Scilab Textbook Companion for Thyristors Theory And Applications by R. K. Sugandhi And K. K. Sugandhi[1]

Created by
Santosh Paila
bachelor of technology
Electronics Engineering
Visvesvaraya National Institute of Technology
College Teacher
K.Surender
Cross-Checked by

August 17, 2013

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website http://scilab.in

Book Description

Title: Thyristors Theory And Applications

Author: R. K. Sugandhi And K. K. Sugandhi

Publisher: Wiley Eastern Limited, New Delhi

Edition: 2

Year: 1986

ISBN: 0-85226-852-1

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes

Chapter 2

THE DEVICE

Scilab code Exa 2.1 vlotage safety factor

```
//chapter 2:THE DEVICE
//Example 2.1
 Vpiv=1500;//peak inverse voltage//
 V=415;//main supply//
 Vf=Vpiv/(sqrt(2)*V);//voltage safety factor//
printf('value of voltage safety factor=%fv',Vf);
```

Scilab code Exa 2.2 peak inverse vlotage

```
//chapter 2:THE DEVICE
//Example 2.2
 Vf=2.1;//voltage safety factor//
 V=230;//main supply//
 Vpiv=sqrt(2)*Vf*V;//peak inverse voltage//
printf('value of peak inverse voltage=%fv',Vpiv);
```

Scilab code Exa 2.3 capacitive current

```
//chapter 2:THE DEVICE
//Example 2.3
 C=30*10^-12;//equivalent capacitance//
 diffV=150*10^6;//dv/dt value of capacitor//
 Ic=C*(diffV);//capacitive current//
printf('value of capacitive current=%fAmp',Ic);
```

Scilab code Exa 2.4 equivalent capacitance

```
//chapter 2:THE DEVICE
//Example 2.4
 Ic=5;//capacitive current in milli amperes//
 difV=175;//dv/dt value in mega V/s//
 C=Ic/(difV)*10^3;//equivalent capacitance in pico
    farad//
printf('value of equivalent capacitance=%fpico farad
   ',C);
```

Scilab code Exa 2.5 value of derivative of v

```
//chapter 2:THE DEVICE
//Example 2.5
 Ic=6*10^-3;//capacitive current//
 C=25*10^-12;//equivalent capacitance//
 diffV=Ic/C;//dv/dt value of capacitor//
printf('value of dv/dt=%fv/s',diffV);
```

Scilab code Exa 2.6 value of rate of voltage

```
//chapter 2:THE DEVICE
//Example 2.6////problem 2.1//
 Ic=5;//capacitive current in milli amperes//
 C=35;//equivalent capacitance in pico farad//
 difV=Ic*10^3/C;//value of dv/dt that can trigger
    the device in V/microseconds//
printf('value of dv/dt that can trigger the device=
   %fV/microseconds',difV);
```

Scilab code Exa 2.7 Voltage safety factor

```
//chapter 2:THE DEVICE//
//Example 2.7//problem2.3//
 Vpiv=1350;//peak inverse voltage in volts//
 V=415;//main supply in volts//
 Vf=Vpiv/(sqrt(2)*V);//voltage safety factor//
printf('value of voltage safety factor=%fv',Vf);
```

Chapter 3

Fabrication and Thermal characteristics

Scilab code Exa 3.1 junction temperature

```
//Fabrication and Thermal characteristics
//Example 3.1
Xa=50;//Ambient temperature//
P=150;//on state power loss in Watts//
Rjc=0.02;//junction_case thermal resistance//
Rcs=0.05;//case_sink thermal resistance//
Rsa=0.08;//sink_atmosphere thermal resistance//
Xj=Xa+P*(Rjc+Rcs+Rsa);//junction temperature//
printf('value of junction temperature=%fc',Xj);
```

Scilab code Exa 3.2 Maximum junction temperature

```
//Fabrication and Thermal characteristics
//Example 3.2
Xa=50;//Ambient temperature//
P20=25;//on state power loss at 20%load in Watts//
```

```
P200=350;//on state power loss at 200%load in Watts
    //
Rjc=0.02;//junction_case thermal resistance//
Rcs=0.05;//case_sink thermal resistance//
Rsa=0.12;//sink_atmosphere thermal resistance at 20%
     load cycle//
T1=60;//time period for the supply of 200% load//
T=((200^2-20^2)*T1)/(100^2-20^2);//time period of
    one cycle//
printf('value of time period of one cycle=%fs',T);
Ts=140;//thermal time constant for heat sink//
Xj20=Xa+P20*(Rjc+Rcs+Rsa);//junction temperature//
printf('\nvalue of junction temperature=%fc',Xj20);
P=P200-P20;//power required to cool down from 200
    %load cycle to 20% load cycle//
printf('\npower required to cool down=%fwatts',P);
Rsa200=((Rsa)*(1-exp(-T1/Ts)))/(1-exp(-T/Ts));//
    sink_atmosphere thermal resistance at 200% load
    cycle//
Xj200=Xj20+(P*(Rjc+Rcs+Rsa200));//maximum junction
    temperature//
printf('\nvalue of maximum junction temperature=%fc'
    ,Xj200);
```

Scilab code Exa 3.3 value of junction temperature

```
//Fabrication and Thermal characteristics
//Example 3.3
Xa=35;//Ambient temperature//
P=150;//on state power loss in Watts//
Rjc=0.01;//junction_case thermal resistance//
Rcs=0.08;//case_sink thermal resistance//
Rsa=0.09;//sink_atmosphere thermal resistance//
Xj=Xa+P*(Rjc+Rcs+Rsa);//junction temperature//
printf('value of junction temperature=%fc',Xj);
```

Scilab code Exa 3.4 on state power loss

```
//Fabrication and Thermal characteristics
//Example 3.4
Xa=45;//Ambient temperature//
Rjs=0.1;//junction_sink thermal resistance//
Rsa=0.08;//sink_atmosphere thermal resistance//
Xj=120;//junction temperature//
P=(Xj-Xa)/(Rjs+Rsa);//on state power loss//
printf('value of on state power loss=%fwatts',P);
```

Scilab code Exa 3.5 case sink thermal resistance

```
//Fabrication and Thermal characteristics
//Example 3.5
Xa=40;//Ambient temperature//
P=300;//on state power loss in Watts//
Rjc=0.015;//junction_case thermal resistance//
Rsa=0.1;//sink_atmosphere thermal resistance//
Xj=105;//junction temperature//
Rcs=((Xj-Xa)/(P))-(Rjc+Rsa);//case_sink thermal resistance//
printf('value of case sink thermal resistance=%fc/w',Rcs);
```

Chapter 4

Series and Parallel Connection of Thyristors

Scilab code Exa 4.1 Derating factor

```
//Series and Parallel Connection of Thyristors//
//Example 4.1//
Vc=3500;//voltage rating of circuit//
Vt=750;//voltage rating of each thyristor//
Ic=1500;//current rating of circuit//
It=500;//current rating of each thyristor//
DF=0.1;//Derating factor of circuit//
Ns=Vc/(Vt*(1-DF));//number of devices in series//
printf('Number of Devices in Series=%f',Ns);
Np=Ic/(It*(1-DF));//number of devices in parallel//
printf('\nNumber of Devices in Parallel=%f',Np);
```

Scilab code Exa 4.2 series connection

```
//Series and Parallel Connection of Thyristors//
//Example 4.2//
```

```
Ed=20;//permissible difference in voltage across devices in Volts//
Id=1*10^-3;//maximum difference in latching current across devices in Amperes//
Qd=10;//difference in recovery charge in Micro coloumbs//
Vd=20;//permissible difference in blocking voltage in Volts//
R=Ed/Id;//equivalent resistance in Ohms//
R1=R;
printf('value of equivalent resistance=R=%fohms',R=R1);
C1=Qd/Vd;//equivalent capacitance in Micro farads//
printf('\nvalue of equivalent capacitance=C1=%fmicrofarads',C1);
```

