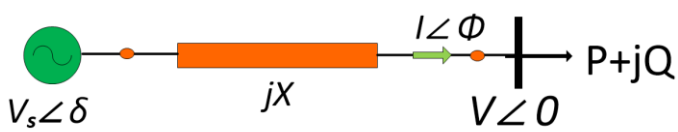


Fig. 2. System diagram.

At the receiving end,

$$V_R I^* = P + jQ$$

$$I = \frac{P - jQ}{V_R}$$

The sending end voltage is:

$$V_S = V_R + \frac{P - jQ}{V_R} jX$$

The corresponding magnitude equation is P_{max}

$$V_S{}^2 = \left\{ V_R + \frac{QX}{V_R} \right\}^2 + \left\{ \frac{PX}{V_R} \right\}^2$$

Thus, the power delivered to the load as a function of receiving end voltage when Q = 0 can be solved as:

$$P = \frac{\sqrt{V_S{}^2 - V_R{}^2}}{X} V_R$$

Since V_S is constant and close to 1 per-unit, and X cannot change, therefore V_R is the only variable that can vary. So, the power will vary with V_R, which is shown in Fig. 3.

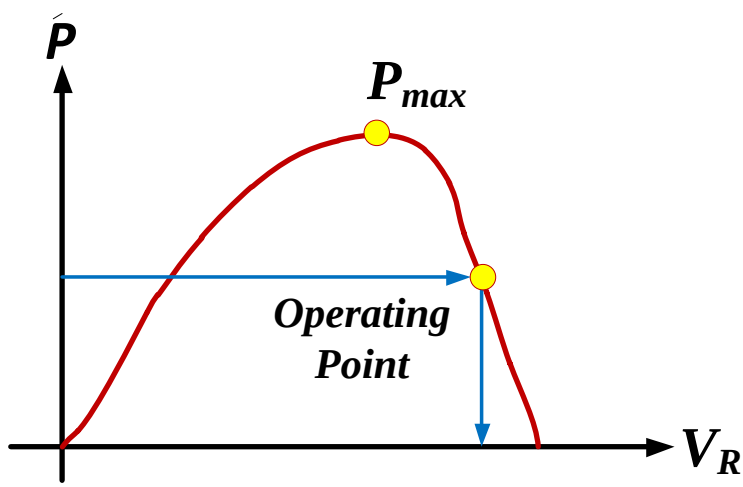


Fig. 3. Transfer capability curve.

The maximum power that can be transmitted is reached when

$$\frac{dP}{dV_R} = 0$$

Which can be given as,

$$P_{max_Vstab} = \frac{dP}{dV_R}$$

The above limit is called the Voltage Stability Limit of power transmission.

Experiment Flowchart:

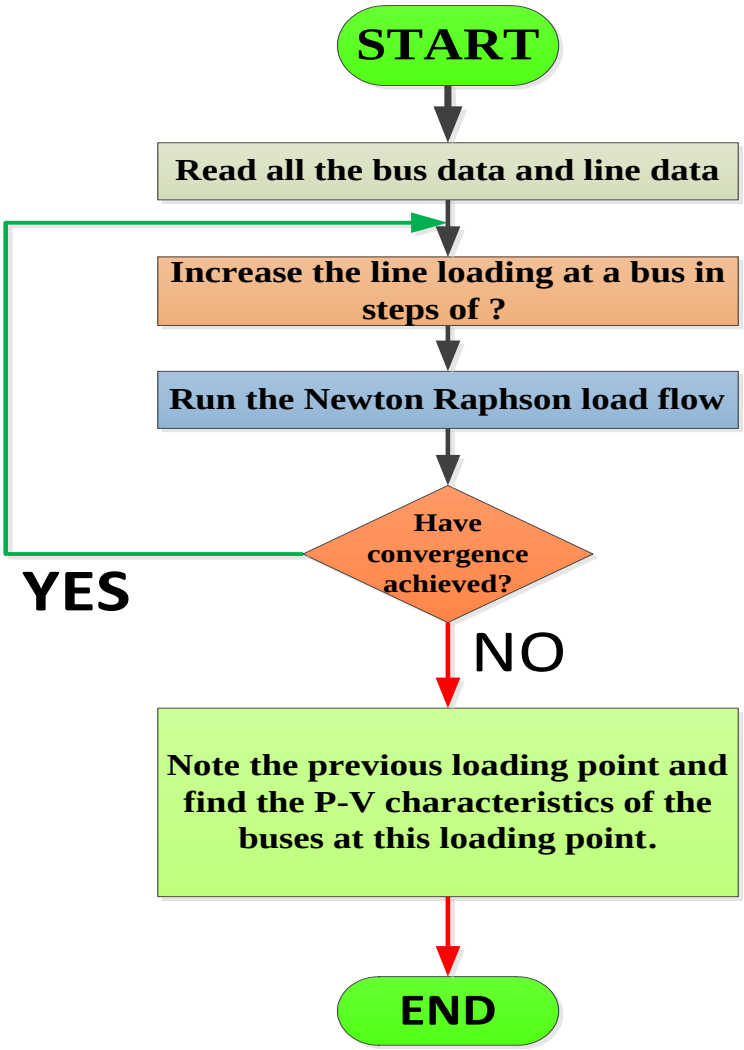


Fig. 4. Flowchart exhibiting the process of determining the loadability and voltage stability of the buses

➢ FIRST CASE: BY INCREASING POWER OF ALL BUSES

ALGORITHM:

1. First, we have to call Datafile of the given 30 bus standard system.
2. After defining appropriate variables we'll use a **while loop** for the case when NRLF is converging.
3. In that loop first we increase the power of each bus by %5 of its previous value for the entire system using a **for loop** & then we'll use **LFYBUS & LFNEWTON** subroutines for the Load flow solution.
4. After NRLF subroutine for all the cases where it is converging, we plot P with respect to V by solid lines it will give the **P vs V plot** for each bus.

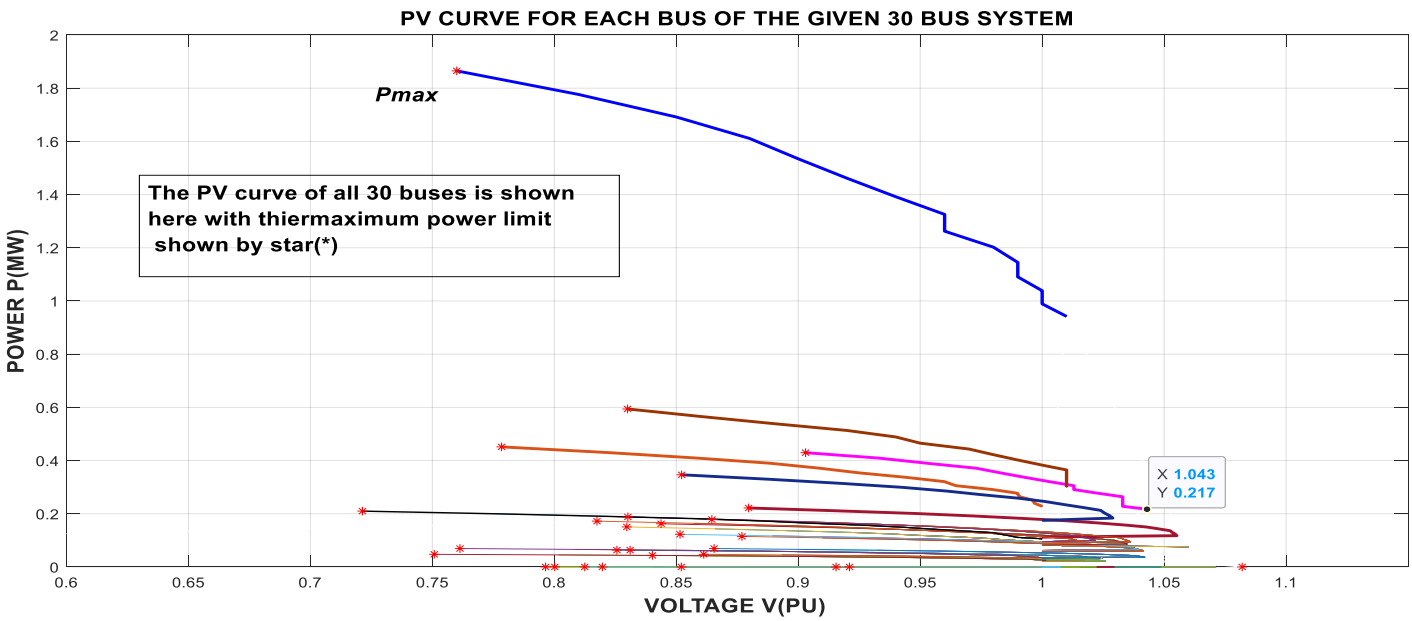
5. As our aim is to find the maximum line loadability & Voltage stability limit so we'll plot all the values of P vs V of the previous iteration where NRLF solution is not converging because it represents the knee point of **Transfer capability curve**.
6. Now we when it became non convergent while loop automatically will be ended & now, we have to show that for how many times we have increase the power i. e. how many iterations have been taken to get the ***Pmax*** limit.
7. Finally, we'll display the (Pmax, V) value for each bus at the end of the program.

➢ MATLAB CODES

CASE 1:

```
% CASE 1: By increasing the power of all the buses by 5% then plotting their
% Respective PV curve to get maximum power limit.
      clear; clc;
      Datafile
      nbus = length(busdata(:,1));
      bc = busdata(:,2);
      Pb=100; % mva base
      converge = 1; q = 0;
       while converge ~= 0
          q=q+1;
          for u = 1:nbus
            if bc(u) ~= 1
               pdm(q,u) = busdata(u,5);
               vltg(q,u) = busdata(u,3);
               busdata(u,5) = 1.05*busdata(u,5);
            end
          end
          LFYBUS
          LFNEWTON
          Ppu=pdm./Pb;
          if converge ~= 0
            plot(vltg(:,:), Ppu(:,:))
            title('PV CURVE FOR EACH BUS OF THE GIVEN 30 BUS SYSTEM')
            xlabel('VOLTAGE V(PU)')
            ylabel('POWER P(MW)')
            hold on
          end
           if converge == 0
             plot(vltg(q-1,:), Ppu(q-1,:), 'r*')
             Vlim = vltg(q-1,:)
             Plim = pdm(q-1,:)
             M(:,1) = Vlim;
             M(:,2) = Plim;
          end
       end
       grid on
       fprintf('\n\t"Loop ran for %g times"',q);
       fprintf('"\n\n Voltage  vs Maximum Power Limit  at all load buses
      is:\n');
       M
```

➢ RESULT OBTAINED:



- ✓ "Loop ran for 16 times""
- ✓ Voltage vs Maximum Power Limit at all load buses is: –

M =

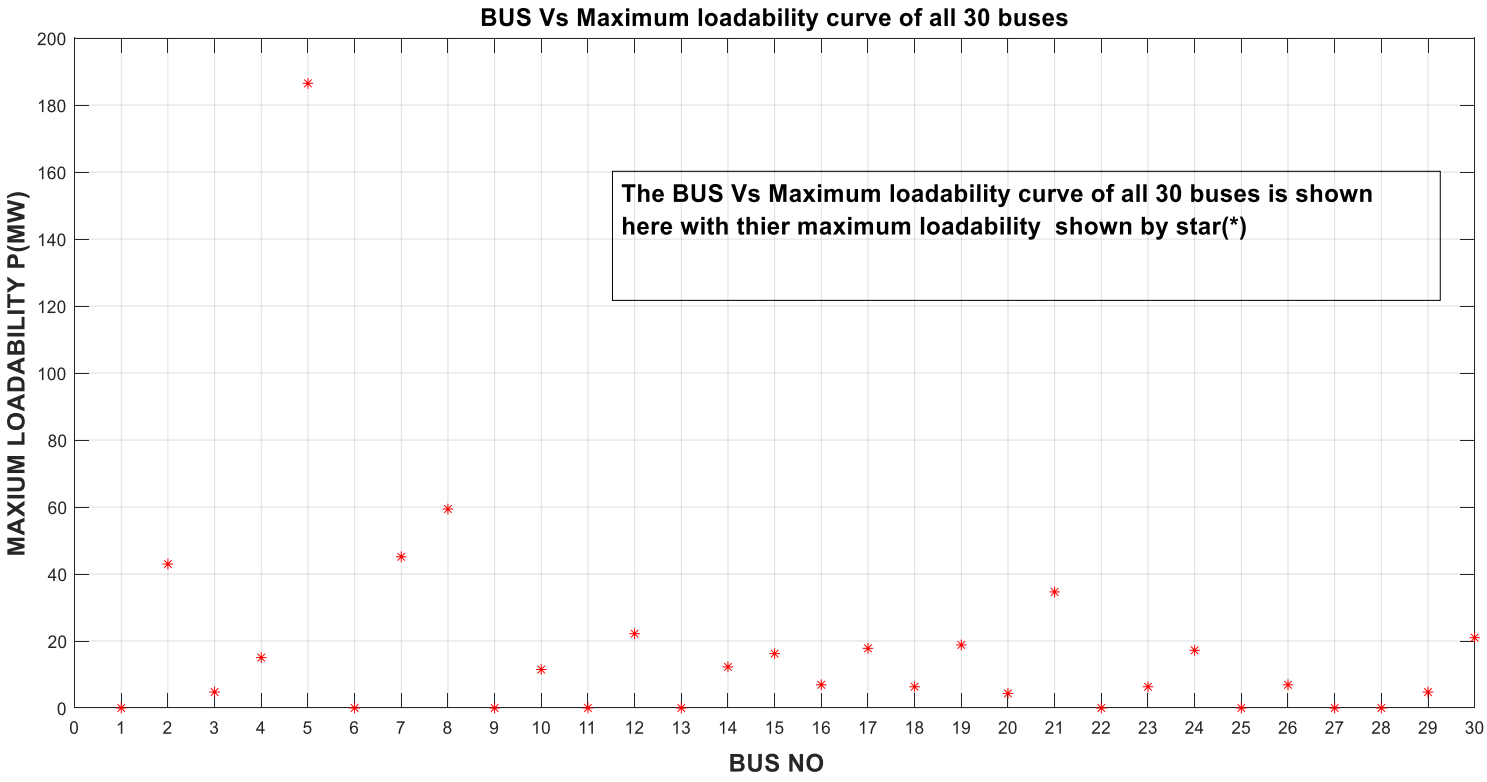
BUS NO	V	P
1.	0	0
2.	0.9030	42.9645
3.	0.8612	4.7518
4.	0.8297	15.0475
5.	0.7600	186.5096
6.	0.8197	0
7.	0.7783	45.1424
8.	0.8300	59.3979
9.	0.9155	0
10.	0.8769	11.4836
11.	1.0820	0
12.	0.8796	22.1752
13.	0.9210	0
14.	0.8515	12.2756
15.	0.8438	16.2354
16.	0.8655	6.9298
17.	0.8646	17.8194
18.	0.8311	6.3358
19.	0.8302	18.8094
20.	0.8402	4.3558
21.	0.8522	34.6488
22.	0.8521	0
23.	0.8257	6.3358

24.	0.8175	17.2254
25.	0.7964	0
26.	0.7613	6.9298
27.	0.8002	0
28.	0.8125	0
29.	0.7508	4.7518
30.	0.7213	20.9873

PART 1: BUS vs Maximum Power Limit

```
% CASE 1, PART1: By increasing the power of all the buses by 5% then
plotting
% the curve of BUS vs Maximum Power Limit at all buses.
clear; clc;
Datafile
nbus = length(busdata(:,1));
bc = busdata(:,2);
Pb=100; % mva base
converge = 1; q = 0;
 while converge ~= 0
    q=q+1;
    for u = 1:nbus
      if bc(u) ~= 1
         pdm(q,u) = busdata(u,5);
         vltg(q,u) = busdata(u,3);
         busdata(u,5) = 1.05*busdata(u,5);
      end
    end
    LFYBUS
    LFNEWTON
    Ppu=pdm./Pb;
for z=1:nbus
     if converge == 0
       plot(z, pdm(q-1,z), 'r*')
       title('MAXIMUM POWER LIMIT FOR EACH BUS FOR THE GIVEN 30 BUS
SYSTEM')
       xlabel('BUS NO')
       ylabel('MAXIUM LOADABILITY P(MW)')
       hold on
       bus =[1:nbus]';
       Pmax = pdm(q-1,:);
       M(:,1) = bus;
       M(:,2) = Pmax;
    end
  end
 end
 grid on
 fprintf('\n\t"Loop ran for %g times"',q);
 fprintf('"\n\n BUS vs Maximum Power Limit  at all  buses is:\n');
 M
```

➢ RESULT OBTAINED:



- ✓ "Loop ran for 16 times""
- ✓ BUS vs Maximum Power Limit at all buses is:

M =

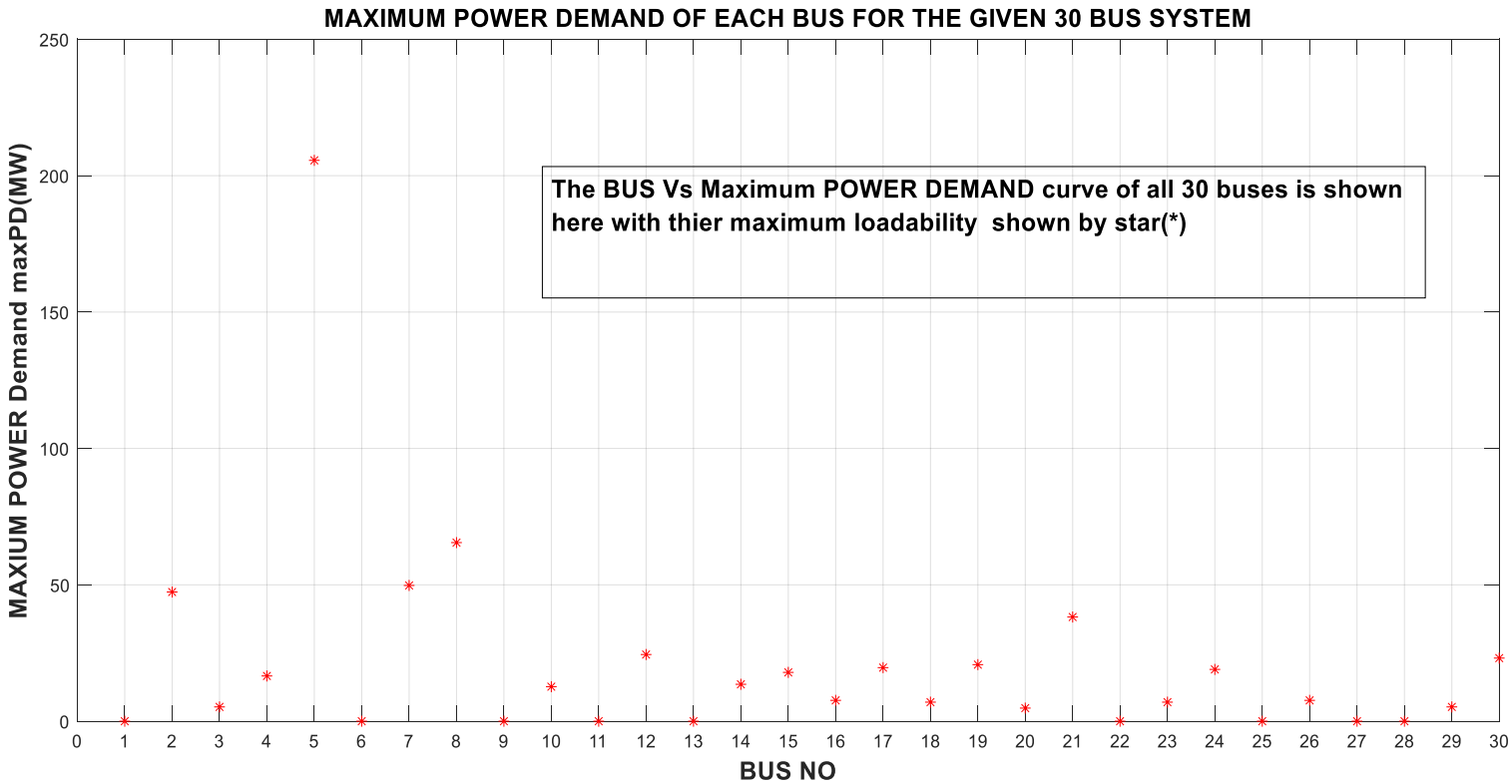
BUS NO.	Pmax
1.0000	0
2.0000	42.9645
3.0000	4.7518
4.0000	15.0475
5.0000	186.5096
6.0000	0
7.0000	45.1424
8.0000	59.3979
9.0000	0
10.0000	11.4836
11.0000	0
12.0000	22.1752
13.0000	0
14.0000	12.2756
15.0000	16.2354
16.0000	6.9298
17.0000	17.8194
18.0000	6.3358
19.0000	18.8094
20.0000	4.3558
21.0000	34.6488
22.0000	0

23.0000	6.3358
24.0000	17.2254
25.0000	0
26.0000	6.9298
27.0000	0
28.0000	0
29.0000	4.7518
30.0000	20.9873

PART 2: BUS vs Maximum Power Demand

```
% CASE 1, PART2: By increasing the power of all the buses by 5% then plotting
% the curve of BUS vs Maximum Power DEMAND at all buses.
    clear; clc;
    Datafile
    nbus = length(busdata(:,1));
    bc = busdata(:,2);
    Pb=100; % mva base
    converge = 1; q = 0;
     while converge ~= 0
        q=q+1;
        for u = 1:nbus
          if bc(u) ~= 1
             pdm(q,u) = busdata(u,5);
             vltg(q,u) = busdata(u,3);
             busdata(u,5) = 1.05*busdata(u,5);
          end
        end
        LFYBUS
        LFNEWTON
        Ppu=pdm./Pb;
      for z=1:nbus
       if converge == 0
         plot(z, Pd(z), 'r*')% Pd is the power demand at each bus calculated in LFNEWTON
         title('MAXIMUM POWER DEMAND OF EACH BUS FOR THE GIVEN 30 BUS SYSTEM')
         xlabel('BUS NO')
         ylabel('MAXIUM POWER Demand maxPD(MW)')
         hold on
         bus = [1:nbus]';
         maxPD = Pd;
         M(:,1) = bus;
         M(:,2) = Pd;
       end
     end
     end
     grid on
     fprintf('\n\t"Loop ran for %g times"',q);
     fprintf('"\n\n BUS  vs Maximum Power DEMAND  at all  buses is:\n');
     M
```

➢ RESULT OBTAINED:



- ✓ "Loop ran for 16 times""
- ✓ BUS vs Maximum Power Limit at all buses is:

