Scilab Textbook Companion for
Power Electronics
by M. D. Singh and K. B. Khanchandani[1]

Created by
Mohit Garg
B.Tech
Electronics Engineering
ABES Engineering College, Ghaziabad
College Teacher
None
Cross-Checked by
Spandana

July 11, 2017

---

[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: Power Electronics

Author: M. D. Singh and K. B. Khanchandani

Publisher: Tata McGraw-Hill

Edition: 2

Year: 2006

ISBN: 978-0070583894

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

Thyristors Principles and Characteristics

Scilab code Exa 2.1 Will the thyristor get fired

```
// chapter 2
// example 2.1
// fig. E2.1
// Will the thyristor get fired
// page-21-22
clear;
clc;
// given
I_L=50; // in mA (Latching current)
t=50; // in us (duration of fireing pulse)
R=20; // in ohm (resistance of ciicuit)
L=0.5; // in H (Inductance of circuit)
V=100; // in V (Battery voltage)
// calculate
tou=L/R; // calculation of total time period
t=t*1E-6; // changing unit from us to sec
i=(V/R)*(1-exp(-t/tou)); // calculation of current
   at t=50 us
i=i*1E3; // changing unit from A to mA
```

```
printf("\nThe value of current is \t i(50 us) = %.2f mA\n",i);
if i<I_L then
    printf("\nSCR will not get fired");

    else printf("\nSCR will get fired");
end
```

Scilab code Exa 2.2 Obtain the minimum gating pulse

```
// chapter 2
// example 2.2
// fig. E2.2
// Obtain the minimum gating pulse
// page-22
clear;
clc;
// given
i=4; // in mA (Latching current)
L=0.1; // in H (Inductance of circuit)
V=100; // in V (Battery voltage)
// calculate
i=i*1E-3; // changing unit from mA to A
// Since V=L*(di/dt), therefore we get
t_min=(L/V)*i; // calculation of minimum gating pulse
t_min=t_min*1E6; // changing unit from sec to us
printf("\nThe minimum gating pulse is \t t_min=%.f us",t_min);
```

Scilab code Exa 2.3 Compute average power loss

```
// chapter 2
```

```
// example 2.3
// fig. E2.3
// Compute average power loss
// page-22-23
clear;
clc;
// given
Beta1=180, Beta2=360; // in degrees (conduction
   angles)
Iav=100; // in A (average current)
// calculate
// since Iav=Im*Beta/360, therefore
Im1=Iav*360/Beta1; // calculation of current during
   180 conduction
V_T1= 1.8; // in V (given corresponding to value of
   Im1)
Pavg1=V_T1*Im1*(Beta1/360); // calculation of
   average power loss during 180 conduction
printf("\nThe average power loss during %.f
   conduction is %.f W",Beta1,Pavg1);
Im2=Iav*360/Beta2; // calculation of current during
   360 conduction
V_T2= 1.5; // in V (given corresponding to value of
   Im2)
Pavg2=V_T2*Im2*(Beta2/360); // calculation of
   average power loss during 360 conduction
printf("\n\nThe average power loss during %.f
   conduction is %.f W",Beta2,Pavg2);
```

Scilab code Exa 2.4 Find the value of series resistor and average power

```
// chapter 2
// example 2.4
// Find the value of series resistor and average
   power
```

```
// page-31
clear;
clc;
// given
// Vg=1.5+8*Ig; // relation between Vg and Ig
Vgs=12; // in V
t=50; // in us
D=0.2; // duty cucle
P=5; // in W
// calculate
// since Vgs=Ig*Rg+Vg=Ig*Rg+1.5+8*Ig or
// Vgs=(Rg+8)*Ig+1.5, (i)
// since P=Vg*Ig and therefore
// P=(1.5+8*Ig)*Ig ,
// P=1.5*Ig+8*Ig^2.
// This is quadratic equation and can be written as
// 8*Ig^2+1.5*Ig-P=0 solving it
Ig1=(-1.5+sqrt(1.5^2-(4*8*(-P))))/(2*8);
Ig2=(-1.5-sqrt(1.5^2-(4*8*(-P))))/(2*8);
if Ig1>0 then
    Ig=Ig1
else
    Ig=Ig2
end
Rg=((Vgs-1.5)/Ig)-8; // calculation of series resistance
Pavg=P*D; // calculation of average power
printf("\nThe series resistance is \t Rg=%.f ohm",Rg);
printf("\nThe average power is \t\t Pavg=%.f W",Pavg);
```

Scilab code Exa 2.5 Calculate the required gate resistance source

```
// chapter 2
```

```
// example 2.5
// Calculate the required gate resistance source
// page-31
clear;
clc;
// given
Pg=0.012; // in W (power) Pg=Vg*Ig
gradient=3E3; // gradient=Vg/Ig
Egs=10; // in V
// calculate
// Since gradient*Pg=Vg^2 therefore we get
Vg=sqrt(gradient*Pg); // calculation of gate voltage
Ig=Pg/Vg; // calculation of gate current
Rg=Vg/Ig; // calculation of gate resistance
Ig=Ig*1E3; // changing unit from A to mA
Rg=Rg*1E-3; // changing unit from ohm to kohm
printf("\nThe required gate voltage is \tVg=%.f V\n",Vg);
printf("\nThe required gate current is \tIg=%.f mA\n",Ig);
printf("\nThe required gate source resistance is \tRg=%.f kohm\n",Rg);
// Note: the value of gate source resistance has not been calculated in the book.
```

Scilab code Exa 2.6 Calculate the mean power loss

```
// chapter 2
// example 2.6
// Fig. E2.6
// Calculate the mean power loss
// page-31-32
clear;
clc;
// given
```

```
// V_T=1.0+(1.1*I/60)  (from graph)
I_a=23; // in A (steady current)\
I_b=18; // in A (Half-sine wave)
I_c=39.6; // in A (level current for half cycle)
I_d=48.5; // in A (level current for one-third cycle)
// calculate
V_T_a=1.0+(1.1*I_a/60); // calculation of ON state voltage corresponding to  I_a
P_a=V_T_a*I_a; // calculation of power loss corresponding to I_a
printf("\nThe Power loss is \t%.1f W\n",P_a);
I_m_b=I_b*%pi; // maximum value of sine wave current
P_b=(1/(2*%pi))*(integrate('((1+(1.1*I_m_b*sin(x)/60))*I_m_b*sin(x))','x',0,%pi)); // calculation of power loss corresponding to I_b
printf("\nThe Power loss is \t%.2f W\n",P_b);
V_T_c=1.0+(1.1*I_c/60); // calculation of ON state voltage corresponding to  I_c
P_c=V_T_c*I_c/2; // calculation of power loss corresponding to I_c
printf("\nThe Power loss is \t%.1f W\n",P_c);
V_T_d=1.0+(1.1*I_d/60); // calculation of ON state voltage corresponding to  I_d
P_d=V_T_d*I_d/3; // calculation of power loss corresponding to I_d
printf("\nThe Power loss is \t%.1f W\n",P_d);
// Note: the answer of part (b) varies slightly due to precise calculation and round off
```

Scilab code Exa 2.7 Compute the peak inverse voltage

```
// chapter 2
// example 2.7
// Compute the peak inverse voltage
```

```
// page-32
clear;
clc;
// given
Vin=415; // in V (input voltage)
Vf=2.1; // voltage safety factor
// calculate
PIV=sqrt(2)*Vin*Vf; // calculation of peak inverse voltage
printf("\nThe peak inverse voltage is \tPIV=%.2f V",PIV);
```

Scilab code Exa 2.8 Calculate the resistance gate power dissipation and maximum triggering frequency

```
// chapter 2
// example 2.8
// Fig. E2.8
// Calculate the resistance, gate power dissipation and maximum triggering frequency
// page-32-33
clear;
clc;
// given
Ig_min=500; // in mA (minimum gate current)
gradient=16; // in V/A
Egs=15; // in V (gate source voltage)
T_on=4; // in us (minimum turn on time)
Pg_av=0.3; // in W (average gate power dissipation)
// calculate
Ig_min=Ig_min*1E-3; // changing unit from mA to A
Vg=gradient*Ig_min; // calculation of gate voltage
Rs=(Egs-Vg)/Ig_min; // calculation of resistance
printf("\nThe resistance to be connected in series with SCR gate is \tRs= %.f ohm",Rs);
```

```
Pg=Vg*Ig_min; // calculation of power dissipation
printf("\n\nThe gate power dissipation is \t\t\t\t\tPg= %.f W",Pg);
// Since Pg=Pg_av/(f*T_on), therefore
T_on=T_on*1E-6; // changing unit from us to sec
f=Pg_av/(Pg*T_on); // calculation of maximum triggering frequency
f=f*1E-3; // changing unit from Hz to khz
printf("\n\nThe maximum triggering frequency is \tf= %.2f kHz \t or \tF= %.f kHz",f,f);
```

Scilab code Exa 2.9 Calculate the value of commutating capacitance

```
// chapter 2
// example 2.9
// Fig. 2.17
// Calculate the value of commutating capacitance
// page-54
clear;
clc;
// given
Edc=120; // in V (source voltage)
I=20; // in A (current)
t_off=60; // in us (turn-off time of both SCR)
// calculate
R1=Edc/I; // calculation of R1
R2=R1; // calculation of R2
C=1.44*t_off/R1; // calculation of commutating capacitance
printf("\n\nThe commutating capacitance for succcessful commutation is \tC= %.1f uF",C);
```

Scilab code Exa 2.10 Compute the value of commutating capacitance and inductance

```
// chapter 2
// example 2.10
// Fig. 2.19
// Compute the value of commutating capacitance and inductance
// page-54
clear;
clc;
// given
Edc=50; // in V (source voltage)
I_L=50; // in A (current)
t_off=30; // in us (turn-off time of SCR1)
f=500; // in Hz (chopping frequency)
neta1=10, neta2=100; // in % (load voltage variation)
// calculate
// Let  us assume 50 % tolerance on turn-off time
t_off=t_off+(50*t_off/100); // calculation of turn-off time for reliable operation
printf("\nThe turn-off time for reliable operation is \tt_off= %.f us",t_off);
C=I_L*t_off/Edc; // calculation of commutating capacitance
printf("\n\nThe commutating capacitance is \t\t\tC= %.f uF",C);
V0_min=neta1*Edc/100; // calculation of minimum V0
T=1/f; // calculation of chopping time period
L1=((V0_min/Edc)^2)*(T^2)/(((%pi)^2)*C*1E-6);
L1=L1*1E6; // changing unit from H to uH
 //L1=((V0_min/Edc)^2); // calculation of upper limit of commutating inductance
L2=C*(Edc/I_L)^2; // calculation of lower limit of commutating inductance
printf("\n\nThe commutating inductance lies in range \t %.f uH < L < %.f uH",L2,L1);
```

Scilab code Exa 2.11 Calculate shortest fault clearance time

```
// chapter 2
// example 2.11
// Fig. E2.11
// Calculate shortest fault clearance time
// page-54-55
clear;
clc;
// given
Vm=120; // in V (sinosoidal ac supply)
integration=15; // in a^2.s (integral of square of
   current)
// calculate
t=integration/Vm^2; // calculation of fault
   clearance time
t=t*1E3; // changing unit from s to ms
printf("\n\nThe fault clearance time is \t= %.2f ms"
   ,t);
```

Scilab code Exa 2.12 Calculate one cycle surge current rating

```
// chapter 2
// example 2.12
// Calculate one cycle surge current rating
// page-54-55
clear;
clc;
// given
I_sub=3000; // in A (half cycle surge current rating
   )
```

```
f=50; // in Hz (supply frequency)
T=100, t=50; // time ration for one and half cycle respectively
// calculate
// I^2*T=I_sub^2*t, therefore we get
I=sqrt(I_sub^2*(t/T)); // calculation of one cycle surge current rating
printf("\nThe one cycle surge current rating \tI= %.2f A",I);
rating=(I_sub/(5*sqrt(2)))^2*(1/(2*f)); // calculation of i^2*t rating
printf("\n\nThe i^2*t  rating is \t %.f A^2.s",rating);
// Note: The answer in the book for i^2*t  rating is wrong due to calculation mistake
```

Chapter 3

Gate Triggering Circuits

Scilab code Exa 3.1 Determine the trigger angle

```
// chapter 3
// example 3.1
// fig. E3.11
// Determine the trigger angle
// page-87
clear;
clc;
// given
Ig=0.1; // in mA (minimum gate current)
Vg=0.5; // in V (minimum gate voltage)
Emax=24; // in V (peak amplitude of input)
Rv=100; // in k-ohm
Rmin=10; // in k-ohm
Vd=0.7; // in V (threshold voltage for diode)
// calculate
e_s=Ig*(Rv+Rmin)+Vd+Vg;  // Applying KVL in the loop
// since e_s=Emax* sin(alpha)
alpha=asind(e_s/Emax); // calculation of trigger
   angle
printf("The trigger angle is \t %.1f degree",alpha);
```

Scilab code Exa 3.2 Determine the value of Vee

```
// chapter 3
// example 3.2
// Determine the value of Vee
// page-99
clear;
clc;
// given
Re=1; // in k-ohm
Iv=5; // in mA
Vv=2; // in V (voltage at valley point)
// calculate
Ie=Iv;
Ve=Vv;
Vee=Ie*Re+Ve; // calculation of Vee
printf("The required value to switch OFF UJT is \
   tVee=%.f V",Vee);
```

Scilab code Exa 3.3 Design UJT relaxation oscillator

```
// chapter 3
// example 3.3
// Design UJT relaxation oscillator
// page-99-100
clear;
clc;
// given
neta=0.7; // (intrinsic stand-off ratio)
I_P=50; // in uA (peak point curent)
Vv=2; // in V (valley point voltage)
I_V=6; // in mA (valley point current)
```

```
V_BB=20; // in V (maximum interbase voltage)
R_BB=7; // in k-ohm (interbase resistance)
I_E0=2; // in mA (emitter leakage current)
Vg_min=0.2 // in V (minimum gate voltage required to trigger SCR)
C=0.1; // in uF (assumption as done in the book)
// calculate
I_P=I_P*1E-6; // changing unit from uA to A
Rmax=V_BB*(1-neta)/I_P; // calculation of Rmax
I_V=I_V*1E-3; // changing unit from mA to A
Rmin=(V_BB-Vv)/I_V; // calculation of Rmin
R2=1E4/(neta*V_BB);
I_E0=I_E0*1E-3; // changing unit from mA to A
R1=Vg_min/I_E0;
C=C*1E-6; // changing unit from uF to F
Tmax=Rmax*C*log(1/(1-neta)); // calculation of maximum time period
Tmin=Rmin*C*log(1/(1-neta)); // calculation of minimum time period
fmin=1/Tmax; // calculation of minimum frequency
fmax=1/Tmin; // calculation of maximum frequency
Rmax=Rmax*1E-3; // changing unit from ohm to k-ohm
Rmin=Rmin*1E-3; // changing unit from ohm to k-ohm
Tmax=Tmax*1E3; // changing unit from s to ms
Tmin=Tmin*1E3; // changing unit from s to ms
fmax=fmax*1E-3; // changing unit from Hz to kHz
printf("\nThe maximum value of R is \t\t Rmax=%.f k-ohm",Rmax);
printf("\nThe minimum value of R is \t\t Rmin=%.f k-ohm",Rmin);
printf("\nThe value of R2 is \t\t\t R2=%.2f ohm",R2);
printf("\nThe value of R1 is \t\t\t R1=%.f ohm",R1);
printf("\nThe maximum value of time period is \t Tmax=%.2f ms",Tmax);
printf("\nThe minimum value of time period is \t Tmin=%.2f ms",Tmin);
printf("\nThe minimum value of frequency is \t fmin=
```

```
%.2f Hz",fmin);
printf("\nThe minimum value of frequency is \t fmax=%.2f kHz",fmax);
// Note : the answee of Tmax, fmin, fmax varies slightly due to exact calculation
```

Scilab code Exa 3.5 Design free running UJT relaxation oscillator

```
// chapter 3
// example 3.5
// Design free-running UJT relaxation oscillator
// page-105-106
clear;
clc;
// given
fmin=5; // in Hz (minimum frequency)
fmax=50; // in Hz (maximum frequency)
E_dc=12; // in V (DC supply)
I_P=80; // in mA (peak current)
T=8; // in us (trigger time)
V_drop=1; // in V (voltage drop across PUT)
Rs=39; // in ohm (Assumption as done in the book)
Ig=1; // in mA (assumption as done in the book
// calculate
// since T=Rs*C
C=T/Rs; // calculation of capacitance (in uF)
printf("\nThe value of capacitance is \t\t C=%.2f uF",C);
I_P=I_P*1E-3; // changing unit from mA to A
V_P=(I_P*Rs)+V_drop; // calculation of peak point voltage
// since V_P=neta*E_dc+V_D, neglecting V_D, we get
neta=V_P/E_dc; // calculation of intrinsic stand-off ratio
Tmax=1/fmin; // calculation of maximum time period
```

```
Tmin=1/fmax; // calculation of maximum time period
C=C*1E-6; // changing unit from uF to F
Rmax=Tmax/(C*log(E_dc/(E_dc-V_P))); // calculation of maximum value of R
Rmin=Tmin/(C*log(E_dc/(E_dc-V_P))); // calculation of minimum value of R
I_V_max=E_dc/Rmin; // calculation of maximum anode current
I_V_min=E_dc/Rmax; // calculation of minimum anode current
// since Ig=2*neta*E_dc/Rg , therefore
Rg=2*neta*E_dc/Ig; // calculation of gate resistance in k-ohm
R1=Rg/neta; // calculation of R1 resistance
R2=Rg/(1-neta); // calculation of R1 resistance
Rmax=Rmax*1E-6; // changing unit from ohm to M-ohm
Rmin=Rmin*1E-6; // changing unit from ohm to M-ohm
I_V_max=I_V_max*1E6; // changing unit from A to uA
I_V_min=I_V_min*1E6; // changing unit from A to uA
printf("\nThe peak point voltage is \t\t V_P=%.2f V",V_P);
printf("\nThe intrinsic stand-off ratio is \t neta=%.2f",neta);
printf("\nThe maximum value of R is \t\t Rmax=%.2f M-ohm",Rmax);
printf("\nThe minimum value of R is \t\t Rmin=%.2f M-ohm",Rmin);
printf("\nThe maximum value of anode current is \t I_V_max=%.2f uA",I_V_max);
printf("\nThe minimum value of anode current is \t I_V_min=%.f uA",I_V_min);
printf("\nThe value of gate resistance is \t Rg=%.2f k-ohm",Rg);
printf("\nThe value of R1 is \t\t\t R1=%.f k-ohm",R1);
printf("\nThe value of R2 is \t\t\t Rg=%.2f k-ohm",R2);
// Note : the answee of Rmax, I_V_max, Rg and R2
```

```
    varies slightly due to exact calculation
```

Chapter 4

Series and Parallel Operation of Thyristors

Scilab code Exa 4.1 Calculate the value of R and C

```
// chapter 4
// example 4.1
// Calculate the value of R and C
// page-120-121
clear;
clc;
// given
n_s=18; // number of thyristors connected is series
E_D=500; // in V (maximum permissible blocking
    voltage)
E_s=7500; // in V (string voltage)
dI_b=1; // in mA (range of blocking current)
dQ=30; // in uC (maximum permissible diffrenece
    between reverse recovery charge)
// calculate
R=(n_s*E_D-E_s)/((n_s-1)*dI_b); // calculation of R
    in k-ohm
C=(n_s-1)*dQ/(n_s*E_D-E_s); // calculation of C in
    uC
```

```
printf("\nThe value of R is \tR=%.2f k-ohm",R);
printf("\nThe value of C is \tC=%.2f uF",C);
```

Scilab code Exa 4.2 Calculate the series resistance

```
// chapter 4
// example 4.2
// Calculate the series resistance
// page-128-129
clear;
clc;
// given
I_T1=100; // in A (current of SCR 1)
I_T2=150; // in A (current of SCR 2)
V_T1=2.1; // in V (on-state voltage of SCR 1)
V_T2=1.75; // in V (on-state voltage of SCR 2)
// calculate
// since V_T1+I_T1*(R+R_T1)=V_T2+I_T2*(R+R_T2) and
   assume R+R_T1=R1 (resistance to be connected in
   series)
// therefore V_T1+I_T1*R1=V_T2+I_T2*R1 or we get
R1=(V_T1-V_T2)/(I_T2-I_T1);
printf("The value of required series resistance is \
   t R1=%.3f ohm",R1);
```

Scilab code Exa 4.3 Calculate the number of thyristors and value of resistance and capacitance

```
// chapter 4
// example 4.3
// Calculate the number of thyristors and value of
   resistance and capacitance
// page-130-131
```

```
clear;
clc;
// given
Es=6; // in kV (string voltage)
Im=4; // in kA (string current)
E_D=1.2; // in kV (thyristor voltage)
I_T=1; // in kA (thyristor current)
neta=90; // in percentage (string efficiency)
I_B_max=15; // in mA (maximum blockin current)
dQ_max=25; // in uC (maximum permissible difference between reverse reovery charge)
// calculate
neta=neta/100; //changing efficiency from percentage to ratio
n_s_float=Es/(E_D*neta); // calculation of number of thyristos in series
n_s= int16(n_s_float)+1; // since n_s will be reduced by decimal value it contaisn if type conversion is done because the decimal part would be removed so we need to add 1 to it
n_p_float=Im/(I_T*neta); // calculation of number of thyristos in parallel
n_p= int16(n_p_float)+1; // since n_p will be reduced by decimal value it contaisn if type conversion is done because the decimal part would be removed so we need to add 1 to it
n_s=double(n_s); // converting again to double for various calculatios
n_p=double(n_p); // converting again to double for various calculatios
printf("\nThe number of thyristors in series is \t n_s=%.f",n_s);
printf("\nThe number of thyristors in parallel is n_p=%.f",n_p);
E_D=(E_D*1E3); // changing unit from kV to V
Es=Es*1E3; // changing unit from kV to V
dI_b=(I_B_max); // (assumption as done in the book)
R=(n_s*E_D-Es)/((n_s-1)*dI_b); // calculation of
```

```
resistance in k-ohm
C=(n_s-1)*dQ_max/(n_s*E_D-Es); // calculation of capacitance in uF
printf("\nThe value of resistance is \t\t R<=%.f k-ohm",R);
printf("\nThe value of capacitance is \t\t C>=%.3f uF",C);
```

Scilab code Exa 4.4 Calculate the number of thyristors

```
// chapter 4
// example 4.4
// Calculate the number of thyristors
// page-132
clear;
clc;
// given
Es=7.5; // in kV (total applied voltage)
Im=1000; // in A (total forward current)
E_D=500; // in V (thyristor voltage)
I_T=75; // in A (thyristor current)
neta=14; // in percentage (derating factor)
// calculate
Es=Es*1E3; // changing unit from kV to V
neta=neta/100; //changing efficiency from percentage to ratio
// since neta=1-(Es/(n_s*E_D)) therefore
n_s_float=Es/(E_D*(1-neta)); // calculation of number of thyristos in series
n_s= int16(n_s_float)+1; // since n_s will be reduced by decimal value it contaisn if type conversion is done because the decimal part would be removed so we need to add 1 to it
// since neta=1-(Im/(n_s*I_T)) therefore
n_p_float=Im/(I_T*(1-neta)); // calculation of
```

```
number of thyristos in parallel
n_p= int16(n_p_float)+1; // since n_p will be reduced by decimal value it containsn if type conversion is done because the decimal part would be removed so we need to add 1 to it
printf("\nThe number of thyristors in series is \t n_s=%.f",n_s);
printf("\nThe number of thyristors in parallel is n_p=%.f",n_p);
```

Scilab code Exa 4.5 Determine the current taken by each SCR and value of equal resistors

```
// chapter 4
// example 4.5
// Determine the current taken by each SCR and value of equal resistors
// page-132
clear;
clc;
// given
// V1=0.9+2.4E-4*I_T1  (voltage characteristics of SCR 1)
// V2=1.0+2.3E-4*I_T2  (voltage characteristics of SCR 2)
I1=500, I2=1000, I3=1500, I4=2000; // in A (total current)
neta=10; // in percentage
// calculate
// since SCR are in parallel, therefore V1=V2 or
// 0.9+2.4E-4*I_T1=1.0+2.3E-4*I_T2. Simplifying this we get
// 2.4E-4*I_T1-2.3E-4*I_T2=0.1     (i)
// since I_T1+I_T2=I   (ii)
// from (i) in (ii), we get
```

```
// 2.4E-4*I_T1 -2.3E-4*(I-I_T1)=0.1  or
// 4.7E-4*I_T1=0.1+2.3E-4*I
// simplifying for I_T1, we get
// I_T1=(0.1+2.3E-4*I)/4.7E-4
for I=500:500:2000
    I_T1=(0.1+2.3E-4*I)/4.7E-4;
    I_T2=I-I_T1;
    printf("\n\nFor I=%.f A,\t I_T1=%.f A \t and \t I_T2=%.f A",I,I_T1,I_T2);
end
// For 10 % sharing I_T1=1100 A and I_T2=900 A , therefore
I_T1=1100, I_T2=900; // in A
R=(0.1+2.3E-4*I-4.7E-4*I_T1)/(I_T1-I_T2);
printf("\n\nThe value of equal resistors is \t R=%.3f m-ohm",R*1E3);
```

Chapter 5

Power Semiconductor Devices

Scilab code Exa 5.1 What will be the maximum and minimum firing angle

```
// chapter 5
// example 5.1
// Fig. 5.15
// What will be the maximum and minimum firing angle
// page-155
clear;
clc;
// given
Vc=40; // in V (breakdown voltage)
R1_min=1; R1_max=25; // in k-ohm
C=470; // in nF
Erms=240; // in V
f=50; // in Hz (AC supply frequency)
// calculate
C=C*1E-9; // changing unit from nF to F
R1_min=R1_min*1E3; // changing unit from k-ohm to
    ohm
R1_max=R1_max*1E3; // changing unit from k-ohm to
    ohm
Zc=1/(2*%pi*f*C); // impedence of capacitor
phi_min=90-atand(1/(2*%pi*f*R1_min*C));  //
```

```
calculation of minimum phase angle
phi_max=90-atand(1/(2*%pi*f*R1_max*C)); // calculation of minimum phase angle
Zd_min=sqrt(R1_min^2+Zc^2); // calculation of minimum impedence
Zd_max=sqrt(R1_max^2+Zc^2); // calculation of maximum impedence
Em=Erms*sqrt(2); // calculation of maximum value of voltage
Id_min=Em/Zd_max; //
Id_max=Em/Zd_min; // calculation of maximum impedence
Vc_min_peak=Id_min*Zc;
Vc_max_peak=Id_max*Zc;
// Now Vc=Vc_max_peak*sin(wt_min+phi_min) and Vc=Vc_min_peak*sin(wt_max+phi_max)

// evaluating both these, we get  wt_min=asind(Vc/Vc_max_peak)+phi_min and wt_max=asind(Vc/Vc_min_peak)+phi_max

wt_min=asind(Vc/Vc_max_peak)+phi_min;
wt_max=asind(Vc/Vc_min_peak)+phi_max;
printf("\nThe value of impedence of capacitor is \t\t Zc=%.f ohm",Zc);
printf("\n\nThe value of minimum phase difference is \t %.1f degree",phi_min);
printf("\nThe value of maximum phase difference is \t %.2f degree",phi_max);
printf("\n\nThe value of minimum total impedence is \t Zd_min=%.f ohm",Zd_min);
printf("\nThe value of maximum total impedence is \t Zd_max=%.f ohm",Zd_max);
printf("\n\nThe value of minimum peak current is \t\t Id_min=%.3f A",Id_min);
printf("\nThe value of maximum peak current is \t\t Id_max=%.3f A",Id_max);
printf("\n\nThe value of minimum peak total voltage
```

```
is \t Vc_min_peak=%.2f V",Vc_min_peak);
printf("\nThe value of maximum peak total voltage is \t Vc_max_peak=%.1f V",Vc_max_peak);
printf("\n\nThe value of minimum delay is \t\t\t =%.2f degree",wt_min);
printf("\nThe value of maximum delay is \t\t\t =%.1f degree",wt_max);
```

Scilab code Exa 5.2 Compute the drain current and differnces in current sharing

```
// chapter 5
// example 5.2
// Fig. E5.2
// Compute the drain current and differnces in current sharing
// page-174-175
clear;
clc;
// given
I_T=30; // in A (total current)
V_DS1=4; // in V (drain to source voltage of MOSFET 1)
V_DS2=4.5; // in V (drain to source voltage of MOSFET 2)
R_S1_a=0.4, R_S2_a=0.3; // in ohm (current sharing series resistance for part-a)
R_S1_b=0.7, R_S2_b=0.7; // in ohm (current sharing series resistance for part-b)
// calculate
// Since I_T=I_D1+I_D2 and V_DS1+ID1*R_S1=V_DS2+ID2*R_S2, simplifying both equations for I_D1, we get
// I_D1=(V_DS2-V_DS1+R_S2*I_T)/(R_S1+R_S2)
I_D1_a=(V_DS2-V_DS1+R_S2_a*I_T)/(R_S1_a+R_S2_a); // calculation of drain current of MOSFET 1 for part
```

```
-a
I_D1_a_percent=I_D1_a*100/I_T; // equivalent value in terms of percentage of total current
I_D2_a=I_T-I_D1_a; // calculation of drain current of MOSFET 2 for part-a
I_D2_a_percent=I_D2_a*100/I_T; // equivalent value in terms of percentage of total current
dI_a=abs(I_D2_a-I_D1_a); // calculation of difference in current for part-a
dI_a_percent=abs(I_D2_a_percent-I_D1_a_percent); // equivalent value in terms of percentage of total current
printf("\npart-a\n");
printf("\nThe drain current of MOSFET 1 is \t I_D1=%.2f A    or    %.2f %%",I_D1_a,I_D1_a_percent);
printf("\nThe drain current of MOSFET 2 is \t I_D2=%.2f A    or    %.2f %%",I_D2_a,I_D2_a_percent);
printf("\n\nThe difference in current sharing is \t%.2f A    or    %.2f %%",dI_a,dI_a_percent);
I_D1_b=(V_DS2-V_DS1+R_S2_b*I_T)/(R_S1_b+R_S2_b); // calculation of drain current of MOSFET 1 for part-a
I_D1_b_percent=I_D1_b*100/I_T; // equivalent value in terms of percentage of total current
I_D2_b=I_T-I_D1_b; // calculation of drain current of MOSFET 2 for part-a
I_D2_b_percent=I_D2_b*100/I_T; // equivalent value in terms of percentage of total current
dI_b=abs(I_D2_b-I_D1_b); // calculation of difference in current for part-a
dI_b_percent=abs(I_D2_b_percent-I_D1_b_percent); // equivalent value in terms of percentage of total current
printf("\n\npart-b\n");
printf("\nThe drain current of MOSFET 1 is \t I_D1=%.2f A    or    %.1f %%",I_D1_b,I_D1_b_percent);
printf("\nThe drain current of MOSFET 2 is \t I_D2=%.2f A    or    %.1f %%",I_D2_b,I_D2_b_percent);
```

```
printf("\n\nThe difference in current sharing is \t%.2f A   or   %.1f %%",dI_b,dI_b_percent);
// Note: the answers for some of the quantity may vary slightly due to precise calculation
```

Scilab code Exa 5.3 Determine series resistance and safe highest frequency

```
// chapter 5
// example 5.3
// Determine series resistance and safe highest frequency
// page-175
clear;
clc;
// given
I_T=120; // in A (total current)
C_gs1=2200; // in pF (gate to source capacitance of MOSFET 1)
C_gd1=390; // in pF (gate to drain capacitance of MOSFET 1)
L_g1=2; // in nH (stray inductance of MOSFET 1)
C_gs2=2700; // in pF (gate to source capacitance of MOSFET 2)
C_gd2=330; // in pF (gate to drain capacitance of MOSFET 2)
L_g2=2.3; // in nH (stray inductance of MOSFET 2)
// calculate
C_gs1=C_gs1*1E-12; // changing unit from pF to F
C_gd1=C_gd1*1E-12; // changing unit from pF to F
L_g1=L_g1*1E-9; // changing unit from nF to F
C_gs2=C_gs2*1E-12; // changing unit from pF to F
C_gd2=C_gd2*1E-12; // changing unit from pF to F
L_g2=L_g2*1E-9; // changing unit from nF to F
f1=1/(2*%pi*sqrt(L_g1*(C_gs1+C_gd1))); // calculation of oscillating frequency of MOSFET 1
```

```
f2=1/(2*%pi*sqrt(L_g2*(C_gs2+C_gd2))); // calculation of oscillating frequency of MOSFET 1
f_h=min(f1,f2)/10; // calculation of safe highest frequency
Rg=1/(2*%pi*(C_gs1+C_gd1)*f_h); // calculation of series resistance in the gate
f_h=f_h*1E-6; // changing unit from Hz to MHz
printf("\nThe oscillating frequency of MOSFET 1 is \t f1=%.2E Hz",f1);
printf("\nThe oscillating frequency of MOSFET 2 is \t f2=%.2E Hz",f2);
printf("\n\n Since the safe highest frequency should be at least 1 decade lower than oscillating frequency, therefore the safe highest frequency is \t %.f MHz",f_h);
printf("\n\nThe series resistance in the gate terminal of both MOSFET is \t Rg=%.2f ohm",Rg);
// Note: The answer for the series resistance varies due to precise calculation
```

Scilab code Exa 5.4 Determine the power dissipated by each MOSFET

```
// chapter 5
// example 5.4
// fig. E5.4
// Determine the power dissipated by each MOSFET
// page-175-176
clear;
clc;
// given
I_T=12; // in A (total current)
R1=0.2, R2=0.1 // in ohm
// calculate
I1=(R2/(R1+R2))*I_T; // applying current divider rule to calculate current through MOSFET 1
```

```
I2=I_T-I1; // calculation of current through MOSFET
    2
P1=I1^2*R1; // calculation of power dissipated by
    MOSFET 1
P2=I2^2*R2; // calculation of power dissipated by
    MOSFET 2
printf("\nThe power dissipated by MOSFET 1 is \t P1=
    %.1f W",P1);
printf("\nThe power dissipated by MOSFET 2 is \t P2=
    %.1f W",P2);
```

Scilab code Exa 5.5 Calculate the rise time of driving waveform

```
// chapter 5
// example 5.5
// Calculate the rise time of driving waveform
// page-178
clear;
clc;
// given
C_iss=470; /// in pf (input capacitance)
Rg=200; // in ohm
R_L=2.2; // in k-ohm
// calculate
C_iss=C_iss*1E-12; // changing unit from pF to F
// Since Rg=t_r/(2.2*Ciss) therefore
t_r=2.2*Rg*C_iss; // calculation of current through
    MOSFET 2
t_r=t_r*1E9; // changing unit from s to ns
printf("\nThe rise time is \t t_r=%.1f ns",t_r);
```

Scilab code Exa 5.6 Determine power loss

```
// chapter 5
// example 5.6
// fig. E 5.6
// Determine power loss
// page-182-183
clear;
clc;
// given
t_r=2; // in us (rise time)
R_DS_on=0.2; // in ohm (drain to source resistance when MOSFET is ON)
D=0.7; // duty cycle
f=30; // in kHz (frequency)
V_DS=100; // in V (DC power supply)
R_L=12; // in ohm (load resistance)
// calculate
I_D=V_DS/(R_L+R_DS_on); // calculation of drain current
f=f*1E3; // changing unit from kHz to Hz
T=1/f; // calculation of switching period
t_on=D*T; // calculation of on-time
W_on=I_D^2*R_DS_on*t_on; // calculation of energy loss during on-time
P_on=W_on*f; // calculation of power loss during on-time
printf("\nThe power loss in the on-state is \t\t P_on=%.2f W",P_on);
t_r=t_r*1E-6; // changing unit from us to s
W_ON=V_DS*I_D*t_r/6; // calculation of energy loss during turn-on interval
P_ON=W_ON*f; // calculation of power loss during turn-on interval
printf("\nThe power loss during the turn-on interval is \t P_ON=%.1f W",P_ON);
```