Chapter 5

Line Commutated converters

Scilab code Exa 5.1 AC terminal power

```
//Line commuted Converters//
//Example 5.1//
Edc=440;//dc terminal voltage of the thyristor in volts//
E2=415;//input voltage of the thyristor in volts//
Id=100;//dc motor current in amps//
C=Edc/(1.35*E2);
printf('cosine of the firing angle=C=%f',C);
A=acos(C)*180/%pi;
printf('firing angle of the converter=A=%fdegrees',A);
Pac=1.05*1.35*E2*Id/1000;//Ac terminal power in Kilo watts//
printf('AC terminal power=Pac=%fKW',Pac);
```

Scilab code Exa 5.2 Voltage regulation

```
//Line commuted Converters//
```

```
//Example 5.2//
Id=200;//rated dc current in amperes//
I2=0.817*Id;//AC line current in amperes//
printf('AC line current of the thyristor=I2=%famperes',I2);
E2=415;//AC line voltage in volts//
Xt=0.06*E2/I2;//effective reactance of the thyristor in ohms//
printf('\neffective reactance of the thyristor=Xt=%fohms',Xt);
C=1-((Id*Xt)/(E2*sqrt(3)));//cosine value of the commutational angle//
printf('\ncosine value of the commutational angle=C=%f',C);
CA=acos(C)*180/%pi;
printf('\ncommutation angle=CA=%fdegrees',CA);
IVR=(1-C)/2;//inductive voltage regulation//
printf('\nInductive voltage regulation=IVR=%f',IVR);
```

Scilab code Exa 5.3 Maximum dc voltage

```
//Line commuted Converters//
//Example 5.3//
E2=415;//input voltage in volts//
Edc=1.17*E2;//dc terminal voltage in volts//
Emax2=sqrt(2)*E2;//maximum value of dc voltage//
Z=2;//total impedance in ohms//
printf('maximum value of dc voltage=Emax2=%fvolts',Emax2);
Irms=Emax2*sqrt(%pi/3+sqrt(3)/4)/(2*%pi*Z);
printf('\nrms current through the device=Irms=%famps',Irms);
```

Scilab code Exa 5.4 Firing angle

```
//Line commuted Converters//
//Example 5.4//
Edc=460;//dc terminal voltage of the thyristor in
    volts//
E2=415;//input voltage of the thyristor in volts//
Id=200;//dc motor current in amps//
C=Edc/(1.35*E2);
printf('cosine of the firing angle=C=%f',C);
A=acos(C)*180/%pi;
printf('\nfiring angle of the converter=A=%fdegrees'
    ,A);
Pdc=Edc*Id/1000;//dc power delivered by the
    converter in kilo Watts
printf('\ndc power delivered by the converter=Pdc=
    %fKW',Pdc);
Pac=1.05*Pdc;//Ac terminal power in KVA//
printf('\nAC terminal power=Pac=%fKVA',Pac);
Iac=Pac*1000/(sqrt(3)*E2);
printf('\nAC line current=Iac=%famps',Iac);
Ib=0.58*Id;//Branch current through the device in
    amps//
printf('\nBranch current through the device=Ib=
    %famps',Ib);
```

Scilab code Exa 5.5 Reactance of the reactor

```
//Line commuted Converters//
//Example 5.5//
Id=150;//rated dc current in amperes//
E2=415;//AC line voltage in volts//
Emax=sqrt(2)*E2;
C=cos(16*%pi/180);//cosine value of the
    commutational angle//
```

```
printf('\ncosine value of the commutational angle=C=
    %f',C);
Xt=(1-C)*E2*sqrt(3)/Id;//effective reactance of the
    thyristor in ohms//
printf('\neffective reactance of the thyristor=Xt=
    %fohms',Xt);
```

Scilab code Exa 5.6 AC load current

```
//Line commuted Converters//
//Example 5.6//
E2=230;//AC line voltage in volts//
Emax=sqrt(2)*E2;
C=cos(13*%pi/180);//cosine value of the
    commutational angle//
Xt=0.16;//effective reactance of the thyristor in
    ohms//
Id=(1-C)*E2*sqrt(3)/Xt;//AC load current in amperes
    //
printf('AC load current=Id=%famps',Id);
```

Scilab code Exa 5.7 Average value of voltage

```
//Line commuted Converters//
//Example 5.7//
E2=230;//input voltage in volts//
Emax=sqrt(2)*E2;//maximum value of dc voltage//
A=%pi/6;
Edc=Emax*(1+cos(A))/(2*%pi);
printf('Average value of dc voltage=Edc=%fvolts',Edc
    );
Eeff=Emax*sqrt((%pi-A)/(4*%pi)+(sin(2*A)/(8*%pi)));
```

```
printf('\nEffective value of voltage=Eeff=%fvolts',
   Eeff);
R=10;//total impedance in ohms//
Id=Edc/R;
printf('\nLoad current=Id=%famps',Id);
```

Scilab code Exa 5.8 DC output voltage

```
//Line commuted Converters//
//Example 5.8//
E2=415;//input voltage in volts//
Emax=sqrt(2)*E2;//maximum value of dc voltage//
A=%pi/6;//triggering angle in degrees//
Edc=Emax*cos(A)/%pi;//dc output voltage in volts//
printf('dc output voltage=Edc=%fvolts',Edc);
```

Chapter 7

Inverter Circuits

Scilab code Exa 7.1 Attenuation factor

```
//Inverter Circuits//
//Example 7.1//
L=10*10^-3;//Inductance of series inverter circuit
   in Henry//
C=0.1*10^-6;//Capacitance of series inverter circuit
    in Farads//
R=400;//Load Resistance in Ohms//
Toff=0.2*10^-3;//Off time of Duty cycle in sec//
w=sqrt((1/(L*C))-(R^2/(4*L^2)));//Angular Frequency
   in rad/sec//
printf('value of w=%f',w);
F=w/(3.14+(w*Toff));//Output Frequency in Hertz//
printf('\nvalue of the Output Frequency=F=%fHertz',F
   );
T=1/F;//Time period of Output in sec//
AF=exp((-R/(2*L))*T);//Attenuation Factor//
printf('\nvalue of the Attenuation Factor=AF=%f',AF)
   ;
```

Scilab code Exa 7.2 Value of inductance

```
//Inverter Circuits//
//Example 7.2//
C=1*10^-6;//Capacitance of series inverter circuit
   in Farads//
f=5*10^3;//operating Frequency of series Inverter in
    Hertz//
L=1/(C*(f^2));//value of Inductance under Resonance
   condition in Henry//
printf('value of Inductance at resonance=L=%fHenry',
   L);
```

Scilab code Exa 7.3 Value of R1

```
//Inverter Circuits//
//Example 7.3//
L=5*10^-3;//Inductance of series inverter circuit in
    Henry//
C=1*10^-6;//Capacitance of series inverter circuit
   in Farads//
Rl=400;//Load Resistance in Ohms//
R2=10^4;//value of the second resistance in Ohms//
DF=0.7;//Damping Factor value of LC filter//
R1=(2*(DF)*(sqrt(L/C))-R2-(1/(Rl*C));//value of the
    first resistance in Ohms//
printf('value of resistance=R1=%fOhms',R1);
```

