Scilab Textbook Companion for Power Electronics by P. S. Bimbra[1]

Created by
Navdeep Kumar
B.E.
Electrical Engineering
Thapar University
College Teacher
Dr. Sunil Kumar Singla
Cross-Checked by
Lavitha Pereira

May 23, 2016

[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: P. S. Bimbra

Publisher: Khanna Publishers, New Delhi

Edition: 5

Year: 2012

ISBN: 978-81-7409-279-3

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

Power Semiconductor Diodes and Transistors

Scilab code Exa 2.1 to find switching freq of the transistor

```
clear;
clc;
B=40;
R_c=10;//ohm
V_cc=130;//V
V_B=10;//V
V_CES=1.0;//V
V_BES=1.5;//V
I_CS=(V_cc-V_CES)/R_c;//A
I_BS=I_CS/B;//A
R_B=(V_B-V_BES)/I_BS;    printf("value of R_B in
    saturated state= %.3f ohm",R_B);
P_T=V_BES*I_BS+V_CES*I_CS;    printf("\n power loss
    in transistor=%.3f W",P_T);

ODF=5;
I_B=ODF*I_BS;
R_B=(V_B-V_BES)/I_B;    printf("\nvalue of R_B for
    an overdrive factor 5 =%.2f ohm",R_B);
```

```
P_T=V_BES*I_B+V_CES*I_CS;    printf("\npower loss in transistor=%.2f W",P_T)

B_f=I_CS/I_B;    printf("\nforced current gain=%.0f",B_f);
```

Scilab code Exa 2.2 to determine avg power loss due to collector current and also to find peak instantaneous power loss due to collector current

```
clear;
clc;

I_CEO=2*10^-3;//A
V_CC=220;//V
P_dt=I_CEO*V_CC;//instant. power loss during delay time

t_d=.4*10^-6;//s
f=5000;
P_d=f*I_CEO*V_CC*t_d;//avg power loss during delay time

V_CES=2;//V
t_r=1*10^-6;//s
I_CS=80;//A
P_r=f*I_CS*t_r*(V_CC/2-(V_CC-V_CES)/3);//avg power loss during rise time

t_m=V_CC*t_r/(2*(V_CC-V_CES));
P_rm=I_CS*V_CC^2/(4*(V_CC-V_CES));//instant. power loss during rise time

P_on=P_d+P_r;    printf("avg power loss during turn on=%.4f W",P_on);
P_nt=I_CS*V_CES;    printf("\ninstantaneous power
```

```
    loss during turn on=%.0 f W" ,P_nt);
t_n=50*10^-6;
P_n=f*I_CS*V_CES*t_n;    printf("\navg power loss
    during conduction period=%.0 f W" ,P_n);
```

Scilab code Exa 2.3 to obtain power loss during turn off time and turn off period and also instant power loss during fall time

```
clear
clc;
I_CEO=2*10^-3;//A
V_CC=220;//V
t_d=.4*10^-6;//s
f=5000;
V_CES=2;//V
t_r=1*10^-6;//s
I_CS=80;//A
t_n=50*10^-6;//s
t_0=40*10^-6;//s
t_f=3*10^-6;//s
P_st=I_CS*V_CES;// instant. power loss during t_s
P_s=f*I_CS*V_CES*t_f;//avg power loss during t_s
P_f=f*t_f*(I_CS/6)*(V_CC-V_CES);//avg power loss
    during fall time
P_fm=(I_CS/4)*(V_CC-V_CES);//peak instant power
    dissipation

P_off=P_s+P_f;    printf("total avg power loss
    during turn off=%.0 f W" ,P_off);
P_0t=I_CEO*V_CC;    printf("\ninstantaneous power
    loss during t_0=%.2 f W" ,P_0t);
P_0=f*I_CEO*V_CC*t_0;//avg power loss during t_s
P_on=14.9339;//W    from previous eg
P_n=40;//W    from previous eg
P_T=P_on+P_n+P_off+P_0;    printf("\ntotal power
```

```
    loss=%.3f W",P_T);
```

Scilab code Exa 2.4 to find switching freq of the transistor

```
clear
clc;
I_CS=100;
V_CC=200;
t_on=40*10^-6;
P_on=(I_CS/50)*10^6*t_on*(V_CC*t_on/2-(V_CC*10^6*
    t_on^2/(40*3)));//energy during turn on
t_off=60*10^-6;
P_off=(I_CS*t_off/2-(I_CS/60)*10^6*(t_off^2)/3)*((
    V_CC/75)*10^6*t_off);//energy during turn off
P_t=P_on+P_off;//total energy
P_avg=300;
f=P_avg/P_t;    printf("allowable switching
    frequency=%.1f Hz",f);
//in book ans is: f=1123.6 Hz. The difference in
    results due to difference in rounding of of
    digits
```

