ภาคผนวก

โปรมแกรมการหาค่า Bus Admittance Matrix

```
clc;clear;
%Line data
%          Bus   Bus   R         X         1/2 B    Transformer data
%          nl    nr    pu        pu        pu       Tap setting
linedata=[ 1     2     0.01938   0.05917   0.0264   1
           1     5     0.05403   0.22304   0.0246   1
           2     3     0.04699   0.19797   0.0219   1
           2     4     0.05811   0.17632   0.0187   1
           2     5     0.05695   0.17388   0.0170   1
           3     4     0.06701   0.17103   0.0173   1
           4     5     0.01355   0.04211   0.0064   1
           4     7     0         0.20912   0        0.978
           4     9     0         0.55618   0        0.969
           5     6     0         0.25202   0        0.932
           6     11    0.09498   0.19890   0        1
           6     12    0.12291   0.25581   0        1
           6     13    0.01655   0.13027   0        1
           7     8     0         0.17615   0        1
           7     9     0         0.11001   0        1
           9     10    0.03181   0.08450   0        1
           9     14    0.12711   0.27038   0        1
           10    11    0.08205   0.19207   0        1
           12    13    0.22092   0.19988   0        1
           13    14    0.17093   0.34802   0        1  ];
nl=linedata(:,1); nr=linedata(:,2); R=linedata(:,3); X=linedata(:,4); BSH=linedata(:,5);
TS=linedata(:,6); nbr=length(nl); nbus=max(max(nl),max(nr));
 Z = R + j*X; y= ones(nbr,1)./Z;      %branch admittance
for n = 1:nbr
if TS(n) <= 0
```

```
   TS(n) = 1;
 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)/TS(k);
     Ybus(nr(k),nl(k))=Ybus(nl(k),nr(k));
   end
 end
         % formation of the diagonal element
 for  n=1:nbus
    for k=1:nbr
       if nl(k)==n
       Ybus(n,n) = Ybus(n,n)+y(k)/(TS(k)^2) + BSH(k)*i;
       elseif nr(k)==n
       Ybus(n,n) = Ybus(n,n)+y(k) +BSH(k)*i;
       else, end
    end
 end
 Ybus(9,9)=Ybus(9,9)+0.19*i;
 Ybus;
 Yb=abs(Ybus);
 Ang=angle(Ybus);
 real(Ybus);
 imag(Ybus);
 G=real(Ybus);
 B=imag(Ybus);
```

