เอกสารอ้างอิง

1. K.R.C. Mamandur and R.D. Chenoweth , " Optimal control of reactive power flow for improvement in voltage profile and real power losses minimization ", IEEE Trans, P.A.S.,Vol.100, No.7,July 1981, pp. 3165-3193.
2. S. Ramalyer,R. Ramachandran,and S. Hariharan," New technique for optimal reactive-power allocation for loss minimization in power system ", IEEE Proceeding,Vol.130,No.4,July 1983,pp. 178-182.
3. เทอดทรงชัย พุทธิศรี, " การลดกำลังงานสูญเสียในระบบไฟฟ้ากำลังให้น้อยที่สุดด้วยการควบคุมกำลังรีแอคทีฟของระบบให้เหมาะสม ", วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย,จุฬาลงกรณ์มหาวิทยาลัย,2532.
4. W.J. Lyman, " Controlling power flow with phase shiftering transformer ", AIEEE Transaction , Vol.49,July 1930,pp.825-831.
5. W.J. Lyman and J.R. North, " Application of large phase -shifter transformer on an interconnected system loop ", AIEEE Transaction ,Vol.57, Oct. 1938, pp 579-587.
6. Z.X. Han," Phase shifter and power flow control ", IEEE Trans,P.A.S.,Vol.101,No.10,October 1982, pp. 3790-3795.
7. N. Srinivasan K.S. Prakasa,C.S. Indulkar." On-line computation of phase shifter distribution factors and line load alleviation ",IEEE Trans, P.A.S.,Vol.104,No.7,July 1985,pp. 1656-1662.
8. ชำนาญ ห่อเกียรติ," ระบบไฟฟ้ากำลัง ",ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์,2526.

9. หลักฐาน ทองนพคุณ," การจ่ายโหลดอย่างประหยัดโดยใช้การจัดสรรกำลังจริงและกำลังรีแอคทีฟ ",วิทยานิพนธ์ปริญญามหาบัณฑิต, ภาควิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2530.

10. Kenneth L. Gebert,Kenneth R.Edwards," Transformers ", American Technical Society,1955.

11. A. Thanikachalam,V.T. Sulzberger,P.Van Olinda and J.N. Wrubel, " On-line Opertion of Phase Shifters Using Control Center Computer",IEEE Transaction Power Apparatus and Systems , Vol.Pas-93, pp. 1563-1570,1974.

12. Stagg,G.W and El-abaid," Computer Methods In Power SystemAnalysis ",New York,Mc Graw-Hill,1968

13. Stevenson,D. william Jr.," Element Of Power System Analysis ",4th ed,Mc Graw-Hill,1982.

14. G.T. Heydt," Computer Analysis Method For Power System ",New York,Macmillan Publishing Company, 1986.

15. N.M. Peterson and W.S. Meyer,"Automatic Adjustment Of Transformer And Phase-Shifter Taps ", IEEE TransactionOn Power Apparatus And System, Vol. Pas-90,pp.104-106,1971.

ภาคผนวก

ภาคผนวก ก.

การคำนวนกระแสฉีด(Injected Current)
และกำลังฉีด(Injected Power)ที่บัส

ถ้าเราไม่คำนึงถึงกำลังสูญเสียในตัวเลื่อนเฟส จะได้ความสัมพันธ์ระหว่างinput power และ output power ดังนี้

$$V_i I_i = V_j I_j \qquad (ก.1)$$

หรือ

$$V_j/V_i = I_i/I_j = e^{j\phi} \qquad (ก.2)$$

จากสมการ (4.3) และรูปที่ 4.2 จะได้ว่า

กระแสฉีดที่บัส j $I_{jm} = I_m'$ (ก.3)

$= (V_j - V_i)/Z_{ij}$ (ก.4)

จากสมการ (ก.2) จะได้ว่า

$$V_j = V_i e^{j\phi} \qquad (ก.5)$$

ดังนั้น

$$I_{jm} = (V_i e^{j\phi} - V_i)/Z_{ij} \qquad (ก.6)$$

$$= V_i(e^{j\phi} - 1)/Z_{ij} \qquad (ก.7)$$

และเช่นเดียวกันสำหรับกระแสฉีดที่บัส i

จะได้ว่า $I_{im} = -V_j(1 - e^{-j\phi})/Z_{ij}$ (ก.8)

จากสมการ (4.4) และสมการ (ก.8) กำลังฉีด(Injected Power)ที่บัส i สามารถแสดงได้ดังสมการ

$$S_{im} = V_i * I_{im} \qquad (ก.9)$$

$$= V_i * [-V_j(1-e^{-j\phi})/Z_{ij}] \qquad (ก.10)$$

เมื่อ $e^{-j\phi} = \cos\phi - j\sin\phi$ สมการ (ก.10) สามารถเขียนได้ดังนี้

$$S_{im} = V_i * [-V_j(1-(\cos\phi - j\sin\phi))/Z_{ij}] \qquad (ก.11)$$

ในระบบไฟฟ้ากำลังเราถือว่า G << B รวมทั้งคิดว่ามุมของตัวเลื่อนเฟส(ϕ)เป็นมุมที่มีขนาดเล็กๆซึ่งจะทำเราสามารถอนุโลมให้ $\cos\phi = 1$ ดังนั้นสมการ(ก.11) สามารถเขียนให้ได้ดังนี้

$$S_{i*} = -V_i V_j B_{ij} \sin\phi \qquad (ก.12)$$

และเช่นเดียวกันสำหรับกำลังฉีดที่บัส j

จะได้ว่า $$S_{j*} = V_i V_j B_{ij} \sin\phi \qquad (ก.13)$$

หรือเมื่อเป็นกำลังจริงฉีด(Injected Real Power)ที่มองเข้าทางปลายบัสจะได้ว่า

$$P_{i*} = S_{i*} = V_i V_j B_{ij} \sin\phi\cos(\delta_i - \delta_j) \qquad (ก.14)$$

$$P_{j*} = S_{j*} = -V_i V_j B_{ij} \sin\phi\cos(\delta_i - \delta_j) \qquad (ก.15)$$

สมการ (ก.14) และ (ก.15) คือ กำลังจริงฉีดที่บัส i และบัส j เนื่องมาจากตัวเลื่อนเฟสที่มีมุม(ϕ)ตามลำดับ ซึ่งก็คือสมการ (4.6)และ(4.7) ส่วนกำลังรีแอคทีฟฉีดที่บัส i และบัส j สามารถแสดงได้ดังสมการ (ก.16) และ (ก.17) ตามลำดับดังนี้

$$Q_{i*} = V_i V_j B_{ij} \sin\phi\sin(\delta_i - \delta_j) \qquad (ก.16)$$

$$Q_{j*} = -V_i V_j B_{ij} \sin\phi\sin(\delta_i - \delta_j) \qquad (ก.17)$$

เนื่องจากค่า $(\delta_i - \delta_j)$ มีค่าน้อยจึงทำให้ค่ากำลังรีแอคทีฟฉีดดังกล่าวนี้มีค่าน้อยมากเมื่อเทียบกับค่าของกำลังจริงฉีด

ภาคผนวก ข.