M =

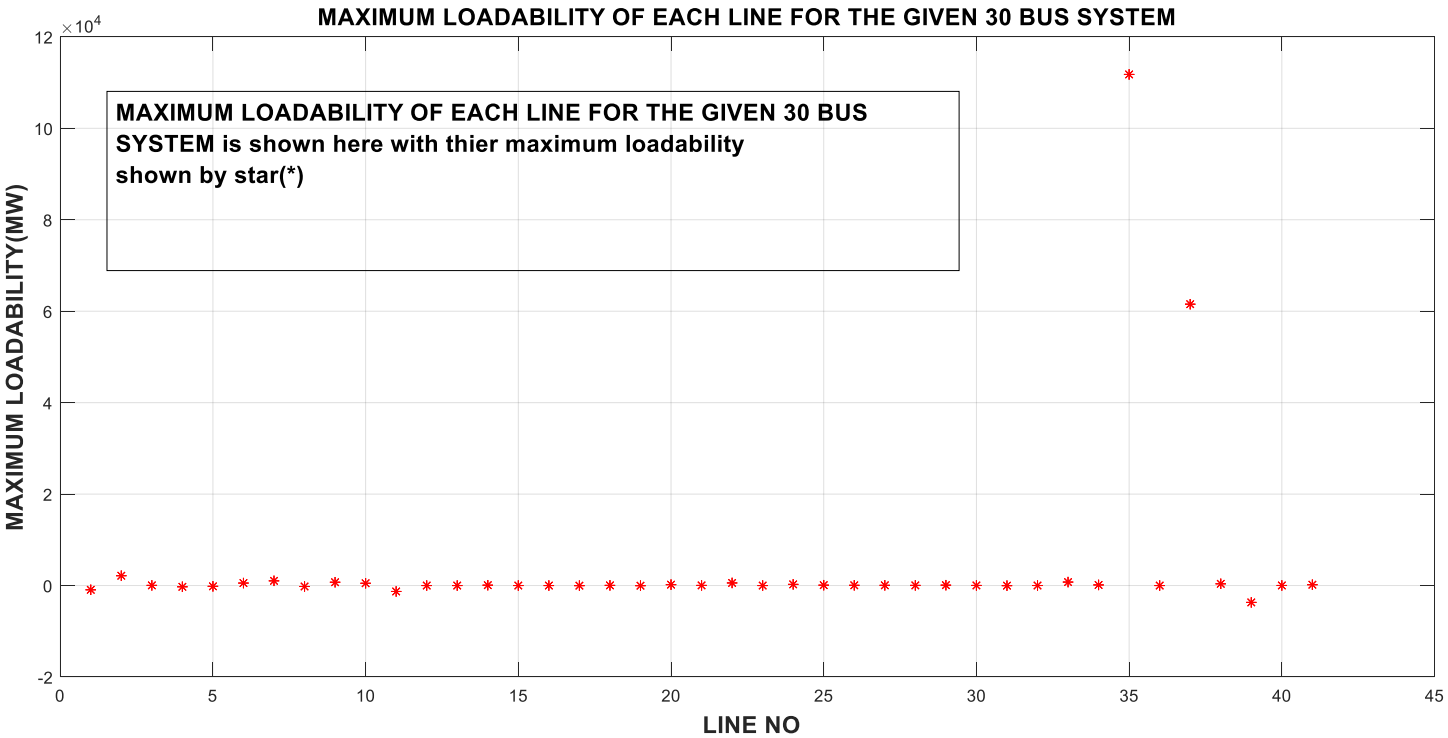
BUS NO.	Pmax
1.0000	0
2.0000	47.3684
3.0000	5.2389
4.0000	16.5898
5.0000	205.6268
6.0000	0
7.0000	49.7695
8.0000	65.4862
9.0000	0
10.0000	12.6607
11.0000	0
12.0000	24.4482
13.0000	0
14.0000	13.5338
15.0000	17.8996
16.0000	7.6401
17.0000	19.6459
18.0000	6.9852
19.0000	20.7373
20.0000	4.8023

21.0000	38.2003
22.0000	0
23.0000	6.9852
24.0000	18.9910
25.0000	0
26.0000	7.6401
27.0000	0
28.0000	0
29.0000	5.2389
30.0000	23.1385

PART 3: LINE vs Maximum LOADABILITY

```
% CASE 1, PART3: By increasing the power of all the buses by 5% then plotting
% the curve of LINE  vs Maximum Loadability of each line.%
    clear; clc;
    Datafile
    nbus = length(busdata(:,1));
    bc = busdata(:,2);
    Pb=100; % mva base
    converge = 1; q = 0;
     while converge ~= 0
        q=q+1;
        for u = 1:nbus
          if bc(u) ~= 1
             pdm(q,u) = busdata(u,5);
             vltg(q,u) = busdata(u,3);
             busdata(u,5) = 1.05*busdata(u,5);
          end
        end
        LFYBUS
        LFNEWTON
        LINEFLOW
        Ppu=pdm./Pb;
     for z=1:nbr
         if converge == 0
           plot(z, PLine(z), 'r*')
           % PLine is the vector having all line flows of all brances defined
           % in LINEFLOW
           title('MAXIMUM LOADABILITY OF EACH LINE FOR THE GIVEN 30 BUS SYSTEM')
           xlabel('LINE NO')
           ylabel('MAXIUM LOADABILITY maxPD(MW)')
           hold on
           line = [1:nbr]';
           M(:,1) = line;
           M(:,2) = PLine; %Line Loadability
        end
      end
     end
     grid on
     fprintf('\n\t"Loop ran for %g times"',q);
     fprintf('"\n\n LINE  vs Maximum Loadability of each line is:\n');
     M
```

➢ RESULT OBTAINED:



- ✓ "Loop ran for 16 times""
- ✓ BUS vs Maximum Power Limit at all buses is:

M =

1.0e+05 * [Line Loadability]

1. -0.0093
2. 0.0215
3. 0.0004
4. -0.0027
5. -0.0017
6. 0.0054
7. 0.0103
8. -0.0020
9. 0.0073
10. 0.0050
11. -0.0129
12. 0.0000
13. -0.0000
14. 0.0008
15. 0.0000
16. -0.0000
17. -0.0002
18. 0.0004
19. -0.0003
20. 0.0020

21. 0.0003
22. 0.0057
23. -0.0000
24. 0.0026
25. 0.0008
26. 0.0005
27. 0.0005
28. 0.0003
29. 0.0008
30. 0.0002
31. -0.0002
32. 0.0001
33. 0.0076
34. 0.0013
35. 1.1178
36. -0.0000
37. 0.6154
38. 0.0038
39. -0.0369
40. 0.0002
41. 0.0018

SECOND CASE: BY INCREASING POWER FOR ONLY ONE BUS AT A TIME

ALGORITHM:

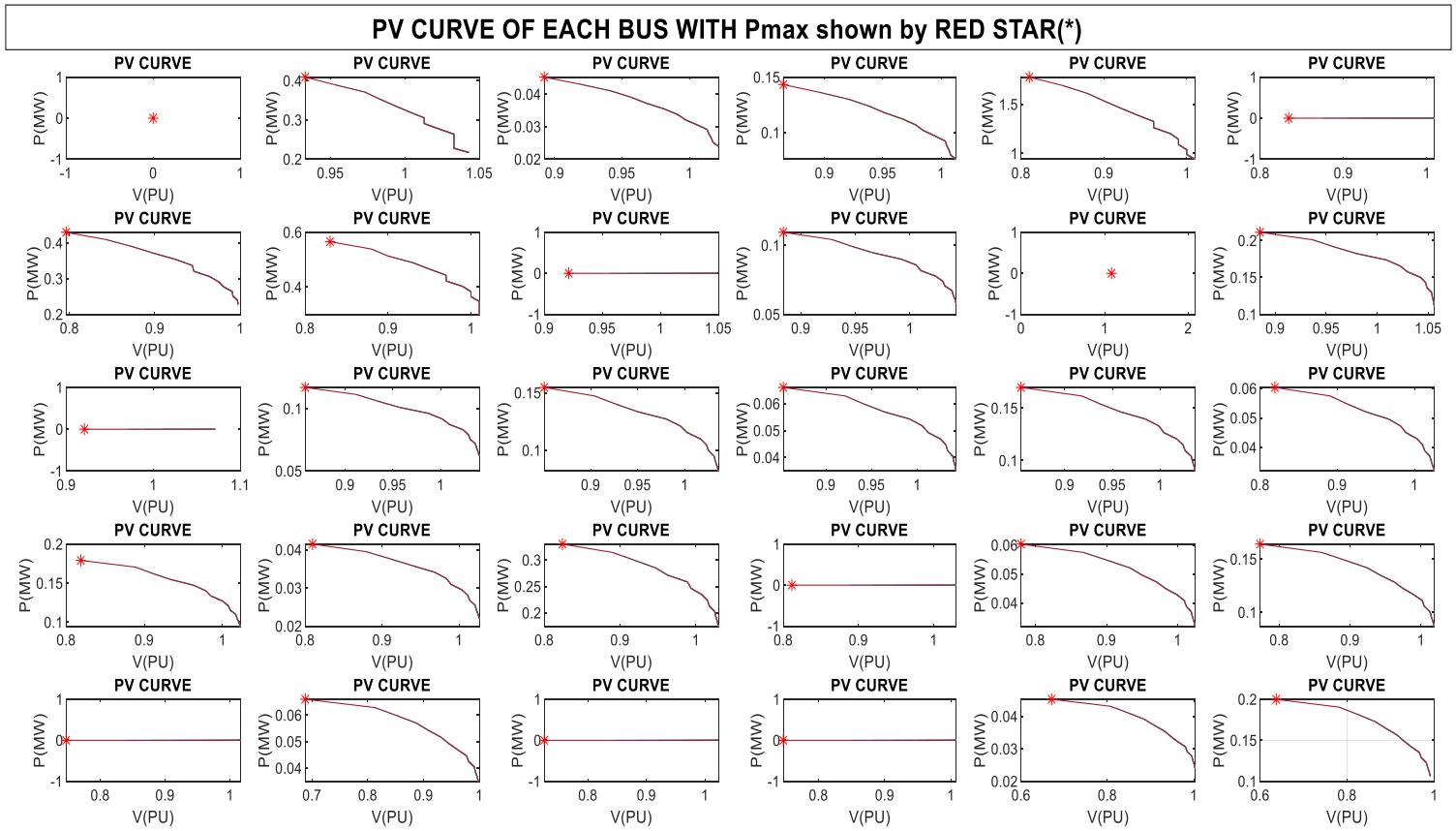
1. First, we have to call Datafile of the given 30 bus standard system.
2. After defining appropriate variables we'll use a **while loop** for the case when NRLF is converging.
3. In that loop first we increase the power of only one bus at a time by %5 of its previous value & then we'll use LFYBUS & LFNEWTON subroutines for the Load flow solution.
4. After NRLF subroutine for all the cases where it is converging, we plot P with respect to V by solid lines it will give the P vs V plot for each bus.
5. All the above things should be done inside a use a **for loop** to do the same thing for all the buses.
6. As our aim is to find the maximum line loadability & Voltage stability limit so we'll plot all the values of P vs V of just previous iteration where NRLF solution is not converging because it represents the knee point of **Transfer capability curve**.
7. Now we when it became non convergent while loop automatically will be ended & now, we have to show that for how many times we have increase the power i. e. how many iterations have been taken to get the ***Pmax*** limit.
8. Finally, we'll display the (Pmax, V) value for each bus at the end of the program.

➢ MATLAB CODES

➢ CASE 2:

```
% CASE 2: By increasing the power of all the buses by 5% then plotting thier
% respective PV curve to get maximum power limit.%
   clear; clc;
   Datafile
   nbus = length(busdata(:,1));
   bc = busdata(:,2);
   Pb=100; % mva base
   converge = 1; q = 0;
    while converge ~= 0
       q=q+1;
       for u = 1:nbus
         if bc(u) ~= 1
            pdm(q,u) = busdata(u,5);
            vltg(q,u) = busdata(u,3);
            busdata(u,5) = 1.05*busdata(u,5);
         end
         LFYBUS
         LFNEWTON
       end
       Ppu=pdm./Pb;
     for z=1:nbus
       if converge ~= 0
         subplot(5,6,z), plot(vltg(:,z), Ppu(:,z))
         title('PV CURVE')
         xlabel('V(PU)')
         ylabel('P(MW)')
         hold on
       end
        if converge == 0
          subplot(5,6,z), plot(vltg(q-1,z), Ppu(q-1,z), 'r*')
          Vlim = vltg(q-1,:)
          Plim = pdm(q-1,:)
          M(:,1) = Vlim;
          M(:,2) = Plim;
       end
     end
    end
    grid on
    fprintf('\n\t"Loop ran for %g times"',q);
    fprintf('"\n\n Voltage  vs Maximum Power Limit  at all load buses is:\n');
    M
```

➢ RESULT OBTAINED:



- ✓ "Loop ran for 15 times""
- ✓ Voltage vs Maximum Power Limit at all load buses is:

M =

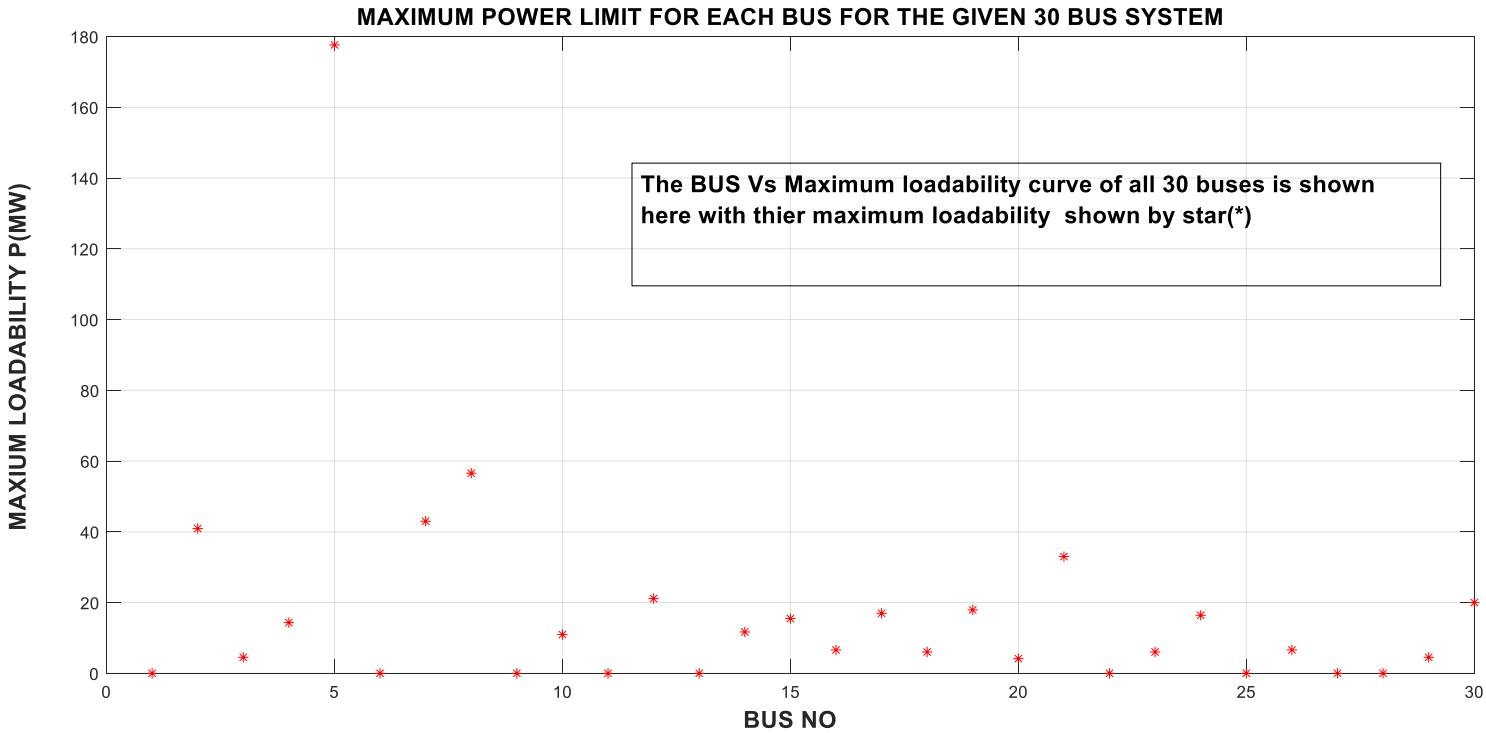
BUS NO	V	P
1.	0	0
2.	0.9330	40.9186
3.	0.8923	4.5256
4.	0.8647	14.3309
5.	0.8100	177.6281
6.	0.8343	0
7.	0.7961	42.9928
8.	0.8300	56.5695
9.	0.9211	0
10.	0.8840	10.9368
11.	1.0820	0
12.	0.8852	21.1193
13.	0.9210	0
14.	0.8582	11.6910
15.	0.8506	15.4623
16.	0.8532	6.5998
17.	0.8540	16.9708
18.	0.8192	6.0341
19.	0.8187	17.9137

20.	0.8096	4.1484
21.	0.8240	32.9989
22.	0.8114	0
23.	0.7797	6.0341
24.	0.7729	16.4051
25.	0.7474	0
26.	0.6877	6.5998
27.	0.7253	0
28.	0.7467	0
29.	0.6708	4.5256
30.	0.6371	19.9879

PART 1: BUS vs Maximum Power Limit

```
% CASE 2, PART1: By increasing the power of each bus first by 5% then plotting
% the curve of BUS vs Maximum Power Limit at all buses.
   Datafile
   nbus = length(busdata(:,1));
   bc = busdata(:,2);
   Pb=100; % mva base
   converge = 1; q = 0;
    while converge ~= 0
       q=q+1;
       for u = 1:nbus
         if bc(u) ~= 1
            pdm(q,u) = busdata(u,5);
            vltg(q,u) = busdata(u,3);
            busdata(u,5) = 1.05*busdata(u,5);
         end
         LFYBUS
         LFNEWTON
       end
       Ppu=pdm./Pb;
    for z=1:nbus
        if converge == 0
          plot(z, pdm(q-1,z), 'r*')
          title('MAXIMUM POWER LIMIT FOR EACH BUS FOR THE GIVEN 30 BUS SYSTEM')
          xlabel('BUS NO')
          ylabel('MAXIUM LOADABILITY P(MW)')
          hold on
          bus =[1:nbus]';
          Pmax = pdm(q-1,:);
          M(:,1) = bus;
          M(:,2) = Pmax;
       end
     end
    end
    grid on
    fprintf('\n\t"Loop ran for %g times"',q);
    fprintf('"\n\n BUS vs Maximum Power Limit  at all  buses is:\n');
    M
```

➢ RESULT OBTAINED:



- ✓ "Loop ran for 15 times""
- ✓ BUS vs Maximum Power Limit at all buses is:

M =

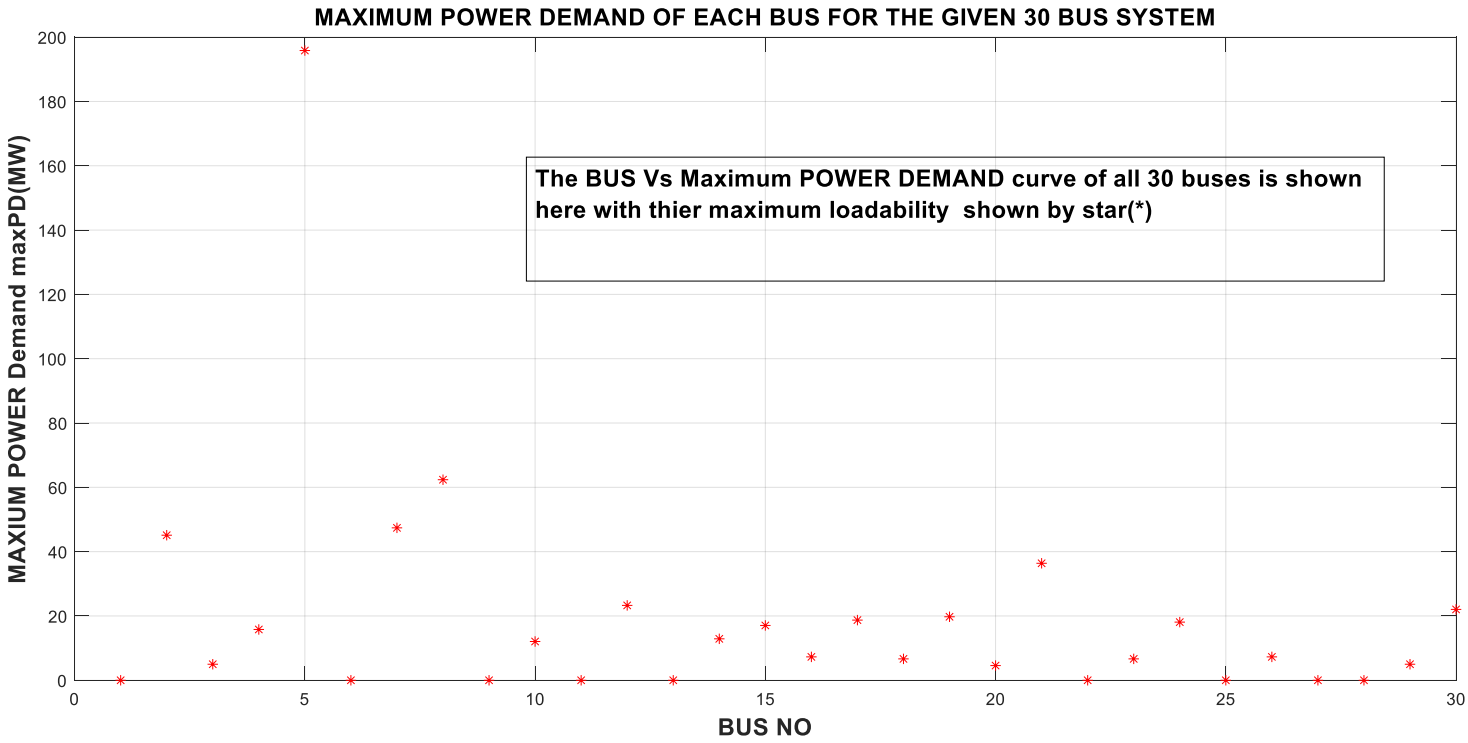
BUS NO.	Pmax
1.0000	0
2.0000	40.9186
3.0000	4.5256
4.0000	14.3309
5.0000	177.6281
6.0000	0
7.0000	42.9928
8.0000	56.5695
9.0000	0
10.0000	10.9368
11.0000	0
12.0000	21.1193
13.0000	0
14.0000	11.6910
15.0000	15.4623
16.0000	6.5998
17.0000	16.9708
18.0000	6.0341
19.0000	17.9137
20.0000	4.1484

21.0000	32.9989
22.0000	0
23.0000	6.0341
24.0000	16.4051
25.0000	0
26.0000	6.5998
27.0000	0
28.0000	0
29.0000	4.5256
30.0000	19.9879

PART 2: BUS vs Maximum Power Demand

```
% CASE 2, PART2: By increasing the power of each bus first by 5% then plotting
% the curve of BUS  vs Maximum Power DEMAND  at all  buses.%
   clear; clc;
   Datafile
   nbus = length(busdata(:,1));
   bc = busdata(:,2);
   Pb=100; % mva base
   converge = 1; q = 0;
    while converge ~= 0
       q=q+1;
       for u = 1:nbus
         if bc(u) ~= 1
            pdm(q,u) = busdata(u,5);
            vltg(q,u) = busdata(u,3);
            busdata(u,5) = 1.05*busdata(u,5);
         end
         LFYBUS
         LFNEWTON
       end
       Ppu=pdm./Pb;
     for z=1:nbus
        if converge == 0
          plot(z, Pd(z), 'r*')
          title('MAXIMUM POWER DEMAND OF EACH BUS FOR THE GIVEN 30 BUS SYSTEM')
          xlabel('BUS NO')
          ylabel('MAXIUM POWER Demand maxPD(MW)')
          hold on
          bus = [1:nbus]';
          maxPD = Pd;
          M(:,1) = bus;
          M(:,2) = Pd;
       end
     end
    end
    grid on
    fprintf('\n\t"Loop ran for %g times"',q);
    fprintf('"\n\n BUS  vs Maximum Power DEMAND  at all  buses is:\n');
    M
```

➢ RESULT OBTAINED:



- ✓ "Loop ran for 15 times""
- ✓ BUS vs Maximum Power DEMAND at all buses is:

