

Scilab Textbook Companion for  
Thyristors Theory And Applications  
by R. K. Sugandhi And K. K. Sugandhi<sup>1</sup>

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

May 20, 2016

<sup>1</sup>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	4
2 THE DEVICE	5
3 Fabrication and Thermal characteristics	8
4 Series and Parallel Connection of Thyristors	11
5 Line Commutated converters	13
7 Inverter Circuits	18
8 Harmonic and PowerFactor with the converter system	20
11 Control of DC Motors	28
12 Controllers and Their Optimisation	35
13 Choppers and Transportation system Application	41
15 The AC motor control	46
16 Faults and Protection	55

# List of Scilab Codes

Exa 2.1	voltage safety factor . . . . .	5
Exa 2.2	peak inverse voltage . . . . .	5
Exa 2.3	capacitive current . . . . .	6
Exa 2.4	equivalent capacitance . . . . .	6
Exa 2.5	value of derivative of v . . . . .	6
Exa 2.6	value of rate of voltage . . . . .	6
Exa 2.7	Voltage safety factor . . . . .	7
Exa 3.1	junction temperature . . . . .	8
Exa 3.2	Maximum junction temperature . . . . .	8
Exa 3.3	value of junction temperature . . . . .	9
Exa 3.4	on state power loss . . . . .	10
Exa 3.5	case sink thermal resistance . . . . .	10
Exa 4.1	Derating factor . . . . .	11
Exa 4.2	series connection . . . . .	11
Exa 5.1	AC terminal power . . . . .	13
Exa 5.2	Voltage regulation . . . . .	13
Exa 5.3	Maximum dc voltage . . . . .	14
Exa 5.4	Firing angle . . . . .	15
Exa 5.5	Reactance of the reactor . . . . .	15
Exa 5.6	AC load current . . . . .	16
Exa 5.7	Average value of voltage . . . . .	16
Exa 5.8	DC output voltage . . . . .	17
Exa 7.1	Attenuation factor . . . . .	18
Exa 7.2	Value of inductance . . . . .	19
Exa 7.3	Value of R1 . . . . .	19
Exa 8.1	Shunt filter . . . . .	20
Exa 8.2	DC reactor circuit . . . . .	21
Exa 8.3	Commutation angle . . . . .	21

Exa 8.4	Rating of shunt compensator . . . . .	23
Exa 8.5	Shunt filter . . . . .	23
Exa 8.6	Maximum current ripple . . . . .	24
Exa 8.7	Voltage ripple . . . . .	25
Exa 8.8	Triggering angle . . . . .	26
Exa 8.9	Power factor . . . . .	26
Exa 11.1	Designing a thyristor . . . . .	28
Exa 11.2	Blocking angle . . . . .	29
Exa 11.3	Firing angle . . . . .	30
Exa 11.4	Reactive power . . . . .	30
Exa 11.5	Active and Reactive power . . . . .	32
Exa 11.6	Power at given load . . . . .	33
Exa 11.7	Triggering angle . . . . .	33
Exa 12.1	Permanent error of p controller . . . . .	35
Exa 12.2	Motor armature time constant . . . . .	35
Exa 12.3	Controller parameters . . . . .	36
Exa 12.4	Designing a PI regulator . . . . .	36
Exa 12.5	Time constant of the controller . . . . .	37
Exa 12.6	Maximum overshoot . . . . .	38
Exa 12.7	Settling time . . . . .	38
Exa 12.8	Difference in response . . . . .	39
Exa 13.1	Instantaneous current . . . . .	41
Exa 13.2	Conduction and Blocking period . . . . .	42
Exa 13.3	Optimum frequency . . . . .	42
Exa 13.4	Required pulse width . . . . .	43
Exa 13.5	Pulse width . . . . .	43
Exa 13.6	Motor Torque . . . . .	44
Exa 13.7	Chopper Frequency . . . . .	44
Exa 15.1	Stator current . . . . .	46
Exa 15.2	Designing a thyristor converter . . . . .	46
Exa 15.3	Torque developed by the motor . . . . .	47
Exa 15.4	Torque developed by the motor . . . . .	47
Exa 15.5	Rotor Frequency . . . . .	48
Exa 15.6	Input voltage to the motor . . . . .	48
Exa 15.7	Rotor copper loss . . . . .	49
Exa 15.8	Stator frequency . . . . .	50
Exa 15.9	Distortion Factor . . . . .	50
Exa 15.10	Range of variation . . . . .	51