โปรแกรมการลดกำลังไฟฟ้าสูญเสียให้มีค่าน้อยที่สุด
โดยการควบคุมการไหลของกำลังจริงด้วยตัวเลื่อนเฟส

```
c     *** This program is used to calculate load flow ***
c     *** By NEWTON-RAPHSON method ***
c     *** Main program ***
      Program PAN1
      Integer se1,se2,ic1,in,sp
      Real jacob
      complex zser,ztser,ycap,ysht,v,yser,ytser,y,vn,y1,s,ys,r,ts,ttr,
     *        y2,cz1,cz2,cz3,cz4,cz5,yp,yq,ycon,cz6,cz7
      common nvar,nc,pbase,error,alpha,lnit,ntb(30),zser(40),
     *        vmax(30),vmin(30),ycap(30),nct(30),mtb(5),power,
     *       vbase,tr(5),nb,nt,mab(5),pg(30),pd(30),qg(30),qd(30),
     *       qmax(30),qmin(30),nl,nsb(40),neb(40),ztser(5),
     *       ysht(40),v(30),yser(40),ytser(5),y(30,30),nload,ncon,
     *       noob,nld,nswg,nk(30),pdn(30),pgn(30),qdn(30),qgn(30),
     *       vn(30),nbtypn(30),qmaxn(30),qminn(30),vmaxn(30),vminn(30),
     *       mtbn(5),mabn(5),dp(30),dq(30),da(30),db(30),del(60),
     *       delv(60),g(30,30),h(30,30),aa(30),b(30),jacob(60,60),
     *       p(30),q(30),pcal(30),qcal(30),dmax,nit,cc(30),vmag(30),
     *       angle(30),t,s(40),y1,nsj,nj,nog,ngn,na,n,tplol,tqlol,r(40),
     *       ys,plol(40),qlol(40),tpol(5),tqol(5),yss(40),ts(5),ttr(5),
     *       y22(30),npst,nsbp(10),nebp(10),zeta(10),nsbpn(10),
     *       nebpn(10),nlp(10),yp(40),yq(40),cpst,cz1,cz2,cz3,cz4,cz5,
     *       ycon(40,40),nyy,cz6,cz7,fe(10),xm(30,30),zc(30,30),fed(10),
     *       xser(40),d(40,40),rser(40),pf(40),am(10,10),bm(10),cnol,
     *       amv(10,10),tl(40)
      open (3,file='LFLOW',status='new')
      write (*,10)
```

```
      write (3,10)
10    format (5x,'*******************************************')
      write (*,20)
      write (3,20)
20    format (5x,'*** Load Flow calculation by NR method ***',/,
     *5x,'*** Limit of this program ***',/,
     *5x,'maximum bus number = 30 ',/,
     *5x,'maximum line number = 40 ',/,
     *5x,'maximum transformer = 5 ',/,
     *5x,'maximum phase shifter = 10')
      write (*,10)
      write (3,10)
      write (*,30)
      write (3,30)
30    format (/,5x,'** START TO CALCULATE NEWTON-RAPHSON LOAD FLOW **'
     *,/,5x,'** FOR USE IN BASE CASE STUDY **')
      write (*,31)
      write (3,31)
31    format (/,5x,'**** MODE TO INPUT SYSTEM DATA ****',/,
     *5x,'1 = input data by interactive mode',/,
     *5x,'0 = input data by disk reading mode',/,
     *5x,'** Your selection = ',$)
      read (*,32) in
32    format (i3)
      if (in.eq.1) then
      call indata
      else
      call didata
      endif
55    continue
      call pindat
      write (*,60)
      write (3,60)
60    format (/,5x,'** Now we will start LF. calculation **',/,5x,
```

```
     *'1 means continue this program',/,5x,
     *'0 means stop and end this program',/,5x,
     *'** Your selection = ',$)
      read (*,50) se2
50    format (i2)
      if (se2.eq.1) then
      go to 70
      else
      go to 100
      endif
70    continue
      call modat
      call ybus
      call cob
      call nrlf
      if (t.eq.1) then
      go to 75
      else
      call output
      go to 80
      endif
75    continue
      write (*,77)
      write (3,77)
77    format (/,10x,'THIS BASE CASE IS NOT SATISFY NR. MATHOD !!!')
      go to 100
80    total = power
      write (*,85) power
      write (3,85) power
85    format (//,5x,'*** POWER LOSS IN BASE CASE = ',f10.4,'  MW ***')
      call xdcal
199   continue
      write (*,200)
      write (3,200)
```

```
200   format (/,5x,'** Do you want to install pst. in same system ? **'
     *,/,5x,'1 means = yes',/,
     *5x,'0 means = no',/,
     *5x,'Your selection =',$)
      read (*,32) sp
      if (sp.eq.1) then
      go to 250
      else
      go to 100
      endif
250   continue
      call didata
      call pster
      call fecal
      if (cnol.eq.1.0) go to 199
      call modat
      call ybus
      call cob
      call nrlf
      if (t.eq.1) then
      go to 260
      else
      call output
      go to 270
      endif
260   continue
      write (*,261)
      write (3,261)
261   format (/,10x,'THIS CASE IS NOT SATISFY NR. MATHOD !!!')
      go to 300
270   continue
      do 271 i=1,npst
      write (*,272) nlp(i),fed(i)
      write (3,272) nlp(i),fed(i)
```

```
272   format(/,5x,'we put pst. at line No.=',i2,'  angle(deg.) =',f10.5)
271   continue
      write (*,273) power
      write (3,273) power
273   format (//,5x,'** POWER LOSS IS =',f10.4,'  MW **')
      go to 300
300   continue
      go to 199
100   continue
      write (*,101)
      write (3,101)
101   format (//,15x,'** END THIS PROGRAM **')
      close (3)
      stop
      end

c     *** subroutine to input system data by interactive mode ***
      subroutine indata
      Real jacob
      complex zser,ztser,ycap,ysht,v,yser,ytser,y,vn,yl,s,ys,r,ts,ttr,
     *        y2,cz1,cz2,cz3,cz4,cz5,yp,yq,ycon,cz6,cz7
      common nvar,nc,pbase,error,alpha,lnit,ntb(30),zser(40),
     *        vmax(30),vmin(30),ycap(30),nct(30),mtb(5),power,
     *       vbase,tr(5),nb,nt,mab(5),pg(30),pd(30),qg(30),qd(30),
     *       qmax(30),qmin(30),nl,nsb(40),neb(40),ztser(5),
     *       ysht(40),v(30),yser(40),ytser(5),y(30,30),nload,ncon,
     *       noob,nld,nswg,nk(30),pdn(30),pgn(30),qdn(30),qgn(30),
     *       vn(30),nbtypn(30),qmaxn(30),qminn(30),vmaxn(30),vminn(30),
     *       mtbn(5),mabn(5),dp(30),dq(30),da(30),db(30),del(60),
     *       delv(60),g(30,30),h(30,30),aa(30),b(30),jacob(60,60),
     *       p(30),q(30),pcal(30),qcal(30),dmax,nit,cc(30),vmag(30),
     *       angle(30),t,s(40),yl,nsj,nj,nog,ngn,na,n,tplol,tqlol,r(40),
     *       ys,plol(40),qlol(40),tpol(5),tqol(5),yss(40),ts(5),ttr(5),
     *       y22(30),npst,nsbp(10),nebp(10),zeta(10),nsbpn(10),
```

```
     *       nebpn(10),nlp(10),yp(40),yq(40),cpst,cz1,cz2,cz3,cz4,cz5,
     *       ycon(40,40),nyy,cz6,cz7,fe(10),xm(30,30),zc(30,30),fed(10),
     *       xser(40),d(40,40),rser(40),pf(40),am(10,10),bm(10),cnol,
     *       amv(10,10),t1(40)
      write (*,1)
      write (3,1)
1     format (/,10x,'** INPUT SYSTEM DATA BY INTERACTIVE MODE **')
      write (*,10)
      write (3,10)
10    format (//,2x,'*** !!! IN PUT SYSTEM DATA !!! ***',//,10x,
     *'number of bus =',$)
      read (*,20) nb
20    format (i5)
      write (*,30)
      write (3,30)
30    format (/,10x,'number of line =',$)
      read (*,20) nl
      write (*,240)
      write (3,240)
240   format (/,10x,'number of transformer =',$)
      read (*,20) nt
      write (*,320)
      write (3,320)
320   format (/,10x,'number of var.source  =',$)
      read (*,20) nvar
      write (*,480)
      write (3,480)
480   format (/,10x,'number of shunt capaciter =',$)
      read (*,20) nc
      write (*,40)
      write (3,40)
40    format (/,10x,'base mva =',$)
      read (*,50) pbase
      write (*,60)
```

```
      write (3,60)
60    format (/,10x,'max. error =',$)
      read (*,50) error
50    format (f12.6)
      write (*,70)
      write (3,70)
70    format (/,10x,'accelerating factor =',$)
      read (*,50) alpha
      write (*,80)
      write (3,80)
80    format (/,10x,'maximium iteration in NR method acceptacle =',$)
      read (*,20) lnit
      write (*,90)
      write (3,90)
90    format (//,2x,'***** BUS DATA *****')
      do 5 i=1,nb
      write (*,100) i
      write (3,100) i
100   format (//,10x,'** BUS NO.',i3, '**',/,
     *15x,'type=1---load bus',/,
     *15x,'type=2---voltage controlled bus',/,
     *15x,'type=3---swing bus',//,
     *15x,'!! type of this bus  =',$)
      read (*,20) ntb(i)
      write (*,110) i
      write (3,110) i
110   format (/,5x,'bus no.',i3,' specify voltage(pu) =',$,3x,$)
      read (*,51) v(i)
51    format (2f12.6)
      write (*,120) i
      write (3,120) i
120   format (/,5x,'bus no.',i3,' base voltage (kv) = ',$)
      read (*,50) vbase
      write (*,130) i
```

```
      write (3,130) i
130   format (/,5x,'bus no.',i3,' real power generate(mw) = ',$)
      read (*,50) pg(i)
      write (*,140) i
      write (3,140) i
140   format (/,5x,'bus no.',i3,' reactive power generate(mvar) =',$)
      read (*,50) qg(i)
      write (*,150) i
      write (3,150) i
150   format (/,5x,'bus no.',i3,' real power demand(mw) =',$)
      read (*,50) pd(i)
      write (*,160) i
      write (3,160) i
160   format (/,5x,'bus no.',i3,' reactive power demand(mvar) =',$)
      read (*,50) qd(i)
5     continue
      write (*,180)
      write (3,180)
180   format (//,2x,'*** LINE DATA ***')
      do 15 i=1,nl
      write (*,190) i
      write (3,190) i
190   format (//,2x,'*line no.*',i3,/,10x,'sending bus (p) =',$)
      read (*,20) nsb(i)
      write (*,200) i
      write (3,200) i
200   format (/,2x,'*line no.*',i3,/,10x,'ending bus (q) = ',$)
      read (*,20) neb(i)
      write (*,210) i
      write (3,210) i
210   format (/,5x,'line no.',i3,'  series resistance and reactance(pu)
     *=',$,3x,$)
      read (*,51) zser(i)
      write (*,230) i
```

```
      write (3,230) i
230   format (/,5x,'line no.',i3,'  susceptance of line changing(pu)
     *=',$,3x,$)
      read (*,51) ysht(i)
      write (*,233) i
      write (3,233) i
233   format (/,5x,'line No.',i3,'  type of line = ',$)
      read (*,234) tl(i)
234   format (f5.2)
15    continue
      if (nt.eq.0)then
      go to 195
      endif
      write (*,250)
      write (3,250)
250   format (//,2x,'*** TRANSFORMER DATA ***')
      do 25 i=1,nt
      write (*,260) i
      write (3,260) i
260   format (//,2x,'** transformer no.',i3,' **',
     *//,15x,'connect from bus (e) =',$)
      read (*,20) mtb(i)
      write (*,270)
      write (3,270)
270   format (/,15x,'connect to bus (f) =',$)
      read (*,20) mab(i)
      write (*,280)
      write (3,280)
280   format (/,2x,' series resistance and reactance(pu) =',$,3x,$)
      read (*,51) ztser(i)
      write (*,300) i
      write (3,300) i
300   format (/,2x,'transformer no.',i3,' ratio = ',$)
      read (*,50) tr(i)
```

```
25    continue
195   continue
      if (nc.eq.0) then
      go to 145
      endif
      write (*,490)
      write (3,490)
490   format (//,2x,'*** SHUNT CAPACITER DATA ***')
      do 135 i=1,nc
      write (*,500) i
      write (3,500) i
500   format (/,10x,'** shunt cap. no.',i3,' **',
     *//,15x,'connect at bus (h) = ',$)
      read (*,20) nct(i)
      write (*,510) i
      write (3,510) i
510   format (/,5x,'shunt cap.no.',i3,' its shunt susceptance(pu) ='
     *,$,3x,$)
      read (*,51) ycap(i)
135   continue
145   continue
      write (*,350)
      write (3,350)
350   format (//,2x,'*** LIMIT OF BUS VOLTAGE AND GEN. REACTIVE ***' )
      do 45 i=1,nb
      qmax(i) = 0.0
      qmin(i) = 0.0
      vmax(i) = 0.0
      vmin(i) = 0.0
      write (*,360) i,ntb(i)
      write (3,360) i,ntb(i)
360   format (//,10x,'* bus no.',i3,'*  this bus type = ',i5)
      if (ntb(i).eq.1) then
      go to 390
```

```
      endif
      write (*,370) i
      write (3,370) i
370   format (/,10x,'bus no.',i3,' max. reactive power generate(mvar)
     *=',$)
      read (*,50) qmax(i)
      write (*,380) i
      write (3,380) i
380   format (/,10x,'bus no.',i3,' min. reactive power generate(mvar)
     *=',$)
      read (*,50) qmin(i)
390   continue
      write (*,400) i
      write (3,400) i
400   format (/,10x,'bus no.',i3,' max. voltage (pu) =',$)
      read (*,50) vmax(i)
      write (*,410) i
      write (3,410) i
410   format (/,10x,'bus no.',i3,' min. voltage (pu) =',$)
      read (*,50) vmin(i)
45    continue
      return
      end

c     *** subroutine to  print out system input data  ***
      subroutine pindat
      Integer i
      Real jacob
      complex zser,ztser,ycap,ysht,v,yser,ytser,y,vn,y1,s,ys,r,ts,ttr,
     *        y2,cz1,cz2,cz3,cz4,cz5,yp,yq,ycon,cz6,cz7
      common nvar,nc,pbase,error,alpha,lnit,ntb(30),zser(40),
     *        vmax(30),vmin(30),ycap(30),nct(30),mtb(5),power,
     *       vbase,tr(5),nb,nt,mab(5),pg(30),pd(30),qg(30),qd(30),
     *       qmax(30),qmin(30),nl,nsb(40),neb(40),ztser(5),
```

```
     *       ysht(40),v(30),yser(40),ytser(5),y(30,30),nload,ncon,
     *       noob,nld,nswg,nk(30),pdn(30),pgn(30),qdn(30),qgn(30),
     *       vn(30),nbtypn(30),qmaxn(30),qminn(30),vmaxn(30),vminn(30),
     *       mtbn(5),mabn(5),dp(30),dq(30),da(30),db(30),del(60),
     *       delv(60),g(30,30),h(30,30),aa(30),b(30),jacob(60,60),
     *       p(30),q(30),pcal(30),qcal(30),dmax,nit,cc(30),vmag(30),
     *       angle(30),t,s(40),yl,nsj,nj,nog,ngn,na,n,tplol,tqlol,r(40),
     *       ys,plol(40),qlol(40),tpol(5),tqol(5),yss(40),ts(5),ttr(5),
     *       y22(30),npst,nsbp(10),nebp(10),zeta(10),nsbpn(10),
     *       nebpn(10),nlp(10),yp(40),yq(40),cpst,cz1,cz2,cz3,cz4,cz5,
     *       ycon(40,40),nyy,cz6,cz7,fe(10),xm(30,30),zc(30,30),fed(10),
     *       xser(40),d(40,40),rser(40),pf(40),am(10,10),bm(10),cnol,
     *       amv(10,10),t1(40)
      write (*,101)
      write (3,101)
101   format (///,10x,'*********************************************')
      write (*,102)
      write (3,102)
102   format (/,10x,'** INPUT DATA OF ELECTRICAL POWER SYSTEM **')
      write (*,103)
      write (3,103)
103   format (/,10x,'*********************************************')
      write (*,104) nb,nl,nt,nvar,nc,pbase,error,alpha,lnit
      write (3,104) nb,nl,nt,nvar,nc,pbase,error,alpha,lnit
104   format (/,10x,'total bus = ',i5,/,10x,
     *'total line =',i5,/,10x,
     *'total transformer =',i5,/,10x,
     *'total var source bus =',i5,/,10x,
     *'total shunt capaciter =',i5,/,10x,
     *'base mva =',f8.2,/,10x,
     *'maximun error =',f8.6,/,10x,
     *'acceleration factor =',f8.3,/,10x,
