MGM'S

Jawaharlal Nehru Engineering College, Aurangabad

PSA Laboratory Manual

For

Third Year Students

Manual made by

Miss. M.P. Saware

LABORATORY MANNUAL CONTENTS

This manual is intended for the second year students of Electronics Engineering in the subject of Programmable logic controller.

Students are advised to thoroughly go through this manual rather than only topics mentioned in the syllabus as practical aspects are the key to understanding and conceptual visualization of theoretical aspects covered in the books.

Good Luck for your Enjoyable Laboratory Sessions

Miss. Molleshree Saware

FOREWORD

It is my great pleasure to present this laboratory manual for second year engineering students for the subject of Programmable logic control. Keeping in view the vast coverage required for visualization of concepts of Programmable logic controller components with simple language.

As a student, many of you may be wondering with some of the questions in your mind regarding the subject and exactly what has been tried is to answer through this manual.
Faculty members are also advised that covering these aspects in initial stage itself, will greatly relived them in future as much of the load will be taken care by the enthusiasm energies of the students once they are conceptually clear

Prof. Dr. S.D.Deshmukh

Principal

Experiment no 01:-

Aim : Introduction of MATLAB:-

Theory:

Q. How to start MATLAB ?

Q. write the matrix initialization and basic operation ?

Q. How to Run programs given in this appendix ?

Conclusion :

Experiment no 02:-

Aim : To study singular transformation using MATLAB

Theory:

Q. write down algorithmic steps of singular transformation using MATLAB ?

Conclusion :

Experiment no 03:-

Aim : Solution of power flow problem using gauss seidal method

Theory:

Q. what is the procedure to obtain the solution of power flow problem using gauss seidal method and its advantages ? and drawbacks ?

MATLAB PROGRAM :

```
%Program for load flow by Gauss-Seidal Method
Clear;
N=4
V=[1.04 1.04 11]
Y=   [3-j*9      - 2+j*6          -1+j*3          0
      -2+j*6       3.66-j*11      -0.666+j*2      -1+j*3
      -1+j*3      -0.666+j*2      3.66-j*11       -2+j*6
      0            -1+j*3          -2+j*6          3-j*9 ]
Type=ones(n,1)
Typechanged=zeros(n,1)
```

```
Qlimitmax=zeros(n,1)
Vmagfixed= zeros(n,1)

Type(2)=2

Qlimitmax(2)=1.0

Qlimitmax(2)=0.2

Vmagfixed(2)=1.04

Diff=10;noofiter=1

Vprev=V;

While(diff>0.00001/ noofiter==1),

Abs(V)

Abs(Vprev)

Vprev=V;

P = [inf   0.5   -1    0.3];

Q = [inf  0.0   0.5   -0.1];

S=[inf+j*inf (0.5-j*0.2)  (-1.0+j*0.5)  (0.3-j*0.1)];

For i=2:n,

If type(i)==2 | typechenged(i)==1,

If type(i)>Qlimitmax(i) | Q(i)<Qlimitmin(i),

If (Q(i)<Qlimitmin(i),

   Q(i)=Qlimitmin(i);

Else

Q(i)=Qlimitmax(i);
```

```
End
Type(i)=1;
Typechanged(i)=1;
Else
Type(i)=2;
Typechanged(i)=0;
End

End

End
Sumyu=0;
Fork=1:n,
If(i~=k)
  Sumyv=sumyu=y(i,k)*V(k);
     end
end
V(i)=(1/y(i,i)*( (p(i)-j*Q(i))/conj(V(i))-sumyv);
If type(i)==2 & typechanged(i)~=1,
V(i)=PolarTorect(Vmagfixed(i),angle(V(i))*180/pi);
      End
```

```
End

Diff=max(abs(abs(V(2:n))-abs(Vprev(2:n))));

Noofiter=noofiter=+1;
```

Output :

Conclusion :

Experiment no 04:-

Aim : solution of power flow problem using N-R method

Theory :

Q. what is the procedure to obtain the solution of power flow problem using N - R method and its advantages over gauss sidal?