Exa 15.11	Load powerfactor . . . . .	52
Exa 15.12	Input displacement factor . . . . .	52
Exa 15.13	Firing angle . . . . .	52
Exa 15.14	Firing angle . . . . .	53
Exa 15.15	Input currents . . . . .	53
Exa 15.16	Load powerfactor . . . . .	54
Exa 16.1	Peak inverse voltage . . . . .	55
Exa 16.2	Voltage safety factor . . . . .	55
Exa 16.3	Choke power . . . . .	56
Exa 16.4	Snubber circuit . . . . .	56
Exa 16.5	Suitable circuit . . . . .	57
Exa 16.6	Suitable circuit . . . . .	57
Exa 16.7	Energy dissipated per plate . . . . .	57
Exa 16.8	Protection circuit . . . . .	58
Exa 16.9	Additional value of inductance . . . . .	60

# Chapter 2

## THE DEVICE

Scilab code Exa 2.1 vlotage safety factor

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

---

Scilab code Exa 2.2 peak inverse vlotage

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

---



**Scilab code Exa 2.3** capacitive current

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

---

**Scilab code Exa 2.4** equivalent capacitance

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

---

**Scilab code Exa 2.5** value of derivative of v

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

---

**Scilab code Exa 2.6** value of rate of voltage

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

---

**Scilab code Exa 2.7** Voltage safety factor

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

---

# Chapter 3

## Fabrication and Thermal characteristics

Scilab code Exa 3.1 junction temperature

```
1 //Fabrication and Thermal characteristics
2 //Example 3.1
3 Xa=50; //Ambient temperature//
4 P=150; //on state power loss in Watts//
5 Rjc=0.02; //junction-case thermal resistance//
6 Rcs=0.05; //case-sink thermal resistance//
7 Rsa=0.08; //sink-atmosphere thermal resistance//
8 Xj=Xa+P*(Rjc+Rcs+Rsa); //junction temperature//
9 printf('value of junction temperature=%fc',Xj);
```

---

Scilab code Exa 3.2 Maximum junction temperature

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

```

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

```

---

**Scilab code Exa 3.3** value of junction temperature

```

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

```

---

**Scilab code Exa 3.4** on state power loss

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

---

**Scilab code Exa 3.5** case sink thermal resistance

```
1 //Fabrication and Thermal characteristics
2 //Example 3.5
3 Xa=40; //Ambient temperature//
4 P=300; //on state power loss in Watts//
5 Rjc=0.015; //junction_case thermal resistance//
6 Rsa=0.1; //sink_atmosphere thermal resistance//
7 Xj=105; //junction temperature//
8 Rcs=((Xj-Xa)/(P))-(Rjc+Rsa); //case_sink thermal
   resistance//
9 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

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

---

Scilab code Exa 4.2 series connection

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

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

---

# Chapter 5

## Line Commutated converters

Scilab code Exa 5.1 AC terminal power

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

---

Scilab code Exa 5.2 Voltage regulation

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



```

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

```

---

### Scilab code Exa 5.3 Maximum dc voltage

```

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

```

---

#### Scilab code Exa 5.4 Firing angle

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

---

#### Scilab code Exa 5.5 Reactance of the reactor

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

```

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

```

---

#### Scilab code Exa 5.6 AC load current

```

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

```

---

#### Scilab code Exa 5.7 Average value of voltage

```

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

```

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

---

**Scilab code Exa 5.8** DC output voltage

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

---

# Chapter 7

## Inverter Circuits

Scilab code Exa 7.1 Attenuation factor

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

---

**Scilab code Exa 7.2** Value of inductance

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

---

**Scilab code Exa 7.3** Value of R1

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

---

# Chapter 8

## Harmonic and PowerFactor with the converter system

Scilab code Exa 8.1 Shunt filter

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

```

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

```

---

### Scilab code Exa 8.2 DC reactor circuit

```

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

```

---

### Scilab code Exa 8.3 Commutation angle



```

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

```

of the program//

---

#### Scilab code Exa 8.4 Rating of shunt compensator

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

---

#### Scilab code Exa 8.5 Shunt filter

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

