เอกสารอ้างอิง

1. F. William Payne, "The Cogeneration Sourcebook", The Faitmont Press, 1985
2. Philip A. Nobile, "Power System Studies for Cogeneration: What's Really Needed?" IEEE Trans Ind. Appl., Vol.IA-23, Sep./oct. 1987, pp. 777-785
3. เทอดทรงชัย พุทธิศรี "การลดกำลังงานสูญเสียในระบบไฟฟ้ากำลังให้น้อยที่สุดด้วยการควบคุมกำลังรีแอคทีพของระบบให้เหมาะสม" วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, จุฬาลงกรณ์มหาวิทยาลัย, 2533
4. Glenn W. Stagg, Ahmed H. El-Abiad, "Computer Methods in Power System Analysis", McGraw-Hill, 1987
5. G. T. Heyedt, "Computer Abalysis Methods for Power System", Macmillan Publishing Company, 1986
6. J. Arrilage, C. P. Arnold, "Computer Modelling of Electrical Power System", John Willey & Sons, 1984
7. W. Lee, M. Chen, L. B. Williams, "Load Model for Stability Studies", IEEE Trans. Ind. Appl., Vol. IA-23, Jan./Feb. 1987, pp. 159-165
8. G. J. Rogers, D. Shirmohamadi, "Induction Machine Modelling for Electromagnetic Transient Program", IEEE Trans. Energy Conversion, Vol. EC-2, Dec. 1987, pp. 622-628
9. P. M. Anderson, A. A. Found, "Power System Control and Stability", Science Press, 1977.
10. A. E. Fitzgerald, Charles Kingsley, Stephen D. Umans, "Electric Machinery", McGraw-Hill, 1983

11. P. C. Krause, "Analysis of Electric Machinery", McGraw Hill, NewYork, 1986

12. W. W. Hung, "Dynamic Simulation of Gas-turbine Generating Unit", IEEE Proc.-C, Vol. 138, No 4, pp. 342-350, July 1991.

13. J. S. Mayer, O. Wasynczuk, "An Efficient Method of Simulating Stiffly Connected Power Systems with Stator and Network Transients Included", IEEE Trans PWRS., Vol.6, No.3, Aug 1991 pp. 922-929

14. "Computer Representation of Excitation Systems", IEEE Committee Report, IEEE Trans. on PAS, Vol. PAS-87, pp. 1460-1464, June 1968

15. พรชัย ปฏิภาณปรีชาวุฒิ "เสถียรภาพชั่วครู่ของระบบโคเยนเนอเรชั่นที่ต่อเข้ากับระบบจ่ายไฟฟ้าของรัฐ", วิทยานิพนธ์วิศวกรรมศาสตร์มหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย, 2532

16. สมนึก ชยพรกุล "Development of Load Models in Eleetrical Power System", วิทยานิพนธ์วิศวกรรมศาสตร์มหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย, 2532

ภาคผนวก

ภาคผนวก ก.

ค่าอิมพีแดนซ์และค่าคงที่ทางเวลาของเครื่องจักรกลซิงโครนัส

Operational Impedance and Time Constant of Synchronous Machine [11]

Standard Synchronous Machine Reactances

เราได้กล่าวถึงรีแอกแตนซ์ที่ใช้สำหรับโรเตอร์ของเครื่องจักรกลซิงโครนัสที่ประกอบด้วยขดลวดในโรเตอร์จำนวน 4 ชุด (ขดลวดหน่วง 3 ชุด และขดลวดสนาม 1 ชุด) และค่ารีแอกแตนซ์บน q-axis และ d-axis คือ

$$X_q = X_{ls} + X_{mq} \qquad (ก.1)$$

$$X_d = X_{ls} + X_{md} \qquad (ก.2)$$

ซึ่งค่าเหล่านี้จะแสดงคุณสมบัติของเครื่องจักรกลซิงโครนัส ในระหว่างการทำงานที่ภาวะอยู่ตัว (Steady State) ซึ่งค่าตัวแปรบนแกนอ้างอิงโรเตอร์จะมีค่าคงที่

สำหรับค่าทรานเซียนท์รีแอกแตนซ์ (Transient Reactance) บน q-axis และ d-axis จะถูกกำหนดดังนี้

$$X'_q = X_{ls} + \frac{X_{mq} X'_{lkq1}}{X'_{lkq1} + X_{mq}} \qquad (ก.3)$$

$$X'_d = X_{ls} + \frac{X_{md} X'_{lfd}}{X'_{lfd} + X_{md}} \qquad (ก.4)$$

และค่าสับทรานเซียนท์รีแอกแตนซ์ (Subtransient Reactance) บน q-axis และ d-axis จะถูกกำหนดดังนี้

$$X''q = X_{ls} + \frac{X_{mq}X'_{lkq1}X'_{lkq2}}{X_{mq}X'_{lkq1}+X_{mq}X'_{lkq2}+X'_{lkq1}X'_{lkq2}} \qquad (ก.5)$$

$$X''d = X_{ls} + \frac{X_{md}X'_{lfd}X'_{lkd}}{X_{md}X'_{lfd}+X_{md}X'_{lkd}+X'_{lfd}X'_{lkd}} \qquad (ก.6)$$

ในการศึกษาในช่วงหลัง เราจะได้ขดลวดเพียง 1 ชุดในการแสดงคุณสมบัติทางไฟฟ้าบน q-axis ซึ่งถือว่าเพียงพอสำหรับโรเตอร์ชนิดซาเลียนท์โพล (Salient-pole Machine) ซึ่งจากสมการข้างต้นจะเห็นได้ว่าขดลวด kq1 จะสัมพันธ์กับทรานเซียนท์รีแอกแตนซ์และขดลวด kq2 จะสัมพันธ์กับสับทรานเซียนท์รีแอกแตนซ์ ดังนั้นในกรณีของการใช้ขดลวดเพียงขอเดียวเราจะใช้ขดลวด kq2 ในการแสดงคุณสมบัติทางไฟฟ้าบน q-axis

Standard Synchronous Mchine Time Constants

ค่าคงที่ทางเวลาของเครื่องจักรกลซิงโครนัสที่มีขดลวด 4 ชุดบนโรเตอร์จะเป็นดังนี้

$$\tau'q = \frac{1}{\omega b r'_{kq1}}\left(X'_{lkq1} + \frac{X_{mq}X_{ls}}{X_{mq}+X_{ls}}\right) \qquad (ก.7)$$

$$\tau'd = \frac{1}{\omega b r'_{fd}}\left(X'_{lfd} + \frac{X_{md}X_{ls}}{X_{md}+X_{ls}}\right) \qquad (ก.8)$$

$$\tau''q = \frac{1}{\omega b r'_{kq2}}\left(X'_{lkq2} + \frac{X_{mq}X_{ls}X'_{lkq1}}{X_{mq}X_{ls}+X_{mq}X'_{lkq1}+X_{ls}X'_{lkq1}}\right) \qquad (ก.9)$$

$$\tau''d = \frac{1}{\omega b r'_{kd}}\left(X'_{lkd} + \frac{X_{md}X_{ls}X'_{lfd}}{X_{md}X_{ls}+X_{md}X'_{lfd}+X_{ls}X'_{lfd}}\right) \qquad (ก.10)$$

ซึ่งสมการดังที่กล่าวนี้จะใช้ในการหาความสัมพันธ์ของค่าคงที่ทางเวลากับค่ารีแอกแตนซ์ ดังนั้นเราสามารถนำไปใช้ในการแปลงค่าตัวแปรระหว่างตัวแปรทั้ง 2 แบบ

ภาคผนวก ข.