Chapter 8

Harmonic and PowerFactor with the converter system

Scilab code Exa 8.1 Shunt filter

```
//Harmonic and Powerfactor with the Converter system//
//Example 8.1//
I5=0.2;//amplitude of 5th harmonic current in Kilo Amperes//
Vp= 11/(sqrt(3));//Input supply phase voltage in Kilo Volts//
P=5;//supply power per phase of filter in MVAR//
Pc=P+((Vp^2*I5^2)/(5*P));//AC Converter power per phase in MVAR//
printf('\nvalue of AC converter power=Pc=%f MVAR',Pc);
C=(Pc*10^3*3)/(11^2*314);//capacitance of the ShuntFilter in milliFarad//
printf('\nvalue of the capacitance of shunt filter=C=%fmillifarads',C);
L=(106*10^6)/(400*4*25*250*3.14^2);//inductance of filter in mHenry//
printf('\nInductance of filter=L=%fmilliHenry',L);
```

```
Q=50;//value of Q//
W5=2*3.14*5*50;//angular frequency of 5th harmonic//
R=(W5*L)/Q;//Resistance of filter in milliOhms//
printf('\nResistance of filter=R=%fmilliOhms',R);
```

Scilab code Exa 8.2 DC reactor circuit

```
//Harmonic and Powerfactor with the Converter system//
//Example 8.2//
printf('For six pulse converter most effective harmonic is 6th and for worst case a=90 degree\n');
Wv=24.1;//voltage ripple in percentage//
printf('voltage ripple=Wv=%fpercent',Wv);
Id=200;
I6=(5*Id)/100;//Harmonic current for 6th harmonic in amp//
printf('\nHarmonic current for 6th harmonic=I6=%famp',I6);
Edc=460;//dc voltage in volts//
W=2*3.14*50;
La=1;//inductance already present in the circuit in milliHenry//
L=((Wv*Edc*10)/(I6*6*W))-La;//additional inductance required in milliHenry//
L=5.93-1;
printf('\nadditional inductance required=L=%fmilliHenry',L);
```

Scilab code Exa 8.3 Commutation angle

```
//Harmonic and Powerfactor with the Converter system //
//Example 8.3//
Id=200;//rated dc current in amperes//
I2=0.817*Id;//AC line current in amperes//
printf('AC line current of the thyristor=I2=%famperes',I2);
E2=415;//AC line voltage in volts//
Edc=400;//dc terminal voltage in volts//
Xt=0.04*E2/I2;//effective reactance of the thyristor in ohms//
printf('\neffective reactance of the thyristor=Xt=%fohms',Xt);
C=1-((Id*Xt)/(E2*sqrt(3)));//cosine value of the commutational angle//
printf('\ncosine value of the commutational angle=C=%f',C);
CA=acos(C)*180/%pi;
printf('\ncommutation angle=CA=%fdegrees',CA);
F=Edc/(1.35*E2*(1+C)/2);//cosine value of the firing angle//
printf('\ncosine value of the firing value=F=%f',F);
FA=acos(F)*180/%pi;
printf('\nfiring angle=FA=%fdegrees',FA);
I2=0.817*Id;//AC line current in amps//
printf('\nAC line current=I2=%famps',I2);
Ied=0.58*Id;//current through each device in amps//
printf('\nCurrent through each device=Ied=%famps',Ied);
PF=F*(1+C)/2;//power factor//
printf('\npower factor=PF=%f',PF);
AP=sqrt(3)*E2*I2*PF;//active power drawn from the mains in Watts//
printf('\nactive power drawn from the mains=AP=%fWatts',AP);
RP=sqrt(3)*E2*I2*sqrt(1-PF^2);//reactive power in VAR//
printf('\nReactive power drawn=RP=%fVAR',RP);//end
```

```
of the program//
```

Scilab code Exa 8.4 Rating of shunt compensator

```
//Harmonic and Powerfactor with the Converter system
   //
//Example 8.4//
Id=100;//rated dc current in amperes//
I2=0.817*Id;//AC line current in amperes//
printf('AC line current of the thyristor=I2=
   %famperes',I2);
E2=230;//AC line voltage in volts//
Edc=200;//dc terminal voltage in volts//
PF=cos(%pi/4)*(1+cos(%pi/10))/2;//power factor//
printf('\npower factor=PF=%f',PF);
RP=sqrt(3)*E2*I2*sqrt(1-PF^2);//reactive power to be
    supplied by shunt compensator in VAR//
printf('\nReactive power to be supplied by shunt
   compensator=RP=%fVAR',RP);//end of the program//
```

Scilab code Exa 8.5 Shunt filter

```
//Harmonic and Powerfactor with the Converter system
   //
//Example 8.5//
I11=400/11;//amplitude of 11th harmonic current in
   Amperes//
V1= 11/(sqrt(3));//Input supply phase voltage in
   Kilo Volts//
P=7;//supply power per phase of filter in MVAR//
Pc=P+((V1^2*I11^2*10^-3)/(11*P));//AC Converter MVAR
    rating of the capacitor//
```

```
printf('value of MVAR rating of the capacitor=Pc=%fMVAR',Pc);
W=2*3.14*50;
C=(Pc*10^6)/(V1^2*W);//capacitance of the ShuntFilter in microFarad//
printf('\nvalue of the capacitance of shunt filter=C=%fmicrofarads',C);
W11=11*W;
L=10^8/(C*W11^2);//inductance of filter in mHenry//
printf('\nInductance of filter=L=%fmilliHenry',L);
Q=35;//value of Q//
R=(W11*L)/Q;//Resistance of filter in milliOhms//
printf('\nResistance of filter=R=%fmilliOhms',R);
```

Scilab code Exa 8.6 Maximum current ripple

```
//Harmonic and Powerfactor with the Converter system//
//Example 8.6//
printf('For six pulse converter most effective harmonic is 6th and for worst case a=90 degree\n');
h=6;
Wv=24.1;//voltage ripple in percentage//
printf('voltage ripple=Wv=%fpercent',Wv);
Edc=460;//dc voltage in volts//
W=2*3.14*50;
Ldc=6;//total dc circuit inductance in milliHenry//
I6=Wv*Edc*10/(Ldc*h*W);//Harmonic current for 6th harmonic in amp//
printf('\nHarmonic current for 6th harmonic=I6=%famp',I6);
Id=300;
Wi=100*I6/Id;//maximum value of current ripple in percentage//
```

```
printf('\nmax. value of current ripple=Wi=%fpercent'
    ,Wi);//end of program//
```

Scilab code Exa 8.7 Voltage ripple

```
//Harmonic and Powerfactor with the Converter system
    //
//Example 8.7//
A=%pi/4;
h=6;
Wv=sqrt(2)*sqrt(h^2-cos(A)^2*(h^2-1))*100/(h^2-1);
printf('voltage ripple of the 6th harmonic=Wv=
    %fpercent',Wv);
printf('\nFor six pulse converter most effective
    harmonic is 6th and for worst case A=90degrees\n'
    );
A=%pi/2;
Wv6=sqrt(2)*sqrt(h^2-cos(A)^2*(h^2-1))*100/(h^2-1);
    //maximum voltage ripple in percentage//
printf('\nmaximum voltage ripple=Wv6=%fpercent',Wv6)
    ;
A=%pi/4;
h=12;
Wv=sqrt(2)*sqrt(h^2-cos(A)^2*(h^2-1))*100/(h^2-1);
printf('\nvoltage ripple of the 12th harmonic=Wv=
    %fpercent',Wv);
A=%pi/2;
Wv12=sqrt(2)*sqrt(h^2-cos(A)^2*(h^2-1))*100/(h^2-1);
    //maximum voltage ripple in percentage//
printf('\nmaximum voltage ripple=Wv12=%fpercent',
    Wv12);
PR=(Wv6-Wv12)*100/Wv6;//percentage reduction in max.
     voltage ripple//
printf('\npercentage reduction in max. voltage
    ripple=PR=%fpercent',PR);
```