โปรมแกรม MATLAB คำนวณค่า Load Flow

```
basemva=100; accuraey=0.001; accel=1.8; maxiter=100;
load ybus.dat.m
% Initial bus voltages and scheduled bus power
%                 Voltage      Demand        Generation    QG limits    Injection
%       bus type  p.u.|V| deg  p.u.PD p.u.QD  p.u.PG p.u.QG  min   max    Mvar
busdata=[ 1   1   1.060  0.00  0.00   0.00   0.00   0.00   0.00  0.00   0
          2   2   1.045  0.00  21.70  12.7   40.0   0.00   -999  999    0
          3   2   1.010  0.00  94.2   19.0   0.00   0.00   -999  999    0
          4   0   1.000  0.00  47.8   3.90   0.00   0.00   0.00  0.00   0
          5   0   1.000  0.00  7.60   1.80   0.00   0.00   0.00  0.00   0
          6   2   1.070  0.00  11.2   7.50   0.00   0.00   -999  999    0
          7   0   1.000  0.00  0.00   0.00   0.00   0.00   0.00  0.00   0
          8   2   1.090  0.00  0.00   0.00   0.00   0.00   -999  999    0
          9   0   1.000  0.00  29.5   16.6   0.00   0.00   0.00  0.00   0
          10  0   1.000  0.00  9.00   5.80   0.00   0.00   0.00  0.00   0
          11  0   1.000  0.00  3.50   1.80   0.00   0.00   0.00  0.00   0
          12  0   1.000  0.00  6.10   1.60   0.00   0.00   0.00  0.00   0
          13  0   1.000  0.00  13.5   5.80   0.00   0.00   0.00  0.00   0
          14  0   1.000  0.00  14.9   5.60   0.00   0.00   0.00  0.00   0
  ];
ns=0; ng=0; Vm=0; delta=0; yload=0; deltad=0;
nbus = length(busdata(:,1));bl=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)*(eos(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;cnd % the spccified 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
 Vm;
 delta;
  maxerror=max(abs(DC));
   if iter == maxiter & maxerror > accuracy
```

```
  fprintf('\nWARNING: Iterative solution did not converged after ')
  fprintf('%g', iter), fprintf(' iterations.\n\n')
  fprintf('Press Enter to terminate the iterations and print the results \n')
  converge = 0; pause, else, end
  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);
   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);
 end
yload(n) = (Pd(n)- j*Qd(n)+j*Qsh(n))/(basemva*Vm(n)^2);
end
Ps=Pg-Pd;
Qs=Qg-Qd;
```

```
Table1=[bl';Vm*230;delta;Pg;Qg;Pd;Qd];
disp('--------------------------------------------------')
disp('BUS  V(pu)  delta|V|   Pgen   Qgen  Pload   Qload ')
disp('--------------------------------------------------')
fprintf('%2.0f %2.4f %2.4f  %4.2f %4.2f %4.2f  %4.2f\n',Table1)
disp('--------------------------------------------------')

for k=1:nbr
   i=nl(k);
   j=nr(k);
   Pij(k)=funcPij(Vm(i),Vm(j),delta(i),delta(j),Yb(i,j),Ang(i,j),G(i,j));
end
for k=1:nbr
   i=nl(k);
   j=nr(k);
   Pji(k)=funcPji(Vm(i),Vm(j),delta(i),delta(j),Yb(j,i),Ang(j,i),G(j,i));
end
for k=1:nbr
   i=nl(k);
   j=nr(k);
   Qij(k)=funcQij(Vm(i),Vm(j),delta(i),delta(j),Yb(i,j),Ang(i,j),B(i,j),BSH(i));
end
for k=1:nbr
   i=nl(k);
   j=nr(k);
   Qji(k)=funcQji(Vm(i),Vm(j),delta(i),delta(j),Yb(j,i),Ang(j,i),B(j,i),BSH(j));
end

disp(' ')
T2=[nl';nr';Pij*100;Qij*100;Pji*100;Qji*100];
disp('--------------------------------------------------------------------------------')
disp(' Busi-busj    Pij      Qij      Pji      Qji' )
```

```
disp('-----------------------------------------------------------------------------------')
fprintf(' %2.0f  %2.0f   %3.2f   %3.2f  %3.2f  %3.2f\n',T2)
disp('-----------------------------------------------------------------------------------')
```