โปรแกรมสำหรับใช้ในการคำนวณ

```
PROGRAM Cogen;
{$N+}
{$M 65520,0,655360}

USES
 Common,Crt;

CONST
 Freq = 60;
 TimEnd = 8.000;
 dT = 0.001;
 MVA = 3.125;
 NoBs = 3;  {No of Bus}
 Dimen = NoBs*2;
 NoG = 1;   {No of Gen}
 NoM = 6;   {No of Motor}
TYPE
 Float = Double;
 Cmpx  = RECORD
  Re   : Float;
  Im   : Float;
 END;
 Mtx2 = Array[1..2,1..2] of Float;
 Mtx2G = Array[1..2,1..2,1..NoG] of Float;
 Mtx2M = Array[1..2,1..2,1..NoM] of Float;
 Vtr2 = Array[1..2] of Float;
 MVtr = Array[1..NoM] of Float;
 MVtr2 = Array[1..2,1..NoM] of Float;
 MByte = Array[1..NoM] of Byte;
 GVtr = Array[1..NoG] of Float;
 GVtr2 = Array[1..2,1..NoG] of Float;
 GByte = Array[1..NoG] of Byte;
 CfG1 = Array[1..2,1..2,1..NoG] of Float;
 CfG2 = Array[1..3,1..3,1..NoG] of Float;
 Bs = Array[1..NoBs] of Cmpx;
 CfM = Array[1..5,1..NoM] of Float;
 BsMtx = Array[1..2,1..2,1..NoBs] of Float;
 BsVtr2 = Array[1..2,1..NoBs] of Float;
 GMtx = Array[1..2,1..2,1..NoG] of Float;
 MMtx = Array[1..2,1..2,1..NoM] of Float;
 Mtx = Array[1..NoBs,1..NoBs] of Cmpx;
 Vtr = Array[1..NoBs] of Cmpx;
VAR
 VF,MF,Gf,CR,EX : Text;
 Tim,StTim : Real;
 Inm,I,Loop,Error : Byte;
 Yt1,Yt2,Ypq1,Ypq2,Zt1,Zt2 : Mtx2;
 Vqd1,Vqd2 : Vtr2;
 V,P : Bs;
 Y,YL : Mtx;
 Qn1,Qn2 : Cmpx;
```

```
 Iqd,KL,Pg,Pl,PQ,yp,Ypp : Vtr;
 VO1,VO2 : Cmpx;
 Ib : Float;

 RsM,XsM,XmM,XrM,RrM,HM,HpM : MVtr;
 FqsM,FdsM,FqrM,FdrM,WrM : MVtr;
 MtrBs : MByte;
 Cf : CfM;
 YsM : MMtx;
 JsM : MVtr2;

 Rs,Rkq,Rfd,Rkd,Xs,Xlkq,Xlfd,Xlkd,Xmq,Xmd,H : GVtr;
 Fds,Fqs,Fkq,Ffd,Fkd : GVtr;
 Wr,Ti,Ang,Exfd : GVtr;
 GnBs : GByte;
 Ys : GMtx;
 Js : GVtr2;
 A : CfG1;
 B : CfG2;

 Vref,V1,V3,Vr,Aex,Bex,W3 : GVtr;
 dWrb,pWrb,Y4,Y6,Y7 : GVtr;

FUNCTION  Magnitude(X : Cmpx) : Float;
BEGIN
 Magnitude := SQRT(SQR(X.Re)+SQR(X.Im));
END;

FUNCTION  Angle(X : Cmpx) : Float;
BEGIN
 IF X.Re <> 0.0 Then
 Begin
  If      (X.Im >=0.0) and (X.Re > 0.0) Then Angle := ARCTAN(X.Im/X.Re)
  Else If (X.Im > 0.0) and (X.Re < 0.0) Then Angle := PI-ARCTAN(ABS(X.Im/X.Re))
  Else If (X.Im <=0.0) and (X.Re < 0.0) Then Angle := PI+ARCTAN(ABS(X.Im/X.Re))
  Else If (X.Im < 0.0) and (X.Re > 0.0) Then Angle := -ARCTAN(ABS(X.Im/X.Re))
 End
 Else
 Begin
  If      X.Im = 0.0 Then Angle := 0.0
  Else If X.Im > 0.0 Then Angle := PI/2.0
  Else Angle := -PI/2.0
 End;
END;

PROCEDURE  XYtoCmpx(X,Y : Float;VAR XY : Cmpx);
BEGIN
 XY.Re := X;
 XY.Im := Y;
END;
```

```
PROCEDURE  Conjugate(X : Cmpx;VAR CONJ : Cmpx);
BEGIN
 CONJ.Re := X.Re;
 CONJ.Im := -X.Im;
END;

PROCEDURE  Sum(X,Y :Cmpx;VAR Sum :Cmpx);
BEGIN
 Sum.Re := X.Re+Y.Re;
 Sum.Im := X.Im+Y.Im;
END;

PROCEDURE  Sub(X,Y :Cmpx;VAR Sub :Cmpx);
BEGIN
 Sub.Re := X.Re-Y.Re;
 Sub.Im := X.Im-Y.Im;
END;

PROCEDURE  Product(X1,Y1 : Cmpx;VAR PROD : Cmpx);
BEGIN
 PROD.Re  := (X1.Re*Y1.Re - X1.Im*Y1.Im);
 PROD.Im  := (X1.Re*Y1.Im + X1.Im*Y1.Re);
END;

PROCEDURE  Devide(A,B :Cmpx;VAR D : Cmpx);
VAR
 MagB : Float;
 C : Cmpx;
BEGIN
 IF (B.Re = 0) AND (B.Im = 0) THEN
 BEGIN
  WRITELN('ERROR : DEVIDED BY ZERO');
  HALT;
  END
 ELSE
 BEGIN
  MagB := SQR(B.Re)+SQR(B.Im);
  C.Re := B.Re; C.Im := -B.Im;
  Product(A,C,C);
  D.Re := C.Re/MagB;
  D.Im := C.Im/MagB;
 END
END;

PROCEDURE  MultiCmpx(X : Float;Y :Cmpx;VAR Z : Cmpx);
BEGIN
 Z.Re := X*Y.Re;
 Z.Im := X*Y.Im;
END;

PROCEDURE InvMtx
```

```
(A : Mtx2;
 VAR B : Mtx2);
VAR
 Det : Float;
BEGIN
 Det := A[1,1]*A[2,2]-A[1,2]*A[2,1];
 B[1,1] := A[2,2]/Det; B[1,2] := -A[1,2]/Det;
 B[2,1] := -A[2,1]/Det; B[2,2] := A[1,1]/Det;
END;

PROCEDURE SumMtx
(A,B : Mtx2;
 VAR C : Mtx2);
BEGIN
 C[1,1] := A[1,1]+B[1,1]; C[1,2] := A[1,2]+B[1,2];
 C[2,1] := A[2,1]+B[2,1]; C[2,2] := A[2,2]+B[2,2];
END;

PROCEDURE MxV
( A : Mtx2;
  B : Vtr2;
  VAR C : Vtr2);
BEGIN
 C[1] := A[1,1]*B[1]+A[1,2]*B[2];
 C[2] := A[2,1]*B[1]+A[2,2]*B[2];
END;

PROCEDURE ParaM
(VAR HpM,RsM,XsM,XmM,XrM,RrM,HM : MVtr);
VAR
 Zb : Float;
 Vb : MVtr;
BEGIN
 Vb[1] := 450;
 HpM[1] := 200;
 RsM[1] := 0.01/(HpM[1]*746/1E6)*MVA;
 XsM[1] := 0.0655/(HpM[1]*746/1E6)*MVA;
 XmM[1] := 3.225/(HpM[1]*746/1E6)*MVA;
 XrM[1] := 0.0655/(HpM[1]*746/1E6)*MVA;
 RrM[1] := 0.0261/(HpM[1]*746/1E6)*MVA;
 HM[1] := 0.922*(HpM[1]*746/1E6)/MVA;