Scilab code Exa 8.8 Triggering angle

```
//Harmonic and Powerfactor with the Converter system//
//Example 8.8//
Wv=18.6;
h=6;
C=sqrt(h^2-(Wv^2*((h^2-1)^2)/2*10^4))*10^6/sqrt(h^2-1);//cosine of triggering angle//
C=sqrt(14.68/35);
printf('cosine of triggering angle=C=%f',C);
A=acos(C)*180/%pi;
printf('\ntriggering angle of the device=A=%fdegrees',A);//endof program//
```

Scilab code Exa 8.9 Power factor

```
//Harmonic and Powerfactor with the Converter system//
//Example 8.9//
E2=415;//AC line voltage in volts//
Edc=380;//dc terminal voltage in volts//
C=1.1*Edc/(1.35*E2);
printf('cosine of the triggering angle=C=%f',C);
A=acos(C)*180/%pi;
printf('\ntriggering angle of the device=A=%fdegrees',A);
PF=C*(1+cos(%pi/12))/2;//power factor//
printf('\npower factor=PF=%f',PF);
Id=200;
I2=0.817*Id;
```

```
RP=sqrt(3)*E2*I2*sqrt(1-PF^2)/1000;//reactive power
    to be supplied by shunt compensator in KVAR//
printf('\nReactive power to be supplied by
    shuntcompensator=RP=%fKVAR',RP);//end of the
    program//
```

Chapter 11

Control of DC Motors

Scilab code Exa 11.1 Designing a thyristor

```
//Control of DC motors//
//Example 11.1//
//Since the speed control is required in both directions we will have to use a dual converter for the application.It would be prefarable to use six pulse dual converter with thyristors connected in antiparallel connection//
//speed control from 20% rated speed to 100% rated speed will be obtained by armature control//
//Control and speed above 100% will be possible by field weakening//
Idc=200/460*1000;//Rated motor current in amps//
printf('Rated motor current=Idc=%famps',Idc);
//Thus the main armature converter will be having dc side rating of 500Amps and 460volts//
//If 20% drop is allowed in cables,ac transformer,converter etc., then No load dc voltage required =460*1.2=552Volts//
printf('\nHence AC voltage for six pulse configuration=552/1.35=410volts');
//Hence a 3phase,415v AC supply will be adequate for
```

```
armature control//
//Field converter rating will be 230V,10A. Arrangement will be six pulse,non reversible. since AC supply of 415V,3 phase is available,we shall make use of it for field converter also.//
printf('\nAC rating of field converter=230/1.35=170V');
//However we shall provide a standard AC voltage of 230V AC and will lock the field converter firing angle to suitable value so as to produce 230V dc//
printf('\nDC power=230*10=2300Watts');
printf('\nAC power=1.05*2300=2415Watts');
printf('\nThus tranformer of 2.5KVA,415/230V will be required');
Edca=(170+170/10)*1.35;//available voltage in volts//
Edc=1.35*230;
A=acos(Edca/Edc)*180/%pi;
printf('\nField converter shall be locked at an angle of A=%fdegrees',A);
```

Scilab code Exa 11.2 Blocking angle

```
//Control of DC motors//
//Example 11.2//
Vdc=440;//Rated dc voltage in volts//
Edca=Vdc+Vdc/10;//Required voltage after allowing 10% drop//
printf('Required voltage after allowing 10percent drop=Edca=%fvolts',Edca);
Edc=1.35*415;
C=Edca/Edc;
printf('\nCosine of the locked angle=C=%f',C);
A=acos(C)*180/%pi;//locked angle in degrees//
```

```
printf('\nConverter shall be locked at an angle of A
   =%fdegrees',A);
```

Scilab code Exa 11.3 Firing angle

```
//Control of DC motors//
//Example 11.3//
Edca1=230;
N1=1000;
N2=500;
Eb1=210;
printf('Eb1=230-20=210volts');
Eb2=Eb1*N2/N1;
printf('\nEb2=%fvolts',Eb2);
V=40;//motor armature drop at rated load in volts//
Edca2=Eb2+V;
printf('\nEdca2=%fvolts',Edca2);
C1=1;//cosine of the firing angle corresponding to
   1000 rpm load//
C2=C1*Edca2/Edca1;//cosine of the firing angle
   corresponding to 500 rpm load//
printf('\nCosine of the firing angle corresponding
   to 500 rpm load=C2=%f',C2);
A=acos(C2)*180/%pi;//firing angle corresponding to
   500 rpm load in degrees//
printf('\nfiring angle corresponding to 500 rpm load
    A=%fdegrees',A);
```

Scilab code Exa 11.4 Reactive power

```
//Control of DC motors//
//Example 11.4//
```

```
Edca1=1.15*440;//Rated output voltage from the converter for rated speed of750rpm//
printf('Rated output voltage from the converter=Edca1=%fvolts',Edca1);
N1=750;
N2=500;
Edca2=Edca1*N2/N1;
E2=415;
C2=Edca2/(1.35*E2);
printf('\nCosine of the triggering angle=C2=%f',C2);
A2=C2*180/%pi;
printf('\nTriggering angle=A2=%fdegrees',A2);
PF2=C2*(1+cos(15*%pi/180))/2;
printf('\nPowerfactor=PF2=%f',PF2);
Id=200;//dc current in amps//
I2=0.75*0.817*Id;//Current at 75percent load in amps//
RP2=sqrt(3)*E2*I2*sqrt(1-PF2^2)/1000;//Reactive power drawn at 75% load//
printf('\nReactive power at 75percent load=RP2=%fKVAR',RP2);
h=6;
Wv=24.17;//maximum voltage ripple in percent//
Wi=8;//maximum permissible current ripple in percent//
I6=Wi*Id/100;
printf('\nSixth harmonic ripple current=I6=%fAmps',I6);
W=314;
L=(Wv*Edca1*10)/(I6*h*W);
printf('\nInductance required in dc circuit=L=%fmH',L);
C1=Edca1/(1.35*E2);
printf('\nCosine of the triggering angle=C1=%f',C1);
A1=C1*180/%pi;
printf('\nTriggering angle=A1=%fdegrees',A1);
PF1=C1*(1+cos(15*%pi/180))/2;
printf('\nPowerfactor=PF1=%f',PF1);
```

```
I1=0.817*Id;//Current at 75percent load in amps//
RP1=sqrt(3)*E2*I1*sqrt(1-PF1^2)/1000;//Reactive power drawn at 75% load//
printf('\nReactive power at 75percent load=RP1=%fKVAR',RP1);
```

Scilab code Exa 11.5 Active and Reactive power

```
//Control of DC motors//
//Example 11.5//
Edca=460;
E2=415;
C=Edca/(1.35*E2);
printf('\nCosine of the triggering angle=C=%f',C);
A=C*180/%pi;
printf('\nTriggering angle=A=%fdegrees',A);
Edca10=0.1*460;
C10=Edca10/(1.35*E2);
printf('\nCosine of the triggering angle=C10=%f',C10);
A10=C10*180/%pi;
printf('\nTriggering angle=A10=%fdegrees',A10);
Id=10^5/Edca;//dc current in amps//
I2=0.817*Id;//Current at rated speed in amps//
AP=sqrt(3)*E2*I2*C/1000;
printf('\nActive power drawn from the system at rated speed=AP=%fKW',AP);
RP=sqrt(3)*E2*I2*sqrt(1-C^2)/1000;//Reactive power drawn from the system//
printf('\nReactive power drawn from the system=RP=%fKVAR',RP);
AP10=sqrt(3)*E2*I2*C10/1000;
printf('\nActivepower drawn from thesystem at 10 percentrated speed=AP10=%fKW',AP10);
RP10=sqrt(3)*E2*I2*sqrt(1-C10^2)/1000;//Reactive
```

```
power drawn from the system//
printf('\nReactive power drawn from the system=RP10=%fKVAR',RP10);
P=RP10/RP;
printf('\nP=%f',P);
printf('\nThus reactive power has increased by 74.5893percent due to reduction in motor speed');
```

