Laboratory Experiment 4

Power Flow using MATLAB and PowerWorld

Objective: To carry out power flow calculations using *MATLAB* and *PowerWorld* program.

Laboratory Tasks and Report:

1. The MATLAB files to calculate power flow in the example 3-bus power system using both Newton-Raphson and Gauss-Seidel methods are included in this folder (see video clip# 6).
    a. Execute the Newton_Raphson_task_1a.m file and obtain power and reactive power flow through all the transmission lines (both ends) and provided by the generators at buses 1 and 2 using both the methods. Draw a diagram of the three bus system and add the values of the P and Q flows and the P and Q bus injections to the diagram.
    b. Execute the Gauss_seidel_task_1a program. Does it get to the same solution? Why does it take so many more iterations?
    c. The line shunt capacitances have been added as described in Chapter 5, execute the Newton_Raphson_task_1b.m file to obtain power and reactive power flow and compare the results with part (a). Explain all differences.
2. The PowerWorld file to calculate power flow in the example 3-bus power system is **Three_Bus_PowerFlow.pwb** (see video clip# 5), which is included in this folder.
    a. Execute this file and obtain the results to confirm those from the MATLAB programs in step 1.
    b. Comment on the nature of buses 1 and 2.
    c. Add the line shunt capacitances as in part 1(c) and compare results.
    d. Limit the reactive power from generator 2 to be in a range $\pm 200\,MVA$ and see the influence on the bus 2 voltage and power flows on the lines.
    e. Raise the power delivered to bus 3 by clicking on the up arrow next to the real power displayed. How much can you deliver until voltage drops to 0.95 pu? How much can you deliver until it drops to 0.90 pu?