```

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

```

---

#### Scilab code Exa 8.6 Maximum current ripple

```

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

```

```

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

```

---

### Scilab code Exa 8.7 Voltage ripple

```

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

```

---

**Scilab code Exa 8.8** Triggering angle

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

---

**Scilab code Exa 8.9** Power factor

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

```
13 RP=sqrt(3)*E2*I2*sqrt(1-PF^2)/1000;//reactive power
    to be supplied by shunt compensator in KVAR//
14 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

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

```

    armature control//
12 //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.//
13 printf('\nAC rating of field converter=230/1.35=170V
    ');
14 //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
    //
15 printf('\nDC power=230*10=2300Watts ');
16 printf('\nAC power=1.05*2300=2415Watts ');
17 printf('\nThus tranformer of 2.5KVA,415/230V will be
    required ');
18 Edca=(170+170/10)*1.35;//available voltage in volts
    //
19 Edc=1.35*230;
20 A=acos(Edca/Edc)*180/%pi;
21 printf('\nField converter shall be locked at an
    angle of A=%fdegrees ',A);

```

---

### Scilab code Exa 11.2 Blocking angle

```

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

```



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

---

### Scilab code Exa 11.3 Firing angle

```
1 //Control of DC motors//
2 //Example 11.3//
3 Edca1=230;
4 N1=1000;
5 N2=500;
6 Eb1=210;
7 printf('Eb1=230-20=210 volts');
8 Eb2=Eb1*N2/N1;
9 printf('\nEb2=%fvolts',Eb2);
10 V=40;//motor armature drop at rated load in volts//
11 Edca2=Eb2+V;
12 printf('\nEdca2=%fvolts',Edca2);
13 C1=1;//cosine of the firing angle corresponding to
    1000 rpm load//
14 C2=C1*Edca2/Edca1;//cosine of the firing angle
    corresponding to 500 rpm load//
15 printf('\nC cosine of the firing angle corresponding
    to 500 rpm load=C2=%f',C2);
16 A=acos(C2)*180/%pi;//firing angle corresponding to
    500 rpm load in degrees//
17 printf('\nfiring angle corresponding to 500 rpm load
    A=%fdegrees',A);
```

---

### Scilab code Exa 11.4 Reactive power

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

```

3 Edca1=1.15*440; //Rated output voltage from the
  converter for rated speed of750rpm//
4 printf('Rated output voltage from the converter=
  Edca1=%fvolts ',Edca1);
5 N1=750;
6 N2=500;
7 Edca2=Edca1*N2/N1;
8 E2=415;
9 C2=Edca2/(1.35*E2);
10 printf('\nCosine of the triggering angle=C2=%f',C2);
11 A2=C2*180/%pi;
12 printf('\nTriggering angle=A2=%fdegrees ',A2);
13 PF2=C2*(1+cos(15*%pi/180))/2;
14 printf('\nPowerfactor=PF2=%f',PF2);
15 Id=200; //dc current in amps//
16 I2=0.75*0.817*Id; //Current at 75percent load in amps
  //
17 RP2=sqrt(3)*E2*I2*sqrt(1-PF2^2)/1000; //Reactive
  power drawn at 75% load//
18 printf('\nReactive power at 75percent load=RP2=
  %fKVAR ',RP2);
19 h=6;
20 Wv=24.17; //maximum voltage ripple in percent//
21 Wi=8; //maximum permissible current ripple in percent
  //
22 I6=Wi*Id/100;
23 printf('\nSixth harmonic ripple current=I6=%fAmps',
  I6);
24 W=314;
25 L=(Wv*Edca1*10)/(I6*h*W);
26 printf('\nInductance required in dc circuit=L=%fmH',
  L);
27 C1=Edca1/(1.35*E2);
28 printf('\nCosine of the triggering angle=C1=%f',C1);
29 A1=C1*180/%pi;
30 printf('\nTriggering angle=A1=%fdegrees ',A1);
31 PF1=C1*(1+cos(15*%pi/180))/2;
32 printf('\nPowerfactor=PF1=%f',PF1);