โปรแกรม MATLAB คำนวณค่า State Estimation

```
state
%state estimation
clc;clear;
accuracy=0.0001;max_iteration=100;
kVbase=230;MVAbase=100;
load PQinjectmeas.dat.m
load PQflowmeas.dat.m
load PQinjeetsigm.dat.m
load PQflowsigm.dat.m
% Measurement values
Vm=PQinjectmeas(:,2)/kVbase;
Pmi=PQinjectmeas(:,3)/MVAbase;
Qmi=PQinjectmeas(:,4)/MVAbase;
Pmij=PQflowmeas(:,3)/MVAbase;
Qmij=PQflowmeas(:,4)/MVAbase;
Pmji=PQflowmeas(:,5)/MVAbase;
Qmji=PQflowmeas(:,6)/MVAbase;
Zi=[Vm;Pmi;Qmi;Pmij;Pmji;Qmij;Qmji];
%Sigma values
S_V=PQinjectsigm(:,2)/kVbase;
S_Pi=PQinjectsigm(:,3)/MVAbase;
S_Qi=PQinjectsigm(:,4)/MVAbase;
S_Pij=PQflowsigm(:,3)/MVAbase;
S_Qij=PQflowsigm(:,4)/MVAbase;
S_Pji=PQflowsigm(:,5)/MVAbase;
S_Qji=PQflowsigm(:,6)/MVAbase;
S_Zi=[S_V;S_Pi;S_Qi;S_Pij;S_Pji;S_Qij;S_Qji];

Nm=0;
```

```
d=0;
for m=1:length(Zi)
   if abs(Zi(m))<3
      Nm=Nm+1;
      Zmcas(Nm)=Zi(m);
      Sigm(Nm)=S_Zi(m);
   else
      d=d+1;
      Non(d)=m;
   end
end
%Variancc of the measurement
Rn=Sigm.^2;
Rm=1./Rn;
R=diag(Rn);
invR=diag(Rm);
Formybus
%Initial value State variables
delt(1:Nbus)=0;
V(1:Nbus)=1;
x=[delt  V];
TOL=1;
count=0;
%Newton-Ralpson Iterative
while TOL>accuracy & count<max_iteration
   count=count+1;
%Calculate Estimated value,fx
%power injcction

for i=1:Nbus
   Pi=0;Qi=0;
   for n=1:Nbus
```

```
      Pi=Pi+V(i)*V(n)*Y(i,n)*cos(O(i,n)+delt(n)-delt(i));
      Qi=Qi+V(i)*V(n)*Y(i,n)*sin(O(i,n)+delt(n)-delt(i));
    end
    Pci(i)=Pi;
    Qci(i)=-Qi;
end
%power flow
for n=1:nbr
    i=nl(n);j=nr(n);
    K=V(i)*V(j)*Y(i,j);
    Pcij(n)=-(V(i).^2)*G(i,j)+K*cos(O(i,j)+delt(j)-delt(i));
    Pcji(n)=-(V(j).^2)*G(i,j)+K*cos(O(i,j)+delt(i)-delt(j));
    Qcij(n)=-(V(i).^2)*(Bc(i,j)-B(i,j))-K*sin(O(i,j)+delt(j)-delt(i));
    Qcji(n)=-(V(j).^2)*(Bc(i,j)-B(i,j))-K*sin(O(i,j)+delt(i)-delt(j));
end
fxc=[V Pci Qci Pcij Pcji Qcij Qcji];
%Form fx : d,Non
if d>0
    for k=1:d
      fxc(Non(k))=1000;
    end
    u=0;
    for t=1:length(fxc)
      if fxc(t)<1000
        u=u+1;
        fx(u)=fxc(t);
      end
    end
else
    fx=fxc;
end
%culculate e=z-f(x)
```

```
for n=1:Nm
   e(n)=Zmeas(n)-fx(n);
end
%The measurment residual,Jx at the beginning of iteration
for n=1:Nm
   J(n)=(e(n)^2)/R(n,n);
end
Jx=sum(J)
Jxn(count)=Jx;
%From Jacobian matrix,Hx
%H1:PDE(Vi/delti)
H1=zeros(Nbus);
%H6:PDE(Vi)/(Vi)
H8=eye(Nbus);
%H2:PDE(Pi/delti)
%H3:PDE(Qi/delti)
%H9:PDE(Pi/Vi)
%H10:PDE(Qi/Vi)
for i=1:Nbus
   for j=1:Nbus
      if i==j
         Pd=0;Qd=0;
         Pv=2*V(i)*G(i,i);Qv=2*V(i)*B(i,i);
         for n=1:Nbus
            if n~=i