<u>**Newton Raphson Power Flow Program**</u>

```
% Example 5-4 Power Flow in a 3-bus Test System
clear
j = sqrt(-1);

V = zeros(3,1);
S = zeros(3,1);
Mismatch = zeros(3,1);
```

```
% ---------- Input line impedances ------------- %

Z = [0 0.0047 + 0.0474i 0.0062 + 0.0632i
     0.0047 + 0.0474i 0 0.0047 + 0.0474i
     0.0062 + 0.0632i 0.0047 + 0.0474i 0];

%---------Base Values ---------%
kVLL=345;
MVA3Ph=100;
Zbase=kVLL^2/MVA3Ph;

XL_km=0.376;                        % ohm/km at 60 Hz
RL_km= 0.037; B_km=4.5;             % B in micro-mho/km

%---------Line Susceptances--------%

B13_Micro_Mho=4.5*200;               %200 km long
B12_Micro_Mho=4.5*150;               %150 km long
B23_Micro_Mho=4.5*150;               %150 km long

%---------Line impedances-----------%
Z13_ohm=(RL_km+j*XL_km)*200;         %200 km long
Z12_ohm=(RL_km+j*XL_km)*150;         %150 km long
Z23_ohm=(RL_km+j*XL_km)*150;         %150 km long

%------- line impedances in per unit--------%
Z13=Z13_ohm/Zbase;
Z12=Z12_ohm/Zbase;
Z23=Z23_ohm/Zbase;

%-------- susceptances in per unit----------%
B13=B13_Micro_Mho*Zbase*10^-6;
B12=B12_Micro_Mho*Zbase*10^-6;
B23=B23_Micro_Mho*Zbase*10^-6;

%---------- YBUS Creation------------%
Y(1,1)=1/Z12 + 1/Z13;
Y(1,2)=-1/Z12;
Y(1,3)=-1/Z13;
Y(2,1)=-1/Z12;
Y(2,2)=1/Z12 + 1/Z23;
Y(2,3)=-1/Z23;
Y(3,1)=-1/Z13;
Y(3,2)=-1/Z23;
Y(3,3)=1/Z13 + 1/Z23;
Y                % Print Y=G+jB Admittance Matrix

%----------Conductance Values-----------%
G(1,1)=real(Y(1,1));
G(1,2)=real(Y(1,2));
G(1,3)=real(Y(1,3));
G(2,1)=real(Y(2,1));
G(2,2)=real(Y(2,2));
G(2,3)=real(Y(2,3));
G(3,1)=real(Y(3,1));
G(3,2)=real(Y(3,2));
G(3,3)=real(Y(3,3));
```



```
%--------Susceptance Values----------%
B(1,1)=imag(Y(1,1));
B(1,2)=imag(Y(1,2));
B(1,3)=imag(Y(1,3));
B(2,1)=imag(Y(2,1));
B(2,2)=imag(Y(2,2));
B(2,3)=imag(Y(2,3));
B(3,1)=imag(Y(3,1));
B(3,2)=imag(Y(3,2));
B(3,3)=imag(Y(3,3));

%--------- Given Specifications in pu----------%
V1MAG=1.0;
ANG1=0;
V2MAG=1.05;
P2sp=2.0;
P3sp=-5.0;
Q3sp=-1.0;

% -------------Calulate ANG2, V3MAG and ANG3---------------%

% ----Solution Parameters----%
Tolerance= 0.001;
Iter_Max=10;

%----- Initialization-------%
Iter=0;
i=0;
ConvFlag=1;
ANG2=0;
ANG3=0;
V3MAG=1.0;
delANG2=0;
delANG3=0;
delMAG3=0;


%------------------ Start Iteration Process for N-R-----------------%
while( ConvFlag==1 & Iter < Iter_Max)
    Iter=Iter+1;
    i=i+1;
ANG2=ANG2+delANG2;
ANG3=ANG3+delANG3;
V3MAG=V3MAG+delMAG3;

%------- Creation of Jacobian J--------%
% J(1,1)=dP2/dAng2; Eq. 9-26; k=2, m=1,3
J(1,1)=V2MAG*(V1MAG*(-G(2,1)*sin(ANG2-ANG1)+B(2,1)*cos(ANG2-ANG1)) +
V3MAG*(-G(2,3)*sin(ANG2-ANG3)+B(2,3)*cos(ANG2-ANG3)));

% J(1,2)=dP2/dAng3; Eq. 9-27; k=2, j=3
J(1,2)=V2MAG*(V3MAG*(G(2,3)*sin(ANG2-ANG3)-B(2,3)*cos(ANG2-ANG3)));

% J(1,3)=dP2/dMAG3; Eq. 9-29; k=2, j=3
J(1,3)=V2MAG*((G(2,3)*cos(ANG2-ANG3)+B(2,3)*sin(ANG2-ANG3)));
```

```
% J(2,1)=dP3/dAng2; Eq. 9-27; k=3, j=2
J(2,1)=V3MAG*(V2MAG*(G(3,2)*sin(ANG3-ANG2)-B(3,2)*cos(ANG3-ANG2)));

% J(2,2)=dP3/dAng3; Eq. 9-26; k=3, m=1,2
J(2,2)=V3MAG*(V1MAG*(-G(3,1)*sin(ANG3-ANG1)+B(3,1)*cos(ANG3-ANG1)) +
V2MAG*(-G(3,2)*sin(ANG3-ANG2)+B(3,2)*cos(ANG3-ANG2)));