M =

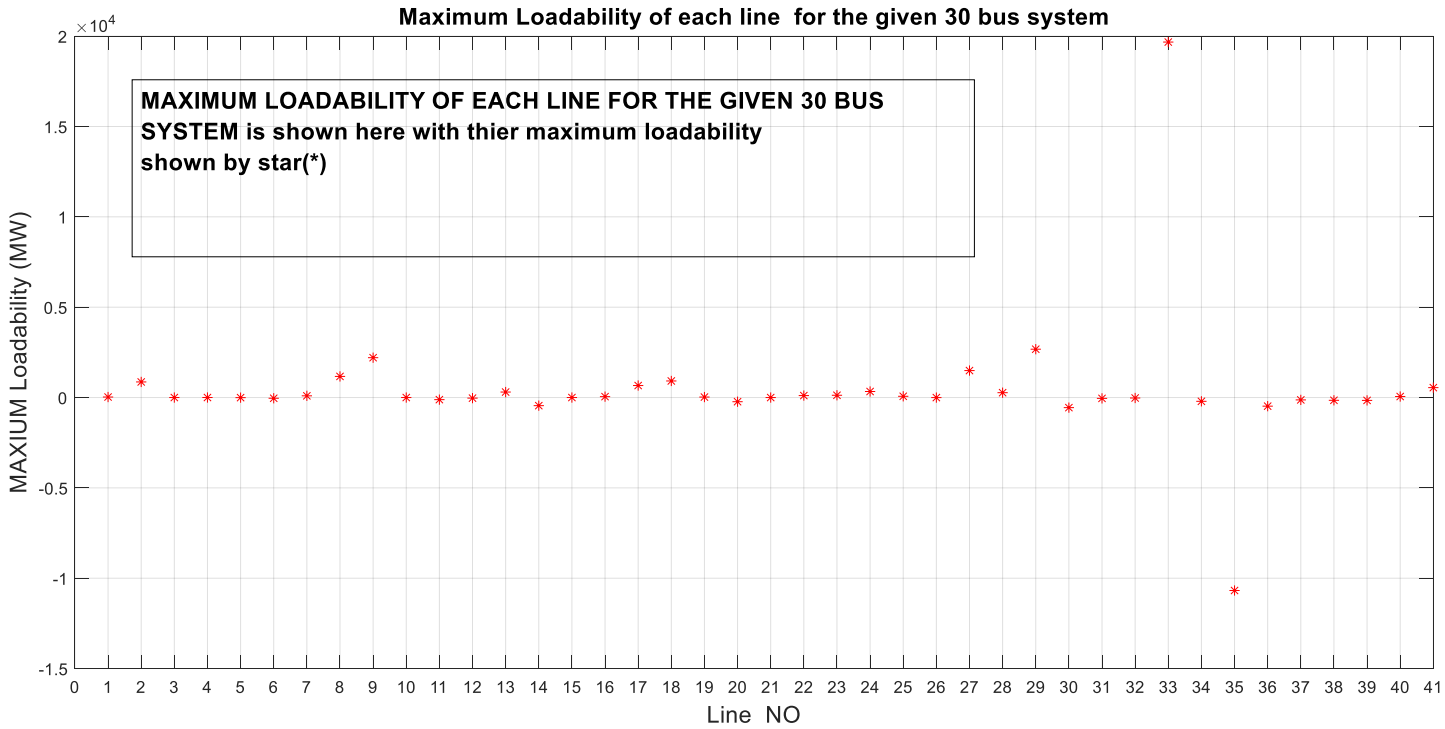
BUS NO.	Pmax
1.0000	0
2.0000	45.1127
3.0000	4.9894
4.0000	15.7999
5.0000	195.8350
6.0000	0
7.0000	47.3996
8.0000	62.3678
9.0000	0
10.0000	12.0578
11.0000	0
12.0000	23.2840
13.0000	0
14.0000	12.8894
15.0000	17.0472
16.0000	7.2762
17.0000	18.7104
18.0000	6.6526
19.0000	19.7498

20.0000	4.5736
21.0000	36.3812
22.0000	0
23.0000	6.6526
24.0000	18.0867
25.0000	0
26.0000	7.2762
27.0000	0
28.0000	0
29.0000	4.9894
30.0000	22.0366

PART 3: LINE vs Maximum LOADABILITY

```
% CASE 2, PART3: By increasing the power of each bus first by 5% then plotting
% the curve of LINE  vs Maximum Loadability of each line.%
    clear; clc;
    Datafile
    nbus = length(busdata(:,1));
    bc = busdata(:,2);
    Pb=100; % mva base
    converge = 1; q = 0;
     while converge ~= 0
        q=q+1;
        for u = 1:nbus
          if bc(u) ~= 1
             pdm(q,u) = busdata(u,5);
             vltg(q,u) = busdata(u,3);
             busdata(u,5) = 1.05*busdata(u,5);
          end
          LFYBUS
          LFNEWTON
          LINEFLOW
        end
        Ppu=pdm./Pb;
      for z=1:nbr
         if converge == 0
           plot(z, PLine(z), 'r*')
           title('Maximum Loadability of each line  for the given 30 bus system')
           xlabel('Line  NO')
           ylabel('MAXIUM Loadability (MW)')
           hold on
           line = [1:nbr]';
           M(:,1) = line;
           M(:,2) = PLine; %Line Loadability
        end
      end
     end
     grid on
     fprintf('\n\t"Loop ran for %g times"',q);
     fprintf('"\n\n LINE  vs Maximum Loadability of each line is:\n');
     M
```

➢ RESULT OBTAINED:



- ✓ "Loop ran for 15 times""
- ✓ Voltage vs Maximum Power Limit at all load buses is:

M =

1.0e+04 * [Line Loadability]

0.0001	0.0028
0.0002	0.0863
0.0003	-0.0000
0.0004	0.0000
0.0005	-0.0006
0.0006	-0.0040
0.0007	0.0095
0.0008	0.1169
0.0009	0.2205
0.0010	-0.0000
0.0011	-0.0117
0.0012	-0.0033
0.0013	0.0304
0.0014	-0.0449
0.0015	-0.0000
0.0016	0.0050
0.0017	0.0664
0.0018	0.0917
0.0019	0.0026

0.0020	-0.0231
0.0021	-0.0001
0.0022	0.0108
0.0023	0.0126
0.0024	0.0334
0.0025	0.0068
0.0026	-0.0003
0.0027	0.1492
0.0028	0.0269
0.0029	0.2675
0.0030	-0.0559
0.0031	-0.0046
0.0032	-0.0025
0.0033	1.9674
0.0034	-0.0209
0.0035	-1.0680
0.0036	-0.0477
0.0037	-0.0131
0.0038	-0.0153
0.0039	-0.0159
0.0040	0.0056
0.0041	0.0548

➢ CONCLUSION:

AS stated in the objective we have Obtained the maximum Power limit or steady state limit & the respective P vs V Curve, BUS vs Pmax curve, Bus vs max power Demand, LINE vs Line loadability are attached along with all the knee points (Pmax, V) for both the cases.

ASSIGHNMENT NO. # 03

(08/02/23 & 15/02/23)

NAME: SHIVRAJ VISHWAKARMA **ROLL. NO.:** 224102112

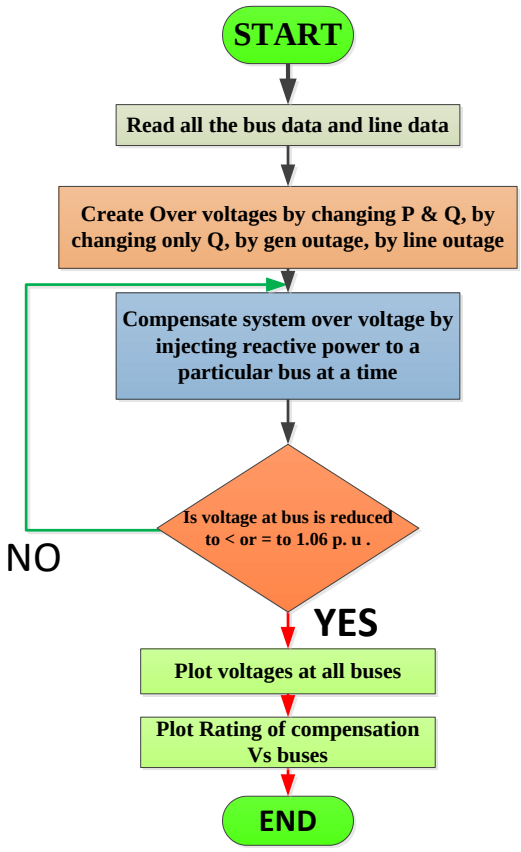
OBJECTIVE

To create overvoltage at all buses of IEEE 30-bus test system & to determine suitable rating of SVC for reactive power compensation at that specificbus using MATLAB.

- **SOFTWARE:** MATLAB R2022a
- **THEORETICAL ANALYSIS**

The main function of a SVC is to absorb or supply reactive power based on the changing VAR requirement of the load. With the reactive power compensation techniques, reactive power is controlled in such a way that the performance of electric power system such as transmission efficiency is improved.

Experiment Flowchart:



➢ FIRST CASE: By Varying P & Q Both to create over voltage at some buses:

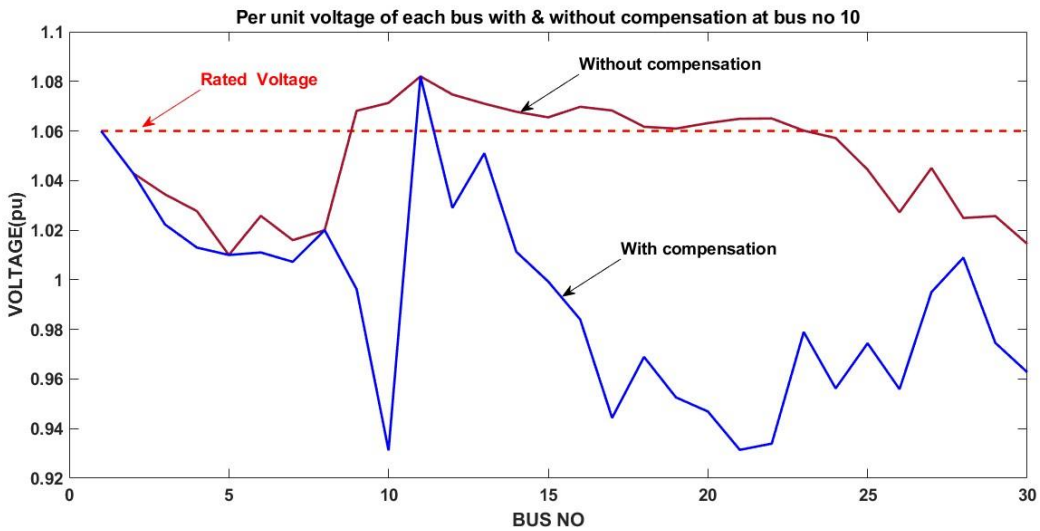
1. First, we have to call Datafile of the given 30 bus standard system.
2. After defining appropriate variables we'll reduce P & Q for some buses (say from 2 to 25) by some extent (say 50% to create more over voltages)
3. Now we'll compensate system over voltage by using compensation at all PQ buses having over voltage.
4. After that we'll plot the voltage with respect to buses for both the cases with compensation and without compensation.

➢ MATLAB CODES

```
% CASE 1: By varying P & Q BOTH to create overvoltage.
clear; clc;
Datafile
q=0;
nbus = length(busdata(:,1));
busdata(2:25, 5)= 0.5*busdata(2:25, 5);
busdata(2:25, 6)= 0.5*busdata(2:25, 6);
LFYBUS
LFNEWTON
vcc1=1.06*ones(1,30)
plot(vcc1,'y--')
hold on
plot(Vm)
title('Per unit voltage of each bus')
xlabel('BUS NO')
ylabel('VOLTAGE(pu)')
hold on

% By using compensation at bus no. 10
Qo=busdata(10, 6);
V10uc = busdata(10, 3);
while busdata(9, 3)> 1.06 || busdata(10, 3)> 1.06 || busdata(12, 3)> 1.06 ||
busdata(13, 3)> 1.06 || busdata(17, 3)> 1.06
    q = q+1;
busdata(10, 6) = 1.01*busdata(10, 6);
   LFYBUS
   LFNEWTON
end
   plot(Vm)
   title('Per unit voltage of each bus with & without compensation at bus no 10')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
   V10uc
   V10c=busdata(10, 3)
   Q10c = busdata(10, 6)- Qo
% grid on
 fprintf('\n\t"Loop ran for %g times"',q);
```

➢ RESULT OBTAINED:

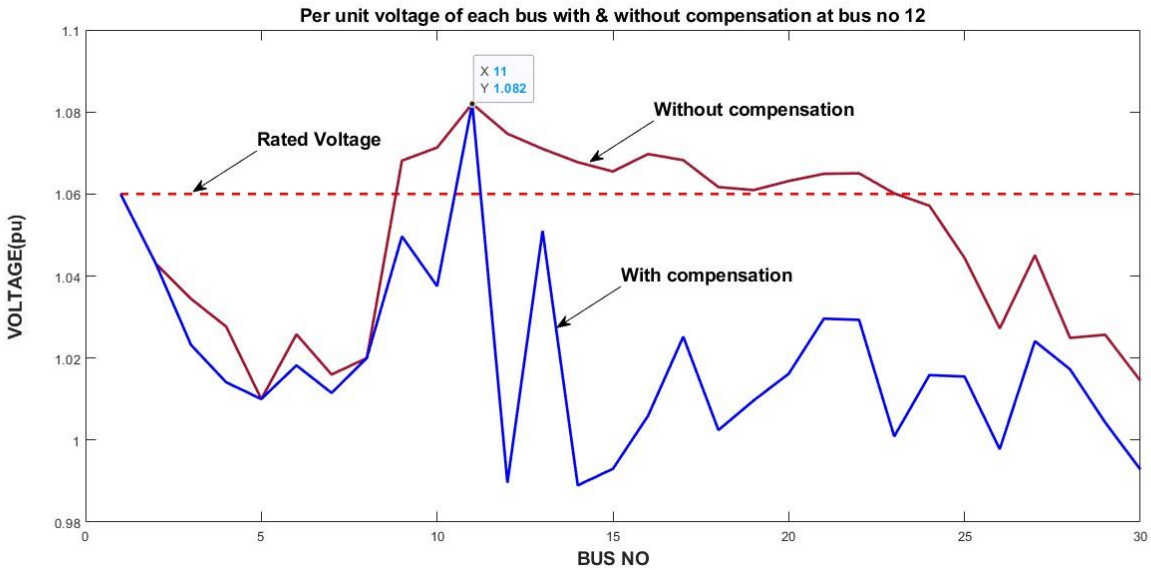
```
V10uc= 1.0713
V10c = 0.9312
Q10c = 121.2429
      "Loop ran for 483 times">>
```




```
% By using compensation at bus no. 12
Qo=busdata(12, 6);
V12uc = busdata(12, 3);
while busdata(9, 3)> 1.06 || busdata(10, 3)> 1.06 || busdata(12, 3)> 1.06 ||
busdata(13, 3)> 1.06 || busdata(17, 3)> 1.06
    q = q+1;
busdata(12, 6) = 1.01*busdata(12, 6);
   LFYBUS
   LFNEWTON
end
   plot(Vm)
   title('Per unit voltage of each bus with & without compensation at bus no 12')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
   V12uc
   V12c=busdata(12, 3)
   Q12c = busdata(12, 6)- Qo
% grid on
 fprintf('\n\t"Loop ran for %g times"',q);
```

➢ RESULT OBTAINED:

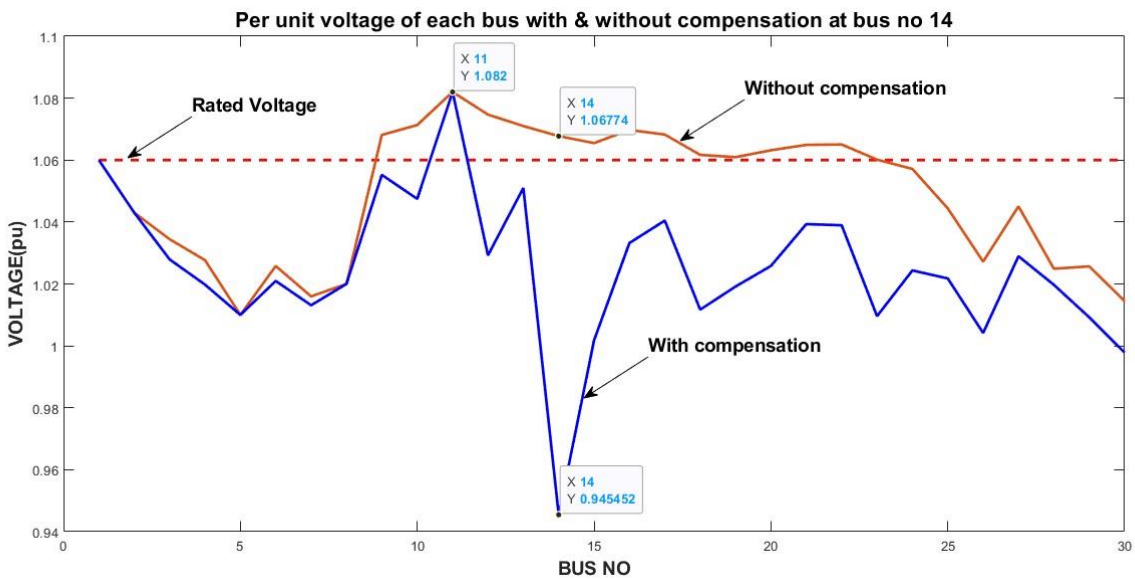
```
V12uc= 1.0747
V12c = 0.9896
Q12c = 101.371
      "Loop ran for 335 times">>
```




```
% By using compensation at bus no. 14
Qo=busdata(14, 6);
V14uc = busdata(14, 3);
while busdata(9, 3)> 1.06 || busdata(10, 3)> 1.06 || busdata(12, 3)> 1.06 ||
busdata(13, 3)> 1.06 || busdata(17, 3)> 1.06
    q = q+1;
busdata(14, 6) = 1.01*busdata(14, 6);
   LFYBUS
   LFNEWTON
end
   plot(Vm)
   title('Per unit voltage of each bus with & without compensation at bus no 14')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
   V14uc
   V14c=busdata(14, 3)
   Q14c = busdata(14, 6)- Qo
% grid on
 fprintf('\n\t"Loop ran for %g times"',q);
```

➢ RESULT OBTAINED:

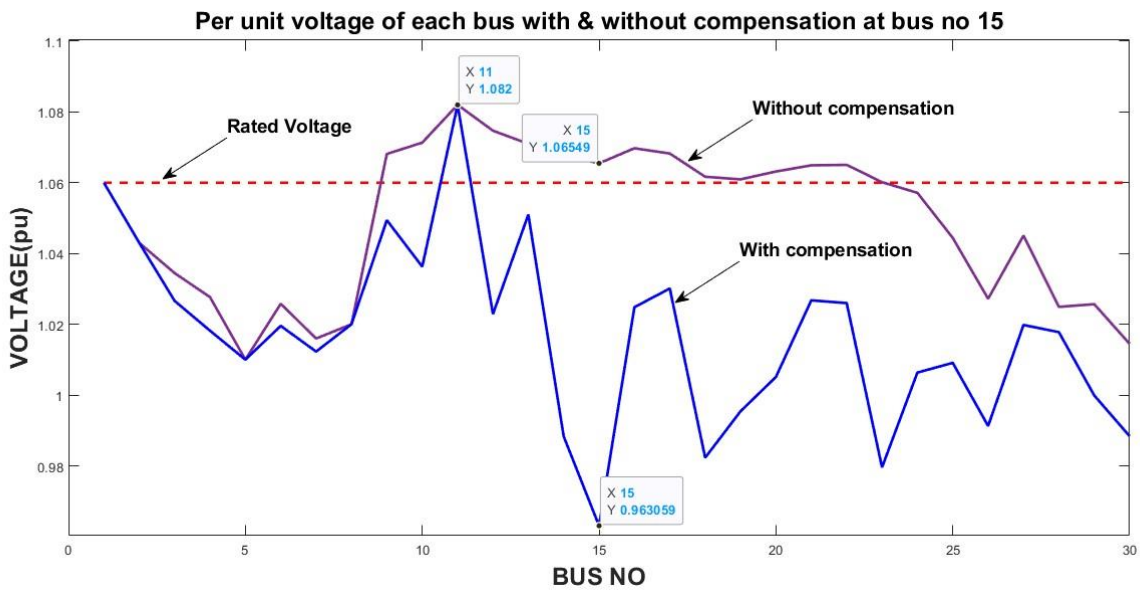
```
V14uc= 1.0677
V14c = 0.9455
Q14c = 52.4979

       "Loop ran for 422 times">>
```




```
% By using compensation at bus no. 15
Qo=busdata(15, 6);
V15uc = busdata(15, 3);
while busdata(9, 3)> 1.06 || busdata(10, 3)> 1.06 || busdata(12, 3)> 1.06 ||
busdata(13, 3)> 1.06 || busdata(17, 3)> 1.06
    q = q+1;
busdata(15, 6) = 1.01*busdata(15, 6);
   LFYBUS
   LFNEWTON
end
   plot(Vm)
   title('Per unit voltage of each bus with & without compensation at bus no 15')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
   V15uc
   V15c = busdata(15, 3)
   Q15c = busdata(15, 6)- Qo
% grid on
 fprintf('\n\t"Loop ran for %g times"',q);
```

➢ RESULT OBTAINED:

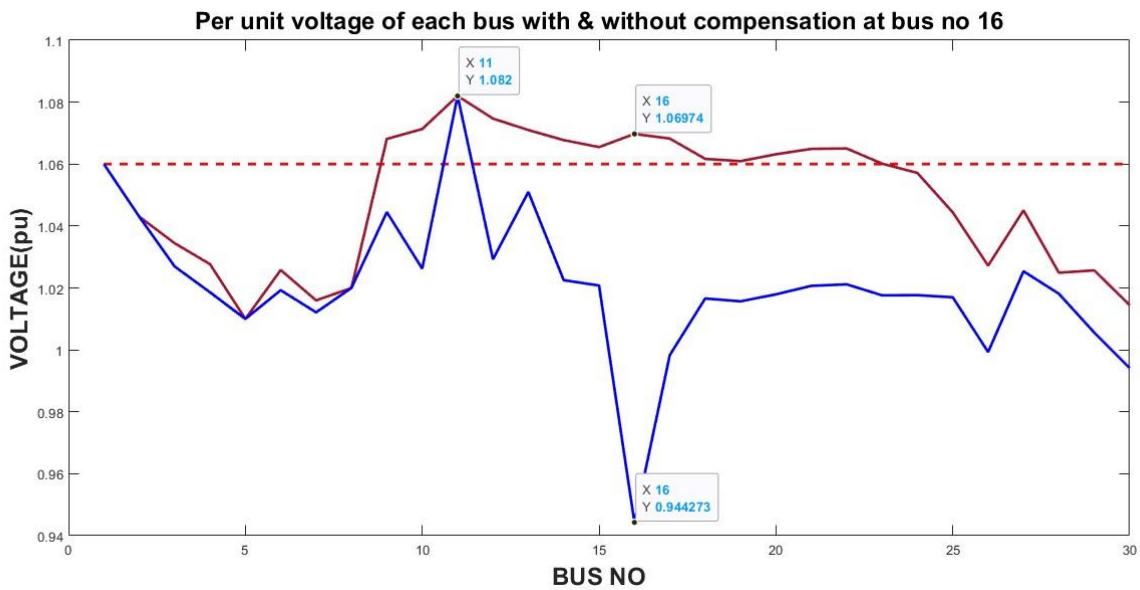
```
V15uc= 1.0655
V15c = 0.9631
Q15c = 67.6824

      "Loop ran for 403 times">>
```




```
% By using compensation at bus no. 16
Qo=busdata(16, 6);
V16uc = busdata(16, 3);
while busdata(9, 3)> 1.06 || busdata(10, 3)> 1.06 || busdata(12, 3)> 1.06 ||
busdata(13, 3)> 1.06 || busdata(17, 3)> 1.06
    q = q+1;
busdata(16, 6) = 1.01*busdata(16, 6);
   LFYBUS
   LFNEWTON
end
   plot(Vm)
   title('Per unit voltage of each bus with & without compensation at bus no 16')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
   V16uc
   V16c = busdata(16, 3)
   Q16c = busdata(16, 6)- Qo
% grid on
 fprintf('\n\t"Loop ran for %g times"',q);
```