     *'max. iteration in load flow acceptable =',i4)
      write (*,105)
```

```
      write (3,105)
105   format (///,2x,' *** BUS DATA ***',/)
      write (*,106)
      write (3,106)
106   format (2x,'|----|----|-------|---|-------|-------|-------|',
     *'-------|-------|')
      write (*,107)
      write (3,107)
107   format (2x,'|bus |bus | volt  |   | volt |   generation  |',
     *'      load     |')
      write (*,108)
      write (3,108)
108   format (2x,'|    |    |       |   | base |-------|-------|',
     *'-------|-------|')
      write (*,109)
      write (3,109)
109   format (2x,'| no.|type| (pu)  |deg| (kv) |   MW   |  MVAR |',
     *'  MW   |  MVAR |')
      write (*,106)
      write (3,106)
      do 111 i=1,nb
      write (*,110)i,ntb(i),v(i),vbase,pg(i),qg(i),pd(i),qd(i)
      write (3,110)i,ntb(i),v(i),vbase,pg(i),qg(i),pd(i),qd(i)
110   format (2x,'|',2(1x,i2,1x,'|'),f7.5,'|',f3.2,'|',5(f7.2,'|'))
111   continue
      write (*,106)
      write (3,106)
      write (*,112)
      write (3,112)
112   format (//,15x,'type = 1 -------> LOAD BUS',/,
     *15x,'type = 2 -------> VOLTAGE CONTROLLED BUS',/,
     *15X,'type = 3 -------> SWING BUS')
      write (*,113)
      write (3,113)
```

```
113   format (///,2x,'*** LINE DATA ***',/)
      write (*,114)
      write (3,114)
114   format (2x,'|------|------|------|--------|--------|',
     *'------------|')
      write (*,115)
      write (3,115)
115   format (2x,'| line | send |  end |    impedance    |',
     *'   line     |')
      write (*,116)
      write (3,116)
116   format (2x,'|      | bus  | bus  |--------|--------|',
     *' charging   |')
      write (*,117)
      write (3,117)
117   format (2x,'|  no. | (p)  | (q)  |    r   |   x    |',
     *'  (pu)      |')
      write (*,114)
      write (3,114)
      do 118 i=1,nl
      write (*,119) i,nsb(i),neb(i),zser(i),ysht(i)
      write (3,119) i,nsb(i),neb(i),zser(i),ysht(i)
119   format (2x,'|',3(' ',i2,'   |'),2(f7.4,1x,'|'),f3.2,'|',f7.4,
     *1x,'|')
118   continue
      write (*,114)
      write (3,114)
      if (nt.eq.0) then
      go to 150
      endif
      write (*,120)
      write (3,120)
120   format (//,'*** TRANSFORMER DATA ***',/)
      write (*,121)
```

```
      write (3,121)
121   format (2x,'|--------|------|------|--------|--------|',
     *'--------|')
      write (*,122)
      write (3,122)
122   format (2x,'| transf.| from |  to  |  tr. impedance  |',
     *' transf.|')
      write (*,123)
      write (3,123)
123   format (2x,'|   no.  | bus  | bus  |    r   |    x   |',
     *' ratio  |')
      write (*,121)
      write (3,121)
      do 124 i=1,nt
      write (*,125) i,mtb(i),mab(i),ztser(i),tr(i)
      write (3,125) i,mtb(i),mab(i),ztser(i),tr(i)
125   format (2x,'|','   ',i2,'   ','|',2('  ',i2,'  |'),3(f7.4,1x,'|'))
124   continue
      write (*,121)
      write (3,121)
150   continue
      if (nc.eq.0) then
      go to 170
      endif
      write (*,151)
      write (3,151)
151   format (///,2x,'*** SHUNT CAPACITER DATA ***',/)
      write (*,152)
      write (3,152)
152   format (2x,'|--------|---------|-------|-------|')
      write (*,153)
      write (3,153)
153   format (2x,'|  shunt | connect |     shunt     |')
      write (*,154)
```

```
      write (3,154)
154   format (2x,'|  cap.  |  at    |   susceptance |')
      write (*,155)
      write (3,155)
155   format (2x,'|  no.   |  bus   |   r   |   x   |')
      write (*,152)
      write (3,152)
      do 156 i=1,nc
      write (*,157) i,nct(i),ycap(i)
      write (3,157) i,nct(i),ycap(i)
157   format (2x,'|',3x,i2,3x,'|',3x,i2,4x,'|',f5.2,2x,'|',f7.5,'|')
156   continue
      write (*,152)
      write (3,152)
170   continue
      if (nvar.eq.0) then
      go to 190
      endif
      write (*,171)
      write (3,171)
171   format (///,2x,'*** VAR SOURCE DATA ***',/)
      write (*,173) nvar
      write (3,173) nvar
173   format (2x,'total number of var source  = ',i2)
190   continue
      write (*,191)
      write (3,191)
191   format (///,2x,'*** LIMIT OF BUS VOLTAGE AND GEN.REACTIVE ***',/)
      write (*,192)
      write (3,192)
192   format (2x,'|-----|-------|-------|-------|-------|')
      write (*,193)
      write (3,193)
193   format (2x,'| bus |    voltage    | Q - generate |')
```

```
      write (*,194)
      write (3,194)
194   format (2x,'|     |  max. |  min. |  max. |  min. |')
      write (*,195)
      write (3,195)
195   format (2x,'| no. |      (pu)     |     (MVAR)    |')
      write (*,192)
      write (3,192)
      do 200 i=1,nb
      write (*,196) i,vmax(i),vmin(i),qmax(i),qmin(i)
      write (3,196) i,vmax(i),vmin(i),qmax(i),qmin(i)
196   format (2x,'|',i3,2x,'|',4(f7.2,'|'))
200   continue
      write (*,192)
      write (3,192)
      return
      end
c     *** subroutine to input system data by disk reading mode ***
      subroutine didata
      character*8 finame
      Real jacob
      complex zser,ztser,ycap,ysht,v,yser,ytser,y,vn,yl,s,ys,r,ts,ttr,
     *        y2,cz1,cz2,cz3,cz4,cz5,yp,yq,ycon,cz6,cz7
      common nvar,nc,pbase,error,alpha,lnit,ntb(30),zser(40),
     *        vmax(30),vmin(30),ycap(30),nct(30),mtb(5),power,
     *       vbase,tr(5),nb,nt,mab(5),pg(30),pd(30),qg(30),qd(30),
     *       qmax(30),qmin(30),nl,nsb(40),neb(40),ztser(5),
     *       ysht(40),v(30),yser(40),ytser(5),y(30,30),nload,ncon,
     *       noob,nld,nswg,nk(30),pdn(30),pgn(30),qdn(30),qgn(30),
     *       vn(30),nbtypn(30),qmaxn(30),qminn(30),vmaxn(30),vminn(30),
     *       mtbn(5),mabn(5),dp(30),dq(30),da(30),db(30),del(60),
     *       delv(60),g(30,30),h(30,30),aa(30),b(30),jacob(60,60),
     *       p(30),q(30),pcal(30),qcal(30),dmax,nit,cc(30),vmag(30),
     *       angle(30),t,s(40),yl,nsj,nj,nog,ngn,na,n,tplol,tqlol,r(40),
```

```
*       ys,plol(40),qlol(40),tpol(5),tqol(5),yss(40),ts(5),ttr(5),
*       y22(30),npst,nsbp(10),nebp(10),zeta(10),nsbpn(10),
*       nebpn(10),nlp(10),yp(40),yq(40),cpst,cz1,cz2,cz3,cz4,cz5,
*       ycon(40,40),nyy,cz6,cz7,fe(10),xm(30,30),zc(30,30),fed(10),
*       xser(40),d(40,40),rser(40),pf(40),am(10,10),bm(10),cnol,
*       amv(10,10),tl(40)
      write (*,10)
      write (3,10)
10    format (//,10x,'** INPUT SYSTEM DATA DISK READING MODE **')
      write (*,15)
15    format (/,5x,'NAME OF FILE TO READ SYSTEM DATA',/,
     *5x,'(do not exceed 8 character)')
      write (*,20)
20    format (/,15x,'DATA FILE NAME =',$)
      read (*,25) finame
25    format (a8)
      open (4,file = finame ,status = 'old')
      write (*,30) finame
      write (3,30) finame
30    format (/,5x,'YOU GET SYSYEM DATA FROM FILE NAME = ',a8)
      read (4,40) nb,nl,nt,nvar,nc,pbase,error,alpha,vbase
40    format (5i5,3f12.6,f6.2)
      read (4,50) lnit
50    format (i5)
      do 70 i=1,nb
      read (4,60) ntb(i),v(i),pg(i),qg(i),pd(i),qd(i)
60    format (i5,f12.6,f3.2,4f12.6)
70    continue
      do 90 i=1,nl
      read (4,80) nsb(i),neb(i),zser(i),ysht(i),tl(i)
80    format (2i5,2f12.6,f3.2,f12.6,f5.2)
90    continue
      if (nt.eq.0) then
      go to 200
```

```
      endif
      do 110 i=1,nt
      read (4,100) mtb(i),mab(i),ztser(i),tr(i)
100   format (2i5,3f12.6)
110   continue
200   continue
      if (nc.eq.0) then
      go to 250
      endif
      do 220 i=1,nc
      read (4,210) nct(i),ycap(i)
210   format (i5,2f12.6)
220   continue
250   continue
      do 270 i=1,nb
      read (4,260) qmax(i),qmin(i),vmax(i),vmin(i)
260   format (4f12.6)
270   continue
      close (4)
      return
      end
c     *** subroutine to modifide input data to per unit base ***
c     *** and change impedance to admittance ***
      subroutine modat
      Real jacob
      complex zser,ztser,ycap,ysht,v,yser,ytser,y,vn,y1,s,ys,r,ts,ttr,
     *        y2,cz1,cz2,cz3,cz4,cz5,yp,yq,ycon,cz6,cz7
      common nvar,nc,pbase,error,alpha,lnit,ntb(30),zser(40),
     *        vmax(30),vmin(30),ycap(30),nct(30),mtb(5),power,
     *       vbase,tr(5),nb,nt,mab(5),pg(30),pd(30),qg(30),qd(30),
     *       qmax(30),qmin(30),nl,nsb(40),neb(40),ztser(5),
     *       ysht(40),v(30),yser(40),ytser(5),y(30,30),nload,ncon,
     *       noob,nld,nswg,nk(30),pdn(30),pgn(30),qdn(30),qgn(30),
     *       vn(30),nbtypn(30),qmaxn(30),qminn(30),vmaxn(30),vminn(30),
```

```
*       mtbn(5),mabn(5),dp(30),dq(30),da(30),db(30),del(60),
*       delv(60),g(30,30),h(30,30),aa(30),b(30),jacob(60,60),
*       p(30),q(30),pcal(30),qcal(30),dmax,nit,cc(30),vmag(30),
*       angle(30),t,s(40),y1,nsj,nj,nog,ngn,na,n,tplol,tqlol,r(40),
*       ys,plol(40),qlol(40),tpol(5),tqol(5),yss(40),ts(5),ttr(5),
*       y22(30),npst,nsbp(10),nebp(10),zeta(10),nsbpn(10),
*       nebpn(10),nlp(10),yp(40),yq(40),cpst,cz1,cz2,cz3,cz4,cz5,
*       ycon(40,40),nyy,cz6,cz7,fe(10),xm(30,30),zc(30,30),fed(10),
*       xser(40),d(40,40),rser(40),pf(40),am(10,10),bm(10),cnol,
*       amv(10,10),t1(40)
      write (*,10)
      write (3,10)
10    format (/,15x,'*** call modat ***')
      do 20 i=1,nb
      pd(i) = pd(i)/pbase
      qd(i) = qd(i)/pbase
      pg(i) = pg(i)/pbase
      qg(i) = qg(i)/pbase
20    continue
      do 30 i=1,nb
      qmax(i) = qmax(i)/pbase
      qmin(i) = qmin(i)/pbase
30    continue
      do 40 i=1,nl
      yser(i) = cmplx(1.0,0.0)/zser(i)
40    continue
      if (nt.eq.0) then
      go to 100
      endif
      do 50 i=1,nt
      ytser(i) = cmplx(1.0,0.0)/ztser(i)
50    continue
100   continue
      return
```

```
      end
c     *** subroutine to form bus admittance matrix ***
      subroutine ybus
      Real jacob
      complex zser,ztser,ycap,ysht,v,yser,ytser,y,vn,y1,s,ys,r,ts,ttr,
     *        y2,cz1,cz2,cz3,cz4,cz5,yp,yq,ycon,cz6,cz7
      common nvar,nc,pbase,error,alpha,lnit,ntb(30),zser(40),
     *        vmax(30),vmin(30),ycap(30),nct(30),mtb(5),power,
     *       vbase,tr(5),nb,nt,mab(5),pg(30),pd(30),qg(30),qd(30),
     *       qmax(30),qmin(30),nl,nsb(40),neb(40),ztser(5),
     *       ysht(40),v(30),yser(40),ytser(5),y(30,30),nload,ncon,
     *       noob,nld,nswg,nk(30),pdn(30),pgn(30),qdn(30),qgn(30),
     *       vn(30),nbtypn(30),qmaxn(30),qminn(30),vmaxn(30),vminn(30),
     *       mtbn(5),mabn(5),dp(30),dq(30),da(30),db(30),del(60),
     *       delv(60),g(30,30),h(30,30),aa(30),b(30),jacob(60,60),
     *       p(30),q(30),pcal(30),qcal(30),dmax,nit,cc(30),vmag(30),
     *       angle(30),t,s(40),y1,nsj,nj,nog,ngn,na,n,tplol,tqlol,r(40),
     *       ys,plol(40),qlol(40),tpol(5),tqol(5),yss(40),ts(5),ttr(5),
     *       y22(30),npst,nsbp(10),nebp(10),zeta(10),nsbpn(10),
     *       nebpn(10),nlp(10),yp(40),yq(40),cpst,cz1,cz2,cz3,cz4,cz5,
     *       ycon(40,40),nyy,cz6,cz7,fe(10),xm(30,30),zc(30,30),fed(10),
     *       xser(40),d(40,40),rser(40),pf(40),am(10,10),bm(10),cnol,
     *       amv(10,10),tl(40)
      write (*,10)
      write (3,10)
10    format (/,15x,'*** call ybus ***')
      do 20 i=1,nb
        do 30 j=1,nb
        y(i,j) = cmplx(0.0,0.0)
30      continue
20    continue
      do 40 i=1,nl
      l = nsb(i)
      m = neb(i)
```

```
      y(l,l) = y(l,l)+yser(i)+ysht(i)/2.0
      y(m,m) = y(m,m)+yser(i)+ysht(i)/2.0
      y(l,m) = y(l,m)-yser(i)
      y(m,l) = y(m,l)-yser(i)
40    continue
      if (nc.eq.0) then
      go to 60
      endif
      do 50 i=1,nc
      n = nct(i)
      y(n,n) = y(n,n)+ycap(i)
50    continue
60    continue
      if (nt.eq.0) then
      go to 100
      endif
      do 70 i=1,nt
      k = mab(i)
      j = mtb(i)
      y(k,k) = y(k,k)+ytser(i)
      y(j,j) = y(j,j)+ytser(i)*(tr(i)**2)
      y(k,j) = y(k,j)-ytser(i)*tr(i)
      y(j,k) = y(j,k)-ytser(i)*tr(i)
70    continue
100   continue
      if (cpst.eq.0) then
      go to 110
      endif
      do 260 i=1,npst
      la = nsbp(i)
      ma = nebp(i)
      ycon(la,la) = y(la,la)
      ycon(ma,ma) = y(ma,ma)
260   continue
```

```
      do 300 i=1,npst
      cz1 = cmplx(0.0,0.0)
      cz2 = cmplx(0.0,0.0)
      cz4 = cmplx(0.0,0.0)
      cz5 = cmplx(0.0,0.0)
      ll = nsbp(i)
      mm = nebp(i)
      nyy = nlp(i)
      cz1 = cmplx(cos(-zeta(i)),sin(-zeta(i)))
      cz2 = cmplx(cos(zeta(i)),sin(zeta(i)))
      cz3 = cmplx(1.0,0.0)
      cz4 = cz3-cz1
      cz5 = cz3-cz2
      yp(nyy) = yser(nyy)*cz4*v(mm)/v(ll)
      yq(nyy) = yser(nyy)*cz5*v(ll)/v(mm)
      y(ll,ll) = y(ll,ll)+yp(nyy)
      y(mm,mm) = y(mm,mm)+yq(nyy)
300   continue
110   continue
      if (cpst.eq.1) go to 240
        do 220 i=1,nb
        do 220 j=1,nb
        write (*,230) i,j,y(i,j)
        write (3,230) i,j,y(i,j)
230     format (/,5x,'Y(',i2,',',i2,')=',2f12.6)
220     continue
240     continue
      return
      end

c     *** subroutine to change order bus for NR. calculation ***
      subroutine cob
      Real jacob
      complex zser,ztser,ycap,ysht,v,yser,ytser,y,vn,yl,s,ys,r,ts,ttr,
```

```
*        y2,cz1,cz2,cz3,cz4,cz5,yp,yq,ycon,cz6,cz7
 common nvar,nc,pbase,error,alpha,lnit,ntb(30),zser(40),
*        vmax(30),vmin(30),ycap(30),nct(30),mtb(5),power,
*       vbase,tr(5),nb,nt,mab(5),pg(30),pd(30),qg(30),qd(30),
*       qmax(30),qmin(30),nl,nsb(40),neb(40),ztser(5),
*       ysht(40),v(30),yser(40),ytser(5),y(30,30),nload,ncon,
*       noob,nld,nswg,nk(30),pdn(30),pgn(30),qdn(30),qgn(30),
*       vn(30),nbtypn(30),qmaxn(30),qminn(30),vmaxn(30),vminn(30),
*       mtbn(5),mabn(5),dp(30),dq(30),da(30),db(30),del(60),
*       delv(60),g(30,30),h(30,30),aa(30),b(30),jacob(60,60),
*       p(30),q(30),pcal(30),qcal(30),dmax,nit,cc(30),vmag(30),
*       angle(30),t,s(40),y1,nsj,nj,nog,ngn,na,n,tplol,tqlol,r(40),
*       ys,plol(40),qlol(40),tpol(5),tqol(5),yss(40),ts(5),ttr(5),