Chapter 3

Diode Circuits and Rectifiers

Scilab code Exa 3.2 to find the conduction time of diode peak current through diode and final steady state voltage

```
clear;
clc;
V_s=400; //V
V_o=100; //V
L=100; //uH
C=30; //uF
t_o=%pi*sqrt(L*C);    printf("conduction time of diode=%.2f us",t_o);
//in book solution is t_o=54.77 us. The ans is incorrect as %pi is not muliplied in ans. Formulae mentioned in correct.
I_p=(V_s-V_o)*sqrt(C/L);    printf("\npeak current through diode=%.2f A",I_p);
v_D=-V_s+V_o;    printf("\nvoltage across diode=%.0f V",v_D);
```

Scilab code Exa 3.6 to determine conduction time of diode and rate of change of current

```
clc
clear
R=10;//ohm
L=.001;//H
C=5*10^-6;//F
V_s=230;//V
xi=R/(2*L);
w_o=1/sqrt(L*C);
w_r=sqrt((1/(L*C))-(R/(2*L))^2);
t=%pi/w_r;    printf('conduction time of diode=%.3f
   us',t*10^6);
t=0;
//di=di/dt
di=V_s/L;    printf('\nrate of change of current at
   t=0 is %.0f A/s',di);
```

Scilab code Exa 3.7 to find the time required to deliver a charge of 200Ah

```
clear;
clc;
I_or=100;//A
R=1;//assumption
V_m=I_or*2*R;
I_o=V_m/(%pi*R);
q=200;//Ah
t=q/I_o;    printf("time required to deliver charge=
   %.04f hrs",t);
```

Scilab code Exa 3.8 to calculate the power delivered to the heater and find peak diode current and input pf

```
clear;
clc;
```

```
V_s=230;//V
P=1000;//W
R=V_s^2/P;
V_or=sqrt(2)*V_s/2;
P_h=V_or^2/R;    printf("power delivered to the heater=%.0f W",P_h);
V_m=sqrt(2)*230;
I_m=V_m/R;    printf("\npeak value of diode current=%.4f A",I_m);
pf=V_or/V_s;    printf("\ninput power factor=%.3f",pf);
```

Scilab code Exa 3.9 to find 1 the value of avg chargng current 2 power supplied to battery and that dissipated in the resistor 3 supply pf 4 charging time 5 rectifier efficiency and PIV of diode

```
clear;
clc;
V_s=230;//V
V_m=V_s*sqrt(2);
E=150;//V
theta1=asind(E/(sqrt(2)*V_s));
R=8;//ohm
f=50;//Hz
I_o=(1/(2*%pi*R))*((2*sqrt(2)*V_s*cosd(theta1))-E*(%pi-2*theta1*%pi/180));
printf("avg value of charging current=%.4f A",I_o);
P_d=E*I_o;    printf("\npower delivered to battery=%.2f W",P_d);
I_or=sqrt((1/(2*%pi*R^2))*((V_s^2+E^2)*(%pi-2*theta1*%pi/180)+V_s^2*sind(2*theta1)-4*V_m*E*cosd(theta1)));    printf("\nrms value of the load current=%.4f A",I_or);
pf=(E*I_o+I_or^2*R)/(V_s*I_or);    printf("\nsupply pf=%.3f",pf);
```

```
P_dd=I_or^2*R;    printf("\npower dissipated in the resistor=%.2f W",P_dd);
q=1000.00;//Wh
t=q/P_d;    printf("\ncharging time=%.3f hr",t);
n=P_d*100/(P_d+P_dd);    disp(n,"rectifier efficiency(%)");
PIV=sqrt(2)*V_s+E;    printf("PIV of diode=%.2f V",PIV);
//solutions have small variations due to difference in rounding off of digits
```

Scilab code Exa 3.10 to determine the effect of reverse recovery time on the avg output voltage

```
clear;
clc;
V_s=230;//V;
t_rr=40*10^-6;//s reverde recovery time
V_o=2*sqrt(2)*V_s/%pi;
V_m=sqrt(2)*V_s;
disp("when f=50Hz");
f=50;
V_r=(V_m/%pi)*(1-cosd(2*%pi*f*t_rr*180/%pi));
v_avg=V_r*100/V_o*10^3;    printf("percentage reduction in avg o/p voltage=%.3fx10^-3",v_avg);

disp("when f=2500Hz");
f=2500;
V_r=(V_m/%pi)*(1-cosd(2*%pi*f*t_rr*180/%pi));
v_avg=V_r*100/V_o;    printf("percentage reduction in avg o/p voltage=%.3f",v_avg);
```