 MtrBs[1] := 2;
 HpM[2] := 150;
 Vb[2] := 450;
 RsM[2] := 0.0051/(HpM[2]*746/1E6)*MVA;
 XsM[2] := 0.00553/(HpM[2]*746/1E6)*MVA;
 XmM[2] := 2.778/(HpM[2]*746/1E6)*MVA;
 XrM[2] := 0.0553/(HpM[2]*746/1E6)*MVA;
 RrM[2] := 0.0165/(HpM[2]*746/1E6)*MVA;
 HM[2] := 1.524*(HpM[2]*746/1E6)/MVA;
 MtrBs[2] := 2;
```

```
 HpM[3] := 40;
 Vb[3] := 450;
 RsM[3] := 0.005/(HpM[3]*746/1E6)*MVA;
 XsM[3] := 0.0587/(HpM[3]*746/1E6)*MVA;
 XmM[3] := 2.952/(HpM[3]*746/1E6)*MVA;
 XrM[3] := 0.0587/(HpM[3]*746/1E6)*MVA;
 RrM[3] := 0.0165/(HpM[3]*746/1E6)*MVA;
 HM[3] := 1.054*(HpM[3]*746/1E6)/MVA;
 MtrBs[3] := 2;
|10 hp|
 HpM[4] := 10;
 Vb[4] := 220;
 RsM[4]  := 0.0453/(10*746/1E6)*MVA;
 XsM[4] := 0.0775/(10*746/1E6)*MVA;
 XmM[4]  := 2.042/(10*746/1E6)*MVA ;
 RrM[4]  := 0.0222/(10*746/1E6)*MVA;
 XrM[4] := 0.0322/(10*746/1E6)*MVA;
 HM[4]  := 0.5*(10*746/1E6)/MVA;
 MtrBs[4] := 2;
|1 MW|
 HpM[5] := 1E6/746;
 Vb[5] := 450;
 RsM[5]  := 3.0672E-3*MVA;
 XsM[5] := 0.02390*MVA;
 XmM[5]  := 1.37918*MVA;
 RrM[5]  := 2.3268E-3*MVA;
 XrM[5] := 0.02390*MVA;
 HM[5] := 0.676/MVA;
 MtrBs[5] := 2;
|2250 Hp|
 Zb := 2300/421;
 Vb[6] := 2300;
 HpM[6] := 2250;
 RsM[6]  := 0.029/Zb/(2250*746/1E6)*MVA;
 XsM[6] := 0.226/Zb/(2250*746/1E6)*MVA;
 XmM[6]  := 13.04/Zb/(2250*746/1E6)*MVA;
 RrM[6]  := 0.022/Zb/(2250*746/1E6)*MVA;
 XrM[6] := 0.226/Zb/(2250*746/1E6)*MVA;
 HM[6]  := 63.87/8/2250/746*SQR(2*Pi*Freq)*(2250*746/1E6)/MVA;
 MtrBs[6] := 2;
 Ib := HpM[Inm]*746/Vb[Inm]*SQRT(2)/SQRT(3);
END;

PROCEDURE IniMotor
(XmM,XrM,RsM,XsM : MVtr;
 VAR YsM : MMtx);
VAR
 I : Byte;
 Xb : Float;
 ZsM,YsM2 : Mtx2;
BEGIN
```

```
 I := Inm;
 Xb := 1/(1/XmM[I]+1/XrM[I]);
 ZsM[1,1] := RsM[I];
 ZsM[2,2] := RsM[I];
 ZsM[1,2] := XsM[I]+Xb;
 ZsM[2,1] := XsM[I]+Xb;
 InvMtx(ZsM,YsM2);
 YsM[1,1,I] := YsM2[1,1]; YsM[1,2,I] := YsM2[1,2];
 YsM[2,1,I] := YsM2[2,1]; YsM[2,2,I] := YsM2[2,2];
END;

PROCEDURE Motor
(XmM,XrM,FdrM,FqrM : MVtr;
 YsM : MMtx;
 VAR JsM : MVtr2);
VAR
 I : Byte;
 Xb : Float;
 EsM,JsM2 : Vtr2;
 ZsM,YsM2 : Mtx2;
BEGIN
 I := Inm;
 Xb := 1/(1/XmM[I]+1/XrM[I]);
 EsM[1] := Xb*FdrM[I]/XrM[I];
 EsM[2] := -Xb*FqrM[I]/XrM[I];
 YsM2[1,1] := YsM[1,1,I]; YsM2[1,2] := YsM[1,2,I];
 YsM2[2,1] := YsM[2,1,I]; YsM2[2,2] := YsM[2,2,I];
 MxV(YsM2,EsM,JsM2);
 JsM[1,I] := JsM2[1]; JsM[2,I] := JsM2[2];
END;

PROCEDURE IniStartM
(V : Bs;
 RsM,XsM,XmM,XrM : MVtr;
 VAR FdsM,FqsM,FdrM,FqrM,WrM : MVtr;
 VAR Cf : CfM);
VAR
 I : Byte;
 Vqs,Vds : Float;
 Iqs,Ids,Iqr,Idr : Float;
 We,Wb,D : Float;
BEGIN
 I := Inm;
{Motor Parameter}
 We  := 2*Pi*Freq;
 Wb  := We;
{Inital Condition for Start Motor}
 If Loop = 1 Then
 Begin
  Vqs := 0;
  Vds := 0;
```

```
 End
 Else
 Begin
  Vqs := V[2].Re;
  Vds := V[2].Im;                          {Vqr=0,Vdr=0}
 End;
 WrM[I] := 0;
 Iqs := 0;
 Ids := 0;
{Finding Initial Flux}
 FdsM[I] := Wb/We*(Vqs-RsM[I]*Iqs);
 FqsM[I] := Wb/We*(RsM[I]*Ids-Vds);
 Iqr := (FqsM[I]-(XsM[I]+XmM[I])*Iqs)/XmM[I];
 Idr := (FdsM[I]-(XsM[I]+XmM[I])*Ids)/XmM[I];
 FqrM[I] := XrM[I]*Iqr+XmM[I]*(Iqs+Iqr);
 FdrM[I] := XrM[I]*Idr+XmM[I]*(Ids+Idr);
{Define Reactant}
 D := (XsM[I]+XmM[I])*(XrM[I]+XmM[I])-SQR(XmM[I]);
 Cf[1,I] := RrM[I]*(XsM[I]+XmM[I])/D;
 Cf[2,I] := RsM[I]*(XrM[I]+XmM[I])/D;
 Cf[3,I] := RsM[I]*XmM[I]/D;
 Cf[4,I] := RrM[I]*XmM[I]/D;
 Cf[5,I] := XmM[I]/D;
END;

PROCEDURE StartM
(V : Bs;
 Cf : CfM;
 RsM,XmM,XsM,HM : MVtr;
 VAR FdsM,FqsM,FdrM,FqrM,WrM : MVtr);
TYPE
 Pnt = (PA,PB);
CONST
 Err = 0.0000001;
{Load Torque Constant}
 a = 0; b = 0; c = 0;
{Voltage at Induction Motor}
 Vqr = 0; Vdr = 0;
VAR
 I : Byte;
 Vqs,Vds : Float;
 Iqs,Ids,Iqr,Idr : Float;
 FqsMB,FdsMB,FqrMB,FdrMB,WrMB : Float;
 FqsMO,FdsMO,FqrMO,FdrMO,WrMO : Float;
 pFqsM,pFdsM,pFqrM,pFdrM,pWrM : Array [Pnt] of Float;
 We,Wb : Float;
 Te,Tm,TeTm : Float;
 Tl : MVtr;
BEGIN
 I := Inn;
 We := 2*Pi*Freq;
```

```
 Wb := We;
 If Loop >= 2 Then
 Begin
  Vqs := V[2].Re;
  Vds := V[2].Im;
 End
 Else
 Begin
  Vqs := 0;
  Vds := 0;
 End;
 Tl[I] := a+b*WrM[I]+c*SQR(WrM[I]);
 {By Modified Euler Method}
 pFqrM[PA] := Wb*(Vqr-Cf[1,I]*FqrM[I]+Cf[4,I]*FqsM[I]-(We-WrM[I])/Wb*FdrM[I]);
 FqrMB := FqrM[I]+pFqrM[PA]*dT;