*       y22(30),npst,nsbp(10),nebp(10),zeta(10),nsbpn(10),
*       nebpn(10),nlp(10),yp(40),yq(40),cpst,cz1,cz2,cz3,cz4,cz5,
*       ycon(40,40),nyy,cz6,cz7,fe(10),xm(30,30),zc(30,30),fed(10),
*       xser(40),d(40,40),rser(40),pf(40),am(10,10),bm(10),cnol,
*       amv(10,10),tl(40)
      write (*,10)
      write (3,10)
10    format (/,10x,'*** call cob ***')
      nload = 0
      ncon = 0
      noob = 1
      nld = 0
      do 405 i=1,nb
      if (ntb(i).eq.1) nload = nload+1
      if (ntb(i).eq.2) ncon = ncon+1
405   continue
      ngn = nload
415   continue
      if (ntb(noob).eq.1) go to 800
      if (ntb(noob).eq.2) go to 810
      if (ntb(noob).eq.3) go to 820
```

```
      go to 840
800   continue
      nld = nld+1
      n = nld
      go to 830
810   continue
      ngn = ngn+1
      n = ngn
      go to 830
820   continue
      nswg = noob
      n = nb
830   continue
      nk(n) = noob
      pdn(n) = pd(noob)
      pgn(n) = pg(noob)
      qdn(n) = qd(noob)
      qgn(n) = qg(noob)
      vn(n) = v(noob)
      nbtypn(n) = ntb(noob)
      qmaxn(n) = qmax(noob)
      qminn(n) = qmin(noob)
      vmaxn(n) = vmax(noob)
      vminn(n) = vmin(noob)
      if (nt.eq.0) go to 778
      do 708 i=1,nt
      if (mtb(i).eq.noob) mtbn(i) = n
      if (mab(i).eq.noob) mabn(i) = n
708   continue
778   continue
      if (npst.eq.0) go to 888
      do 790 i=1,npst
      if (nsbp(i).eq.noob) nsbpn(i) = n
      if (nebp(i).eq.noob) nebpn(i) = n
```

```
790   continue
888   continue
840   continue
      noob = noob+1
      if (noob.le.nb) go to 415
      write (*,301) nload,ncon
      write (3,301) nload,ncon
301   format (//,5x,'NO. OF LOAD BUS =',i5,/,
     *5x,'NO. OF GEN. BUS =',i5,/,
     *5x,'NO. OF SWING BUS = 1')
748   continue
      write (*,1120)
      write (3,1120)
1120  format (/,5x,'PASS CHANGE ORDER BUS ')
      return
      end

c     *** this subroutine is used to calculate loadflow ***
c     *** By NEWTON-RAPHSON mathod ***
      subroutine nrlf
      Real jacob
      complex zser,ztser,ycap,ysht,v,yser,ytser,y,vn,y1,s,ys,r,ts,ttr,
     *        y2,cz1,cz2,cz3,cz4,cz5,yp,yq,ycon,cz6,cz7
      common nvar,nc,pbase,error,alpha,lnit,ntb(30),zser(40),
     *        vmax(30),vmin(30),ycap(30),nct(30),mtb(5),power,
     *       vbase,tr(5),nb,nt,mab(5),pg(30),pd(30),qg(30),qd(30),
     *       qmax(30),qmin(30),nl,nsb(40),neb(40),ztser(5),
     *       ysht(40),v(30),yser(40),ytser(5),y(30,30),nload,ncon,
     *       noob,nld,nswg,nk(30),pdn(30),pgn(30),qdn(30),qgn(30),
     *       vn(30),nbtypn(30),qmaxn(30),qminn(30),vmaxn(30),vminn(30),
     *       mtbn(5),mabn(5),dp(30),dq(30),da(30),db(30),del(60),
     *       delv(60),g(30,30),h(30,30),aa(30),b(30),jacob(60,60),
     *       p(30),q(30),pcal(30),qcal(30),dmax,nit,cc(30),vmag(30),
     *       angle(30),t,s(40),y1,nsj,nj,nog,ngn,na,n,tplol,tqlol,r(40),
```

```
     *       ys,plol(40),qlol(40),tpol(5),tqol(5),yss(40),ts(5),ttr(5),
     *       y22(30),npst,nsbp(10),nebp(10),zeta(10),nsbpn(10),
     *       nebpn(10),nlp(10),yp(40),yq(40),cpst,cz1,cz2,cz3,cz4,cz5,
     *       ycon(40,40),nyy,cz6,cz7,fe(10),xm(30,30),zc(30,30),fed(10),
     *       xser(40),d(40,40),rser(40),pf(40),am(10,10),bm(10),cno1,
     *       amv(10,10),t1(40)
      write (*,10)
      write (3,10)
10    format (/,10x,'*** call nrlf ***')
      write (*,20)
20    format (//,10x,'*** NOW LOAD FLOW IS RUNNING ***')
      do 22 i=1,nb
      dp(i) = 0.0
      dq(i) = 0.0
      da(i) = 0.0
      db(i) = 0.0
22    continue
      do 14  i=1,2*nb
      del(i) = 0.0
      delv(i) = 0.0
14    continue
      nsj = nb-1
      nj = 2*nsj
      n = nload+nsj
      nog = ncon
      ngn = nload
      nit = 0
      nld = 0
      t = 0.0
      na = nsj+nb
38    continue
      do 341 i=1,nb
      dr = real(vn(i))
      di = aimag(vn(i))
```

```
      aa(i) = cabs(vn(i))
      if (di.lt.0.0) b(i) = 4.172388980
      if (di.gt.0.0) b(i) = 1.570796327
      if (di.eq.0.0) b(i) = 0.0
      if (dr.ne.0.0) b(i) = atan2(di,dr)
341   continue
342   continue
      do 1022 i=1,nb
      do 1022 j=1,nb
      g(i,j) = 0.0
      h(i,j) = 0.0
1022  continue
      do 332 i=1,nb
      k = nk(i)
      do 332 j=1,nb
      l = nk(j)
      dr = real(y(k,l))
      di = aimag(y(k,l))
      g(i,j) = cabs(y(k,l))
      if (di.lt.0.0) h(i,j) = -4.712388980
      if (di.gt.0.0) h(i,j) = -1.570796327
      if (di.eq.0.0) h(i,j) = 0.0
      if (dr.ne.0.0) h(i,j) = -atan2(di,dr)
332   continue
      do 301 i=1,nj
      do 301 j=1,nj
      jacob(i,j) = 0.0
301   continue
      dmax = 0.0
      do 351 i=1,nb
      p(i) = pgn(i)-pdn(i)
      q(i) = qgn(i)-qdn(i)
      pcal(i) = 0.0
      qcal(i) = 0.0
```

```
351   continue
      do 352 i=1,nsj
      do 352 j=1,nb
      di = aa(i)*aa(j)*g(i,j)
      dr = h(i,j)+b(i)-b(j)
      pcal(i) = pcal(i)+di*cos(dr)
      qcal(i) = qcal(i)+di*sin(dr)
352   continue
      do 361 i=1,nsj
      dp(i) = p(i)-pcal(i)
      if (abs(dp(i)).gt.dmax) dmax = abs(dp(i))
361   continue
      do 362 i=1,nload
      dq(i) = q(i)-qcal(i)
      if (abs(dq(i)).gt.dmax) dmax = abs(dq(i))
362   continue
c     * find J1
      do 371 i=1,nsj
      do 371 j=1,nsj
      if (i.eq.j) go to 3771
      jacob(i,j) = aa(i)*aa(j)*g(i,j)*sin(h(i,j)+b(i)-b(j))
      go to 371
3771  continue
      do 1 iq=1,nb
      if (iq.eq.i) go to 1
      jacob(i,i)=jacob(i,i)-aa(i)*aa(iq)*g(i,iq)*sin(h(i,iq)+b(i)-b(iq))
1     continue
371   continue
c     * find J2
      do 372 i=1,nsj
      do 372 j=1,nload
      jj = nsj+j
      if (i.eq.j) go to 3772
      jacob(i,jj) = aa(i)*g(i,j)*cos(h(i,j)+b(i)-b(j))
```

```
      go to 372
3772  continue
      do 2 iq=1,nb
      if (iq.eq.i) go to 2
      jacob(i,jj) = jacob(i,jj)+aa(iq)*g(i,iq)*cos(h(i,iq)+b(i)-b(iq))
2     continue
      jacob(i,jj) = (2.0*aa(i)*g(i,i)*cos(h(i,i)))+jacob(i,jj)
372   continue
c     * find J3
      do 373 i=1,nload
      do 373 j=1,nsj
      ii = nsj+i
      if (i.eq.j) go to 3773
      jacob(ii,j) = -aa(i)*aa(j)*g(i,j)*cos(h(i,j)+b(i)-b(j))
      go to 373
3773  continue
      do 3 iq=1,nb
      if (iq.eq.i) go to 3
      jacob(ii,j) = jacob(ii,j)+aa(i)*aa(iq)*g(i,iq)*cos(h(i,iq)+
     *b(i)-b(iq))
3     continue
373   continue
c     * find J4
      do 374 i=1,nload
      do 374 j=1,nload
      ii = nsj+i
      jj = nsj+j
      if (i.eq.j) go to 3774
      jacob(ii,jj) = aa(i)*g(i,j)*sin(h(i,j)+b(i)-b(j))
      go to 374
3774  continue
      do 4 iq=1,nb
      if (iq.eq.i) go to 4
      jacob(ii,jj) = jacob(ii,jj)+aa(iq)*g(i,iq)*sin(h(i,iq)+b(i)-b(iq))
```

```
4       continue
        jacob(ii,jj) = (2.0*aa(i)*g(i,i)*sin(h(i,i)))+jacob(ii,jj)
374     continue
        if (dmax.lt.error) go to 3333
48      continue
        do 6666 i=1,nsj
        del(i) = dp(i)
6666    continue
        do 7777 i=1,nload
        ii = nsj+i
        del(ii) = dq(i)
7777    continue
        call jordan(jacob,del,delv,n)
        do 412 i=1,nsj
        db(i) = delv(i)
        ii = nsj+i
        da(i) = delv(ii)
        aa(i) = aa(i)+da(i)
        b(i)  = b(i)+db(i)
412     continue
        if (nog.eq.0) go to 421
        i = nload+1
        j = nload+nog
        do 422 noob=i,j
        if (nbtypn(noob).eq.2) go to 431
        go to 422
431     continue
        q(noob) = 0.0
        aa(i) = real(vn(i))
        do 432 jj=1,nb
        dr = h(noob,jj)-b(jj)+b(noob)
        q(noob) = q(noob)+aa(noob)*aa(jj)*g(noob,jj)*sin(dr)
432     continue
        qgn(noob) = q(noob)+qdn(noob)
```

```
      if (qgn(noob)-qmaxn(noob)) 441,422,442
441   continue
      if (qgn(noob)-qminn(noob)) 451,422,422
442   continue
      qgn(noob) = qmaxn(noob)
      go to 422
451   continue
      qgn(noob) = qminn(noob)
422   continue
421   continue
      if (cpst.eq.0) then
      go to 910
      endif
      do 920 i=1,npst
      la = nsbp(i)
      ma = nebp(i)
      y(la,la) = ycon(la,la)
      y(ma,ma) = ycon(ma,ma)
920   continue
910   continue
      if (cpst.eq.0) then
      go to 800
      endif
      do 810 i=1,npst
      cz1 = cmplx(0.0,0.0)
      cz2 = cmplx(0.0,0.0)
      cz4 = cmplx(0.0,0.0)
      cz5 = cmplx(0.0,0.0)
      cz6 = cmplx(0.0,0.0)
      cz7 = cmplx(0.0,0.0)
      ll = nsbp(i)
      mm = nebp(i)
      lp = nsbpn(i)
      mp = nebpn(i)
```

```
      nyy = nlp(i)
      cz1 = cmplx(cos(-zeta(i)),sin(-zeta(i)))
      cz2 = cmplx(cos(zeta(i)),sin(zeta(i)))
      cz3 = cmplx(1.0,0.0)
      cz4 = cz3-cz1
      cz5 = cz3-cz2
      rwx = b(lp)-b(mp)
      rwy = b(mp)-b(lp)
      cz6 = cmplx(cos(rwy),sin(rwy))
      cz7 = cmplx(cos(rwx),sin(rwx))
      yp(nyy) = yser(nyy)*cz4*aa(mp)/aa(lp)*cz6
      yq(nyy) = yser(nyy)*cz5*aa(lp)/aa(mp)*cz7
      y(ll,ll) = y(ll,ll)+yp(nyy)
      y(mm,mm) = y(mm,mm)+yq(nyy)
810   continue
800   continue
      nit = nit+1
      if (nit.gt.lnit) go to 4444
      go to 342
3333  continue
      do 561 i=1,nb
      k = nk(i)
      cc(k) = aa(i)
      db(k) = b(i)
      dq(k) = qgn(i)
561   continue
      do 562 i=1,nb
      vmag(i) = cc(i)
      angle(i) = db(i)
      qg(i) = dq(i)
562   continue
      do 571 i=1,nb
      di = vmag(i)*sin(angle(i))
      dr = vmag(i)*cos(angle(i))
```

```
      v(i) = cmplx(dr,di)
571   continue
      if (ncon.eq.0) go to 572
      do 581 i=1,nb
      if (ntb(i)-2) 581,582,581
582   continue
      y1 = cmplx(0.0,0.0)
      do 591 j=1,nb
      y1 = y1+y(i,j)*v(j)
591   continue
      y1 = y1*conjg(v(i))
      qg(i) = -aimag(y1)+qd(i)
581   continue
572   continue
      y1 = cmplx(0.0,0.0)
      do 592 i=1,nb
      y1 = y1+y(nswg,i)*v(i)
592   continue
      y1 = y1*conjg(v(nswg))
      p(nswg) = real(y1)
      q(nswg) = -aimag(y1)
      pg(nswg) = p(nswg)+pd(nswg)
      qg(nswg) = q(nswg)+qd(nswg)
      pgn(nb) = pg(nswg)
      qgn(nb) = qg(nswg)
      pdn(nb) = pd(nswg)
      qdn(nb) = qd(nswg)
      go to 601
4444  continue
      write (*,602 ) lnit
      write (3,602) lnit
602   format (//,2x,'the solution is not converged in',i3,'iteration')
      t = t+1
601   continue
```

```
      return
      end
c     *** this subroutine is used to solve loadflow variable ***
c     *** by GAUSS JORDAN elemination ***
      subroutine jordan (s,tt,x,nn)
      dimension s(60,60),tt(60),x(60)
      do 10 i=1,nn
      d = s(i,i)
      do 20 j=1,nn
      s(i,j) = s(i,j)/d
20    continue
      tt(i) = tt(i)/d
      do 30 j=1,nn
      if (i.eq.j) go to 30
      tt(j) = tt(j)-tt(i)*s(j,i)
30    continue
      do 40 k = 1,nn
      if (k.eq.i) go to 40
      d = s(k,i)
      do 50 j=1,nn
      s(k,j) = s(k,j)-s(i,j)*d
50    continue
40    continue
10    continue
      do 60 i=1,nn
      x(i) = tt(i)
60    continue
      return
      end
c     *** this routine is used to calculate power flow in lines, ***
c     *** transformer,power loss,line charging,static capaciter,***
c     *** mismatch and print output ***
      subroutine output
      Real jacob
```

```
      complex zser,ztser,ycap,ysht,v,yser,ytser,y,vn,y1,s,ys,r,ts,ttr,
     *        y2,cz1,cz2,cz3,cz4,cz5,yp,yq,ycon,cz6,cz7
      common nvar,nc,pbase,error,alpha,lnit,ntb(30),zser(40),
     *        vmax(30),vmin(30),ycap(30),nct(30),mtb(5),power,
     *       vbase,tr(5),nb,nt,mab(5),pg(30),pd(30),qg(30),qd(30),
     *       qmax(30),qmin(30),nl,nsb(40),neb(40),ztser(5),
     *       ysht(40),v(30),yser(40),ytser(5),y(30,30),nload,ncon,
     *       noob,nld,nswg,nk(30),pdn(30),pgn(30),qdn(30),qgn(30),
     *       vn(30),nbtypn(30),qmaxn(30),qminn(30),vmaxn(30),vminn(30),
     *       mtbn(5),mabn(5),dp(30),dq(30),da(30),db(30),del(60),
     *       delv(60),g(30,30),h(30,30),aa(30),b(30),jacob(60,60),
     *       p(30),q(30),pcal(30),qcal(30),dmax,nit,cc(30),vmag(30),
     *       angle(30),t,s(40),y1,nsj,nj,nog,ngn,na,n,tplol,tqlol,r(40),
     *       ys,plol(40),qlol(40),tpol(5),tqol(5),yss(40),ts(5),ttr(5),
     *       y22(30),npst,nsbp(10),nebp(10),zeta(10),nsbpn(10),
     *       nebpn(10),nlp(10),yp(40),yq(40),cpst,cz1,cz2,cz3,cz4,cz5,
     *       ycon(40,40),nyy,cz6,cz7,fe(10),xm(30,30),zc(30,30),fed(10),
     *       xser(40),d(40,40),rser(40),pf(40),am(10,10),bm(10),cnol,
     *       amv(10,10),tl(40)
      write (*,10)
      write (3,10)
10    format (/,10x,'*****************************************')
      write (*,20)
      write (3,20)
20    format (/,10x,'***  NEWTON-RAPHSON LOADFLOW SOLUTION ***')
      write (*,10)
      write (3,10)
      write (*,603) nit
      write (3,603) nit
603   format (//,2x,'this solution is converged in',i3,'  iteration')
      write (*,604)
      write (3,604)
604   format(///,2x,'*** VOLTAGE AND POWER GENERATION ***',/)
      write (*,613)
```

```
      write (3,613)
613   format (1x,'|-----|-----|-------|-------|-------|',
     *'-------|-------|-------|-------|')
      write (*,614)
      write (3,614)
614   format (1x,'| bus | bus |    bus voltage        |',
     *'  generation   |     load      |')
      write (*,623)
      write (3,623)
623   format (1x,'| no. |type |  pu   |   kv  | deg. |',
     *'  MW   |  MVAR |  MW   | MVAR  |')
      write (*,613)
      write (3,613)
      do 633 i=1,nb
      vd = vmag(i)
      vk = vd*vbase
      dl = angle(i)*57.29578
      gp = pg(i)*pbase
      gq = qg(i)*pbase
      dpp = pd(i)*pbase
      dqq = qd(i)*pbase
      write (*,634) i,ntb(i),vd,vk,dl,gp,gq,dpp,dqq
      write (3,634) i,ntb(i),vd,vk,dl,gp,gq,dpp,dqq
634   format (1x,'|',2(1x,i3,1x,'|'),f7.3,'|',6(f7.2,'|'))
633   continue
      write (*,613)
      write (3,613)
      write (*,643)
      write (3,643)
643   format (///,2x,'*** LINE FLOW ***',/)
      write (*,644)
      write (3,644)
644   format (1x,'|-----|-----|-----|-------|-------|-------|-------|',