Scilab code Exa 3.11 to determine 1 avg value of output voltage and output current 2 avg and rms values of diode currents 3 rms values of output and input currrents and supply pf

```
clear;
clc;
V_s=230;//V
R=10;//ohm
V_m=sqrt(2)*V_s;
V_o=2*V_m/%pi;    printf("avg value of o/p voltage=%
    .2f V",V_o);
I_o=V_o/R;    printf("\navg value of o/p current=%.3
    f A",I_o);
I_DA=I_o/2;    printf("\navg value of diode current=
    %.3f A",I_DA);
I_Dr=I_o/sqrt(2);    printf("\nrms value of diode
    current=%.3f A",I_Dr);

printf("\nrms value of o/p current=%.3f A",I_o);
printf("\nrms value of i/p current=%.3f A",I_o);
pf=(V_o/V_s);    printf("\nsupply pf=%.2f",pf);
```

Scilab code Exa 3.12 to calculate 1 peak load current 2 dc load current 3 dc diode current 4 percentage regulation from no load to full load

```
clear;
clc;
V_s=230;//V
R=1000;//ohm
R_D=20;//ohm
V_m=sqrt(2)*V_s;
I_om=V_m/(R+R_D);    printf("peak load current=%.4f
    A",I_om);
I_o=I_om/%pi;    printf("\ndc load current=%.5f A",
    I_o);
```

```
V_D=I_o*R_D-V_m/%pi;     printf("\ndc diode voltage=%.1f V",V_D);
V_on=V_m/%pi;     printf("\nat no load, load voltage=%.3f V",V_on);
V_o1=I_o*R;     printf("\nat given load, load voltage=%.3f V",V_o1);
vr=(V_on-V_o1)*100/V_on;     printf("\nvoltage regulation(in percent)=%.3f",vr);
```

Scilab code Exa 3.13 to design a zener voltage regulator to meet given specifications

```
clear;
clc;
V_L=6.8;//V
V_smax=20*1.2;//V
V_smin=20*.8;//V
I_Lmax=30*1.5;//mA
I_Lmin=30*.5;//mA
I_z=1;//mA

R_smax=(V_smax-V_L)/((I_Lmin+I_z)*10^-3);     printf("max source resistance=%.0f ohm",R_smax);
R_smin=(V_smin-V_L)/((I_Lmax+I_z)*10^-3);     printf("\nmin source resistance=%.0f ohm",R_smin);//in book solution, error is committed in putting the values in formulea(printing error) but solution is correct
R_Lmax=V_L*1000/I_Lmin;     printf("\nmax load resistance=%.1f ohm",R_Lmax);
R_Lmin=V_L*1000/I_Lmax;     printf("\nmin load resistance=%.1f ohm",R_Lmin);

V_d=.6;//V
V_r=V_L-V_d;     printf("\nvoltage rating of zener
```

```
diode=%.1f V",V_r);
```

Scilab code Exa 3.14 to find R1 and R2

```
clear;
clc;
I2=200*10^-6;//A
V_z=20;//V
R_G=500;//ohm
R2=(V_z/I2)-R_G;     printf("R2=%.1f kilo-ohm",R2
    /1000);

V_v=25;//V
I1=I2;
R1=(V_v-V_z)/I1;     printf("\nR1=%.0f kilo-ohm",R1
    /1000);
```

Scilab code Exa 3.15 to find the VA rating of the transformer

```
clear;
clc;
V_s=2*230;//V
V_o=(sqrt(2)*V_s)/%pi;
R=60;//ohm
P_dc=(V_o)^2/R;
TUF=0.2865;
VA=P_dc/TUF;     printf("kVA rating of the
    transformer=%.1f kVA",VA/1000);
```

Scilab code Exa 3.16 to determine 1 avg value of output voltage 2 input current distortion factor 3 input displacement factor 4 input pf 5 input current harmonic factor and 6 creast ratio

```
clear;
clc;
tr=0.5;    //turns ratio
I_o=10;
V=230;
V_s=V/tr;
V_m=sqrt(2)*V_s;
V_o=2*V_m/%pi;    printf("o/p voltage=%.2f V",V_o);
phi1=0;    //displacemnt angle=0 as fundamnetal
    component of i/p source current in phase with
    source voltage
DF=cosd(phi1);    printf("\ndistortion factor=%.0f",
    DF);
I_s1=4*I_o/(sqrt(2)*%pi);
I_s=sqrt(I_o^2*%pi/%pi);
CDF=I_s1/I_o;    printf("\ncurrent displacent factor
    =%.1f",CDF);
pf=CDF*DF;    printf("\ni/p pf=%.2f",pf);
HF=sqrt((I_s/I_s1)^2-1);    printf("\nharmonic
    factor=%.2f",HF);
CF=I_o/I_s;    printf("\ncreast factor=%.2f",CF);
```