➢ RESULT OBTAINED:

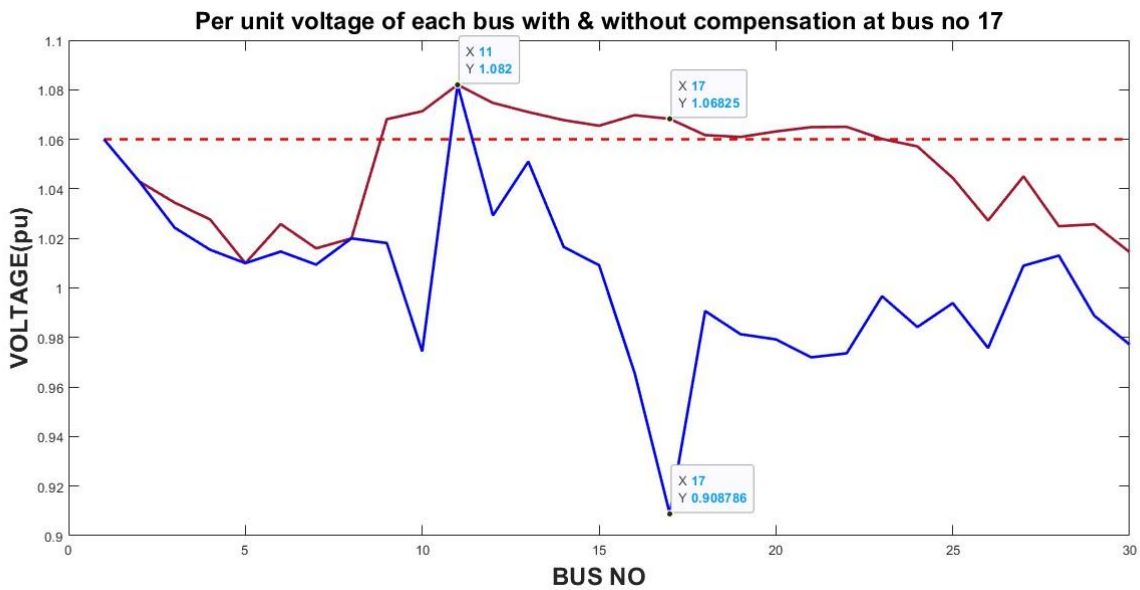
```
V16uc= 1.0697
V16c = 0.9443
Q16c = 64.6775
      "Loop ran for 431 times">>
```




```
% By using compensation at bus no. 17
Qo=busdata(17, 6);
V17uc = busdata(17, 3);
while busdata(9, 3)> 1.06 || busdata(10, 3)> 1.06 || busdata(12, 3)> 1.06 ||
busdata(13, 3)> 1.06 || busdata(17, 3)> 1.06
    q = q+1;
busdata(17, 6) = 1.01*busdata(17, 6);
   LFYBUS
   LFNEWTON
end
   plot(Vm)
   title('Per unit voltage of each bus with & without compensation at bus no 17')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
   V17uc
   V17c = busdata(17, 3)
   Q17c = busdata(17, 6)- Qo
% grid on
 fprintf('\n\t"Loop ran for %g times"',q);
```

➢ RESULT OBTAINED:

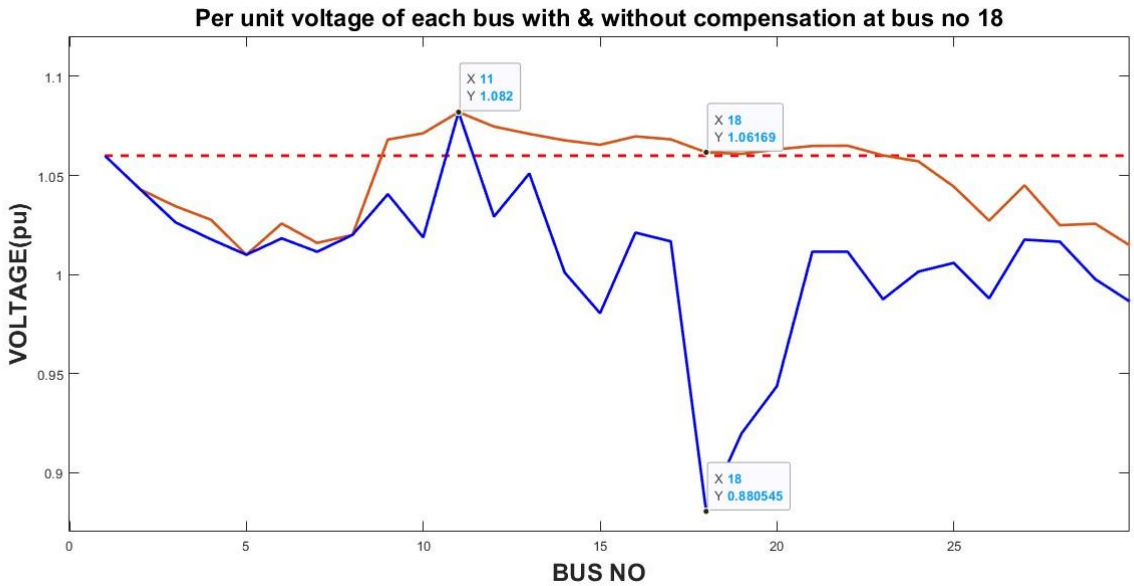
```
V17uc= 1.0682
V17c = 0.9088
Q17c = 92.4235
      "Loop ran for 351 times">>
```




```
% By using compensation at bus no. 18
Qo=busdata(18, 6);
V18uc = busdata(18, 3);
while busdata(9, 3)> 1.06 || busdata(10, 3)> 1.06 || busdata(12, 3)> 1.06 ||
busdata(13, 3)> 1.06 || busdata(17, 3)> 1.06
    q = q+1;
busdata(18, 6) = 1.01*busdata(18, 6);
   LFYBUS
   LFNEWTON
end
   plot(Vm)
   title('Per unit voltage of each bus with & without compensation at bus no 18')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
   V18uc
   V18c = busdata(18, 3)
   Q18c = busdata(18, 6)- Qo
% grid on
 fprintf('\n\t"Loop ran for %g times"',q);
```

➢ RESULT OBTAINED:

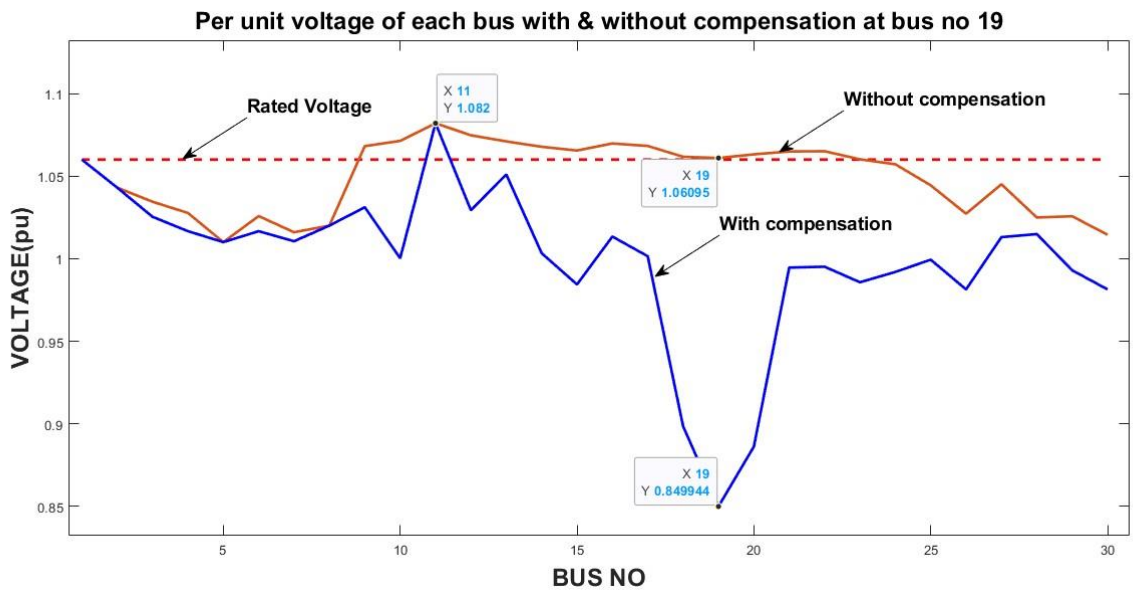
```
V18uc= 1.0617
V18c = 0.8805
Q18c = 65.3492

      "Loop ran for 501 times">>
```




```
% By using compensation at bus no. 19
Qo=busdata(19, 6);
V19uc = busdata(19, 3);
while busdata(9, 3)> 1.06 || busdata(10, 3)> 1.06 || busdata(12, 3)> 1.06 ||
busdata(13, 3)> 1.06 || busdata(17, 3)> 1.06
    q = q+1;
busdata(19, 6) = 1.01*busdata(19, 6);
   LFYBUS
   LFNEWTON
end
   plot(Vm)
   title('Per unit voltage of each bus with & without compensation at bus no 19')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
   V19uc
   V19c = busdata(19, 3)
   Q19c = busdata(19, 6)- Qo
% grid on
 fprintf('\n\t"Loop ran for %g times"',q);
```

➢ RESULT OBTAINED:

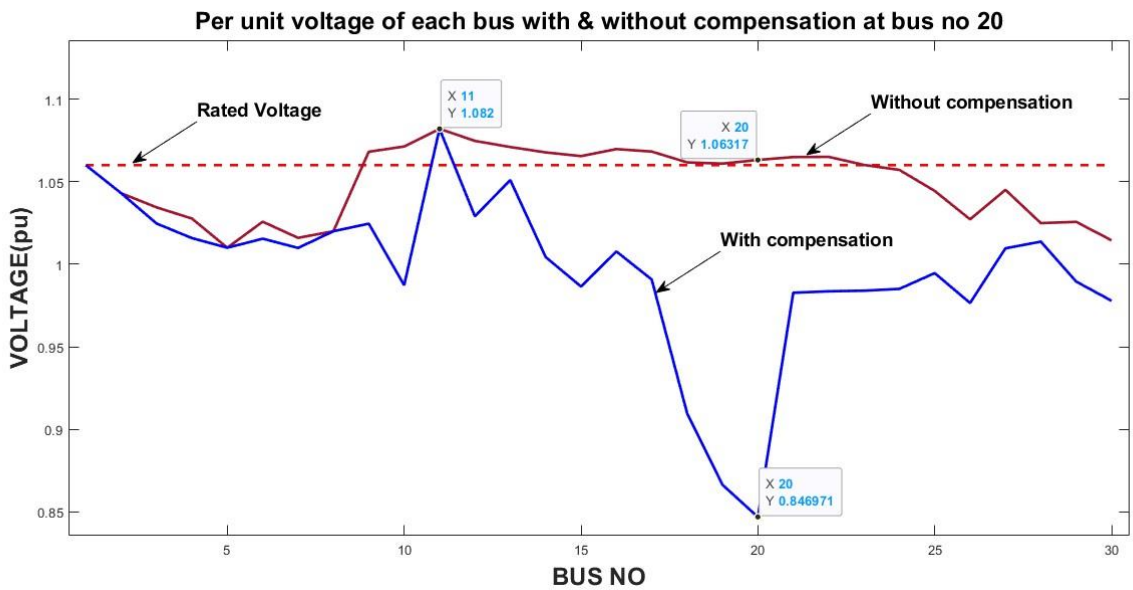
```
V19uc= 1.0609
V19c = 0.8499
Q19c = 72.8712
       "Loop ran for 380 times">>
```




```
% By using compensation at bus no. 20
Qo=busdata(20, 6);
V20uc = busdata(20, 3);
while busdata(9, 3)> 1.06 || busdata(10, 3)> 1.06 || busdata(12, 3)> 1.06 ||
busdata(13, 3)> 1.06 || busdata(17, 3)> 1.06
    q = q+1;
busdata(20, 6) = 1.01*busdata(20, 6);
   LFYBUS
   LFNEWTON
end
   plot(Vm)
   title('Per unit voltage of each bus with & without compensation at bus no 20')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
   V20uc
   V20c = busdata(20, 3)
   Q20c = busdata(20, 6)- Qo
% grid on
 fprintf('\n\t"Loop ran for %g times"',q);
```

➢ RESULT OBTAINED:

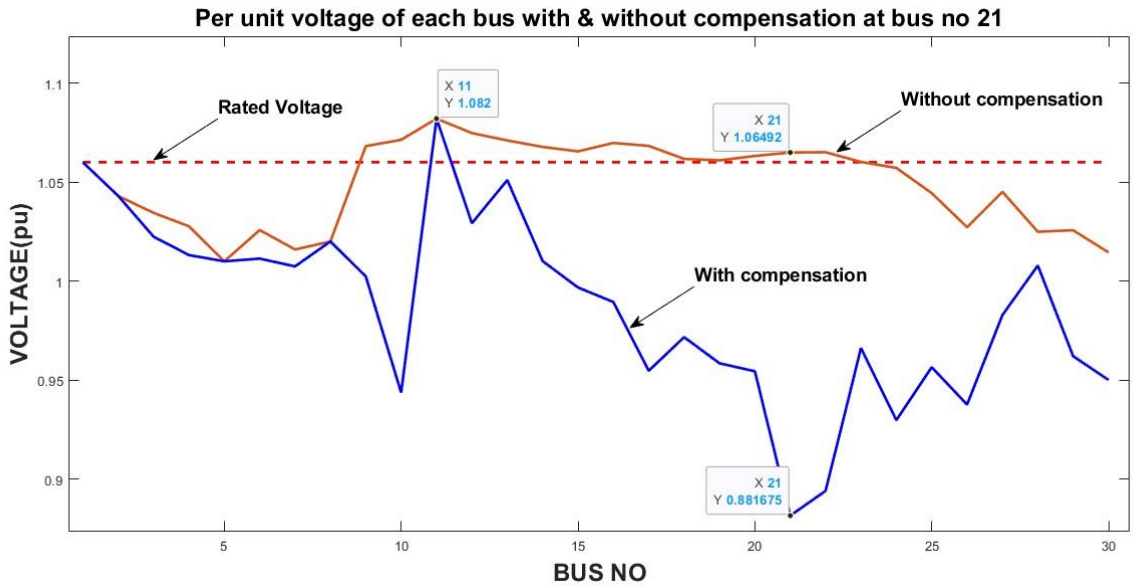
```
V20uc= 1.0632
V20c = 0.8470
Q20c = 79.7326
      "Loop ran for 546 times">>
```



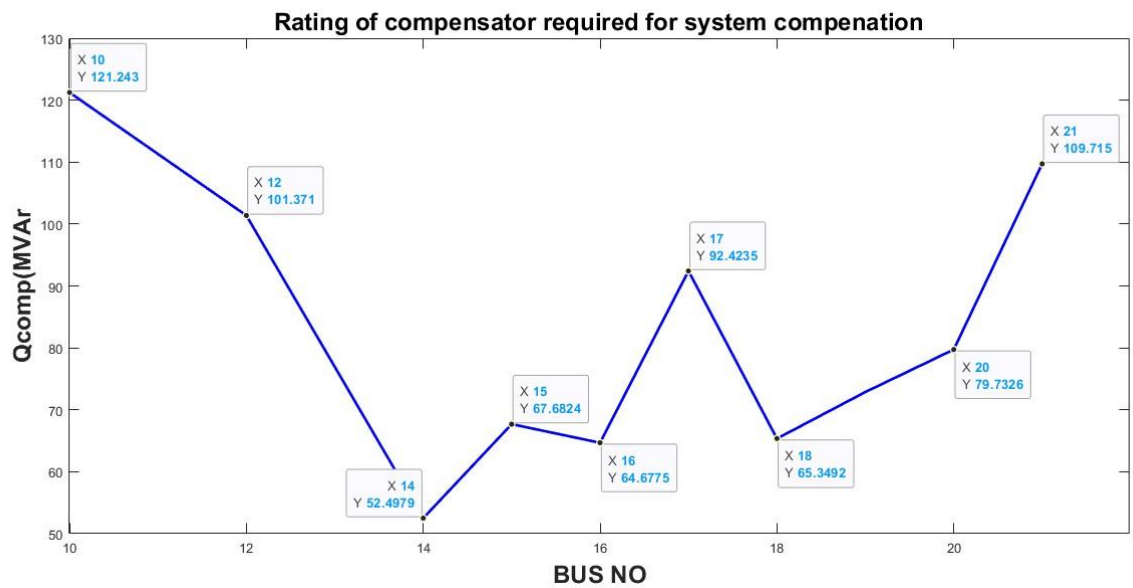

```
% By using compensation at bus no. 21
Qo=busdata(21, 6);
V21uc = busdata(21, 3);
while busdata(9, 3)> 1.06 || busdata(10, 3)> 1.06 || busdata(12, 3)> 1.06 ||
busdata(13, 3)> 1.06 || busdata(17, 3)> 1.06
    q = q+1;
busdata(21, 6) = 1.01*busdata(21, 6);
   LFYBUS
   LFNEWTON
end
   plot(Vm)
   title('Per unit voltage of each bus with & without compensation at bus no 21')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
   V21uc
   V21c = busdata(21, 3)
   Q21c = busdata(21, 6)- Qo
% grid on
 fprintf('\n\t"Loop ran for %g times"',q);
```

➢ RESULT OBTAINED:

```
V21uc= 1.0649
V21c = 0.8817
Q21c = 109.7150
      "Loop ran for 304 times">>
```




```
% Plot of Rating of Compensator Vs buses.
Qcomp= [121.2429 101.3715 52.4979 67.6824 64.6775 92.4235 65.3492 72.8712 79.7326
109.7150];
OVBus=[10 12 14 15 16 17 18 19 20 21];
plot(OVBus, Qcomp)
      title('Rating of compensator required for system compenation')
      xlabel('BUS NO')
      ylabel('Qcomp(MVAr')
```



➢ CONCLUSION:

- ✓ In this analysis we are not compensating the voltage at bus 11 because it is a PV Bus and also nothing is specified in generator columns also. Excluding it we have compensated the entire system by placing suitable SVC at all PV buses having over voltage.

- ✓ One thing is also noticeable here that after using SVC the voltage at that bus is reduced by a large extent this is because we are compensating the entire system as it was the objective of our experiment to know the required rating of SVC & for that the voltage of that particular bus should be reduced up to that particular value.
- ✓ If we compensate only that bus voltage where SVC is placed then the compensated voltage will be at the boundary.

➢ SECOND CASE: BY GENERATOR OUTAGE CONNECTED AT BUS 2

ALGORITHM:

1. First, we have to call Datafile of the given 30 bus standard system.
2. After defining appropriate variables we'll assign zero values to columns of bus 2 corresponding to generator.
3. In this case there will be overvoltage only at 11th (PV) & 13th (PV) buses.
4. To reduce overvoltage at 13th bus we have used compensation at bus 12.
5. Now after that we have plot the voltage with respect to buses for both the cases with compensation and without compensation.

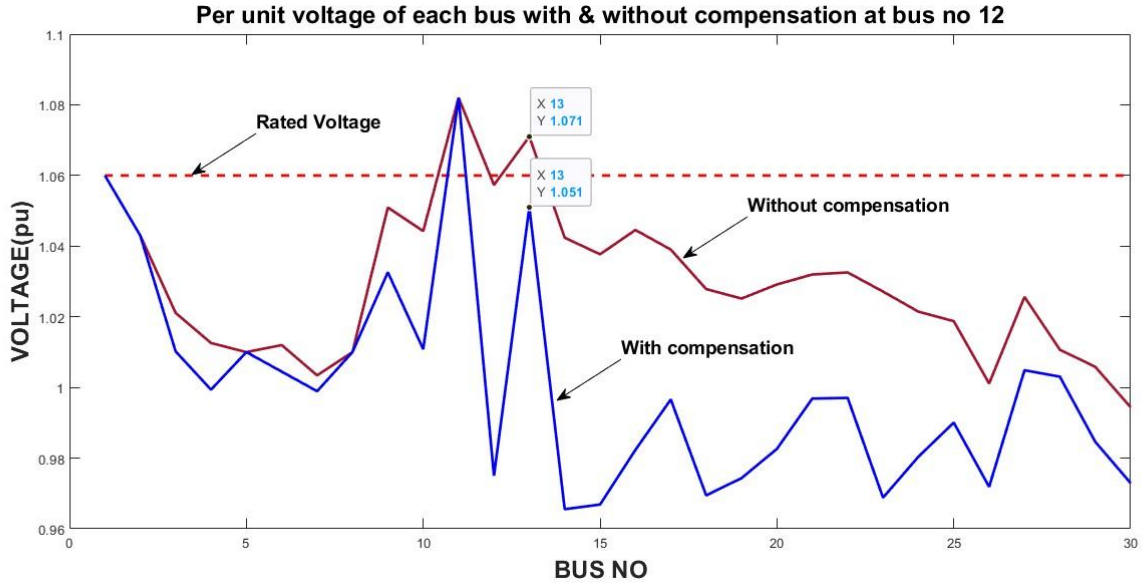
➢ MATLAB CODES

```
% CASE 2: Generator Outage
clear; clc;
Datafile
nbus = length(busdata(:,1));
q=0;
busdata(2, 7)=0; busdata(2, 8)=0; busdata(2, 9)=0; busdata(2, 10)=0;
   LFYBUS
   LFNEWTON
   vcc1=1.06*ones(1,30)
   plot(vcc1,'y--')
   hold on
   plot(Vm)
   title('Per unit voltage of each bus')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on

% By using compensation at bus no. 12
Qo=busdata(12, 6);
V13uc = busdata(13, 3);
while busdata(13, 3)> 1.06
    q = q+1;
busdata(12, 6) = 1.01*busdata(12, 6);
   LFYBUS
   LFNEWTON
end
   plot(Vm)
   title('Per unit voltage of each bus with & without compensation at bus no 12')
   xlabel('BUS NO')
```

```
   ylabel('VOLTAGE(pu)')
   hold on
   V13uc
   V13c=busdata(13, 3)
   Q12c = busdata(12, 6)- Qo
% grid on
 fprintf('\n\t"Loop ran for %g times"',q);
```

➢ RESULT OBTAINED:

```
V13uc= 1.0710
V13c = 1.0510
Q12c = 95.2029
                    "Loop ran for 263 times">>
```



➢ CONCLUSION:

- ✓ In this case there is only two buses having over voltages i.e., 11th & 13th & both are PV buses.
- ✓ By using SVC on either of these two buses compensation is not occurring that's why I have placed the SVC at bus no. 12 to compensate voltage of 13th bus.
- ✓ On thing should be taken into consideration that bus no 12 doesn't have any over voltage here nut just to compensate 13thy bus voltage I have placed SVC at this bus(bus no. 12).