```

```

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

```

---

### Scilab code Exa 11.5 Active and Reactive power

```

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

```

```

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

```

---

#### Scilab code Exa 11.6 Power at given load

```

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

```

---

#### Scilab code Exa 11.7 Triggering angle

```

1 //Control of DC motors//
2 //Example 11.7//
3 N1=500;
4 N2=400;
5 Eb1=410;
6 Eb2=Eb1*N2/N1;
7 printf('Eb2=%fvolts',Eb2);
8 V=440;//operating voltage of dc motor in volts//
9 P=100;//input power of dc motor in KW//
10 Ia=P*1000/V;
11 printf('\nIa=%fAmps',Ia);
12 Ra=(V-Eb1)/Ia;

```

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

---

# Chapter 12

## Controllers and Their Optimisation

Scilab code Exa 12.1 Permanent error of p controller

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

---

Scilab code Exa 12.2 Motor armature time constant

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

```

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

```

---

### Scilab code Exa 12.3 Controller parameters

```

1 //Controllers and Their Optimisation//
2 //Example 12.3//
3 f=50; //frequency in hz//
4 p=6; //pulse number//
5 t1=1000/(2*f*p); //time constant for the current loop
    in ms//
6 printf('time constant for the current loop=t1=%fms',
    t1);
7 t2=1.5; //time constant of feedback channel in ms//
8 G=t1+t2; //smaller time constant in ms//
9 printf('\nSmaller time constant=G=%fms',G);
10 T1=30; //bigger time constant in ms//
11 Tn=T1; //time constant of the controller in ms//
12 printf('\nTime constant of the controller in AVO=Tn=
    %fms',Tn);
13 V=T1/(2*G); //gain of the control system//
14 printf('\nGain of the control system=V=%f',V);
15 Vg=14; //gain of the regulating current link//
16 Vr=V/Vg; //gain of the PI controller//
17 printf('\nGain of the PI controller=Vr=%f',Vr);
18 R2=11; //R2 in KiloOhms//
19 R1=R2/Vr; //R1 in kiloohms//
20 printf('\nR1=%fKiloohms',R1);
21 C1=Tn/R1; //C1 in microfarads//
22 printf('\nC1=%fmicrofarads',C1);

```

---

### Scilab code Exa 12.4 Designing a PI regulator

```

1 //Controllers and Their Optimisation//
2 //Example 12.4//
3 G=20;//smaller time constant in ms//
4 T1=350;//bigger time constant in ms//
5 Tn=4*G;//time constant of the controller in ms//
6 printf('\nTime constant of the controller in SO=Tn=
    %fms ',Tn);
7 V=T1/(2*G);//gain of the control system//
8 printf('\nGain of the control system=V=%f',V);
9 Vg=1;//gain of the regulating current link//
10 Vr=V/Vg;//gain of the PI regulator//
11 printf('\nGain of the PI regulator=Vr=%f',Vr);
12 R1=11;//R1 in KiloOhms//
13 R2=R1*Vr;//R2 in kilohms//
14 printf('\nR2=%fKilohms ',R2);
15 C2=Tn/R2;//C1 in microfarads//
16 printf('\nC2=%fmicrofarads ',C2);

```

---

#### Scilab code Exa 12.5 Time constant of the controller

```

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

```



```

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

```

---

#### Scilab code Exa 12.6 Maximum overshoot

```

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

```

---

#### Scilab code Exa 12.7 Settling time

```

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

```

```

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

```

---

#### Scilab code Exa 12.8 Difference in response

```

1 //Controllers and Their Optimisation//
2 //Example 12.8//
3 printf('Response for an AVO system');
4 G=10; //smaller time constant in ms//
5 Tf=4.7*G; //time taken by the system to achieve its
   final output for firsttime//
6 printf('\ntime taken by the system to achieve its
   final value=Tf=%fms',Tf);
7 printf('\nMaximum overshoot for a symmetrically
   optimised system is 4.3 percent');
8 Ts=8.4*G; //settling time in ms//
9 printf('\nSettling time=Ts=%fms',Ts);
10 printf('\nResponse for an SO system');
11 G=10; //smaller time constant in ms//
12 Tf=3.1*G; //time taken by the system to achieve its
   final output for firsttime//
13 printf('\ntime taken by the system to achieve its
   final value=Tf=%fms',Tf);
14 printf('\nMaximum overshoot for a symmetrically
   optimised system is 43 percent');