Scilab code Exa 3.17 to determine diode rating and transformer rating

```
clear;
clc;
V_o=230;
R=10;
V_s=V_o*%pi/(2*sqrt(2));
I_o=V_o/R;
I_m=sqrt(2)*V_s/R;    printf("peak diode current=%.2
```

```
f A",I_m);
I_DAV=I_m/%pi;    printf("\nI_DAV=%.2f A",I_DAV);    //avg value of diode current
I_Dr=I_m/2;    printf("\nI_Dr=%.2f A",I_Dr);    //rms value of diode current
PIV=sqrt(2)*V_s;    printf("\nPIV=%.1f V",PIV);
I_s=I_m/sqrt(2);
TF=V_s*I_s;    printf("\ntransformer rating=%.2f kVA",TF/1000);
```

Scilab code Exa 3.18 to calculate avg value of output voltage avg and rms values of diode current and power delivered to diode

```
clear;
clc;
tr=5;
V=1100;
R=10;
disp("in case of 3ph-3pulse type");
V_ph=V/tr;
V_mp=sqrt(2)*V_ph;
V_o=3*sqrt(3)*V_mp/(2*%pi);    printf("avg o/p voltage=%.1f V",V_o);
I_mp=V_mp/R;
I_D=(I_mp/%pi)*sin(%pi/3);    printf("\navg value of diode current=%.3f A",I_D);
I_Dr=I_mp*sqrt((1/(2*%pi))*(%pi/3+.5*sin(2*%pi/3)));    printf("\nrms value of diode current=%.2f A",I_Dr);
V_or=V_mp*sqrt((3/(2*%pi))*(%pi/3+.5*sin(2*%pi/3)));
P=(V_or^2)/R;    printf("\npower delivered=%.1f W",P);

disp("in case of 3ph-M6 type");
V_ph=V_ph/2;
```

```
V_mp=sqrt(2)*V_ph;
V_o=3*V_mp/(%pi);     printf("avg o/p voltage=%.2f V",V_o);
I_mp=V_mp/R;
I_D=(I_mp/%pi)*sin(%pi/6);     printf("\navg value of diode current=%.4f A",I_D);
I_Dr=I_mp*sqrt((1/(2*%pi))*(%pi/6+.5*sin(2*%pi/6)));     printf("\nrms value of diode current=%.3f A",I_Dr);
V_or=V_mp*sqrt((6/(2*%pi))*(%pi/6+.5*sin(2*%pi/6)));
P=(V_or^2)/R;     printf("\npower delivered=%.0f W",P);
```

Scilab code Exa 3.19 determine the ratings of diodes and of 3ph deltastar transformer

```
clear;
clc;
V_o=400;
R=10;
V_ml=V_o*%pi/3;
V_s=V_ml/(sqrt(2)*sqrt(3));
I_m=V_ml/R;
I_s=.7804*I_m;
tr=3*V_s*I_s;     printf("transformer rating=%.1f VA",tr);

I_Dr=.5518*I_m;     printf("\nrms value of diode current=%.3f A",I_Dr);
I_D=I_m/%pi;     printf("\navg value of diode current=%.3f A",I_D);
printf("\npeak diode current=%.2f A",I_m);
PIV=V_ml;     printf("\nPIV=%.2f V",PIV);
```

Scilab code Exa 3.20 to determine 1 power delivered to battery and load 2 input displacement factor 3 current distotion factor 4 input pf 5 input HF 6 transformer rating

```
clear;
clc;
V_l=230;
E=240;
R=8;
V_ml=sqrt(2)*V_l;
V_o=3*V_ml/%pi;
I_o=(V_o-E)/R;
P_b=E*I_o;    printf("power delivered to battery=%.1f W",P_b);
P_d=E*I_o+I_o^2*R;    printf("\npower delivered to load=%.2f W",P_d);

phi1=0;
DF=cosd(phi1);
printf("\ndisplacement factor=%.0f",DF);
I_s1=2*sqrt(3)*I_o/(sqrt(2)*%pi);
I_s=sqrt(I_o^2*2*%pi/(3*%pi));
CDF=I_s1/I_s;    printf("\ncurrent distortion factor=%.3f",CDF);
pf=DF*CDF;    printf("\ni/p pf=%.3f",pf);
HF=sqrt(CDF^-2-1);    printf("\nharmonic factor=%.4f",HF);
tr=sqrt(3)*V_l*I_o*sqrt(2/3);    printf("\ntranformer rating=%.2f VA",tr);
//answers have small variations from the book due to difference in rounding off of digits
```