➢ THIRD CASE: BY LINE OUTAGE

ALGORITHM:

1. First, we have to call Datafile of the given 30 bus standard system.
2. After defining appropriate variables we'll assign null matrix to the particular line (rows in line data file) to be removed.

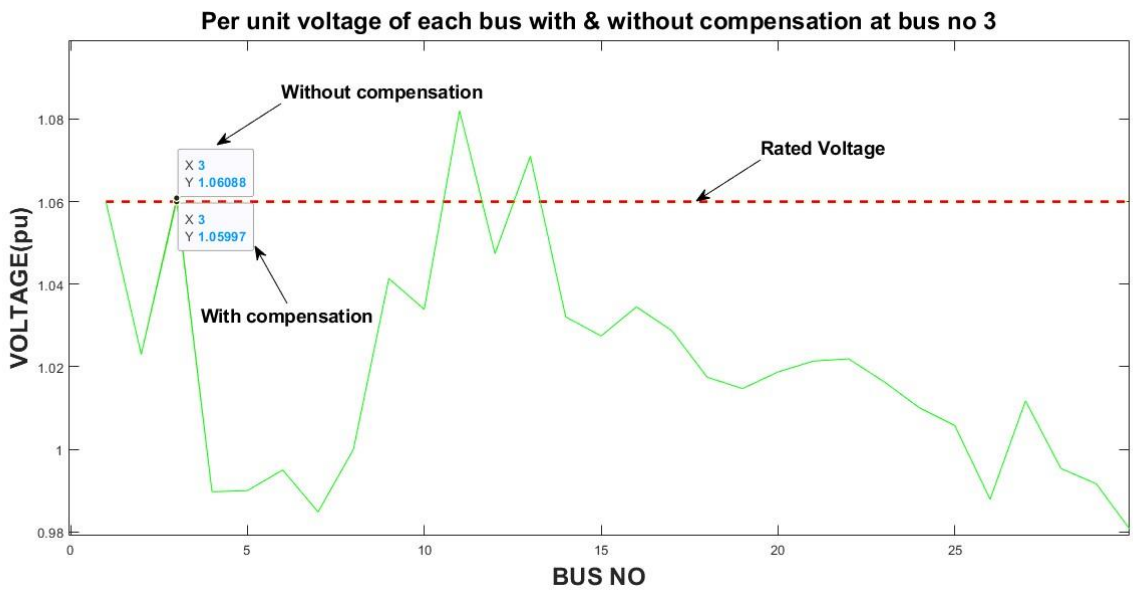
3. There will be over voltage at some buses now we'll compensate it using appropriate svc at those buses.
4. Now after that we have plot the voltage with respect to buses for both the cases with compensation and without compensation.

➢ MATLAB CODES

```
CASE-3 LINE OUTAGE
clear; clc;
Datafile
nbus = length(busdata(:,1));
bc = busdata(:,2); q=0;

% % By removing 4rth line
linedata(4,:) = []
   LFYBUS
   LFNEWTON
   vcc1=1.06*ones(1,30)
   plot(vcc1,'y--')
   hold on
   plot(Vm)
   title('Per unit voltage of each bus')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
%to compensate overvoltage at bus no. 3
Qo=busdata(3, 6);
V3uc = busdata(3, 3);
while busdata(3, 3)> 1.06
    q = q+1;
busdata(3, 6) = 1.01*busdata(3, 6);
   LFYBUS
   LFNEWTON
end
   plot(Vm)
   title('Per unit voltage of each bus with & without compensation at bus no 3')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
   V3uc
   V3c = busdata(3, 3)
   Q3c = busdata(3, 6)- Qo
% grid on
 fprintf('\n\t"Loop ran for %g times"',q);
```

➢ RESULT OBTAINED:

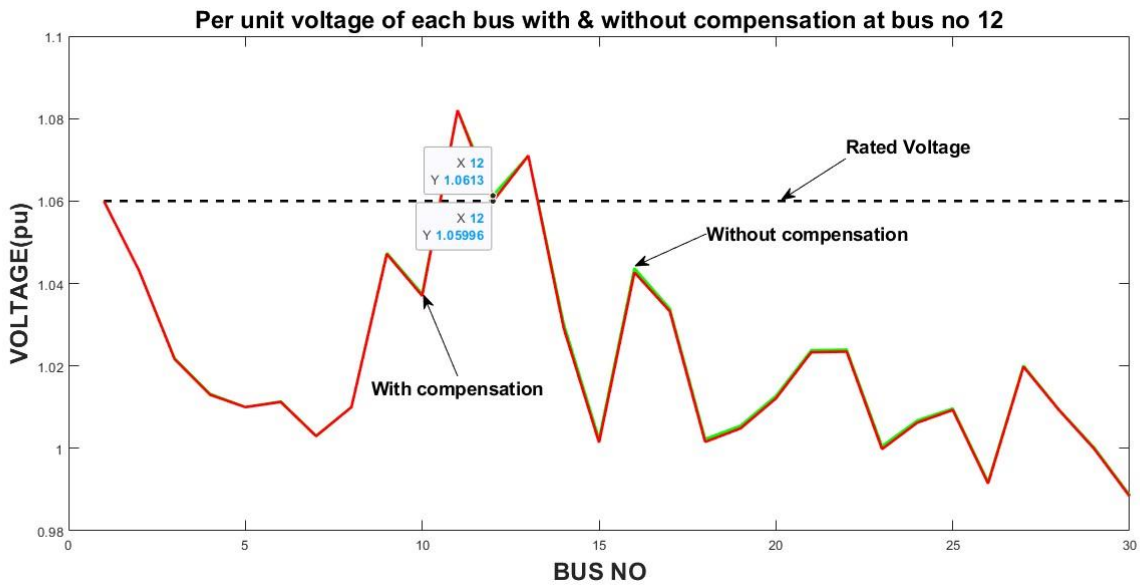
```
V3uc = 1.0609
V3c = 1.0600
Q3c = 0.5169
        "Loop ran for 36 times">>
```




```
% % By removing 18th line
linedata(18,:) = []
   LFYBUS
   LFNEWTON
   vcc1=1.06*ones(1,30)
   plot(vcc1,'y--')
   hold on
   plot(Vm)
   title('Per unit voltage of each bus')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
%to compensate overvoltage at bus no. 12
Qo=busdata(12, 6);
V12uc = busdata(12, 3);
while busdata(12, 3)> 1.06
    q = q+1;
busdata(12, 6) = 1.01*busdata(12, 6);
   LFYBUS
   LFNEWTON
end
   plot(Vm)
   title('Per unit voltage of each bus with & without compensation at bus no 12')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
   V12uc
   V12c = busdata(12, 3)
   Q12c = busdata(12, 6)- Qo
% grid on
 fprintf('\n\t"Loop ran for %g times"',q)
```

➢ RESULT OBTAINED:

V12uc = 1.0613
V12c = 1.0600
Q12c = 1.7429
"Loop ran for 21 times">>



➢ FOURTH CASE: BY VARYING ONLY Q

ALGORITHM:

1. First, we have to call Datafile of the given 30 bus standard system.
2. After defining appropriate variables we'll reduce Q for some buses (say from 2 to 25) by some extent (say 40% to create over voltages)
3. Now we'll compensate system over voltage by using compensation at all PQ buses having over voltage.
4. After that we'll plot the voltage with respect to buses for both the cases with compensation and without compensation.

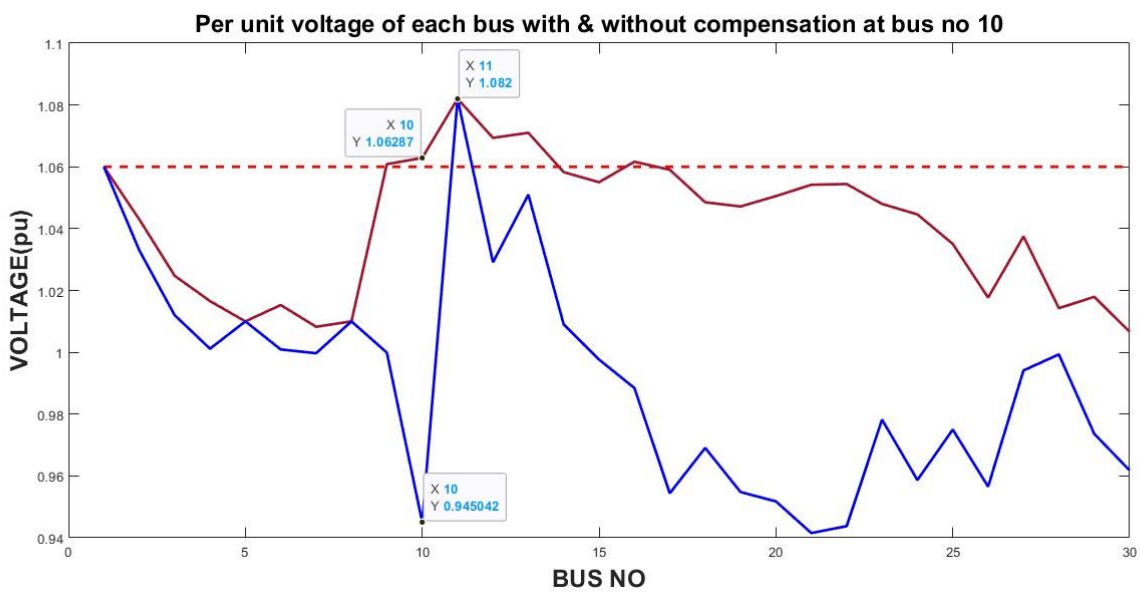
➢ MATLAB CODES

```
% CASE 4: BY VARYING Only Q.
clear; clc;
Datafile
q=0;
nbus = length(busdata(:,1));
busdata(2:25, 6)= 0.45*busdata(2:25, 6);
Qo=busdata(12, 6);
LFYBUS
LFNEWTON
vcc1=1.06*ones(1,30);
plot(vcc1,'y--')
hold on
plot(Vm)
```

```
title('PER UNIT VOLTAGE OF EACH BUS FOR THE GIVEN 30 BUS SYSTEM')
xlabel('BUS NO')
ylabel('VOLTAGE(VOLT)')
hold on

% By using compensation at bus no. 10
Qo=busdata(10, 6);
V10uc = busdata(10, 3);
while busdata(9, 3)> 1.06 || busdata(10, 3)> 1.06 || busdata(12, 3)> 1.06 ||
busdata(13, 3)> 1.06 || busdata(16, 3)> 1.06
    q = q+1;
busdata(10, 6) = 1.01*busdata(10, 6);
   LFYBUS
   LFNEWTON
end
   plot(Vm)
   title('Per unit voltage of each bus with & without compensation at bus no 10')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
   V10uc
   V10c=busdata(10, 3)
   Q10c = busdata(10, 6)- Qo
% grid on
 fprintf('\n\t"Loop ran for %g times"',q);
```

➢ RESULT OBTAINED:

V10uc = 1.0629
V10c = 0.9450
Q10c = 101.7163
"Loop ran for 476 times">>

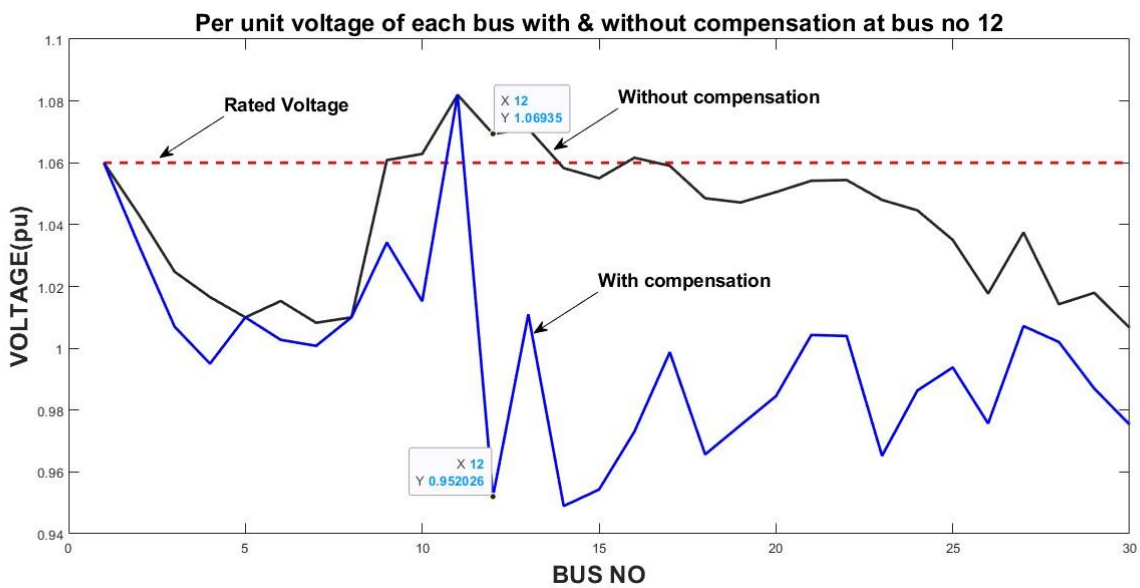



```
% By using compensation at bus no. 12
Qo=busdata(12, 6);
```

```
V12uc = busdata(12, 3);
while busdata(9, 3)> 1.06 || busdata(10, 3)> 1.06 || busdata(12, 3)> 1.06 ||
busdata(13, 3)> 1.06 || busdata(16, 3)> 1.06
    q = q+1;
busdata(12, 6) = 1.01*busdata(12, 6);
   LFYBUS
   LFNEWTON
end
   plot(Vm)
   title('Per unit voltage of each bus with & without compensation at bus no 12')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
   V12uc
   V12c=busdata(12, 3)
   Q12c = busdata(12, 6)- Qo
% grid on
 fprintf('\n\t"Loop ran for %g times"',q);
```

➢ RESULT OBTAINED:

V12uc = 1.0694
V12c = 0.9520
Q12c = 101.1326
"Loop ran for 345 times">>

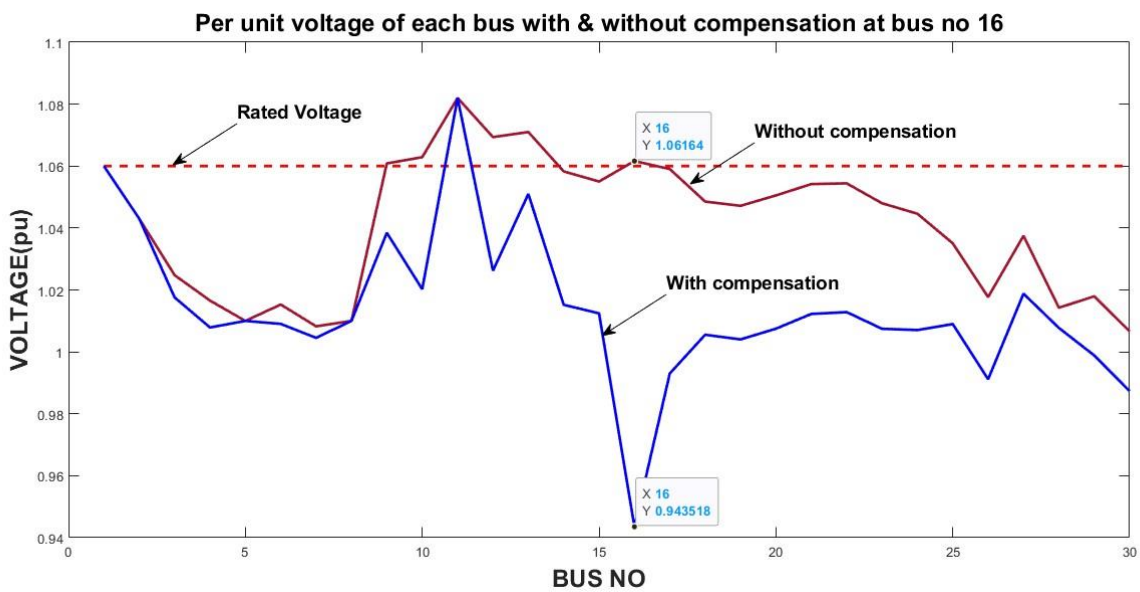



```
% By using compensation at bus no. 16
Qo=busdata(16, 6);
V16uc = busdata(16, 3);
while busdata(9, 3)> 1.06 || busdata(10, 3)> 1.06 || busdata(12, 3)> 1.06 ||
busdata(13, 3)> 1.06 || busdata(16, 3)> 1.06
    q = q+1;
busdata(16, 6) = 1.01*busdata(16, 6);
   LFYBUS
   LFNEWTON
```

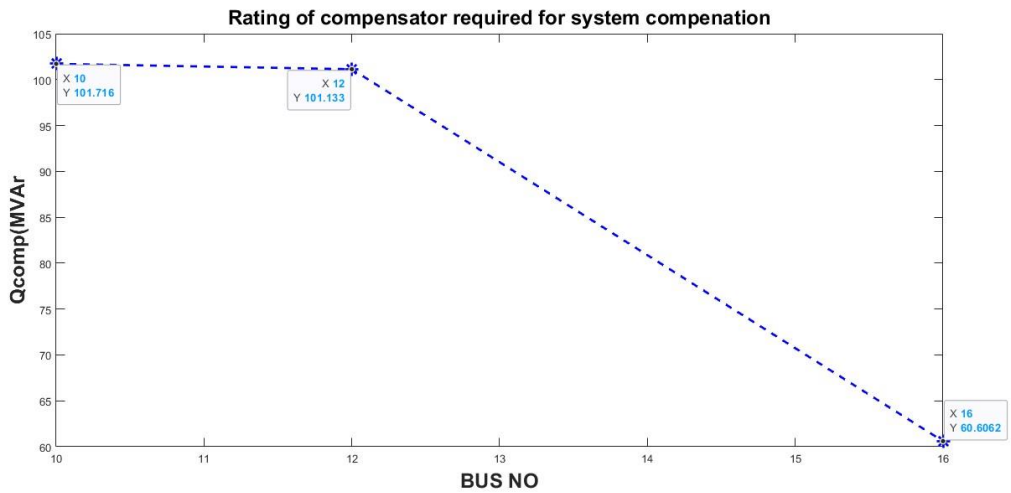
```
end
   plot(Vm)
   title('Per unit voltage of each bus with & without compensation at bus no 16')
   xlabel('BUS NO')
   ylabel('VOLTAGE(pu)')
   hold on
   V16uc
   V16c = busdata(16, 3)
   Q16c = busdata(16, 6)- Qo
% grid on
 fprintf('\n\t"Loop ran for %g times"',q);
```

➢ RESULT OBTAINED:

V16uc = 1.0616
V16c = 0.9435
Q16c = 60.6062
"Loop ran for 435 times">>




```
% Plot of Rating of Compensator Vs buses.
Qcomp=[101.7163 101.1326 60.6062];
OVBus=[10 12  16];
plot(OVBus, Qcomp)
title('Rating of compensator required for system compenation')
xlabel('BUS NO')
ylabel('Qcomp(MVAr')
```



➢ CONCLUSION:

- ✓ In this analysis we are not compensating the voltage at bus 11 because it is a PV Bus and also nothing is specified in generator columns also. Excluding it we have compensated the entire system by placing suitable SVC at nus no. 10, 12 & 16.
- ✓ One thing is also noticeable here that after using SVC the voltage at that bus is reduced by a large extent this is because we are compensating the entire system as it was the objective of our experiment to know the required rating of SVC & for that the voltage of that particular bus should be reduced up to that particular value.
- ✓ If we compensate only that bus voltage where SVC is placed then the compensated voltage will be at the boundary.

ASSIGHNMENT NO. # 04

(29/03/23 & 05/04/23)

NAME: SHIVRAJ VISHWAKARMA **ROLL. NO.:** 224102112

OBJECTIVE

Backward-forward sweep load flow analysis for radial distribution network

- **SOFTWARE:** MATLAB R2022a
- **THEORETICAL ANALYSIS**

Load flow studies are one of the most important aspects of power system planning and operation. Load flow analysis provides a way for steady state analysis of the power system and thus provides the steady state values of different quantities like, bus voltages, line currents, losses, real and reactive power generated or absorbed. Since the governing equations for load flow studies are nonlinear, for transmission networks, the load flow is carried out using iterative numerical techniques like Newton-Raphson method and Gauss-Seidel method. However, for distribution networks the methods used for transmission system are not effective because of the following reasons.

- The distribution networks are highly radial in nature and weakly meshed unlike transmission system.
- The R/X ratio of distribution networks is high unlike transmission networks for which it is high.
- The distribution systems have various sourced connected at various locations known as distributed generations.

Thus, keeping in view, the above-mentioned points, the load flow analysis for distribution networks is carried out using other techniques, among which 'Backward and Forward Sweep' is one of the effective technique/ algorithms.

The Backward and Forward Sweep algorithm is carried out using following steps:

<u>Backward sweep</u>

- Initialize all bus voltages
- Calculate load currents using equation -

$$I_{l(i)} = \frac{(P_i - jQ_i)}{V_i}$$

where i represents the bus number where load is connected.

- Calculate branch currents starting from load end towards substation, using the equation –

$$I_{ki} = I_i + \sum_{J=i+1}^{N} I_{ij}$$

Where, N is the number of buses.

Forward sweep:

- Calculate all voltages (except slack bus which is chosen as 1pu) using the equation –

$$Vj = Vi - Iij * Zij$$

- The above process is carried out iteratively till the solution converges.

- **Here** the backward forward sweep load flow analysis is carried out for various distribution networks like 33 bus radial distribution system and 69 radial distribution networks.

- For carrying out the load flow algorithm we need to have bus-data, which provides data for all the loads present in the system, and line data which specifies the impedances of all the lines.

ALGORITHM

- ✓ First, we have to call Datafile of the given various (33/69/52) bus standard system.
- ✓ Then we'll calculate the load current at each bus.
- ✓ After defining appropriate variables we'll find the terminal bus because at terminal bus load current will be equal to line current so we'll assign those line current values.
- ✓ Then by using Backward sweep algorithm we'll calculate the other line currents.
- ✓ After that using Forword sweep algorithm we'll find the voltages at each bus.
- ✓ Finally, we'll calculate losses occurring in the lines & then we'll plot both the voltages & line losses.