% J(2,3)=dP3/dMAG3; Eq. 9-28; k=3, m=1,2
J(2,3)=2*G(3,3)*V3MAG + V1MAG*(G(3,1)*cos(ANG3-ANG1)+B(3,1)*sin(ANG3-
ANG1)) + V2MAG*(G(3,2)*cos(ANG3-ANG2)+B(3,2)*sin(ANG3-ANG2));

% J(3,1)=dQ3/dAng2; Eq. 9-31; k=3, j=2
J(3,1)=V3MAG*(V2MAG*(G(3,2)*cos(ANG3-ANG2)-B(3,2)*sin(ANG3-ANG2)));

% J(3,2)=dQ3/dAng3; Eq. 9-30; k=3, m=1,2
J(3,2)=V3MAG*(V1MAG*(G(3,1)*cos(ANG3-ANG1)+B(3,1)*sin(ANG3-ANG1)) +
V2MAG*(G(3,2)*cos(ANG3-ANG2)+B(3,2)*sin(ANG3-ANG2)));

% J(3,3)=dQ3/dMAG3; Eq. 9-32; k=3, m=1,2
J(3,3)=- 2*B(3,3)*V3MAG + V1MAG*(G(3,1)*sin(ANG3-ANG1)-B(3,1)*cos(ANG3-
ANG1))  + V2MAG*(G(3,2)*sin(ANG3-ANG2)-B(3,2)*cos(ANG3-ANG2));

% ---------Bus Voltages--------%
V(1)=V1MAG*exp(j*ANG1);
V(2)=V2MAG*exp(j*ANG2);
V(3)=V3MAG*exp(j*ANG3);

% -------Injected currents into Buses-------%
Iinj=Y*V;

%------- P and Q Injected into Buses--------%
S(1)=V(1)*conj(Iinj(1));
S(2)=V(2)*conj(Iinj(2));
S(3)=V(3)*conj(Iinj(3));

% -----Mismatch at PQ and PV buses-------%
Mismatch(1)=P2sp-real(S(2));
Mismatch(2)=P3sp-real(S(3));
Mismatch(3)=Q3sp-imag(S(3));

% -------calculate new delta values for ANG2, ANG3, and MAG3------&
del=inv(J)*Mismatch;
delANG2=del(1);
delANG3=del(2);
delMAG3=del(3);

% -------Calculate Power Flow on the Transmission Lines-------%
P12=real(V(1)*conj((V(1)-V(2))/Z12));    Q12=imag(V(1)*conj((V(1)-
V(2))/Z12)); % at Bus 1
P13=real(V(1)*conj((V(1)-V(3))/Z13));    Q13=imag(V(1)*conj((V(1)-
V(3))/Z13)); % at Bus 1

P21=real(V(2)*conj((V(2)-V(1))/Z12));    Q21=imag(V(2)*conj((V(2)-
V(1))/Z12)); % at Bus 2
P23=real(V(2)*conj((V(2)-V(3))/Z23));    Q23=imag(V(2)*conj((V(2)-
V(3))/Z23)); % at Bus 2
```

```
P31=real(V(3)*conj((V(3)-V(1))/Z13));    Q31=imag(V(3)*conj((V(3)-
V(1))/Z13)); % at Bus 3
P32=real(V(3)*conj((V(3)-V(2))/Z23));    Q32=imag(V(3)*conj((V(3)-
V(2))/Z23)); % at Bus 3

P1=real(S(1));
Q1=imag(S(1));
P2=real(S(2));
Q2=imag(S(2));
P3=real(S(3));
Q3=imag(S(3));

%--------Display voltage and P,Q values at each iteration-------%

fprintf('\n %s %2d %s \n', 'Iter ',Iter, '  Mismatch 1   Mismatch 2
Mismatch 3 ')

fprintf('             %7.4f        %7.4f      %7.4f \n', Mismatch(1),
Mismatch(2), Mismatch(3) );


 if max(abs(Mismatch)) > Tolerance,
     ConvFlag=1;
 else
     ConvFlag=0;
 end
end

J,                                              %  Print Final Jacobian
Matrix
ANG2DEG=ANG2*180/pi,
ANG3DEG=ANG3*180/pi,
V3MAG,




%-------Display Power Flows on the buses---------%

fprintf('\n %s','Flows on Line 1-2')
fprintf('\n  %s \n', ' P12        Q12         P21         Q21 ')
fprintf('   %7.4f   %7.4f   %7.4f   %7.4f  \n',  P12, Q12 ,P21, Q21)

fprintf('\n %s','Flows on Line 1-3')
fprintf('\n  %s \n', ' P13        Q13         P31         Q31 ')
fprintf(' %7.4f   %7.4f   %7.4f   %7.4f  \n', P13, Q13 ,P31, Q31)

fprintf('\n %s','Flows on Line 2-3')
fprintf('\n  %s \n',  ' P23        Q23         P32         Q32 ')
fprintf('  %7.4f   %7.4f   %7.4f   %7.4f  \n',  P23, Q23 ,P32, Q32)

fprintf('\n %s','Net P,Q at Buses')
fprintf('\n %s \n', '  P1          Q1          P2          Q2
P3          Q3  ')
fprintf('  %7.4f    %7.4f    %7.4f    %7.4f     %7.4f     %7.4f    \n',
P1, Q1, P2, Q2, P3, Q3)
```