Scilab code Exa 3.21 design capaitor filter and avg value of output voltage with and without filter

```
clear;
clc;
f=50;      //Hz
V=230;
V_m=sqrt(2)*V;
R=400;
RF=0.05;
C=(1/(4*f*R))*(1+(1/(sqrt(2)*RF)));     printf("capacitor value=%.1f uF",C/10^-6);
V_o=V_m*(1-1/(4*f*R*C));     printf("\no/p voltage with filter=%.3f V",V_o);
V_o=2*V_m/%pi;     printf("\no/p voltage without filter=%.2f V",V_o);
```

Scilab code Exa 3.22 to find the value of L with different R and CRF without L

```
clear;
clc;
f=50;
CRF=0.05;
R=300;
L=sqrt((CRF/(.4715*R))^-2-R^2)/(2*2*%pi*f);     printf("L=%.4f H",L);
R=30;
L=sqrt((CRF/(.4715*R))^-2-R^2)/(2*2*%pi*f);     printf("\nL=%.4f H",L);
L=0;
CRF=.4715*R/sqrt(R^2+(2*2*%pi*f*L)^2);     printf("\nCRF=%.4f",CRF);
```

Scilab code Exa 3.23 to design LC filter

```
clear;
clc;
R=50;
L_L=10*10^-3;
f=50;
w=2*%pi*f;
C=10/(2*w*sqrt(R^2+(2*w*L_L)^2));    printf("C=%.2f uF",C*10^6);
VRF=0.1;
L=(1/(4*w^2*C))*((sqrt(2)/(3*VRF))+1);    printf("\nL=%.3f mH",L*10^3);
```

Chapter 4

Thyristors

Scilab code Exa 4.2 to plot allowable gate voltage as the function of gate current and to find avg gate power dissipation

```
clc
clear
P=.5;     //P=V_g*I_g
V=[0.01:0.01:.25];
plot2d(P./V);
xlabel('I_g in A');
ylabel('V_g in V');

t=poly(0,'t');
P1=P*t/(2*t);
printf('average gate power dissipation(in watts)');
disp(P1);
```

Scilab code Exa 4.3 to compute gate source resistance

```
clear;
clc;
```

```
P=.5;      //P=V_g*I_g
s=130;      //s=V_g/I_g
I_g=sqrt(P/s);
V_g=s*I_g;
E=15;
R_s=(E-V_g)/I_g;      printf("gate source resistance=%.2f ohm",R_s);
//Answers have small variations from that in the book due to difference in the rounding off of digits.
```

Scilab code Exa 4.4 to compute gate source resistance trigger voltage and trigger current

```
clear;
clc;
R_s=120;      //slope of load line is -120V/A. This gives gate source resistance
printf("gate source resistance=%.0f ohm",R_s);

P=.4;      //P=V_g*I_g
E_s=15;
//E_s=I_g*R_s+V_g,      after solving this
//120*I_g^2-15*I_g+0.4=0      so
a=120;     b=-15;     c=0.4;
D=sqrt((b^2)-4*a*c);
I_g=(-b+D)/(2*a);     V_g=P/I_g;
printf("\ntrigger current=%.2f mA",I_g*10^3);    printf("\nthen trigger voltage=%.3f V",V_g);

I_g=(-b-D)/(2*a);     V_g=P/I_g;
printf("\ntrigger current=%.2f mA",I_g*10^3);    printf("\nthen trigger voltage=%.2f V",V_g);
```

Scilab code Exa 4.5 to compute 1 resistance 2 triggering freq 3 duty cycle of the triggering pulse

```
clear;
clc;
//V_g=1+10*I_g
P_gm=5;      //P_gm=V_g*I_g
// after solving , eqn becomes 10*I_g^2+I_g-5=0
a=10;    b=1;    c=-5;
I_g=(-b+sqrt(b^2-4*a*c))/(2*a);
E_s=15;
// using E_s=R_s*I_g+V_g
R_s=(E_s-1)/I_g-10;     printf("reistance=%.3f ohm",
    R_s);

P_gav=.3;     //W
T=20*10^-6;
f=P_gav/(P_gm*T);     printf("\ntriggering freq=%.0f
    kHz",f/1000);

dl=f*T;     printf("\nduty cycle=%.2f",dl);
```

Scilab code Exa 4.6 to compute min width of gate pulse current

```
clear;
clc;
I=.1;
E=200;
disp("in case load consists of (a)L=.2H");
L=.2;
t=I*L/E;     printf("min gate pulse width=%.0f us",t
    *10^6);
```

```
disp("(b)R=20ohm in series with L=.2H");
R=20;
t=(-L/R)*log(1-(R*I/E));    printf("min gate pulse
    width=%.3f us",t*10^6);
disp("(c)R=20ohm in series with L=2H");
L=2;
t=(-L/R)*log(1-(R*I/E));    printf("min gate pulse
    width=%.2f us",t*10^6);
```