FIRST CASE: 33 Bus radial distribution network

Bus-data:

Bus No.	Bus code	Voltage (pu)	Angle (°)	Load (kW)	Load (kVAr)
1	1	1	0	0	0
2	0	1	0	100	60
3	0	1	0	90	40
4	0	1	0	120	80
5	0	1	0	60	30
6	0	1	0	60	20
7	0	1	0	200	100
8	0	1	0	200	100
9	0	1	0	60	20
10	0	1	0	60	20
11	0	1	0	45	30

12	0	1	0	60	35
13	0	1	0	60	35
14	0	1	0	120	80
15	0	1	0	60	10
16	0	1	0	60	20
17	0	1	0	60	20
18	0	1	0	90	40
19	0	1	0	90	40
20	0	1	0	90	40
21	0	1	0	90	40
22	0	1	0	90	40
23	0	1	0	90	50
24	0	1	0	420	200
25	0	1	0	420	200
26	0	1	0	60	25
27	0	1	0	60	25
28	0	1	0	60	20
29	0	1	0	120	70
30	0	1	0	200	600
31	0	1	0	150	70
32	0	1	0	210	100
33	0	1	0	60	40

Line-data:

Line no.	From bus	To bus	Resistance (r)	Reactance (x)
1	1	2	0.0922	0.0470
2	2	3	0.4930	0.2511
3	3	4	0.3660	0.1864
4	4	5	0.3811	0.1941
5	5	6	0.8190	0.7070
6	6	7	0.1872	0.6188
7	7	8	0.7114	0.2351
8	8	9	1.0300	0.7400
9	9	10	1.0440	0.7400
10	10	11	0.1966	0.0650
11	11	12	0.3744	0.1238
12	12	13	1.4680	1.1550
13	13	14	0.5416	0.7129
14	14	15	0.5910	0.5260
15	15	16	0.7463	0.5450
16	16	17	1.2890	1.7210

17	17	18	0.7320	0.5740
18	2	19	0.1640	0.1565
19	19	20	1.5042	1.3554
20	20	21	0.4095	0.4784
21	21	22	0.7089	0.9373
22	3	23	0.4512	0.3083
23	23	24	0.8980	0.7091
24	24	25	0.8960	0.7011
25	6	26	0.2030	0.1034
26	26	27	0.2842	0.1447
27	27	28	1.0590	0.9337
28	28	29	0.8042	0.7006
29	29	30	0.5075	0.2585
30	30	31	0.9744	0.9630
31	31	32	0.3105	0.3619
32	32	33	0.3410	0.5302

➢ MATLAB CODES

```
busata_33bus;
j=sqrt(-1);
basemva = 100;
basekv = 12.66;
    linedata(:,3) = line_data(:,3)*basemva/(basekv^2);
    linedata(:,4) = line_data(:,4)*basemva/(basekv^2);
accuracy = 0.0001; maxiter=100;

%%%%%%%%%%%%%%%Extracting the line data and the busdata%%%%%%%%%%%%%%%%%

nbr=length(line_data(:,1));   %  Number of Line
nbus=length(bus_data(:,1));    %  Number of Bus
P = bus_data(:,5)/(1000*basemva);    %Load Active Power
Q = bus_data(:,6)/(1000*basemva);    %Load Reactive Power
V = bus_data(:,3);  % Bus voltage except substation bus
angle = bus_data(:,4);


ns = line_data(:,1);        %  Starting Node
nr = line_data(:,2);        %  To Node
R = line_data(:,3);
X = line_data(:,4);
ln=line_data(:,end);        %line number
Z =R+j*X;
nl=nbr;
```

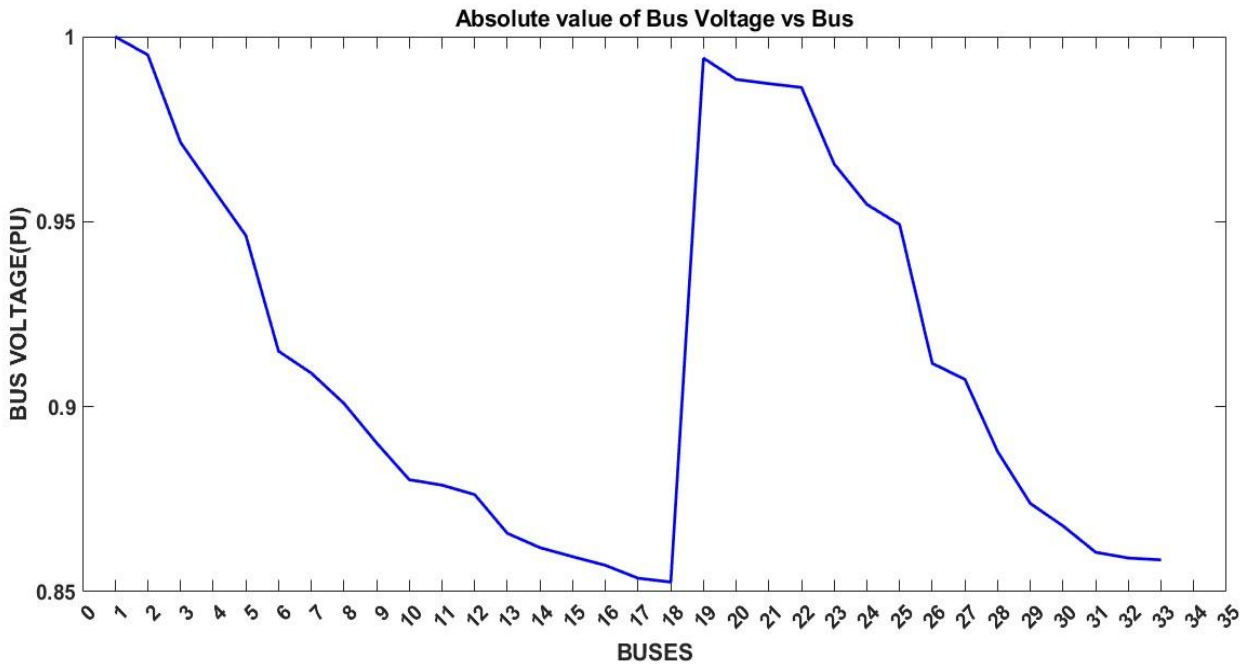
```
Vlt=bus_data(:,3)+j*bus_data(:,4);

%%%%%%%%%%%%%%%%%%%%%%%%%Load current
calculation%%%%%%%%%%%%%%%%%%%%%%%%%%
bc=bus_data(:,2);
iter=0;
g=1;
while g==1
iter=iter+1;
for p=1:nbus
   if bc(p)==0
   I_l(p)=(P(p)-j*Q(p))/conj(Vlt(p));
   end
end
I_l = I_l(2:end)
%%%%%%%%%%%%%%%%%%%%%%%%%%%% Backword_Sweep
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
v=1;
z=nbr;
for m =nl:-1:1
      term_nr = any(ns == nr(m));
      if term_nr == 0
        i_L(z)=I_l(m);
      else
         u=find(ns==nr(m));
        if length(u)==1
          i_L(z)=I_l(u(1)-1);
          for c=1:length(u)
            i_L(z)=i_L(z)+i_L(u(c));
          end
        else
           if length(u)==2
           i_L(z)=I_l(u(1)-1);
           for c=1:length(u)
            i_L(z)=i_L(z)+i_L(u(c));
           end
           end
        end
      end
      z=z-1;
end
i_L;
aa = [1:32];
curent = [aa' i_L' I_l']
%%%%%%%%%%%%%%%%%%%%%%%%%%%% Forword sweep
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for x=1:nbr
    w=find(ns==x);
```

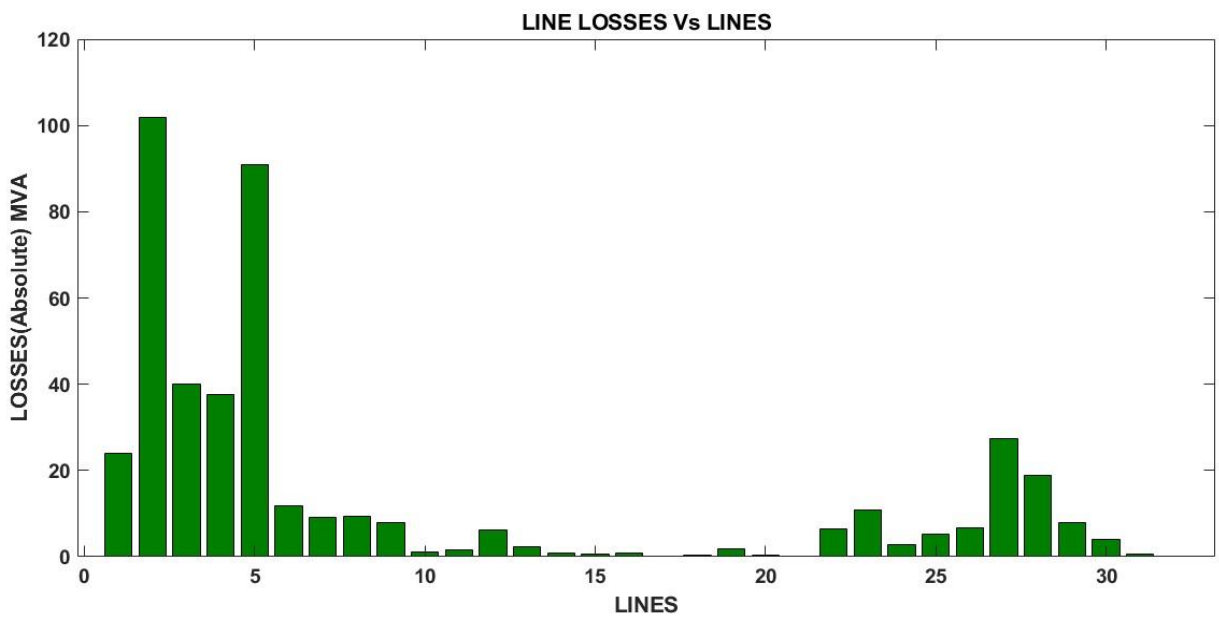
```
    if length(w)==1
    V(x+1)=V(x)-i_L(x)*Z(x);
    end
end
for x=1:nbr
    w=find(ns==x);
  if length(w)==0
V(x)=V(x-1)-i_L(x-1)*Z(x-1);
  end
end
for x=1:nbr
    w=find(ns==x);
    if length(w)==2
       V(w(1)+1)=V(x)-i_L(x)*Z(x);
       V(w(2)+1)=V(w(1))-i_L(w(2))*Z(w(2));
    end
end
V(1)=Vlt(1);
err=(abs(Vlt) - abs(V));
error = max(err);
if error>accuracy
    if iter>maxiter
        fprintf('\nWARNING: Iterative solution did not converged after ')
   fprintf('%g', iter), fprintf(' iterations.\n\n')
    g=0;
    else
     g=1;
    end
else
    if iter>maxiter
        fprintf('\nWARNING: Iterative solution did not converged after ')
   fprintf('%g', iter), fprintf(' iterations.\n\n')
    g=0;
    else
       fprintf('\nWARNING: Iterative solution  converges after ')
   fprintf('%g', iter), fprintf(' iterations.\n\n')
g=0
    end
end
   Vlt=V;
end
V
figure
plot(bus_data(:,1), abs(V))
%%%%%%%%%%%%%%%%%%%%%%%%% LOSSES CALCULATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for d=1:nbr
    loss(d)=((abs(i_L(d)))^2)*Z(d)*1000*basemva;
end
```

```
totalloss = 0;
for e = 1:length(loss)
 totalloss =(totalloss + loss(e));
end
MINIMUM_VOLTAGE = min(abs(V))
MINIMUM_VOLTAGE_BUS = find(abs(V) == MINIMUM_VOLTAGE)
REAL_LOSS = real(totalloss)
REACTIVE_LOSS = imag(totalloss)
figure
bar(line_data(:,5), abs(loss))
```

➢ RESULT OBTAINED:

```
WARNING: Iterative solution converges after 9 iterations.
g =
    0
V =
  1.0000 + 0.0000i                    0.8525 - 0.0132i
  0.9950 + 0.0004i                    0.9942 + 0.0001i
  0.9714 + 0.0028i                    0.9884 - 0.0017i
  0.9588 + 0.0046i                    0.9873 - 0.0023i
  0.9462 + 0.0065i                    0.9863 - 0.0028i
  0.9150 + 0.0037i                    0.9656 + 0.0019i
  0.9091 - 0.0027i                    0.9547 - 0.0006i
  0.9009 - 0.0017i                    0.9492 - 0.0018i
  0.8902 - 0.0036i                    0.9117 + 0.0048i
  0.8803 - 0.0053i                    0.9073 + 0.0063i
  0.8788 - 0.0051i                    0.8878 + 0.0085i
  0.8762 - 0.0048i                    0.8738 + 0.0104i
  0.8657 - 0.0072i                    0.8677 + 0.0132i
  0.8618 - 0.0093i                    0.8606 + 0.0108i
  0.8594 - 0.0103i                    0.8590 + 0.0102i
  0.8571 - 0.0109i                    0.8585 + 0.0100i
  0.8535 - 0.0130i

MINIMUM_VOLTAGE =
                    0.8526

MINIMUM_VOLTAGE_BUS =

                    18

REAL_LOSS =

                    360.3151

REACTIVE_LOSS =

                    240.6419
```





CONCLUSION:

- ✓ It can be seen from the above plot that the voltage magnitude close to 1 pu for the buses which are closer to the substation bus (slack bus) and the more away a node from the substation node, the voltage gets reduced due to line drop.
- ✓ The minimum voltage of the network is **0.8526 pu** at **bus no. 18**. The total real loss of the network comes out to be **360.3151 kW** and the reactive power loss of the distribution network is **240.6419 kVAr.**

➢ SECOND CASE: 69 Bus radial distribution network.

Bus-data:

Bus No.	Bus code	Voltage (pu)	Angle (°)	Load (kW)	Load (kVAr)
1	1	1	0	0	0
2	0	1	0	0	0
3	0	1	0	0	0
4	0	1	0	0	0
5	0	1	0	0	0
6	0	1	0	2.6	2.2
7	0	1	0	40.4	30
8	0	1	0	75	54
9	0	1	0	30	22
10	0	1	0	28	19
11	0	1	0	145	104
12	0	1	0	145	104
13	0	1	0	8	5.5
14	0	1	0	8	5.5
15	0	1	0	0	0
16	0	1	0	45.5	30
17	0	1	0	60	35
18	0	1	0	60	35
19	0	1	0	0	0
20	0	1	0	1	0.6
21	0	1	0	114	81
22	0	1	0	5.3	3.5
23	0	1	0	0	0
24	0	1	0	28	20
25	0	1	0	0	0
26	0	1	0	14	10
27	0	1	0	14	10
28	0	1	0	26	18.6
29	0	1	0	26	18.6
30	0	1	0	0	0
31	0	1	0	0	0
32	0	1	0	0	0
33	0	1	0	14	10
34	0	1	0	19.5	14
35	0	1	0	6	4

36	0	1	0	26	18.55
37	0	1	0	26	18.55
38	0	1	0	0	0
39	0	1	0	24	17
40	0	1	0	24	17
41	0	1	0	1.2	1
42	0	1	0	0	0
43	0	1	0	6	4.3
44	0	1	0	0	0
45	0	1	0	39.22	26.3
46	0	1	0	39.22	26.3
47	0	1	0	0	0
48	0	1	0	79	56.4
49	0	1	0	384.7	274.5
50	0	1	0	384.7	274.5
51	0	1	0	40.5	28.3
52	0	1	0	3.6	2.7
53	0	1	0	4.35	3.5
54	0	1	0	26.4	19
55	0	1	0	24	17.2
56	0	1	0	0	0
57	0	1	0	0	0
58	0	1	0	0	0
59	0	1	0	100	72
60	0	1	0	0	0
61	0	1	0	1244	888
62	0	1	0	32	23
63	0	1	0	0	0
64	0	1	0	227	162
65	0	1	0	59	42
66	0	1	0	18	13
67	0	1	0	18	13
68	0	1	0	28	20
69	0	1	0	28	20

Line data:

Line no.	From bus	To bus	Resistance (r)	Reactance (x)
1	1	2	0.0005	0.0012
2	2	3	0.0005	0.0012
3	3	4	0.0015	0.0036

4	4	5	0.0251	0.0294
5	5	6	0.3660	0.1864
6	6	7	0.3811	0.1941
7	7	8	0.0922	0.0470
8	8	9	0.0493	0.0251
9	9	10	0.8190	0.2707
10	10	11	0.1872	0.0691
11	11	12	0.7114	0.2351
12	12	13	1.0300	0.3400
13	13	14	1.0440	0.3450
14	14	15	1.0580	0.3496
15	15	16	0.1966	0.0650
16	16	17	0.3744	0.1238
17	17	18	0.0047	0.0016
18	18	19	0.3276	0.1083
19	19	20	0.2106	0.0696
20	20	21	0.3416	0.1129
21	21	22	0.0140	0.0046
22	22	23	0.1591	0.0526
23	23	24	0.3463	0.1145
24	24	25	0.7488	0.2745
25	25	26	0.3089	0.1021
26	26	27	0.1732	0.0572
27	3	28	0.0044	0.0108
28	28	29	0.0640	0.1565
29	29	30	0.3978	0.1315
30	30	31	0.0702	0.0232
31	31	32	0.3510	0.1160
32	32	33	0.8390	0.2816
33	33	34	1.7080	0.5646
34	34	35	1.4740	0.4873
35	3	36	0.0044	0.0108
36	36	37	0.0640	0.1565
37	37	38	0.1053	0.1230
38	38	39	0.0304	0.0355
39	39	40	0.0018	0.0021
40	40	41	0.7283	0.8509
41	41	42	0.3100	0.3623
42	42	43	0.0410	0.0478
43	43	44	0.0092	0.0116
44	44	45	0.1089	0.1373

45	45	46	0.0009	0.0012
46	4	47	0.0034	0.0084
47	47	48	0.0851	0.2083
48	48	49	0.2898	0.7091
49	49	50	0.0822	0.2011
50	8	51	0.0928	0.0473
51	51	52	0.3319	0.1114
52	9	53	0.1740	0.0886
53	53	54	0.2030	0.1034
54	54	55	0.2842	0.1447
55	55	56	0.2813	0.1433
56	56	57	1.5900	0.5337
57	57	58	0.7837	0.2630
58	58	59	0.3042	0.1006
59	59	60	0.3861	0.1172
60	60	61	0.5075	0.2585
61	61	62	0.0974	0.0496
62	62	63	0.1450	0.0738
63	63	64	0.7105	0.3619
64	64	65	1.0410	0.5302
65	11	66	0.2012	0.0611
66	66	67	0.0047	0.0014
67	12	68	0.7394	0.2444
68	68	69	0.0047	0.0016

➢ MATLAB CODES

```
clc; clear all; clc;
busata_69
bd = busdataa69; ld = linedataa69;
mvabase = 10; kvbase  = 12.66;
zbase   = kvbase^2./mvabase;
ld(:,4:5) = ld(:,4:5)/zbase;
bd(:,5:6) = bd(:,5:6)/(1000*mvabase);
bd(30,6)  = bd(30,6) - 0/(1000*mvabase);
 z  = complex(ld(:,4),ld(:,5));
 s  = complex(bd(:,5),bd(:,6));          % complex power S = P + j.Q
 nb = max(max(ld(:,2:3)));
 v  = eye(nb,nb); voltage_old  = diag(v);
zmatrix = zeros(nb,nb);
for i = 1:size(zmatrix,1)-1
    for j = 1:size(zmatrix,1)-1
        zmatrix(ld(i,2),ld(i,3)) = z(i); % line impedence z = r + j.x
    end
end
```

```
%  iter  = 100;                          %  for k = 1:iter
count    = 1;  accuracy = 0.0001;  error    = 1;
 while error > accuracy

%%%%%%%%%%%%% BACKWARD SWEEP %%%%%%%%%%%%%%%%%%%
imatrix = zeros(nb,nb);
for i=1:nb
 iload(i) = conj(s(i)./v(i,i)); % load currents at the buses.
 imatrix(i,i) = iload(i);
end
for i = nb:-1:1
 [q,w] = find(linedataa69(:,2:3)==i);
 for j=nb:-1:1
 if j~=i && zmatrix(j,i) ~= 0
 if size(q)==1 % lines connected to end buses
 imatrix(j,i) =iload(i);
 else % if not line to the end bus
 for k = nb:-1:i
 imatrix(j,i) = imatrix(j,i) + imatrix(i,k);
 end
 end
 end
 end
end

%%%%%%%%%% FORWARD SWEEP%%%%%%%%%%%%%
for i=1:nb-1
 for j=1:nb
 if j~=i
 v(i,j) = zmatrix(i,j)*imatrix(i,j); % line drop calculation
 end
 end
end
for j=1:nb
 for i=1:nb
 if i<j && imatrix(i,j) ~= 0
 v(j,j) = v(i,i) - v(i,j);
 end
 end
end
 votage_new = diag(v);
 error = max(abs(voltage_old - votage_new));
 voltage_old = votage_new;
 count = count + 1;
end
count;

%%%%%%%%%%%% LOSS CALCULATION %%%%%%%%%%%%%%%%
linecurrent = zeros(nb-1,1);
impedence = zeros(nb-1,1);
losses = zeros(nb-1,1);
k=1;
for j=2:nb
 for i=1:nb
 if j ~= i && imatrix(i,j) ~= 0
 linecurrent(k) = imatrix(i,j);
```

```
 impedence(k) = zmatrix(i,j);
 losses(k) = ((abs(linecurrent(k)))^2)*(impedence(k))*1000*mvabase;
 end
 end
 k=k+1;
end
totalloss = 0;
for i = 1:size(losses)
 totalloss =(totalloss + losses(i));
end
busvoltage = diag(v);
BUS_VOLTAGES = busvoltage
MINIMUM_VOLTAGE = min(abs(busvoltage))
MINIMUM_VOLTAGE_BUS = find(abs(busvoltage) == MINIMUM_VOLTAGE)
REAL_LOSS = real(totalloss)
REACTIVE_LOSS = imag(totalloss)
figure(1)
plot(abs(busvoltage),'r')
title('Voltage magnitudes of a 69 bus radial distribution network')
xlabel('Bus No.');
ylabel('Bus voltage (in pu)')
figure(2)
bar(abs(losses),'g')
title('Line Losses Vs Line curve of a 69 bus radial distribution network')
xlabel('Line No.');
ylabel(' Line losses (MVA)')
%%%
```

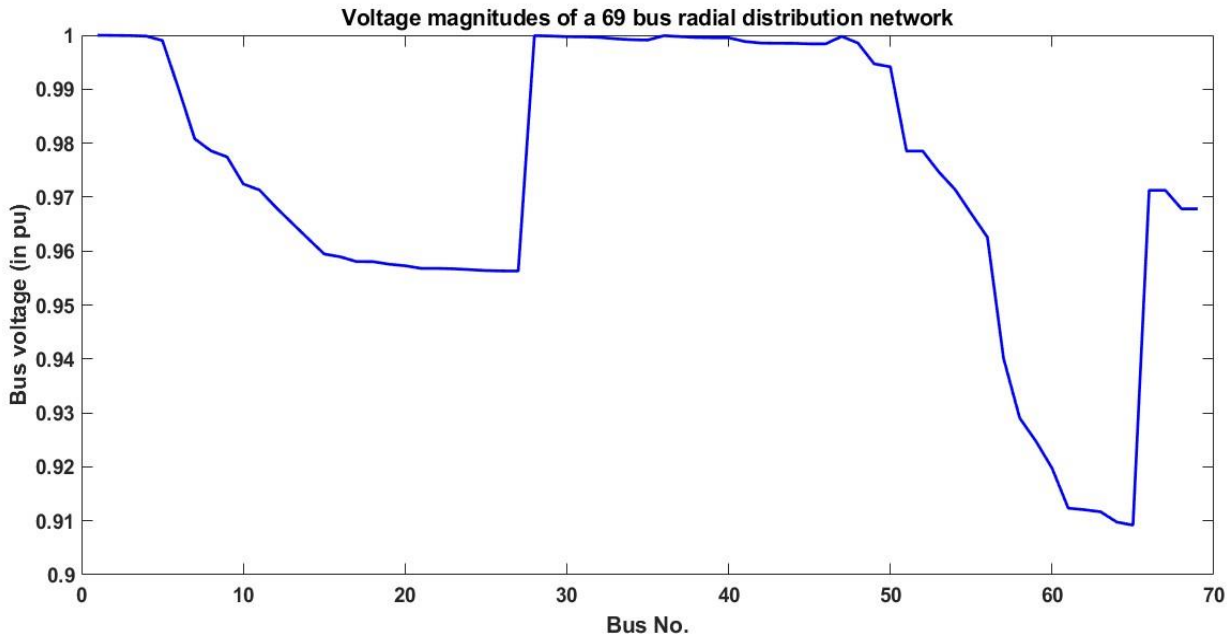
MINIMUM_VOLTAGE = 0.9092

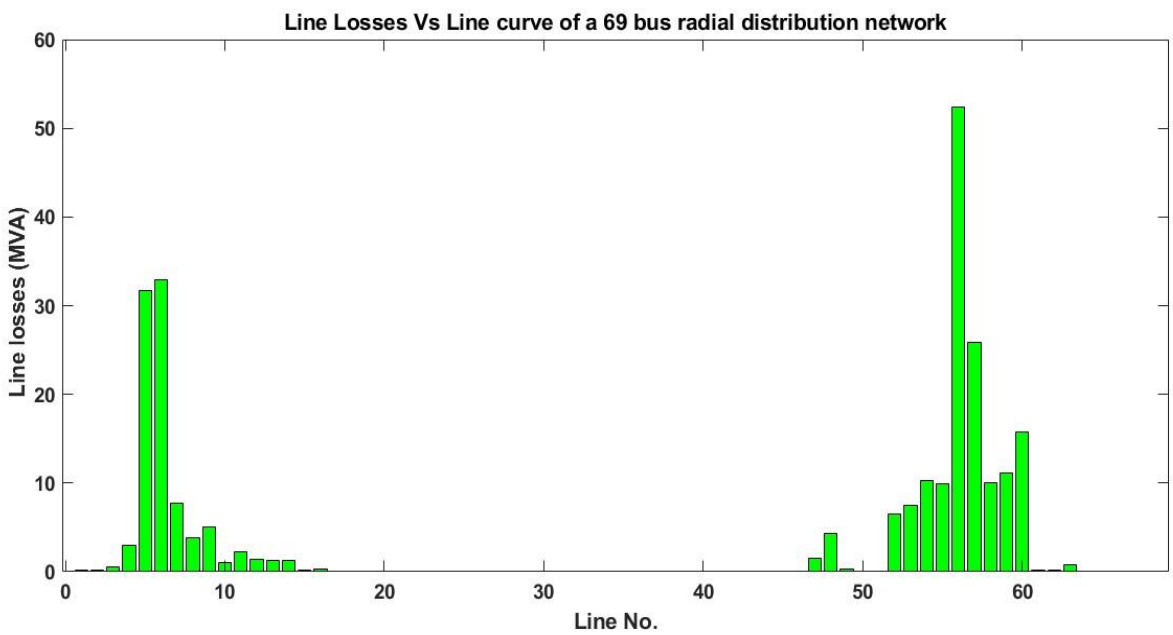
MINIMUM_VOLTAGE_BUS = 65

REAL_LOSS = 224.9722

REACTIVE_LOSS = 102.1948

"Loop ran for 263 times">>





➢ CONCLUSION:

- ✓ The minimum voltage of the network is **0.9092 pu** at **bus no. 65.**

- ✓ The total real loss of the 69-bus radial distribution network comes out to be **224.9722 kW** and the reactive power loss of the distribution network is **102.1948 kVAr.**

➢ THIRD CASE: 52 Bus radial distribution network.