 pFdrM[PA] := Wb*(Vdr-Cf[1,I]*FdrM[I]+Cf[4,I]*FdsM[I]+(We-WrM[I])/Wb*FqrM[I]);
 FdrMB := FdrM[I]+pFdrM[PA]*dT;
 pWrM[PA] := Wb/HM[I]/2*(Cf[5,I]*(FqsM[I]*FdrM[I]-FqrM[I]*FdsM[I])-TL[I]);
 WrMB := WrM[I]+pWrM[PA]*dT;
 FqsMB := (Vqs-We/Wb/Cf[2,I]*(Vds+Cf[3,I]*FdrMB)+Cf[3,I]*FqrMB)/(Cf[2,I]+SQR(We/Wb)/Cf[2,I]);
 FdsMB := (Vds+We/Wb*FqsMB+Cf[3,I]*FdrMB)/Cf[2,I];
 Repeat                              {Converted Value at Any Time}
  FqrMO := FqrMB; FdrMO := FdrMB;
  FqsMO := FqsMB; FdsMO := FdsMB;
  WrMO  := WrMB;
  pFqrM[PB] := Wb*(Vqr-Cf[1,I]*FqrMB+Cf[4,I]*FqsMB-(We-WrMB)/Wb*FdrMB);
  FqrMB := FqrM[I]+dT/2*(pFqrM[PA]+pFqrM[PB]);
  pFdrM[PB] := Wb*(Vdr-Cf[1,I]*FdrMB+Cf[4,I]*FdsMB+(We-WrMB)/Wb*FqrMB);
  FdrMB := FdrM[I]+dT/2*(pFdrM[PA]+pFdrM[PB]);
  FqsMB := (Vqs-We/Wb/Cf[2,I]*(Vds+Cf[3,I]*FdrMB)+Cf[3,I]*FqrMB)/(Cf[2,I]+SQR(We/Wb)/Cf[2,I]);
  FdsMB := (Vds+We/Wb*FqsMB+Cf[3,I]*FdrMB)/Cf[2,I];
  pWrM[PB] := Wb/HM[I]/2*(Cf[5,I]*(FqsMB*FdrMB-FqrMB*FdsMB)-TL[I]);
  WrMB  := WrM[I]+dT/2*(pWrM[PA]+pWrM[PB]);
 Until (ABS(FqrMB-FqrMO) < Err) and (ABS(FdrMB-FdrMO) < Err) and
       (ABS(FqsMB-FqsMO) < Err) and (ABS(FdsMB-FdsMO) < Err) and
       (ABS(WrMB-WrMO)   < Err);
 FdsM[I] := FdsMB; FqsM[I] := FqsMB;
 FdrM[I] := FdrMB; FqrM[I] := FqrMB;
 WrM[I]  := WrMB;
{ Iqs := (Vqs-We/Wb*FdsM[I])/RsM[I];
 Ids := We/Wb*FqsM[I]/RsM[I];
 Iqr := (FqsM[I]-(XsM[I]+XmM[I])*Iqs)/XmM[I];
 Idr := (FdsM[I]-(XsM[I]+XmM[I])*Ids)/XmM[I];}
 Te  := Cf[5,I]*(FqsM[I]*FdrM[I]-FqrM[I]*FdsM[I]);
{ TeTm := Cf[5,I]*(FqsM[I]*FdrM[I]-FqrM[I]*FdsM[I])-a-b*WrM[I]-c*SQR(WrM[I]);}
{ Writeln(MF,WrM[I]/Wb,Te);}
END;

PROCEDURE IniAVR
(VAR V1,V3,Vref,Vr,W3,Aex,Bex : GVtr;
 Exfd : GVtr;
```

```
 V : Bs);
CONST
{ Tr = 0; Ka = 400; Ta = 0.01; Vmx = 8.4; Vmn = 0;
 Kf = 0.01; Tf1 = 0.15; Tf2 = 0.06; Ke = 1; Te = 0.1;
 Semx = 0.860; Senmx = 0.5; |          {SEnmx = SE0.75mx|
{ Tr = 0.04; Ka = 1375; Ta = 0.0025; Vmx = 16.44; Vmn = 0;
 Kf = 0.0124; Tf1 = 1.716; Tf2 = 0.154; Ke = 1; Te = 0.96;
 Semx = 1.82; Senmx = 1.71; |          {SEnmx = SE0.75mx|
 Tr = 0.02; Ka = 1660; Ta = 0.0017; Vmx = 20; Vmn = 0;
 Kf = 0.018; Tf1 = 1.632; Tf2 = 0.232; Ke = 1; Te = 1.02;
 Semx = 2.21; Senmx = 2.21;           {SEnmx = SE0.75mx|
VAR
 I : Byte;
 Ve,Se,Vt,Emx : Float;
BEGIN
 I := 1;
 Vt := Magnitude(V[I]);
 Emx := Vmx/(Ke+Semx);
 Aex[I] := (Senmx*Senmx*Senmx*Senmx)/(Semx*Semx*Semx);
 Bex[I] := (4/Emx)*ln(Semx/Senmx);
 Se := Aex[I]*EXP(Bex[I]*Exfd[I]);
 Vr[I] := (Ke+Se)*Exfd[I];
 Ve := Vr[I]/Ka;
 V3[I] := Kf*Ka*Ve/Ta-Kf*Vr[I]/Ta;
 Vref[I] := Ve+Vt+V3[I];
 V1[I] := Vt;
 W3[I] := 0;
END;

PROCEDURE AVR
(Aex,Bex,Vref : GVtr;
 V : Bs;
 VAR Exfd,W3,V1,Vr,V3 : GVtr);
CONST
{ Tr = 0; Ka = 400; Ta = 0.01; Vmx = 8.4; Vmn = 0;
 Kf = 0.01; Tf1 = 0.15; Tf2 = 0.06; Ke = 1; Te = 0.1;|
{ Tr = 0.04; Ka = 1375; Ta = 0.0025; Vmx = 16.44; Vmn = 0;
 Kf = 0.0124; Tf1 = 1.716; Tf2 = 0.154; Ke = 1; Te = 0.96;|
 Tr = 0.02; Ka = 1660; Ta = 0.0017; Vmx = 20; Vmn = 0;
 Kf = 0.018; Tf1 = 1.632; Tf2 = 0.232; Ke = 1; Te = 1.02;
 Err = 0.0000000000001;
VAR
 I : Byte;
 Vt,Se,V1O,pV1A,pV1B,V1B,Ve,VeB,VrO,pVrA,pVrB,VrB,ExfdO,pExfdA,pExfdB,ExfdB : Float;
 T12,T12a,pW3A,pW3B,W3B,W3O : Float;
BEGIN
 I := 1;
 Vt := Magnitude(V[1]);
 If Tr = 0 Then V1[I] := Vt
 Else
 Begin
```

```
 pV1A := (Vt-V1[I])/Tr;
 V1B := V1[I]+pV1A*dT;
 Repeat
  V10 := V1B;
  pV1B := (Vt-V1B)/Tr;
  V1B := V1[I]+(pV1A+pV1B)*dT/2;
 Until ABS(V1B-V10) < Err;
 V1[I] := V1B;
End;
Ve := Vref[I]-V1[I]-V3[I];
pVrA := (Ka*Ve-Vr[I])/Ta;
VrB := Vr[I]+(pVrA)*dT;
Repeat
 VrO := VrB;
 pVrB := (Ka*Ve-VrB)/Ta;
 VrB := Vr[I]+(pVrA+pVrB)*dT/2;
Until ABS(VrB-VrO) < Err;
If VrB >= Vmx Then Vr[I] := Vmx
Else If VrB > Vmn Then Vr[I] := VrB
Else If VrB <= Vmn Then Vr[I] := Vmn
Else Writeln('ERROR!');
Se := Aex[I]*Exp(Bex[I]*Exfd[I]);
pExfdA := (Vr[I]-(Ke+Se)*Exfd[I])/Te;
ExfdB := Exfd[I]+pExfdA*dT;
Repeat
 Exfd0 := ExfdB;
 Se := Aex[I]*Exp(Bex[I]*ExfdB);
 pExfdB := (Vr[I]-(Ke+Se)*ExfdB)/Te;
 ExfdB := Exfd[I]+(pExfdA+pExfdB)*dT/2;
Until ABS(ExfdB-Exfd0) < Err;
Exfd[I] := ExfdB;
T12 := 1/(Tf1*Tf2);
T12a := 1/(Tf1*Tf2*Ta);
pW3A := Kf*Ka*T12a*Ve-Kf*T12a*Vr[I]-(Tf1+Tf2)*T12*W3[I]-T12*V3[I];