Scilab code Exa 11.6 Power at given load

```
//Control of DC motors//
//Example 11.6//
printf('Reactive power at rated speed and rated load =72.79KVAR');
printf('\nReactive power at rated speed and 10 percent load=0.1*72.79=7.279KVAR');
printf('\nSimilarly reactive power at 10percent speed and 10 percent load=0.1*127.08=12.71KVAR');
```

Scilab code Exa 11.7 Triggering angle

```
//Control of DC motors//
//Example 11.7//
N1=500;
N2=400;
Eb1=410;
Eb2=Eb1*N2/N1;
printf('Eb2=%fvolts',Eb2);
V=440;//operating voltage of dc motor in volts//
P=100;//input power of dc motor in KW//
Ia=P*1000/V;
printf('\nIa=%fAmps',Ia);
Ra=(V-Eb1)/Ia;
```

```
printf('\nRa=%fohms',Ra);
E2=415;
Edca=Eb2+(0.75*Ia*Ra);//terminal voltage of dc motor at 500 rpm and 75% load//
printf('\nTerminal voltage of dc motor at 500 rpm and 75percent load=Edca=%fvolts',Edca);
C=Edca/(1.35*E2);//cosine of the triggering angle of the converter//
printf('\nCosine of the triggering angle of the converter=C2=%f',C);
A=acos(C)*180/%pi;//triggering angle of the converter in degrees//
printf('\ntriggering angle of the converter A=%fdegrees',A);
```

Chapter 12

Controllers and Their Optimisation

Scilab code Exa 12.1 Permanent error of p contoller

```
//Controllers and Their Optimisation//
//Example 12.1//
V=40;//gain of the controller in volts//
P=100/(1+V);//permanent error of p controller in
    percent//
printf('permanent Error of P controller=P=%fpercent'
    ,P);
```

Scilab code Exa 12.2 Motor armature time constant

```
//Controllers and Their Optimisation//
//Example 12.2//
P=1.8;//permanent error of p controller in percent//
V=100/1.8-1;//gain of the controller in volts//
printf('gain of the controller=V=%fvolts',V);
G=8;//sum of all time constants in milliseconds//
```

```
T1=2*G*V;//motor armature time constant//
printf('\nMotor armature time constant=T1=
   %fmilliseconds',T1);
```

Scilab code Exa 12.3 Controller parameters

```
//Controllers and Their Optimisation//
//Example 12.3//
f=50;//frequency in hz//
p=6;//pulse number//
t1=1000/(2*f*p);//time constant for the current loop
    in ms//
printf('time constant for the current loop=t1=%fms',
   t1);
t2=1.5;//time constant of feedback channel in ms//
G=t1+t2;//smaller time constant in ms//
printf('\nSmaller time constant=G=%fms',G);
T1=30;//bigger time constant in ms//
Tn=T1;//time constant of the controller in ms//
printf('\nTime constant of the controller in AVO=Tn=
   %fms',Tn);
V=T1/(2*G);//gain of the control system//
printf('\nGain of the control system=V=%f',V);
Vg=14;//gain of the regulating current link//
Vr=V/Vg;//gain of the PI controller//
printf('\nGain of the PI controller=Vr=%f',Vr);
R2=11;//R2 in KiloOhms//
R1=R2/Vr;//R1 in kiloohms//
printf('\nR1=%fKiloohms',R1);
C1=Tn/R1;//C1 in microfarads//
printf('\nC1=%fmicrofarads',C1);
```

Scilab code Exa 12.4 Designing a PI regulator

```
//Controllers and Their Optimisation//
//Example 12.4//
G=20;//smaller time constant in ms//
T1=350;//bigger time constant in ms//
Tn=4*G;//time constant of the controller in ms//
printf('\nTime constant of the controller in SO=Tn=
    %fms',Tn);
V=T1/(2*G);//gain of the control system//
printf('\nGain of the control system=V=%f',V);
Vg=1;//gain of the regulating current link//
Vr=V/Vg;//gain of the PI regulator//
printf('\nGain of the PI regulator=Vr=%f',Vr);
R1=11;//R1 in KiloOhms//
R2=R1*Vr;//R2 in kiloohms//
printf('\nR2=%fKiloohms',R2);
C2=Tn/R2;//C1 in microfarads//
printf('\nC2=%fmicrofarads',C2);
```

Scilab code Exa 12.5 Time constant of the controller

```
//Controllers and Their Optimisation//
//Example 12.5//
G=6;//smaller time constant in ms//
T1=80;//bigger time constant in ms//
Tn=T1;//time constant of the controller in ms//
printf('Time constant of the controller=Tn=%fms',Tn)
    ;
V=T1/(2*G);//gain of the control system//
printf('\nGain of the control system=V=%f',V);
Wn=1/(sqrt(2)*G);//Natural frequency of the system
    in rad/ms//
printf('\nNatural frequency of the system=Wn=%frad/
    ms',Wn);
Tf=4.7*G;//time taken by the system to achiecve its
    desired output for firsttime//
```

```
printf('\ntime taken by the system to achieve its desired value=Tf=%fms',Tf);
printf('\nMaximum overshoot for a symmetrically optimised system is 4.3 percent');
Tmax=6.24*G;//time at which maximum overload will occur in ms//
printf('\nTime at which maximum overload will occur=Tmax=%fms',Tmax);
```

Scilab code Exa 12.6 Maximum overshoot

```
//Controllers and Their Optimisation//
//Example 12.6//
G=20;//smaller time constant in ms//
Tn=4*G;//time constant of the controller in ms//
printf('time constant of the controller=Tn=%fms',Tn);
T1=170;//bigger time constant in ms//
V=T1/(2*G);//gain of the control system//
printf('\nGain of the control system=V=%f',V);
Tf=3.1*G;//time taken by the system to achiecve its final value on step input//
printf('\ntime taken by the system to achieve its final value=Tf=%fms',Tf);
printf('\nMaximum overshoot for a symmetrically optimised system is 43 percent');
```

Scilab code Exa 12.7 Settling time

```
//Controllers and Their Optimisation//
//Example 12.7//
G=10;//smaller time constant in ms//
```

```
Tf=4.7*G;//time taken by the system to achiecve its
    final output for firsttime//
printf('time taken by the system to achieve its
    final value=Tf=%fms',Tf);
printf('\nMaximum overshoot for a symmetrically
    optimised system is 4.3 percent');
Tmax=6.24*G;//time at which maximum overshoot will
    occur in ms//
printf('\nTime at which maximum overshoot will occur
    =Tmax=%fms',Tmax);
Ts=8.4*G;//settling time in ms//
printf('\nSettling time=Ts=%fms',Ts);
```