     *'-------|-------|-------|')
```

```
      write (*,653)
      write (3,653)
653   format (1x,'|line |from |  to |flow from bus p| flow to bus q |',
     *'  line loss    |  line |')
      write (*,654)
      write (3,654)
654   format (1x,'|     | bus | bus |-------|-------|-------|-------|',
     *'-------|-------|charg. |')
      write (*,663)
      write (3,663)
663   format (1x,'| no. |  p  |  q  |  MW   |  MVAR |  MW   |  MVAR |',
     *'  MW   |  MVAR | MVAR  |')
      write (*,644)
      write (3,644)
      tplol = 0.0
      tqlol = 0.0
      ys = cmplx(0.0,0.0)
      do 673 i=1,nl
      l = nsb(i)
      m = neb(i)
      s(i) = v(l)*conjg((v(l)-v(m))*yser(i)+v(l)*ysht(i)/2.0+v(l)*yp(i))
      r(i) = v(m)*conjg((v(m)-v(l))*yser(i)+v(m)*ysht(i)/2.0+v(m)*yq(i))
      if (cpst.eq.0) then
      pf(i) =  -real(r(i))
      endif
      plol(i) = (abs(abs(real(s(i)))-abs(real(r(i)))))*pbase
      qlol(i) = (abs(abs(aimag(s(i)))-abs(aimag(r(i)))))*pbase
      ax = real(s(i))*pbase
      ay = aimag(s(i))*pbase
      az = real(r(i))*pbase
      ry = aimag(r(i))*pbase
      yss(i) = (aimag(ysht(i)*(cabs(v(m))**2+cabs(v(l))**2)/2.0))*pbase
      ys = ys+cmplx(0.0,yss(i))
      write (*,674) i,nsb(i),neb(i),ax,ay,az,ry,plol(i),qlol(i),yss(i)
```

```
      write (3,674) i,nsb(i),neb(i),ax,ay,az,ry,plol(i),qlol(i),yss(i)
674   format (1x,'|',3(1x,i3,1x,'|'),7(f7.2,'|'))
673   continue
      write (*,644)
      write (3,644)
      if (nt.eq.0) go to 683
      write (*,218)
      write (3,218)
218   format(///,2x,'*** POWER FLOW IN TRANSFORMER ***',/)
      write (*,693)
      write (3,693)
693   format (1x,'|-----|-----|-----|-------|-------|-------|-------|',
     *'-------|-------|')
      write (*,694)
      write (3,694)
694   format (1x,'|trans|from |  to |flow from bus e| flow to bus f |',
     *' transf. loss  |')
      write (*,703)
      write (3,703)
703   format (1x,'|     | bus | bus |-------|-------|-------|-------|',
     *'-------|-------|')
      write (*,704)
      write (3,704)
704   format (1x,'| no. |  e  |  f  |  MW   | MVAR  |  MW   | MVAR |',
     *'  MW   |  MVAR |')
      write (*,693)
      write (3,693)
      do 684 i=1,nt
      n = mtb(i)
      m = mab(i)
      ts(i) = v(n)*tr(i)*conjg((v(n)*tr(i)-v(m))*ytser(i))
      ttr(i) = v(m)*conjg((v(m)-v(n)*tr(i))*ytser(i))
      tpol(i) = (abs(abs(real(ts(i)))-abs(real(ttr(i)))))*pbase
      tqol(i) = (abs(abs(aimag(ts(i)))-abs(aimag(ttr(i)))))*pbase
```

```
      px = real(ts(i))*pbase
      py = aimag(ts(i))*pbase
      pz = real(ttr(i))*pbase
      pe = aimag(ttr(i))*pbase
      tplol = tplol+tpol(i)
      tqlol = tqlol+tqol(i)
      write (*,713) i,mtb(i),mab(i),px,py,pz,pe,tpol(i),tqol(i)
      write (3,713) i,mtb(i),mab(i),px,py,pz,pe,tpol(i),tqol(i)
713   format (1x,'|',3(1x,i3,1x,'|'),6(f7.2,'|'))
684   continue
      write (*,693)
      write (3,693)
683   continue
      do 733 i=1,nl
      tplol = tplol+plol(i)
      tqlol = tqlol+qlol(i)
733   continue
      y2 = cmplx(0.0,0.0)
      if (nc.eq.0) go to 734
      write (*,208)
      write (3,208)
208   format (///,2x,'*** POWER FLOW IN SHUNT CAPACITOR ***',/)
      write (*,743)
      write (3,743)
743   format (1x,'|-----|-------|-------|')
      write (*,744)
      write (3,744)
744   format (1x,'|shunt|connect|   Q   |')
      write (*,753)
      write (3,753)
753   format (1x,'| cap.|  at   |  flow |')
      write (*,754)
      write (3,754)
754   format (1x,'|  no.|  bus  |  MVAR |')
```

```
      write (*,743)
      write (3,743)
      do 763 i=1,nc
      n = nct(i)
      y22(i) = (aimag(ycap(i)*cabs(v(n))**2))*pbase
      y2 = y2+cmplx(0.0,y22(i))
      write (*,764) i,nct(i),y22(i)
      write (3,764) i,nct(i),y22(i)
764   format (1x,'|',1x,i3,1x,'|',2x,i3,2x,'|',f7.2,'|')
763   continue
      write (*,743)
      write (3,743)
734   continue
      sax = 0.0
      say = 0.0
      saz = 0.0
      sry = 0.0
      do 783 i=1,nb
      sax = sax+pg(i)*pbase
      say = say+qg(i)*pbase
      sry = sry+pd(i)*pbase
      saz = saz+qd(i)*pbase
783   continue
      power = tplol
      write (*,784)
      write (3,784)
784   format (//,25x,'*** TOTAL SYSTEM SOLUTION ***')
      write (*,793)
      write (3,793)
793   format (/,t34,'MW',t45,'MVAR')
      write (*,794) sax,say
      write (3,794) sax,say
794   format (/,10x,'system generation',t30,2(f7.2,5x))
      write (*,803) sry,saz
```

```
      write (3,803) sry,ssz
803   format (/,10x,'system load',t30,2(f7.2,5x))
      write (*,804) ys
      write (3,804) ys
804   format (/,10x,'line charging',t30,2(f7.2,5x))
      write (*,813) y2
      write (3,813) y2
813   format (/,10x,'shunt capacitor ',t30,2(f7.2,5x))
      ys = ys+y2
      ssx = ssx-sry-real(ys)
      ssy = ssy-ssz+simag(ys)
      ssx = abs(ssx-tplol)
      ssy = abs(ssy-tqlol)
      write (*,814) tplol,tqlol
      write (3,814) tplol,tqlol
814   format (/,10x,'system loss',t30,2(f8.4,5x))
      write (*,823) ssx,ssy
      write (3,823) ssx,ssy
823   format (/,10x,'mismatch',t30,2(f7.4,5x))
      return
      end
c     *** this subroutine is used to input data of phase shifter ***
      subroutine pster
      Real jacob
      complex zser,ztser,ycap,ysht,v,yser,ytser,y,vn,y1,s,ys,r,ts,ttr,
     *        y2,cz1,cz2,cz3,cz4,cz5,yp,yq,ycon,cz6,cz7
      common nvar,nc,pbase,error,alpha,lnit,ntb(30),zser(40),
     *        vmax(30),vmin(30),ycap(30),nct(30),mtb(5),power,
     *       vbase,tr(5),nb,nt,mab(5),pg(30),pd(30),qg(30),qd(30),
     *       qmax(30),qmin(30),nl,nsb(40),neb(40),ztser(5),
     *       ysht(40),v(30),yser(40),ytser(5),y(30,30),nload,ncon,
     *       noob,nld,nswg,nk(30),pdn(30),pgn(30),qdn(30),qgn(30),
     *       vn(30),nbtypn(30),qmaxn(30),qminn(30),vmaxn(30),vminn(30),
     *       mtbn(5),mabn(5),dp(30),dq(30),da(30),db(30),del(60),
```

```
*       delv(60),g(30,30),h(30,30),aa(30),b(30),jacob(60,60),
*       p(30),q(30),pcal(30),qcal(30),dmax,nit,cc(30),vmag(30),
*       angle(30),t,s(40),y1,nsj,nj,nog,ngn,na,n,tplol,tqlol,r(40),
*       ys,plol(40),qlol(40),tpol(5),tqol(5),yss(40),ts(5),ttr(5),
*       y22(30),npst,nsbp(10),nebp(10),zeta(10),nsbpn(10),
*       nebpn(10),nlp(10),yp(40),yq(40),cpst,cz1,cz2,cz3,cz4,cz5,
*       ycon(40,40),nyy,cz6,cz7,fe(10),xm(30,30),zc(30,30),fed(10),
*       xser(40),d(40,40),rser(40),pf(40),am(10,10),bm(10),cnol,
*       amv(10,10),tl(40)

      do 700 i=1,40
      yp(i) = cmplx(0.0,0.0)
      yq(i) = cmplx(0.0,0.0)
700   continue
      do 710 i=1,40
      do 720 j=1,40
      ycon(i,j) = cmplx(0.0,0.0)
720   continue
710   continue
      do 750 i=1,10
      nlp(i) = 0
      nsbp(i) = 0
      nebp(i) = 0
      nsbpn(i) = 0
      nebpn(i) = 0
      zeta(i)= 0.0
750   continue
      write (*,10)
10    format (/,5x,'** NOW WE WILL INSTALL THE PHASE SHIFTER **',//,
     *5x,'!! input the data of phase shifters !!')
      write (*,20)
20    format (/,5x,'number of phase shifter = ',$)
      read (*,25) npst
25    format (i2)
```

```
      if (npst.ne.0) then
      cpst = 1
      else
      cpst = 0
      endif
      do 70 i=1,npst
      write (*,27) i
27    format (/,5x,'NO. of line that install pst. no.',i2,' is =',$)
      read (*,25) nlp(i)
      nsbp(i) = nsb(nlp(i))
      nebp(i) = neb(nlp(i))
70    continue
      return
      end

c     *** this subroutine is used to form x - d parameter ***
      subroutine  xdcal
      Real jacob
      complex zser,ztser,ycap,ysht,v,yser,ytser,y,vn,yl,s,ys,r,ts,ttr,
     *        y2,cz1,cz2,cz3,cz4,cz5,yp,yq,ycon,cz6,cz7
      common nvar,nc,pbase,error,alpha,lnit,ntb(30),zser(40),
     *        vmax(30),vmin(30),ycap(30),nct(30),mtb(5),power,
     *       vbase,tr(5),nb,nt,mab(5),pg(30),pd(30),qg(30),qd(30),
     *       qmax(30),qmin(30),nl,nsb(40),neb(40),ztser(5),
     *       ysht(40),v(30),yser(40),ytser(5),y(30,30),nload,ncon,
     *       noob,nld,nswg,nk(30),pdn(30),pgn(30),qdn(30),qgn(30),
     *       vn(30),nbtypn(30),qmaxn(30),qminn(30),vmaxn(30),vminn(30),
     *       mtbn(5),mabn(5),dp(30),dq(30),da(30),db(30),del(60),
     *       delv(60),g(30,30),h(30,30),aa(30),b(30),jacob(60,60),
     *       p(30),q(30),pcal(30),qcal(30),dmax,nit,cc(30),vmag(30),
     *       angle(30),t,s(40),yl,nsj,nj,nog,ngn,na,n,tplol,tqlol,r(40),
     *       ys,plol(40),qlol(40),tpol(5),tqol(5),yss(40),ts(5),ttr(5),
     *       y22(30),npst,nsbp(10),nebp(10),zeta(10),nsbpn(10),
     *       nebpn(10),nlp(10),yp(40),yq(40),cpst,cz1,cz2,cz3,cz4,cz5,
```

```
     *       ycon(40,40),nyy,cz6,cz7,fe(10),xm(30,30),zc(30,30),fed(10),
     *       xser(40),d(40,40),rser(40),pf(40),am(10,10),bm(10),cnol,
     *       amv(10,10),tl(40)
      integer i,j,m,ip,iq,l,ii,jj,ix,jx,pp,qq

      write (*,10)
      write (3,10)
10    format (/,10x,'*** call xdcal ***')
      do 20 i=1,nl
      rser(i) = real(zser(i))
      xser(i) = aimag(zser(i))
20    continue
      write (*,101)
      write (3,101)
101   format (/,5x,'** THIS IS REACTANCE MATRIX **')
      m = 1
      do 999 i=1,nl
      ip = nsb(i)
      iq = neb(i)
      if (ip.ne.1) go to 500
      if (tl(i).eq.1.0) then
      go to 120
      else
      go to 130
      endif
c     ** branch , ip = 1 **
120   continue
      m = m+1
      do 122 ii=1,m
      if (ii.eq.iq) go to 124
      xm(iq,ii) = 0.0
      xm(ii,iq) = xm(iq,ii)
      xm(iq,iq) = xser(i)
124   continue
```

```
122   continue
      go to 999
c     ** link , ip = 1 **
130   continue
      l = m+1
      do 131 ii=1,m
      zc(l,ii) = -xm(iq,ii)
      zc(ii,l) = zc(l,ii)
131   continue
      zc(l,l) = -zc(iq,l)+xser(i)
      do 132 ix=1,m
      do 133 jx=1,m
      xm(ix,jx) = xm(ix,jx)-(zc(ix,l)*zc(l,jx)/zc(l,l))
133   continue
132   continue
      go to 999
500   continue
      if (tl(i).eq.1.0) then
      go to 520
      else
      go to 510
      endif
c     ** link , ip not = 1 **
510   continue
      l = m+1
      do 511 ii=1,m
      zc(l,ii) = xm(ip,ii)-xm(iq,ii)
      zc(ii,l) = zc(l,ii)
511   continue
      zc(1,1) = zc(ip,l)-zc(iq,l)+xser(i)
      do 512 ix=1,m
      do 513 jx=1,m
      xm(ix,jx) = xm(ix,jx)-(zc(ix,l)*zc(l,jx)/zc(l,l))
513   continue
```

```
512   continue
      go to 999
c     ** branch , ip not = 1 **
520   continue
      m = m+1
      do 521 ii=1,m
      if (ii.eq.iq) go to 522
      xm(iq,ii) = xm(ip,ii)
      xm(ii,iq) = xm(iq,ii)
      xm(iq,iq) = xm(ip,iq)+xser(i)
522   continue
521   continue
      go to 999
999   continue
      do 1000 i=1,m
      do 1000 j=1,m
      write (*,1001) i,j,xm(i,j)
      write (3,1001) i,j,xm(i,j)
1001  format (/,5x,'XM(',i2,',',i2,')=',f10.6)
1000  continue
      write (*,1201)
      write (3,1201)
1201  format (/,5x,'** THIS IS ALL D(pq,ij) FOR ALL LINE **')
c     * J is line of phase shifter *
c     * I is line of line *
      do 2200 j=1,nl
      do 2201 i=1,nl
      ii = nsb(j)
      jj = neb(j)
      pp = nsb(i)
      qq = neb(i)
      if (i.eq.j) go to 1700
      d(i,j) = (-1/(xser(j)*xser(i)))*(xm(ii,pp)+xm(jj,qq)-xm(ii,qq)
     *          -xm(jj,pp))
```

```
      write (*,1500) pp,qq,ii,jj,d(i,j)
      write (3,1500) pp,qq,ii,jj,d(i,j)
1500  format (/,5x,'D(',i2,'-',i2,',',i2,'-',i2,')=',f9.4)
      go to 1999
1700  continue
      d(i,j) = (1/xser(j)**2)*(xser(j)-xm(ii,ii)-xm(jj,jj)+2*xm(ii,jj))
      write (*,1750) pp,qq,ii,jj,d(i,j)
      write (3,1750) pp,qq,ii,jj,d(i,j)
1750  format (/,5x,'D(',i2,'-',i2,',',i2,'-',i2,')=',f9.4)
1999  continue
2201  continue
2200  continue
      return
      end

c     *** this subroutine is used to calculate optimum angle  ***
c     *** of any phase shifter for loss minimization ***
      subroutine fecal
      Real jacob
      complex zser,ztser,ycap,ysht,v,yser,ytser,y,vn,yl,s,ys,r,ts,ttr,
     *        y2,cz1,cz2,cz3,cz4,cz5,yp,yq,ycon,cz6,cz7
      common nvar,nc,pbase,error,alpha,lnit,ntb(30),zser(40),
     *        vmax(30),vmin(30),ycap(30),nct(30),mtb(5),power,
     *       vbase,tr(5),nb,nt,mab(5),pg(30),pd(30),qg(30),qd(30),
     *       qmax(30),qmin(30),nl,nsb(40),neb(40),ztser(5),
     *       ysht(40),v(30),yser(40),ytser(5),y(30,30),nload,ncon,
     *       noob,nld,nswg,nk(30),pdn(30),pgn(30),qdn(30),qgn(30),
     *       vn(30),nbtypn(30),qmaxn(30),qminn(30),vmaxn(30),vminn(30),
     *       mtbn(5),mabn(5),dp(30),dq(30),da(30),db(30),del(60),
     *       delv(60),g(30,30),h(30,30),aa(30),b(30),jacob(60,60),
     *       p(30),q(30),pcal(30),qcal(30),dmax,nit,cc(30),vmag(30),
     *       angle(30),t,s(40),yl,nsj,nj,nog,ngn,na,n,tplol,tqlol,r(40),
     *       ys,plol(40),qlol(40),tpol(5),tqol(5),yss(40),ts(5),ttr(5),
     *       y22(30),npst,nsbp(10),nebp(10),zeta(10),nsbpn(10),
```

```
     *       nebpn(10),nlp(10),yp(40),yq(40),cpst,cz1,cz2,cz3,cz4,cz5,
     *       ycon(40,40),nyy,cz6,cz7,fe(10),xm(30,30),zc(30,30),fed(10),
     *       xser(40),d(40,40),rser(40),pf(40),am(10,10),bm(10),cnol,
     *       amv(10,10),tl(40)
      integer i,ii,iii,j,jjj,ir,ic,ipp,s1,s2,s3,ip
      write (*,10)
      write (3,10)
10    format (/,10x,'*** call fecal ***')
      cnol = 0.0
      do 6245 i=1,10
      do 6245 j=1,10
      am(i,j) = 0.0
      amv(i,j) = 0.0
      bm(i) = 0.0
      fe(i) = 0.0
6245  continue
      do 710 iii=1,npst
      s1 = nlp(iii)
      do 720 jjj=1,npst
      s2 = nlp(jjj)
      do 730 ii=1,nl
      am(iii,jjj) = am(iii,jjj)+rser(ii)*d(ii,s1)*d(ii,s2)
730   continue
720   continue
710   continue
      do 760 i=1,npst
      s3 = nlp(i)
      do 770 ipp=1,nl
      bm(i) = bm(i)+pf(ipp)*rser(ipp)*d(ipp,s3)
770   continue
      bm(i) = -bm(i)
760   continue
      write (*,771)
      write (3,771)
```