W3B := W3[I]+pW3A*dT;
Repeat
 W30 := W3B;
 pW3B := Kf*Ka*T12a*Ve-Kf*T12a*Vr[I]-(Tf1+Tf2)*T12*W3[I]-T12*V3[I];
 W3B := W3[I]+(pW3A+pW3B)*dT/2;
Until (ABS(W3B-W30) < Err);
V3[I] := V3[I]+(W3[I]+W3B)*dT/2;
W3[I] := W3B;
END;

PROCEDURE IniGov
(Ti : GVtr;
VAR Y4,Y6,Y7 : GVtr);
CONST
 Wf = 0.23; C2 = 0.251; C1 = 1.3523; Cg = 0.5;
VAR Y11 : Float; I : Byte;
BEGIN
```

```
 I := 1;
 Y4[I] := Ti[I]/C1+C2-Wf; {dWrb = 0}
 Y6[I] := Ti[I]/C1+C2;
 Y7[I] := Y6[I];
END;

PROCEDURE Governer
(dWrb,pWrb : GVtr;
 VAR Y4,Y6,Y7,Ti : GVtr);
CONST
 Kc = 22.5; Tc = 0.55; Tfv =0.01; Tft = 0.05;
 Wf = 0.23; C2 = 0.251; C1 = 1.3523; Cg = 0.5;
 Err = 0.0000000001;
VAR
 pY4,pY6,pY7,pY4B,pY6B,pY7B,Y4B,Y6B,Y7B,Y4O,Y6O,Y7O : Float;
 I : Byte;
BEGIN
 I := 1;
 pY4 := Kc/Tc*dWrb[I]-Kc*pWrb[I];
 Y4B := Y4[I]+pY4*dT;
 pY6 := (Y4[I]+Wf-Y6[I])/Tfv;
 Y6B := Y6[I]+pY6*dT;
 pY7 := (Y6[I]-Y7[I])/Tft;
 Y7B := Y7[I]+pY7*dT;
 Repeat
  Y6O := Y6B; Y7O := Y7B;
  pY6B := (Y4B+Wf-Y6B)/Tfv;
  Y6B := Y6[I]+(pY6+pY6B)*dT/2;
  pY7B := (Y6B-Y7B)/Tft;
  Y7B := Y7[I]+(pY7+pY7B)*dT/2;
 Until (ABS(Y6B-Y6O) < Err) AND (ABS(Y7B-Y7O) < Err);
 Y4[I] := Y4B; Y6[I] := Y6B; Y7[I] := Y7B;
 Ti[I] := C1*(Y7[I]-C2)+Cg*dWrb[I];
END;

PROCEDURE ParaG
(VAR Rs,Rkq,Rfd,Rkd,Xs,Xlkq,Xlfd,Xlkd,Xmq,Xmd,H : GVtr);
VAR
 I : Byte;
 MVAG : Float;
BEGIN
 I := 1;
 MVAG := 3.125;
 Rs[I] := 0.00515/MVAG*MVA;
 Rkq[I] := 0.0613/MVAG*MVA;
 Rfd[I] := 0.00111/MVAG*MVA;
 Rkd[I] := 0.02397/MVAG*MVA;
 Xs[I] := 0.8/MVAG*MVA;
 Xlkq[I] := 0.3298/MVAG*MVA;
 Xlfd[I] := 0.13683/MVAG*MVA;
 Xlkd[I] := 0.33383/MVAG*MVA;
```

```
 Xmq[I] := 1.0/MVAG*MVA;
 Xmd[I] := 1.768/MVAG*MVA;
 H[I] := 2.137*MVAG/MVA;
 GnBs[I] := 1;
END;

PROCEDURE TransferG
(VAR Rs,Rkq,Rfd,Rkd,Xs,X1kq,X1fd,X1kd,Xmq,Xmd,H : GVtr);
VAR
 I : Byte;
 f,Wb,MVAG,kVG,Xd,Xq,X1d,X2d,X2q,T1d,T2d,T2q,Ta,Xa,Ra,Rf,HG : Float;
BEGIN
{ f := 50;
 Wb := 2*Pi*f;
 MVAG := 16;
 kVG := 6.6;
 Xd := 2.0;
 Xq := 1.6;
 X1d := 0.263;
 X2d := 0.173;
 X2q := 0.20;
 T1d := 1.105;
 T2d := 0.035;
 T2q := 0.035;
 Ta := 0.20;
 Xa := 0.20;
 Ra := 0.003;
 Rf := 0.006;
 HG := 2.6;|
 f := 60;
 Wb := 2*Pi*f;
 MVAG := 18.875;
 kVG := 11;
 Xd := 1.78;
 Xq := 1.78;
 X1d := 0.272;
 X2d := 0.194;
 X2q := 0.194;
 T1d := 1.12;
 T2d := 0.045;
 T2q := 0.045;
 Ta := 0.192;
 Xa := 0.12;
 Ra := 0.0023;
 Rf := 0.00064;
 HG := 2.5;
 I := 1;
 Rs[I] := Ra;
 Xs[I] := Xa;
 Xmq[I] := Xq-Xs[I];
 Xmd[I] := Xd-Xs[I];
```

```
 Xlkq[I] := (X2q-Xs[I])*Xmq[I]/(Xmq[I]+Xs[I]-X2q);
 Xlfd[I] := (X1d-Xs[I])*Xmd[I]/(Xmd[I]+Xs[I]-X1d);
 Xlkd[I] := (X2d-Xs[I])*Xmd[I]*Xlfd[I]/(Xmd[I]*Xlfd[I]+(Xs[I]-X2d)*Xmd[I]+(Xs[I]-X2d)*Xlfd[I]);
 Rkq[I] := (Xlkq[I]+(Xmq[I]*Xs[I])/(Xmq[I]+Xs[I]))/Wb/T2q;
 Rfd[I] := (Xlfd[I]+(Xmd[I]*Xs[I])/(Xmd[I]+Xs[I]))/Wb/T1d;
 Rkd[I] := (Xlkd[I]+(Xmd[I]*Xs[I]*Xlfd[I])/(Xmd[I]*Xs[I]+Xmd[I]*Xlfd[I]+Xs[I]*Xlfd[I]))/Wb/T2d;
 Rs[I] := Rs[I]/MVAG*MVA;
 Rkq[I] := Rkq[I]/MVAG*MVA;
 Rfd[I] := Rfd[I]/MVAG*MVA;
 Rkd[I] := Rkd[I]/MVAG*MVA;
 Xs[I] := Xs[I]/MVAG*MVA;
 Xlkq[I] := Xlkq[I]/MVAG*MVA;
 Xlfd[I] := Xlfd[I]/MVAG*MVA;
 Xlkd[I] := Xlkd[I]/MVAG*MVA;
 Xmq[I] := Xmq[I]/MVAG*MVA;
 Xmd[I] := Xmd[I]/MVAG*MVA;
 H[I] := HG*MVAG/MVA;
 GnBs[I] := 1;
END;

PROCEDURE Generator
(Xmq,Xmd,Xlfd,Xlkd,Xlkq,Rs,Xs,Ffd,Fkd,Fkq,Ang : GVtr;
 VAR Ys : GMtx;
 VAR Js : GVtr2);
VAR
 Xbq,Xbd,Vqs,Vds : Float;
 Vs : GVtr;
 Es : Vtr2;
 Zs,Ys2 : Mtx2;
 Js2 : Vtr2;
 I : Byte;
BEGIN
 I := 1;
 Xbq := 1/(1/Xmq[I]+1/Xlkq[I]);
 Xbd := 1/(1/Xmd[I]+1/Xlfd[I]+1/Xlkd[I]);
 Es[1] := Xbd*((Ffd[I]/Xlfd[I]+Fkd[I]/Xlkd[I])*COS(Ang[I])-Xbq*(Fkq[I]/Xlkq[I])*SIN(Ang[I]));
 Es[2] := -Xbd*(Ffd[I]/Xlfd[I]+Fkd[I]/Xlkd[I])*SIN(Ang[I])-Xbq*(Fkq[I]/Xlkq[I])*COS(Ang[I]);
 Zs[1,1] := Rs[I]+(Xbd-Xbq)*COS(Ang[I])*SIN(Ang[I]);
 Zs[1,2] := Xs[I]+Xbd*SQR(COS(Ang[I]))+Xbq*SQR(SIN(Ang[I]));
 Zs[2,1] := -Xs[I]-Xbd*SQR(SIN(Ang[I]))-Xbq*SQR(COS(Ang[I]));
 Zs[2,2] := Rs[I]-(Xbd-Xbq)*COS(Ang[I])*SIN(Ang[I]);
 InvMtx(Zs,Ys2);
 MxV(Ys2,Es,Js2);
 Ys[1,1,I] := Ys2[1,1]; Ys[1,2,I] := Ys2[1,2];
 Ys[2,1,I] := Ys2[2,1]; Ys[2,2,I] := Ys2[2,2];
 Js[1,I] := Js2[1]; Js[2,I] := Js2[2];
END;

PROCEDURE  IniGn
(Xs,Xmq,Xmd,Xlkq,Xlfd,Xlkd : GVtr;