Scilab code Exa 5.7 Calculate the total power loss

```
// chapter 5
// example 5.7
// Calculate the total power loss
// page-183
clear;
clc;
// given
V_DS=120; // in V (DC power supply)
I_D=4; // in A (drain current)
t_r=80; // in ns (rise time)
t_f=120; // in ns (fall time)
I_DSS=2; // in mA (drain current at saturation)
R_DS_on=0.2; // in ohm (drainn ot source resistance when MOSFET is ON)
D=50; // in percentage (duty cycle)
f=45; // in kHz (swithcing frequency)
// calculate
f=f*1E3; // changing unit from kHz to Hz
I_DSS=I_DSS*1E-3; // changing unit from mA to A
t_r=t_r*1E-9; // changing unit from ns to s
t_f=t_f*1E-9; // changing unit from ns to s
T=1/f; // calculation of total period
t_on=D*T/100; // calculation of on-time
t_off=(100-D)*T/100; // calculation of on-time
P_on=I_D^2*R_DS_on*t_on/T; // calculation of on-state power loss
P_off=V_DS*I_DSS*t_off/T; // calculation of off-state power loss
Pw_on=(V_DS*I_D*t_r/6)*f; // calculation of turn-on switching power loss
Pw_off=(V_DS*I_D*t_f/6)*f; // calculation of turn-off switching power loss
P_T=P_on+P_off+Pw_on+Pw_off; // calculation of total power loss
printf("\nThe total power loss is \t P_T=%.2f W",P_T);
```

Scilab code Exa 5.8 Determine avg load current and power losses

```
// chapter 5
// example 5.8
// Fig. E 5.8
// Determine avg load current and power losses
// page-201
clear;
clc;
// given
t_ON=3; // in us (turn-on time)
t_OFF=1.2; // in us (tunr-off time)
D=0.7; // duty cycle
V_CE_sat=2; // in V (collector to emitter voltage at
    saturation)
f=1; // in kHz (swithcing frequency)
R_L=10; // in ohm (load resistance)
V_CC=200; // in V (collector to emitter volatge)
// calculate
t_ON=t_ON*1E-6; // changing unit from us to s
t_OFF=t_OFF*1E-6; // changing unit from us to s
f=f*1E3; // changing unit from kHz to Hz
V_CE_max=V_CC;
I_C_max=(V_CC-V_CE_sat)/R_L; // calculation of
   maximum collector(load) current
I_C_avg=D*I_C_max; // calculation of average
   collector(load) current
P_con=V_CE_sat*I_C_avg; // calculation of conduction
    power loss
Pw_ON=(V_CE_max*I_C_max*t_ON/6)*f; // calculation of
    swithcing power loss during turn-on
Pw_OFF=(V_CE_max*I_C_max*t_OFF/6)*f; // calculation
   of swithcing power loss during turn-off
printf("\nThe average load current is \t\t\t I_C_avg
```

```
=%.2f A",I_C_avg);
printf("\nThe conduction power loss is \t\t\t %.2f W",P_con);
printf("\nThe swithcing power loss during turn-on is \t %.2f W",Pw_ON);
printf("\nThe swithcing power loss during turn-off is \t %.3f W",Pw_OFF);
```

Chapter 6

Phase Controlled Convertors

Scilab code Exa 6.2 Determine the heater power

```
// chapter 6
// example 6.2
// Determine the heater power
// page-270
clear;
clc;
// given
P=1; // in kW (power rating)
E=230; // in V (voltage rating)
f=50; // in Hz (supply frequency)
alpha1=45, alpha2=90; // in degree (delay angle)
// calculate
P=P*1E3; // changing unit from kW to W
Em=E*sqrt(2); // calculation of maximum value of supply voltage
alpha1_rad=(%pi/180)*alpha1; // changing from degree to radian
alpha2_rad=(%pi/180)*alpha2; // changing from degree to radian
Erms1=Em*sqrt(((%pi-alpha1_rad)/(4*%pi))+((sin(2*alpha1_rad))/(8*%pi))); // calculation of rms
```

```
   value of E
R=E^2/P; // calculation of resistance
W=Erms1^2/R; // calculation of heater power at 45
   degree
Erms2=Em*sqrt(((%pi-alpha2_rad)/(4*%pi))+((sin(2*
   alpha2_rad))/(8*%pi))); // calculation of rms
   value of E
W2=Erms2^2/R; // calculation of heater power at 45
   degree
printf("\nThe heater power at %.f degree is \t W1=%
   .2f W",alpha1,W);
printf("\nThe heater power at %.f degree is \t W2=%.
   f W",alpha2,W2);
```

Scilab code Exa 6.3 Determine various parameters

```
// chapter 6
// example 6.3
// Determine various parameters
// page-270-271
clear;
clc;
// given
E=120; // in V (voltage supply)
f=50; // in Hz (supply frequency)
R=10; // in ohm
neta=25; // in percentage
// calculate
Em=E*sqrt(2); // calculation of maximum value of
   supply voltage
Edc=(neta/100)*(Em/%pi); // calculation of dc value
   of voltage
// since Edc=(Em/(2*%pi))*(1+cos(alpha)) solving for
    alpha, we get
alpha=acosd(Edc*(2*%pi/Em)-1); // calculation of
```

```
   firing angle
Idc=Edc/R; // calculation of average output current
alpha_rad=(%pi/180)*alpha; // calculation of firing
   angle in radian
Erms=Em*sqrt(((%pi-alpha_rad)/(4*%pi))+((sin(2*
   alpha_rad))/(8*%pi))); // calculation of rms
   voltage
Irms=Erms/R; // calculation of rms output current
Idc_SCR=Idc; // calculation of average SCR current
Irms_SCR=Irms; // calculation of rms SCR current
printf("\nThe firing angle is \t%.f",alpha);
printf("\nThe average output current is \tIdc=%.2f A
   ",Idc);
printf("\nThe rms output current is \tIrms=%.2f A",
   Irms);
printf("\nThe average SCR current is \t%.2f A",
   Idc_SCR);
printf("\nThe rms output current is \t%.2f A",
   Irms_SCR);
// Note: There is calculation mistake in the book
   for alpha due to which other values are also
   coming different.That's why all the answer vary
```

Scilab code Exa 6.4 Determine average current power supplied and dissipated and power factor

```
// chapter 6
// example 6.4
// Determine average current, power supplied and
   dissipated and power factor
// page-271-272
clear;
clc;
// given
Erms=230; // in V (voltage supply)
```

```
f=50; // in Hz (supply frequency)
R=5; // in ohm
Eb=150; // in V
// calculate
Em=Erms*sqrt(2); // calculation of maximum value of supply voltage
alpha1=asin(Eb/Em); // calculation of firing angle
Id=(1/(2*%pi*R))*(2*Em*cos(alpha1)-Eb*(%pi-2*alpha1)); // calculation of average charging current
P_s=Eb*Id; // calculation of power supplied to battery
Irms=sqrt((1/(2*%pi*R^2))*((Eb^2+Erms^2)*(%pi-2*alpha1)+(Erms^2*sin(2*alpha1))-(4*Em*Eb*cos(alpha1)))); // calculation of rms current
P_d=Irms^2*R; // calculation of power dissipated in resistor
pf=(P_d+P_s)/(Erms*Irms); // calculation of supply power factor
printf(”\nThe average charging current is \tId=%.2f A”,Id);
printf(”\nThe power supplied to battery is \t%.2f W or \t %.2f kW”,P_s,P_s*1E-3);
printf(”\nThe rms current is \tIrms=%.2f A”,Irms);
printf(”\nThe power dissipated in resistor is \t%.2f W or \t %.2f kW”,P_d,P_d*1E-3);
printf(”\nThe supply power factor is \t%.3f”,pf);
// Note: 1.There is calculation mistake in the book for P_s .
//       2. The formula used in the book to calculate Irms has little printing error for Em. It should be Erms for the first two Em used
//       3. The answer slightly vary due to precise calculations
```

Scilab code Exa 6.5 Determine design details

```
// chapter 6
// example 6.5
// fig. 6.14
// Determine design details
// page-287-288
clear;
clc;
// given
E_supply=230; // in V (voltage supply)
Im=12; // in A
Edc=150; // in V
alpha=30; // in degree
V_drop=1.5; // in V (voltage drop across SCR)
// calculate
// Since Edc=(2*Em/%pi)*cos(alpha)-V_drop, evaluating it for Em we get
Em=(%pi*(Edc+V_drop))/(2*cosd(alpha)); // calculation of peak voltage
Erms=Em/sqrt(2); // calculation of rms voltage
Irms=Im/sqrt(2); // calculation of rms current
TSR=2*Erms*Irms; // calculation of Transformer secondory rating
K=E_supply/Erms; // calculation of Transformer voltage ratio
I_P=Im*(1/K); // calculation of Transformer primary current
PIV=2*Em;// calculation of PIV for each SCR
printf("\nFor M-2 connection\n")
printf("\n\tThe peak voltage is \t\t\tEm=%.1f V",Em);
printf("\n\tThe rms volatge is \t\t\tErms=%.2f V",Erms);
printf("\n\tThe rms current is \t\t\tIrms=%.2f A",Irms);
printf("\n\tThe Transformer secondory rating is \t %.1f kVA",TSR*1E-3);
printf("\n\tThe Transformer voltage ratio is \tK=%.2f",K);
```

```
printf("\n\tThe Transformer primary current is \tI_P=%.2f A",I_P);
printf("\n\tThe PIV for each SCR is \t\tPIV=%.f V",PIV);
Em=(%pi*(Edc+2*V_drop))/(2*cosd(alpha)); // calculation of peak voltage
Erms=Em/sqrt(2); // calculation of rms voltage
Irms=Im; // calculation of rms current
TSR=Erms*Irms; // calculation of Transformer secondory rating
K=E_supply/Erms; // calculation of Transformer voltage ratio
I_P=Im*(1/K); // calculation of Transformer primary current
PIV=Em;// calculation of PIV for each SCR
I_rms_SCR=Im/sqrt(2); // calculation of rms current for each SCR
printf("\nFor B-2 connection\n")
printf("\n\tThe peak voltage is \t\t\tEm=%.2f V",Em);
printf("\n\tThe rms volatge is \t\t\tErms=%.2f V",Erms);
printf("\n\tThe rms current is \t\t\tIrms=%.f A",Irms);
printf("\n\tThe Transformer secondory rating is \t %.2f kVA",TSR*1E-3);
printf("\n\tThe Transformer voltage ratio is \tK=%.2f",K);
printf("\n\tThe Transformer primary current is \tI_P=%.2f A",I_P);
printf("\n\tThe PIV for each SCR is \t\tPIV=%.2f V",PIV);
printf("\n\tThe rms current for each SCR is \t%.1f A",I_rms_SCR);
SCR_loss_M2=(V_drop/Edc)*100; // SCR loss in M2 connection
SCR_loss_B2=(2*V_drop/Edc)*100; // SCR loss in B2 connection
```

```
printf("\n\nThe SCR  loss in M2 connection is \t%.f percent of the load power",SCR_loss_M2);
printf("\nThe SCR  loss in B2 connection is \t%.f percent of the load power",SCR_loss_B2);
```

Scilab code Exa 6.6 Determine variuos parameters

```
// chapter 6
// example 6.6
// fig. 6.21
// Determine variuos parameters
// page-288-290
clear;
clc;
// given
Erms=230; // in V (voltage supply)
alpha=%pi/3; // in degree (firing angle)
Id=4; // in A (load current)
// calculate
Em=Erms*sqrt(2); // calculation of rms voltage
Edc=(2*Em/%pi)*cos(alpha); // calculation of dc output voltage
Pi=(2*Em/%pi)*Id*cos(alpha); // calculation of active power input
Qi=(2*Em/%pi)*Id*sin(alpha); // calculation of reactive power input
R=Edc/Id; // calculation of load resistance
Edc_1=(Em/%pi)*(1+cos(alpha)); // calculation of dc output voltage when free wheeling diode is connected
Id_1=Edc_1/R;
I1=(2*sqrt(2)*Id_1/%pi)*cos(alpha/2); // calculation of fundamental value of alternating current
Pi_1=(Erms*I1)*cos(alpha/2); // calculation of active input power input when free wheeling diode
```

```
is connected
Qi_1=(Erms*I1)*sin(alpha/2); // calculation of reactive power input when free wheeling diode is connected
Edc_2=Em/(2*%pi)*(1+cos(alpha)); // calculation of dc output voltage when SCR3 is damaged
Id_2=Edc_2/R; // calculation of dc output current when SCR3 is damaged
printf("\nThe dc output voltage is \tEdc=%.2f V",Edc);
printf("\nThe active power input is \tPi=%.2f W",Pi);
printf("\nThe reactive power input is \tQi=%.2f Vars",Qi);
printf("\nThe load resistance is \t\tR=%.2f ohm",R);
printf("\n\nWhen free wheeling diode is connected\n");
printf("\n\tThe dc output voltage is \tEdc=%.2f V",Edc_1);
printf("\n\tThe dc output current is \tIdc=%.f V",Id_1);
printf("\n\tThe fundamental value of alternating current is \tI1=%.2f A",I1);
printf("\n\tThe active power input is \tPi=%.2f W",Pi_1);
printf("\n\tThe reactive power input is \tQi=%.f Vars",Qi_1);
printf("\n\nWhen SCR3 is damaged\n");
printf("\n\tThe dc output voltage is \tEdc=%.2f V",Edc_2);
printf("\n\tThe dc output current is \tIdc=%.f A",Id_2);
```

Scilab code Exa 6.7 Determine the firing angle

```
// chapter 6
// example 6.7
// fig. E6.7
// Determine the firing angle
// page-290-291
clear;
clc;
// given
Erms=220; // in V (voltage supply)
R=0.5; // in ohm (load resistance)
Idc=10; // in A (output current)
Ec_a=135; Ec_b=-145; // in V (capacitor voltage)
// calculate
Em=Erms*sqrt(2); // calculation of rms voltage
Edc_a=Idc*R+Ec_a; // calculation of dc output
    voltage
alpha_a=acosd(%pi/(2*Em)*Edc_a); // calculation of
    firing angle when Vc=135 V
Edc_b=Idc*R+Ec_b; // calculation of dc output
    voltage
alpha_b=acosd(%pi/(2*Em)*Edc_b); // calculation of
    firing angle when Vc=-145 V
printf("\nThe firing angle when Ec= %.f V is \t= %.f
     degree",Ec_a,alpha_a);
printf("\n\nThe firing angle when Ec= %.f V is \t= %
    .f degree",Ec_b,alpha_b);
```

Scilab code Exa 6.8 Determine firing angle output current and thyristor current

```
// chapter 6
// example 6.8
// Determine firing angle, output current and
    thyristor current
// page-298-299
```

```
clear;
clc;
// given
E=120; // in V (voltage supply)
f=50; // supply frequency
R=10; // in ohm (load resistance)
neta=25; // in percent
alpha_min=0; // angle for which average output voltage will be maximum
// calculate
Em=E*sqrt(2); // calculation of rms voltage
Edc_max=(Em/%pi)*(1+cos(alpha_min)); // calculation of maximum average output voltage
Edc=(neta/100)*Edc_max; // calculation of average output voltage
// Since Edc=(Em/%pi)*(1+cos(alpha)) , therefore we get
alpha=acos((%pi/Em)*Edc-1); // calculation of firing angle in radian
Erms=Em*sqrt(((%pi-alpha)/(2*%pi))+((sin(2*alpha))/(4*%pi))); // calculation of rms load voltage
Irms=Erms/R; // calculation of rms load current
Idc=Edc/R; // calculation of average load current
Im=Em/R; // calculation of maximum thyristor current
I_TH_avg=(Im/(2*%pi))*(integrate('sin(wt)','wt',alpha,%pi)); // calculation of average thyristor current
I_TH_rms=sqrt((Im^2/(2*%pi))*(integrate('(sin(wt))^2','wt',alpha,%pi))); // calculation of rms thyristor current
printf("\nThe firing angle is \t\t\t %.f degree",alpha*(180/%pi));
printf("\nThe average output voltage is \t\t Edc=%.f V",Edc);
printf("\nThe rms load voltage is \t\t Erms=%.f V",Erms);
printf("\nThe average output current is \t\t Idc=%.1f A",Idc);
```

```
printf("\nThe rms load current is \t\t Irms=%.1f A",Irms);
printf("\nThe average thyristor current is \t I_TH_avg=%.2f A",I_TH_avg);
printf("\nThe rms thyristor current is \t\t I_TH_rms=%.2f A",I_TH_rms);
```

Scilab code Exa 6.9 Find average output current and power delivered

```
// chapter 6
// example 6.9
// fig. Ex.6.9
// Find average output current and power delivered
// page-299-300
clear;
clc;
// given
E=230; // in V (voltage supply)
f=50; // supply frequency
alpha=%pi/4; // in radian (firing angle)
R=10; // in ohm (load resistance)
L=10; // in mH
Ec=120; // in V (capacitor voltage)
// calculate
Em=E*sqrt(2); // calculation of peak voltage
Edc=(Em/(2*%pi))*(integrate('sin(wt)','wt',alpha,%pi)+abs(integrate('(sin(wt))','wt',%pi,2*%pi))); // The formula used in the book has little mistake
Idc=(Edc-Ec)/R; // calculation of average output current
P=Ec*Idc; // calculation of power delivered
printf("\nThe average output current is \t Idc=%.2f A",Idc);
printf("\nThe power delivered is \t\t P=%.2f W",P);
// Note: 1. The integrating formula used in the book
```

```
has little mistake but the answer is same because they have mistake in calculation. I have used the correct formula
//       2. The answers vary slightly due to precise calculation upto 6 decimal digit. In the book, the calculation is upto 2 decimal digit
```

Scilab code Exa 6.10 Determine various parameters

```
// chapter 6
// example 6.10
// Determine various parameters
// page-300-301
clear;
clc;
// given
E=230; // in V (voltage supply)
alpha=%pi/6; // in radian (firing angle)
// calculate
Em=E*sqrt(2); // calculation of peak output voltage
Edc=(2*Em/%pi)*cos(alpha); // calculation of average output voltage
Irms_supply=’Id’; // Supply RMS current
Irms_fundamental=’(2∗sqrt(2)/%pi)∗Id’; // Supply fundamental RMS current
pf_fundamental=cos(alpha); // calculation of fundamental power factor
pf_supply=(2*sqrt(2)/%pi)*cos(alpha); // calculation of supply power factor
hf_supply=sqrt(%pi^2/8-1); // calculation of fundamental harmonic factor
KV=sqrt(%pi^2/(8*(cos(alpha))^2)-1); // calculation of voltage ripple factor
printf(”\nThe average output voltage is \t\t Edc=%.2f V”,Edc);
```

```
printf("\nThe supply RMS current is \t\t Irms_supply=%s",Irms_supply);
printf("\nThe supply fundamental RMS current is \t Irms_fundamental=%s",Irms_fundamental);
printf("\nThe fundamental power factor is \t pf_fundamental=%.2f",pf_fundamental);
printf("\nThe supply power factor is \t\t pf_supply=%.2f",pf_supply);
printf("\nThe fundamental harmonic factor is \t hf_supply=%.3f",hf_supply);
printf("\nThe voltage ripple factor is \t\t KV=%.3f",KV);
```

Scilab code Exa 6.11 Find displacement factor harmonic factor and input power factor

```
// chapter 6
// example 6.11
// Find displacement factor , harmonic factor and input power factor
// page-301
clear;
clc;
// given
E=120; // in V (voltage supply)
f=50; // supply frequency
alpha=%pi/6; // in radian (firing angle)
// calculate
DF=cos(-alpha/2); // calculation of displacement factor
// Since HF=sqrt((Irms/Irms1)^2-1) where Irms=Idc*sqrt(1-(alpha/%pi)) and Irms1=(2*sqrt(2)*Idc/%pi)*cos(alpha/2)
//putting both value in formula for HF, we get HF=sqrt((sqrt(1-(alpha/%pi))/((2*sqrt(2)/%pi)*cos(
```

```
alpha/2)))^2-1)
HF=sqrt((sqrt(1-(alpha/%pi))/((2*sqrt(2)/%pi)*cos(alpha/2)))^2-1); // calculation of harmonic factor of input current
// since pf=(Irms1/Irms)*cos(alpha/2) , putting the value of both we get pf=(2*sqrt(2)/%pi)*cos(alpha/2)/(qrt(1-(alpha/%pi)))
PF=(2*sqrt(2)/%pi)*cos(alpha/2)/(sqrt(1-(alpha/%pi))); // calculation of input power factor
printf("\nThe displacement factor is \t\t DF=%.4f",DF);
printf("\nThe harmonic factor of input current is HF=%.2f percent",HF*100);
printf("\nThe input power factor is \t\t PF=%.3f ",PF);
if PF>0 then
    printf("(lagging)");
else
    printf("(leading)");
end
// Note: The answer for DF, HF, PF varies slightly due to precise calculation upto 6 decimal digit where as there is 2 decimal digit calculation in the book
```

Scilab code Exa 6.12 Determine average output voltage

```
// chapter 6
// example 6.12
// Determine average output voltage
// page-319-320
clear;
clc;
// given
E=200; // in V (voltage supply per phase)
```

```
alpha1=0, alpha2=30, alpha3=60; // in degree (firing angle)
V_drop=1.5; // in V (voltage drop across SCR)
// calculate
Em=E*sqrt(2); // calculation of peak value of voltage
Edc1=(3*sqrt(3)/(2*%pi))*Em*cosd(alpha1)-V_drop; // calculation of average load voltage for alpha=0 degree
Edc2=(3*sqrt(3)/(2*%pi))*Em*cosd(alpha2)-V_drop; // calculation of average load voltage for alpha=30 degree
Edc3=(3*sqrt(3)/(2*%pi))*Em*cosd(alpha3)-V_drop; // calculation of average load voltage for alpha=60 degree
printf("\nThe average load voltage for alpha=%.f degree\ is t Edc1=%.2f V",alpha1,Edc1);
printf("\n\nThe average load voltage for alpha=%.f degree\ is t Edc1=%.2f V",alpha2,Edc2);
printf("\n\nThe average load voltage for alpha=%.f degree\ is t Edc1=%.2f V",alpha3,Edc3);
```

Scilab code Exa 6.13 Determine average load voltage current PIV and power dssipation

```
// chapter 6
// example 6.13
// Determine average load voltage, current, PIV and power dssipation
// page-320-321
clear;
clc;
// given
E_line=380; // in V (voltage line supply)
I=32; // in A (load-current)
```

```
alpha1=0, alpha2=45; // in degree (firing angle)
Vt=1.2; // in V (voltage drop across SCR)
// calculate
Em=E_line*sqrt(2/3); // calculation of peak value of voltage
Edc1=(3*sqrt(3)/(2*%pi))*Em*cosd(alpha1)-Vt; // calculation of average load voltage for alpha=0 degree
Edc2=(3*sqrt(3)/(2*%pi))*Em*cosd(alpha2)-Vt; // calculation of average load voltage for alpha=30 degree
Irms=I/sqrt(3); // calculation of rms current
PIV=sqrt(2)*E_line;// calculation of peak inverse voltage
Pavg=(1/(2*%pi))*(Vt*I)*integrate('1','wt',alpha1,(alpha1+(2*%pi/3)));// calculation of average power dissipation in each thyristor
printf("\nThe average load voltage for alpha=%.f degree is \t\t Edc1=%.1f V",alpha1,Edc1);
printf("\nThe average load voltage for alpha=%.f degree is \t Edc1=%.1f V",alpha2,Edc2);
printf("\nThe rms current is \t\t\t\t\t Irms=%.2f A",Irms);
printf("\nThe Peak Inverse Voltage is \t\t\t\t PIV=%.1f V",PIV);
printf("\nThe average power dissipation in each thyristor is \t Pavg=%.1f W",Pavg);
```

Scilab code Exa 6.14 Calculate thyristor current ratings

```
// chapter 6
// example 6.14
// Calculate thyristor current ratings
// page-321
clear;
```

```
clc;
// given
E_line=415; // in V (voltage line supply) The value is used  in the solution but not mentioned
R_L=10; // in ohm
alpha=30; // in degree (firing angle)
// calculate
Em=E_line*sqrt(2/3); // calculation of peak value of voltage
Edc=(3*sqrt(3)/(2*%pi))*Em*cosd(alpha); // calculation of average load voltage
Idc=Edc/R_L; // calculation of average load current
I_T_rms=Idc/sqrt(3); // calculation of thyristor rms current
I_T_dc=Idc/3; // calculation of thyristor average load current
I_T_peak=Idc; // calculation of thyristor peak current
PIV=E_line*sqrt(2); // calculation of thyristor peak inverse voltage
printf("\nThe average load current \t\t Idc=%.1f A",Idc);
printf("\nThe thyristor average load current is \t I_T_dc=%.1f A",I_T_dc);
printf("\nThe thyristor rms current is \t\t I_T_rms=%.1f A",I_T_rms);
printf("\nThe thyristor peak current is \t\t I_T_peak=%.1f A",I_T_peak);
printf("\nThe thyristor peak inverse voltage is \t PIV=%.1f V",PIV);
```

Scilab code Exa 6.15 Determine various parameters

```
// chapter 6
// example 6.15
```

```
// Determine various parameters
// page-322-323
clear;
clc;
// given
E_L=220; // in V (line voltage)
f=50; // in Hz (supply frequency)
R_L=10; // in ohm
neta=25; // in percent
// calculate
Es=E_L/sqrt(3); // calculation of phase voltage
Em=sqrt(2)*Es; // calculation of peak voltage
Edc_max=(3*sqrt(3)/(2*%pi))*Em; // calculation of maximum average load voltage
Edc=(neta/100)*Edc_max; // calculation of average load voltage
if neta>86.6 then
    alpha=acosd((2*%pi/(3*sqrt(3)*Em))*Edc);
else
    alpha=acosd((2*%pi/(3*Em))*Edc-1)-30;

end
printf("\nThe delay angle \t\t\t %.1f degree",alpha);
Idc=Edc/R_L; // calculation of average load current
alpha=alpha*%pi/180; // changing from degree to radian
Erms=sqrt((Em^2/(2*%pi/3))*integrate('(sin(wt))^2','wt',0.691*%pi,%pi)); // calculation of rms voltage
Irms=Erms/R_L;// calculation of rms current
I_T_dc=Idc/3; // calculation of thyristor average load current
I_T_rms=Irms/sqrt(3); // calculation of thyristor rms current
Pdc=Idc^2*R_L; // calculation of dc power
Pac=Irms^2*R_L; // calculation of ac power
neta=(Pdc/Pac)*100; // calculation of rectification
```

```
efficiency
Input_VA=3*Es*I_T_rms; // calculation of input volt-ampere rating
TUF=(Pdc/Input_VA)*100; // calculation of transformer utilisation factor
pf=Pac/Input_VA; // calculation of input power factor
printf("\nThe average output current is \t\t Idc=%.3f A",Idc);
printf("\nThe rms output current is \t\t Irms=%.2f A",Irms);
printf("\nThe thyristor average current is \t I_T_dc=%.2f A",I_T_dc);
printf("\nThe thyristor rms current is \t\t I_T_rms=%.2f A",I_T_rms);
printf("\nThe rectification efficiency is \t %.2f percent",neta);
printf("\nThe transformer utilisation factor is \t TUF=%.2f percent",TUF);
printf("\nThe input power factor is \t\t pf=%.2f",pf);
if pf>0 then
    printf(" (lagging)");
else
    printf(" (leading)");
end
// Note: The answer for Erms, Irms, I_T_rms, neta, TUF, pf vary slightly due to precise calculation
```

Scilab code Exa 6.17 Calculate firing angle and supply power factor

```
// chapter 6
// example 6.17
// Calculate firing angle and supply power factor
// page-341-342
```

```
clear;
clc;
// given
Vs=230; // in V (voltage supply)
f=50; // supply frequency
E_batt=200; // in V (battery emf)
r=0.5; // in ohm (internal resistance of battery)
Id=20; // in A (load current)
// calculate
Edc=E_batt+Id*r; // calculation of battery terminal voltage
Em=Vs*sqrt(2/3); // calculation of peak voltage
// Since Edc=(3*sqrt(3)*Em/%pi)*cos(alpha) we get
alpha=acosd(%pi/(3*sqrt(3)*Em)*Edc); // calculation of firing angle
Is=sqrt((1/%pi)*Id^2*(2*%pi/3)); // calculation of supply current over one-half cycle
Idc_rms=Id; // calculation of rms value of output current
Pd=Edc*Id+Idc_rms^2*r; // calculation of power delivere to the load
// Since Pd=sqrt(3)*Vs*Is*cos1(alpha1) where cos(alpha1) is supply power factor therefore we get
pf=Pd/(sqrt(3)*Vs*Is); // calculation of supply power factor
Edc_batt=E_batt-Id*r; // calculation of dc voltage when battery is delivering power
// since (-)Edc_batt=(3*sqrt(3)/%pi*Emph*cosd(alpha)) therefore we get
alpha2=acosd((%pi/(3*sqrt(3)))*(-Edc_batt)/Em); // calculation of firing angle
printf("\nThe firing angle is \t\t %.2f degree",alpha);
printf("\nThe supply power factor is \t %.2f",pf);
if pf>0 then
    printf(" (lagging)");
else
    printf(" (leading)");
```

```
end
printf(”\n\nWhen battery is delivering the power, the firing angle is \t%.3f degree”,alpha2);
// Note:  The answers for all required values varies slightly due to precise calculation upto 6 digit
```

Scilab code Exa 6.19 Find out SCR rating

```
// chapter 6
// example 6.19
// Find out SCR rating
// page-343-344
clear;
clc;
// given
Es=415; // in V (voltage supply)
f=50; // supply frequency
R=100; // in ohm (load resistance)
alpha=0; // in radian
// calculate
Em=Es*sqrt(2/3); // calculation of peak voltage
Edc=(3*sqrt(3)*Em/%pi)*cos(alpha); // calculation of average voltage
Idc=Edc/R; // calculation of average output current
I_T_av=(1/(2*%pi))*Idc*integrate(’1’,’wt’,(alpha+%pi/6),(alpha+5*%pi/6)); // calculation of SCR average current
I_T_rms=sqrt((1/(2*%pi))*Idc^2*integrate(’1’,’wt’,(alpha+%pi/6),(alpha+5*%pi/6))); // calculation of SCR rms current
I_T_peak=Idc; // calculation of SCR peak current
PIV=sqrt(3)*Em; // calculation of peak inverse voltage of SCR
printf(”\nThe SCR average current is \t I_T_av=%.3f A”,I_T_av);
```

```
printf("\nThe SCR rms current is \t\t I_T_rms=%.3f A",I_T_rms);
printf("\nThe SCR peak current is \t I_T_peak=%.3f A",I_T_peak);
printf("\n\nThe peak inverse voltage of SCR is \t PIV>=%.f A",PIV);
// Note: The answers vary slightly due to precise calculations upto 6 decimal digit
```

Scilab code Exa 6.20 Determine fundamental and various harmonics for various angle

```
// chapter 6
// example 6.20
// Determine fundamental and various harmonics for various angle
// page-344-345
clear;
clc;
clf;
// given
Es=415; // in V (voltage supply)
f=50; // supply frequency
R_L=100; // in ohm (load resistance)
alpha=0:30:180; // in radian
n1=1, n5=5, n7=7, n11=11, n13=13;
// calculate
Em=Es*sqrt(2/3); // calculation of peak voltage
printf("\nThe angles are\t\t\t\t")
for alpha=0:30:180
    printf("\t%.f\t",alpha);

end
printf("\nFundamental component of the supply current is");
```

```
for alpha=0:30:180
    Edc=(3*sqrt(3)*Em/%pi)*cosd(alpha); // calculation of average voltage
    Idc=Edc/R_L; // calculation of average output current
    Isn1=abs((2*sqrt(2)*Idc/%pi)*cosd(30*n1)/n1);
    printf(”\t%.3f\t”,Isn1);

end
printf(”\n%.fth harmonics component is\t\t”,n5);
for alpha=0:30:180
    Edc=(3*sqrt(3)*Em/%pi)*cosd(alpha); // calculation of average voltage
    Idc=Edc/R_L; // calculation of average output current
    Isn5=abs((2*sqrt(2)*Idc/%pi)*cosd(30*5)/n5);
    printf(”\t%.3f\t”,Isn5);
end
printf(”\n%.fth harmonics component is\t\t”,n7);
for alpha=0:30:180
    Edc=(3*sqrt(3)*Em/%pi)*cosd(alpha); // calculation of average voltage
    Idc=Edc/R_L; // calculation of average output current
    Isn7=abs((2*sqrt(2)*Idc/%pi)*cosd(30*n7)/n7);
    printf(”\t%.3f\t”,Isn7);
end
printf(”\n%.fth harmonics component is\t\t”,n11);
for alpha=0:30:180
    Edc=(3*sqrt(3)*Em/%pi)*cosd(alpha); // calculation of average voltage
    Idc=Edc/R_L; // calculation of average output current
    Isn11=abs((2*sqrt(2)*Idc/%pi)*cosd(30*n11)/n11);
    printf(”\t%.3f\t”,Isn11);
end
printf(”\n%.fth harmonics component is\t\t”,n13);
for alpha=0:30:180
```

```
    Edc=(3*sqrt(3)*Em/%pi)*cosd(alpha); // calculation of average voltage
    Idc=Edc/R_L; // calculation of average output current
    Isn13=abs((2*sqrt(2)*Idc/%pi)*cosd(30*n13)/n13);
    printf("\t%.3f\t",Isn13);
end
// plotting
// since to obtain a smooth graph alpha has been chosen as given below
alpha=[0:5:180]';
Edc=(3*sqrt(3)*Em/%pi)*cosd(alpha);
Idc=Edc/R_L;
Isn1=abs((2*sqrt(2)*Idc/%pi)*cosd(30*n1)/n1);
Isn5=abs((2*sqrt(2)*Idc/%pi)*cosd(30*5)/n5);
Isn7=abs((2*sqrt(2)*Idc/%pi)*cosd(30*n7)/n7);
Isn11=abs((2*sqrt(2)*Idc/%pi)*cosd(30*n11)/n11);
Isn13=abs((2*sqrt(2)*Idc/%pi)*cosd(30*n13)/n13);
plot2d(alpha,[Isn1 Isn5 Isn7 Isn11 Isn13],[1,2,3,4,5],leg="Isn1@Isn5@Isn7@Isn11@Isn11",nax=[2,7,2,10],rect=[0,0,180,5]);
xlabel("alpha (degrees)");
title("Fundamental and harmonic current with alpha");
// Note: The answers vary slightly due to precise calculations upto 6 decimal digit
```

Scilab code Exa 6.21 Determine average voltage

```
// chapter 6
// example 6.21
// Determine average voltage
// page-345
clear;
clc;
```

```
clf;
// given
Es=415; // in V (voltage supply)
f=50; // supply frequency
alpha=[0:30:120]; // in degree
// calculate
Em=Es*sqrt(2/3); // calculation of peak voltage
for alpha=[0:30:120];
    if alpha<=60 then
        Edc=(3*sqrt(3)*Em/%pi)*cosd(alpha); // calculation of average voltage for alpha <=60
   else
        Edc=(3*sqrt(3)*Em/%pi)*(1+cosd(alpha+60)); // calculation of average voltage for alpha >60
    end
    printf("\nFor alpha=%.f,\tEdc=%.1f",alpha,Edc);
end
// plot
alpha=[0:5:60];
Edc1=(3*sqrt(3)*Em/%pi)*cosd(alpha);
plot2d(alpha,Edc1);
alpha=[60:5:120];
Edc2=(3*sqrt(3)*Em/%pi)*(1+cosd(alpha+60));
plot2d(alpha,Edc2);
xlabel("alpha (degrees)");
ylabel("Edc (avg)  (in V)");
title("Average output voltage with alpha");
```