Scilab code Exa 12.8 Difference in response

```
//Controllers and Their Optimisation//
//Example 12.8//
printf('Response for an AVO system');
G=10;//smaller time constant in ms//
Tf=4.7*G;//time taken by the system to achiecve its
    final output for firsttime//
printf('\ntime taken by the system to achieve its
    final value=Tf=%fms',Tf);
printf('\nMaximum overshoot for a symmetrically
    optimised system is 4.3 percent');
Ts=8.4*G;//settling time in ms//
printf('\nSettling time=Ts=%fms',Ts);
printf('\nResponse for an SO system');
G=10;//smaller time constant in ms//
Tf=3.1*G;//time taken by the system to achiecve its
    final output for firsttime//
printf('\ntime taken by the system to achieve its
    final value=Tf=%fms',Tf);
printf('\nMaximum overshoot for a symmetrically
    optimised system is 43 percent');
```

```
Ts=16.6*G;//settling time in ms//
printf('\nSettling time=Ts=%fms',Ts);
```

Chapter 13

Choppers and Transportation system Application

Scilab code Exa 13.1 Instantaneous current

```
//Choppers and Transportation System Application//
//Example 13.1//
E=220;//dc supply voltage in volts//
El=22;//Load voltage in volts//
Ton=1000;//conducting period in microseconds//
T=2500;//Total timeperiod in microseconds//
L=1;//inductance in milliHenry//
R=0.25;//resistance in ohms//
t=L/R;//time constant in milliseconds//
printf('time constant=t=%fmilliseconds',t);
A=0.133;
Td=A*T;//Discontinuous condition starts at//
printf('\nDiscontinuous condition starts from Td=
    %fmicroseconds',Td);
Eo=0.4*E;//output voltage in volts//
printf('\nOutput voltage=Eo=%fvolts',Eo);
Iav=(Eo-El)/R;//Average current in amps//
printf('\nAverage current=Iav=%famp',Iav);
Imax=((E*(1-exp(-Ton/(t*1000))))/(R*(1-exp(-T/(t
```

```
   *1000)))))-(El/R);
printf('\nMaximum current=Imax=%famp',Imax);
Imin=((E*(exp(Ton/(t*1000))-1))/(R*(exp(T/(t*1000))
   -1)))-(El/R);
printf('\nMinimum current=Imin=%famp',Imin);
```

Scilab code Exa 13.2 Conduction and Blocking period

```
//Choppers and Transportation System Application//
//Example 13.2//
f=1;//operating frequency in KHZ//
E=220;//dc supply voltage in volts//
El=165;//Load voltage in volts//
Ton=El/(E*f);//conduction period in ms//
printf('Conduction period=Ton=%fms',Ton);
T=1/f;//total time period in ms//
printf('\nTotal time period=T=%fms',T);
Toff=T-Ton;//blocking period in ms//
printf('\nBlocking period=Toff=%fms',Toff);
```

Scilab code Exa 13.3 Optimum frequency

```
//Choppers and Transportation System Application//
//Example 13.3//
E=220;//dc supply voltage in volts//
Toff=200;//blocking period in microseconds//
Il=50;//load current in amps//
C=%pi*Toff*Il/(2*E);//capacitance for optimum
   frequency in microfarad//
C=75;
printf('Load capacitance required for optimum
   frequency=C=%fmicrofarad',C);
```

```
L1=Toff^2*10^-3/C;//inductance required in milliHenry//
L2=L1;
printf('\nInductance parameters=L1=L2=%fmilliHenry',L1);
```

Scilab code Exa 13.4 Required pulse width

```
//Choppers and Transportation System Application//
//Example 13.4//
E=220;//dc supply voltage in volts//
El=660;//Load voltage in volts//
Toff=100;//blocking period in microseconds//
Ton=(El/E-1)*Toff;//Conduction period in microseconds//
printf('Conduction period=Ton=%fmicroseconds',Ton);
```

Scilab code Exa 13.5 Pulse width

```
//Choppers and Transportation System Application//
//Example 13.5//
f=200;//chopper frequency in HZ//
E=220;//dc supply voltage in volts//
Iav=100;//Average current in the circuit in amps//
Ra=0.02;//Armature resistance in ohms//
Rf=0.01;//Field resistance in ohms//
Ebav=50;//Average value of the Back emf in volts//
Eav=Iav*(Ra+Rf)+Ebav;//Average voltage in the circuit in volts//
printf('Average voltage in the circuit=Eav=%fvolts',Eav);
Ton=Eav*1000/(E*f);//conduction period in ms//
printf('\nConduction period=Ton=%fms',Ton);
```

Scilab code Exa 13.6 Motor Torque

```
//Choppers and Transportation System Application//
//Example 13.6//
f=200;//chopper frequency in HZ//
T=1000/f;//total time period in ms//
Toff=4;//Blocking period in ms//
Ton=T-Toff;//conduction period in ms//
R1=2;//R1 in ohms//
R2=4;//R2 in ohms//
R=((R1*Ton)+(R1+R2)*Toff)/T;//rotor resistance
    referred to stator in ohms//
printf('Rotor resistance referred to stator=R=%fohms
    ',R);
V=415;//stator voltage in volts//
s=0.02;//slip of the motor//
MT=V^2*s/R;//motor torque in Syn. Watts//
printf('\nMotor torque=MT=%fSnc. Watts',MT);
```

Scilab code Exa 13.7 Chopper Frequency

```
//Choppers and Transportation System Application//
//Example 13.7//
//R1=rotor resistance before introduction of control
    //
//R2=rotor resistance after introduction of control
    //
printf('R2=1.5*R1');
R2=((R1*Ton)+(R1+R1)*Toff)/T;//rotor resistance
    referred to stator in ohms//
printf('\nthe above condition satisfies when Ton=
    Toff');
```

```
T=4;//total time period in ms//
f=1000/T;//chopper frequency in hz//
printf('\nChopper frequency=f=%fhz',f);
```

Chapter 15

The AC motor control

Scilab code Exa 15.1 Stator current

```
//The ac Motor Control//
//Example 15.1//
S1=2;//value of slip in percentage of slip ring
   induction motor//
Ns=1000;//value of stator speed in rpm//
Nr=500;//value of rotor speed in rpm//
S2=(Ns-Nr)*100/Ns;//valu of slip in percentage of
   motor//
printf('value of slip of motor=S2=%fpercentage',S2);
I1=50;//stator current in amps//
I2=I1*sqrt(S2/S1);
printf('\nvalue of new stator current=I2=%fAmp',I2);
```

Scilab code Exa 15.2 Designing a thyristor converter

```
//The ac Motor Control//
//Example 15.2//
Imr=50;//motor field rating in amp//
```

```
Icr=1.5*Imr;//converter rated current in amp//
printf('value of converter rated current=Icr=%famp',
   Icr);
Vdc=100;//converter dc rating in volts//
Vac=Vdc/1.35;//converter ac rating voltage required
   //
printf('\nvalue of converter rated ac voltage=Vac=
   %fvolts',Vac);
Pkva=(1.05*100*75)/1000;//KVA rating of the
   transformer//
printf('\nKVA rating of transformer=Pkva=%fKVA',Pkva
   );
```

Scilab code Exa 15.3 Torque developed by the motor

```
//The ac Motor Control//
//Example 15.3//
S1=0.04;//value of slip in of induction motor//
Ns=1500;//value of initial speed in rpm//
N2=1300;//value of speed reduced to in rpm//
N1=Ns*(1-S1);//valu of speed N1 in rpm//
printf('value of speed N1=%frpm',N1);
f=(Ns-N1)/(Ns-N2);
printf('\nvalue of f=%f',f);
T1=2000;//developing torque in induction motor in
   watts//
T2=T1/f;//new value of torque developed by the motor
    in watts//
printf('\nvalue of new torque developed=T2=%fWatts',
   T2);
```