```

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

---

# Chapter 13

## Choppers and Transportation system Application

Scilab code Exa 13.1 Instantaneous current

```
1 //Choppers and Transportation System Application//
2 //Example 13.1//
3 E=220;//dc supply voltage in volts//
4 E1=22;//Load voltage in volts//
5 Ton=1000;//conducting period in microseconds//
6 T=2500;//Total timeperiod in microseconds//
7 L=1;//inductance in milliHenry//
8 R=0.25;//resistance in ohms//
9 t=L/R;//time constant in milliseconds//
10 printf('time constant=t=%fmilliseconds',t);
11 A=0.133;
12 Td=A*T;//Discontinuous condition starts at//
13 printf('\nDiscontinuous condition starts from Td=
    %fmicroseconds',Td);
14 Eo=0.4*E;//output voltage in volts//
15 printf('\nOutput voltage=Eo=%fvolts',Eo);
16 Iav=(Eo-E1)/R;//Average current in amps//
17 printf('\nAverage current=Iav=%famp',Iav);
18 Imax=((E*(1-exp(-Ton/(t*1000)))))/(R*(1-exp(-T/(t
```

```

    *1000)))))-(E1/R);
19 printf('\nMaximum current=Imax=%famp', Imax);
20 Imin=(E*(exp(Ton/(t*1000))-1))/(R*(exp(T/(t*1000))-1)))-(E1/R);
21 printf('\nMinimum current=Imin=%famp', Imin);

```

---

### Scilab code Exa 13.2 Conduction and Blocking period

```

1 //Choppers and Transportation System Application//
2 //Example 13.2//
3 f=1; //operating frequency in KHZ//
4 E=220; //dc supply voltage in volts//
5 E1=165; //Load voltage in volts//
6 Ton=E1/(E*f); //conduction period in ms//
7 printf('Conduction period=Ton=%fms', Ton);
8 T=1/f; //total time period in ms//
9 printf('\nTotal time period=T=%fms', T);
10 Toff=T-Ton; //blocking period in ms//
11 printf('\nBlocking period=Toff=%fms', Toff);

```

---

### Scilab code Exa 13.3 Optimum frequency

```

1 //Choppers and Transportation System Application//
2 //Example 13.3//
3 E=220; //dc supply voltage in volts//
4 Toff=200; //blocking period in microseconds//
5 I1=50; //load current in amps//
6 C=%pi*Toff*I1/(2*E); //capacitance for optimum
   frequency in microfarad//
7 C=75;
8 printf('Load capacitance required for optimum
   frequency=C=%fmicrofarad', C);

```

```

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

```

---

#### Scilab code Exa 13.4 Required pulse width

```

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

```

---

#### Scilab code Exa 13.5 Pulse width

```

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

```

---

### Scilab code Exa 13.6 Motor Torque

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

---

### Scilab code Exa 13.7 Chopper Frequency

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

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

---



# Chapter 15

## The AC motor control

### Scilab code Exa 15.1 Stator current

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

---

### Scilab code Exa 15.2 Designing a thyristor converter

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

```

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

```

---

**Scilab code Exa 15.3** Torque developed by the motor

```

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

```

---

**Scilab code Exa 15.4** Torque developed by the motor

```

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

```

---

#### Scilab code Exa 15.5 Rotor Frequency

```

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

```

---

#### Scilab code Exa 15.6 Input voltage to the motor

```

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

```

```

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

```

---

#### Scilab code Exa 15.7 Rotor copper loss

```

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

```

```

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

```

---

### Scilab code Exa 15.8 Stator frequency

```

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

```

---

### Scilab code Exa 15.9 Distortion Factor

```

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

```

```

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

```

---

### Scilab code Exa 15.10 Range of variation

```

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

```

```

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

```

---

#### Scilab code Exa 15.11 Load powerfactor

```

1 //The ac Motor Control//
2 //Example 15.11//
3 PFm=0.5;//highest value of input factor//
4 Am=3.14/6;//highest value of input powerfactor
    occurs at 30 degrees//
5 A=cos(Am);//highest value of cosAm if firingangle
    ranging from 30 to 150//
6 printf('highest value of cosAm=%f',A);
7 PF1=(sqrt(2)*PFm)/A;
8 printf('\nload power factor of cyclo converter=%f',
    PF1);