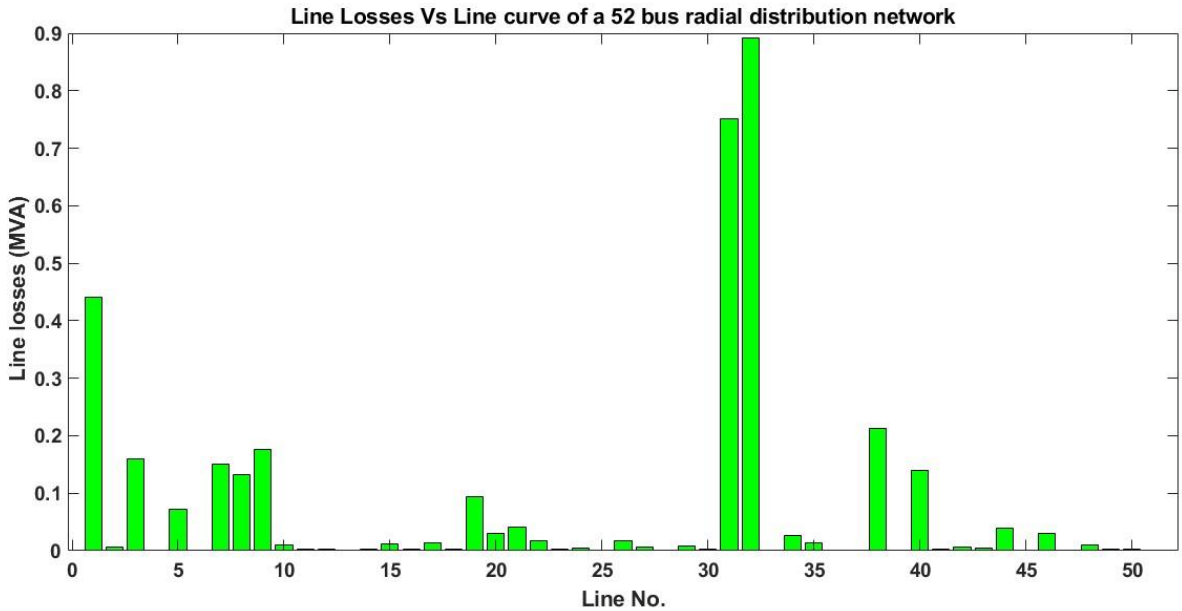
```
%%%%%%%%%%%%%%%%%% BUS DATA OF 52 BUS SYSTEM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% bus no. | bus code | voltage | angle | P | Q
busdata52=[
1 1 1 0 0 0;
2 0 1 0 81 39;
3 0 1 0 135 65;
4 0 1 0 108 52;
5 0 1 0 108 52;
6 0 1 0 27 13;
7 0 1 0 54 26;
8 0 1 0 135 65;
9 0 1 0 81 39;
10 0 1 0 67 32;
11 0 1 0 27 13;
12 0 1 0 27 13;
13 0 1 0 108 52;
14 0 1 0 54 26;
15 0 1 0 94 45;
16 0 1 0 67 33;
17 0 1 0 67 33;
18 0 1 0 108 52;
19 0 1 0 81 39;
20 0 1 0 108 52;
21 0 1 0 94 46;
22 0 1 0 81 39;
23 0 1 0 108 52;
24 0 1 0 108 52;
25 0 1 0 102 50;
26 0 1 0 41 20;
27 0 1 0 108 52;
28 0 1 0 162 79;
29 0 1 0 68 33;
30 0 1 0 68 33;
31 0 1 0 95 46;
32 0 1 0 41 20;
33 0 1 0 121 59;
34 0 1 0 41 20;
35 0 1 0 41 20;
36 0 1 0 135 66;
37 0 1 0 81 40;
38 0 1 0 68 33;
39 0 1 0 95 46;
40 0 1 0 108 52;
41 0 1 0 41 20;
42 0 1 0 95 46;
43 0 1 0 27 13;
44 0 1 0 122 59;
```

```
45 0 1 0 108 52;
46 0 1 0 81 39;
47 0 1 0 68 33;
48 0 1 0 41 20;
49 0 1 0 68 33;
50 0 1 0 81 39;
51 0 1 0 108 52;
52 0 1 0 41 20];
%%%
```

```
43 43 44 0.043 0.0185;
44 41 45 0.0129 0.0056;
45 45 46 0.0301 0.013;
46 45 47 0.0215 0.0093;
47 47 48 0.0129 0.0056;
48 47 49 0.0129 0.0056;
49 49 50 0.0344 0.0148;
50 49 51 0.0129 0.0056;
51 51 52 0.0086 0.0037];
%%%
```

```
%%%%%%%%%% FORWARD SWEEP %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:nb-1
 for j=1:nb
 if j~=i
 v(i,j) = zmatrix(i,j)*imatrix(i,j); % line drop calculation
 end
 end
end
for j=1:nb
 for i=1:nb
 if i<j && imatrix(i,j) ~= 0
 v(j,j) = v(i,i) - v(i,j);
 end
 end
end
 votage_new = diag(v);
 error = max(abs(voltage_old - votage_new));
 voltage_old = votage_new;
 count = count + 1;
end
count;
%%%%%%%%%%%% LOSS CALCULATION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
linecurrent = zeros(nb-1,1);
impedence = zeros(nb-1,1);
losses = zeros(nb-1,1);
k=1;
for j=2:nb
 for i=1:nb
 if j ~= i && imatrix(i,j) ~= 0
 linecurrent(k) = imatrix(i,j);
 impedence(k) = zmatrix(i,j);
 losses(k) = ((abs(linecurrent(k)))^2)*(impedence(k))*1000*mvabase;
 end
 end
 k=k+1;
end
totalloss = 0;
for i = 1:size(losses)
 totalloss =(totalloss + losses(i));
end
busvoltage = diag(v);
BUS_VOLTAGES = busvoltage
MINIMUM_VOLTAGE = min(abs(busvoltage))
MINIMUM_VOLTAGE_BUS = find(abs(busvoltage) == MINIMUM_VOLTAGE)
REAL_LOSS = real(totalloss)
REACTIVE_LOSS = imag(totalloss)
figure(1)
plot(abs(busvoltage),'r')
title('Voltage magnitudes of a 52 bus radial distribution network')
xlabel('Bus No.');
ylabel('Bus voltage (in pu)')
figure(2)
bar(abs(losses),'g')
title('Line Losses Vs Line curve of a 52 bus radial distribution network')
xlabel('Line No.');
```

```
ylabel(' Line losses (MVA)')
%%%
```



➢ CONCLUSION:

- ✓ The minimum voltage of the network is **0.9986 pu** at **bus no. 50.**
- ✓ The total real loss of the 69-bus radial distribution network comes out to be **3.2662 kW** and the reactive power loss of the distribution network is **1.4074 kVAr.**

ASSIGHNMENT NO. # 05

(04/03/23 & 12/04/23)

NAME: SHIVRAJ VISHWAKARMA **ROLL. NO.:** 224102112

OBJECTIVE

To simulate a static VAr compensator in power system using PSCAD

- **SOFTWARE:** PSCAD v45
- **THEORETICAL ANALYSIS**

Static VAr compensators (SVC) are used to prevent annoying voltage flickers caused by industrial loads such as arc furnaces, which cause very rapid changes in the reactive power and also introduce a fluctuating load imbalance between the three phases. Another use for static VAr compensators is to provide a dynamic voltage regulation to enhance the stability of the interconnection between two ac systems. There are primarily three types of static VAr compensators:

1. Thyristors-controlled reactor (TCR)

2. Thyristors-controlled capacitors (TSC)

3. Switching converters with minimum energy storage elements.

The SVC is constructed by means of fixed capacitor and Thyristors-controlled reactor, which is placed in shunt with the load bus as shown in Fig. 1. The firing angles of the TCR are varied for different loading conditions to make the reactive power zero at the source. The TCR comprises a pair of back-to-back Thyristors in series with a linear air-core reactor. Control and firing unit is used to vary the firing angle α by which the start of conduction of the Thyristors pair is delayed.

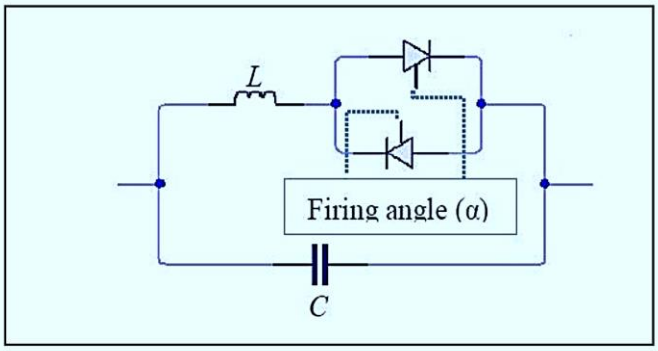


Fig 1. Illustration of SVC

ALGORITHM:

- First, we'll Open PSCAD and create a new case.
- Then we'll Open the main page of our new case.

- we'll Build the circuit to study the operation of SVC in power system as shown in Fig.2.
- The SVC is connected to the power system through a circuit breaker (BRK) that closes after **2.5 seconds** from the start of the simulation.
- Thyristor firing pulses generation and firing angle control are achieved using the blocks as shown in Fig. 3.
  - A reference signal is fed to the VCO (Voltage-controlled-oscillator) function block, which will generate triangular pulse. Through variable input slider value of Alpha (α =firing angle) is selected. Interpolated Firing Pulses is used to turn on the forward Thyristors. The same is repeated for the backward thyristor, except its firing angle Alpha is delayed by 180° as illustrated in Fig. 3.
- Finally, We'll Plot the load voltage in p.u., and the currents at the source, load, capacitor and inductor.
  - The obtained waveforms for α=147° are shown in Results section at the end.

System Parameters	Value
Transmission line resistance and reactance	7.2 ohm and 0.23 H
Load	100 MVA at power factor 0.8 lagging
Voltage source	132.79 kV line-neutral; single phase
Frequency	60 Hz
SVC	0.127 H; 20 μF

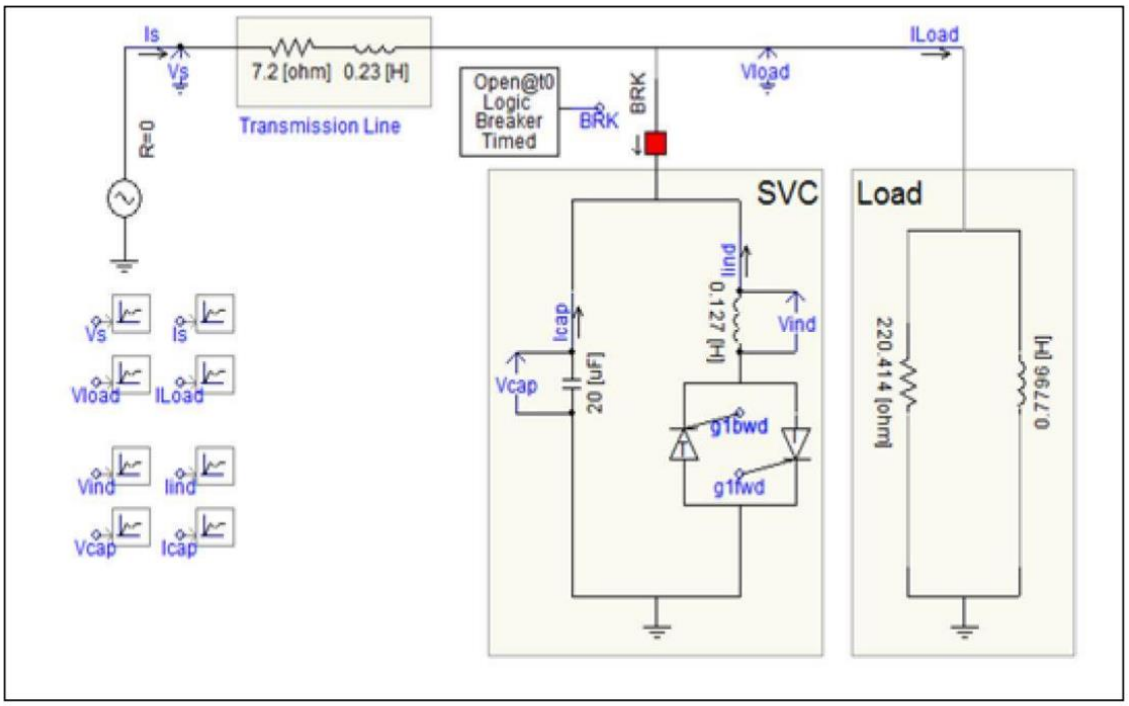


Fig 2. Circuit to study the operation of SVC in power system

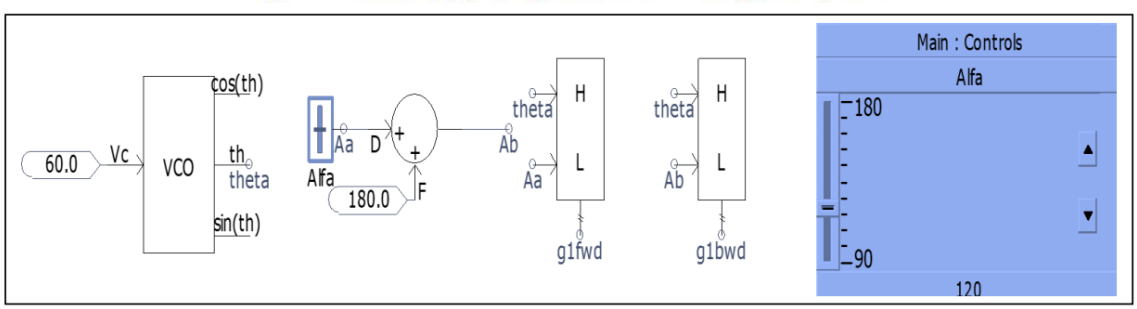


Fig 3. Generation of firing pulses

➢ SIMULATION CIRCUIT:

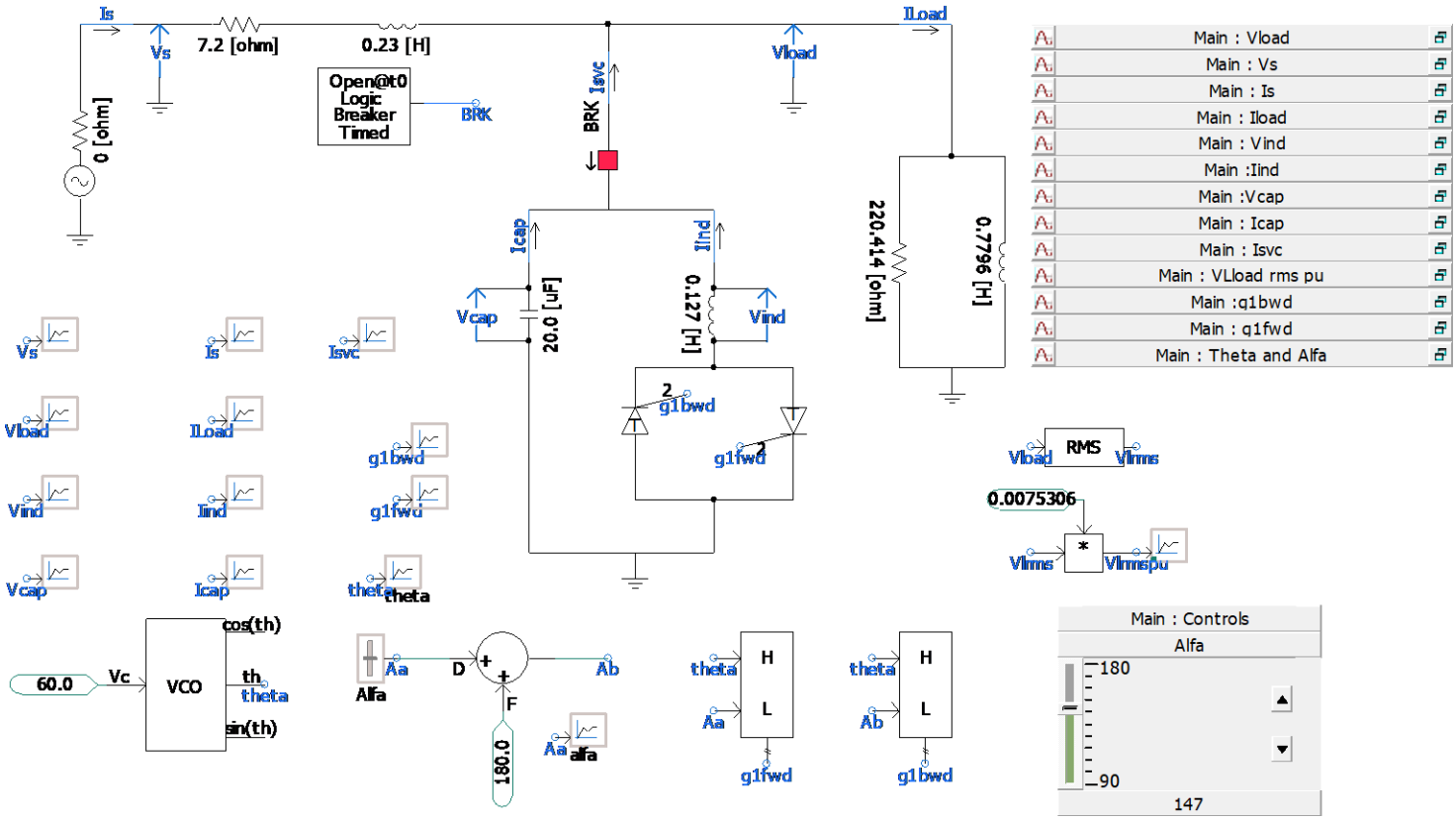


Fig-4: Simulated Circuit.

➢ RESULT OBTAINED:

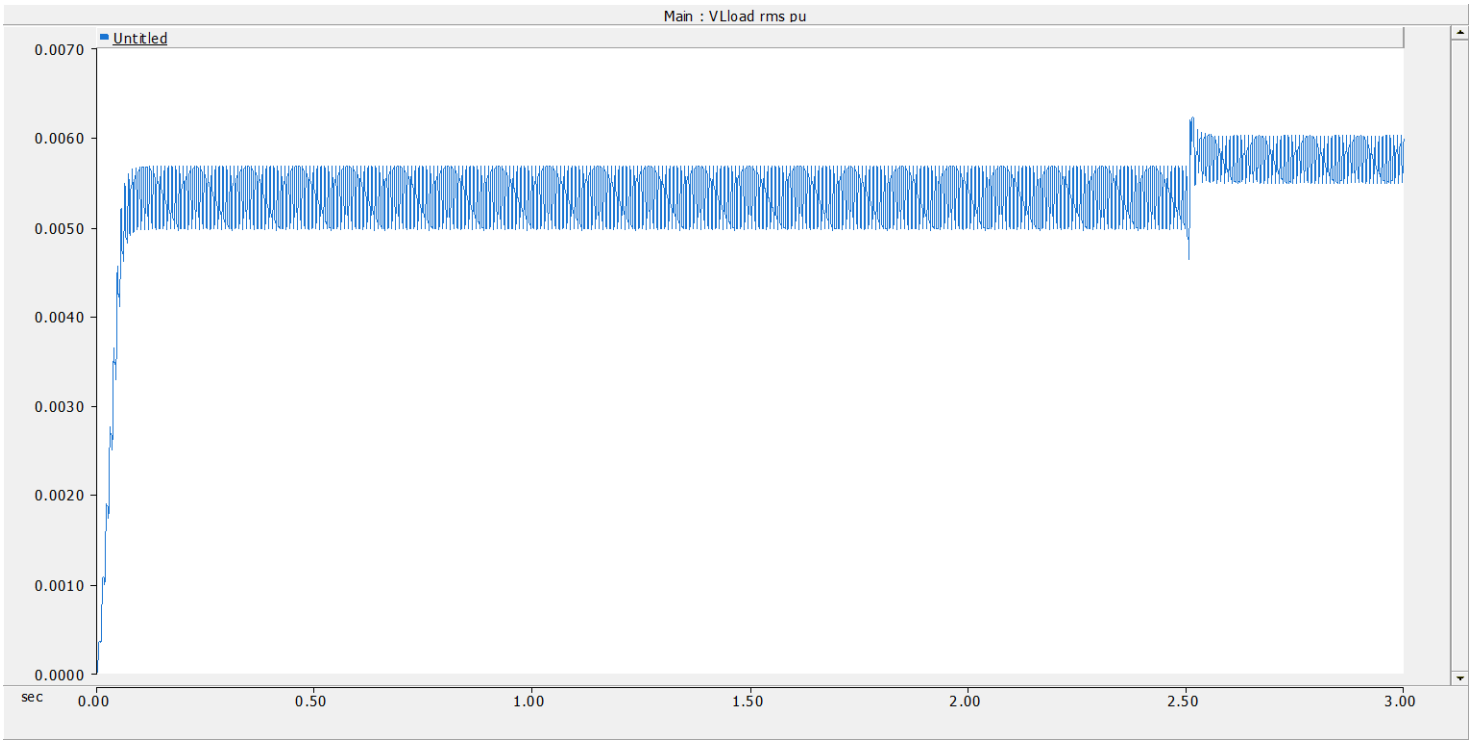


Fig-5: Output Load Voltage (rms Pu).