Scilab code Exa 6.22 Determine the minimum width of gate pulse

```
// chapter 6
// example 6.22
// Determine the minimum width of gate pulse
```

```
// page-345-346
clear;
clc;
// given
I_H=0.2; // in A (holding current)
td=2.5; // in us (delay time)
wt=%pi/3; // in radian (delay angle)
L1=8, L2=0; // in mH
R=2; // in ohm
E_line=208; // in V (line voltage)
f=50; // supply frequency
// calculate
L1=L1*1E-3;// changing unit from mH to H
td=td*1E-6;// changing unit from us to s
Ep=E_line/sqrt(3); // calculation of phase voltage
Em=Ep*sqrt(2); // calculation of peak voltage
e=sqrt(3)*Em*sin(wt+%pi/6);
di_dt_1=e/L1; // calculation of rate of rise of anode current when L=8 mH
// Since I_H=t1*(di_dt), therefore we get
t1=I_H/di_dt_1;// calculation of time required for anode current to rise to holding value when L=8 mH
tp1=t1+td; // calculation of minimum width of gate pulse when L=8 mH
//di_dt_2=e/L2; // rate of rise of anode current when L=0
t2=I_H/e*L2;// calculation of time required for anode current to rise to holding value when L=0
tp2=t2+td; // calculation of minimum width of gate pulse when L=0
printf("\nThe minimum width of gate pulse when L=8 mH is \t tp1=%.3f us",tp1*1E6);
printf("\nThe minimum width of gate pulse when L=0 H is \t tp2=%.1f us",tp2*1E6);
```

Scilab code Exa 6.23 Plot average voltage and load voltage

```
// chapter 6
// example 6.23
// Plot average voltage and load voltage
// page-356-357
clear;
clc;
clf;
// given
Es=415; // in V (voltage supply)
f=50; // supply frequency
alpha=[0:30:120]; // in degree
// calculate
Em=Es*sqrt(2/3); // calculation of peak voltage
// plot
alpha=[0:5:60];
Edc1=(3*sqrt(3)*Em/%pi)*cosd(alpha);
plot2d(alpha,Edc1);
alpha=[60:5:120];
Edc2=(3*sqrt(3)*Em/%pi)*(1+cosd(alpha+60));
plot2d(alpha,Edc2);
xlabel("alpha (degrees)");
ylabel("Edc (avg)  (in V)");
title("Average output voltage with alpha");
// since 3-phase waveform can not be plotted, so i
   have only plotted the average voltage
```

Scilab code Exa 6.24 Determine various parameters

```
// chapter 6
// example 6.24
```

```
// Determine various parameters
// page-358-359
clear;
clc;
// given
E_line=220; // in V (line voltage)
f=50; // supply frequency
R_L=10; // in ohm
ratio=25; // in percent (ratio of  given to maximum average output voltage)
alpha_min=0; // in radian (firing angle for max average output voltage)
// calculate
Ep=E_line/sqrt(3); // calculation of phase voltage
Em=Ep*sqrt(2); // calculation of peak voltage
Edc_max=(3*sqrt(3)*Em/(2*%pi))*(1+cos(alpha_min)); // calculation of maximum average output voltage
Edc=(ratio/100)*Edc_max; // calculation of average output voltage
// since Edc=(3*sqrt(3)*Em/(2*%pi)*(1+cos(alpha));, therefore we get
alpha=acosd((2*%pi/(3*sqrt(3)*Em))*Edc-1); // calculation of firing angle
printf("\nThe firing angle is \t\t\t %.f degree",alpha);
Idc=Edc/R_L; // calculation of average output current
alpha=alpha*(%pi/180); // changing unit from degree to radian
Erms=(3*Em/2)*sqrt((%pi-alpha+0.5*sin(2*alpha))/%pi); // calculation of rms voltage
Irms=Erms/R_L; // calculation of rms current
I_T_avg=Idc/3; // calculation of SCR average current
I_T_rms=Irms*sqrt(2)/3; // calculation of SCR rms current
neta=(Edc*Idc)/(Erms*Irms)*100; // calculation of Rectification efficiency
Is=Irms*sqrt(2/3);
```

```
Es=3*Ep;
TUF=(Edc*Idc)/(Es*Is); // calculation of Transformer Utilisation Factor
printf("\nThe average output current is \t\t Idc=%.2f A",Idc);
printf("\nThe rms current is \t\t\t Irms=%.2f A",Irms);
printf("\nThe thyristor average current is \t I_T_avg=%.2f A",I_T_avg);
printf("\nThe thyristor rms current is \t\t I_T_rms=%.2f A",I_T_rms);
printf("\nThe Rectification efficiency is \t neta=%.1f percent",neta);
printf("\nThe Transformer Utilisation Factor is \t TUF=%.3f",TUF);
// Note : The formula written in the book to calculate Erms and Is is wrong but the answer in the book is according to the correct formula. I have used the correct formula.
```

Scilab code Exa 6.25 Determine DPF DF HF and PF

```
// chapter 6
// example 6.25
// Determine DPF, DF, HF and PF
// page-361
clear;
clc;
// given
Es=415; // in V (supply voltage)
f=50; // supply frequency
alpha=0:30:180; // in degree
// calculate
// Since Is_n=(2*sqrt(2)*Idc/%pi)*(cosd(30*n)/n)=0.9*Idc*(cosd(30*n)/n)
```

```
// and Is_fund=0.9*Idc*cosd(30)=0.78*Idc and Is_rms=sqrt(2/3)*Idc,
// Since DF=Is_fund/Is_rms therefore we get
DF=(0.78/sqrt(2/3)); // calclation of distortion factor
HF=sqrt((1/DF)^2-1); // calclation of harmonic factor
printf("\nangle\t\tDPF\t\tDF\t\tHF\t\tPF");
for alpha=0:30:180
    DPF=cosd(alpha); // calclation of displacement power factor
    SPF=(3/%pi)*cosd(alpha); // calclation of supply power factor
    printf("\n%.f\t\t%.3f\t\t%.1f %%\t\t%.1f %%\t\t%.3f",alpha,DPF,DF*100,HF*100,SPF);
end
```

Scilab code Exa 6.26 Determine overlap angles

```
// chapter 6
// example 6.26
// fig. 6.72
// Determine overlap angles
// page-370-371
clear;
clc;
// given
Em=120; // in V
f=50; // supply frequency
L=0.33; // in mH
Id=4; // in A (load current)
alpha=15; // in degree (firing angle)
// calculate
L=L*1E-3; // changing unit from mH to H
// since Id=(Em/(2*%pi*f*L))*(1-cosd(u)) therefore
```

```
we get
u1=acosd(1-Id*(2*%pi*f*L/Em)); // calclation of overlap angle for case I
u2=acosd(cosd(alpha)-Id*(2*%pi*f*L/Em))-alpha; // calclation of overlap angle for case II
printf("\nThe overlap angle for transfer of current from a conducting thyristor to the commutating diode is \t u1=%.3f degree",u1);
printf("\nThe overlap angle for transfer of current from  the commutating diode to a thyristoris \t\t\t u2=%.3f degree",u2);
// Note: The answers for u1 and u2 vary slightly due to precise calculation upto 6 decimal digit
```

Scilab code Exa 6.27 Find the mean voltage at the load

```
// chapter 6
// example 6.27
// fig. Ex. 6.27
// Find the mean voltage at the load
// page-371-372
clear;
clc;
// given
E_line=415; // in V (line voltgae)
Beta=18; // in degree (advance angle)
u=3.8; // in degree (overlap angle)
// calculate
Em=E_line*sqrt(2/3); // calclation of peak voltage
Edc=(3*sqrt(3)/(4*%pi))*Em*(cosd(Beta)+cosd(Beta-u)); // calclation of mean voltage at the load
printf("\nThe mean voltage at the load is \t Edc=%.1f V",Edc);
```

Scilab code Exa 6.28 Compute source inductance load resistance and overlap angle

```
// chapter 6
// example 6.28
// Compute source inductance, load resistance and overlap angle
// page-371-372
clear;
clc;
// given
E=400; // in V (supply voltage)
f=50; // in Hz (supply frequency)
alpha=%pi/4; // in radian (firing angle)
Id=10; // in A (load current)
Edc=360; // in V (load voltage)
// calculate
alpha=alpha*(180/%pi);
disp(alpha);
Emph=E*sqrt(2/3);// calculation of peak voltage
// Since Edc=(3*sqrt(3)*Emph/%pi)*cosd(alpha)-(3*2*%pi*f*Ls/%pi)*Id, we get
Ls=(%pi/(3*2*%pi*f*Id))*((3*sqrt(3)*Emph/%pi)*cosd(alpha)-Edc);// calculation of source inductance
R=Edc/Id;// calculation of load resistance
// since Edc=(3*sqrt(3)*Emph/%pi)*cosd(alpha+u)-(3*2*%pi*f*Ls/%pi)*Id, we get
u=acosd((%pi/(3*sqrt(3)*Emph))*(Edc-(3*2*%pi*f*Ls/%pi)*Id))-alpha;// calculation of overlap angle
printf(”\nThe source inductance is \t Ls=%.1f mH”,Ls*1E3);
printf(”\nThe load resistance is \t\t R=%.1f ohm”,R);
printf(”\nThe overlap angle is \t\t u=%.f degree”,u)
```

```
;
```

Scilab code Exa 6.29 Compute the average load voltage

```
// chapter 6
// example 6.29
// Compute the average load voltage
// page-372-373
clear;
clc;
// given
Eph=150; // in V (supply voltage per phase)
f=50; // in Hz (supply frequency)
Ls=1.2; // in mH (source inductance)
R=0.07; // in ohm
V_drop_Thyristor=1.5; // in V (voltage drop across
   Thyristor)
Id=30; // in A (continuous load current)
alpha=0:30:60; // in degree (firing angles)
// calculate
Ls=Ls*1E-3; // changing unit from mH to H
V_drop_reactance=(3*2*%pi*f*Ls/(2*%pi))*Id; //
   voltage drop due to source reactance
Em=Eph*sqrt(2);// calculation of peak voltage
for alpha=0:30:60
    Edc=((3*sqrt(3)/(2*%pi))*Em*cosd(alpha))-
       V_drop_Thyristor-V_drop_reactance;;//
       calculation of average load voltage
    printf("\nFor %.f degree, the average load
       voltage is Edc=%.1f V",alpha,Edc);
end
// Note: The answers vary slightly due to precise
   calculation upto 6 decimal digits
```

Scilab code Exa 6.30 Compute the average generator voltage

```
// chapter 6
// example 6.30
// Compute the average generator voltage
// page-373
clear;
clc;
// given
E=415; // in V (supply voltage per phase)
f=50; // in Hz (supply frequency)
X_L=0.3; // in ohm (source reactance)
R=0.05; // in ohm (resistance per phase)
V_drop_Thyristor=1.5; // in V (voltage drop across Thyristor)
Id=60; // in A (continuous load current)
Beta=35; // in degree (firing advance angle)
u=0; // in degree (overlap angle at no load)
// calculate
V_drop_reactance=(3*X_L/%pi)*Id; // voltage drop due to overlap
V_drop_Thyristors=2*V_drop_Thyristor; // voltage drop due to SCRs
V_drop_resistance=2*R*Id; // voltage drop due to supply resistance
Emph=E*sqrt(2/3);// calculation of peak voltage
Edc_noload=-((3*sqrt(3)/%pi)*Emph*cosd(u-Beta));// calculation of average voltage at no load
Edc=abs(Edc_noload-V_drop_Thyristors-V_drop_reactance-V_drop_resistance);// calculation of average generator voltage
printf("\nThe average generator voltage is Edc=%.2f V",Edc);
// Note: The answers vary slightly due to precise
```

```
calculation upto 6 decimal digits
```

Scilab code Exa 6.32 Compute the firing angle and overlap angle

```
// chapter 6
// example 6.32
// Compute the firing angle and overlap angle
// page-373-375
clear;
clc;
// given
Eph=230; // in V (supply voltage per phase)
f=50; // in Hz (supply frequency)
Ls=0.3; // in mH (source inductance)
Id=15; // in A (load current)
R=1; // onm (internal resistance)
E_loadsource=400; // in V (dc source voltage)
// calculate
Ls=Ls*1E-3; // changing unit from mH to H
Edc=E_loadsource+Id*R;// calculation of average voltage
Emph=Eph*sqrt(2);// calculation of peak voltage
// Since Edc=(3*sqrt(3)/%pi)*Emph*cosd(alpha)-(3*2*%pi*f*Ls/%pi)*Id, therefore we get
alpha=acosd((%pi/(3*sqrt(3)*Emph))*(Edc+(3*2*%pi*f*Ls/%pi)*Id));// calculation of firing angle
// Since Edc=(3*sqrt(3)/%pi)*Emph*cosd(alpha+u)+(3*2*%pi*f*Ls/%pi)*Id, therefore we get
u=acosd(%pi/(3*sqrt(3)*Emph)*(Edc-(3*2*%pi*f*Ls/%pi)*Id))-alpha;// calculation of overlap angle
printf(”\nThe firing angle is \t%.2f degree”,alpha);
printf(”\nThe overlap angle is \t%.3f degree”,u);
// Note: The answers vary slightly due to precise calculation upto 6 decimal digits.In the book, the calculation is done upto 2 decimal digit
```

Scilab code Exa 6.33 Design the 3 phase bridge convertor

```
// chapter 6
// example 6.33
// Design the 3-phase bridge convertor
// page-375-377
clear;
clc;
// given
E=2.3; // in kV (supply voltage)
n=10; // delta-star transformer ratio
f=50; // in Hz (supply frequency)
Id=90; // in A (load current)
Edc1=500, Edc2=-500; // in V (range of varying DC voltage)
Ls=50; // in uH (commutating inductance per phase)
V_T=1.5; // in V (assumption for voltage drop across SCR as done in the book)
Tc_max=104; // in degree C (assumption for maximum temperature as done in the book)
T_A=25; // in degree C (assumption as done in the book)
theta_CS=0.675; // in dgree C/W (assumption as done in the book)
Pav=43; // in W (assumption  as done in the book)
// calculate
Ls=Ls*1E-6; //changing unit from uH to H
E=E*1E3; //changing unit from kV to V
Edc_red=(3*2*%pi*f*Ls/%pi)*Id;// calculation of reduction in output voltage due to overlap
Edc=Edc1+Edc_red+2*V_T;// calculation of average output voltage
E_L=(E/n)*sqrt(3);// calculation of line rms voltage
// Since Edc=1.35*E_L*cosd(alpha), therefore we get
```

```
alpha=acosd(Edc/(1.35*E_L));// calculation of firing angle
E_DRM=E_L*sqrt(2)*V_T;// calculation of voltage ratings of SCR
ERRM=E_DRM;// calculation of voltage ratings of SCR
Iavg=Id/3;// calculation of average output current
Irms=Id/sqrt(3);// calculation of rms current
printf("\nThe average output voltage is \t\tEdc=%.2f V",Edc);
printf("\nThe line rms voltage is \t\tE_L=%.1f V",E_L);
printf("\nThe firing angle is \t\t\t%.1f degree",alpha);
printf("\nThe minimum voltgae rating of SCR are \tE_DRM=E_RRM=%.f V",E_DRM);
printf("\nThe average output current is \t\tIavg=%.f A",Iavg);
printf("\nThe rms current is \t\t\tIrms=%.2f A",Irms);
printf("\n\nTherefore we can select SCRs with rating 900 V and 63 A (rms)");
Edc_2=(Edc2+Edc_red+2*V_T);// calculation of average output voltage in inverting mode. Here minus sign indicates inverting mode
PF=Edc_2/(1.35*E_L);// calculation of cos(alpha) in inverting mode
// since cosd(alpha+u)=cosd(alpha)-(2*%pi*f*Ls*Id*sqrt(3/2)/E_L), therefore we get
alpha_mu=acosd(PF-(2*%pi*f*Ls*Id*sqrt(3/2)/E_L));// calculation of (alpha+mu)
gama=180-alpha_mu;// calculation of gama
t_off=gama/(2*%pi*f*(180/%pi));// calculation of off time
printf("\n\nThe off time is \t t_off=%.3f ms",t_off*1E3);
theta_CA=(Tc_max-T_A)/Pav;
// since theta_CA=theta_CS+theta_SA, therefore we get
```

```
theta_SA=theta_CA-theta_CS;
printf(”\n\nThe heat sink parameters is \t theta_SA=%.2f degree C/W”, theta_SA);
Irms_secondary=sqrt(2/3)*Id;// calculation of rms secondary current
Irms_primary=sqrt(2/3)*Id/n;// calculation of rms primary current
VA_rating=3*E_L*Irms_secondary/sqrt(3);// calculation of tranformer VA rating
I_Lf=(3/%pi)*(2*Id/n)*(1/sqrt(2));// calculation of lin rms fundamental current
I_L=sqrt(2)*Id/n;// calculation of rms current
printf(”\n\nThe rms secondary current is \t\t%.2f A”,Irms_secondary);
printf(”\nThe rms primary current is \t\t%.2f A”,Irms_primary);
printf(”\nThe tranformer VA rating is \t\t%.1f kVA”,VA_rating*1E-3);
printf(”\nThe line rms fundamental current is \tI_Lf=%.2f A”,I_Lf);
printf(”\nThe rms current is \t\t\tI_L=%.2f A”,I_L);
DPF=cosd(alpha);// calculation of displacement power factor
DF=I_Lf/I_L;// calculation of distortion factor
PF=DPF*DF; // calculation of supply power factor
Pi=sqrt(3)*E*I_Lf*DPF; // calculation of fundamental input power
Po=Edc1*Id; // calculation of output power
neta=(Po/Pi)*100; // calculation of efficiency
printf(”\n\nThe displacement power factor is \tDPF=%.2f”,DPF);
printf(”\nThe distortion factor is \t\tDF=%.2f”,DF);
printf(”\nThe power factor is \t\t\tPF=%.2f ”,PF);
printf(”\nThe efficiency is \t\t\t%.1f percent”,neta);
// Note: The answers vary slightly due to precise calculation upto 6 decimal digits.In the book, the calculation is done upto 2 decimal digit
```

Scilab code Exa 6.34 Calculate delay angle power delivered and fundamental reactive power

```
// chapter 6
// example 6.34
// Calculate delay angle, power delivered and
   fundamental reactive power
// page-377-378
clear;
clc;
// given
E_line=580; // in V (line to line supply voltage)
f=50; // in Hz (supply frequency)
I0=3464; // in A (load current)
Edc=648; // in V (average output voltage)
// calculate
Em=sqrt(2/3)*E_line;
// since Edc=(3*sqrt(3)/%pi)*Em*cosd(alpha),
   therefore we get
alpha=acosd(%pi/(3*sqrt(3)*Em)*Edc);
Pdc=Edc*I0;
Pdc=Pdc*1E-3;
Pac=1.05*Pdc;
Q_KVAR=sqrt(Pac^2-Pdc^2);
printf("\nThe delay angle is \t\t\t\t\t\t\t %.2f
   degree",alpha);
printf("\nThe power delivered to the load is \t\t\t\
   t\t Pdc=%.2f kW",Pdc);
printf("\nThe fundamental reactive power drawn from
   the supply in KVAR is \t %.2f",Q_KVAR);
// Note: The answers vary slightly due to precise
   calculation upto 6 decimal digits.In the book,
   the calculation is done upto 2 decimal digit
```

Scilab code Exa 6.35 Calculate reducation in dc output voltage and overlap angle

```
// chapter 6
// exapmle 6.35
// Calculate reducation in dc output voltage and
   overlap angle
// page-378
clear;
clc;
// given
E_line=400; // in V (line to line supply voltage)
alpha=30; // in degree (control angle)
r=15; // in degree (overlap angle)
alpha_inv_mode=120; // in degree (control angle in
   inverting mode)
P=6; // number of pulses
// calculate
Emax=sqrt(2/3)*E_line;// calculation of peak voltage
Edc_with_ovelap=(P/(2*%pi))*Emax*sin(%pi/P)*(cosd(
   alpha)+cosd(alpha+r));// calculation of average
   output voltage with overlap
Edc_without_overlap=(P/%pi)*Emax*(%pi/P)*cosd(alpha)
   ;// calculation of average output voltage without
    overlap
Edc_drop=Edc_without_overlap-Edc_with_ovelap;//
   calculation of reducation in dc output voltage
Beta=180-alpha_inv_mode;// calculation of beta
r=Beta-acosd((25/(Emax*sin(%pi/P)))-cosd(Beta));//
   calculation of overlpa angle
printf("\nThe reducation in dc output voltage due to
    overlap is \t %.2f V",Edc_drop);
printf("\nThe overlap angle in the inverting mode is
    \t\t r= %.f degree",r);
```

```
// Note: The answers vary slightly due to precise calculation upto 6 decimal digits.In the book, the calculation is done upto 2 decimal digit
```

Scilab code Exa 6.36 Calculate various parameters

```
// chapter 6
// example 6.36
// Calculate various parameters
// page-393-395
clear;
clc;
// given
E=230; // in V (supply voltage)
f=50; // in Hz (supply frequency)
Ia=10; // in A
Edc=175; // in V (average output voltage)
alpha1=33, alpha2=78; // in degree
Beta1=62; // in degree
// calculate
Em=sqrt(2)*E;// calculation of peak voltage
// EAC control
// Since Edc=(Em/%pi)*(1-cosd(Beta)), therefore we get
Beta=acosd(1-(%pi/Em)*Edc);// calculation of extinction angle
Is_rms=Ia*sqrt(Beta/180);// calculation of rms supply current
Is_fund=(2*sqrt(2)*Ia/%pi)*sind(Beta/2);// calculation of rms fundamental current
DPF=sind(Beta/2);// calculation of displacement factor
DF=Is_fund/Is_rms;// calculation of distortion factor
HF=sqrt(1/DF^2-1);// calculation of harmonic factor
```

```
PF=DF*DPF;// calculation of power factor
printf("\nEAC Control");
printf("\n\t\tThe extinction angle is \t Beta=%.2f degree",Beta);
printf("\n\t\tThe rms supply current is \t Is_rms=%.2f A",Is_rms);
printf("\n\t\tThe rms fundamental current is \t Is_fund=%.2f A",Is_fund);
printf("\n\t\tThe displacement factor is \t DPF=%.2f",DPF);
if DPF>=0 then
    printf(" (lagging)");
else
    printf(" (leading)");
end
printf("\n\t\tThe distortion factor is \t DF=%.2f percent",DF*100);
printf("\n\t\tThe harmonic factor is \t\t HF=%.2f percent",HF*100);
printf("\n\t\tThe power factor is \t\t PF=%.4f",PF);
if PF>=0 then
    printf(" (lagging)");
else
    printf(" (leading)");
end
// SAC control
// Since Edc=(2*Em/%pi)*cosd(alpha), therefore we get
alpha=acosd(%pi/(2*Em)*Edc);// calculation of trigger angle
Is_rms=Ia*sqrt(1-(2*alpha/180));// calculation of rms supply current
Is_fund=(2*sqrt(2)*Ia/%pi)*cosd(alpha);// calculation of rms fundamental current
DPF=1;// calculation of displacement factor
DF=Is_fund/Is_rms;// calculation of distortion factor
HF=sqrt(1/DF^2-1);// calculation of harmonic factor
```

```
PF=DF*DPF;// calculation of power factor
printf(”\n\nSAC Control”);
printf(”\n\t\tThe trigger angle is \t\t alpha=%.2f degree”,alpha);
printf(”\n\t\tThe rms supply current is \t Is_rms=%.f A”,Is_rms);
printf(”\n\t\tThe rms fundamental current is \t Is_fund=%.1f A”,Is_fund);
printf(”\n\t\tThe displacement factor is \t DPF=%.f”,DPF);
printf(”\n\t\tThe distortion factor is \t DF=%.f percent”,DF*100);
printf(”\n\t\tThe harmonic factor is \t\t HF=%.1f percent”,HF*100);
printf(”\n\t\tThe power factor is \t\t PF=%.2f”,PF);
// PWM control
Beta3=180-alpha1;
Beta2=180-alpha2;
alpha3=180-Beta1;
Edc=(Em/%pi)*(cosd(alpha1)-cosd(Beta1)+cosd(alpha2)-cosd(Beta2)+cosd(alpha3)-cosd(Beta3));// calculation of average output voltage
Is_rms=(Ia/sqrt(180))*sqrt(Beta1-alpha1+Beta2-alpha2+Beta3-alpha3);// calculation of rms supply current
Is_fund=(sqrt(2)*Ia/%pi)*(cosd(alpha1)-cosd(Beta1)+cosd(alpha2)-cosd(Beta2)+cosd(alpha3)-cosd(Beta3));// calculation of rms fundamental current
DPF=1;// calculation of displacement factor
DF=Is_fund/Is_rms;// calculation of distortion factor
HF=sqrt(1/DF^2-1);// calculation of harmonic factor
PF=DF*DPF;// calculation of power factor
printf(”\n\nPWM Control”);
printf(”\n\t\tThe average output voltage is \t Edc=%.2f V”,Edc);
printf(”\n\t\tThe rms supply current is \t Is_rms=%.2f A”,Is_rms);
```

```
printf("\n\t\tThe rms fundamental current is \t Is_fund=%.1f A",Is_fund);
printf("\n\t\tThe displacement factor is \t DPF=%.f",DPF);
printf("\n\t\tThe distortion factor is \t DF=%.f percent",DF*100);
printf("\n\t\tThe harmonic factor is \t\t HF=%.2f percent",HF*100);
printf("\n\t\tThe power factor is \t\t PF=%.2f",PF);
// Note: The answers vary slightly due to precise calculation upto 6 decimal digits.In the book, the calculation is done upto 2 decimal digit
```

Chapter 7

Dual Converters

Scilab code Exa 7.1 Compute the peak value of circulating current

```
// chapter 7
// example 7.1
// Compute the peak value of circulating current
// page-431
clear;
clc;
// given
Erms=230; // in V (per phase voltage supply)
w=315; // in rad/s
L=12; // in mH
alpha1=60, alpha2=120; // in degree
// calculate
L=L*1E-3; // changing unit from mH to H
i_cp=(3*sqrt(2)*Erms/(w*L))*(1-cosd(30)); // calculation of peak value of circulating current
printf("\nThe peak value of circulating current is \t i_cp=%.2f A",i_cp);
// Note: The answer varies slightly due to precise calculation
```

Scilab code Exa 7.2 Design dual convertor

```
// chapter 7
// example 7.2
// Design dual convertor
// page-432
clear;
clc;
// given
Ea=220; // in V
Ia=30; // in A
N=1500; // in rpm (speed of motor)
Eac=400; // in V (3-phase supply)
f=50; // in Hz (supply frequency)
drop=15; // in percent (drop in the circuit)
// calculate
E_drop=(drop/100)*Ea;// calculation of total drop in the system
Edc_alpha=Ea+E_drop;// calculation of total dc voltage
// since Edc_alpha=1.35*Eac*cosd(alpha), therefore we get
alpha1=acosd(Edc_alpha/(1.35*Eac));// calculation of firing angle
Iac=0.817*Ia;// calculation of AC line current
Pac=sqrt(3)*Eac*Iac;// calculation of AC terminal power
// since Pac=1.35*Pdc, therefore we get
Pdc=Pac/1.35;// calculation of DC average power
I_ripple=Ia/5;// calculation of ripple current
w=2*%pi*f;// calculation of angular velocity
Lc=(2*1.35*Eac/(6*w*I_ripple))*(1/7+1/5);// calculation of current limiting inductance
alpha2=180-alpha1;// calculation of firing angle
```

```
PIV=2*sqrt(2)*Eac;// calculation of peak inverse voltage of SCR
I_T=2*sqrt(2)*Iac;// calculation of current rating of SCR
printf("\nThe total drop in the system is \t E_drop=%.f V",E_drop);
printf("\nThe total dc voltage is \t\t Edc_alpha=%.f V",Edc_alpha);
printf("\nThe firing angle is \t\t\t alpha1=%.f degree",alpha1);
printf("\nThe AC line current is \t\t\t Iac=%.2f A",Iac);
printf("\nThe AC terminal power is \t\t Pac=%.2f kW",Pac*1E-3);
printf("\nThe DC average power is \t\t Pdc=%.2f kW",Pdc*1E-3);
printf("\nThe current limiting inductance is \t Lc=%.f mH",Lc*1E3);
printf("\nThe firing angle is \t\t\t alpha2=%.f degree",alpha2);
printf("\nThe peak inverse voltage of SCR is \t PIV=%.f V",PIV);
printf("\nThe current rating of SCR is \t\t I_T=%.f A",I_T);
```

Chapter 8

Choppers

Scilab code Exa 8.1 Determine time ratio of the chopper

```
// chapter 8
// example 8.1
// Determine time ratio of the chopper
// page-439
clear;
clc;
// given
Edc=100; // in V
L=40; // in mH
R=5; // in ohm
I1=10, I2=12; // in A (current limits)
// calculate
Iavg=(I1+I2)/2;// calculation of average value of
    load current
Imax=Edc/R;// calculation of maximum value of load
    current
E0_av=Edc*(Iavg/Imax);// calculation of average
    value of voltage
// since Ton/(Ton+Toff)=E0_av/Edc, therefore we get
Ton_Toff=E0_av/(Edc-E0_av);// calculation of time
    ratio of the chopper
```

```
printf("\nThe time ratio of the chopper is \t Ton/
    Toff=%.3f ",Ton_Toff);
```

Scilab code Exa 8.3 Determine average and rms output voltage chopper efficiency and effective input resistance

```
// chapter 8
// example 8.3
// Determine average and rms output voltage, chopper
    efficiency and effective input resistance
// page-440-441
clear;
clc;
// given
Edc=200; // in V (input voltage)
R=15; // in ohm
Ed=2.5; // in V (voltage drop when the chopper is ON
    )
f=1; // in KHz (chopper frequency)
alpha=50; // in percent (duty cycle)
// calculate
f=f*1E3;
T=1/f;// calculation of time period of chopper
E0=(alpha/100)*(Edc-Ed);// calculation of average
    output voltage
E0_rms=sqrt(alpha/100)*(Edc-Ed);// calculation of
    rms output voltage
I0_rms=E0_rms/R;// calculation of rms output current
Po=E0_rms*I0_rms;// calculation of output power
Is=(Edc-Ed)/R;// calculation of average output
    current
Pi=(1/T)*Edc*Is*integrate('1','t',0,((alpha/100)*T))
    ;// calculation of input power
neta=Po/Pi;// calculation of chopper efficiency
printf("\nThe average output voltage is \t E0=%.2f V
```

```
",E0);
printf("\nThe rms output voltage is \t E0_rms=%.3f V",E0_rms);
printf("\nThe chopper efficiency is \t neta=%.2f percent",neta*100);
```

Scilab code Exa 8.4 Calculate the required pulse width

```
// chapter 8
// example 8.4
// Calculate the required pulse width
// page-442
clear;
clc;
// given
Edc=220; // in V (dc source)
E0=500; // in V (load voltage)
Toff=80; // in us (blocking period of thyrostor)
// calculate
// since E0=Edc*(Ton+Toff)/Toff, therefore we get
Ton=(E0/Edc)*Toff-Toff;// calculation of required pulse width
printf("\nThe time required pulse width is \t Ton=%.1f us",Ton);
// Note: the answer varies slightly due to precise calculation
```

Scilab code Exa 8.5 Calculate the conduction and blocking period of the thyristor

```
// chapter 8
// example 8.5
```

```
// Calculate the conduction and blocking period of the thyristor
// page-446
clear;
clc;
// given
Edc=220; // in V (dc source)
f=2; // in Khz (chopper frequency)
E0=170; // in V (load voltage)
// calculate
f=f*1E3;
// since E0=Edc*Ton*f, therefore we get
Ton=E0/(Edc*f);// calculation of conduction period of the thyristor
T=1/f; // calculation of chopping period
Toff=T-Ton; // calculation of  blocking period of the thyristor
printf("\nThe conduction period of the thyristor is \t Ton=%.3f s",Ton*1E3);
printf("\nThe blocking period of the thyristor is \t Toff=%.3f s",Toff*1E3);
// Note: the answer varies slightly due to precise calculation
```

Scilab code Exa 8.6 Calculate limit of current pulsation chopping frequency duty cycle and output voltage

```
// chapter 8
// example 8.6
// Calculate limit of current pulsation, chopping frequency, duty cycle and output voltage
// page-446
clear;
clc;
// given
```

```
Edc=110; // in V (dc source)
Ton=15; // in ms (conduction period of the thyristor)
Toff=12; // in ms (blocking period of the thyristor)
I0_max=300; // in A (maximum possible accelerating current)
I0_min=140; // in A (lower limit of current plusation)
// calculate
limit=I0_max-I0_min; // calculation of limit of current pulsation
T=Ton+Toff; // calculation of chopping period
f=1/(T*1E-3); // calculation of chopping frequency
alpha=Ton/T; // calculation of duty cycle
E0=alpha*Edc; // calculation of output voltage
printf("\nThe limit of current pulsation is \t %.f A",limit);
printf("\nThe chopping frequency is \t\t f=%.f Hz",f);
printf("\nThe duty cycle is \t\t\t alpha=%.2f ",alpha);
printf("\nThe output voltage is \t\t\t E0=%.2f V",E0);
// Note: the answer varies slightly due to precise calculation
```

Scilab code Exa 8.7 Compute following quantities

```
// chapter 8
// example 8.7
// Compute following quantities
// whether load current is continuous, Average output current, maximum & minimum values of steady-state output current, rms values of 1st, 2nd, 3rd harmonics of load current, Average value
```

```
   of source current, input power, power absorbed by
    the back emf Eb and power loss in the reistor,
   RMS value of uotput current
// page-459-460
clear;
clc;
// given
Edc=220; // in V (dc source)
f=500; // in Hz (chopping frequency)
alpha=0.3; // duty cycle
R=1; // in ohm
L=3; // in mH
Eb=23; // in V (back emf)
// calculate
L=L*1E-3; // changing unit from mH to H
tou=L/R;
T=1/f;
g=Eb/Edc;
w=2*%pi*f;
alpha_1=(tou/T)*log(1+g*(exp(T/tou)-1));
if alpha>alpha_1 then
    printf("\nThe load current is continuous.");
else
    printf("\nThe load current is discontinuous.");
end
I0_avg=(alpha*Edc-Eb)/R;
printf("\n\nThe average output current is \t\t\t\t
   I0_avg=%.f A",I0_avg);
// Since alpha=Ton/T, therefore we get
Ton=alpha*T;
I0_max=(Edc/R)*((1-exp(-Ton/tou))/(1-exp(-T/tou)))-
   Eb/R;
printf("\n\nThe maximum steady-state output current
   is \t\t I0_max=%.2f A",I0_max);
I0_min=(Edc/R)*((exp(Ton/tou)-1)/(exp(T/tou)-1))-Eb/
   R;
printf("\n\nThe minimum steady-state output current
   is \t\t I0_min=%.2f A",I0_min);
```

```
Value=0; // a temporary variable to calculate few variables
for n=1:3
    Z=sqrt(R^2+(w*n*L)^2);
    I=(2*Edc*sin(n*%pi*alpha))/(n*sqrt(2)*%pi)/Z;
    printf(”\n\nThe %.fth harmonic of steady-state output current is \t I_%.f=%.3f A”,n,n,I);
    Value=Value+I^2;
end
I_TAV=(Edc-Eb)*alpha/R-(L/(R*T))*(I0_max-I0_min);
printf(”\n\nThe average value of source current is \t\t\t I_TAV=%.f3 A”,I_TAV);
Pin=Edc*I_TAV;
P_absorbed=Eb*I0_avg;
P_loss=Pin-P_absorbed;
printf(”\n\nThe input power is \t\t\t\t\t Pin=%.1f W”,Pin);
printf(”\n\nThe power absorbed by load emf is \t\t\t P_absorbed=%.f W”,P_absorbed);
printf(”\n\nThe power loss in the resistor is \t\t\t P_loss=%.1f W”,P_loss);
Irms_2_3=sqrt(I0_avg^2+Value);
printf(”\n\nThe rms current using the result of (ii) and (iii) is \t Irms=%.2f A”,Irms_2_3);
Irms_4=sqrt(P_loss/R);
printf(”\n\nThe rms current using the result of (iv) is \t\t Irms=%.2f A”,Irms_4);
// Note: the answer varies slightly due to precise calculation
```