Scilab code Exa 4.8 to calculate trigger voltage and trigger current

```
clc
clear
E_s=16;
R_s=128;
P=.5;
y=poly([P -E_s R_s],'i','coeff');
a=roots(y);
printf('trigger current=%.1f mA',a(1)*1000);
printf('\ntrigger voltage=%.0f V',P/a(1));
```

Scilab code Exa 4.9 to compute avg on current rating for half sine wave-current for various conduction angles

```
clear;
clc;
function [I_TAV]=theta(th)
    I_m=1;    //supposition
    I_av=(I_m/(2*%pi))*(1+cosd(th));
    I_rms=sqrt((I_m/(2*%pi))*((180-th)*%pi/360+.25*
       sind(2*th)));
    FF=I_rms/I_av;
    I_rms=35;
```

```
    I_TAV=I_rms/FF;
endfunction
disp("when conduction angle=180");
th=0;
I_TAV=theta(th);
printf("avg on current rating=%.3f A",I_TAV);
disp("when conduction angle=90");
th=90;
I_TAV=theta(th);
printf("avg on current rating=%.3f A",I_TAV);
disp("when conduction angle=30");
th=150;
I_TAV=theta(th);
printf("avg on current rating=%.3f A",I_TAV);
```

Scilab code Exa 4.10 to compute avg on current rating for half sine wave current for various conduction angles

```
clear;
clc;
clear
function [I_TAV]=theta(th)
    n=360/th;
    I=1;    //supposition
    I_av=I/n;
    I_rms=I/sqrt(n);
    FF=I_rms/I_av;
    I_rms=35;
    I_TAV=I_rms/FF;
endfunction
disp("when conduction angle=180");
th=180;
I_TAV=theta(th);
printf("avg on current rating=%.3f A",I_TAV);
disp("when conduction angle=90");
```

```
th=90;
I_TAV=theta(th);
printf("avg on current rating=%.1f A",I_TAV);
disp("when conduction angle=30");
th=30;
I_TAV=theta(th);
printf("avg on current rating=%.4f A",I_TAV);
```

Scilab code Exa 4.11 to calculate surge current rating and power rating

```
clear;
clc;
f=50;     //Hz
I_sb=3000;
t=1/(4*f);
T=1/(2*f);
I=sqrt(I_sb^2*t/T);     printf("surge current rating=
    %.2f A",I);
r=(I_sb/sqrt(2))^2*T;     printf("\nI^2*t rating=%.0f
     A^2.s",r);
```

Scilab code Exa 4.12 to find maximum value of the remedial parameter

```
clc
clear
V_s=300;//V
R=60;//ohm
L=2;//H
t=40*10^-6;//s
i_T=(V_s/R)*(1-exp(-R*t/L));
i=.036;//A
R1=V_s/(i-i_T);
```

```
printf("maximum value of remedial parameter=%.3f kilo-ohm",R1/1000);
```

Scilab code Exa 4.16 to calculate fault clearance time

```
clear;
clc;
V_p=230*sqrt(2);
R=1+((1)^-1+(10)^-1)^-1;
A=V_p/R;
s=1;     //s
t_c=20*A^-2*s;     printf("fault clearance time=%.4f ms",t_c*10^3);
```

Scilab code Exa 4.17 1 to calculate the max values of change in current and change in voltage for SCR 2 rms and avg current rating and 3 suitable voltage rating

```
clear;
clc;
V_s=sqrt(2)*230;     //V
L=15*10^-6;     //H
I=V_s/L;     //I=(di/dt)_max
printf("(di/dt)_max=%.3f A/usec",I/10^6);
R_s=10;    //ohm
v=I*R_s;     //v=(dv/dt)_max
printf("\n(dv/dt)_max=%.2f V/usec",v/10^6);

f=50;     //Hz
X_L=L*2*%pi*f;
R=2;
I_max=V_s/(R+X_L);     printf("\nI_rms=%.3f A",I_max);
```

```
disp("when conduction angle=90");
FF=%pi/sqrt(2);
I_TAV=I_max/FF;    printf("I_TAV=%.3f A",I_TAV);
disp("when conduction angle=30");
FF=3.98184;
I_TAV=I_max/FF;    printf("I_TAV=%.3f A",I_TAV);

printf("\nvoltage rating=%.3f V",2.75*V_s);//rating
    is taken 2.75 times of peak working voltage
    unlike 2.5 to 3 times as mentioned int book.
```