Scilab code Exa 15.4 Torque developed by the motor

```
//The ac Motor Control//
//Example 15.4//
f1a=50;//intial frequency in hertz//
f1b=75;//value of frequency increased to in hertz//
Ta=1500;//developing torque in induction motor in
   watts//
Tb=Ta*f1a/f1b;//new value of torque developed by the
    motor in watts//
printf('value of new torque developed=Tb=%fWatts',Tb
   );
```

Scilab code Exa 15.5 Rotor Frequency

```
//The ac Motor Control//
//Example 15.5//
V=415;//operating input voltage of induction motor
   in volts//
S=0.04;//input slip//
r2=1;//rotor resistance referred to stator in ohms//
T=(S*V^2)/r2;//torque developed by motor in watts//
printf('torque developed by motor=T=%fwatts',T);
f1=75;//input stator frequency in hertz//
f2=S*f1;//rotor frequency in hertz//
printf('\nvalue of rotor frequency=f2=%fhertz',f2);
```

Scilab code Exa 15.6 Input voltage to the motor

```
//The ac Motor Control//
//Example 15.6//
f1a=50;//intial frequency in hertz//
f1b=30;//value of frequency reduced to in hertz//
Va=415;//operating voltage of induction motor in
   volts//
```

```
Vb=Va*f1b/f1a;//input voltage to the motor in volts//
printf('value of input voltage to the motor=Vb=%fvolts',Vb);
Pa=100;//operating power of induction motor in KVA//
Pb=Pa*f1b/f1a;//input power to the motor in KVA//
printf('\nvalue of input power to the motor=Pb=%fKVA',Pb);
```

Scilab code Exa 15.7 Rotor copper loss

```
//The ac Motor Control//
//Example 15.7//
f1a=40;//intial frequency in hertz//
Pa=200;//input power of squirrel cage motor in KVA//
Pb=150;//input power to the motor after change in speed in KVA//
f1b=f1a*Pb/Pa;//frequency changed to in hertz//
printf('value of frequency changed to f1b=%fhz',f1b);
Nsa=1200;//motor initial syncronous speed in rpm//
Nsb=Nsa*f1b/f1a;
Sb=0.04;
Nb=Nsb*(1-Sb);//speed in rpm at 4% slip//
printf('\nspeed at 4 percent slip=Nb=%frpm',Nb);
Va=325;//operating voltage of induction motor in volts//
Vb=Va*f1b/f1a;//stator voltage to the motor in volts//
printf('\nvalue of stator voltage to the motor=Vb=%fvolts',Vb);
Pag=150;//power transferred from stator to rotor at 30 hz in KVA//
Ws=2*3.14*Nsb/60;
T=Pag*1000/Ws;//torque if stator drop is negligible
```

```
    in watts//
printf('\ntorque if stator drop is negligible=T=
    %fwatts',T);
P2=Sb*Pag;//rotor copper loss in KVA//
printf('\nrotor copper loss=P2=%fKVA',P2);
```

Scilab code Exa 15.8 Stator frequency

```
//The ac Motor Control//
//Example 15.8//
f1a=50;//intial input frequency in hertz//
Ta=2000;//developing torque in induction motor in
    watts//
Tb=1500;//new value of torque reduced to in watts//
f1b=f1a*sqrt(Ta/Tb);//value of stator frequency
    increased to in hertz//
printf('value of stator frequency increased to f1b=
    %fhertz',f1b);
```

Scilab code Exa 15.9 Distortion Factor

```
//The ac Motor Control//
//Example 15.9//
Vom1=sqrt(2)*41.5;//starting rms value of output
    voltage //
Vom2=sqrt(2)*166;//ending rms value of output
    voltage//
V=415;//operating voltage of cyclo converter//
A1=(acos(Vom1/(1.35*V)))*180/%pi;//firing angle
    starts from//
printf('firing angle starts from A1=%fdegrees',A1);
A2=(acos(Vom2/(1.35*V)))*180/%pi;//firing angle ends
     at//
```

```
printf('\nfiring angle ends at A2=%fdegrees',A2);
PF1=0.8;//load power factor//
IPF=cos(%pi*7/15)*PF1/sqrt(2);//input power factor//
DF=0.7;//input displacement factor//
printf('\ninput power factor=IPF=%f',IPF);
Mh=cos(%pi*0.3627)*PF1/(sqrt(2)*DF);
printf('\ndistortion factor=Mh=%f',Mh);
```

Scilab code Exa 15.10 Range of variation

```
//The ac Motor Control//
//Example 15.10//
Vo5m=sqrt(2)*41.5;//rms value of output voltage //
V=415;//operating voltage of cyclo converter//
A5=(acos(Vo5m/(1.35*V)))*180/%pi;//trigger angle ranges from//
printf('trigger angle ranges fromA5=%fdegrees',A5);
A51=180-A5;//trigger angle ranges upto//
printf('\ntrigger angle ranges upto A51=%fdegrees',A51');
LPF=0.9;//load power factor//
CA15=0.3132;//maximum cosine value corresponding to operating frequency 15hz//
HIPF=CA15*LPF/sqrt(2);//highest value of input power factor//
printf('\nhighest value of input power factor=HIPF=%f',HIPF);
LIPF=cos(A5*%pi/180)*LPF/sqrt(2);//lowest value of input power factor//
printf('\nlowest value of input power factor=LIPF=%f',LIPF);
IDF=0.75;//input displacement factor//
HDF=CA15*LPF/(sqrt(2)*IDF);//highest value of distortion factor//
printf('\nhighest value of distortion factor=HDF=%f'
```

```
    ,HDF);
LDF=HDF*cos(A5*%pi/180)/CA15;//lowest value of
    distortion factor//
printf('\nlowest value of distortion factor=LDF=%f',
    LDF);
```

Scilab code Exa 15.11 Load powerfactor

```
//The ac Motor Control//
//Example 15.11//
PFm=0.5;//highest value of input factor//
Am=3.14/6;//highest value of input powerfactor
    occurs at 30 degrees//
A=cos(Am);//highest value of cosAm if firingangle
    ranging from 30 to 150//
printf('highest value of cosAm=%f',A);
PFl=(sqrt(2)*PFm)/A;
printf('\nlaod power factor of cyclo converter=%f',
    PFl);
```

Scilab code Exa 15.12 Input displacement factor

```
//The ac Motor Control//
//Example 15.12//
PFi=0.6;//input powerfactor//
DF=0.7;//distortion factor//
IDF=PFi/DF;//input displacement factor//
printf('input displacement factor=%f',IDF);
```

Scilab code Exa 15.13 Firing angle

```
//The ac Motor Control//
//Example 15.13//
PFi=0.1;//input powerfactor//
PFl=0.9;//load powerfactor//
A=(acos(sqrt(2)*PFi/PFl))*180/3.14;//firing angle
   indegrees//
printf('firing angle of cyclo converter drive=A=
   %fdegrees',A);
IDF=0.7;//leading input displacement factor//
DF=PFi/IDF;//distortion factor//
printf('\ndistortion factor=DF=%f',DF);
```

Scilab code Exa 15.14 Firing angle

```
//The ac Motor Control//
//Example 15.14//
Ap=30;//triggering angle of positive group in
   degrees//
An=180-Ap;//triggering angle of negative group in
   degrees//
printf('triggering angle of negative group=An=
   %fdegrees',An);
```

Scilab code Exa 15.15 Input currents

```
//The ac Motor Control//
//Example 15.15//
V=415;//input operating voltage of cycloconverter in
    volts//
Pi=50;//input power of the cycloconverter in KVA//
PF=0.8;//input power factor//
A=0.785;//firing angle in radians//
```

```
I=(Pi*1000*sqrt(2))/(3*V*PF*cos(A));//input current
    to the converter in amp//
printf('input current to the converter=I=%famp',I);
```