Scilab code Exa 8.8 Calculate the mean values of load voltage and current

```
// chapter 8
// example 8.8
// Calculate the mean values of load voltage and
```

```
   current
// page-460-462
clear;
clc;
// given
Edc=80; // in V (dc source)
T=2; // in ms (chopping period)
L=8; // in mH
R=4; // in ohm
Ton_Toff_1= 1/1;
Ton_Toff_2= 4/1;
Ton_Toff_3= 1/4;
// calculate
L=L*1E-3; // changing unit from mH to H
tou=L/R;
T=T*1E-3;
//part-a
// since Ton/Toff=1/1 therefore Ton=Toff or
Ton=T/2;
Toff=Ton;
I0_max=(Edc/R)*((1-exp(-Ton/tou))/(1-exp(-T/tou)));
I0_min=I0_max*exp(-Toff/tou);
E0=Edc*(Ton/T);
I0_av=E0/R;
printf("\nWhent Ton/Toff=1/1,")
printf("\n\t\tThe maximum output current is \t\t I0_max=%.2f A", I0_max);
printf("\n\t\tThe minimum output current is \t\t I0_min=%.2f A", I0_min);
printf("\n\t\tThe mean value of load voltage is \t E0=%.f V",E0);
printf("\n\t\tThe mean value of load current is \t I0_av=%.f A",I0_av);
//part-b
// since Ton/Toff=4/1 therefore Ton=4Toff and Ton+ Toff=T or 5*Toff=T therefore we get
Toff=T/5;
Ton=4*Toff;
```

```
I0_max=(Edc/R)*((1-exp(-Ton/tou))/(1-exp(-T/tou)));
I0_min=I0_max*exp(-Toff/tou);
E0=Edc*(Ton/T);
I0_av=E0/R;
printf("\nWhent Ton/Toff=4/1,")
printf("\n\t\tThe maximum output current is \t\t I0_max=%.2f A", I0_max);
printf("\n\t\tThe minimum output current is \t\t I0_min=%.2f A", I0_min);
printf("\n\t\tThe mean value of load voltage is \t E0=%.f V",E0);
printf("\n\t\tThe mean value of load current is \t I0_av=%.f A",I0_av);
//part-c
// since Ton/Toff=1/4 therefore Toff=4*Toff and Ton+Toff=T or 5*Ton=T therefore we get
Ton=T/5;
Toff=5*Ton;
I0_max=(Edc/R)*((1-exp(-Ton/tou))/(1-exp(-T/tou)));
I0_min=I0_max*exp(-Toff/tou);
E0=Edc*(Ton/T);
I0_av=E0/R;
printf("\nWhent Ton/Toff=1/4,")
printf("\n\t\tThe maximum output current is \t\t I0_max=%.2f A", I0_max);
printf("\n\t\tThe minimum output current is \t\t I0_min=%.2f A", I0_min);
printf("\n\t\tThe mean value of load voltage is \t E0=%.f V",E0);
printf("\n\t\tThe mean value of load current is \t I0_av=%.f A",I0_av);

// Note: 1 The answer for I0_min of part-b is wrong in the book due to calculation mistake.
//       2.the answers varies slightly due to precise calculation
```

Scilab code Exa 8.9 Find the chopping frequency

```
// chapter 8
// example 8.9
// fig. Ex. 8.9
// Find the chopping frequency
// page-463
clear;
clc;
// given
Edc=400; // in V (dc source)
L=0.05; // in H
R=0; // in ohm
alpha=0.3; // duty cycle
dI0=8; // in A (load current excursion)
// calculate
E0=alpha*Edc;
Eb=E0;
// Since VT_area=L*dI0 and VT_area=(Edc-Eb)*Ton, therefore we get
// (Edc-Eb)*Ton=L*dI0 or
Ton=L*dI0/(Edc-Eb); // calculation of ON period of chopper
// Since alpha=Ton/T, therefore
T=Ton/alpha; // calculation of chopping period
f=1/T; // calculation of chopping frequency
printf("\nThe chopping frequency to limit the load current excursion to %.f A is \t f=%.2f Hz",dI0,f);
// Note: The answer varies slightly due to precise calculation
```

Scilab code Exa 8.10 Find the on period average load current magnitude of ripple current and its rms value

```
// chapter 8
// example 8.10
// Find the on period, average load current, magnitude of ripple current and its rms value
// page-464
clear;
clc;
// given
f=2; // in Khz (chopping frequency)
Edc=96; // in V (dc source)
R=8; // in ohm
tou=6; // in ms (time constant)
E0=57.6; // in V (average load voltage)
// calculate
f=f*1E3; // changing unit from Khz to Hz
tou=tou*1E-3; // changing unit from ms to s
T=1/f; // calculation of chopping period
// Since tou=L/R, therefore we get
L=tou*R; // calculation of Inductance
// Since Eo=Edc*(Ton/T), therefore we get
Ton=(E0/Edc)*T; // calculation of ON period of chopper
Toff=T-Ton; // calculation of OFF period of chopper
F_L_rms=Edc*sqrt(Ton/T); // calculation of rms value of load voltage
I0=E0/R; // calculation of average load current
del_i=(Edc-E0)*Ton/L; // calculation of current ripple
I0_max=(Edc/R)*((1-exp(-Ton/tou))/(1-exp(-T/tou))); // calculation of maximum output current
I0_min=I0_max*exp(-Toff/tou); // calculation of minimum output current
Ir_rms=(I0_max-I0_min)/(2*sqrt(3)); // calculation of rms value of ripple current
printf("\nThe ON period of chopper is \t\t Ton=%.1f
```

```
   ms",Ton*1E3);
printf("\nThe rms value of load voltage is \t %.2f V
   ",F_L_rms);
printf("\nThe average load current is \t\t I0=%.1f A
   ",I0);
printf("\nThe magnitude of ripple current is \t %.2f
    A",del_i);
printf("\nThe maximum load current is \t\t I0_max=%
   .2f A",I0_max);
printf("\nThe minimum load current is \t\t I0_min=%
   .2f A",I0_min);
printf("\nThe RMS value of ripple current is \t
   Ir_rms=%.4f A",Ir_rms);
// Note: The answer varies slightly due to precise
   calculation
```

Scilab code Exa 8.11 Compute the current in the motor

```
// chapter 8
// example 8.11
// Compute the current in the motor
// page-464-465
clear;
clc;
// given
Edc=180; // in V (dc source)
Ra=0.4; // in ohm(armature resistance)
Eb=80; // in V (back emf)
i_a=80; // in A(armature current)
La=8; // in mH (armature inductance)
t=8; // in ms
t_off=1; // in ms
// calculate
La=La*1E-3; // changing unit from mH to H
tou=La/Ra; // calculation of time constant
```

```
t_off=t_off*1E-3; // changing unit from ms to s
t=t*1E-3; // changing unit from ms to s
i_t=((Edc-Eb)/Ra)*(1-exp(-t_off/tou))+i_a*exp(-t_off/tou);
t8=t_off+t; // time instant 8ms after SCR turns off
i_f=i_t*exp(-t8/tou);
printf(”\nThe current at the instant of Thyristor turns OFF is \t %.3f A”,i_t);
printf(”\nThe current %.f ms after Thyristor turns OFF is \t %.2f A”,t*1E3,i_f);
// Note: The value of tou has been calculated wrong in the book due to which answer in the book are wrong
```

Scilab code Exa 8.12 Determine the ON time of the chopper

```
// chapter 8
// example 8.12
// Determine the ON time of the chopper
// page-465
clear;
clc;
// given
Edc=230; // in V (dc source)
F=400; // in Hz (chopping frequency)
E0=150; // in V
// calculate
// Siince E0=alpha*Edc, therefore we get,
alpha=E0/Edc;
T=1/F;
// Since alpha=Ton/T, therefore we get,
Ton=alpha*T;
printf(”\nThe ON time of chopper is \t Ton=%.4f ms”,Ton*1E3);
// Note: The answer varies slightly due to precise
```

```
   calculation upto 6 decimal digit
```

Scilab code Exa 8.13 Determine average and rms output voltage and form and ripple factor

```
// chapter 8
// example 8.13
// Determine average and rms output voltage and form
    and ripple factor
// page-465
clear;
clc;
// given
Edc=220; // in V (dc source)
Ton=1; // in ms (ON period of chopper)
Toff=1.5; // in ms (OFF period of chopper)
// calculate
T=Ton+Toff; // calculation of chopping period
alpha=Ton/T; // calculation of duty cycle
E0=alpha*Edc; // calculation of average output
   voltage
E0_rms=sqrt(alpha)*Edc; // calculation of rms output
    voltage
FF=E0_rms/E0; // calculation of form factor
RF=sqrt(FF^2-1); // calculation of ripple factor
printf("\nThe average output voltage is \t E0=%.f V"
   ,E0);
printf("\nThe rms output voltage is \t E0_rms=%.2f V
   ",E0_rms);
printf("\nThe form factor is \t\t FF=%.2f ",FF);
printf("\nThe ripple factor is \t\t RF=%.2f ",RF);

// Note: The answer varies slightly due to precise
   calculation upto 6 decimal digit
```

Scilab code Exa 8.14 Determine duty cycle for motoring regenerating mode and to achieve regenerating mode and poweer returned

```
// chapter 8
// example 8.14
// Determine duty cycle for motoring, regenerating mode and to achieve regenerating mode and poweer returned
// page-477
clear;
clc;
// given
Edc=220; // in V (dc source)
R=0.1; // in ohm
L=10; // in mH
Eb=100; // in V (back emf)
I0=10; // in A
// calculate
L=L*1E-3; // changing unit from mH to H
// since I0=(E0-Eb)/R therefore we get,
E0=I0*R+Eb; // calculation of average load voltage in motoring mode
// since E0=alpha*Edc, therefore we get
alpha_1=E0/Edc; // calculation of duty cycle for motoring mode
alpha_2=Eb/Edc; // calculation of duty cycle for generating mode
alpha_3=(Eb-I0*R)/Edc; // calculation of duty cycle to achieve regenerating mode
P=Eb*I0-I0^2*R; // calculation of power returned to source during braking
printf("\nThe duty cycle for motoring mode is \t\t %.3f ",alpha_1);
printf("\nThe critical duty cycle for generating
```

```
mode is \t %.4f ”,alpha_2);
printf(”\nThe duty cycle to achieve regenerating mode is \t %.2f ”,alpha_3);
printf(”\nThe power returned to source during braking is \t %.f W”,P);
```

Scilab code Exa 8.15 Determine duty cycle for motoring regenerating mode and to achieve regenerating mode power returned and switching frequency

```
// chapter 8
// example 8.14
// Determine duty cycle for motoring, regenerating mode and to achieve regenerating mode, poweer returned and switching frequency
// page-478
clear;
clc;
// given
Edc=220; // in V (dc source)
R=0.1; // in ohm
L=10; // in mH
Eb=100; // in V (back emf)
I0=10; // in A
f=5; // in KHz (output frequency)
// calculate
L=L*1E-3; // changing unit from mH to H
// since I0=(E0-Eb)/R therefore we get,
E0_m=I0*R+Eb; // calculation of average load voltage in motoring mode
// since E0=(2*alpha-1)*Edc, therefore we get
alpha_m=(E0_m/Edc+1)/2; // calculation of duty cycle for motoring mode
alpha_c=1-alpha_m; // calculation of duty cycle for generating mode
// since I0=(Eb-E0)/R during regeneration mode,
```

```
therefore we get
E0_r=Eb-I0*R;
// since E0=-(2*alpha_r -1)*Edc
alpha_r=(1-E0_r/Edc)/2;
P=Eb*I0-I0^2*R; // calculation of power returned to source during braking
f_s=f/2;
printf("\nThe duty cycle for motoring mode is \t\t alpha_m=%.2f ",alpha_m);
printf("\nThe critical duty cycle for generating mode is \t alpha_c=%.2f ",alpha_c);
printf("\nThe duty cycle to achieve regenerating mode is \t alpha_r=%.3f ",alpha_r);
printf("\nThe power returned to source during braking is \t P=%.f W",P);
printf("\nThe switching frequency of the device is \t fs=%.1f KHz",f_s);
```

Scilab code Exa 8.16 Determine duty cycle of chopper

```
// chapter 8
// example 8.16
// Determine duty cycle of chopper
// page-480
clear;
clc;
// given
Edc=200; // in V (dc source)
R=0.1; // in ohm
L=10; // in mH
Eb1=150, Eb2=-110; // in V (back emf)
I0=10; // in A
// calculate
L=L*1E-3; // changing unit from mH to H
// since I0=(E0-Eb)/R therefore we get ,
```

```
E01=I0*R+Eb1; // calculation of average load voltage when Eb=150 V
E02=I0*R+Eb2; // calculation of average load voltage when Eb=-110 V
// since E0=2*Edc(alpha-0.5), therefore we get
alpha1=(E01/(2*Edc))+0.5; // calculation of duty cycle when Eb=150 V
alpha2=(E02/(2*Edc))+0.5; // calculation of duty cycle when Eb=-110 V
printf("\nThe duty cycle when Eb=%.f V is \t %.3f ",Eb1,alpha1);
if alpha1>0.5 then
    printf("\t\t It is forwarding mode");
else
    printf("\t\t It is reversing motoring mode");
end
printf("\nThe duty cycle when Eb=%.f V is \t %.3f ",Eb2,alpha2);
if alpha2>0.5 then
    printf("\t\t It is forwarding mode");
else
    printf("\t\t It is reversing motoring mode");
end
// Note : the answer vary slightly due to precise calculation
```

Scilab code Exa 8.17 Compute the value of commutating capacitor and inductor

```
// chapter 8
// example 8.17
// fig. 8.26
// Compute the value of commutating capacitor and inductor
// page-483-484
```

```
clear;
clc;
// given
Edc=80; // in V (dc source)
Toff=20; // in us (turn off time of SCR)
T=2000; // in us (chopping period)
I0_m=80; // in A (starting current)
// calculate
Toff=Toff*1E-6;
T=T*1E-6;
dt=Toff; // assumption as done in the book
t_q=Toff+dt;
C=I0_m*(t_q)/Edc; // calculation of commutating capacitance
L_min=C*(Edc/I0_m); // calculation of minimum commutating inductance
L_max=0.01*T^2/(%pi^2*C); // calculation of maximum commutating inductance
printf("\nThe commutating capacitance is \t C=%.f uF",C*1E6);
printf("\nThe commutating inductance lies in the range \t %.f uH <= L <= %.2f uH",L_min*1E6,L_max*1E6);
```

Scilab code Exa 8.18 Compute the value of commutating capacitor and inductor Maximum capacitor voltage and peak commutating current

```
// chapter 8
// example 8.18
// Compute the value of commutating capacitor and inductor, Maximum capacitor voltage, and peak commutating current
// page-491
clear;
clc;
```

```
// given
Edc=220; // in V (dc source)
Toff=20; // in us (turn off time of SCR)
I0_m=180; // in A (starting current)
x=3
// calculate
Toff=Toff*1E-6;
dt=Toff; // assumption as done in the book
t_q=Toff+dt;
L=(Edc*t_q)/(x*I0_m*(%pi-2*asin(1/x))); // calculation of commutating inductance
C=x*I0_m*t_q/(Edc*(%pi-2*asin(1/x))); // calculation of commutating capacitance
V_cp=Edc+I0_m*sqrt(L/C); // calculation of Maximum capacitor voltage
i_cp=x*I0_m; // calculation of peak commutating current
printf("\nThe commutating inductance is \t\t L=%.2f uH",L*1E6);
printf("\nThe commutating capacitance is \t\t C=%.2f uF",C*1E6);
printf("\nThe maximum capacitor voltage is \t V_cp=%.2f V",V_cp);
printf("\nThe peak commutating current is \t i_cp=%.f A",i_cp);
// Note: the answer for i_cp is wrong in the book due to use of wrong value in the calculation
```

Scilab code Exa 8.19 Compute the value of commutating capacitance average output voltage circuit off time total commutation interval

```
// chapter 8
// example 8.19
// Compute the value of commutating capacitance, average output voltage, circuit off-time, total
```

```
commutation interval
// page-495
clear;
clc;
// given
Edc=230; // in V (dc source)
I0=50; // in A (load current)
alpha=0.4; // duty cycle
f=2; // in KHz (chopping frequency)
// calculate
f=f*1E3;// changing unit from KHz to Hz
T=1/f; // calculation of chopping period
// since alpha=Ton/T, therefore we get
Ton=alpha*T; // calculation of ON period of chopper
C=Ton*I0/(2*Edc); // calculation of commutating capacitance
E0=2*Edc^2*C*f/I0; // calculation of average output voltage
t_q=Ton/2; // calculation of circuit off-time
T_interval=2*C*Edc/I0; // calculation of total commutation interval
printf("\nThe commutating capacitance is \t\t\t C=%.2f uF",C*1E6);
printf("\nThe average output voltage is \t\t\t E0=%.f V",E0);
printf("\nThe circuit tunr-off time for one SCR pair is \t t_q=%.f us",t_q*1E6);
printf("\nThe total commutation interval is \t\t %.f us",T_interval*1E6);
```

Scilab code Exa 8.20 Compute the value of commutating capacitance and transformer inductances

```
// chapter 8
// example 8.20
```

```
// Compute the value of commutating capacitance and transformer inductances
// page-499
clear;
clc;
// given
Edc=60; // in V (dc source)
I0_m=140; // in A (load current)
t_q=20; // in us
g=4;
// calculate
t_q=t_q*1E-6; // changing unit from us to s
Rm=Edc/I0_m;
// since g=(1/Rm)*sqrt(L1/C) and t_q=sqrt(L1*C),
// Multiplying both these equations we get g*t_q=(1/Rm)*L1 or we get
L1=g*t_q*Rm; // calculation of transformer inductance L1
C=t_q^2/L1; // calculation of commutating capacitance
L2=L1; // calculation of transformer inductance L2
printf("\nThe commutating capacitance is \t C=%.2f uF",C*1E6);
printf("\nThe transformer inductances is \t L1=%.1f uH \t L2=%.1f uH",L1*1E6, L2*1E6);
// Note: the answers vary slightly due to precise calculation
```

Scilab code Exa 8.21 Compute the inductance steady state speed and current swing

```
// chapter 8
// example 8.21
// fig. 8.32
// Compute the inductance, steady-state speed and
```

```
   current swing
// page-499-500
clear;
clc;
// given
di_a=6; // in A
n=1750; // in rpm (rated speed of motor)
HP=150; // in HP (horse power rating)
Ra=0.0099; // in ohm (armature resistance)
Eff=90; // in percent
Ea=1200; // in V (armature voltage)
Edc=Ea;
T=2500; // in us (chopping period)
alpha=0.10; // duty cycle
L1=125; // in mH
// calculate
T=T*1E-6; // changing unit from us to s
// since dIa/d(alpha)=(Edc/L)*T*(1-2*alpha), and dIa/d(alpha)=0 therefore we get
// 1-2*alpha_worst=0
alpha_worst=1/2; // calculation of duty cycle for worst case of motor
// since di_a=(Edc-alpha_worst*Edc)/L, therefore we get
L=(Edc-alpha_worst*Edc)/di_a*alpha_worst*T; // calculation of inductance
Pin=HP*746/(Eff/100); // calculation of power input to the motor
Ia=Pin/Edc; // calculation of rated current in the armature
Ea_rated=Edc-Ia*Ra; // calculation of armature voltage under rated torque
Va=alpha*Edc-Ia*Ra; // calculation of voltage at the armature
N=(Va/Ea_rated)*n; // calculation of steady-state speed
del_i_a=(Edc-alpha*Edc)/L*alpha*T; // calculation of current swing
```

```
printf("\nThe inductance is \t\t L=%.f mH",L*1E3);
printf("\nThe steady-state speed is \t N=%.2f rpm",N);
printf("\nThe current swing is \t\t del_i_a=%.2f A",del_i_a);
// Note: the answers vary slightly due to precise calculation
```

Scilab code Exa 8.22 Design the jones chopper

```
// chapter 8
// example 8.22
// Design the jones chopper
// page-500-501
clear;
clc;
// given
Edc=200; // in V (source voltage)
I0=50; // in A (load current)
t_q=200; // in us
safety_factor=1.5;
// calculate
t_q=t_q*1E-6;  // changing unit from us to s
C=(%pi/2)*(t_q/Edc)*I0; // calculation of commutating capacitance
V_C=safety_factor*Edc; // calculation of capacitor voltage rating
// since t_q=sqrt(L1*C), therefore we get
L1=t_q^2/C; // calculation of commutating inductance L1
L2=L1; // calculation of commutating inductance L2
V_BO=safety_factor*Edc; // for SCR 1
I_T=safety_factor*I0; // for SCR 1
t_q2=(%pi/2)*sqrt(L1*C); // for SCR 1
PIV=V_BO; // calculation of Peak inverse voltage for
```

```
diode D1
I_D=I_T; // calculation of diode current for diode D1
dV_dt_1=I0/C; // dynamic characteristics for SCR 1
di_dt_1=safety_factor*Edc/L1; // dynamic characteristics for SCR 1
E_peak=safety_factor*Edc;
dV_dt_2=E_peak/sqrt(L1*C); // dynamic characteristics for SCR 2
L=2; // in uH (assumption as done in the book)
di_dt_2=E_peak/L; // dynamic characteristics for SCR 2
printf("\nThe commutating capacitance is \t\t C=%.2f uF",C*1E6);
printf("\n\nThe capacitor voltage rating is \t V_C=%.f V",V_C);
printf("\n\nThe commutating inductances are \t L1=%.2f mH \t or L2=%.2f mH",L1*1E3,L2*1E3);
printf("\n\nThe characteristics of SCR1 are \t V_BO=%.f V \t I_T=%.f A",V_BO,I_T);
printf("\n\nThe characteristics of SCR2 are \t t_q<=%.f us or t_q=250 us \t V_BO=%.f V \t I_T=%.f A",t_q2*1E6,V_BO,I_T);
printf("\n\nThe characteistics of diode D1 are \t PIV=%.f V \t I_D=%.f A",V_BO,I_D);
printf("\n\nThe dynamic characteistics of SCR1 are \t dV/dt=%.2f V/us \t di/dt=%.2f A/us",dV_dt_1*1E-6,di_dt_1*1E-6);
printf("\n\nThe dynamic characteistics of SCR2 are \t dV/dt=%.2f V/us \t di/dt=%.f A/us",dV_dt_2*1E-6,di_dt_2*1E-6);
// Note: the answers vary slightly due to precise calculation
```

Scilab code Exa 8.23 Find duty cycle filter inductance and filter capacitance

```
// chapter 8
// example 8.23
// fig 8.43
// Find duty cycle, filter inductance and filter capacitance
// page-514-515
clear;
clc;
// given
Edc=14; // in V (dc source)
E0=6; // in V (average output voltage)
del_Vc=15; // mV (peak to peak ripple voltage)
del_I=0.6; // in A (peak to peak ripple current)
f=30; // in KHz (switching frequency)
// calculate
del_Vc=del_Vc*1E-3; // changing unit from mV to V
f=f*1E3; // changing unit from KHz to Hz
alpha=E0/Edc; // calculation of duty cycle
L=E0*(Edc-E0)/(f*Edc*del_I); // calculation of filter inductance
C=del_I/(8*f*del_Vc); // calculation of filter capacitance

printf("\nThe duty cycle is \t\t alpha=%.4f \t or %.2f percent",alpha,alpha*100);
printf("\nThe filter inductance is \t L=%.2f uH",L*1E6);
printf("\nThe filter capacitance is \t C=%.2f uF",C*1E6);
// Note: the answers vary slightly due to precise calculation
```

Scilab code Exa 8.24 Determine maximum ON period of MOSFET Battery drain current and the value of choke required

```
// chapter 8
// example 8.24
// Determine maximum ON-period of MOSFET, Battery drain current and the value of choke required
// page-515
clear;
clc;
// given
Edc=12; // in V
Edc_max=13.5; // in V (maximum dc source)
Edc_min=10; // in V (minimum dc source)
E0=5; // in V (average output voltage)
I0=10; // in A (load current)
f=50; // in KHz
del_I=500; // in mA (ripple current)
// calculate
f=f*1E3; // changing unit from KHz to Hz
del_I=del_I*1E-3; // changing unit from mA to A
T=1/f; // calculation of time period
alpha_max=E0/Edc_min; // calculation of duty cycle
// since alpha=Ton/T, therefore we get
Ton_max=alpha_max*T; // calculation of maximum ON-period of MOSFET
// since Edc*Is=E0*I0,  therefore we get
Is=E0*I0/Edc; // calculation of Battery drain current
L=E0*(Edc_min-E0)/(f*Edc_min*del_I); // calculation of the value of choke required
printf("\nThe maximum ON-period of MOSFET is \t Ton_max=%.f us",Ton_max*1E6);
printf("\nThe Battery drain current is \t\t Is=%.2f A",Is);
printf("\nThe value of choke required is \t\t L=%.f uH",L*1E6);
// Note: the answers vary slightly due to precise
```

```
calculation
```

Scilab code Exa 8.25 Determine duty cycle ripple and peak current of inductor ripple voltage

```
// chapter 8
// example 8.25
// fig. 8.44
// Determine duty cycle, ripple and peak current of inductor, ripple voltage
// page-518
clear;
clc;
// given
Edc=6; // in V (dc source)
E0=18; // in V (average output voltage)
I0=0.4; // in A (average load current)
f=20; // in KHz (switching frequency)
L=250; // in uH
C=420; // in uH
// calculate
f=f*1E3; // changing unit from KHz to Hz
L=L*1E-6; // changing unit from uH to H
C=C*1E-6; // changing unit from uF to F
// since E0=Edc/(1-alpha), therefore we get
alpha=1-Edc/E0; // calculation of duty cycle
del_I=Edc*(E0-Edc)/(f*L*E0); // calculation of ripple current of inductor
Is=I0/(1-alpha);
I2=Is+del_I/2; // calculation of peak current of inductor
del_Vc=I0*alpha/(f*C); // calculation of ripple voltage
printf("\nThe duty cycle is \t\t\t\t alpha=%.4f \t or \t %.2f percent",alpha,alpha*100);
```

```
printf("\nThe ripple current of inductor is \t\t del_I=%.2f A",del_I);
printf("\nThe peak current of inductor is \t\t I2=%.1f A",I2);
printf("\nThe ripple voltage of filter cqpacitor is \t del_Vc=%.2f mV",del_Vc*1E3);5
// Note: the answers vary slightly due to precise calculation
```

Scilab code Exa 8.26 Compute average output voltage peak to peak output ripple voltage peak to peak current of the inductor and peak current of the device

```
// chapter 8
// example 8.26
// fig. 8.45
// Compute average output voltage, peak-to-peak output ripple voltage, peak-to-peak current of the inductor and peak current of the device
// page-522-523
clear;
clc;
// given
Edc=14; // in V (dc source)
alpha=0.6; // duty cycle
f=25; // in KHz (switching frequency)
L=180; // in uH
C=220; // in uH
I0=1.5; // in A (average load current)
// calculate
f=f*1E3; // changing unit from KHz to Hz
L=L*1E-6; // changing unit from uH to H
C=C*1E-6; // changing unit from uF to F
E0=-Edc*alpha/(1-alpha); // calculation of average output voltage
```

```
del_Vc=I0*alpha/(f*C); // calculation of peak-to-peak output ripple voltage
del_I=Edc*alpha/(f*L); // calculation of ripple current of inductor
Is=I0*alpha/(1-alpha); // calculation of average input current
Ip=Is/alpha+del_I/2; // calculation of peak current of the device
printf("\nThe average output voltage is \t\t\t E0=%.f V",E0);
printf("\nThe peak-to-peak output ripple voltage is \t del_Vc=%.2f V",del_Vc);
printf("\nThe peak-to-peak ripple current of inductor is \t del_I=%.2f A",del_I);
printf("\nThe peak current of the device is \t\t Ip=%.2f A",Ip);
// Note: the answers vary slightly due to precise calculation
```

Scilab code Exa 8.27 Determine range of duty cycle peak to peak choke ripple current and average supply current

```
// chapter 8
// example 8.27
// Determine range of duty cycle , peak-to-peak choke ripple current and average supply current
// page-523-524
clear;
clc;
// given
Edc=24; // in V (dc source)
f=50; // in KHz (switching frequency)
L=500; // in uH
E0=15; // in V (average output voltage)
Edc_max=26; // in V (maximum voltage of dc source)
```

```
Edc_min=21; // in V (maximum voltage of dc source)
I0=2; // in A (average load current)
// calculate
f=f*1E3; // changing unit from KHz to Hz
L=L*1E-6; // changing unit from uH to H
// since E0=Edc*alpha/(1-alpha), therefore we get
alpha_max=1/((Edc_min/E0)+1); // calculation of upper limit of duty cycle
alpha_min=1/((Edc_max/E0)+1); // calculation of lower limit of duty cycle
alpha_normal=1/((Edc/E0)+1); // calculation of normal duty cycle
del_I=Edc*alpha_normal/(f*L); // calculation of peak-to-peak choke ripple current
// since Edc*Is=E0*I0, therefore we get
Is=E0*I0/Edc; // calculation of average supply current
printf("\nThe duty cycle varies from \t\t\t\t\t\t %.3f to %.3f",alpha_min,alpha_max);
printf("\nThe peak-to-peak choke ripple current for normal supply voltage is \t del_I=%.1f mA",del_I*1E3);
printf("\nThe average supply current drawn from battery is \t\t\t Is=%.2f A",Is);
// Note: the answers vary slightly due to precise calculation
```

Scilab code Exa 8.28 Compute average output voltage average input current peak to peak current of the inductors peak to peak output ripple voltages and peak current of the device

```
// chapter 8
// example 8.28
// fig. 8.46
// Compute average output voltage, average input
```

```
current, peak-to-peak current of the inductors, peak-to-peak output ripple voltages and peak current of the device
// page-529-530
clear;
clc;
// given
Edc=15; // in V (dc source)
alpha=0.4; // duty cycle
f=25; // in KHz (switching frequency)
L1=250; // in uH
C1=400; // in uH
L2=380; // in uH
C2=220; // in uH
I0=1.25; // in A (average load current)
// calculate
f=f*1E3; // changing unit from KHz to Hz
L1=L1*1E-6; // changing unit from uH to H
C1=C1*1E-6; // changing unit from uF to F
L2=L2*1E-6; // changing unit from uH to H
C2=C2*1E-6; // changing unit from uF to F
E0=-Edc*alpha/(1-alpha); // calculation of average output voltage
Is=I0*alpha/(1-alpha); // calculation of average input current
del_I1=Edc*alpha/(f*L1); // calculation of ripple current of inductor L1
del_Vc1=Is*(1-alpha)/(f*C1); // calculation of peak-to-peak output ripple voltage of capacitor C1
del_I2=Edc*alpha/(f*L2); // calculation of ripple current of inductor L2
del_Vc2=alpha*Edc/(8*C2*L2*f^2); // calculation of peak-to-peak output ripple voltage of capacitor C2
V_DF=E0; // calculation of average voltage across diode
I_L2=I0*E0/V_DF; // calculation of average current in inductor L2
```

```
Ip=Is+del_I1/2+I_L2+del_I2/2; // calculation of peak current of the device
printf(”\nThe average output voltage is \t\t\t\t\t E0=%.f V”,E0);
printf(”\nThe average input voltage is \t\t\t\t\t Is=%.2f A”,Is);
printf(”\nThe peak−to−peak ripple current of inductor L1 is \t\t del_I1=%.2f A”,del_I1);
printf(”\nThe peak−to−peak output ripple voltage of capacitor C1 is \t del_Vc1=%.2f mV”,del_Vc1*1E3);
printf(”\nThe peak−to−peak ripple current of inductor L2 is \t\t del_I2=%.2f A”,del_I2);
printf(”\nThe peak−to−peak output ripple voltage  of capacitor C2 is \t del_Vc2=%.2f mV”,del_Vc2*1E3);
printf(”\nThe peak current of the device is \t\t\t\t Ip=%.2f A”,Ip);
// Note: the answers vary slightly due to precise calculation
```

Chapter 9

Inverters

Scilab code Exa 9.1 Compute rms value of output voltage fundamental component of the output voltage waveform first five harmonics of the output voltage fundamental power consumed rms power consumed rms value by harmonic summation method

```
// chapter 9
// example 9.1
// Compute rms value of output voltage, fundamental
   component of the output-voltage waveform, first
   five harmonics of the output voltage, fundamental
    power consumed,rms power consumed, rms value by
   harmonic summation method
// page-544
clear;
clc;
// given
Edc_2=96; // in V (dc source at center tapped)
R=10; // in ohm
// calculate
Edc=2*Edc_2;
E0_rms=Edc/2; // calculation of rms value of output
   voltage
printf("\nThe rms value of output voltage is \t\t\t
```

```
    E0_rms=%.f V",E0_rms);
E1_fund=sqrt(2)/%pi*Edc; // calculation of
    fundamental component of the output-voltage
    waveform
printf("\n\nThe fundamental component of the output
    voltage is \t E1_fund=%.2f V\n",E1_fund);
P0_fund=E1_fund^2/R; // calculation of fundamental
    power consumed
P0_rms=E0_rms^2/R; // calculation of rms power
    consumed
E0_rms_H=E1_fund^2;
for n=3:2:11
    E0=E1_fund/n;
    printf("\n The %.f harmonic voltage is \t %.2f V
        ",n,E0); // calculation of rms value of
        voltage by harmonic summation method
    E0_rms_H=E0_rms_H+E0^2;
end
E0_rms_H=sqrt(E0_rms_H); // calculation of rms value
     by harmonic summation method
printf("\n\nThe fundamental power consumed is \t\t\t
    \t P0_fund=%.2f W",P0_fund);
printf("\nThe rms power consumed is \t\t\t\t\t
    P0_rms=%.2f W",P0_rms);
printf("\n\nThe rms value determined by harmonic
    summation method is \t E0_rms=%.2f V",E0_rms_H);
printf("\n\n The two values of rms voltages as
    determined are almost equal.");
// Note: The answer varies slightly due to precise
    calculation
```