Scilab code Exa 4.19 1 to check heat sink selection is satisfactory 2 to choose heat sink for given requirements and ckt efficiency 3compute case and junction temp

```
clear;
clc;
T_jm=125;
th_jc=.15;    //degC/W
th_cs=0.075;    //degC/W

dT=54;    //dT=T_s-T_a
P_av=120;
th_sa=dT/P_av;
T_a=40;    //ambient temp
P_av=(T_jm-T_a)/(th_sa+th_jc+th_cs);
if((P_av-120)<1)
    disp("selection of heat sink is satisfactory");
end
dT=58;    //dT=T_s-T_a
P_av=120;
th_sa=dT/P_av;
T_a=40;    //ambient temp
P_av=(T_jm-T_a)/(th_sa+th_jc+th_cs);
if((P_av-120)<1)
```

```
    disp("selection of heat sink is satisfactory");
end

V_m=sqrt(2)*230;
R=2;
I_TAV=V_m/(R*%pi);
P_av=90;
th_sa=(T_jm-T_a)/P_av-(th_jc+th_cs);
dT=P_av*th_sa;
disp("for heat sink");    printf("T_s-T_a=%.2f degC",dT);    printf("\nP_av=%.0f W",P_av);
P=(V_m/2)^2/R;
eff=P/(P+P_av);    printf("\nckt efficiency=%.3f pu",eff);

a=60;    //delay angle
I_TAV=(V_m/(2*%pi*R))*(1+cosd(a));
printf("\nI_TAV=%.2f A",I_TAV);
dT=46;
T_s=dT+T_a;
T_c=T_s+P_av*th_cs;    printf("\ncase temp=%.2f degC",T_c);
T_j=T_c+P_av*th_jc;    printf("\njunction temp=%.2f degC",T_j);
```

Scilab code Exa 4.20 to find total avg power loss and percentage inc in device rating

```
clear;
clc;
T_j=125;    //degC
T_s=70;    //degC
th_jc=.16;    //degC/W
th_cs=.08;    //degC/W
P_av1=(T_j-T_s)/(th_jc+th_cs);    printf("total avg
```

```
    power loss in thristor sink combination=%.2f W",
    P_av1);

T_s=60;     //degC
P_av2=(T_j-T_s)/(th_jc+th_cs);

inc=(sqrt(P_av2)-sqrt(P_av1))*100/sqrt(P_av1);
    printf("\npercentage inc in rating=%.2f",inc);
```

Scilab code Exa 4.21 to determine voltage across each SCR and discharge current across each capacitor

```
clear;
clc;
R=25000;
I_l1=.021;     //I_l=leakage current
I_l2=.025;
I_l3=.018;
I_l4=.016;
//V1=(I-I_l1)*R;
//V2=(I-I_l2)*R;
//V3=(I-I_l3)*R;
//V4=(I-I_l4)*R;
//V=V1+V2+V3+V4
V=10000;
I_l=I_l1+I_l2+I_l3+I_l4;
//after solving
I=((V/R)+I_l)/4;
R_c=40;
V1=(I-I_l1)*R;     printf("voltage across SCR1=%.0f V
    ",V1);
V2=(I-I_l2)*R;     printf("\nvoltage across SCR2=%.0f
     V",V2);
V3=(I-I_l3)*R;     printf("\nvoltage across SCR3=%.0f
     V",V3);
```

```
V4=(I-I_14)*R;     printf("\nvoltage across SCR4=%.0f V",V4);

I1=V1/R_c;     printf("\ndischarge current through SCR1=%.3f A",I1);
I2=V2/R_c;     printf("\ndischarge current through SCR2=%.3f A",I2);
I3=V3/R_c;     printf("\ndischarge current through SCR3=%.3f A",I3);
I4=V4/R_c;     printf("\ndischarge current through SCR4=%.3f A",I4);
```

Scilab code Exa 4.22 to calculate number of series and parrallel units of SCRs

```
clear;
clc;
V_r=1000;     //rating of SCR
I_r=200;     //rating of SCR
V_s=6000;     //rating of String
I_s=1000;     //rating of String
disp("when DRF=.1");
DRF=.1;
n_s=V_s/(V_r*(1-DRF));     printf("number of series units=%.0f",ceil(n_s));
n_p=I_s/(I_r*(1-DRF));     printf("\nnumber of parrallel units=%.0f",ceil(n_p));
disp("when DRF=.2");
DRF=.2;
n_s=V_s/(V_r*(1-DRF));     printf("number of series units=%.0f",ceil(n_s));
n_p=I_s/(I_r*(1-DRF));     printf("\nnumber of parrallel units=%.0f",ceil(n_p));
```

Scilab code Exa 4.23 to calculate the resistance

```
clear;
clc;
V1=1.6;      //on state voltage drop of SCR1
V2=1.2;      //on state voltage drop of SCR2
I1=250;      //current rating of SCR1
I2=350;      //current rating of SCR2
R1=V1/I1;
R2=V2/I2;
I=600;     //current to be shared
//for SCR1,      I*(R1+R)/(total resistance)=k*I1      (1)
//for SCR2,      I*(R2+R)/(total resistance)=k*I2      (2)
//(1)/(2)
R=(R2*I2-R1*I1)/(I1-I2);
printf("reqd value of resistance=%.3f ohm",R);
```