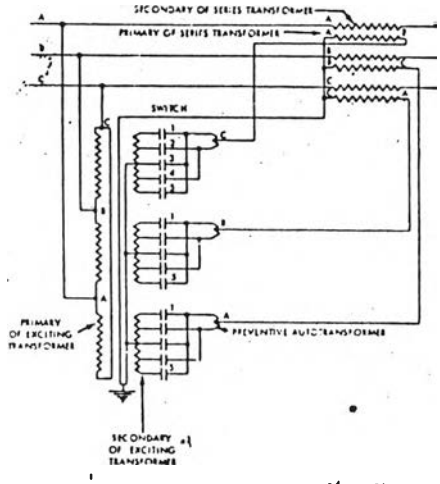
```
771   format (/,5x,'*** THIS IS ALL AM(I,J) ***')
      do 780 i=1,npst
      do 790 j=1,npst
      write (*,7100) i,j,am(i,j)
      write (3,7100) i,j,am(i,j)
7100  format (/,5x,'am(',i2,',',i2,')=',f10.5)
790   continue
780   continue
      write (*,772)
      write (3,772)
772   format (/,5x,'*** THIS IS ALL BM(I) ***')
      do 7110 i=1,npst
      write (*,7112) i,bm(i)
      write (3,7112) i,bm(i)
7112  format (/,5x,'bm(',i2,')=',f10.5)
7110  continue
      do 210 ip=1,npst
      do 215 ir=1,npst
      if (ir.eq.ip) go to 215
      do 220 ic=1,npst
      if (ic.eq.ip) go to 220
      if (am(ip,ip).eq.0) go to 275
      am(ir,ic) = am(ir,ic)-(am(ir,ip)*am(ip,ic)/am(ip,ip))
220   continue
215   continue
      if (am(ip,ip).eq.0) go to 275
      am(ip,ip) = -1/am(ip,ip)
      do 225 i=1,npst
      if (i.eq.ip) go to 225
      am(i,ip) = am(i,ip)*am(ip,ip)
      am(ip,i) = am(ip,i)*am(ip,ip)
225   continue
210   continue
      do 230 ir=1,npst
```