Scilab code Exa 15.16 Load powerfactor

```
//The ac Motor Control//
//Example 15.15//
Vo=200;//input operating voltage of cycloconverter
    in volts//
Po=50*10^3;//input power of the cycloconverter in VA
    //
Io=100;//drawing current from motor in amp//
PF=Po/(3*Vo*Io);//load power factor//
printf('load power factor of motor=PF=%f',PF);
```

Chapter 16

Faults and Protection

Scilab code Exa 16.1 Peak inverse voltage

```
//Faults and Protection//
//Example 16.1//
V=415;//AC input voltage//
Vf=2.53;//voltage safety factor//
PIV=2*sqrt(2)*V*Vf;//peak inverse voltage of the device//
printf('peak inverse voltage of the device=PIV=%fVolts',PIV);
```

Scilab code Exa 16.2 Voltage safety factor

```
//Faults and Protection//
//Example 16.2//
V=415;//AC input voltage in volts//
PIV=1350;//peak inverse voltage of the device in volts//
Vf=PIV/(sqrt(2)*V);//voltage safety factor of the device//
```

```
printf('voltage safety factor of the device=Vf=%f', Vf);
```

Scilab code Exa 16.3 Choke power

```
//Faults and Protection//
//Example 16.3//
P=100;//input power in KVA//
Xt=0.04;//limiting ac reactance value//
Fov=2;//current ovarload factor//
Pc=Xt*P*Fov;//choke power of the converter in KVA//
printf('choke power of the converter=Pc=%fKVA',Pc);
```

Scilab code Exa 16.4 Snubber circuit

```
//Faults and Protection//
//Example 16.4//
Ls=0.1;//stray inductance in the circuit in milli Henry//
L=2*Ls;//inductance required for the snubber ckt for protection in mH//
Im=250;//mean value of current in amp//
C=2.5*Im;//capacitance required for the snubber ckt in nano Farads//
printf('capacitance in snubber circuit=C=%fnanofarads',C);
R=2*100*sqrt(L/C);//resistance in snubber circuit in Kilo Ohms//
printf('\nResistance in snubber circuit=R=%fKilo Ohms',R);
Pdif=1*30;//permissible dv/dt of the circuit//
printf('\nPermissible dv/dt of the circuit=%fMV/s',Pdif);
```

Scilab code Exa 16.5 Suitable circuit

```
//Faults and Protection//
//Example 16.5//
V=240;//dc input voltage in volts//
Vh=25;//each selenium plate handling voltage in volts//
N=V/Vh;//number of plates in series in the circuit//
printf('number of plates in series in the circuit=N=%f',N);
printf('\nso we will use 10 plates in the circuit');
```

Scilab code Exa 16.6 Suitable circuit

```
//Faults and Protection//
//Example 16.6//
V=230;//ac input voltage in volts//
Vh=30;//each selenium plate handling voltage in volts//
N=((V/Vh)1)+1;//number of plates in series in each direction in the ckt//
printf('number of plates in series in each direction=N=%f',N);
Nt=2*N;//total number of plates in series in the circuit//
printf('\ntotal number of plates in series in both directions=Nt=%f',Nt);
```

Scilab code Exa 16.7 Energy dissipated per plate

```
//Faults and Protection//
//Example 16.7//
V=415;//ac input voltage in volts//
Vdc=440;//supplied voltage to dc motor in volts//
Vh=30;//each selenium plate handling voltage in volts//
N=Vdc/Vh;//number of plates in series in each direction in the ckt//
N=15;
printf('number of plates in each branch=N=%f',N);
Nt=3*N;//total number of plates in series in the circuit//
printf('\ntotal number of plates=Nt=%f',Nt);
Ipa=136;//peak armature current in amperes//
T=30;//time constant in milliseconds//
R=0.175;//Armature resistance in Ohms//
L=T*R;//Armature circuit Inductance in milliHenry//
printf('\nArmature circuit inductance=L=%fmH',L);
Es=0.5*L*Ipa^2*10^-3;//Energy stored in armature circuit in wattsec//
printf('\nEnergy stored in armature circuit=Es=%fwattsec',Es);
Ed=Es/N;//Energy dissipated per plate in wattsec//
printf('\nEnergy dissipated per plate=Ed=%fwattsec',Ed);
```

Scilab code Exa 16.8 Protection circuit

```
//Faults and Protection//
//Example 16.8//
printf(' As the thyristor converter is required for both rectification/inversion and as the fuse has to protect against inverter fault also,fuses will have to be located in branches and minimum of six fuses will be required.');
```

```
Id=765;//dc current in amps//
Ib=0.58*Id;//Current through each branch in amps//
printf('\n Current through each thyristor branch=Ib=%famps',Ib);
printf('\n Inverter short through:');
printf('\n Voltage causing inverter shoot through current=E2+eb');
printf('\n Maximum value of the voltage causing inverter shoot through current=sqrt(2)*E2+eb');
//E2=Input voltage of the thyristor converter//
//eb=Back emf of the motor causing regeneration//
printf('\n Recovery voltage across each fuse=Ew=E2/2+eb/(2*sqrt(2))');
//eb=Edi*cos(5*%pi/6)//
//Edi=Maximum dc value of the voltage on the thyristor converter=1.35*E2 for 6 pulse connection under discussion//
//Angle of 5*%pi/6 is normally taken as the limiting value of the firing angle beyond which inverter shoot through will takes place//
printf('\n Further fuse rated voltage=En=E2+Eb/sqrt(2)=2*Ew');
E2=500;
Ew=0.914*E2;
printf('\n Ew=%fvolts',Ew);
En=2*Ew;
printf('\n En=%fvolts',En);
printf('\n Ew/En=455/1000=0.45');
printf('\n Ita^2=1.4*Itm^2\n Total It^2 value of fuse=Ita^2+Itm^2=2.4*Itm^2=2.4*65000A^2s=1,56,000A^2s');
printf('\n I^2t of thyristor=1,90,000A^2s');
printf('\n I^2t of thyristor>I^2t of fuse or the fuse will protect the device');
printf('\n Short circuit on dc Bushers');
//The fault is shown in fig 16.9(c) along with path of the fault current//
printf('\n Maximum voltage causing fault current=
```

```
    sqrt(2)*E2');
printf('\n Recovery voltage across each fuse=0.5*E2
    =0.5*500=250volts');
printf('\n Ew/En=250/1000=0.25 and at this value Ita
    ^2=0.4*Itm^2');
printf('\n It^2 of fuse=Ita^2+Itm^2=1.4*Itm
    ^2=1.4*65000=91000A^2s');
printf('\n It^2 of thyristor=1,50,000A^2s\n It^2 of
    thyristor>>It^2 of fuse\n the fuse will protect
    thyristor');
printf('\n Puncture of a device:\n In this case also
     maximum voltage causing fault current=sqrt(2)*E2
    \n Thus as per case It^2 value of thyristor will
    be more than that of fuse');
printf('\n Short circuit between phase and bridge:\n
     In this case also as per case above fuse will
    protect device\n Thus the fuse will protect for
    all faults');
```

Scilab code Exa 16.9 Additional value of inductance

```
//Faults and Protection//
//Example 16.9//
printf('    Thus from the table we see at a value of
     circuit inductance 1.592mH,I^2t value of breaker
     is 4.9*10^5A^2s and selectivity between fuse and
     breakeris I^2tFuse/I^2t Breaker
    =4*10^5/4.97*10^5=1.01');
printf('\nAs this is just the border case we will go
     for the next value of inductancei.e,1.91mH.where
     selectivity=5*10^5/4.34*10^5=1.18');
printf('\nThus the additional inductance required is
    =1.91-1.273=0.637mH');
```