Scilab code Exa 9.2 Compute fundamental output voltage first five harmonics of the output voltage rms value by direct integration and harmonic summation method output rms and fundamental power and transistor switch ratings

```
// chapter 9
// example 9.2
// Compute fundamental output-voltage, first five harmonics of the output voltage,rms value by direct integration and harmonic summation method,output rms and fundamental power  and transistor switch ratings
// page-547
clear;
clc;
// given
Edc=48; // in V (dc source)
R=10; // in ohm
// calculate
E1_fund=2*sqrt(2)/%pi*Edc; // calculation of fundamental component of the output-voltage waveform
printf("\n\nThe fundamental component of the output voltage is \t E1_fund=%.2f V\n",E1_fund);
E0_rms=Edc; // calculation of rms value of output voltage
printf("\nThe rms value of output voltage is \t\t\t E0_rms=%.f V\n",E0_rms);
E0_rms_H=E1_fund^2;
for n=3:2:11
    E0=E1_fund/n;
    printf("\n The %.f harmonic voltage is \t %.2f V",n,E0); // calculation of rms value of voltage by harmonic summation method
    E0_rms_H=E0_rms_H+E0^2;
end
E0_rms_H=sqrt(E0_rms_H); // calculation of rms value by harmonic summation method
printf("\n\nThe rms value determined by harmonic summation method is \t E0_rms=%.2f V",E0_rms_H);
P0_rms=E0_rms^2/R; // calculation of rms power consumed
printf("\n\nThe rms power consumed is \t\t\t\t\t
```

```
P0_rms=%.2f W",P0_rms);
P0_fund=E1_fund^2/R; // calculation of fundamental power consumed
printf("\n\nThe fundamental power consumed is \t\t\t\t P0_fund=%.2f W",P0_fund);
V_CE0=Edc;
I_T_peak=Edc/R;
I_T_rms=Edc/(sqrt(2)*R);
I_T_avg=Edc/(2*R);
printf("\n\nThe transistor switch ratings are \t VCE0=%.f V \t I_T_peak=%.1f A \t I_T_rms=%.3f A \t I_T_avg=%.1f A",V_CE0,I_T_peak,I_T_rms,I_T_avg);
// Note: The answer varies slightly due to precise calculation
```

Scilab code Exa 9.3 Determine rms load current at fundamental frequency rms value of load current power output and average supply current

```
// chapter 9
// example 9.3
// fig. 9.5
// Determine rms load current at fundamental frequency, rms value of load current, power output and average supply current
// page-550
clear;
clc;
// given
Edc=220; // in V (dc source)
R=10; // in ohm
L=10; // in mH
C=52; // in uF
f=400; // in Hz
// calculate
```

```
L=L*1E-3; // changing unit from mH to H
C=C*1E-6; // changing unit from uF to F
X_L=2*%pi*f*L; // calculation of inductive reactance
X_C=1/(2*%pi*f*C); // calculation of inductive reactance
I=0; // intialisation of variable for rms load current
// since Impedence offered to the nth harmonic component Zn=sqrt(R^2+(n*X_L-X_C/n)^2)
printf("\nn\t\tZn\t\t\tIn");
for n=1:2:9
    Zn=sqrt(R^2+(n*X_L-X_C/n)^2); // calculation of Impedence offered to the nth harmonic component
    En=0.9*Edc/n; // calculation of rms value of the nth harmonic component of the output voltage
    In=En/Zn; // calculation of rms value of nth harmonic component of the current
    printf("\n%.f\t\t%.2f ohm\t\t%.3f A",n,Zn,In);
    I=I+In^2;
end
I=sqrt(I);
printf("\n\nThe rms value of load current is \t I=%.2f A",I);
P0=I^2*R; // calculation of output power
Iav=P0/Edc; // calculation of average supply current
printf("\n\nThe output power is \t\t\t P0=%.2f W",P0);
printf("\n\nThe average supply current is \t\t Iav=%.2f A",Iav);
// Note: The answer varies slightly due to precise calculation
```

Scilab code Exa 9.4 Determine IGBT ratings THD DF and HF and DF of the lowest order harmonic

```
// chapter 9
// example 9.4
// fig. 9.5
// Determine IGBT ratings, THD, DF and HF and DF of the lowest order harmonic
// page-552-554
clear;
clc;
// given
Edc=24; // in V (dc source)
R=3; // in ohm
// calculate
I_peak=(Edc/2)/R; // calculation of IGBT peak current
V_BR=2*(Edc/2); // calculation of peak reverse voltage of each IGBT
printf("\nThe IGBT ratings are \t I_peak=%.f A \t V_BR=%.f V",I_peak,V_BR);
E1_rms=2*Edc/(sqrt(2)*%pi);
E0_rms=(Edc/2);
THD=sqrt(E0_rms^2-E1_rms^2)/E1_rms; // calculation of total harmonic distortion
printf("\nThe total harmonic distortion is \t THD=%.3f or \t %.1f percent",THD,THD*100);
K=0;
for n=3:2:13
    En_rms=2*Edc/(n*%pi*sqrt(2));
    En_rms_n2=(En_rms/n^2)^2;
    K=K+En_rms_n2;
end
K=sqrt(K);
DF=K/E1_rms; // calculation of distortion factor
printf("\nThe distortion factor is \t\t DF=%.3f or \t %.1f percent",DF,DF*100);
E3_rms=2*Edc/(3*%pi*sqrt(2));
HF3=E3_rms/E1_rms; // calculation of lowest order of harmonic distortion
printf("\nThe lowest order harmonic factor is \t HF3
```

```
=%.3f or \t %.2f percent",HF3,HF3*100);
DF3=(E3_rms/3^2)/E1_rms; // calculation of lowest order distortion factor
printf("\nThe lowest order distortion factor is \t DF3=%.3f or \t %.2f percent",DF3,DF3*100);
// Note: The answer varies slightly due to precise calculation
```

Scilab code Exa 9.5 Determine transistor ratings THD DF and HF and DF of the lowest order harmonic

```
// chapter 9
// example 9.5
// Determine transistor ratings , THD, DF and HF and DF of the lowest order harmonic
// page-554
clear;
clc;
// given
Edc=48; // in V (dc source)
R=3; // in ohm
// calculate
Ip=Edc/R; // calculation of transistor peak current
I_avg=Ip/2; // calculation of transistor average current
V_BR=Edc; // calculation of peak reverse voltage of each IGBT
printf("\nThe transistor ratings are \t Ip=%.f A \t I_avg=%.f A \t V_BR=%.f V",Ip,I_avg,V_BR);
E1_rms=2*Edc/(sqrt(2)*%pi);
E0_rms=(Edc/2);
THD=sqrt(E0_rms^2-E1_rms^2)/E1_rms; // calculation of total harmonic distortion
printf("\nThe total harmonic distortion is \t THD=%.3f or \t %.1f percent",THD,THD*100);
```

```
K=0;
for n=3:2:13
    En_rms=2*Edc/(n*%pi*sqrt(2));
    En_rms_n2=(En_rms/n^2)^2;
    K=K+En_rms_n2;
end
K=sqrt(K);
DF=K/E1_rms; // calculation of distortion factor
printf("\n\nThe distortion factor is \t\t DF=%.3f or  \t %.1f percent",DF,DF*100);
E3_rms=2*Edc/(3*%pi*sqrt(2));
HF3=E3_rms/E1_rms; // calculation of lowest order of harmonic distortion
printf("\n\nThe lowest order harmonic factor is \t HF3=%.3f  or  \t %.2f percent",HF3,HF3*100);
DF3=(E3_rms/3^2)/E1_rms; // calculation of lowest order distortion factor
printf("\n\nThe lowest order distortion factor is \t DF3=%.4f  or  \t %.3f percent",DF3,DF3*100);
// Note: The answer varies slightly due to precise calculation
```

Scilab code Exa 9.6 Determine carrier ratio and the number of pulses per cycle Fundamental output voltage DF and HF of the output voltage and the order of first five significant harmonics

```
// chapter 9
// example 9.6
// Determine carrier ratio and the number of pulses per cycle, Fundamental output voltage, DF and HF of the output voltage and the order of first five significant harmonics
// page-569-570
clear;
clc;
```

```
// given
Edc_2=240; // in V (dc source)
fm=50; // in Hz (fundamental output frequency)
fc=1.2; // in KHz (carrier frequency)
M=0.8; // modulation index
// calculate
fc=fc*1E3;
Mf=fc/fm; // calculation of carrier ratio
p=Mf; // calculation of number of pulses per cycle
E0_fund=(1/sqrt(2))*M*Edc_2; // calculation of Fundamental output voltage
E0_rms=Edc_2;
DF=E0_fund/E0_rms; // calculation of DF of the output voltage
HF=sqrt(1/DF^2-1); // calculation of HF of the output voltage
printf(”\nThe carrier ratio is \t\t\t\t Mf=%.f”,Mf);
printf(”\nThe number of pulses per cycle is \t\t p=%.f”,p);
printf(”\nThe distortion factor of the output voltage is \t DF=%.2f percent”,DF*100);
printf(”\nThe harmonic factor of the output voltage is \t HF=%.1f percent”,HF*100);
printf(”\n\n The order of first five harmonics are \n\nn=”);
for n=0:2:2
    N1=Mf-n;
    N2=Mf+n;
    printf(”%.f, %.f, ”,N1,N2);
end
printf(” etc\n\nn=”);
for n=1:2:3
    N1=2*Mf-n;
    N2=2*Mf+n;
    printf(”%.f, %.f, ”,N1,N2);
end
printf(” etc\n\nn=”);
for n=0:2:2
```

```
    N1=3*Mf-n;
    N2=3*Mf+n;
    printf("%.f, %.f, ",N1,N2);
end
printf(" etc\n\nn=");
for n=1:2:3
    N1=4*Mf-n;
    N2=4*Mf+n;
    printf("%.f, %.f, ",N1,N2);
end
printf(" etc\n\nn=");
for n=0:2:4
    N1=5*Mf-n;
    N2=5*Mf+n;
    printf("%.f, %.f, ",N1,N2);
end
printf(" etc");
// Note: 1. The answer varies slightly due to precise calculation
//       2. The calculation of order of first five significant harmonics is wrong in the book due to calculation mistake
```

Scilab code Exa 9.7 Compute DC and switch voltage ratings

```
// chapter 9
// example 9.7
// Compute DC and switch voltage ratings
// page-570
clear;
clc;
// given
E0_rms=220; // in V (Fundamental output voltage)
M=0.8; // modulation index
// calculate
```

```
// since E0_rms=sqrt(1/2)*M*(Edc_2), therefore we get
Edc_2=E0_rms/(sqrt(1/2)*M); // calculation of DC voltage ratings
V_CEO=2*Edc_2; // calculation of switch voltage ratings
printf("\nThe DC voltage ratings is \t %.2f V",Edc_2);
printf("\nThe switch voltage ratings is \t %.f V",V_CEO);
// Note: The answer varies slightly due to precise calculation
```

Scilab code Exa 9.8 Compute Total rms output voltage fundamental output voltage DF and HF and gain of the invertor

```
// chapter 9
// example 9.8
// Compute Total rms output voltage, fundamental output voltage, DF and HF and gain of the invertor
// page-572
clear;
clc;
// given
Edc=240; // in V (battery voltage)
M=0.8; // modulation index
// calculate
E0_rms=Edc; // calculation of rms output voltage
E0_fund=M*Edc/sqrt(2); // calculation of Fundamental output voltage
DF=M/sqrt(2); // calculation of distortion factor
HF=sqrt(2/M^2-1); // calculation of harmonic factor
G=M/sqrt(2); // calculation of gain of the invertor

```

```
printf("\nThe rms output voltage is \t\t E0_rms=%.f V",E0_rms);
printf("\nThe Fundamental output voltage is \t E0_fund=%.f V",E0_fund);
printf("\nThe distortion factor is \t\t DF=%.4f ",DF);
printf("\nThe harmonic factor is \t\t\t HF=%.3f ",HF);
printf("\nThe gain of the invertor is \t\t G=%.4f ",G);
// Note: The answer varies slightly due to precise calculation
```

Scilab code Exa 9.9 What should be te amplitude of modulation index

```
// chapter 9
// example 9.9
// What should be te amplitude of modulation index
// page-572
clear;
clc;
// given
Edc=120; // in V (battery voltage)
K=1/3; // transformer turn ratio
E01_rms=210; // in V (Fundamental output voltage of secondary)
// calculate
E0_fund=E01_rms*K; // calculation of Fundamental voltage component at primary of the transformer
// since E0_fund=M*Edc/sqrt(2), therefore we get
M=E0_fund*sqrt(2)/Edc; // calculation of amplitude of modulation index
printf("\nThe amplitude of modulation index is \t M=%.2f",M);
// Note: The answer varies slightly due to precise
```

```
calculation
```

Scilab code Exa 9.10 Determine fundamental output voltage and first five dominant harmonics

```
// chapter 9
// example 9.10
// Determine fundamental output voltage and first five dominant harmonics
// page-574
clear;
clc;
// given
Edc=280; // in V (battery voltage)
M=0.8; // modulation index
Mf=24; // carrier ratio
// calculate
E0_fund=M*Edc/sqrt(2); // calculation of Fundamental output voltage
printf("\nThe Fundamental output voltage is \t E0_rms=%.2f V",E0_fund);
printf("\n\nThe first five dominant harmonics are \t n= ")
for n=2:2:6
    N1=n*Mf-1;
    N2=n*Mf+1;
    printf("%.f, %.f, ",N1,N2);
end
printf(" ... etc");
// Note: The answer varies slightly due to precise calculation
```

Scilab code Exa 9.12 Determine rms value of the output line and phase voltages rms value of the fundamental component of the line and phase voltage

```
// chapter 9
// example 9.12
// Determine rms value of the output line and phase voltages, rms value of the fundamental component of the line and phase voltage
// page-582
clear;
clc;
// given
Edc=500; // in V (source voltage)
// calculate
E_line_rms=sqrt(2/3)*Edc; // calculation of rms value of the output line voltage
E_phase_rms=E_line_rms/sqrt(3); // calculation of rms value of the phase voltage
E_line_fund=4*Edc*cosd(30)/(sqrt(2)*%pi); // calculation of rms value of the fundamental component of the line voltage
E_phase_fund=E_line_fund/sqrt(3); // calculation of rms value of the fundamental component of the phase voltage
printf("\nThe rms value of the output line voltage is \t\t\t\t E_line_rms=%.f V",E_line_rms);
printf("\nThe rms value of the phase voltage is \t\t\t\t\t E_phase_rms=%.2f V",E_phase_rms);
printf("\nThe rms value of the fundamental component of the line voltage is \t E_line_fund=%.f V",E_line_fund);
printf("\nThe rms value of the fundamental component of the line voltage is \t E_phase_fund=%.2f V",E_phase_fund);
// Note: The answer varies slightly due to precise calculations and the answer in the book is rounded off
```

Scilab code Exa 9.14 Determine various parameters

```
// chapter 9
// example 9.14
// Determine rms line and phase voltage, rms line and phase voltage at fundamental frequency, THD, DR, HF and DF of lowest order harmonic, load power and average and rms switch current
// page-584-585
clear;
clc;
// given
Edc=200; // in V (source voltage)
R=10; // in ohm
L=20; // in mH
f0=50; // in Hz (invertor frequency)
// calculate
L=L*1E-3; // changing unit from mH to H
w=2*%pi*f0; // calculation of angular velocity
E_L=sqrt(2/3)*Edc; // calculation of rms value of the output line voltage
E_P=E_L/sqrt(3); // calculation of rms value of the phase voltage
E_L1=4*Edc*cosd(30)/(sqrt(2)*%pi); // calculation of rms value of the fundamental component of the line voltage
E_P1=E_L1/sqrt(3); // calculation of rms value of the fundamental component of the phase voltage
printf("\nThe rms value of the output line voltage is \t\t\t\t E_L=%.2f V",E_L);
printf("\nThe rms value of the phase voltage is \t\t\t\t\t E_P=%.2f V",E_P);
printf("\nThe rms value of the fundamental component of the line voltage is \t E_L1=%.2f V",E_L1);
```

```
printf("\nThe rms value of the fundamental component of the line voltage is \t E_P1=%.2f V",E_P1);
E1_rms=2*Edc/(sqrt(2)*%pi);
E0_rms=(Edc/2);
THD=sqrt(E_L^2-E_L1^2)/E_L1; // calculation of total harmonic distortion
printf("\\nnThe total harmonic distortion is \t THD=%.4f or \t %.2f percent",THD,THD*100);
K=0;
for n=5:2:11
    E_Ln=4*Edc*cos(n*%pi/6)/(n*%pi*sqrt(2));
    E_Ln_n2=(E_Ln/n^2)^2;
    K=K+E_Ln_n2;
end
K=sqrt(K);
DF=K/E_L1; // calculation of distortion factor
printf("\nThe distortion factor is \t\t DF=%f or \t %.3f percent",DF,DF*100);
E_L5=4*Edc*cos(5*%pi/6)/(5*%pi*sqrt(2));
HF5=abs(E_L5/E_L1); // calculation of lowest order of harmonic factor
DF5=abs((E_L5/5^2)/E_L1); // calculation of lowest order of distortion factor
printf("\nThe lowest order harmonic factor is \t HF5=%.3f or \t %.2f percent",HF5,HF5*100);
printf("\nThe lowest order distortion factor is \t DF5=%.3f or \t %.2f percent",DF5,DF5*100);
I=0; // initialisation
for n=1:2:17
    Z_L=sqrt(R^2+(n*w*L)^2);
    theta=atand(n*w*L/R);
    I_L=4*Edc*cos(n*%pi/6)/(n*%pi*Z_L);
    I=I+I_L^2;
end
I=sqrt(I)/sqrt(2); // calculation of load current
P0=3*I^2*R; // calculation of load power
Is=P0/Edc; // calculation of average supply current
I_T_avg=Is/3; // calculation of average switch
```

```
current
I_T_rms=I/sqrt(2); // calculation of rms switch current
printf("\n\nThe load power is \t\t P0=%.2f kW",P0*1E-3);
printf("\nThe average switch current is \t I_T_avg=%.2f A",I_T_avg);
printf("\nThe rms switch current is \t I_T_rms=%.2f A",I_T_rms);
// Note: 1. There is calculation mistake while calculating load current and hence the answers for P0, I_T_avg, I_T_rms are wrong in the book
//       2.The answer varies slightly due to precise calculations and the answer in the book is rounded off
```

Scilab code Exa 9.17 Determine dc voltage rms line and phase voltages and Device voltage ratings

```
// chapter 9
// example 9.17
// Determine dc voltage, rms line and phase voltages and Device voltage ratings
// page-591
clear;
clc;
// given
E1_line=415; // fundamental line voltage
// calculate
// since E1_line=(sqrt(2)/%pi)*Edc*1.5, therefore we get
Edc=E1_line*(%pi/(sqrt(2)*1.5)); // calculation of dc voltage
E_rms_line=Edc/sqrt(2); // calculation of rms line voltage
```

```
E_rms_phase=E_rms_line/sqrt(3); // calculation of rms phase voltage
V_CEO=1.5*Edc; // calculation of Device voltage ratings
printf("\nThe dc voltage is \t\t Edc=%.f V", Edc);
printf("\nThe rms line voltage is \t E_rms_line=%.2f V",E_rms_line);
printf("\nThe rms phase voltage is \t E_rms_phase=%.2f V",E_rms_phase);
printf("\nThe Device voltage ratings is \t V_CEO=%.1f V",V_CEO);
// Note: The answer varies slightly due to precise calculations
```

Scilab code Exa 9.18 Design a series invertor circuit

```
// chapter 9
// example 9.18
// Desig a series invertor circuit
// page-606
clear;
clc;
// given
f0=1, fr=5; // in KHz (range of frequency of operation)
Rmin=25, Rmax=100; // in ohm (range of load resistance)
Ip=3;  // in A
Edc=100; // in V (dc source)
AF=0.5; // attenuation factor (assumption as done in the book)
// calculate
f0=f0*1E3; // changing unit from KHz to Hz
fr=fr*1E3; // changing unit from KHz to Hz
wr=2*%pi*fr; // calculation of upper limit of
```

```
angular velocity
w0=2*%pi*f0; // calculation of lower limit of angular velocity
L=-Rmax/(8*fr*log(AF)); // calculation of inductance
C=(1/L)*(1/(wr^2+(Rmax/(2*L))^2)); // calculation of capacitance
Ec=Edc*(exp(-Rmin*%pi/(2*wr*L))/(1-exp(-Rmin*%pi/(2*wr*L)))); // calculation of capacitor voltage
I_peak=(Ec+Edc)/(wr*L)*exp(-Rmin*%pi/(4*wr*L)); // calculation of peak current
// since V_BO>=Ec+Edc, therefore
V_BO=Ec+Edc; // calculation of forward blocking voltage rating
// since I_T>=I-peak, therefore
I_T=(I_peak+1); // calculation of Thyristor current
Toff=%pi*((1/w0)-(1/wr)); // calculation of Thyristor off-time
// since I_T>=I-peak, therefore
tq=Toff*1E3-0.1; // calculation of invertor trun off-time
printf("\nThe inductance \t\t\t L=%.1f mH",L*1E3);
printf("\nThe capacitance \t\t C=%.2f uF",C*1E6);
printf("\nThe capacitor voltage \t\t Ec=%.2f V",Ec);
printf("\nThe peak current \t\t I_peak=%.2f A",I_peak);
printf("\nThe forward blocking voltage \t V_BO>=%.f V",V_BO);
printf("\nThe Thyristor current \t\t I_T=%.f A",I_T);
printf("\nThe Thyristor turn-off time \t Toff=%.1f ms",Toff*1E3);
printf("\nThe invertor turn-off time \t tq=%.1f ms",tq);
// Note: The answer varies slightly due to precise calculations and round off as done in the book
```

Scilab code Exa 9.19 Design a self commutated invertor circuit and compute output power

```
// chapter 9
// example 9.19
// Design a self-commutated invertor circuit and
   compute output power
// page-608-609
clear;
clc;
// given
f0=3; // in KHz
R=5; // in ohm (load resistance)
L=5; // in mH
Edc=100; // in V (dc source)
// calculate
f0=f0*1E3; // changing unit from KHz to Hz
fr=1.35*f0; // calculation of resonant frequency
L=L*1E-3; // changing unit from mH to H
w0=2*%pi*f0; // calculation of normal angular
   frequency
wr=2*%pi*fr; // calculation of resonant angular
   velocity
// since L/L1=200, therefore we get
L1=L/200; // calculation of inductance L1
// since fr=1/(2*%pi*sqrt(2*L1*C)), therefore we
   get
C=(1/(2*%pi*fr))^2/(2*L1); // calculation of
   capacitance
tou=R*C; // calculation of time constant
T=1/f0; // calculation of time period
K=exp(-T/(2*tou)); // calculation of attenuation
   factor
Z0=(R*%i*w0*L)/((R+(%i*w0*L))); // calculation of
```

```
    output impedence
Z0_magnitude=abs(Z0);
I=(Edc/Z0_magnitude)*(1/(1-4*tou*((1-K)/(1+K)))); // calculation of current flowing through Thyristor
E0_max=I*Z0_magnitude*(1-2*K/(1+K)); // calculation of maximum output vltage
E0_rms=E0_max/sqrt(2); // calculation of rms output voltage
V_BO=2*(Edc+E0_max); // calculation of forward blocking voltage rating
I_T=2*I; // calculation of Thyristor current
tq=tou*log(2/(1+K)); // calculation of invertor trun off-time
Vc=2*E0_max; // calculation of capacitor voltage
P0=(E0_max*E0_rms/R)*cosd(atand(imag(Z0),real(Z0))); // calculation of output power
printf(”\nThe resonant frequency is \t\t\t fr=%.2f KHz”,fr*1E-3);
printf(”\nThe inductance L1 is \t\t\t\t L1=%.f uH”,L1*1E6);
printf(”\nThe capacitance is \t\t\t\t C=%.f uF”,C*1E6);
printf(”\nThe current flowing through Thyristor is \t I=%.f A”,I);
printf(”\nThe maximum output voltage is \t\t\t E0_max=%.2f V”,E0_max);
printf(”\nThe rms output voltage is \t\t\t E0_rms=%.2f V”,E0_rms);
printf(”\nThe forward blocking voltage is \t\t V_BO>=%.2f V”,V_BO);
printf(”\nThe Thyristor current is \t\t\t I_T=%.f A”,I_T);
printf(”\nThe invertor turn-off time is \t\t\t tq=%.1f us”,tq*1E6);
printf(”\nThe output power is \t\t\t\t P0=%.2f W”,P0);
// Note: The answer varies slightly due to precise calculations and round off as done in the book
```

Scilab code Exa 9.20 Design a parallel invertor

```
// chapter 9
// example 9.20
// Design a parallel invertor
// page-614
clear;
clc;
// given
f=50; // in Hz
E=200; // in V
Edc=40; // in V
I_L=2; // in A
tq=40; // in us (assumption as done in the book)
E_amplitude=415; // in V (fundamental frequency
    amplitude of voltage assumption as done in the
    book)
Bmax=1; // in Wb/m^2 (assumption for flux density as
     done in the book)
A=25; // in cm^2 (assumption for cross-section area
    as done in the book)
// calculate
tq=tq*1E-6; // changing unit from us to s
A=A*1E-4; // changing unit from cm^2 to m^2
// since Edc*sqrt(C/L)=I_L and sqrt(C*L)=3*tq/%pi,
    therefore multiplying both we get
// Edc*C=I_L*3*tq/%pi, therefore we get
C=(I_L*3*tq/%pi)/Edc; // calculation of capacitance
L=(1/(I_L/Edc)^2)*C; // calculation of inductance
Vc=2*Edc; // calculation of capacitor voltage
N2=E_amplitude/(sqrt(2)*4.44*f*A); // calculation of
     number of turns in secondary winding
// since N2/(1/2*N1)=6, therefore we get
N1=2*N2/6; // calculation of number of turns in
```

```
primary winding
V_BO=2*Edc; // calculation of forward blocking voltage rating
I_T=2*E/Edc*I_L; // calculation of Thyristor current
printf("\nThe inductance is \t\t\t\t L=%.1f uH",L*1E6);
printf("\nThe capacitance is \t\t\t\t C=%.1f uF",C*1E6);
printf("\nThe capacitor voltgae is \t\t\t Vc=%.f V",Vc)
printf("\nThe number of turns in secondary winding is \t N2=%.f",N2);
printf("\nThe number of turns in primary winding is \t N1=%.f",N1);
printf("\nThe forward blocking voltage is \t\t V_BO>=%.f V",V_BO);
printf("\nThe Thyristor current is \t\t\t I_T>=%.f A",I_T);
// Note: 1. The answer for L and C is wrong in the book due to calculation mistake.
//       2 The answer vary slightly due to precise calculations and round off as done in the book
```

Scilab code Exa 9.21 Calculate rms output voltage at fundamental frequency output power average and peak current and peak reverse blocking voltage

```
// chapter 9
// exapmle 9.21
// Calculate rms output voltage at fundamental frequency, output power, average and peak current and peak reverse-blocking voltage
// page-618
clear;
clc;
```

```
// given
Edc=50; // in V (source voltage)
R=3; // in ohm
D=50; // in percent (universal for the circuit used)
// calculate
E1=2*Edc/(sqrt(2)*%pi); // calculation of rms output voltage at fundamental frequency
E_L=Edc/2; // calculation of rms line voltage
P0=E_L^2/R; // calculation of output power
I_peak=(Edc/2)/R; // calculation of peak current
I_av=(D/100)*I_peak; // calculation of average current
V_BR=2*Edc/2; // calculation of peak reverse-blocking voltage
printf("\nThe rms output voltage at fundamental frequency is \t E1=%.1f V",E1);
printf("\nThe output power is \t\t\t\t\t P0=%.2f W",P0);
printf("\nThe peak current is \t\t\t\t\t I_peak=%.2f A",I_peak);
printf("\nThe average current is \t\t\t\t\t I_av=%.3f A",I_av);
printf("\nThe peak reverse-blocking voltage is \t\t\t V_BR=%.f V",V_BR);
// Note: The answer vary slightly due to precise calculations and round off as done in the book
```

Scilab code Exa 9.22 Calculate rms output voltage at fundamental frequency output power average and peak current and peak reverse blocking voltage

```
// chapter 9
// example 9.22
// fig. 9.43 (a)
// Calculate rms output voltage at fundamental
```

```
   frequency, output power, average and peak current
    and peak reverse-blocking voltage
// page-620
clear;
clc;
// given
Edc=50; // in V (source voltage)
R=3; // in ohm
D=50; // in percent (universal for the circuit used)
// calculate
E1=4*Edc/(sqrt(2)*%pi); // calculation of rms output
    voltage at fundamental frequency
E_L=sqrt(2*Edc^2*integrate(’1’,’t’,0,1/2)); //
   calculation of rms line voltage
P0=Edc^2/R; // calculation of output power
I_peak=Edc/R; // calculation of peak current
I_av=(D/100)*I_peak; // calculation of average
   current
V_BR=Edc; // calculation of peak reverse-blocking
   voltage
printf(”\nThe rms output voltage at fundamental
   frequency is \t E1=%.1f V”,E1);
printf(”\nThe output power is \t\t\t\t\t P0=%.2f W’,
   P0);
printf(”\nThe peak current is \t\t\t\t\t I_peak=%.2f
    A”,I_peak);
printf(”\nThe average current is \t\t\t\t\t I_av=%.3
   f A”,I_av);
printf(”\nThe peak reverse-blocking voltage is \t\t\
   t V_BR=%.f V”,V_BR);
// Note: The answer vary slightly due to precise
   calculations and round off as done in the book
```

Scilab code Exa 9.23 What value of C should the load have in order to obtain load commutation of SCR

```
// chapter 9
// example 9.23
// What value of C should the load have in order to obtain load commutation of SCR
// page-620
clear;
clc;
// given
Edc=50; // in V (source voltage)
R=3; // in ohm
wL=12; // in ohm
T=0.2; // in ms (time period)
tq=12; // in us
Toff=1.5*tq;
// calculate
T=T*1E-3; // changing unit from ms to s
Toff=Toff*1E-6; // changing unit from us to s
phi_w=Toff; // calculation of phi/w
f=1/T; // calculation of frequency
w=2*%pi*f; // calculation of angular velocity
phi=phi_w*w; // calculation of phase angle
X_L=wL;
// since Tan(phi)=(X_C-X_L)/R, therefore we get
X_C=(tan(phi)*R)+X_L; // calculation of capacitive reactance
// since X_c=1/(w*C), therefore we get
C=1/(X_C*w); // calculation of required capacitance
printf("\nThe value of C that the load sholud have in order to obtain load commutation of SCR is \t C=%.2f uF",C*1E6);
// Note: The answer in the book is wrong due to calculation mistake in calculating the variable phi_w
```

Scilab code Exa 9.24 Obtain the value of commutating components L and C

```
// chapter 9
// example 9.24
// fig. 9.45
// Obtain the value of commutating components L and C
// page-626
clear;
clc;
// given
I_L_max=80; // in A (maximum load current)
Edc_min=200; // in V (minimum source voltage)
tq=40; // in us (turn-off time of SCRs)
// calculate
tq=tq*1E-6; // changing unit from us to s
Toff=1.5*tq; // calculation of circuit turn-off time
C=0.893*Toff*I_L_max/Edc_min; // calculation of the value of commutating capacitance
L=0.397*Edc_min*Toff/I_L_max; // calculation of the value of commutating inductance
printf("\nThe value of commutating capacitance is \t C=%.2f uF",C*1E6);
printf("\nThe value of commutating inductance is \t\t L=%.2f uH",L*1E6);
// Note: The answer in the book for C and L is wrong due to calculation mistake
```

Scilab code Exa 9.25 Obtain the value of commutating components and the value of resistance

```
// chapter 9
// example 9.25
// fig. 9.47
```

```
// Obtain the value of commutating components and the value of resistance
// page-628
clear;
clc;
// given
Edc=300; // in V (source voltage)
neta=15; // in percent (variation of maximum and minimum dc voltage from dc source voltage)
I_L_max=100; // in A (maximum load current)
I_L_min=20; // in A (minimum load current)
tq=20; // in us (turn-off time of SCRs)
safety_factor=2; // assumption
// calculate
tq=tq*1E-6; // changing unit from us to s
Toff=2*tq; // calculation of circuit turn-off time
Edc_min=Edc*(1-neta/100); // calculation of minimum dc voltage
Edc_max=Edc*(1+neta/100); // calculation of maximum dc voltage
C=0.893*Toff*I_L_max/Edc_min; // calculation of the value of commutating capacitance
L=0.397*Edc_min*Toff/I_L_max; // calculation of the value of commutating inductance
R=2*sqrt(L/C); // calculation of the value of resistance
printf("\nThe value of commutating capacitance is \t C=%.f uF",C*1E6);
printf("\nThe value of commutating inductance is \t\t L=%.1f uH",L*1E6);
printf("\nthe value of resistance is \t\t\t R=%.1f ohm",R);
// Note: The answer in the book for C and L is wrong due to calculation mistake
```

Scilab code Exa 9.26 Find the rms load current load power and thyristor ratings

```
// chapter 9
// example 9.26
// fig. Ex.9.26(a) and Ex. 9.26(b)
// Find the rms load current, load-power and thyristor ratings
// page-628-630
clear;
clc;
// given
Edc=600; // in V (source voltage)
R=15; // in ohm (resistance per phase)
alpha1=120, alpha2=180; // in degree
// calculate
// part-a
Ip=Edc/(2*R); // calculation of load current amplitude
Irms=sqrt((1/(2*%pi))*Ip^2*(integrate('1','theta',0,(alpha1*(%pi/180)))+integrate('1','theta',(alpha2*(%pi/180)),(5*%pi/3)))); // calculation of rms load current
P0=3*Irms^2*R; // calculation of load-power
I_T_rms=sqrt(Ip^2/3); // calculation of thyristor rms current
printf("\nFor %.f degree conduction, \n",alpha1);
printf("\nThe rms load current is \t Irms=%.2f A",Irms);
printf("\nThe load-power is \t\t P0=%.f kW",P0*1E-3);
printf("\nThe thyristor rms current is \t I_T_rms=%.1f A",I_T_rms);
// part-b
R0=R+R/2; // calculation of load on invertor
I1=Edc/R0; // calculation of current I1
I2=I1/2; // calculation of current I2
Irms1=sqrt((1/(2*%pi))*(I2^2*integrate('1','theta'
```

```
,0,(alpha1*%pi/180))+I1^2*integrate('1','theta',(alpha1*%pi/180),(4*%pi/3))+I2^2*integrate('1','theta',(4*%pi/3),(2*%pi)))); // calculation of rms load current Irms1
Irms2=I1/2; // calculation of rms load current Irms2
P0=3*Irms1^2*R; // calculation of load-power
printf("\n\nFor %.f degree conduction, \n",alpha2);
printf("\nThe rms load currents are \t Irms1=%.2f A \t Irms2=%.2f A",Irms1,Irms2);
printf("\nThe load-power is \t\t P0=%.2f kW",P0*1E-3);
// Note: 1. The wrong values of angle theta has been used in the book for calculating the rms load current in both cases. But the answer in the book is right. So I have used correct values and the answer is same as in the book with little variation due to precise calculation
```

Scilab code Exa 9.27 Determine commutating capacitor load current Fcritical and Rcritical

```
// chapter 9
// example 9.27
// Determine commutating capacitor, load current, Fcritical and Rcritical
// page-649-650
clear;
clc;
// given
f=50; // in Hz
tq=62; // in us (Thyristor turn-off time)
R=5; // in ohm
I=1; // in A (let us assume this value for current source for calculation of load current)
// calculate
```

```
tq=tq*1E-6; // changing unit from us to s
Toff=5*tq; // calculation of circuit turn-off time
T=1/f; // calculation of time period
// since Toff=0.69*R*C, therefore we get
C=Toff/(0.69*R); // calculation of commutating capacitor
I1=I*((1-exp(-T/(2*R*C)))/(1+exp(-T/(2*R*C)))); // calculation of load current
Toff_critical=1.5*tq; // calculation of critical circuit turn-off time
// since Toff=R*C*log(2/(1+exp(-T/(2*R*C)))), therefore we get
Tcritical=2*R*C*log(2/(exp(Toff_critical/(R*C))-1)); // calculation of critical time period
Fcritical=1/Tcritical; // calculation of critical frequency
// since Toff=0.69*R*C, therefore we get
Rcritical=Toff_critical/(0.69*C); // calculation of critical resistance
printf("\nThe commutating capacitor is \t C=%.2f uF",C*1E6);
printf("\nThe load current is \t\t I1=I=%.f A",I1);
printf("\nThe critical frequency is \t Fcritical=%.1f Hz",Fcritical);
printf("\nThe critical resistance is \t Rcritical=%.2f ohm",Rcritical);
// Note: 1.The answer varies slightly as from calculated from the book due to precise calculation.
//       2. for calculating other variables, author has use standard value of C (100 uF) for further calculation but I have used the value which is calculated.
```