 VAR Fds,Fqs,Fkq,Fkd,Ffd,Ti,Wr,Ang,Exfd : GVtr;
```

```
 VAR A : CfG1;
 VAR B : CfG2);
VAR
 Xq,Xd,Xkq,Xfd,Xkd,Dq,Dd,Wb,Vqs,Vds,Te,Ifd,Iqs,Ids,AngIV,Vs : Float;
 Ias,Ea : Cmpx;
 I : Byte;
BEGIN
 I := 1;
 {Initial Freq}
 Wb := 2*Pi*Freq;
 Wr[I] := Wb;
 {Parameter}
 Xq := Xs[I]+Xmq[I];
 Xd := Xs[I]+Xmd[I];
 Xkq := X1kq[I]+Xmq[I];
 Xfd := X1fd[I]+Xmd[I];
 Xkd := X1kd[I]+Xmd[I];
 Dq := SQR(Xmq[I])-Xkq*Xq;
 Dd := SQR(Xmd[I])*(Xd-2*Xmd[I]+Xfd+Xkd)-Xd*Xfd*Xkd;
 A[1,1,I] := (-Rs[I])*Xkq/Dq;
 A[1,2,I] := (-Rs[I])*(-Xmq[I])/Dq;
 A[2,1,I] := Rkq[I]*Xmq[I]/Dq;
 A[2,2,I] := Rkq[I]*(-Xq)/Dq;
 B[1,1,I] := (-Rs[I])*(Xfd*Xkd-SQR(Xmd[I]))/Dd;
 B[1,2,I] := (-Rs[I])*(-Xmd[I]*Xkd+SQR(Xmd[I]))/Dd;
 B[1,3,I] := (-Rs[I])*(-Xmd[I]*Xfd+SQR(Xmd[I]))/Dd;
 B[2,1,I] := Xmd[I]*(Xmd[I]*Xkd-SQR(Xmd[I]))/Dd;
 B[2,2,I] := Xmd[I]*(-Xd*Xkd+SQR(Xmd[I]))/Dd;
 B[2,3,I] := Xmd[I]*(Xd*Xmd[I]-SQR(Xmd[I]))/Dd;
 B[3,1,I] := Rkd[I]*(Xmd[I]*Xfd-SQR(Xmd[I]))/Dd;
 B[3,2,I] := Rkd[I]*(Xd*Xmd[I]-SQR(Xmd[I]))/Dd;
 B[3,3,I] := Rkd[I]*(-Xd*Xfd+SQR(Xmd[I]))/Dd;
 {Find Initial Condition}
 XYtoCmpx(Pg[1].Re,-Pg[1].Im,Qn1);
 Conjugate(V[1],Qn2);
 Devide(Qn1,Qn2,Ias);
 XYtoCmpx(Rs[1],Xq,Qn1);
 Product(Qn1,Ias,Qn1);
 Sum(V[1],Qn1,Ea);
{ V[G2Bs[I]] := Ea;}
 Ang[I] := Angle(Ea)-Angle(V[1]);
 AngIV := Angle(Ias)-Angle(V[1]);
 Ids := -Magnitude(Ias)*SIN(AngIV-Ang[I]);
 Iqs := Magnitude(Ias)*COS(AngIV-Ang[I]);
 Exfd[I] := Magnitude(Ea)+(Xd-Xq)*Ids;
 Ifd := Exfd[I]/Xmd[I];
 Vqs := -Rs[I]*Iqs-Wr[I]/Wb*Xd*Ids+Wr[I]/Wb*Xmd[I]*Ifd;
 Vds := -Rs[I]*Ids+Wr[I]/Wb*Xq*Iqs;
 Writeln('Vqs = ',Vqs,'Vds = ',Vds);;
 {Find Initial Flux}
 Fqs[I] := -Xq*Iqs;          {No Damp So Ikd=Ikq1=Ikq2=0}
```

```
 Fkq[I] := -Xmq[I]*Iqs;
 Fds[I] := Xmd[I]*Ifd-Xd*Ids;
 Ffd[I] := Xfd*Ifd-Xmd[I]*Ids;
 Fkd[I] := Xmd[I]*Ifd-Xmd[I]*Ids;
 {Find Initial Torque}
{Iqs := -(A[1,1,I]*Fqs[I]+A[1,2,I]*Fkq1[I]+A[1,3,I]*Fkq2[I])/RsG[I];
 Ids := -(B[1,1,I]*Fds[I]+B[1,2,I]*Ffd[I]+B[1,3,I]*Fkd[I])/RsG[I];}
{Te := Fds[I]*Iqs-Fqs[I]*Ids;}
 Te := (B[1,1,I]/Rs[I]-A[1,1,I]/Rs[I])*Fqs[I]*Fds[I]+Fds[I]*(-A[1,2,I]/Rs[I]*Fkq[I])
       -Fqs[I]*(-B[1,2,I]/Rs[I]*Ffd[I]-B[1,3,I]/Rs[I]*Fkd[I]);
 Ti[I] := Te;
 Writeln(EX,Exfd[I],Ti[I]);
 Writeln(GF,Ang[I],' ',Wr[I]/Wb);
 {Voltage Magnitude}
 Vs := SQRT(SQR(Vqs)+SQR(Vds));
 Vqs := Vs*COS(Ang[I]);              {Check Vqs,Vds}
 Vds := Vs*SIN(Ang[I]);
 Writeln('Vqs = ',Vqs,'Vds = ',Vds);
 {Find Initial Condition}
 Vqs := V[1].Re*COS(Ang[I])-V[1].Im*SIN(Ang[I]);
 Vds := V[1].Im*COS(Ang[I])+V[1].Re*SIN(Ang[I]);
 Writeln('Vqs = ',Vqs,'Vds = ',Vds);
 Vs := SQRT(SQR(Vqs)+SQR(Vds));
 Vqs := Vs*COS(Ang[I]);              {Check Vqs,Vds}
 Vds := Vs*SIN(Ang[I]);
 Writeln('Vqs = ',Vqs,'Vds = ',Vds);Readln;
END;

PROCEDURE  GnVolt
(V : Bs;
 A : CfG1;
 B : CfG2;
 Rs,Exfd,H,Xmd,Rfd : GVtr;
 VAR Fds,Fqs,Fkq,Fkd,Ffd,Ti,Ang,Wr,dWrb,pWrb : GVtr);
TYPE
 Pnt = (PA,PB);
CONST
 Vkq = 0; Vkd = 0; Err = 0.0000000100;
VAR
 Vqs,Vds,Wb,Te,Vs : Float;
 ExfdB,VqsB,VdsB,Exfd0,Vqs0,Vds0,IfdB,IqsB,IdsB : Float;
 FkqB,FkdB,FfdB,FqsB,FdsB,WrB,AngB : Float;
 Fkq0,Fkd0,Ffd0,Fqs0,Fds0,Wr0,Ang0 : Float;
 pFqs,pFds,pFkq,pFkd,pFfd,pWr,pAng : Array [Pnt] of Float;
 I  : Byte;
BEGIN
 Wb := 2*Pi*Freq;
 I := 1;
 Vqs := V[1].Re*COS(Ang[I])-V[1].Im*SIN(Ang[I]);
 Vds := V[1].Im*COS(Ang[I])+V[1].Re*SIN(Ang[I]);
 Vs := SQRT(SQR(Vqs)+SQR(Vds));
```

```
 Vqs := Vs*COS(Ang[I]);
 Vds := Vs*SIN(Ang[I]);
 pFkq[PA] := Wb*(Vkq-A[2,1,I]*Fqs[I]-A[2,2,I]*Fkq[I]);
 FkqB := Fkq[I]+pFkq[PA]*dT;
 pFfd[PA] := Wb*Rfd[I]*(Exfd[I]-B[2,1,I]*Fds[I]-B[2,2,I]*Ffd[I]-B[2,3,I]*Fkd[I])/Xmd[I];
 FfdB := Ffd[I]+pFfd[PA]*dT;
 pFkd[PA] := Wb*(Vkd-B[3,1,I]*Fds[I]-B[3,2,I]*Ffd[I]-B[3,3,I]*Fkd[I]);
 FkdB := Fkd[I]+pFkd[PA]*dT;
 Te   := (B[1,1,I]/Rs[I]-A[1,1,I]/Rs[I])*Fqs[I]*Fds[I]+Fds[I]*(-A[1,2,I]/Rs[I]*Fkq[I])
        -Fqs[I]*(-B[1,2,I]/Rs[I]*Ffd[I]-B[1,3,I]/Rs[I]*Fkd[I]);
 pWr[PA] := Wb/2/H[I]*(Ti[I]-Te);
 WrB := Wr[I]+pWr[PA]*dT;
 pAng[PA] := Wr[I]-Wb;
 AngB := Ang[I]+pAng[PA]*dT;
{ VqsB := V[1].Re*COS(AngB)-V[1].Im*SIN(AngB);
 VdsB := V[1].Im*COS(AngB)+V[1].Re*SIN(AngB);}
 VqsB  := Vs*COS(AngB);
 VdsB  := Vs*SIN(AngB);