               Pd=Pd+V(i)*V(n)*Y(i,n)*sin(O(i,n)+delt(n)-delt(i));
               Qd=Qd+V(i)*V(n)*Y(i,n)*cos(O(i,n)+delt(n)-delt(i));
               Pv=Pv+V(n)*Y(i,n)*cos(O(i,n)+delt(n)-delt(i));
               Qv=Qv+V(n)*Y(i,n)*sin(O(i,n)+delt(n)-delt(i));
            end
         end
      else
```

```
        Pd=-V(i)*V(j)*Y(i,j)*sin(O(i,j)+delt(j)-delt(i));
        Qd=-V(i)*V(j)*Y(i,j)*cos(O(i,j)+delt(j)-delt(i));
        Pv=V(i)*Y(i,j)*cos(O(i,j)+delt(j)-delt(i));
        Qv=V(i)*Y(i,j)*sin(O(i,j)+delt(j)-delt(i));
      end
      H2(i,j)=Pd;H3(i,j)=Qd;
      H9(i,j)=Pv;H10(i,j)=-Qv;
   end
end
%H4:PDE(Pij/delti)&H5:PDE(Pji/delti)
H4=zeros(nbr,Nbus);H5=H4;
%H6:PDE(Qij/delti)&H7:PDE(Oji/deli)
H6=H4;H7=H4;
%H11:PDE(Pij/Vi)&H12:PDE(Pji/Vi)
H11=H4;H12=H4;
%H13:PDE(Oij/Vi)&H14:PDE(Oji/Vi)
H13=H4;H14=H4;
for n=1:nbr
   i=nl(n);j=nr(n);
   K45=V(i)*V(j)*Y(i,j);
   H4(n,i)=K45*sin(O(i,j)+delt(j)-delt(i));
   H4(n,j)=-H4(n,i);
   H5(n,i)=-K45*sin(O(i,j)+delt(i)-delt(j));
   H5(n,j)=-H5(n,i);
   K67=V(i)*V(j)*Y(i,j);
   H6(n,i)=K67*eos(O(i,j)+delt(j)-delt(i));
   H6(n,j)=-H6(n,i);
   H7(n,i)=-K67*eos(O(i,j)+delt(i)-delt(j));
   H7(n,j)=-H7(n,i);
   K1A=Y(i,j)*eos(O(i,j)+delt(j)-delt(i));
   H11(n,i)=-2*V(i)*G(i,j)+V(j)*K1A;
   H11(n,j)=V(i)*K1A;
```

```
    K2A=Y(i,j)*eos(O(i,j)+delt(i)-delt(j));
    H12(n,i)=V(j)*K2A;
    H12(n,j)=-2*V(j)*G(i,j)+V(i)*K2A;
    K1B=Y(i,j)*sin(O(i,j)+delt(j)-delt(i));
    H13(n,i)=-2*V(i)*(Bc(i,j)-B(i,j))-V(j)*K1B;
    H13(n,j)=-V(j)*K1B;
    K2B=Y(i,j)*sin(O(i,j)+delt(i)-delt(j));
    H14(n,i)=-V(j)*K2B;
    H14(n,j)=-2*V(j)*(Bc(i,j)-B(i,j))-V(i)*K2B;
end
H=[H1 H8;H2 H9;H3 H10;H4 H11;H5 H12;H6 H13;H7 H14];
%[row,col]=[Nm,(2N-1)]
Hc=H(:,2:2*Nbus);
%if not all Measurement
if d>0
   [row,col]=size(Hc);
   for k=1:d
      Hc(Non(k),:)=1000;
   end
   u=0;
   for t=1:row
      if Hc(t,:)<1000
         u=u+1;
         Hx(u,:)=Hc(t,:);
      end
   cnd
else
   Hx=Hc;
end
%Update the state variables
Gx=(Hx')*(invR)*Hx;
[L,U,P]=lu(Gx);
```

```
invGx=inv(U)*inv(L)*P;
S=invGx*(Hx')*invR*e';
dX=[0 S'];
TOL=max(abs(dX));
%Largest delta |delt| and |V|
maxdelta_delt(count)=max(abs(dX(1:Nbus)));
maxdelta_V(count)=max(abs(dX(Nbus+1:2*Nbus)));
x=x+dX;
delt=x(1:Nbus);V=x(Nbus+1:2*Nbus);
end
%==========================end of iteration======================
%Check Bad measurment
Cov=R-Hx*(invGx)*Hx';
for n=1:Nm
  std(n)=e(n)/sqrt(Cov(n,n));
end
if d>0
  yi_norm=zeros(1,Nm+d); % equal 3*Nbus+4*brn
  for n=1:d
    yi_norm(Non(n))=999.9;
  end
  c=0;
  for m=1:(Nm+d)
    if yi_norm(m)==0
      c=c+1;
      yi_norm(m)=std(c);
    end