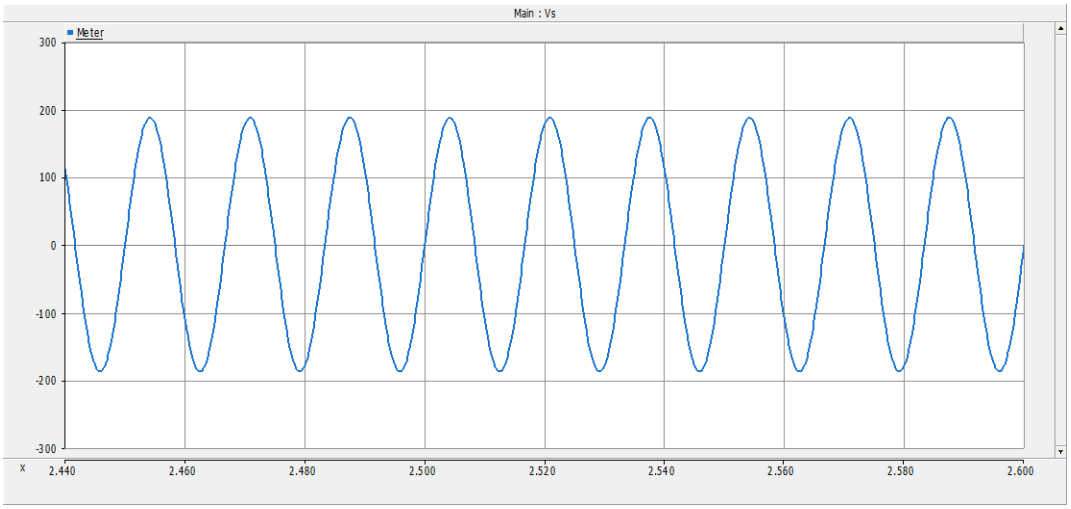


Fig-6: Source Voltage.

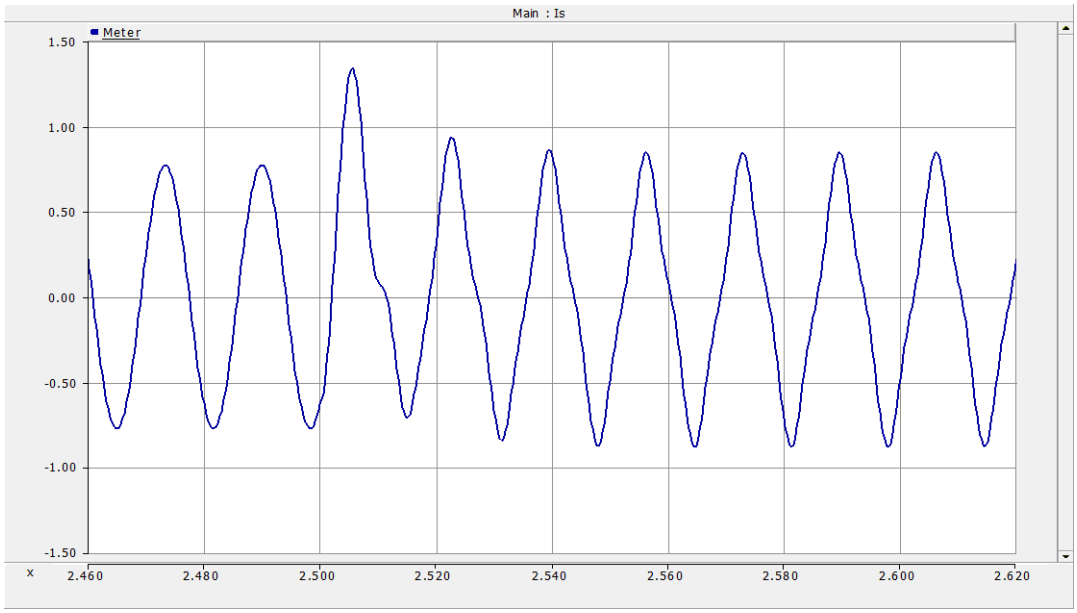


Fig-7: Source Current.

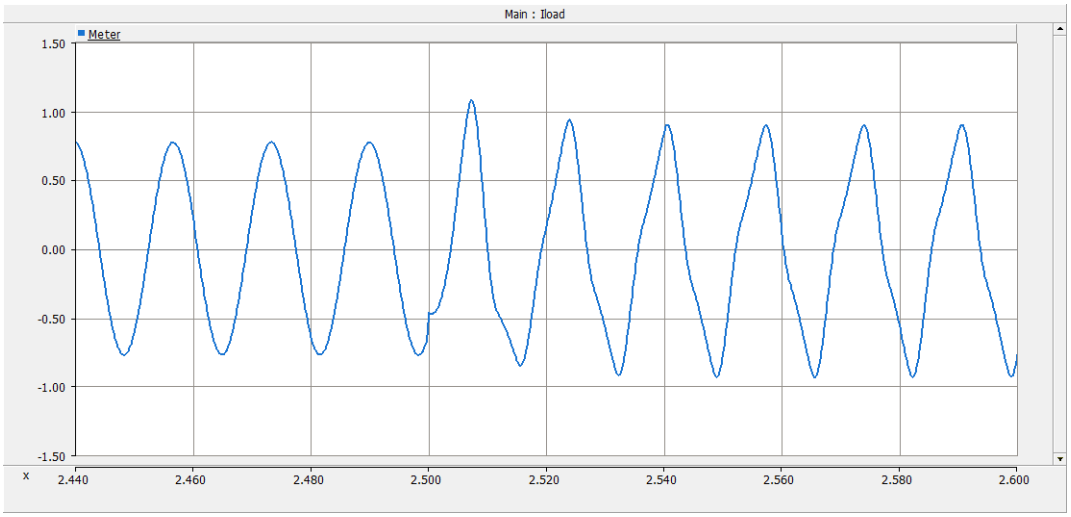


Fig-8: Output Load Current.

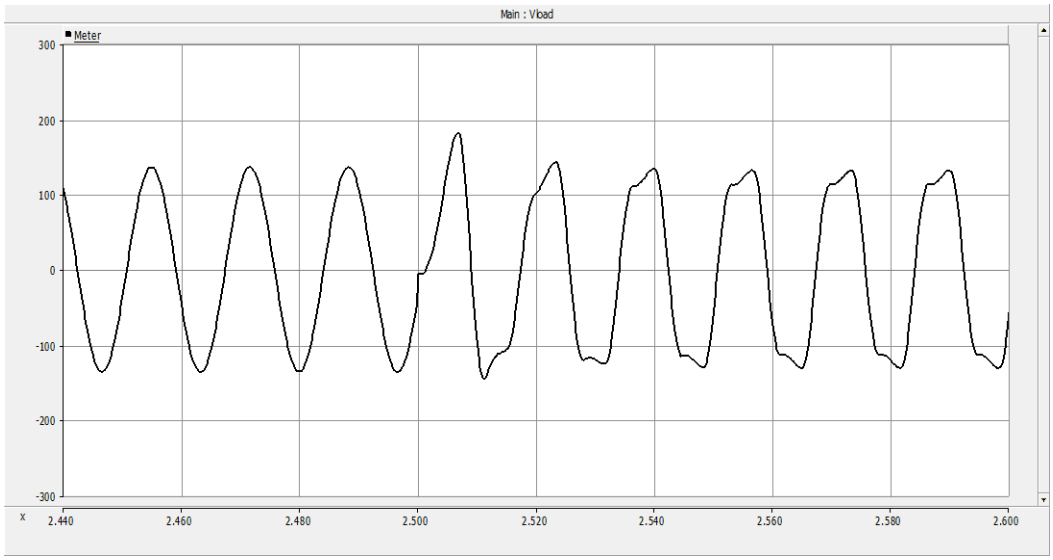


Fig-9: Load Voltage.

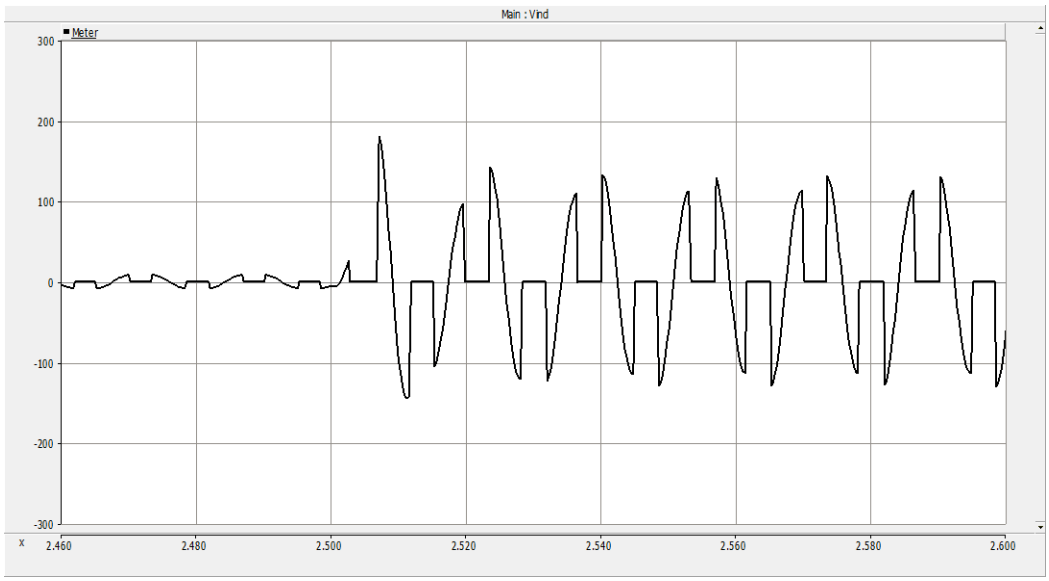


Fig-10: Inductor Voltage.

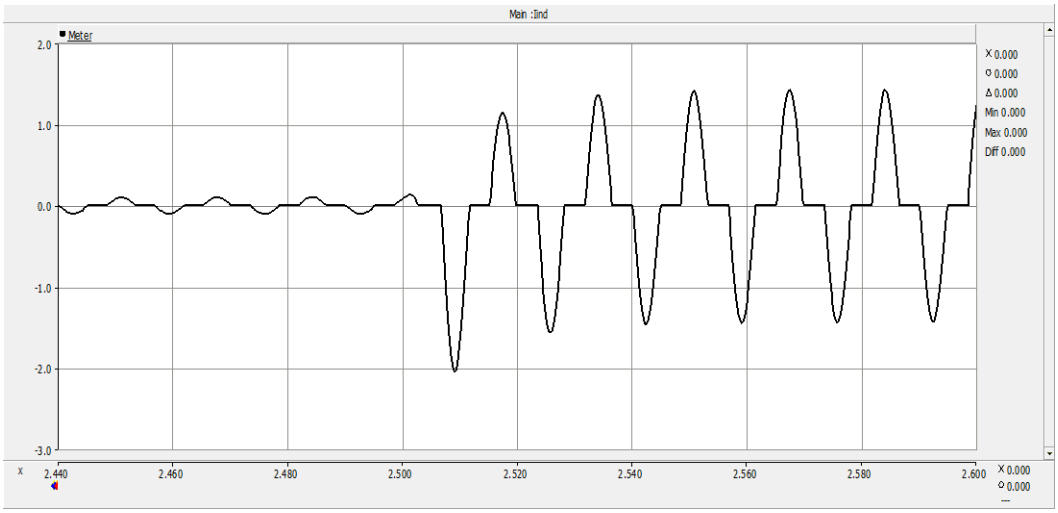


Fig-11: Inductor Current.

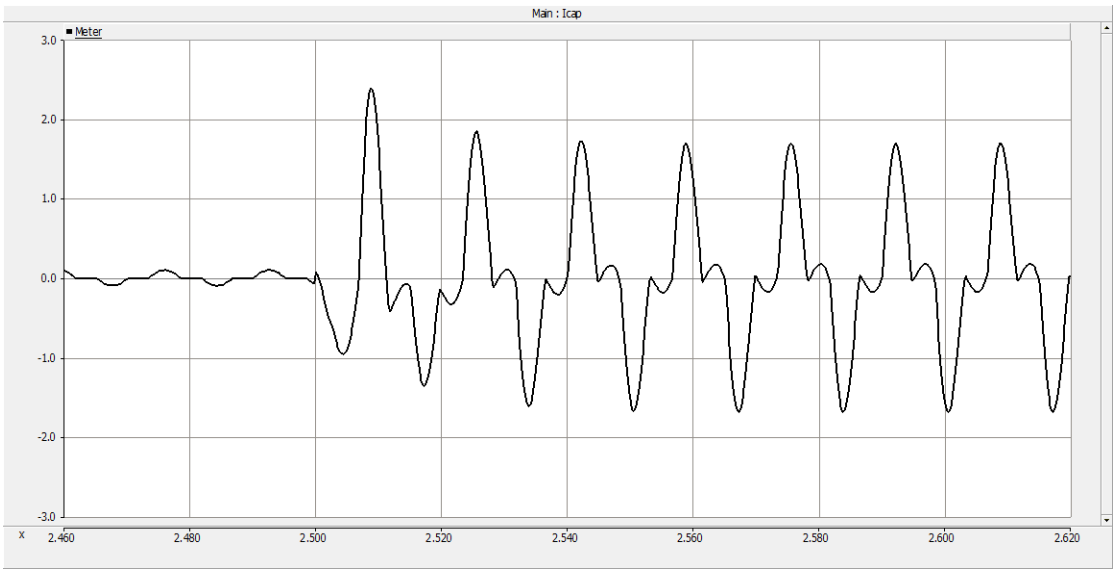


Fig-12: Capacitor Current.

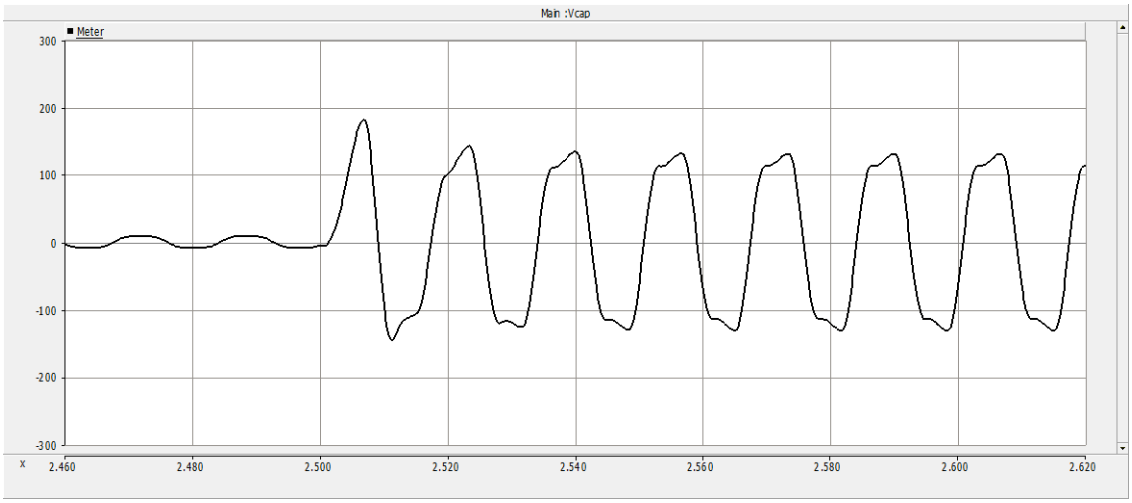


Fig-13: Capacitor Voltage.

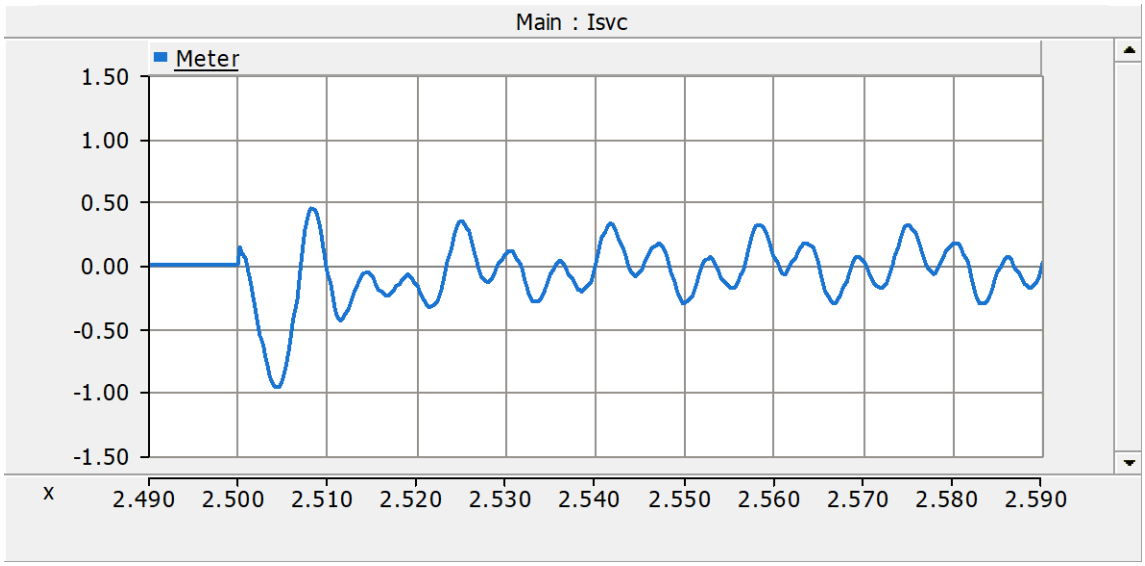


Fig-14: Input Current to SVC.

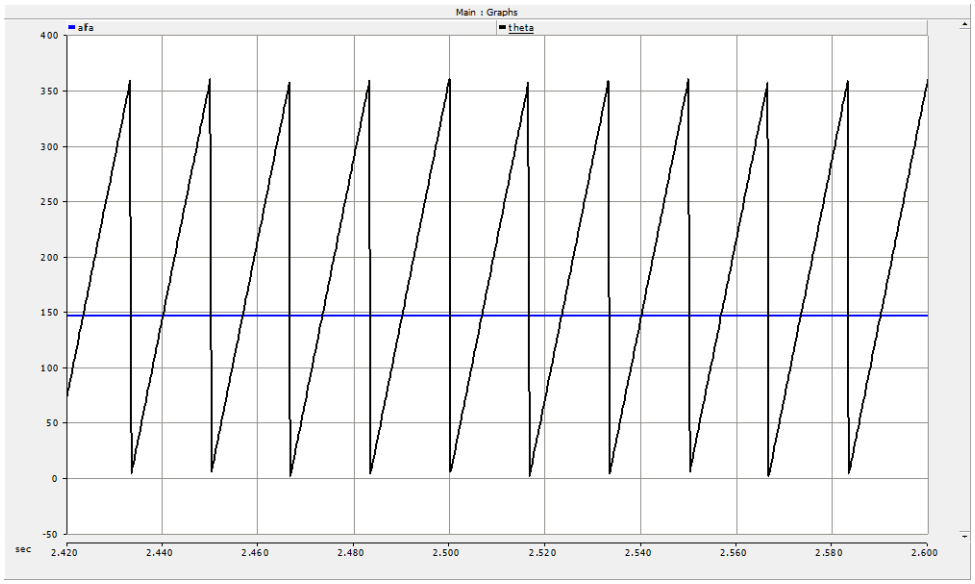


Fig-15: Alpha & theta curve used for firing control circuit.

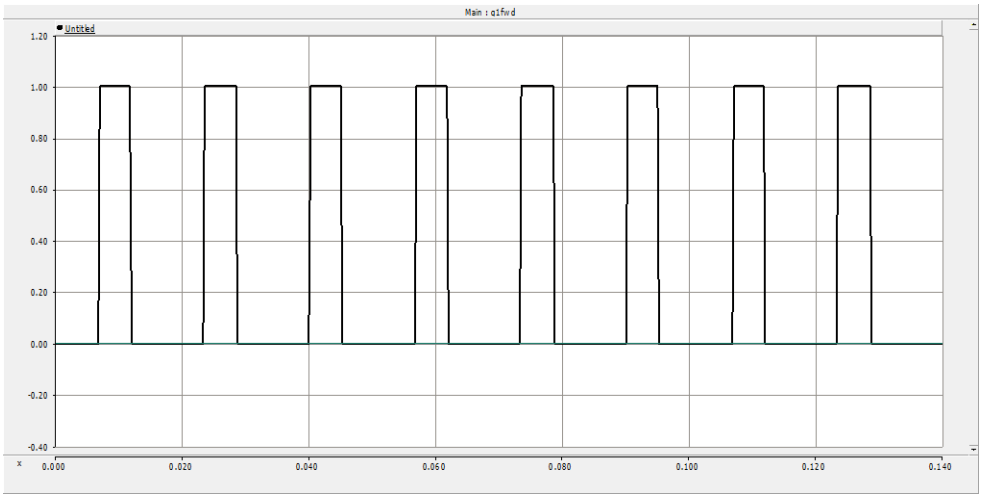


Fig-16: switching scheme of Forword thyristor.

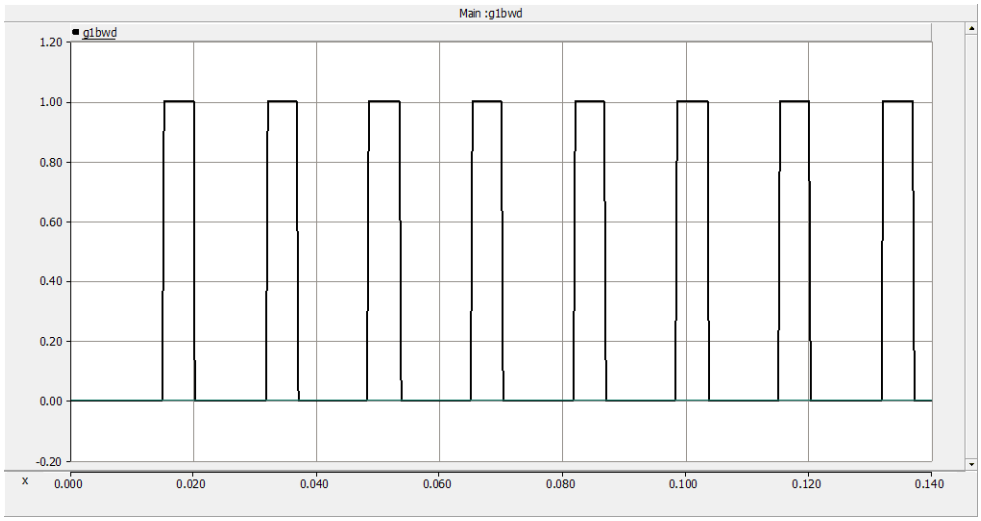


Fig-17: switching scheme of Backward thyristor.

CONCLUSION:

- We have simulated the TCR and studied its operation with the help of PSCAF v45 software with the given specification.
- But at the given value of α =120°, the voltage was getting reduced by some amount instead of increasing as it was shown, which means it was acting as an inductive compensator instead of acting like capacitive compensation.
- The given situation met when we used α = 147° i.e., we found an increment in the voltage at the time of switching at 2.5s, which means it was behaving like capacitive compensation as it was given.
- By seeing these waveforms, we can say that after closing of circuit breakers, SVC gets connected to the system & that's why the output voltage or load voltage had increased. The same is also true for other waveforms.