 FqsB  := (VqsB-WrB/Wb*(VdsB-B[1,2,I]*FfdB-B[1,3,I]*FkdB)/B[1,1,I]-A[1,2,I]*FkqB)
         /(A[1,1,I]+SQR(WrB/Wb)/B[1,1,I]);
 FdsB  := (VdsB+WrB/Wb*(VqsB-A[1,2,I]*FkqB)/A[1,1,I]-B[1,2,I]*FfdB-B[1,3,I]*FkdB)
          /(B[1,1,I]+SQR(WrB/Wb)/A[1,1,I]);
 Repeat
  FkqO := FkqB; FfdO := FfdB; FkdO := FkdB;
  FqsO := FqsB; FdsO := FdsB; WrO  := WrB;
  VqsO := VqsB; VdsO := VdsB; AngO := AngB;
  pFkq[PB] := Wb*(Vkq-A[2,1,I]*FqsB-A[2,2,I]*FkqB);
  FkqB := Fkq[I]+(pFkq[PA]+pFkq[PB])*dT/2;
  pFfd[PB] := Wb*Rfd[I]*(Exfd[I]-B[2,1,I]*FdsB-B[2,2,I]*FfdB-B[2,3,I]*FkdB)/Xmd[I];
  FfdB  := Ffd[I]+(pFfd[PA]+pFfd[PB])*dT/2;
  pFkd[PB]  := Wb*(Vkd-B[3,1,I]*FdsB-B[3,2,I]*FfdB-B[3,3,I]*FkdB);
  FkdB  := Fkd[I]+(pFkd[PA]+pFkd[PB])*dT/2;
  FqsB  := (VqsB-WrB/Wb*(VdsB-B[1,2,I]*FfdB-B[1,3,I]*FkdB)/B[1,1,I]-A[1,2,I]*FkqB)
          /(A[1,1,I]+SQR(WrB/Wb)/B[1,1,I]);
  FdsB  := (VdsB+WrB/Wb*(VqsB-A[1,2,I]*FkqB)/A[1,1,I]-B[1,2,I]*FfdB-B[1,3,I]*FkdB)
          /(B[1,1,I]+SQR(WrB/Wb)/A[1,1,I]);
  Te    := (B[1,1,I]/Rs[I]-A[1,1,I]/Rs[I])*FqsB*FdsB+FdsB*(-A[1,2,I]/Rs[I]*FkqB)
           -FqsB*(-B[1,2,I]/Rs[I]*FfdB-B[1,3,I]/Rs[I]*FkdB);
  pWr[PB] := Wb/2/H[I]*(Ti[I]-Te);
  WrB  := Wr[I]+(pWr[PA]+pWr[PB])*dT/2;
  pAng[PB] := WrB-Wb;
  AngB := Ang[I]+(pAng[PA]+pAng[PB])*dT/2;
{  VqsB := V[1].Re*COS(AngB)-V[1].Im*SIN(AngB);
  VdsB := V[1].Im*COS(AngB)+V[1].Re*SIN(AngB);}
  VqsB := Vs*COS(AngB);
  VdsB := Vs*SIN(AngB);
 Until (ABS(FkqB-FkqO) < Err) and (ABS(FfdB-FfdO) < Err) and (ABS(FkdB-FkdO) < Err) and
       (ABS(FqsB-FqsO) < Err) and (ABS(FdsB-FdsO) < Err) and (ABS(WrB-WrO)   < Err) and
       (ABS(VqsB-VqsO) < Err) and (ABS(VdsB-VdsO) < Err) and (ABS(AngB-AngO) < Err)  ;
{ IqsB := -(A[1,1,I]*FqsB+A[1,2,I]*FkqB)/Rs[I];
 IdsB := -(B[1,1,I]*FdsB+B[1,2,I]*FfdB+B[1,3,I]*FkdB)/Rs[I];
```

```
 IfdB :=  (B[2,1,I]*PdsB+B[2,2,I]*PfdB+B[2,3,I]*PkdB)/Xmd[I];|
 dWrb[I] := (1-WrB/Wb);
 pWrb[I] := (pWr[PA]+pWr[PB])/2/Wb;
 Writeln(EX,Exfd[I],Ti[I]);
 Writeln(GF,AngB,' ',WrB/Wb);
 Fkq[I] := FkqB; Ffd[I] := FfdB; Fkd[I] := FkdB;
 Fqs[I] := FqsB; Fds[I] := FdsB; Wr[I] := WrB; Ang[I] := AngB;
END;

PROCEDURE CalIqd
(Ys : GMtx; Js : GVtr2;
 YsM : MMtx; JsM : MVtr2;
 VAR Iqd : Vtr);
VAR
 I : Byte;
 Yqd : BsMtx;
 Jqd : BsVtr2;
BEGIN
 I := 1;
 Jqd[1,I] := Js[1,1];
 Jqd[2,I] := Js[2,1];
 Yqd[1,1,I] := Ys[1,1,1]; Yqd[1,2,I] := Ys[1,2,1];
 Yqd[2,1,I] := Ys[2,1,1]; Yqd[2,2,I] := Ys[2,2,1];
 Iqd[I].Re := Jqd[1,I]-Yqd[1,1,I]*V[I].Re-Yqd[1,2,I]*V[I].Im;
 Iqd[I].Im := -(Jqd[2,I]-Yqd[2,1,I]*V[I].Re-Yqd[2,2,I]*V[I].Im);
 I := 2;
 Jqd[1,I] := JsM[1,Imm];
 Jqd[2,I] := JsM[2,Imm];
 Yqd[1,1,I] := YsM[1,1,Imm]; Yqd[1,2,2] := YsM[1,2,Imm];
 Yqd[2,1,I] := YsM[2,1,Imm]; Yqd[2,2,2] := YsM[2,2,Imm];
 Iqd[I].Re := Jqd[1,I]-Yqd[1,1,I]*V[I].Re-Yqd[1,2,I]*V[I].Im;
 Iqd[I].Im := -(Jqd[2,I]-Yqd[2,1,I]*V[I].Re-Yqd[2,2,I]*V[I].Im);
{ Writeln(CR,SQRT(SQR(Iqd[I].Re)+SQR(Iqd[I].Im))*MVA*1E6*SQRT(2)/450/SQRT(3)/Ib);|
END;

PROCEDURE Net(VAR Y,YL : Mtx);
VAR
 Rn1,Yn1,Rn2,Yn2,Uni,Zr : Cmpx;
BEGIN
 XYtoCmpx(0,0,Zr);
 XYtoCmpx(0.005,0.01,Rn1);
 XYtoCmpx(1,0,Uni);
 Devide(Uni,Rn1,Yn1);
 MultiCmpx(-1,Yn1,Rn1);
 XYtoCmpx(0.05,0.05,Rn2);
 Devide(Uni,Rn2,Yn2);
 MultiCmpx(-1,Yn2,Rn2);
 Sum(Yn1,Yn2,Y[1,1]); Y[1,2] := Rn1;
 Y[2,1] := Rn1; Y[2,2] := Yn1;
 Y[1,3] := Rn2; Y[2,3] := Zr;
 Y[3,1] := Rn2; Y[3,2] := Zr;Y[3,3] := Yn2;
```

```
 Devide(Y[1,2],Y[1,1],YL[1,2]);
 Devide(Y[1,3],Y[1,1],YL[1,3]);
 Devide(Y[2,1],Y[2,2],YL[2,1]);
END;

PROCEDURE Flow
(Y,YL : Mtx;
 VAR Pg,Pl,PQ,KL : Vtr;
 VAR V : BS);
VAR
 Zr : Cmpx;
BEGIN
 XYtoCmpx(0,0,Zr);
 Pg[1].Re := 1/MVA;
 Pg[1].Im := 1/MVA;
 Pl[2].Re := 1/MVA;
 Pl[2].Im := 1/MVA;
 PQ[1] := Pg[1];                    {PQ = Pg-Pl}
 Sub(Zr,Pl[2],PQ[2]);
 Conjugate(PQ[1],Qn1);
 Devide(Qn1,Y[1,1],KL[1]);
 Conjugate(PQ[2],Qn1);
 Devide(Qn1,Y[2,2],KL[2]);
 Repeat
  V01 := V[1]; V02 := V[2];
  Product(YL[1,2],V[2],Qn1);
  MultiCmpx(-1,Qn1,Qn1);
  Product(YL[1,3],V[3],Qn2);
  MultiCmpx(-1,Qn2,Qn2);
  Sum(Qn2,Qn1,Qn1);
  Conjugate(V[1],Qn2);
  Devide(KL[1],Qn2,Qn2);
  Sum(Qn2,Qn1,V[1]);
  Product(YL[2,1],V[1],Qn1);
  MultiCmpx(-1,Qn1,Qn1);
  Conjugate(V[2],Qn2);
  Devide(KL[2],Qn2,Qn2);