  end
else
  yi_norm=std;
end
%number of Meas.
```

```
Nm;
%Thresthod Check 99% condidentiall +-2.58 >>>0.495*2=0.99
k=Nm-(2*Nbus-1);
u=k;
sdv=sqrt(2*k);
y=2.33;
Tj=y*sdv+u
%display the results
Table1=[1:count;Jxn;maxdelta_V;maxdelta_delt];
disp('-----------------------------------------------')
disp('Iteration Jx  max delta|V|    max delta|delta| ')
disp('-----------------------------------------------')
fprintf('%3.0f  %10.2f  %4.8f   %4.8f\n',Table1)
disp('-----------------------------------------------')
%Calculate Estimated value,Z_estimate
%Power injection
for i=1:Nbus
   Pi=0;Qi=0;
   for n=1:Nbus
      Pi=Pi+V(i)*V(n)*Y(i,n)*cos(O(i,n)+delt(n)-delt(i));
      Qi=Qi+V(i)*V(n)*Y(i,n)*sin(O(i,n)+delt(n)-delt(i));
   end
   Pci(i)=Pi;
   Qci(i)=-Qi;
end
%Power flow
for n=1:nbr
   i=nl(n);j=nr(n);
   K=V(i)*V(j)*Y(i,j);
   Pcij(n)=-(V(i).^2)*G(i,j)+K*cos(O(i,j)+delt(j)-delt(i));
   Pcji(n)=-(V(j).^2)*G(i,j)+K*cos(O(i,j)+delt(i)-delt(j));
   Qcij(n)=-(V(i).^2)*(Bc(i,j)-B(i,j))-K*sin(O(i,j)+delt(j)-delt(i));
```

```
  Qcji(n)=-(V(j).^2)*(Bc(i,j)-B(i,j))-K*sin(O(i,j)+delt(i)-delt(j));
end
V_est=V*kVbase;
Pi_est=Pci*MVAbase;Qi_est=Qci*MVAbase;
Pij_est=Pcij*MVAbase;Pji_est=Pcji*MVAbase;
Qij_est=Qcij*MVAbase;Qji_est=Qcji*MVAbase;
%Estimated Voltage(kV)
disp('No.   Voltage_est(kV) Yi_norm')
disp('---------------------------------------------')
Table2=[(1:Nbus);V_est;yi_norm(1:Nbus)];
fprintf('%3.0f %4.1f  %4.2f\n',Table2)
disp('---------------------------------------------')
%Estimated Power injection(NW,MVar)
disp('No.  Pi_est(MW) Yi_norm  Qi_est(MVar)   yi_norm')
disp('---------------------------------------------')
Table3=[(1:Nbus);Pi_est;yi_norm(Nbus+1:2*Nbus);Qi_est;yi_norm(2*Nbus+1:3*Nbus)];
fprintf('%3.0f %4.1f  %4.2f  %4.1f  %4.2f\n',Table3)
disp('---------------------------------------------')
%Estimated power flow (MW,MVar)
disp('Line  Pi_est(MW) Yi_norm  Pi_est(MVar)   yi_norm')
disp('---------------------------------------------')
Table4=[nl';nr';Pij_est;yi_norm(3*Nbus+1:3*Nbus+nbr);Pji_est;yi_norm(3*Nbus+nbr+1:3*Nbus
+2*nbr)];
fprintf('%3.0f %3.0f  %4.1f  %4.2f  %4.1f  %4.2f\n',Table4)
disp('---------------------------------------------')
%Estimated power flow (MW,MVar)
disp('Line  Qij_est(MW) Yi_norm  Qji_est(MVar)   yi_norm')
disp('---------------------------------------------')
Table5=[nl';nr';Qij_est;yi_norm(3*Nbus+2*nbr+1:3*Nbus+3*nbr);Qji_est;yi_norm(3*Nbus+3*n
br+1:3*Nbus+4*nbr)];
fprintf('%3.0f %3.0f  %4.1f  %4.2f  %4.1f  %4.2f\n',Table5)
disp('---------------------------------------------')
```