Gauss-Seidel Power Flow Program

```
Iter = 0;
Maxiter = 20;
converge = 0;
tolerance  = 0.001;
j = sqrt(-1);

% ---------- Input line impedances ------------- %

Z = [0 0.0047 + 0.0474i 0.0062 + 0.0632i
     0.0047 + 0.0474i 0 0.0047 + 0.0474i
     0.0062 + 0.0632i 0.0047 + 0.0474i 0];

% ---------- Base Values ----------- %

kVLL=345;
MVA3Ph=100;
Zbase=kVLL^2/MVA3Ph;

XL_km=0.376;                                % ohm/km at 60 Hz
RL_km= 0.037; B_km=4.5;                     % B in micro-mho/km

Z13_ohm=(RL_km+j*XL_km)*200;                %200 km long
B13_Micro_Mho=4.5*200;                      %200 km long
Z12_ohm=(RL_km+j*XL_km)*150;                %150 km long
B12_Micro_Mho=4.5*150;                      %150 km long
Z23_ohm=(RL_km+j*XL_km)*150;                %150 km long
B23_Micro_Mho=4.5*150;                      %150 km long

%--------- line impedances in per unit ----------%

Z13=Z13_ohm/Zbase;
Z12=Z12_ohm/Zbase;
Z23=Z23_ohm/Zbase;

%--------- susceptances in per unit ----------%

B13=B13_Micro_Mho*Zbase*10^-6;
B12=B12_Micro_Mho*Zbase*10^-6;
B23=B23_Micro_Mho*Zbase*10^-6;

%-------Finding the admittance matrix---------%

Y(1,1)=1/Z12 + 1/Z13;
Y(1,2)=-1/Z12;
Y(1,3)=-1/Z13;
Y(2,1)=-1/Z12;
Y(2,2)=1/Z12 + 1/Z23;
Y(2,3)=-1/Z23;
Y(3,1)=-1/Z13;
Y(3,2)=-1/Z23;
Y(3,3)=1/Z13 + 1/Z23;
```