  Sum(Qn2,Qn1,V[2]);
{  Write('c');}
 Until
  (ABS(V[1].Re-V01.Re) < 0.00000000001) AND (ABS(V[1].Im-V01.Im) < 0.00000000001) AND
  (ABS(V[2].Re-V02.Re) < 0.00000000001) AND (ABS(V[2].Im-V02.Im) < 0.00000000001);
END;

PROCEDURE Calyp
(Pl : Vtr;
 VAR yp : Vtr);
VAR
 I : Byte;
BEGIN
 I := 2;
```

```
 Conjugate(Pl[I],Qn1);
 MultiCmpx(1/(SQR(V[I].Re)+SQR(V[I].Im)),Qn1,yp[I]);
END;

PROCEDURE ModiYL
(yp : Vtr;
 VAR Ypp : Vtr;
 VAR YL : Mtx);
VAR
 I : Byte;
BEGIN
 I := 1;
 Ypp[I] := Y[I,I];
 I := 2;
 Sum(Y[I,I],yp[I],Ypp[I]);
 I := 3;
 Ypp[I] := Y[I,I];
 Devide(Y[1,2],Ypp[1],YL[1,2]);
 Devide(Y[1,3],Ypp[1],YL[1,3]);
 Devide(Y[2,1],Ypp[2],YL[2,1]);
END;

BEGIN      {Main}
 Assign(VF,'C:\TP\VBUS3.PRN'); Rewrite(VF);
{ Assign(MF,'C:\TP\MOTR.PRN'); Rewrite(MF);}
 Assign(GF,'C:\TP\GENTR3.PRN'); Rewrite(GF);
{ Assign(CR,'C:\TP\CURRENT.PRN'); Rewrite(CR);}
 Assign(EX,'C:\TP\EXFD3.PRN'); Rewrite(EX);
 Tim := 0;
 StTim := 0.002;
 Imm := 1;
 Loop := 1;
 XYtoCmpx(1,0,V[1]);
 XYtoCmpx(1,0,V[2]);
 XYtoCmpx(1.06,0,V[3]);
 Net(Y,YL);
 Flow(Y,YL,Pg,Pl,PQ,KL,V);
 Calyp(Pl,yp);
 ModiYL(yp,Ypp,YL);
{ ParaG(Rs,Rkq,Rfd,Rkd,Xs,Xlkq,Xlfd,Xlkd,Xmq,Xmd,H);}
 TransferG(Rs,Rkq,Rfd,Rkd,Xs,Xlkq,Xlfd,Xlkd,Xmq,Xmd,H);
 IniGn(Xs,Xmq,Xmd,Xlkq,Xlfd,Xlkd,Fds,Fqs,Fkq,Fkd,Ffd,Ti,Wr,Ang,Exfd,A,B);
 IniAVR(V1,V3,Vref,Vr,W3,Aex,Bex,Exfd,V);
{ IniGov(Ti,Y4,Y6,Y7);}
 ParaM(HpM,RsM,XsM,XmM,XrM,RrM,HM);
 IniStartM(V,RsM,XsM,XmM,XrM,FdsM,FqsM,FdrM,FqrM,WrM,Cf);
 IniMotor(XmM,XrM,RsM,XsM,YsM);
 Repeat
  If Tim > StTim then
  Begin
   If Loop = 1 Then
```

```
    Begin
     Loop := 2;
     IniStartM(V,RsM,XsM,XmM,XrM,PdsM,PqsM,PdrM,PqrM,WrM,Cf);
     StartM(V,Cf,RsM,XmM,XsM,HM,PdsM,PqsM,PdrM,PqrM,WrM);
    End
    Else
     StartM(V,Cf,RsM,XmM,XsM,HM,PdsM,PqsM,PdrM,PqrM,WrM);
    GnVolt(V,A,B,Rs,Exfd,H,Xmd,Rfd,Pds,Pqs,Pkq,Pkd,Pfd,Ti,Ang,Wr,dWrb,pWrb);
    AVR(Aex,Bex,Vref,V,Exfd,W3,V1,Vr,V3);
|   Governer(dWrb,pWrb,Y4,Y6,Y7,Ti);|
    Motor(XmM,XrM,PdrM,PqrM,YsM,JsM);
    Generator(Xmq,Xmd,Xlfd,Xlkd,Xlkq,Rs,Xs,Pfd,Pkd,Pkq,Ang,Ys,Js);
    CalIqd(Ys,Js,YsM,JsM,Iqd);
    Conjugate(V[1],Qn1);
    Product(Iqd[1],Qn1,P[1]);
    Devide(P[1],Y[1,1],KL[1]);
    Conjugate(V[2],Qn1);
    Product(Iqd[2],Qn1,P[2]);
    Devide(P[2],Y[2,2],KL[2]);
    Repeat
     VO1 := V[1]; VO2 := V[2];
     Product(YL[1,2],V[2],Qn1);
     MultiCmpx(-1,Qn1,Qn1);
     Product(YL[1,3],V[3],Qn2);
     MultiCmpx(-1,Qn2,Qn2);
     Sum(Qn2,Qn1,Qn1);
     Conjugate(V[1],Qn2);
     Devide(KL[1],Qn2,Qn2);
     Sum(Qn2,Qn1,V[1]);
     Product(YL[2,1],V[1],Qn1);
     MultiCmpx(-1,Qn1,Qn1);
     Conjugate(V[2],Qn2);
     Devide(KL[2],Qn2,Qn2);
     Sum(Qn2,Qn1,V[2]);
    Until
    (ABS(V[1].Re-VO1.Re) < 0.0000001) AND (ABS(V[1].Im-VO1.Im) < 0.0000001) AND
    (ABS(V[2].Re-VO2.Re) < 0.0000001) AND (ABS(V[2].Im-VO2.Im) < 0.0000001);
End
Else
Begin
 StartM(V,Cf,RsM,XmM,XsM,HM,PdsM,PqsM,PdrM,PqrM,WrM);
 GnVolt(V,A,B,Rs,Exfd,H,Xmd,Rfd,Pds,Pqs,Pkq,Pkd,Pfd,Ti,Ang,Wr,dWrb,pWrb);
 AVR(Aex,Bex,Vref,V,Exfd,W3,V1,Vr,V3);
|  Governer(dWrb,pWrb,Y4,Y6,Y7,Ti);|
|  Motor(XmM,XrM,PdrM,PqrM,YsM,JsM);
 Generator(Xmq,Xmd,Xlfd,Xlkd,Xlkq,Rs,Xs,Pfd,Pkd,Pkq,Ang,Ys,Js);
 CalIqd(Ys,Js,YsM,JsM,Iqd);
 Product(Y[2,1],V[1],Qn1);
 MultiCmpx(-1,Qn1,Qn1);
 Sum(Iqd[2],Qn1,Qn1);
 Devide(Qn1,Y[2,2],V[2]);|
```

```
 End;
 Tim := Tim+dT;
 Qn1.IM := SQRT(SQR(V[2].Re)+SQR(V[2].Im));
 Qn1.Re := SQRT(SQR(V[1].Re)+SQR(V[1].Im));
 Writeln(VF,Qn1.Re,' ',Qn1.Im);
 Write('>');
 Until Tim > TimEnd;
 Writeln('SUCCESS !');Readln;
 Close(VF); Close(GF);Close(EX);{ Close(MF);Close(CR)};
END.
```

ประวัติผู้เขียน

นาย ไชยวัฒน์ ผลลาภ เกิดวันที่ 25 ธันวาคม พ.ศ. 2507 ที่ อ.แกลง จ.ระยอง สำเร็จปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า จากจุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2530 แล้วได้ทำงานที่การไฟฟ้าส่วนภูมิภาคและศูนย์วิจัยและอบรมพลังงาน จุฬาลงกรณ์มหาวิทยาลัย หลังจากนั้นได้เข้าศึกษาต่อปริญญาโทในภาควิชาวิศวกรรมไฟฟ้า สาขาพลังงานไฟฟ้า ที่จุฬาลงกรณ์มหาวิทยาลัย ภาคปีการศึกษา 2531 ระหว่างศึกษาต่อได้ทำหน้าที่เป็นผู้ช่วยวิจัย ประจำศูนย์วิจัยและอบรมพลังงานจุฬาลงกรณ์มหาวิทยาลัย และมูลนิธิสถาบันวิจัยเพื่อการพัฒนาประเทศไทย