```

---

#### Scilab code Exa 15.12 Input displacement factor

```

1 //The ac Motor Control//
2 //Example 15.12//
3 PFi=0.6;//input powerfactor//
4 DF=0.7;//distortion factor//
5 IDF=PFi/DF;//input displacement factor//
6 printf('input displacement factor=%f',IDF);

```

---

#### Scilab code Exa 15.13 Firing angle

```

1 //The ac Motor Control//
2 //Example 15.13//
3 PFi=0.1; //input powerfactor//
4 PF1=0.9; //load powerfactor//
5 A=(acos(sqrt(2)*PFi/PF1))*180/3.14; //firing angle
   indegrees//
6 printf('firing angle of cyclo converter drive=A=
   %fdegrees ',A);
7 IDF=0.7; //leading input displacement factor//
8 DF=PFi/IDF; //distortion factor//
9 printf('\ndistortion factor=DF=%f',DF);

```

---

#### Scilab code Exa 15.14 Firing angle

```

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

```

---

#### Scilab code Exa 15.15 Input currents

```

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

```



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

---

#### Scilab code Exa 15.16 Load powerfactor

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

---

# Chapter 16

## Faults and Protection

Scilab code Exa 16.1 Peak inverse voltage

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

---

Scilab code Exa 16.2 Voltage safety factor

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

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

---

### Scilab code Exa 16.3 Choke power

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

---

### Scilab code Exa 16.4 Snubber circuit

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

---

**Scilab code Exa 16.5** Suitable circuit

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

---

**Scilab code Exa 16.6** Suitable circuit

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

---

**Scilab code Exa 16.7** Energy dissipated per plate

```

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

```

---

#### Scilab code Exa 16.8 Protection circuit

```

1 //Faults and Protection//
2 //Example 16.8//
3 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. ');

```

```

4 Id=765;//dc current in amps//
5 Ib=0.58*Id;//Current through each branch in amps//
6 printf('\n Current through each thyristor branch=Ib=
    %famps',Ib);
7 printf('\n Inverter short through:');
8 printf('\n Voltage causing inverter shoot through
    current=E2+eb');
9 printf('\n Maximum value of the voltage causing
    inverter shoot through current=sqrt(2)*E2+eb');
10 //E2=Input voltage of the thyristor converter//
11 //eb=Back emf of the motor causing regeneration//
12 printf('\n Recovery voltage across each fuse=Ew=E2
    /2+eb/(2*sqrt(2))');
13 //eb=Edi*cos(5*%pi/6)//
14 //Edi=Maximum dc value of the voltage on the
    thyristor converter=1.35*E2 for 6 pulse
    connection under discussion//
15 //Angle of 5*%pi/6 is normally taken as the limiting
    value of the firing angle beyond which inverter
    shoot through will takes place//
16 printf('\n Further fuse rated voltage=En=E2+Eb/sqrt
    (2)=2*Ew');
17 E2=500;
18 Ew=0.914*E2;
19 printf('\n Ew=%fvolts',Ew);
20 En=2*Ew;
21 printf('\n En=%fvolts',En);
22 printf('\n Ew/En=455/1000=0.45');
23 printf('\n Ita2=1.4*Itm2\n Total It2 value of
    fuse=Ita2+Itm2=2.4*Itm2=2.4*65000A2s=1,56,000
    A2s');
24 printf('\n I2t of thyristor=1,90,000A2s');
25 printf('\n I2t of thyristor>I2t of fuse or the
    fuse will protect the device');
26 printf('\n Short circuit on dc Bushers');
27 //The fault is shown in fig 16.9(c) along with path
    of the fault current//
28 printf('\n Maximum voltage causing fault current=

```

```

    sqrt(2)*E2');
29 printf('\n Recovery voltage across each fuse=0.5*E2
    =0.5*500=250 volts ');
30 printf('\n Ew/En=250/1000=0.25 and at this value Ita
    ^2=0.4*Itm^2');
31 printf('\n It^2 of fuse=Ita^2+Itm^2=1.4*Itm
    ^2=1.4*65000=91000A^2s');
32 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');
33 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');
34 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

```

1 //Faults and Protection//
2 //Example 16.9//
3 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 ');
4 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 ');
5 printf('\nThus the additional inductance required is
    =1.91-1.273=0.637mH');

```

---