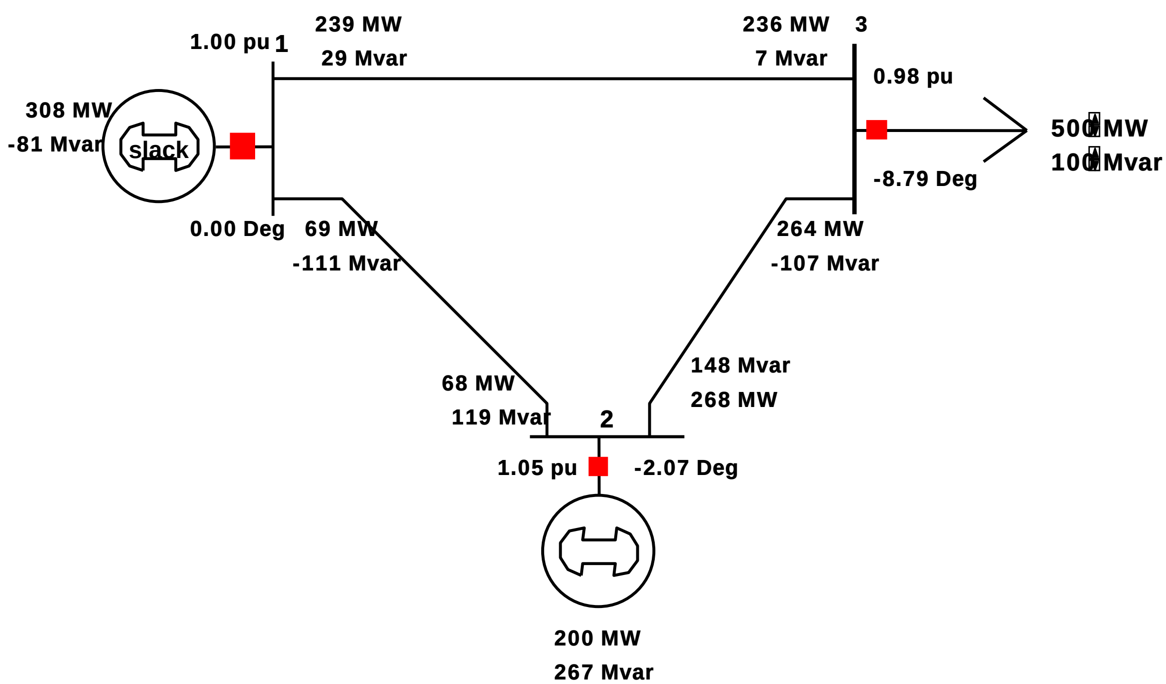
```
G = real(Y);
B = imag(Y);

% ------- Initialize line variables --------- %

I = zeros(1,length(Z));
P = zeros(1,length(Z));
Q = zeros(1,length(Z));
S = zeros(1,length(Z));

% -------- Input bus parameters --------- %

v = ones(1,length(Z));
%angle = zeros(length(Z),1);
Pload = zeros(1,length(Z));
Qload = zeros(1,length(Z));
Psched = zeros(1,length(Z));
Qsched = zeros(1,length(Z));

Bustype(1) = 'S';                            % PQ bus
Bustype(2) = 'G';                            % PV bus
Bustype(3) = 'L';                            % PQ bus

v(2) = 1.05;
Pload(2) = -2;
Pload(3) = 5;
Qload(3) = 1;
% ------ Set up power schedule --------- %

totload = 0.0;
for i = 1:length(Z)
   Psched(i)= -Pload(i);
   Qsched(i)= -Qload(i);
   totload = totload + Pload(i);
end

% --- Calculate P and Q at each bus ----- %

for i = 1:length(Z)
   I(i) = 0.0 + sqrt(-1)*0.0;
   for j = 1:length(Z)
      I(i) = I(i) + Y(i,j)*v(j);
   end
   S(i) = v(i) * conj(I(i));
   P(i) = real(S(i));
   Q(i) = imag(S(i));
end

% ------- Display bus voltages ---------- %

fprintf('\n %s %2d %s \n', 'Iter ',Iter, '  Vreal   Vimag   Vmag
Vangle     P      Q ')
for i = 1:length(Z)
   vitermag = abs(v(i));
   viterang = (180./pi)*atan2(imag(v(i)),real(v(i)));
   fprintf(' %s %2d %7.4f %7.4f %7.4f %7.4f %7.4f %7.4f \n',' Bus ',i,
real(v(i)), imag(v(i)), vitermag, viterang, P(i), Q(i) )
```



```
end

while converge == 0
    Iter = Iter + 1;
    MAXDP = 0.0;
    MAXDPbus = 0;
    MAXDQ = 0.0;
    MAXDQbus = 0;

    for i = 1 : length(Z)
%---------calculate net P and Q at bus i--------%
        I(i) = 0.0 + sqrt(-1)*0.0;
        for j = 1:length(Z)
            I(i) = I(i) + Y(i,j)*v(j);
        end
        S(i) = v(i) * conj(I(i));
        P(i) = real(S(i));
        Q(i) = imag(S(i));

        if Bustype(i) == 'G'
            Qsched(i) = Q(i);
        end
        deltap(i) = abs(P(i) - Psched(i));
        deltaq(i) = 0.0;

        if Bustype(i) == 'L'
            deltaq(i) = abs(Q(i) - Qsched(i));
        end

        if Bustype(i) == 'S'
            deltap(i) = 0.0;
            deltaq(i) = 0.0;
        end

        if Bustype(i) ~= 'S'
            if deltap(i) > MAXDP
                MAXDP = deltap(i);
                MAXDPbus = i;
            end

            if deltaq(i) > MAXDQ
                MAXDQ = deltaq(i);
                MAXDQbus = i;
            end

%--------- Y * V for row i of Y matrix without Yii term---------%

            sum = 0.0;
            for j = 1:length(Z)
                if j ~= i
                    sum = sum + Y(i,j)*v(j);
                end
            end
            vnew = (1.0/Y(i,i))*( ((Psched(i) - sqrt(-
1)*Qsched(i))/(conj(v(i)))) - sum);
            v(i) = vnew;
        end
```

```
    end

%------------ Print and save result from last iteration-----------%

%-----calculate net P and Q at bus i-----%

    for i = 1:length(Z)
        I(i) = 0.0 + sqrt(-1)*0.0;
        for j = 1:length(Z)
            I(i) = I(i) + Y(i,j)*v(j);
        end
        S(i) = v(i) * conj(I(i));
        P(i) = real(S(i));
        Q(i) = imag(S(i));

        if Bustype(i) == 'G'
            Qsched(i) = Q(i);
        end
    end

    fprintf('\n %s %2d %s \n', 'Iter ',Iter, '  Vreal   Vimag   Vmag
Vangle     P      Q ')
    for i = 1:length(Z)
        vitermag = abs(v(i));
        viterang = (180./pi)*atan2(imag(v(i)),real(v(i)));
        fprintf(' %s %2d %7.4f %7.4f %7.4f %7.4f %7.4f %7.4f \n',' Bus
',i, real(v(i)), imag(v(i)), vitermag, viterang, P(i), Q(i) )
    end

% -------- check for convergence ---------- %

    if MAXDP < tolerance
        if MAXDQ < tolerance
            converge = 1;
        end
    end

    if Iter > Maxiter
        converge = 1;
    end
end
```

3Bus_PowerFlow.pwb



Problem 5-8
Confirm the MATLAB Results of
Example 5-4.