Q. write down the difference in gauss seidal and N-R method ?

MATLAB PROGRAM :

```
%Program for load flow by Newton- raphson Method
Clear
n=3;
V=[1.04  1.0  1.04]
Y= [ 5.88228-j*23.50514    -2.9427+j*11.7676        -2.9427+j*11.7676
      -2.9427+j*11.7676         5.88228-j*23.50514     -2.9427+j*11.7676
      -2.8427+j*11.7676        -2.9427+j*117676          5.88228-j*23.50514 ]
type =ones(n,1);
```

```
Typechanged=zeros(n,1):

Qlimitmax=zeros(n,1);

Qlimitmin=zeros(n,1);

Vmagfixed=zero(n,1);

Type(3)=2;

Qlimitmax(3)=1.5;

Qlimitemin(2)=0;

Vmagfixed(2)=1.04;

Diff=10;noofiter=1.0

Pspec=[inf 0.5  -1.5]

Qspec=[inf 1    0];

S=[inf+j*inf(0.5-j*0.2)  (-1.0+j*0..5)   (0.3-j*0.1)];

While (diff>0.00001 | noofiter==1),

Eqcount=1;

For i=2:n,

Abs(Vprev)

Pause

Vprev=V;

For ceq=1:eqcount,

    Am=real(Y(assoeqbus(ceq),assocolbus(ccol))*V(assocolbus(ccol)));

   bm =imag(Y(assoeqbus(ceq),assocolbus(ccol))*V(assocolbus(ccol)));
```

```
  ei= real(V(assoeqbus(ceq)));
fi= imag(V(assoeqbus(ceq)));


if assoeqvar(ceq)==’p’ & assocolbus(col)==’d’,
    if assoeqbus(ceq)~=assocolbus(ccol),
         H=am*fi-bm*ei;
Else
    H=Q(assoeqbus(ceq))-imag(Y(assoeqbus(ceq)))^2);
   End
   Jacob(ceq,ccol)=H
 End
If assoeqvar(ceq)=’p’ & assocolver(ccol)==’v’,
If assoeqbus(ceq)`= assocolver(ccol),
N =am*ei+bm*fi
Else
N=P(assoeqbus(ceq))+real(Y(assoeqbus(ceq),assocolbus(ceq))*abs(V(assocolbus(ceq)))^2);
Scal(i)=0;
Sumyv=0;
Fork=1:n,
Sumyv=sumyv+Y(i,k)*V(k);
End
```

```
Scal(i)=V(i)*conj(sumyv);
P(i)=real(scale(i));
Q(i)=image(Scal(i));
If type(i)==2 | typechanged(i)==1,
If(Q(i)<Qlimitmax(i) | Q(i)<Qlimitmax(i)),
    If(Q(i)<Qlimitmax(i));
        Q(i)<Qlimitmax(i);
Else
Q(i)=Qlimitmax(i);
End
Type(i)=1;
Typechanged(i)=0;
End
End
If type (i)==1,
Assoeqvar(eqcount)='p';
Assoeqbus(eqcount)=i;
Mismatch(eqcount)=Pspec(i)-P(i);
Assoceqvar(eqcount+1)='Q';
Assoeqbus(eqcount+1)=i;
Mismatch(eqcount+1)=Qpec(i)-Q(i);
```

```
Assoeqvar(eqcount)='d';
Assoeqbus(eqcount)=i;
Assoceqvar(eqcount+1)='V';
Assoeqbus(eqcount+1)=i;
Mismatch=(eqcount+2);
Else

Assoeqvar(eqcount)='p';
Assoeqbus(eqcount)=i;
Assoceqvar(eqcount+1)='Q';
Assoeqbus(eqcount+1)=d;
Mismatch(eqcount+1)=Ppec(i)-P(i);
Eqcount=eqcount+1:
End
End
Mismatch
Eqcount=eqcount-1;
Noofeq=eqcount;
Update=Zeros(eqcount,1)
Vprwv=V
Abs(V);
```

```
End
Jacob(ceq,ccol)=N
End
If assoeqvar(ceq)==’Q’ & assocolvar (ccol)==’d’,
If assoeqbus(ceq)~=assocolbus(ccol),
J=am*ei+bm*fi;
Else
J=p(assoeqbus(ceq))+real(Y(ass0eqbus(ceq),assocolbus(ceq)*abs(V(assoeqbus(ceq)))^2);
    End
    Jacob(ceq,ccol)=J
    End

If assoeqvar(ceq)==’Q’ & ass0colvar (ccol)==’V’,
If assoeqbus(ceq)~=assocolbus(ccol),
    L=am*fi-bm*ei;
    else
    L=Q(assoeqbus(ceq))-
Imag(Y(assoeqbus(ceq),assocolbus(ceq))*abs(V(assoeqbus(ceq)))^2);
    End
        Jacob(ceq,ccol)=L;
        End
```

```
End

End

Jacob;

Pause

Update=inv(Jacob)*mismatch;

Noofeq=1;

For i=2:n,

    If i=2:n,

    newchinangV=update(noofeq);

    newangV=angle(V(i))+newchinangV;

    newchinmagV=update(noofeq+1)*abs(V(i));

    newangV=abs(V(i))+newchinangV;

    V(i)=polartorect(newmagV*180/pi);

    Noofeq=noofeq+2;

    Else

    newchinmagV=update(noofeq);

    newangV=angle(V(i))+newchinangV;

    V(i)=polartorect(newmagV*180/pi);

            Noofeq=noofeq+2;

    end

end

clear mismatchJacob update assoeqvar assoeqbus Assoeqlvar assocllous
```

```
diff=min(abs(abs(V(2:n))-abs(Vprwv(2:n))));
noofiter=noofiter+1;
```