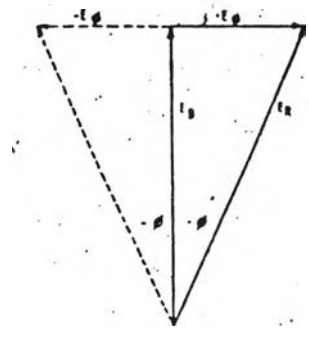
```
      do 230 ic=1,npst
      amv(ir,ic) = -am(ir,ic)
230   continue
      do 3333 i=1,npst
      do 3333 j=1,npst
      write (*,3434) i,j,amv(i,j)
      write (3,3434) i,j,amv(i,j)
3434  format (/,5x,'amv',i2,',',i2,'=',f10.5)
3333  continue
      do 240 i=1,npst
      do 245 j=1,npst
      fe(i) = fe(i)+amv(i,j)*bm(j)
245   continue
240   continue
      do 250 i=1,npst
      ii = nlp(i)
      iii = nsb(ii)
      jjj = neb(ii)
      write (*,255) iii,jjj,fe(i)
      write (3,255) iii,jjj,fe(i)
255   format (/,5x,'** FE(',i2,',',i2,') = ',f9.5,'  radial **')
      zeta(i) = fe(i)
      fed(i) = fe(i)*57.2727
      write (*,256) fed(i)
      write (3,256) fed(i)
256   format(/,18x,'= ',f10.5,'  degree **')
250   continue
      go to 277
275   continue
      write (*,276)
      write (3,276)
276   format (/,5x,'** THIS CASE IS NO SOLUTION OR MULTI SOLUTION **')
      cnol = 1.0
277   continue
```

```
return
end
```