Scilab code Exa 9.28 Design an OTT filter

```
// chapter 9
// example 9.28
// Design an OTT-filter
// page-662
clear;
clc;
// given
E_L=230; // in V (output voltage)
f=50; // in Hz
I_L=1.5; // in A (load current)
Edc=40; // in V (source voltage)
// calculate
W_D=2*%pi*f; // calculation of design radian frequency
Z_L=E_L/I_L; // calculation of load impedence
Z_D=abs(Z_L)/2; // calculation of filter design impedence
C1=1/(6*Z_D*W_D); // calculation of capacitance C1
C2=1/(3*Z_D*W_D); // calculation of capacitance C2
L1=9*Z_D/(2*W_D); // calculation of inductance L1
L2=Z_D/W_D; // calculation of inductance L1
printf("\nThe design radian frequency is \t W_D=%.f rad/s",W_D);
printf("\nThe load impedence is \t\t Z_L=%.2f ohm",Z_L);
printf("\nThe filter design impedence is \t Z_D=%.2f ohm",Z_D);
printf("\nThe capacitance C1 is \t\t C1=%.2f uF",C1*1E6);
printf("\nThe capacitance C2 is \t\t C2=%.2f uF",C2*1E6);
printf("\nThe capacitance L1 is \t\t L1=%.3f H",L1);
printf("\nThe capacitance L2 is \t\t L1=%.3f H",L2);
// Note: The answer varies slightly due to precise calculation
```

Scilab code Exa 9.29 Design a 400 Hz parallel invertor

```
// chapter 9
// example 9.29
// Design a 400 Hz parallel invertor
// page-663-664
clear;
clc;
// given
P0=360; // in W (output power)
E_L=120; // in V (output voltage)
f=400; // in Hz
PF=0.7; // (lagging) (power factor)
Edc=28; // in V (source voltage)
Zin=80-23*%i; // filter input impedence (assumption as done in the book)
neta=85; // in percent (assumption as done in the book)
Toff=12; // in us (assumption for circuit turn-off time as done in the book)
Ip_SCR=14; // in A (assumption for SCR peak current as done in the book)
// calculate
R_L=E_L^2*PF^2/P0; // calculation of load resistance
X_L=(R_L/PF)*sqrt(1-PF^2); // calculation of load reactance
Z_L_magnitude=sqrt(R_L^2+X_L^2); // calculation of load impedence
Z_L_phase=acosd(PF); // calculation of load impedence phase
Z_D=abs(Z_L_magnitude)/2; // calculation of filter design impedence
W_D=2*%pi*f; // calculation of dsign radian frequency
```

```
C1=1/(6*Z_D*W_D); // calculation of capacitance C1
C2=1/(3*Z_D*W_D); // calculation of capacitance C2
L1=9*Z_D/(2*W_D); // calculation of inductance L1
L2=Z_D/W_D; // calculation of inductance L1
Rin=real(Zin); // calculation of filter input resistance
Xin=abs(imag(Zin)); // calculation of filter input reactance
Zin_magnitude=abs(Zin); // calculation of filter input impedence magnitude
E_SQ=(sqrt(2)/4)*%pi*Zin_magnitude*(P0/Rin)^0.5 // calculation of input voltage to filter
n=E_SQ/Edc; // calculation of transfromer turns ratio
Pi=P0/(neta/100); // calculation of input power
I_T_av=P0*Zin_magnitude/(2*Edc*Rin); // calculation of average thyristor current
V_BO=2.5*Edc; // calculation of blocking voltage
L=6*Edc*Toff/(%pi*Ip_SCR); // calculation of commutating inductance
C=3*Toff*Ip_SCR/(8*%pi*Edc); // calculation of commutating capacitance
di_dt=2*Edc/L; // calculation of rate of change of current
printf("\nThe load resistance is \t\t\t\t R_L=%.f ohm",R_L);
printf("\nThe load reactance is \t\t\t\t X_L=%.f ohm",X_L);
printf("\nThe magnitude of load impedence is \t\t Z_L_magnitude=%.1f ohm",Z_L_magnitude);
printf("\nThe phase of load impedence is \t\t\t Z_L_phase=%.f degree",Z_L_phase);
printf("\nThe filter design impedence is \t\t\t Z_D=%.2f ohm",Z_D);
printf("\nThe design radian frequency is \t\t\t W_D=%.f rad/s",W_D);
printf("\nThe capacitance C1 is \t\t\t\t C1=%.1f uF",C1*1E6);
```

```
printf("\nThe capacitance C2 is \t\t\t\t C2=%.1f uF",C2*1E6);
printf("\nThe capacitance L1 is \t\t\t\t L1=%.1f mH",L1*1E3);
printf("\nThe capacitance L2 is \t\t\t\t L1=%.f mH",L2*1E3);
printf("\nThe filter input resistance is \t\t\t Rin=%.f ohm",Rin);
printf("\nThe filter input reactance is \t\t\t Xin=%.f ohm",Xin);
printf("\nThe magnitude of filter input impedence is \t Zin_magnitude=%.f ohm",Zin_magnitude);
printf("\nThe input voltage to filter is \t\t\t E_SQ=%.f V",E_SQ);
printf("\nThe transfromer turns ratio is \t\t\t n=%.f",n);
printf("\nThe input power is \t\t\t\t Pin=%.f W",Pi);
printf("\nThe verage thyristor current is \t\t I_T_av=%.1f A",I_T_av);
printf("\nThe blocking voltage is \t\t\t V_BO=%.f V",V_BO);
printf("\nThe commutating inductance is \t\t\t L=%.1f uH",L);
printf("\nThe commutating capacitance is \t\t\t C=%.2f uF",C);
printf("\nThe rate of change of current at t=0 s is \t di/dt=%.2f A/us",di_dt);
// Note: 1. The answer varies slightly due to precise calculation.
//       2. Most of the answers have been rounded off in the book and then used for further calculation. I have used the exact values for the calculations
```

Chapter 10

Cycloconverters

Scilab code Exa 10.1 Determine required supply voltage thyristor rating and power factor of the supply current

```
// chapter 10
// example 10.1
// Determine required supply voltage, thyristor
   rating and power factor of the supply current
// page-698-699
clear;
clc;
// given
m=3; // number of phases
PF=0.7; // power factor (lagging)
Edc=190; // in V (load voltage)
I_L=45; // in A
// calculate
// since Edc=(m/%pi)*Eph*sin(%pi/m), therefore we
   get
Eph=Edc/((m/%pi)*sin(%pi/m)); // calculation of
   required supply voltage
Eph_max=Eph*sqrt(2); // calculation of required
   supply voltage maximum value
Imax=I_L*sqrt(2); // calculation of maximum value of
```

```
    cycloconvertor current
Irms=Imax/sqrt(3); // calculation of thyristor rms
   current
PIV=sqrt(3)*Eph_max; // calculation of peak inverse
   voltage
Irms_1_3=sqrt(I_L^2/3); // calculation of rms
   current for one−third cycle
Pin=(1/3)*Edc*I_L*PF; // calculation of input power
   per phase
pf=Pin/(Eph*Irms_1_3); // calculation of power
   factor of the supply current
printf(”\nThe required supply voltage is \t\t\t Eph=
   %.2f V”,Eph);
printf(”\nThe thyristor ratings are \t\t\t Irms==%.2
   f A \t PIV=%.2f V”,Irms,PIV);
printf(”\nThe power factor of the supply current is
   \t pf=%.2f”,pf);
// Note: 1. The answers vary slightly due to precise
    calculation.
```

Scilab code Exa 10.2 Estimate peak and rms value of load voltage

```
// chapter 10
// example 10.2
// Estimate peak and rms value of load voltage
// page−699−700
clear;
clc;
// given
E=600; // in V (line voltage)
f=50; // in Hz (supply frequency)
Ls=1.46; // in mH (inductance per phase)
I_L=28; // in A (load current)
p=6; // number of pulses
alpha1=0,alpha2=30, alpha3=60; // in degrees (firing
```

```
angles)
// calculate
Ls=Ls*1E-3; // changing unit from mH to H
Em=E*sqrt(2); // calculation of peak value of supply voltage
w=2*%pi*f; // calculation of angular frequency in radian
// since Epeak=(p/%pi)*Em*sin(%pi/p)*cosd(alpha)-(p*w*Ls*I_L/(2*%pi)), therefore we get
for alpha=0:30:60
    Epeak=(p/%pi)*Em*sin(%pi/p)*cosd(alpha)-(p*w*Ls*I_L/(2*%pi)); // calculation of peak value of load voltage
    Erms=Epeak/sqrt(2); // calculation of rms value of load voltage
    printf("\n\nFor alpha=%.f degree,",alpha);
    printf("\nThe peak value of load voltage is \t Epeak=%.1f V",Epeak);
    printf("\nThe rms value of load voltage is  \t Erms=%.1f V",Erms);
end
// Note: 1. The value of E used in calculation in the book is 660 V while the value given is 600 V. So i have used 600 V for the calculationas given in the question statement. Therefore the answers do not match.
//        2. The answers vary slightly due to precise calculation.
```

Scilab code Exa 10.3 Determine the value of fundamental rms output voltage rms output current and output power

```
// chapter 10
// example 10.3
// Determine the value of fundamental rms output
```

```
voltage , rms output current and output power
// page-700
clear;
clc;
// given
n=3/1; // transformer turn ratio
wL=3; // in ohm (inductive reactance)
R_L=4; // in ohm (load resistance)
alpha=160; // in degree (firing angle)
E=410; // in V (input voltage per phase)
f=50; // in Hz (supply frequency)
m=3; // number of phases
// calculate
Eph=E/2; // calculation of per phase voltage to convertor
r=cosd(180-alpha); // calculation of voltage reduction factor
Eor=r*Eph*(m/%pi)*sin(%pi/m); // calculation of fundamental rms output voltage ,
Ior_magnitude=Eor/sqrt(R_L^2+wL^2); // calculation of rms output current magnitude
Ior_phase=-atand(wL/R_L); // calculation of rms output current phase
P0=Ior_magnitude^2*R_L; // calculation of output power
printf("\nThe fundamental rms output voltage is \t Eor=%.2f V",Eor);
printf("\nThe rms output current is  \t Ior=%.2f A \t and phase is \t %.2f degree",Ior_magnitude,Ior_phase);
printf("\nThe output power is \t P0=%.2f W",P0);
// Note: The answers vary slightly due to precise calculation.
```

Scilab code Exa 10.4 Determine the value of fundamental rms output voltage rms output current and output power

```
// chapter 10
// example 10.4
// Determine the value of fundamental rms output voltage, rms output current and output power
// page-700-701
clear;
clc;
// given
n=3/1; // transformer turn ratio
wL=3; // in ohm (inductive reactance)
R_L=4; // in ohm (load resistance)
alpha=160; // in degree (firing angle)
E=410; // in V (input voltage per phase)
f=50; // in Hz (supply frequency)
m=6; // number of pulses
// calculate
Eph=E/2; // calculation of per phase voltage to convertor
E_line=Eph*sqrt(3); // calculation of input to bridge invertor
r=cosd(180-alpha); // calculation of voltage reduction factor
Eor=r*E_line*(m/%pi)*sin(%pi/m); // calculation of fundamental rms output voltage,
Ior_magnitude=Eor/sqrt(R_L^2+wL^2); // calculation of rms output current magnitude
Ior_phase=-atand(wL/R_L); // calculation of rms output current phase
P0=Ior_magnitude^2*R_L; // calculation of output power
printf("\nThe fundamental rms output voltage is \t Eor=%.2f V",Eor);
printf("\nThe rms output current is \t Ior=%.2f A \t and phase is \t %.2f degree",Ior_magnitude,Ior_phase);
```

```
printf("\nThe output power is \t P0=%.2f W",P0);
// Note: The answers vary slightly due to precise
   calculation.
```

Scilab code Exa 10.5 Determine the input current to the convertors

```
// chapter 10
// example 10.5
// Determine the input current to the convertors
// page-701
clear;
clc;
// given
m=6; // number of pulses
Pi=50; // in kVA ()power of cycloconvertor)
E=415; // in V
theta=45; // in degree (firing angle)
alpha0=0.8; // power factor
// calculate
Pi=Pi*1E3; // changing unit from kVA to VA
// since Pi=3*E*Im_phase=3*E*I*(cos(theta)/sqrt(2))*
   alpha0, therefore we get
I=Pi/(3*E*(cosd(theta)/sqrt(2))*alpha0);//
   calculation of input current to the convertors
printf("\nThe input current to the convertors is \t
   I=%.2f A",I);
```

Chapter 11

A C Regulators

Scilab code Exa 11.1 Determine various parameters

```
// chapter 11
// example 11.1
// Determine the peak values of average and rms thyristor currents, minimum circuit turn-off time, the ratio of third harmonic voltage to fundamental voltage , maximum value of di/dt and the angle at which greatest forward or reverse voltage is applied.
// page-718-719
clear;
clc;
// given
R=4; // in ohm
E=230; // in V (supply voltage)
f=50; // in Hz
alpha=%pi/3; // in degree
n=3
// calculate
Em=E*sqrt(2);// calculation of peak input voltage
I_T_av_m=Em/(%pi*R);// calculation of peak value of average thyristor current
```

```
I_T_rms_m=Em/(2*R);// calculation of peak value rms thyristor current
w=2*%pi*f;// calculation of angluar frequency
Toff=%pi/w;// calculation of minimum circuit turn-off time
A3=(Em/%pi)*(sin((n+1)*alpha)/(n+1)-sin((n-1)*alpha)/(n-1));
B3=(Em/%pi)*((cos((n+1)*alpha)-1)/(n+1)-(cos((n-1)*alpha)-1)/(n-1));
E_3m=sqrt(A3^2+B3^2);
E_1m=(Em/%pi)*sqrt((sin((n-1)*alpha)/2+(%pi-alpha))^2+((cos((n-1)*alpha)-1)/(n-1))^2);// calculation of
ratio=E_3m/E_1m;// calculation of the ratio of third harmonic voltage to fundamental voltage
di_dt=%inf// calculation of maximum value of di/dt
alpha2=%pi/2;// calculation of the angle at which greatest forward or reverse voltage is applied.
Em_max=sqrt(2)*E;// calculation of magnitude of maximum voltage
printf("\nThe peak value of average thyristor current is \t\t\t\t I_T_av_m=%.2f A",I_T_av_m);
printf("\nThe peak value rms thyristor current is \t\t\t\t I_T_rms_m=%.2f A",I_T_rms_m);
printf("\nThe minimum circuit turn-off time is \t\t\t\t\t Toff_m=%.f ms",Toff*1E3);
printf("\nThe ratio of third harmonic voltage to fundamental voltage is \t\t %.3f ",ratio);
printf("\nThe maximum value of di/dt is \t\t\t\t\t\t di/dt=%f A/s",di_dt);
printf("\nThe angle at which greatest forward or reverse voltage is applied is \t alpha=%.2f radian",alpha2);
printf("\nThe magnitude of maximum voltage is \t\t\t\t\t Em_max=%.2f V",Em_max);
```

Scilab code Exa 11.2 Determine rms value of output voltage input power factor and average input current

```
// chapter 11
// example 11.2
// fig. Ex. 11.2
// Determine rms value of output voltage, input power factor and average input current
// page-719-721
clear;
clc;
// given
R=6; // in ohm
Es=230; // in V (supply voltage)
f=50; // in Hz
alpha=%pi/2; // in degree
n=3
// calculate
E0_rms=sqrt((2*Es^2/(4*%pi))*(integrate('1-cos(2*wt)','wt',alpha,%pi)+integrate('1-cos(2*wt)','wt',%pi,(2*%pi))));// calculation of rms value of output voltage
I0_rms=E0_rms/R;// calculation of rms value of output current
P0=I0_rms^2*R;// calculation of load power
Ea=Es*I0_rms;// calculation of input voltage-ampere rating
pf=P0/Ea;// calculation of input power factor
Edc=(1/(2*%pi))*(sqrt(2)*Es*integrate('sin(wt)','wt',alpha,%pi)+sqrt(2)*Es*integrate('sin(wt)','wt',%pi,(2*%pi)));// calculation of average output voltage
Idc=Edc/R;// calculation of average input current
printf("\nThe rms value of output voltage is \t
```

```
E0_rms=%.2f V",E0_rms);
printf("\nThe input power factor is \t\t pf=%.3f ",pf);
if pf>0 then
    printf(" (lagging)");
else
    printf(" (leading)");
end
printf("\nThe average input current is \t\t Idc=%.2f A",Idc);
```

Scilab code Exa 11.3 Determine various parameters

```
// chapter 11
// example 11.3
// Determine control range of firing angle, maximum value of rms load current, maximum power and power factor, maximum values of average and rms thyristor current, maximum possible of di/dt and the conduction angle for alpha=0 and alpha=120
// page-721-722
clear;
clc;
// given
R=4; // in ohm
Es=230; // in V (supply voltage)
f=50; // in Hz
wL=3; // in ohm (inductive reactance)
alpha1=0, alpha2=120; // in degree
// calculate
w=2*%pi*f;// calculation of angular frequency
Em=Es*sqrt(2);// calculation of peak value of input voltage
alpha_min=atand(wL/R);// calculation of minimum firing angle
```

```
alpha_max=180;// calculation of maximum firing angle
Z=sqrt(R^2+wL^2);// calculation of impedence
I0_rms_m=Es/Z;// calculation of maximum value of rms load current
P0=I0_rms_m^2*R;// calculation of maximum power
pf=P0/(Es*I0_rms_m);// calculation of power factor
I_T_av=(Em/(%pi*Z));// calculation of maximum values of average thyristor current
I_T_rms=Em/(2*Z);// calculation of maximum values of rms thyristor current
di_dt=w*Em/Z;// calculation of maximum possible of di/dt
gama1=180;// calculation of the conduction angle for alpha=0
gama2=95;// calculation of the conduction angle for alpha=120
printf("\nThe control range of firing angle is \t %.2f degree <= alpha <= %.2f degree",alpha_min,alpha_max);
printf("\nThe maximum value of rms load current is \t\t I0_rms_m=%.f A",I0_rms_m);
printf("\nThe maximum power is \t\t\t\t\t P0=%.f W",P0);
printf("\nThe power factor is \t\t\t\t\t pf=%.1f",pf);
printf("\nThe maximum values of average thyristor current is \t I_T_av=%.3f A",I_T_av);
printf("\nThe maximum values of rms thyristor current is \t\t I_T_rms=%.3f A",I_T_rms);
printf("\nThe maximum possible of di/dt is \t\t\t di/dt=%1E A/s",di_dt);
printf("\nThe conduction angle for alpha=0 is \t\t\t gama1=%.f degree",gama1);
printf("\nThe conduction angle for alpha=120 is \t\t\t gama2=%.f degree",gama2);
```

Scilab code Exa 11.4 Plot a curve of load power against firing delay angle and determine required thyristor rating

```
// chapter 11
// example 11.4
// Plot a curve of load power against firing delay angle and determine required thyristor rating
// page-722
clear;
clc;
clf;
// given
R=10; // in ohm
E1=100; // in V
E2=70.7; // in V
f=50; // in Hz (used by the author)
// calculate
w=2*%pi*f;
alpha=[0:0.1:%pi];
P0=1000-(500*alpha-250*sin(2*alpha))/%pi;
plot2d(alpha,P0,nax=[2,3,2,3],rect=[0,0,%pi,1000]);
PRV_12=E1*sqrt(2);
PRV_34=E2*sqrt(2);
printf("\nTHe peak reverse voltage for thyristor T1 and T2 is \t PRV=%.f V",PRV_12);
printf("\nTHe peak reverse voltage for thyristor T3 and T3 is \t PRV=%.f V",PRV_34);
I_T_max_12=E1*sqrt(2)/R;
I_T_max_34=E2*sqrt(2)/R;
I_T_rms_12=I_T_max_12/2;
I_T_rms_34=I_T_max_34/2;
printf("\nTHe maximum current for thyristor T1 and T2 is \t I_T_12=%.1f A",I_T_max_12);
printf("\nTHe maximum current for thyristor T3 and
```

```
T3 is \t I_T_34=%.1f A",I_T_max_34);
printf("\nTHe rms current for thyristor T1 and T2 is \t I_T_12=%.1f A",I_T_rms_12);
printf("\nTHe rms current for thyristor T3 and T3 is \t I_T_34=%.1f A",I_T_rms_34);
```

Scilab code Exa 11.5 Determine rms current of tyristor T1 and T2 rms current of Thyristor T3 and T4 and input power factor

```
// chapter 11
// example 11.5
// fig. 11.9
// Determine rms current of tyristor T1 and T2, rms current of Thyristor T3 and T4 and input power factor
// page-722-723
clear;
clc;
// given
e1=130; // in V
e2=130; // in V
Ep=260; // in V
E0=195; // in V
R=6; // in ohm
alpha=%pi/2; // in radian (firing angle)
// calculate
I1=sqrt((1/(2*%pi*R^2))*2*(e1+e2)^2*integrate('(sin(wt))^2','wt',alpha,%pi)); // calculation of rms current of tyristor T1 and T2
I2=sqrt((1/(2*%pi*R^2))*2*(e1^2)*integrate('(sin (wt))^2','wt',0,alpha)); // calculation of rms current of Thyristor T3 and T4
Iw2=sqrt(2)*I1; // calculation of rms current of second winding
Iw1=sqrt((sqrt(2)*I1)^2+(sqrt(2)*I2)^2); //
```

```
   calculation of rms current of first winding
Ea=e1*Iw1+e2*Iw2; // calculation of volt-ampere
   rating
P0=E0^2/R; // calculation of maximum power
PF=P0/Ea; // calculation of power factor
printf("\nThe rms current of tyristor T1 and T2 is \
   t I1=%.2f A",I1);
printf("\nThe rms current of Thyristor T3 and T4 is
   \t I2=%.2f A",I2);
printf("\nThe power factor is \t\t\t pf=%.3f",PF);
if PF>0 then
    printf(" (lagging)");
else
    printf(" (laeding)");
end
```

Scilab code Exa 11.6 Determine rms output phase voltage input power factor

```
// chapter 11
// example 11.6
// Determine rms output phase voltage, input power
   factor
// page-741-742
clear;
clc;
// given
E_L=210; // in V
R=5; // in ohm
alpha=%pi/3; // in radian (firing angle)
// calculate
Es=E_L/sqrt(3);
E0=sqrt((2*Es^2/(4*%pi))*(integrate('1-cos(2*wt)','
   wt',alpha,%pi)+integrate('1-cos(2*wt)','wt',%pi
   ,(2*%pi))));// calculation of rms output pahse
```

```
   voltage
Ia=E0/R; // calculation of rms current
P0=3*Ia^2*R; // calculation of maximum power
I_L=Ia; // calculation of load current
EA=3*Es*I_L; // calculation of volt-ampere rating
pf=P0/EA; // calculation of power factor
printf("\nThe rms output phase voltage is \t E0=%.3f
    V",E0);
printf("\nThe power factor is \t\t\t pf=%.3f",pf);
if pf>0 then
    printf(" (lagging)");
else
    printf(" (laeding)");
end
// Note part-c of this question is a derivation type
    which i havenot done
```

Scilab code Exa 11.7 Determine required ratings of triacs and thyrstors

```
// chapter 11
// example 11.7
// Determine required ratings of triacs and
   thyrstors
// page-742
clear;
clc;
// given
Es=415; // in V
P0=15; // in kW
// calculate
P0=P0*1E3; // changing unit from kW to W
E_line=Es*sqrt(3); // calculation of line voltage
I_line_rms=P0/E_line; // calculation of rms line
   current for traics
V_offstate=Es*sqrt(2); // calculation of peak
```

```
offstate voltage for traics
I_rms=I_line_rms/sqrt(2); // calculation of required rms current rating for thyristor
printf("\n For Traics");
printf("\n\t\tThe rms line current is \t I_line_rms=%.1f A",I_line_rms);
printf("\n\t\tThe peak offstate voltage is \t V_offstate=%.f V",V_offstate);
printf("\n\n For thyristors");
printf("\n\t\tThe required rms current rating is \t\t I_rms=%.1f A",I_rms);
printf("\n\t\tThe voltage requirement will be same as that of triacs  %.f V",V_offstate);
```

Chapter 12

Resonant Convertors

Scilab code Exa 12.1 Determine circuit turn off time maximum possible output frequency capacitor voltage rms load current output power and average and rms SCR currents

```
// chapter 12
// example 12.1
// Determine circuit turn-off time, maximum possible
    output frequency, capacitor voltage, rms load
   current, output power and average and rms SCR
   currents
// page-755-756
clear;
clc;
// given
Edc=200; // in V (dc input)
Lr=40; // in uH (resonant inductor)
Cr=6.8; // in uF (resonant capacitor)
F=6; // in Khz (frequency of output voltage)
t_q=15; // in us (SCR turn-off time)
R=3; // in ohm
// calculate
Lr=Lr*1E-6; // changing unit from uH to H
Cr=Cr*1E-6; // changing unit from uF to F
```

```
t_q=t_q*1E-6; // changing unit from us to s
F=F*1E3; // changing unit from KHz to Hz
wr=sqrt((1/(Lr*Cr))-R^2/(4*Lr^2));
// since wr=2*%pi*fr , therefore we get ,
fr=wr/(2*%pi);
w0=2*%pi*F;
Toff=(%pi/w0)-(%pi/wr); // calculation of circuit turn-off time
Fmax=1/(2*(t_q+(%pi/wr))); // calculation of maximum possible output frequency
Ec=Edc*((exp(R*%pi/(2*Lr*wr))+1)/(exp(R*%pi/(2*Lr*wr))-1)); // calculation of capacitor voltage
I_L_rms=((Edc+Ec)/(wr*Lr))*sqrt(F*(1-exp(-R*%pi/(2*wr*Lr)))*((Lr/R)-(R/Lr)*(1/((2*wr)^2+(R/Lr)^2)))); // calculation of rms load current
P0=I_L_rms^2*R; // calculation of output power
I_rms_SCR=I_L_rms/sqrt(2); // calculation of rms SCR current
I_L_avg=(2*F*(Edc+Ec)/((wr^2+(R/(2*Lr)^2))*Lr))*1+exp(-R*%pi/(4*wr*Lr)); // calculation of average load current
I_avg_SCR=I_L_avg/2; // calculation of average SCR current
printf("\nThe circuit turn-off time is \t\t\t Toff=%.1f us",Toff*1E6);
printf("\nThe maximum possible output frequency is \t Fmax=%.3f KHz",Fmax*1E-3);
printf("\nThe capacitor voltage is \t\t\t Ec=%.2f V",Ec);
printf("\nThe rms load current is \t\t\t I_L_rms=%.2f A",I_L_rms);
printf("\nThe output power is \t\t\t\t P0=%.2f W",P0);
printf("\nThe rms SCR current is \t\t\t\t I_rms_SCR=%.2f A",I_rms_SCR);
printf("\nThe average SCR current is \t\t\t I_avg_SCR=%.3f A",I_avg_SCR);
// Note : The answers vary slightly due to precise
```

```
calculation
```

Scilab code Exa 12.2 Determine resonant frequency maximum operating frequency Peak thyristor current average thyristor current rms thyristor current rms load current and average supply current

```
// chapter 12
// example 12.2
// fig. 12.8
// Determine resonant frequency, maximum operating frequency,Peak thyristor current, average thyristor current, rms thyristor current, rms load current and average supply current
// page-760-762
clear;
clc;
// given
C1=4; // in uF
C2=4; // in uF
Lr=40; // in uH
R=2; // in ohm
Edc=120; // in V (input voltage)
t_q=20; // in us (SCR turn-off time)
// calculate
Lr=Lr*1E-6; // changing unit from uH to H
t_q=t_q*1E-6; // changing unit from us to s
Ceq=C1+C2;
Ceq=Ceq*1E-6; // changing unit from uF to F
wr=sqrt((1/(Lr*Ceq))-R^2/(4*Lr^2)); // calculation of resonant angular frequency
fr=wr/(2*%pi); // calculation of resonant frequency
tr=1/fr; // calculation of resonant time-period
fr_max=1/(2*t_q); // calculation of maximum frequency
f0=0.4*fr; // calculation of output frequency
```

```
T0=1/f0; // calculation of output period
td=T0/2-tr; // calculation of delay time
tp=(1/wr)*atan(2*wr*Lr/R); // calculation of time at which peak current is obtained
Ec1=Edc/(exp(R*2*%pi/(2*Lr*wr))+1); // calculation of initial capacitor voltage
Ip=(Edc+Ec1)/(wr*Lr)*sin(wr*tp)*exp(-R*tp/(2*Lr)); // calculation of peak current
I_av_SCR=(Edc+Ec1)/(wr*Lr)*(1/T0)*integrate('sin(wr*t)*exp(-R*t/(2*Lr))','t',0,tr/2); // calculation of average thyristor current
I_rms_SCR=(Edc+Ec1)/(wr*Lr)*sqrt((1/T0)*(integrate('(sin(wr*t))^2*exp(-R*t/Lr)','t',0,tr/2))); // calculation of rms thyristor current
I0=2*I_rms_SCR; // calculation of rms load current
P0=I0^2*R; // calculation of output power
Is=P0/Edc; // calculation of average supply current
printf("\nThe resonant frequency is \t\t\t fr=%.3f KHz",fr*1E-3);
printf("\nThe maximum possible operating frequency is \t fr_max=%.f KHz",fr_max*1E-3);
printf("\nThe Peak thyristor current is \t\t\t Ip=%.2f A",Ip);
printf("\nThe average thyristor current is \t\t I_av_SCR=%.3f A",I_av_SCR);
printf("\nThe rms thyristor current is \t\t\t I_rms_SCR=%.2f A",I_rms_SCR);
printf("\nThe rms load current is \t\t\t I0=%.1f A",I0);
printf("\nThe average supply current is \t\t\t Is=%.2f A",Is);
// Note : The answers vary slightly due to precise calculation
```

Scilab code Exa 12.3 Determine resonant frequency maximum operating frequency Peak thyristor current average thyristor current rms thyristor current rms load current and average supply current

```
// chapter 12
// example 12.3
// Determine resonant frequency, maximum operating frequency,Peak thyristor current, average thyristor current, rms thyristor current, rms load current and average supply current
// page-765-767
clear;
clc;
// given
Cr=8; // in uF
Lr=40; // in uH
R=2; // in ohm
Edc=120; // in V (input voltage)
t_q=20; // in us (SCR turn-off time)
// calculate
Lr=Lr*1E-6; // changing unit from uH to H
t_q=t_q*1E-6; // changing unit from us to s
Cr=Cr*1E-6; // changing unit from uF to F
wr=sqrt((1/(Lr*Cr))-R^2/(4*Lr^2)); // calculation of resonant angular frequency
fr=wr/(2*%pi); // calculation of resonant frequency
f0=0.4*fr; // calculation of output frequency
tr=1/fr; // calculation of resonant time-period
T0=1/f0; // calculation of output period
td=T0/2-tr; // calculation of delay time
fr_max=1/(2*t_q); // calculation of maximum frequency
f0_max=fr_max/2; // calculation of maximum foutput frequency
tp=(1/wr)*atan(2*wr*Lr/R); // calculation of time at which peak current is obtained
Ec1=Edc*((1-exp(-R*%pi/(Lr*wr)))/(1+exp(-R*%pi/(Lr*wr)))); // calculation of initial capacitor voltage
```

```
Ip=(Edc+Ec1)/(wr*Lr)*sin(wr*tp)*exp(-R*tp/(2*Lr)); // calculation of peak current
I_avg_SCR=(Edc+Ec1)/(wr*Lr)*(1/T0)*integrate('sin(wr*t)*exp(-R*t/(2*Lr))','t',0,tr/2); // calculation of average thyristor current
I_rms_SCR=(Edc+Ec1)/(wr*Lr)*sqrt((1/T0)*(integrate('(sin(wr*t))^2*exp(-R*t/Lr)','t',0,tr/2))); // calculation of rms thyristor current
I0=2*I_rms_SCR; // calculation of rms load current
P0=I0^2*R; // calculation of output power
Is=P0/Edc; // calculation of average supply current
printf("\nThe resonant frequency is \t\t\t fr=%.2f KHz",fr*1E-3);
printf("\nThe maximum possible operating frequency is \t fr_max=%.f KHz",fr_max*1E-3);
printf("\nThe maximum possible output frequency is \t f0_max=%.1f KHz",f0_max*1E-3);
printf("\nThe Peak thyristor current is \t\t\t Ip=%.3f A",Ip);
printf("\nThe average thyristor current is \t\t I_avg_SCR=%.3f A",I_avg_SCR);
printf("\nThe rms thyristor current is \t\t\t I_rms_SCR=%.2f A",I_rms_SCR);
printf("\nThe rms load current is \t\t\t I0=%.1f A",I0);
printf("\nThe average supply current is \t\t\t Is=%.2f A",Is);
// Note : 1. The answer for I_avg_SCR, I_rms_SCR, I0, P0 and Is are wrong in the book due to the use of wrong value of Ec1 in the calculation
//        2.The answers vary slightly due to precise calculation
```

Scilab code Exa 12.4 Design E class resonant invertor

```
// chapter 12
// example 12.4
// Design E-class resonant invertor
// page-769-770
clear;
clc;
// given
R=12; // in ohm
Edc=24; // in V (DC voltage)
fs=20; // in KHz (switching frequency)
Q=7; // assumption for quality factor as done in the book
// calculate
fs=fs*1E3; // changing unit from KHz to Hz
ws=2*%pi*fs; // calculation of switching angular frequency
Ldc=0.4*R/ws; // calculation of source inductance
C1=2.165/(R*ws); // calculation of resonance element
// since Q=ws*Lr/R, therefore we get,
Lr=Q*R/ws; // calculation of inductance from series resonance
// since ws*Lr-1/(ws*Cr)=0.353*R, therefore we get,
Cr=(1/(ws*Lr-0.353*R))/ws; // calculation of capacitance from series resonance
e=(R/2)*sqrt(Cr/Lr); // calculation of damping factor
f0=1/(2*%pi*sqrt(Lr*Cr)); // calculation of resonance frequency
printf("\nThe source inductance is \t\t\t Ldc=%.1f uH",Ldc*1E6);
printf("\nThe resonance element is \t\t\t C1=%.1f uF",C1*1E6);
printf("\nThe inductance from series resonance is \t Lr=%.1f uH",Lr*1E6);
printf("\nThe capacitance from series resonance is \t Cr=%.2f uF",Cr*1E6);
printf("\nThe resonance frequency is \t\t\t f0=%.2f KHz",f0*1E-3);
```

```
// Note :The answers vary slightly due to precise calculation
```

Scilab code Exa 12.5 Determine values of resonant components Lr and Cr peak switching current peak voltage rating of the resonant capacitor

```
// chapter 12
// example 12.5
// Destermine values of resonant components Lr and Cr, peak switching current, peak voltage rating of the resonant capacitor
// page-778-780
clear;
clc;
// given
Edc=18; // in V (unregulated DC voltage)
E0=12; // in V (output voltage)
fr=1; // in MHz
P0=12; // in W
// calculate
fr=fr*1E6; // changing unit from MHz to Hz
// since P0=E0*I0, therefore we get,
I0=P0/E0; // calculation of load current
Zr=Edc/I0; // calculation of load impedance
// since Zr=sqrt(Lr/Cr) and fr=1/(2*%pi*sqrt(Lr*Cr)), multiplying two equations, we get
// Zr*fr=1/(2*%pi*Cr), therefore
Cr=1/(2*%pi*Zr*fr); // calculation of resonant capacitance
Lr=Zr^2*Cr; // calculation of resonant inductance
is_p=I0+Edc*sqrt(Cr/Lr); // calculation of peak switch current
Vc_peak=2*Edc; // calculation of peak voltage rating of capacitor
printf("\nThe resonant capacitance is \t\t\t Cr=%.5f
```

```
uF",Cr*1E6);
printf("\nThe resonant inductance is \t\t\t Lr=%.2f uH",Lr*1E6);
printf("\nThe peak switch current is \t\t\t is_p=%.2f A",is_p);
printf("\nThe peak voltage rating of capacitor is \t Vc_peak=%.f V",Vc_peak);
// Note :The answer in the book are wrong due to calculation mistake while calculating Cr and hence all other answer are also wrong
```