Output :

Conclusion :

Experiment no 05:-

Aim : Solution of formation of bus impedance matrix by building algorithm :-

Theory :

Q. write down the detail procedure of formation of bus impedance matrix by building algorithm ?

Conclusion:

Experiment no 06 :-

Aim : Programme for optimum loading of generator :-

MATLAB PROGRAM :-

```
clear ;
n=2
Pd = 231.25
alpha = [0.20   0.25]
beta = [40  30]
lamda = 20
lamdaprev = lambda
esp = 1
deltalambda = 0.25
Pgmin = [20 20]
 pg = 1008ones(n,1)
while abs (sum (Pg)-Pd)>eps
        for i= 1:n,
            Pg(i) = (lamda-beta(i)) / alpha(i) ;
```

```
 if Pg(i)>Pmax(i)
                    Pg(i) = Pmax (i);
        end
                   if Pg(i)>Pmin(i)
                      Pg(i) = Pmin (i);
       end
 end
if(sum (Pg-Pd)) < 0
   lamdaprev = lambda
   lamda= lamda + deltalamda ;
else
   lamdaprev = lambda
   lamda= lamda - deltalamda ;
 end
end
dis('the final value of lamda is')
lamdaprev
dis('the distribution of load shared by two units is')
Pg
```

Output :

Conclusion :

Experiment no 07 :-

Aim : Programme for building of Z-Bus by addition of branch or link :-

Theory :

Q. write down the algorithymic steps of buildung zbus by addition of branch or link

MATLAB PROGRAM :-

```
clear ;
zprimary = [1100.25
            2210.1
            3310.1
            4200.25
            5230.1 ]
[elements columes] = size(zprimary)
zbus = []
currentbus no = 0
for count = 1:elements,
[row cols] = size(zbus)
from=zprimary(count,2)
```

```
to=zprimary(count,3)

valu=zprimary(count,4)

newbus=max(from,to)

if newbus>currentbusno & ref = = 0

zbus = [zbus zeros(rows,1)

        zeros(1,cols) value]

currentbus no =newbus

continue

end

if newbus>currentbusno & ref ~= 0

  zbus=[zbus zbus(:,ref)

      zbus(ref,:) value+zbus(ref,ref)]

  currentbusno = newbus

  continue

end

if newbus<=currentbusno & ref ~= 0

  zbus=zbus-1/(value+zbus(from,from)+zbus(to,to)-2*zbus(from,to))*(zbus(:,from)-
zbus(:,to))*((zbus(from,:)-zbus(to,:)))

continue

end

end
```

Output :

Conclusion :