ภาคผนวก ค.

การควบคุมมุมเลื่อนเฟส





รูปที่ ง.1 วงจรทางไฟฟ้าและหลักการควบคุมมุมของตัวเลื่อนเฟส

การควบคุมขนาดและทิศทางของมุม ϕ สามารถควบคุมได้โดยการปรับ turn ratio ด้านทุติยภูมิของหม้อแปลงกระตุ้น การปรับขนาด turn ratio ทำให้แรงดันinput ของหม้อแปลงอนุกรมเปลี่ยนแปลงไปตามขนาดของ turn ratio ส่งผลให้แรงดันมุมต่าง 90 องศา(E_ϕ)ที่เป็นตัวควบคุมมุม ϕ มีการเปลี่ยนแปลงเกิดขึ้นและทำให้มุม ϕ เกิดการเปลี่ยนแปลงตามไปด้วย

การที่ turn ratio ของหม้อแปลงกระตุ้นเป็นตัวควบคุมแรงดันมุมต่าง 90 องศา ขณะที่การเปลี่ยนแปลงของแรงดันมุมต่าง 90 องศาก็คือการเปลี่ยนแปลงขนาดความยาวและทิศทางของเวคเตอร์ E_ϕ ดังรูปที่ ง.1 และเวคเตอร์ E_ϕ นี้เป็นตัวกำหนดขนาดและทิศทางของมุม ϕ ดังนั้นเราจึงสามารถควบคุมมุม ϕ ให้มีค่าเป็นเท่าใดก็ได้ตามแต่ขนาดของ turn ratio ที่เราสามารถออกแบบให้มีขนาดตามต้องการ จะพบว่า turn ratio นี้จะเป็น independent variable ที่เราสามารถปรับค่าได้ตามต้องการและจะทำให้มุม ϕ ซึ่งเป็น controlled variable เปลี่ยนตามไปด้วย

แรงดันอ้างอิงที่ต้องการปรับขนาดมุมนั้นถือว่ามีขนาดคงที่ ขนาดของมุม ϕ ไม่ควรมีค่าเกิน $\pm$20 องศา เพราะถ้ามุม ϕ มีขนาดมากกว่า 20 องศาจะทำให้ขนาดแรงดันลัพธ์เปลี่ยนแปลงไปมากกว่า 4 เปอร์เซนต์ ซึ่งในระบบไฟฟ้ากำลังนั้น voltage regulation ไม่ควรเกิน 5 เปอร์เซนต์ ที่มุมประมาณ 20 องศาหม้อแปลงกระตุ้นที่ใช้จะมี turn ratio ประมาณ 1:0.364 และเมื่อมีการใช้วงจรควบคุมที่เหมาะสมมาช่วยในการควบคุมขนาดของมุมเช่น ใช้ thyristor valve หรือ auto transformer จะสามารถควบคุมมุม ϕ ได้อย่างละเอียดและต่อเนื่องตั้งแต่ค่าประมาณ -20 ถึง +20 องศา

ประวัติผู้เขียน

นาย ปานชนก เตมียเสน เกิดวันที่ 22 กรกฎาคม พ.ศ. 2509 มีภูมิลำเนาอยู่ที่จังหวัดกรุงเทพฯ สำเร็จปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้าจากมหาวิทยาลัยเกษตรศาสตร์ เมื่อปี พ.ศ. 2531 หลังจากนั้นได้เข้าศึกษาต่อในบัณฑิตวิทยาลัย ภาควิชาวิศวกรรมศาสตร์ สาขาพลังงานไฟฟ้า ที่จุฬาลงกรณ์มหาวิทยาลัย