โปรแกรม MATLAB ฟังก์ชันคำนวณค่า Power Flow

```
function Pij=funcPij(Vmi,Vmj,deltai,deltaj,Ybij,Angij,Gij)
Pij=-(Vmi.^2)*Gij+Vmi*Vmj*Ybij*cos(Angij+deltaj-deltai);

function Pji=funcPji(Vmi,Vmj,deltai,deltaj,Ybji,Angji,Gji)
Pji=-(Vmj.^2)*Gji+Vmj*Vmi*Ybji*cos(Angji+deltai-deltaj);

function Qij=funcQij(Vmi,Vmj,deltai,deltaj,Ybij,Angij,Bij,BSHi)
Qij=-(Vmi.^2)*(BSHi-Bij)-Vmi*Vmj*Ybij*sin(Angij+deltaj-deltai);

function Qji=funcQji(Vmi,Vmj,deltai,deltaj,Ybji,Angji,Bji,BSHj)
Qji=-(Vmj.^2)*(BSHj-Bji)-Vmj*Vmi*Ybji*sin(Angji+deltai-deltaj);
```

โปรแกรมข้อมูล

%Power flow measurement data

%	i	j	Pij(MW)	Qij(MVar)	Pji(MW)	Qji(MVar)
	1	2	156.87	-20.40	-152.57	27.87
	1	5	75.37	4.19	-72.62	2.46
	2	3	73.33	3.26	-71.00	1.64
	2	4	56.08	-1.11	-54.40	1.58
	2	5	41.46	1.06	-40.56	-2.75
	3	4	-23.20	4.16	23.58	-7.35
	4	5	-61.08	11.65	61.60	-13.74
	4	7	28.05	-23.68	-28.05	23.53
	4	9	16.06	-8.83	-16.06	8.50
	5	6	43.97	-22.06	-43.97	23.63
	6	11	7.39	1.65	-7.34	-3.51
	6	12	7.63	-0.61	-7.56	-1.24
	6	13	17.76	7.64	-17.70	-9.16
	7	8	0.00	-18.30	0.00	18.07
	7	9	28.05	4.72	-28.05	-4.64
	9	10	5.19	4.15	-5.18	-4.12
	9	14	9.42	2.97	-9.31	-2.74
	10	11	-3.82	-1.68	3.84	1.71
	12	13	1.46	-0.36	-1.46	0.37
	13	14	5.66	2.99	-5.59	-2.86

%Power flow measurement's sigma data

%	i	j	Pij(MW)	Qij(MVar)	Pji(MW)	Qji(MVar)
	1	2	1	1	1	1
	1	5	1	1	1	1
	2	3	1	1	1	1
	2	4	1	1	1	1
	2	5	1	1	1	1
	3	4	1	1	1	1
	4	5	1	1	1	1
	4	7	1	1	1	1
	4	9	1	1	1	1
	5	6	1	1	1	1
	6	11	1	1	1	1
	6	12	1	1	1	1
	6	13	1	1	1	1
	7	8	1	1	1	1
	7	9	1	1	1	1
	9	10	1	1	1	1
	9	14	1	1	1	1
	10	11	1	1	1	1
	12	13	1	1	1	1
	13	14	1	1	1	1

%From the measurement matrix,(Nm,zmeas)

%Voltage & Power injection measurement data

%	Bus no.	Vi(kV)	Pi(MW)	Oi(MVar)
	1	243.8000	232.17	-16.02
	2	240.3500	18.30	32.57
	3	232.3000	-94.20	6.23
	4	233.5635	-47.80	-3.90
	5	234.2264	-7.60	-1.80
	6	246.1000	-11.20	7.00
	7	243.9842	0.00	0.00
	8	250.7000	0.00	18.03
	9	242.7784	-29.50	-16.60
	10	241.6549	-9.00	-5.80
	11	243.0516	-3.50	-1.80
	12	243.3610	-6.10	-1.60
	13	242.8181	-13.50	-5.80
	14	238.4651	-14.90	-5.60

%Voltage & Power injection measurement's sigma data

%	Bus no.	Vi(kV)	Pi(MW)	Oi(MVar)
	1	1	1	1
	2	1	1	1
	3	1	1	1
	4	1	1	1
	5	1	1	1
	6	1	1	1
	7	1	1	1
	8	1	1	1
	9	1	1	1
	10	1	1	1
	11	1	1	1
	12	1	1	1
	13	1	1	1
	14	1	1	1