Scilab code Exa 12.6 Determine values of resonant components Lr and Cr peak voltage rating of the resonant capacitor and charging and discharging period of capcaitor

```
// chapter 12
// example 12.6
// Determine values of resonant components Lr and Cr , peak voltage rating of the resonant capacitor and charging and discharging period of capcaitor
// page-785-786
clear;
clc;
// given
Edc=35; // in V
E0=24; // in V (output voltage)
fr=500; // in KHz
P0=24; // in W
// calculate
fr=fr*1E3; // changing unit from KHz to Hz
// since P0=E0*I0 , therefore we get ,
I0=P0/E0; // calculation of load current
Zr=Edc/I0; // calculation of load impedance
// since Zr=sqrt(Lr/Cr) and fr=1/(2*%pi*sqrt(Lr*Cr)) , multiplying two equations , we get
```

```
// Zr*fr=1/(2*%pi*Cr), therefore
Cr=1/(2*%pi*Zr*fr); // calculation of resonant capacitance
Lr=Zr^2*Cr; // calculation of resonant inductance
Vs_p=Edc+I0*Zr; // calculation of peak voltage rating of capacitor
t1=Edc*Cr/I0; // calculation of charging period of capcaitor
t3=sqrt(Lr*Cr)*asin(Edc*sqrt(Cr/Lr)/I0); // calculation of  discharging period of capcaitor
printf("\nThe resonant capacitance is \t\t\t Cr=%.4f uF",Cr*1E6);
printf("\nThe resonant inductance is \t\t\t Lr=%.2f uH",Lr*1E6);
printf("\nThe peak voltage rating of capacitor is \t Vs_p=%.f V",Vs_p);
printf("\nThe charging period of capcaitor is \t\t t1=%.3f us",t1*1E6);
printf("\nThe discharging period of capcaitor is \t\t t3=%.3f us",t3*1E6);
// Note :The answer vary slightly due to precise calculation
```

Scilab code Exa 12.7 Design a ZVS three level PWM convertor

```
// chapter 12
// example 12.7
// Design a ZVS three-level PWM convertor
// page-791-792
clear;
clc;
// given
Edc=300; // in V
E0=60; // in V (output voltage)
I0=10; // in A
```

```
fs=120; // in KHz
Deff=0.5; // effective duty ratio
del_D=25; // assumption for reduction in duty ratio in percentage of duty ratio as done in the book
TLI=3.46; // in uH (assuming transformer leakage inductance as done in the book)
C=500; // in pF (assuming intrinsic capacitance of MOSFET as done in the book)
// calculate
fs=fs*1E3; // changing unit from KHz to Hz
TLI=TLI*1E-6; // changing unit from uH to H
C=C*1E-12; // changing unit from pF to F
// since Deff=2*n*E0/Edc, therefore we get,
n=Deff*Edc/(2*E0); // calculation of transformer turns ratio
// since Deff=D-del_D*D, therefore we get,
D=Deff/(1-del_D/100); // calculation of duty ratio
Lr=((del_D*D/100)*Edc/2)/(4*(I0/n)*fs); // calculation of resonant inductance
Lr_eff=Lr+TLI; // calculation of effecive inductance
I0_min=(n*Edc/2)*sqrt(1.5*C/Lr_eff); // calculation of minimum load current
I_Lr=I0/n; // calculation of inductor current
X_Lr=2*%pi*fs*Lr_eff; // calculation of inductive reactance
V_Lr=I_Lr*X_Lr; // calculation of voltage drop across Lr
// since E0=Deff*Es, therefore we get
Es=E0/Deff; // calculation of secondary voltage
Ep=n*Es; // calculation of primary voltage
VA=Ep*I_Lr; // calculation of transformer volt-ampere
printf("\nThe duty ratio is \t\t D=%.2f",D);
printf("\nThe resonant inductance is \t Lr=%.2f uH",Lr*1E6);
printf("\nThe effective inductance is \t Lr_eff=%.f uH",Lr_eff*1E6);
printf("\nThe minimum load current is \t I0_min=%.2f
```

```
A",I0_min);
printf("\nThe inductor current is \t I_Lr=%.f A",I_Lr);
printf("\nThe voltage drop across Lr is \t V_Lr=%.2f V",V_Lr);
printf("\nThe secondary voltage is \t Es=%.f V",Es);
printf("\nThe primary voltage is \t\t Ep=%.f V",Ep);
printf("\nThe transformer volt-ampere is \t VA=%.f VA",VA);
// Note :The answer vary slightly due to precise calculation
```

Chapter 13

Protection and Cooling of Power Switching Devices

Scilab code Exa 13.1 Compute the values of di dt inductor and the snubber circuit component Rs and Cs

```
// chapter 13
// example 13.1
// fig. 13.11
// Compute the values of di/dt inductor and the
   snubber circuit component Rs and Cs
// page-807-808
clear;
clc;
// given
E=400; // in V
di_dt=50; // in A/us
dv_dt=200; // in V/us
// calculate
// since di/dt=E/L, therefore we get
L=E/di_dt; // calculation of
// since dV/dt=Rs*di/dt, therefore we get
Rs=dv_dt/di_dt; // calculation of
Rs=10; // assuming the desired values
```

```
Cs=0.1; // in uF assuming the desired value
L=E*Rs/(dv_dt);
printf("\nThe value of di/dt inductor is \t\t L=%.f uH",L);
printf("\nThe snubber circuit component are \t Rs=%.f ohm \t Cs=%.1f uF",Rs,Cs);
```

Scilab code Exa 13.2 Compute the required parameters snubber circuit

```
// chapter 13
// example 13.2
// Compute the required parameters snubber circuit
// page-809
clear;
clc;
// given
Em=380; // in V
dv_dt=50; // in V/us
L=0.1; // in mH
sigma=0.65; // assumption for damping factor as done in the book
// calculate
dv_dt=dv_dt/1E-6; // changing unit from V/us to V/s
L=L*1E-3; // changing unit from mH to H
C=(1/(2*L))*(0.564*Em/dv_dt)^2;
R=2*sigma*sqrt(L/C);
printf("\nThe capacitance is \t C=%.3f uF",C*1E6);
printf("\nThe resistance is \t R=%.2f ohm",R);
// Note :The answer vary slightly due to precise calculation
```

Scilab code Exa 13.3 Design a snubber circuit

```
// chapter 13
// example 13.3
// Design a snubber circuit
// page-809
clc;
clear;
// given
VA=5; // in kVA
Es=120; // in V (secondary voltage)
f=400; // in Hz (switching frequency)
L=100; // in uH
Ep=200; // in V (peak transient voltage)
sigma=0.75; // assumption for damping factor as done
    in the book
// calculate
VA=VA*1E3; // changing unit from kVA to VA
L=L*1E-6; // changing unit from uH to H
C=10*VA*60/(Es^2*f); // calculation of capacitance
   in uF
Esp=Es*sqrt(2); // calculation of peak switching
   voltage ratio
Ep_Esp=Ep/Esp; // calculation of peak transient
   voltage Vs peak switching voltage ratio
R=2*sigma*sqrt(L/(C*1E-6)); // calculation
   resistance
printf("\nThe capacitance is \t C=%.2f uF",C);
printf("\nThe peak transient voltage Vs peak
   switching voltage ratio is \t %.2f",Ep_Esp);
printf("\nThe resistance is \t R=%.1f ohm",R);
// Note :The answer vary slightly due to precise
   calculation
```

Scilab code Exa 13.4 Compute initial capacitor voltage magnetizing current and capacitor value to limit the transient capacitor voltage

```
// chapter 13
// example 13.4
// fig. 13.13
// Compute initial capacitor voltage, magnetizing current and capacitor value to limit the transient capcaitor voltage to Ep=330 V
// page-809-810
clear;
clc;
// given
Es=140; // in V
f=50; // in Hz
Lm=3; // in mH
theta=180; // in degree
Ep=330; // in V
// calculate
Lm=Lm*1E-3; // changing unit from mH to H
Em=sqrt(2)*Es; // calculation of peak voltage
Vc_0=Em*sind(theta); // calculation of initial capacitor voltage
w=2*%pi*f; // calculation of angular frequency
Im=(-Em/(w*Lm))*cosd(theta); // calculation of magnetizing current
C=Im*Em/(Ep^2*w); // calculation of capacitor value to limit the transient capcaitor voltage
printf("\nThe initial capacitor voltage is \t\t\t\t\t\t Vc_0= %.f V",abs(Vc_0));
printf("\nThe magnetizing current is \t\t\t\t\t\t Im= %.f A",Im);
printf("\nThe capacitor value to limit the transient capcaitor voltage to Ep=%.f V is \t C= %.1f uF",Ep,C*1E6);
// Note :The answer vary slightly due to precise calculation
```

Scilab code Exa 13.5 Determine the snubber component values for critical damping value of R and power loss

```
// chapter 13
// example 13.5
// fig. 13.14
// Determine the snubber component values for critical damping, value of R and power loss
// page-810-811
clear;
clc;
// given
Edc=30; // in V
fs=40; // in kHz
i_L=30; // in A (load current)
t_r=80; // in ns (rise time)
t_f=30; // in ns (fall time)
neta=10; // in % (ratio of maximum discharge current to load current)
// calculate
fs=fs*1E3; // changing unit from kHz to Hz
t_r=t_r*1E-9; // changing unit from ns to s
t_f=t_f*1E-9; // changing unit from ns to s
L=Edc*t_r/i_L; // calculation of inductance
C=i_L*t_f/Edc; // calculation of capacitance
R=sqrt(4*L/C); // calculation of resistance for critical damping
R1=Edc/((neta/100)*i_L); // calculation of resistance if maximum discharge current is limited to 10 % of load current
Ploss=(1/2)*C*Edc^2*fs; // calculation of power loss
printf("\nThe inductance is \t L=%.f nH",L*1E9);
printf("\nThe capacitance is \t C=%.f nF",C*1E9);
printf("\nThe resistance is \t R=%.2f ohm",R);
printf("\nThe resistance if maximum discharge current is limited to 10 %% of load current is \t R=%.f ohm",R1);
printf("\nThe power loss due to RC snubber circuit
```

```
is \t Ploss=%.2f W',Ploss);
// Note :The answer vary slightly due to precise calculation
```

Scilab code Exa 13.6 Design a suitable selenium voltage protection circuit and compute energy dissipated per plate

```
// chapter 13
// example 13.6
// Design a suitable selenium voltage protection circuit and compute energy dissipated per plate
// page-814
clear;
clc;
// given
Edc=440; // in V
Imax=120; // in A
R=0.175; // in ohm
T=25; // in ms (time constant)
V=40; // in V
// calculate
T=T*1E-3; // changing unit from ms to s
n=Edc/V; // calculation of number of plates in each branch
N=3*n; // calculation of total number of plates
L=T*R; // calculation of armature circuit inductance
E_stored=(1/2)*L*Imax^2; // calculation of energy stored in armature
E_dissipated=E_stored/11; // calculation of energy dissipated per plate
printf("\nThe number of plates in each branch is \t n=%.f",n);
printf("\nThe total number of plates is \t N=%.f",N);
printf("\nThe armature circuit inductance is \t L=%
```

```
.2f mH",L*1E3);
printf("\nThe energy stored in armature is \t E_stored=%.1f W",E_stored);
printf("\nThe energy dissipated per plate is \t E_dissipated=%.2f W",E_dissipated);
// Note : The value of time constant given in the book is 25 ms but in calculation the author has used 30 ms. Thats why answers in the book are wrong
```

Scilab code Exa 13.7 Design a suitable circuit

```
// chapter 13
// example 13.7
// Design a suitable circuit
// page-814
clear;
clc;
// given
Edc=200; // in V
V=30; // in V
// calculate
n=Edc/V; // calculation of number of plates in each branch
printf("\nThe number of plates is \t n=%.f is series",n);
// Note :The answer vary slightly due to precise calculation
```

Scilab code Exa 13.8 Calculate the peak value of the fault current

```
// chapter 13
// example 13.8
```

```
// Calculate the peak value of the fault current
// page-825-826
clear;
clc;
// given
I_2_t=150; // in A^2 sec
f=50; // in Hz
tc=6; // in ms
// calculate
tc=tc*1E-3; // changing unit from ms to s
// since tc=3*I^2*t/Ip^2, therefore we get,
Ip=sqrt(3*I_2_t/tc); // calculation of peak value of the fault current
printf("\nThe peak value of the fault current is \t Ip=%.2f A",Ip);
// Note :The answer vary slightly due to precise calculation
```

Scilab code Exa 13.9 Compute the required fusing time

```
// chapter 13
// example 13.9
// Compute the required fusing time
// page-826-827
clear;
clc;
// given
I=250; // in A (non-repetitive surge current)
f=50; // in Hz
Ip=400; // in A (peak short circuit current)
// calculate
t=1/f; // calculation of time period
I_subcycle=sqrt(I^2*(1/100)/t); // calculation of subcycle surge current rating
tc=3*I_subcycle^2/Ip^2; // calculation of required
```

```
fusing time
printf(”\nThe required fusing time is \t tc=%.2f ms”,tc*1E3);
// Note :The formula used in the book is wrong. The value of I_subcycle also needs to be squared but is not done in the book. Thats why the answer in the book is wrong.
```

Scilab code Exa 13.10 What is the minimum value of load inductance and what will be the new value of L

```
// chapter 13
// example 13.10
// What is the minimum value of load inductance and what will be the new value of L
// page-827
clear;
clc;
// given
di_dt=15; // in A/us
Edc=150; //  in V
R=620; // in ohm
R_L=62; // in ohm
// calculate
di_dt=di_dt/1E-6; // changing unit from A/us to A/s
L=Edc/di_dt; // calculation of minimum value of load inductance
L1=Edc*((R+R_L)/R)/di_dt; // calculation of new value of L
printf(”\nThe minimum value of load inductance that will protect the device is \t L=%.f uH”,L*1E6);
printf(”\nThe required new value of inductance \t\t\t\t\t L=%.f uH”,L1*1E6);
// Note :The answer vary slightly due to precise calculation
```

Scilab code Exa 13.11 Compute the values of RF filter components

```
// chapter 13
// example 13.11
// Compute the values of RF filter components
// page-833
clear;
clc;
// given
R_L=120; // in ohm
f0=50; // in KHz
// calculate
f0=f0*1E3; // changing unit from KHz to Hz
L=R_L/(2*%pi*f0); // calculation of inductance
C=1/(2*%pi*R_L*f0); // calculation of capacitance
printf("\nThe inductance is \t L=%.2f uH",L*1E6);
printf("\nThe capacitance is \t C=%.3f uF",C*1E6);
// Note :The answer vary slightly due to precise
   calculation
```

Scilab code Exa 13.12 Determine the value of thermal resistance heat sink and temperature at base

```
// chapter 13
// example 13.12
// Determine the value of thermal resistance, heat
   sink and temperature at base
// page-840
clear;
clc;
// given
```

```
Pa=30; // in W (power loss)
Tc=125; // in degree C (junction temperature)
Ta=40; // in degree C (ambient temperature)
theta_CJ=0.7; // in degree C/W (junction to heat-
   sink thermal resistance)
// calculate
theta_CA=(Tc-Ta)/Pa; // calculation of total thermal
    resistance
theta_CS=theta_CA-theta_CJ; // calculation of
   thermal resistance of heat sink
T_base=Ta+(Pa*theta_CS); // calculation of
   temperature at base
printf("\nThe  total thermal resistance is \t\t
   theta_CA=%.2f degree C/W",theta_CA);
printf("\nThe  thermal resistance of heat sink is \t
    theta_CS=%.2f degree C/W",theta_CS);
printf("\nThe temperature at base is \t\t\t T_base=%
   .f degree C",T_base);
// Note :The answer vary slightly due to precise
   calculation
```

Scilab code Exa 13.13 Determine junction temperature

```
// chapter 13
// example 13.13
// Determine junction temperature
// page-840
clear;
clc;
// given
E=440; // in V
I0=960; // in A
Ta=80; // in ambient temperature
Pa=150; // in W (on state power loss)
Q_JC=0.15; // in degree C/W
```

```
Q_CS=0.075; // in degree C/W
Q_SA=0.45; // in degree C/W
// calculate
T_J=Ta+Pa*(Q_JC+Q_CS+Q_SA); // calculation of
    junction temperature
printf("\nThe junction temperature is \t T_J=%.2f
    degree C",T_J);
// Note :The answer vary slightly due to precise
    calculation
```

Scilab code Exa 13.14 Determine total average power loss and percentage increase in deivce rating

```
// chapter 13
// example 13.14
// Determine total average power loss and percentage
     increase in deivce rating
// page-840-841
clear;
clc;
// given
T_J=150; // in degree C (maximum junction
    temperature)
Q_JC=0.015; // in degree C/W
Q_CS=0.08; // in degree C/W
Ta_1=60; // in degree (heat sink temperature1)
Ta_2=50 // in degree (heat sink temperature2)
// calculate
Pav1=(T_J-Ta_1)/(Q_JC+Q_CS);
Pav2=(T_J-Ta_2)/(Q_JC+Q_CS);
percent_increase=((sqrt(Pav2)-sqrt(Pav1))/sqrt(Pav1)
    )*100;
printf("\nThe total average power loss is \t\t\t
    Pav1=%.2f W",Pav1);
printf("\nThe required percentage increase in deivce
```

```
rating is \t %.2f %%",percent_increase);

// Note :The answer vary slightly due to precise calculation
```

Scilab code Exa 13.15 Plot instantaneous junction temperature

```
// chapter 13
// example 13.15
// fig. 13.40
// Plot instantaneous junction temperature
// page-841-842
clear;
clc;
clf;
// given
P2=0, P4=0, P6=0; // in W
P1=1000, P3=1400, P5=700; // in W
t1=1, t3=1, t5=1; // in ms
theta1=0.035, theta3=0.035, theta5=0.035; // in degree C/W
t2=0.5, t4=0.5, t6=0.5; // in ms
theta2=0.025, theta4=0.025, theta6=0.025; // in degree C/W
// calculate
del_TJ1=theta1*P1;
del_TJ1_5=del_TJ1-theta2*P1;
del_TJ2_5=del_TJ1_5+theta3*P3;
del_TJ3=del_TJ2_5-theta4*P3;
del_TJ4=del_TJ3+theta5*P5;
del_TJ4_5=del_TJ4-theta6*P5;
printf("\n\ndel_TJ1=%.f degree C",del_TJ1);
printf("\n\ndel_TJ1_5=%.f degree C",del_TJ1_5);
printf("\n\ndel_TJ2_5=%.f degree C",del_TJ2_5);
printf("\n\ndel_TJ3=%.f degree C",del_TJ3);
```

```
printf("\n\ndel_TJ4=%.f degree C",del_TJ4);
printf("\n\ndel_TJ4_5=%.f degree C",del_TJ4_5);
del_TJ=[del_TJ1 del_TJ1_5 del_TJ2_5 del_TJ3 del_TJ4 del_TJ4_5];
plot2d(del_TJ,nax=[1,6,1,7],rect=[0,0,5,60]);
xlabel("t (ms)");
ylabel("del_TJ  (degree C)");
title("Junction temperature rise plot");
// Note :There is calculation mistake in the book while calculating del_TJ_3 and hence all other values are also affected. Thats why answers in the book are wrong
```

Scilab code Exa 13.16 Determine the heat sink required

```
// chapter 13
// example 13.16
// Determine the heat-sink required
// page-841-842
clear;
clc;
// given
Vcc=100; // in V
V_CE_on=1; // in V
I_on=20; // in A
t_on=1; // in us
t_off=2; // in us
fs=10; // in kHz
del=0.9; // duty cycle
T_A=35; // in degree C
T_J_max=125; // in degree C
theta_JC=0.7; // in degree C/W
theta_CS=0.1; // in degree C/W
// calculate
t_on=t_on*1E-6; // changing unit from us to s
```

```
t_off=t_off*1E-6; // changing unit from us to s
fs=fs*1E3; // changing unit from kHz to Hz
P_cond=V_CE_on*I_on*del; // calcultion of average power
P_loss=Vcc*I_on*(t_on+t_off)*fs/2; // calcultion of switching power loss
P_diss=P_cond+P_loss; // calcultion of total power loss
// since P_diss=(T_J_max-T_A)/(theta_JC+theta_CS+theta_SA), therefore we get
theta_SA=((T_J_max-T_A)/P_diss)-theta_JC-theta_CS; // calcultion of heat-sink required
printf("\nThe required heat sink is \t theta_SA=%.1f degree C/W",theta_SA);

// Note :The answer vary slightly due to precise calculation
```

Scilab code Exa 13.17 Determine the thermal resistance of the heat sink

```
// chapter 13
// example 13.17
// Determine the thermal resistance of the heat sink
// page-842-843
clear;
clc;
// given
I_D=20; // in A
t_on=10; // in us
del=0.1; // in % (duty cycle)
T_A=40; // in degree C
T_J_max=150; // in degree C
theta_JC=1.5; // in degree C/W
R_ds_on=5; // in ohm
T_pr=0.03; // assumption as done in the book
```

```
// calculate
t_on=t_on*1E-6; // changing unit from us to s
P_cond=I_D^2*R_ds_on;
P_peak=P_cond;
// since P_peak=(T_J_max-T_C)/(T_pr*theta_JC),
   therefore we get
T_C=T_J_max-P_peak*T_pr*theta_JC;
P_diss=(del/100)*P_peak;
// since P_diss=(T_C-T_A)/theta_CA, therefore we get
theta_CA=(T_C-T_A)/P_diss;
printf("\nThe thermal resistance of the heat sink is
    \t theta_CA=%.f degree C/W",theta_CA);

// Note :The answer vary slightly due to precise
   calculation
```

Scilab code Exa 13.18 Determine the maximum power disspation

```
// chapter 13
// example 13.18
// Determine the maximum power disspation
// page-843
clear;
clc;
// given
Cs=0.2; // in J/degree C (thermal capacity)
theta=0.7; // in degree C/W (thermal resistance)
T_J=40; // in degree C (junction temperature)
t=0.1; // in s
// calculate
power_diss_per_degreeC=1/theta; // calculation of
   power dissipation per degree Celsius rise
tou=Cs/power_diss_per_degreeC; //  calculation of
   thermal time constant
// since T_J=T_J_max*(1-exp(-t/tou)), therefore we
```

```
get
T_J_max=T_J/(1-exp(-t/tou)); //  calculation of maximum junction temperature
P_max_diss=T_J_max*power_diss_per_degreeC; // calculation of maximum power disspation
printf("\nThe maximum power disspation is \t P_max_diss=%.f W",P_max_diss);
// Note :The answer vary slightly due to precise calculation
```

Chapter 14

Control of D C Drives

Scilab code Exa 14.1 Determine motor terminal voltage and the value of flux as a percentage of rated flux

```
// chapter 14
// example 14.1
// Determine motor terminal voltage and the value of
   flux as a percentage of rated flux
// page-862-863
clear;
clc;
// given
E=220; // in V
N1=960; // in rpm
Ia=80; // in A (armature current)
Ra=0.06; // in ohm (armature resistance)
N2=620; // in rpm
N3=1200; // in rpm
// calculate
Eb1=E-Ia*Ra; // calculation of bcak emf at 960 rpm
w1=N1*2*%pi/60; // calculation of rated speed
// since K1*phi1*w1=Eb1, therefore we get
K1_phi1=Eb1/w1;
// since Eb1/Eb2=N1/N2,  therefore we get
```

```
Eb2=(N2/N1)*Eb1; // calculation of back emf at 620
Em=Eb2+Ia*Ra; // calculation of motor terminal
    voltage
w3=N3*2*%pi/60;
// since Eb3=Ka*phi2*w3 and phi2=k*phi1, therefore
    we get
// Eb3=K*phi1*w3  (i)
// since T1=T2 or Ka*phi1*Ia1=Ka*phi2*Ia2 or Ia2=(
    phi1/phi2)*Ia1 or
//   Ia2=Ia1/K   (ii)
// since E=Eb3+Ia2*Ra , therefore Ia2=(E-Eb3)/Ra
        (iii)
// from (i) (ii) and (iii), we get  E=K*K1_phi1*w3+(
    Ia1/K)*Ra
// and hence solving for K we get a quadratic
    equation (K1_phi1*w3)K^2-(E)K+Ia*Ra=0
// solving the quadratic equation
K1=(-(-E)+sqrt((-E)^2-4*K1_phi1*w3*Ia*Ra))/(2*
    K1_phi1*w3); // calculation of value of flux1
K2=(-(-E)-sqrt((-E)^2-4*K1_phi1*w3*Ia*Ra))/(2*
    K1_phi1*w3); // calculation of value of flux2
printf("\nThe motor terminal voltage is \t Em=%.2f V
    ",Em);
printf("\nThe value of flux is \t K1=%.1f ",K1);
printf("\nThe value of flux is \t K2=%.3f ",K2);
printf("\nThe feasible value of K is %.1f",K1)
// Note :The answer vary slightly due to precise
    calculation
```

Scilab code Exa 14.2 Compute the speed at which motor can hold the load by regenerative braking

```
// chapter 14
// example 14.2
// Compute the speed at which motor can hold the
```

```
load by regenerative braking
// page-863-864
clear;
clc;
// given
T=100; // in Nm (torque)
E=220; // in V
N1=960; // in rpm
Ia=80; // in A (armature current)
Ra=0.06; // in ohm (armature resistance)
// calculate
Eb1=E-Ia*Ra; // calculation of bcak emf at 960 rpm
w1=N1*2*%pi/60; // calculation of rated speed
T1=Eb1*Ia/w1; // calculation of rated motor torque
// since T=Ka*phi*Ia, therefore we get
Ka_phi=T1/Ia;
Ia2=T/Ka_phi; // calculation of armature current at T=100 Nm
Eb2=E+Ia2*Ra; // calculation of corresponding back emf
w2=(Eb2/Eb1)*w1; // calculation of speed at which motor can hold the load by regenerative braking
N2=w2*(60/(2*%pi));
printf("\nThe speed at which motor can hold the load by regenerative braking is \t w2=%.2f rad/s or \t N2=%.f rpm",w2,N2)
// Note :The value of torque given is T=100 Nm but the author has used T=1000 Nm in the calculation. Thats why the answer in the book is wrong
```

Scilab code Exa 14.3 Compute the field current firing angle of the convertor and the input power factor

```
// chapter 14
// example 14.3
```

```
// Compute the field current, firing angle of the convertor and the input power factor
// page-864-865
clear;
clc;
// given
E=230; // in V
f=50; // in Hz
La=0.06; // in H (armature inductance)
Ra=0.3; // in ohm (armature resistance)
Ka=0.9; // in V/A rad/s (motor voltage constant)
Rf=104; // in ohm (field resistance)
T_L=50; // in Nm (load torque)
N=800; // in rpm (speed)
alpha0=0; // in radian (firing angle for maximum field voltage)
// calculate
Em=E*sqrt(2); // calculation of peak voltage
Ef=(Em/%pi)*(1+cos(alpha0)); // calculation of maximum field voltage
If=Ef/Rf; // calculation of field current
Ia=T_L/(Ka*If); // calculation of armature current
w=N*(2*%pi/60); // calculation of angular speed
Eb=Ka*w*If; // calculation of back emf
Ea=Eb+Ia*Ra; // calculation of armature voltage
// since Ea=(Em/%pi)*(1+cos(alpha)), therefore we get
alpha=acosd(Ea*(%pi/Em)-1); // calculation of firing angle
P0=Ea*Ia; // calculation of output power
Pa=P0; // calculation of power from supply
Irms=sqrt((2/(2*%pi))*Ia^2*integrate('1','wt',alpha*(%pi/180),%pi)); // calculation of rms current
EI=E*Irms; // calculation of volt-ampere rating
Pf=Pa/(EI); // calculation of power factor
printf("\nThe field current is \t\t\t If=%.2f A",If);
printf("\nThe firing angle of the convertor is \t
```

```
   alpha=%.f degree",alpha);
printf("\nThe power factor is \t\t\t Pf=%.2f",Pf);
if Pf>0 then
    printf(" (lagging)");
else
    printf(" (leading)");
end
```

Scilab code Exa 14.4 Determine motor torque speed of the motor supply power factor firing angle and power fed back to the supply

```
// chapter 14
// example 14.4
// fig. 14.13
// Determine motor torque, speed of the motor,
   supply power factor, firing angle and power fed
   back to the supply
// page-865-866
clear;
clc;
// given
HP=10; // in HP (power of motor)
E=210; // in V
N=1000; // in rpm (speed)
Ia=30; // in A (armature current)
Ra=0.25; // in ohm (armature resistance)
Es=230; // in V (supply voltage)
Ka_phi=0.172; // in V/rpm (motor voltage constant)
alpha=45; // in degree
// calculate
Em=Es*sqrt(2); // calculation of peak voltage
// part- (a)
Ka_phi_rad=Ka_phi*(60/(2*%pi)); // changing unit
   from V/rpm to V/rad/s
T=Ka_phi_rad*Ia; // calculation of motor torque
```

```
Ea=(2*Em/%pi)*cosd(alpha); // calculation of armature voltage
Eb=Ea-Ia*Ra; // calculation of back emf
N=Eb/Ka_phi; // calculation of speed
EI=Es*Ia; // calculation of volt-ampere rating
Ps=Ea*Ia; // calculation of supplied power
Pf=Ps/EI; // calculation of power factor
printf("\nThe motor torque is \t\t T=%.1f Nm",T);
printf("\nThe speed of the motor is \t N=%.2f rpm",N);
printf("\nThe supply power factor is \t Pf=%.2f",Pf);
// part  (b)
Ea2=-Eb+Ia*Ra; // calculation of back emf when in regenerative action Here minus sign indicates regenerative action
// since Ea=(2*Em/%pi)*cosd(alpha), therefore we get
alpha2=acosd(Ea2*(%pi/(2*Em))); // calculation of corresponding firing angle
Pg=Eb*Ia; // calculation of power from dc machine
Pr=Ia^2*Ra; // calculation of power lost in armature resistance
Ps=Pg-Pr; // calculation of supplied power
printf("\n\nThe firing angle to keep the motor current at its rated value is \t alpha2=%.2f degree",alpha2);
printf("\nThe fpower fed back from the supply is \t Ps=%.1f W",Ps);
// Note :The answers vary due to precise calculations
```

Scilab code Exa 14.5 What should be firing angle different speeds and calculate the motor speed

```
// chapter 14
```

```
//example 14.5
//What should be firing angle different speeds and calculate the motor speed
//page-866-867
clear;
clc;
//given
Ea=210; //in V
N=1200; //in rpm (speed)
Ia=10; //in A (armature current)
Ra=1.5; //in ohm (armature resistance)
Es=230; //in V (supply voltage)
N1=800; //in rpm (rated torque)
N2=1200; //in rpm (braking torque)
alpha3=165; //in degree
//calculate
Em=Es*sqrt(2); //calculation of peak voltage
Eb=Ea-Ia*Ra; //calculation of rated back emf
w=abs(N2)*(2*%pi/60); //calculation of rated angular speed
Ka_phi=Eb/w; //calculation of motor voltage constant
Eb1=(N1/N2)*Eb; //calculation of back emf at speed N1
Ea1=Eb1+Ia*Ra; //calculation of armature voltage at speed N1
//since Ea=(2*Em/%pi)*cosd(alpha), therefore we get
alpha1=acosd(Ea1*(%pi/(2*Em))); //calculation of firing angle at speed N1
Ea2=-Eb+Ia*Ra; //calculation of armature voltage at speed N2
alpha2=acosd(Ea2*(%pi/(2*Em))); //calculation of firing angle at speed N2
Ea3=(2*Em/%pi)*cosd(alpha3); //calculation of armature voltage at aplha3
Eb3=Ea3-Ia*Ra; //calculation of back emf at alpha3
w3=abs(Eb3/Ka_phi); //calculation of speed at alpha3
```

```
N3=w3*(60/(2*%pi)); // calculation of speed in rpm
printf("\nThe firing angle at N1=%.f rpm is \t alpha1=%.2f degree",N1,alpha1);
printf("\nThe firing angle at N2=%.f rpm is \t alpha2=%.2f degree",N2,alpha2);
printf("\n\nThe speed of the motor is \t w3=%.2f rad/s or \t N3=%.1f rpm",w3,N3);
// Note : The answers vary slightly due to precise calculations
```

Scilab code Exa 14.6 Determine the motor speed

```
// chapter 14
// example 14.6
// Determine the motor speed
// page-867-868
clear;
clc;
// given
E=240; // in V
alpha1=110; // in degree (triggering angle)
alpha2=50; // in degree
T=1.8; // in Nm (torque)
Ka_phi=1; // in Nm/A (torque motor characteristics)
Ra=6; // in ohm (armature resistance)
// calculate
Em=E*sqrt(2); // calculation of peak voltage
Ia=T/Ka_phi;
VARIABLE=(1/%pi)*Em*integrate('sin(wt)','wt',alpha1*(%pi/180),%pi);
// since Ea=(VARIABLE+Eb*%pi*(alpha1-alpha2)/(%pi*180)) and Eb=Ea-Ia*Ra, putting the value in it, we get
// Eb=VARIABLE+Eb*(alpha1-alpha2)/180)-Ia*Ra, or
// Eb-Eb*(alpha1-alpha2)/180=VARIABLE/%pi-Ia*Ra or
```

```
Eb=(VARIABLE-Ia*Ra)/(1-(alpha1-alpha2)/180); // calculation of back emf
w=Eb/Ka_phi; // calculation of motor speed in rad/s
N=w*(60/(2*%pi)); // calculation of motor speed in rpm
printf("\n\nThe motor speed is \t w=%.2f rad/s or \t N=%.f rpm",w,N);
```

Scilab code Exa 14.7 Determine the triggering angle of the convertor

```
// chapter 14
// example 14.7
// Determine the triggering angle of the convertor
// page-868
clear;
clc;
// given
P=80; // in kW (power rating of motor)
E=440; // in V
N1=800; // in rpm
N2=600; // in rpm
neta=75; // in % (% torque with respect to rated torque)
Eb1=410; // in V (back emf)
Eac=415; // in V (three-phase input)
// calculate
P=P*1E3; // changing unit from kW to W
Em=sqrt(2/3)*Eac; // calculation of peak value of input voltage
// since Eb1/Eb2=N1/N2, therefore we get
Eb2=Eb1*(N2/N1); // calculation of back emf corresponding to N2=600 rpm
Ia=P/E; // calculation of armature current
// since Eb1=E-Ia*Ra, therefore we get
Ra=(E-Eb1)/Ia; // calculation of armature resistance
```

```
Ea=Eb2+((neta/100)*Ia*Ra); // calculation of armature volage
// since Ea=(3*sqrt(3)/%pi)*Em*cosd(alpha), therefore we get
alpha=acosd((Ea/Em)*(%pi/(3*sqrt(3)))); // calculation of triggering angle of the convertor
printf("\n\nThe triggering angle of the convertor is \t alpha=%.2f rdegree",alpha);
// Note: The answer varies slightly due to precise calculations
```

Scilab code Exa 14.8 Determine no load speeds firing angle supply power factor and the speed regulation

```
// chapter 14
// example 14.8
// Determine no load speeds, firing angle, supply power factor and the speed regulation
// page-868-870
clear;
clc;
// given
P=150; // in HP (power rating of motor)
E0=650; // in V (voltgae rating of motor)
N1=1750; // in rpm
Eac=460; // in V (three-phase input)
Ia=170; // in A (rated armature current)
Ra=0.099; // in ohm (armature resistance)
La=0.73; // in mH (armature inductance)
Ka_phi=0.33; // in V/rpm (motor voltage constant)
alpha1=0, alpha2=30; // in degrees (firing angle)
neta=10; // in % (armature current at no load as a percentage of rated armature current)
// calculate
E=Eac/sqrt(3);
```

```
Em=E*sqrt(2);
Ea1=(3*sqrt(3)/%pi)*Em*cosd(alpha1);
Eb1=Ea1-Ia*Ra;
N_NL1=Eb1/Ka_phi;
Ea2=(3*sqrt(3)/%pi)*Em*cosd(alpha2);
Eb2=Ea2-Ia*Ra;
N_NL2=Eb2/Ka_phi;
Eb3=Ka_phi*N1;
Ea3=Eb3+Ia*Ra;
// since Ea=(3*sqrt(3)/%pi)*Em*cosd(alpha), therefore we get
alpha3=acosd((Ea3/Em)*(%pi/(3*sqrt(3))));
I_A=sqrt((1/%pi)*Ia^2*(2*%pi/3));
VA=3*E*I_A;
Ps=Ea3*Ia;
PF=Ps/VA;
Eb4=Ea3-((neta/100)*Ia*Ra);
N_NL3=Eb4/Ka_phi;
SR=((N_NL3-N1)/N1)*100;
printf("\nThe no load speeds are");
printf("\n\t\t\t For alpha=%.f degree, N_NL= %.2f rpm",alpha1,N_NL1);
printf("\n\t\t\t For alpha=%.f degree, N_NL= %.2f rpm",alpha2,N_NL2);
printf("\n\nThe firing angle to obtain a speed of N=%.f rpm is \t alpha= %.2f degree",N1,alpha3);
printf("\nThe supply power factor is \t\t\t\t PF= %.2f",PF);
printf("\nThe speed regulation is \t\t\t\t\t SR= %.2f %%",SR);
// Note: The answer varies slightly due to precise calculations
```

Scilab code Exa 14.9 Determine motor torque motor current and supply power factor

```
// chapter 14
// example 14.9
// Determine motor torque, motor current and supply power factor
// page-878-879
clear;
clc;
// given
P=20; // in HP (power rating of motor)
E0=650; // in V (voltgae rating of motor)
N=1000; // in rpm
Ra=0.25; // in ohm (armature resistance)
K_af=0.03; // in NmA^2
K_res=0.075; // in Vs/rad
Es=230; // in V (supply voltage)
alpha=30; // in degree (firing angle)
// calculate
Em=sqrt(2)*Es; // calculation of peak value of supply voltage
// part- (a) // semiconvertor controlled dc drive
w=N*(2*%pi/60); // calculation of angular speed
T=K_af*((((Em/%pi)*(1+cosd(alpha)))-K_res*w)/(Ra+K_af*w))^2; // calculation of motor torque
Ia=sqrt(T/K_af); // calculation of motor current
Ea=(sqrt(2)*Es/%pi)*(1+cosd(alpha)); // calculation of motor terminal voltage
Ps=Ea*Ia; // calculation of input power
VA=Es*Ia*sqrt(5/6); // calculation of input volt-ampere
PF=Ps/VA; // calculation of supply power factor
printf("\nFor semiconvertor controlled DC drives");
printf("\nThe motor torque is \t\t T= %.2f Nm",T);
printf("\nThe motor current is \t\t Ia= %.2f A",Ia);
printf("\nThe supply power factor is \t PF= %.2f",PF);
// part (b)
T=K_af*((((2*Em/%pi)*cosd(alpha))-K_res*w)/(Ra+K_af*w))^2; // calculation of motor torque
```

```
Ia=sqrt(T/K_af); // calculation of motor current
Ea=(2*sqrt(2)*Es/%pi)*cosd(alpha); // calculation of
    motor terminal voltage
Ps=Ea*Ia; // calculation of input power
VA=Es*Ia; // calculation of input volt-ampere
PF=Ps/VA; // calculation of supply power factor
printf("\n\nFor fullconvertor controlled DC drives")
  ;
printf("\nThe motor torque is \t\t T= %.2f Nm",T);
printf("\nThe motor current is \t\t Ia= %.2f A",Ia);
printf("\nThe supply power factor is \t PF= %.2f",PF
  );
// Note: The answer varies slightly due to precise
  calculations
```

Scilab code Exa 14.10 Determine the firing angle of the armature convertor speed of the motor and firing angle of the field convertor

```
// chapter 14
// example 14.10
// Determine the firing angle of the armature
   convertor, speed of the motor and firing angle of
    the field convertor
// page-885-886
clear;
clc;
// given
P=25; // in HP (power rating of motor)
E0=320; // in V (voltgae rating of motor)
N=960; // in rpm
Eac=210; // in V (ac input voltage)
Ra=0.2; // in ohm (armature resistance)
Rf=130; // in ohm (field resistance)
K_a=1.2; // in V/A rad/s (motor voltage constant)
T=110; // in Nm (torque developed)
```

```
alpha_a=0; // in degree (firing angle for armature convertor)
N2=1750; // in rpm
// calculate
Ep=Eac/sqrt(3);
Em=sqrt(2)*Ep; // calculation of peak value of phase voltage
Ef=(3*sqrt(3)*Em/%pi)*cosd(alpha_a);
If=Ef/Rf;
// since T=Ia*Ka*If, therefore we get
Ia=T/(K_a*If);
w=N*(2*%pi/60);
Eb=K_a*If*w;
Ea=Eb+Ia*Ra;
// since Ea=(3*sqrt(3)*Em/%pi)*cosd(alpha), therefore we get
alpha=acosd((Ea/Em)*(%pi/(3*sqrt(3))));
Ea1=(3*sqrt(3)*Em/%pi)*cosd(alpha_a);
Eb1=Ea1-Ia*Ra;
w1=Eb1/(K_a*If);
N1=w1*(60/(2*%pi));
w2=N2*(2*%pi/60);
// since Eb=K_a*If*w, therefore we get
If=Eb1/(K_a*w2);
Ef=If*Rf;
// since Ef=(3*sqrt(3)*Em/%pi)*cosd(alpha_f), therefore we get
alpha_f=acosd((Ef/Em)*(%pi/(3*sqrt(3))));
printf(”\nThe firing angle of the armature convertor is \talpha= %.2f degree”,alpha);
printf(”\nThe speed of the motor is \t\t\t N1= %.2f rpm”,N1);
printf(”\nThe firing angle of the field convertor is \t alpha_f= %.2f”,alpha_f);
// Note: The answer varies slightly due to precise calculations
```

Scilab code Exa 14.11 Determine the firing angle power factor active power and reactive power at rated speed and its 10 and their ratio

```
// chapter 14
// example 14.11
// Determine the firing angle, power factor, active power and reactive power at rated speed and its 10 % and their ratio
// page-886-887
clear;
clc;
// given
P=100; // in kW (power rating of motor)
N=1000; // in rpm
Ea=460; // in V
I=300; // in A
E=415; // in V (3 phase input voltage)
neta=10; // in % (% of rated speed for new speed)
// calculate
P=P*1E3;
Em=sqrt(2/3)*E; // calculation of peak value of phase voltage
// since Ea=(3*sqrt(3)*Em/%pi)*cosd(alpha), therefore we get
alpha1=acosd((Ea/Em)*(%pi/(3*sqrt(3))));
PF1=cosd(alpha1);
Ea_alpha=(neta/100)*Ea;
alpha2=acosd((Ea_alpha/Em)*(%pi/(3*sqrt(3))));
PF2=cosd(alpha2);
Ia=P/Ea;
I=sqrt(2/3)*Ia;
P_active1=sqrt(3)*E*I*cosd(alpha1);
P_reactive1=sqrt(3)*E*I*sind(alpha1);
P_active2=sqrt(3)*E*I*cosd(alpha2);
```

```
P_reactive2=sqrt(3)*E*I*sind(alpha2);
ratio=P_reactive2/P_reactive1;
printf("\nAt rated speed, the firing angle is \t alpha=%.2f degree and the power factor is \t PF=%.2f",alpha1,PF1);
printf("\n\nAt %.f %% rated speed, the firing angle is \t alpha=%.2f degree and the power factor is \t PF=%.3f",neta,alpha2,PF2);
printf("\n\nAt rated speed, the active power is \t P_active=%.1f kW and reactive power is \t P_reactive=%.2f kVAR",P_active1*1E-3,P_reactive1*1E-3);
printf("\n\nAt %.f %% rated speed, the active power is \t P_active=%.2f kW and reactive power is \t P_reactive=%.2f kVAR",neta,P_active2*1E-3,P_reactive2*1E-3);
printf("\n\nThe ratio of reactive power at  %.f %% rated speed and rated speed is %.2f",neta,ratio);
// Note: The answer vary slightly due to precise calculations
```

Scilab code Exa 14.12 Determine the range of frequencies of the chopper

```
// chapter 14
// example 14.12
// Determine the range of frequencies of the chopper
// page-893
clear;
clc;
// given
I=80; // in A
E=480; // in V
N=960; // in rpm
Ra=0.25; // in ohm (armature resistance)
Rf=120; // in ohm (field resistance)
```

```
N1=400, N2=750; // in rpm (range of speed)
Ton=3; // in ms
Ef=480; // in V
// calculate
Ton=Ton*1E-3; // changing unit from ms to s
If=E/Rf; // calculation of field current
Ia=I-If; // calculation of armature current
Eb=E-Ia*Ra; // calculation of back emf
Eb1=(N1/N)*Eb; // calculation of back emf corresponding to N1=400 rpm
Ea1=Eb1+Ia*Ra; // calculation of terminal voltage corresponding to N1=400 rpm
T1=(E/Ea1)*Ton; // calculation of chopping period corresponding to N1=400 rpm
F1=1/T1; // calculation of frequency corresponding to N1=400 rpm
Eb2=(N2/N)*Eb; // calculation of back emf corresponding to N2=400 rpm
Ea2=Eb2+Ia*Ra; // calculation of terminal voltage corresponding to N2=400 rpm
T2=(E/Ea2)*Ton; // calculation of chopping period corresponding to N2=400 rpm
F2=1/T2; // calculation of frequency corresponding to N2=400 rpm
printf("\n\nThe range of frequencies of the chopper is \t %.2f Hz <= f <= %.2f Hz",F1,F2);
// Note: The answer vary slightly due to precise calculations
```

Chapter 15

Control of A C Drives

Scilab code Exa 15.1 Determine the new value of torque developed by motor

```
// chapter 15
// example 15.1
// Determine the new value of torque developed by motor
// page-912
clear;
clc;
// given
T1=1500; // in sync Watts
f1=50; // in Hz
N=1440; // in rpm
NS=1500; // in rpm (synchronoous speed)
f2=75; // in Hz
// calculate
// since T1/T2=f2/f1, therefore we get
T2=(f1/f2)*T1;
printf("\n\nThe new value of torque developed by motor is \t T2=%.2f sync. watts",T2);
// Note: The answer vary slightly due to precise calculations
```

Scilab code Exa 15.2 Determine the angle of firing advance in different cases voltage ratio and power flow

```
// chapter 15
// example 15.2
// Determine the angle of firing advance in different cases, voltage ratio and power flow
// page-958-959
clear;
clc;
// given
P=6; // number of poles
f=50; // in Hz
N1=600, N2=800; // in rpm
E=600; // in V open circuit standstill slip-ring voltage
Es=415; // in V (supply voltage)
gama1=10; // in degree (overlap angle in rectifier)
gama2=5; // in degree (overlap angle in invertor)
V_diode=1.5; // in V (voltage drop across SCR)
V_SCR=0.7; // in V (voltage drop across SCR)
alpha=0; // in degree (used by the author in the book)
Pin=100; // assumption as done in the book
// calculate
Ns=(f/P)*120; // calculation of synchronous speed
S1=(Ns-N1)/Ns; // calculation of slip corresponding to N1=600
Er1=N1*S1; // calculation of rotor voltage corresponding to N1=600
Edc1=(P*Er1*sqrt(2)/%pi)*sin(%pi/P); // calculation of dc link voltage corresponding to N1=600
Edc_inv1=Edc1;
Emax=Es*sqrt(2);
```

```
// since Edc_inv=(P*Emax/(2*%pi))*sin(%pi/P)*(cos(Beta)+cos(Beta-gama)), and gama=0, therefore we get
Beta1=(acosd((2*%pi/(P*Emax*sin(%pi/P)))*Edc_inv1/2)); // calculation of angle of firing advance corresponding to N1=600
S2=(Ns-N2)/Ns; // calculation of slip corresponding to N2=800
Er2=N2*S2; // calculation of rotor voltage corresponding to N2=800
Edc2=(P*Er2*sqrt(2)/%pi)*sin(%pi/P); // calculation of dc link voltage corresponding to N2=800
Edc_inv2=Edc2;
// since Edc_inv=(P*Emax/(2*%pi))*sin(%pi/P)*(cos(Beta)+cos(Beta-gama)), and gama=0, therefore we get
Beta2=(acosd((2*%pi/(P*Emax*sin(%pi/P)))*Edc_inv2/2)); // calculation of angle of firing advance corresponding to N2=800
Edc3=(P*Emax/(2*%pi))*sin(%pi/P)*(cos(alpha)+cos(alpha+gama1))-2*V_SCR;
Beta3=acosd(((2*%pi/(P*Emax*sin(%pi/P)))*(Edc3-2*V_diode))/(2*cosd(gama2/2)))+gama2/2;
K=Es/Er1;
P0_rotor=Pin*S1;
P0_shaft=Pin-P0_rotor;
printf("\n\nThe angle of firing advance corresponding to N1=%.f rpm is \t Beta1=%.1f degree",N1,Beta1);
printf("\nThe angle of firing advance corresponding to N2=%.f rpm is \t Beta2=%.1f degree",N2,Beta2);
printf("\nThe angle of firing advance in overlapping case is \t\t Beta3=%.1f degree",Beta3);
printf("\n\nThe transfromer ratio is \t K=%.2f",K);
printf("\n\nThe power out of the rotor is \t P0_rotor=%.f %% of power input to the stator",P0_rotor);
printf("\nThe shaft output power is \t P0_shaft=%.f
```

```
%% of power input to the stator”,P0_shaft);
// Note: The answer in the book is wrong for Beta3 due to use of wrong values of Emax and gama1 in the calculation. I have used the correct values
```

Scilab code Exa 15.3 Determine the input voltage armature current excitation voltage torque angle and pull out torque

```
// chapter 15
// example 15.3
// Determine the input voltage, armature current, excitation voltage, torque angle and pull-out torque
// page-977-978
clear;
clc;
// given
P=6; // number of poles
f=50; // in Hz
E=400; // in V
Xs=%i*2; // in ohm (reactance per phase)
T_L=300; // in Nm
N=1000; // in rpm
f_inv=40; // in Hz (invertor frequency)
Nm=560; // in rpm (motor speed)
phi=0; // in degree (used in the book)
// calculate
Pf=cosd(phi);
Es=E/sqrt(3);
Eb=Es;
Ea_rated=Es;
w=2*%pi*f;
ws=2*w/P;
wb=ws;
Ns=(60/(2*%pi))*ws;
```

```
K=Eb/wb;
T_L1=T_L*(Nm/N)^2;
wm1=Nm*(2*%pi/60);
ws1=wm1;
P0=T_L1*wm1;
Ea=K*ws1;
// since P0=3*Ea*Ia*Pf, therefore we get
Ia=P0/(3*Ea*Pf);
Ef=Ea-Ia*(Xs);
Ef_abs=abs(Ef);
Ef_phase=atand(imag(Ef)/real(Ef));
del=Ef_phase;
T_P=3*Ea*Ef_abs/(2*wm1);
printf("\nThe input voltage is \t\t Ea=%.2f V",Ea);
printf("\nThe armature current is \t Ia=%.2f A",Ia);
printf("\nThe excitation voltage is \t Ef=%.2f V <%.2f degree",Ef_abs,Ef_phase);
printf("\nThe torque angle is \t\t %.2f degree",del);
printf("\nThe pull-out torque is \t\t T_P=%.2f Nm",T_P);
// Note: 1. In the book, the value of N given is N=100 rpm, but hte author hase used N=1000 rpm while solving. I have also used N=1000 rpm to make the answers to have practical values.
//       2. The answers in the book vary slightly due to precise calculations
```

Chapter 16

Power Electronic Applications

Scilab code Exa 16.1 Determine kVA rating of invertor Wattage of the rectifier and AH rating of battery

```
// chapter 16
// example 16.1
// Determine kVA rating of invertor , Wattage of the
   rectifier and A–H rating of  battery
// page −996
clear;
clc;
// given
Load=600; // in W
PF=0.8; // (lagging power factor)
efficiency=80; // in %
Edc=24; // in V
backup_time=30; // in min
// calculate
// since PF=Load/kVA_rating , therefore we get
kVA_rating=Load/PF; // calculation of kVA rating of
   invertor
wattage=kVA_rating*PF/(efficiency/100); //
   calculation of Wattage of the rectifier
Idc=kVA_rating/Edc; // calculation of dc current
```

```
AH_rating=Idc*backup_time/60; // calculation of A-H rating of battery
printf("\nThe kVA rating of invertor is \t\t %.2f kW",kVA_rating*1E-3);
printf("\nThe Wattage of the rectifier is \t %.2f kW",wattage*1E-3);
printf("\nThe A-H rating of  battery is \t\t %.2f AH",AH_rating);
// The battery voltage is given as Edc=24 V but the author has  used Edc=48 V. Thats why the answer in the book for AH_rating is wrong.
```

Scilab code Exa 16.2 Select a suitable battery

```
// chapter 16
// example 16.2
// Select a suitable battery
// page-996-997
clear;
clc;
// given
UPS_rating=20; // in kVA
backup_time=15; // in min
efficiency=85; // in %
PF=0.8; // (lagging power factor)
Edc1=147, Edc2=190; // in V (Battery voltage range)
V_cell=1.75; // in V (voltage per cell)
N=6; // in cell groups per jar
// calculate

Battery_kW=UPS_rating*PF/(efficiency/100); // calculation of battery power
num_cell=Edc1/V_cell; // calculation of number of cells
num_jar=num_cell/N; // calculation of number of jars
```

```
cell_size_kW=Battery_kW/num_cell // calculation of cell size
printf(”\nThe battery power is \t %.2f kW”,Battery_kW);
printf(”\nThe number of cells is \t %.f”,num_cell);
printf(”\nThe number of jars is \t %.f”,num_jar);
printf(”\nThe cell size is \t %.3f kW”,cell_size_kW);
printf(”\n\nBased on this data, we can choose 90 A–H battery which has 0.255 kW discharge rate at %.f min back−up time”,backup_time);
```

Scilab code Exa 16.3 Determine AH efficiency and WH efficiency

```
// chapter 16
// example 16.3
// Determine AH efficiency and WH efficiency
// page−997
clear;
clc;
// given
V_cell=2; // in V (voltage of lead acid cell)
I_discharge=35; // in A (discharge current)
T_discharge=5; // in Hrs
Eavg=1.95; // in V
I_charge=45; // in A (4charge current)
T_charge=4; // in Hrs
// calculate
AH_efficiency=((I_discharge*T_discharge)/(I_charge*T_charge))*100; // calculation of AH efficiency
WH_efficiency=AH_efficiency*Eavg/V_cell; // calculation of WH efficiency

printf(”\nThe AH efficiency is \t %.2f %%”,AH_efficiency);
```

```
printf("\nThe WH efficiency is \t %.2f %%", WH_efficiency);
```

Scilab code Exa 16.4 Determine dc supply voltage and charging current

```
// chapter 16
// example 16.4
// Determine dc supply voltage and charging current
// page-997
clear;
clc;
// given
num_cell=18; // number of cells
AH_output=90; // in AH (AH output per cell)
T=10; // in Hrs
charging_time=8; // in Hrs
AH_efficiency=85; // in %
V_cell=2.4; // in V (voltage per cell)
r=0.1; // in ohm (internal resistance of battery)
// calculate
// since AH_efficiency=AH_output/AH_input, therefore we get
AH_input=AH_output/(AH_efficiency/100); // calculation of input AH per cell
// since AH_input=charging_current*charging_time, therefore we get
charging_current=AH_input/charging_time; // calculation of charging current
V_total=num_cell*V_cell; // calculation of total terminal voltage of 18 cells
V_drop=r*charging_current; // calculation of voltage drop across internal resistance
Edc=V_total+V_drop; // calculation of dc supply voltage
printf("\nThe charging current is \t %.2f A",
```

```
charging_current);
printf(”\nThe dc supply voltage is \t %.3f V”,Edc);
// Note: the answers vary slightly due to precise calculation
```

Scilab code Exa 16.5 Calculate back up time and charger peak output power

```
// chapter 16
// example 16.5
// Calculate back-up time and charger peak output power
// page-997-998
clear;
clc;
// given
VA_rating=6; // in kVA
V=230; // in V
E=144; // in V
PF=0.8;
neta=0.85; // invertor efficiency
AH_rating=500; // in AH
E1=10.6, E2=13.4; // in V (range of battery voltage)
E_normal=12; // in V (normal battery voltage)
T=4; // in Hrs (charging time)
t=8; // in Hrs
capacity_derating=0.5;
// calculate
VA_rating=VA_rating*1E3;
Battery_kW=VA_rating*PF/neta; // calculation of battery power
num_Battery=E/E_normal; // calculation of number of batteries
// considering worst case for calculation of discharge current
```

```
Total_battery_voltage=E1*num_Battery; // calculation of total battery voltage
// since Battery_kW=Total_battery_voltage*I_dc , therefore we get ,
I_dc=Battery_kW/Total_battery_voltage; // calculation of battery discharge current
T_backup=AH_rating*capacity_derating/I_dc; // calculation of back-up time
Ic=AH_rating*capacity_derating/T; // calculation of charging current
P_peak=E*Ic; // calculation of charging peak power
printf("\nThe back-up time is \t\t %.3f hours",T_backup);
printf("\nThe charging peak power is \t %.f kW",P_peak*1E-3);
// Note : There is calculation mistake in the book while calculating T_backup. Thats why answer in the book is wrong
```

Scilab code Exa 16.6 Determine maximum duty cycle and flyback convertor turns ratio

```
// chapter 16
// example 16.6
// Determine maximum duty cycle and flyback convertor turns ratio
// page-1007
clear;
clc;
// given
fs=60; // in kHz (switching frequency)
Esp=180; // in V (spike voltage)
Mains=230; // in V (mains supply)
f=50; // in Hz (supply frequency)
E0=12; // in V
```

```
E_DS=1200; // in V (used in the book, voltage across MOSFET)
// calculate
Edc=Mains*sqrt(2); // calculation of peak mains voltage
// since E_DS=(Edc/(1-alpha_max)+Esp, therefore we get
alpha_max=1-(Edc/(E_DS-Esp)); // calculation of maximum duty cycle
// since E0=(alpha_max/(1-alpha_max))*(N2/N1)*Edc, therefore we get
K=(Edc/E0)*(alpha_max/(1-alpha_max)); // calculation of flyback convertor turns ratio
printf("\nThe maximum duty cycle is \t\t alpha_max=%.2f",alpha_max);
printf("\nThe flyback convertor turns ratio is \t N2/N1=%.1f",K);
// Note : the answer vary slightly due to precise calculations
```

Scilab code Exa 16.7 Determine average input power and current ratings of the trasistor primary winding indutance and number of turns and ratings of the diode

```
// chapter 16
// example 16.7
// Determine average input power and current, ratings of the trasistor, primary winding indutance and number of turns, and ratings of the diode
// page-1007-1009
clear;
clc;
// given
Edc_min=14, Edc_max=30; // in V (range of supply
```

```
voltage)
V1=12, V2=-12, // in V (output voltages)
I1=0.6, I2=0.6; // in A (output current)
i_1=0.1, i_2=0.5; // in A (range of load current)
neta=80; // in % (efficiency of the convertor)
fs=50; // in kHz (switching frequency)
alpha_max=0.5; // assumption as done in the book
L0=80; // in mH  (assumption as done in the book)
N0=1100; //  (assumption as done in the book)
// calculate
fs=fs*1E3; // changing unit from kHz to Hz
L0=L0*1E-3; // changing unit from mH to H
P0=(V1*I1)+(-V2*I2); // calculation of full-load power
Pi=P0/(neta/100); // calculation of input power
I_avg_max=Pi/Edc_min; // calculation of maximum average input current
I_avg_min=Pi/Edc_max; // calculation of minimum average input current
Ic_max=2*P0/((neta/100)*Edc_min*alpha_max); // calculation of transistor current rating
V_CE_max=Edc_max/(1-alpha_max); // calculation of transistor voltage rating
Pd_max=V_CE_max*Ic_max; // calculation of transistor power rating
L_prim=Edc_min*alpha_max/(Ic_max*fs); // calculation of primary inductance
N_prim=N0*sqrt(L_prim/L0); // calculation of number of turns in primary winding
// since E0=(alpha_max/(1-alpha_max))*(N2/N1)*Edc, therefore we get
N_sec=(V1/Edc_min)*N_prim*((1-alpha_max)/alpha_max); // calculation of number of turns in secondary winding
Id=(N_prim/N_sec)*Ic_max; // calculation of peak diode current
PIV=V1+(Edc_max/(N_prim/N_sec)); // calculation of peak inverse voltage
```

```
printf("\nThe average input power is \t Pi=%.f W",Pi);
printf("\nThe maximum average input current is \t I_avg_max=%.3f A",I_avg_max);
printf("\nThe minimum average input current is \t I_avg_min=%.3f A",I_avg_min);
printf("\n\nThe ratings of the transistor are \t Ic_max=%.2f A \t V_CE_max=%.f V \t Pd_max=%.f W",Ic_max,V_CE_max,Pd_max);
printf("\n\nThe number of turns in primary winding is \t N_prim=%.f",N_prim);
printf("\n\nThe number of turns in secondary winding is \t N_sec=%.f",N_sec);
printf("\n\nThe ratings of the diode are \t Id>=%.1f A \t PIV>= %.2f V",Id,PIV);
// Note : 1.There is calculation mistake while calculating Id. Thats why answer in the book is wrong.
//        2. The answers vary slightly due to precise calculations
```

Scilab code Exa 16.8 Determine turn ratio of demagnetizing winding with primary winding switch voltagae rating and dc supply current

```
// chapter 16
// example 16.8
// Determine turn ratio of demagnetizing winding with primary winding, switch voltagae rating and dc supply current
// page-1015
clear;
clc;
// given
E0=12; // in V
I0=12; // in A
```

```
fs=60; // in kHz (switching frequency)
E=230; // in V (rectified ac mains)
alpha_max=0.6; // maximum duty cycle
neta1=50; // in % (spike voltage percentage)
neta2=20; // in %
// calculate
fs=fs*1E3; // changing unit from kHz to Hz
Edc=E*sqrt(2); // calculation of dc voltage
Esp=(neta1/100)*Edc; // calculation of spike voltage
// since alpha_max=1/(1+(N3/N1)), therefore we get
// K=N1/N3=1/((1/alpha_max)-1),
K=1/((1/alpha_max)-1); // calculation of turns ratio
Vsw=Edc+K*Edc+Esp; // calculation of switch voltage
// since K1=N1/N2=V1/V2, therefore we get
K1=(1-(neta2/100))*Edc/E0;
// since I1/I2=N2/N1, therefore we get
I1=I0/K1; // calculation of dc supply current
printf("\nThe turn ratio of demagnetizing winding with primary winding is \t N1/N3=%.1f",K);
printf("\nThe required switch voltage is \t\t\t\t\t\t Vsw=%.2f V",Vsw);
printf("\nThe dc supply current at full current is \t\t\t\t I1=%.2f A",I1);
// Note: The answer vary slightly due to precise calculations
```

Scilab code Exa 16.9 Determine the value of coupling capacitor verfiy if the value is acceptable or not If not then determine the new value

```
// chapter 16
// example 16.9
// Determine the value of coupling capacitor, verfiy if the value is acceptable or not, If not then determine the new value
// page-1019-1020
```

```
clear;
clc;
// given
P0=200; // in W
fs=20; // in kHz
n=10; // turns ratio
L=20; // in uH
e=80; // in % (efficiency)
E=230; // in V (used in the book)
alpha_max=0.8; // assumption for duty cycle as done in the book
tolerance=20; // assumption for voltage tolerance as done in the book
// calculate
fs=fs*1E3; // changing unit from kHz to Hz
L=L*1E-6; // changing unit from uH to H
L_R=n^2*L;// calculation of resonant inductance
f_R=0.25*fs;// calculation of resonant freuquency
Cc=1/(4*(%pi*f_R)^2*L_R);// calculation of coupling capacitor
Edc=E*sqrt(2);// calculation of dc voltage
Ic=2*P0/((e/100)*alpha_max*Edc);// calculation of coupling current
I=Ic+(tolerance/100)*Ic;// calculation of worst case coupling current
dt=alpha_max/(2*fs);
Vc=(I/Cc)*dt;// calculation of coupling capacitor charge voltage
// since the range of Vc is 6 to 32 V,
// let us assume dVc=30 V therefore, the new value of capacitor is C=I*(dt/dVc);
dVc=30; // in V
C=I*(dt/dVc);// calculation of new coupling capacitance
printf("\nThe value of coupling capacitor is \t\t Cc=%.2f uF",Cc*1E6);
printf("\nThe peak coupling capacitor charge voltage is \t vc=%.1f V",Vc);
```

```
if Vc>=6 & Vc<=32 then
    printf("\n\nThe value of Vc is acceptable.");
else
    C=I*dt/dVc;
    printf("\nThe new value of coupling capacitor is \t\t C=%.1f uF",C*1E6);
end
// Note : the answer vary slightly due to round off
```

Scilab code Exa 16.10 Determine rms current and peak reverse voltage

```
// chapter 16
// example 16.10
// fig. 16.20
// Determine rms current and peak reverse voltage
// page-1028-1029
clear;
clc;
// given
P0=500; // in MW
E=250; // in kV
// calculate
P0=P0*1E6; // changing unit from MW to W
E=E*1E3; // changing unit from kV to V
Id=P0/(2*E); // calculation of direct current
Irms=Id/sqrt(3); // calculation of rms current
E_line_max=E/2; // calculation of maximum line voltage
 for 6 pulse group
PIV=E_line_max*(%pi/3); // calculation of peak reverse voltage
printf("\nThe rms current is \t\t Irms=%.2f A",Irms);
printf("\nThe peak reverse voltage is \t PIV=%.f kV",PIV*1E-3);
```

Scilab code Exa 16.11 Determine depth of heating and heat generated per unit surface area

```
// chapter 16
// example 16.11
// Determine depth of heating and heat generated per
    unit surface area
// page-1036
clear;
clc;
// given
L=10; // in cm (length)
d=1; // in cm  (diameter)
N=15; // number of turns
I=80; // in A
f=200; // in kHz
ur=1; // relative permeability
rho=5E-8; // in ohm-m (resistivity)
// calculate
L=L*1E-2; // changing unit from cm to m
d=d*1E-2; // changing unit from cm to m
f=f*1E3; // changing unit from KHz to Hz
del=503*sqrt(rho/(ur*f)); // calculation of depth of
    heating
H=2*%pi*(N*I/L)^2*sqrt(rho*ur*f/1E7); // calculation
    of heat generated per unit surface area
printf("\nThe depth of heating is \t\t\t del=%.3f mm
   ",del*1E3);
printf("\nThe heat generated per unit surface area
   is \t H=%.2f kW/m^2",H*1E-3);
// Note : The value of L given is L=10 cm but the
   author has used L=1 cm in the calculation. Thats
   why answer in the book is wrong.
```

Scilab code Exa 16.12 Determine supply frequency

```
// chapter 16
// example 16.12
// Determine supply frequency
// page-1037
clear;
clc;
// given
del=2.5; // in mm (depth of heating)
rho=5E-5; // in ohm-cm (resistivity)
ur=1; // relative permeability
// calculate
del=del*1E-3; // changing unit from mm to m
rho=rho*1E-2; // changing unit from ohm-cm to ohm-m
f=(rho/ur)*(503/del)^2; // calculation of supply frequency
printf("\nThe supply frequency is \t f=%.2f kHz",f*1E-3);
```

Scilab code Exa 16.13 Determine voltage and current

```
// chapter 16
// example 16.13
// Determine voltage and current
// page-1041
clear;
clc;
// given
l=25, b=15, t=1; // in cm (dimensions of piece)
P=1; // in kW
Er=4; // relative permittivity
```

```
PF=0.6; // power factor
f=40; // in MHz
Eo=8.854E-12; // absolute permittivity
// calculate
l=l*1E-2; // changing unit from cm to m
b=b*1E-2; // changing unit from cm to m
t=t*1E-2; // changing unit from cm to m
P=P*1E3; // changing unit from kW to W
f=f*1E6; // changing unit from MHz to Hz
A=l*b; // calculation of area
C=Eo*Er*A/t; // calculation of capacitance
// since P=2*%pi*f*C*V^2*PF, therefore we get
V=sqrt(P/(2*%pi*f*C*PF)); // calculation of voltage
Xc=1/(2*%pi*f*C); // calculation of capacitive reactance
I=V/Xc; // calculation of current
printf(”\nThe voltage is \t V=%.2f V”,V);
printf(”\nThe current is \t I=%.2f A”,I);
```

Scilab code Exa 16.14 Determine supply frequency

```
// chapter 16
// example 16.14
// Determine supply frequency
// page-1041
clear;
clc;
// given
C=5.3; // in uF
V=600; // in V
I=100; // in A
// calculate
C=C*1E-6; // changing unit from uF to F
Xc=V/I;
// since Xc=1/(2*%pi*f*C), therefore we get
```

```
f=1/(2*%pi*C*Xc); // calculation of supply frequency
printf("\nThe supply frequency is \t f=%.f kHz",f*1E
    -3);
```

Scilab code Exa 16.15 Calculate the value of load resistance commutating capacitor and resistance connected in series

```
// chapter 16
// example 16.15
// fig. Ex 16.15
// Calculate the value of load resistance,
    commutating capacitor and resistance connected in
     series
// page-1046
clear;
clc;
// given
Edc=50; // in V (supply voltage)
I_L=10; // in A (load current)
Toff=20; // in us (turn-off time)
I_H=5; // in mA (holding current)
// calculate
Toff=Toff*1E-6; // changing unit from us to s
I_H=I_H*1E-3; // changing unit from mA to A
R2=Edc/I_H; // calculation of resistance connected
    in series
R1=Edc/I_L; // calculation of load resistance
C=1.5*Toff*I_L/Edc; // calculation of ommutating
    capaitor
printf("\nThe value of resistance connected in
    series is \t R2=%.f k-ohm",R2*1E-3);
printf("\nThe value of load resistance is \t R1=%.f
    ohm",R1);
printf("\nThe value of commutating capacitor is \t C
    >=%.f uF",C*1E6);
```

Scilab code Exa 16.16 What will be the firing angle and output power available

```
// chapter 16
// example 16.16
// What will be the firing angle and output power available
// page-1051
clear;
clc;
// given
f=50; // in Hz
neta=60; // in %
T=0.24; // in s (repition period)
// calculate
// part (i) can not be solved
// since T=0.24 s represents 12 cycles at 50 Hz or T=24 half cycles, therefore
T=24;
// since Pload/Pmax=N/24, therefore
N1=1, N2=24
Pload1=N1/T;
Pload2=N2/T;
printf("The available power range from %.2f %% of Pmax (N=%.f) to %.f %% of Pmax (N=%.f) varying in steps of %.2f %% of Pmax",Pload1*1E2,N1,Pload2*1E2,N2,Pload1*1E2);
```