Scilab code Exa 4.25 to compute value of various resistances

```
clear;
clc;
f=2000;      //Hz
C=0.04*10^-6;
n=.72;
R=1/(f*C*log(1/(1-n)));     printf("R=%.2f kilo-ohm",
   R/1000);
V_p=18;
V_BB=V_p/n;
R2=10^4/(n*V_BB);     printf("\nR2=%.2f ohm",R2);
I=4.2*10^-3;     //leakage current
```

```
R_BB=5000;
R1=(V_BB/I)-R2-R_BB;     printf("\nR1=%.0f ohm",R1);
```

Scilab code Exa 4.26 to compute max and min values of R and the corresponding freq

```
clear;
clc;
V_p=18;
n=.72;
V_BB=V_p/n;
I_p=.6*10^-3;
I_v=2.5*10^-3;
V_v=1;
R_max=V_BB*(1-n)/I_p;     printf("R_max=%.2f kilo-ohm",R_max/1000);
R_min=(V_BB-V_v)/I_v;     printf("\nR_min=%.2f kilo-ohm",R_min/1000);

C=.04*10^-6;
f_min=1/(R_max*C*log(1/(1-n)));     printf("\nf_min=%.3f kHz",f_min/1000);
f_max=1/(R_min*C*log(1/(1-n)));     printf("\nf_max=%.2f kHz",f_max/1000);
```

Scilab code Exa 4.27 to find max and min firing angles for triac

```
clear;
clc;
R1=1000;
C=.5*10^-6;
f=50;
w=2*%pi*f;
```

```
V_s=230;
X_c=1/(w*C);
v_c=30;
R=0;
Z=sqrt((R+R1)^2+X_c^2);
phi=atand(X_c/(R+R1));
I1=V_s/(Z*complex(cosd(-phi),sind(-phi)));
V_c=I1*X_c*complex(cosd(-90),sind(-90));
a=abs(V_c);    //magnitude of V_c
b=-atand(imag(V_c)/real(V_c));    //argument of V_c
//v_c=sqrt(2)*a*sind(a1-b)
a1=asind(v_c/(sqrt(2)*a))+b;    printf("min angle=%.1f deg",a1);
R=25000;
Z=sqrt((R+R1)^2+X_c^2);
phi=atand(X_c/(R+R1));
I1=V_s/(Z*complex(cosd(-phi),sind(-phi)));
V_c=I1*X_c*complex(cosd(-90),sind(-90));
a=abs(V_c);    //magnitude of V_c
b=-atand(imag(V_c)/real(V_c));    //argument of V_c
//v_c=sqrt(2)*a*sind(a2-b)
a2=asind(v_c/(sqrt(2)*a))+b;    printf("\nmax angle=%.2f deg",a2);
```

Chapter 5

Thyristor Commutation Techniques

Scilab code Exa 5.1 to determine 1 conduction time and 2 voltage across thyristor

```
clear;
clc;
L=5*10^-3;
C=20*10^-6;
V_s=200;
w_o=sqrt(1/(L*C));
t_o=%pi/w_o;    printf("conduction time of thyristor
   =%.5f ms",t_o*1000);
printf("\nvoltage across thyristor=%.0f V",-V_s);
```

Scilab code Exa 5.2 to calculate 1 conduction time for auxillery thyristor 2 voltage across and 3 circuit turn off time for main thyristor

```
clear;
clc;
```

```
C=20*10^-6;
L=5*10^-6;
V_s=230;
I_p=V_s*sqrt(C/L);
w_o=sqrt(1/(L*C));
t_o=%pi/w_o;    printf("conduction time of auxillery thyristor=%.3f us",t_o*10^6);
I_o=300;
//a=w_o*(t3-t2)=asind(I_o/(2*V_s));
a=asind(I_o/(2*V_s));
V_ab=V_s*cosd(a);    printf("\nvoltage across main thyristor=%.3f V",V_ab);
t_c=C*V_ab/I_o;    printf("\nckt turn off time=%.3f us",t_c*10^6);
```

Scilab code Exa 5.3 to determine 1 peak value of current 2 value of capacitor C

```
clear;
clc;
V_s=200;
R1=10;
R2=100;
I1=V_s*(1/R1+2/R2);    printf("peak value of current through SCR1=%.0f A",I1);
I2=V_s*(2/R1+1/R2);    printf("\npeak value of current through SCR2=%.0f A",I2);
t_c1=40*10^-6;
fos=2;    //factor of safety
C1=t_c1*fos/(R1*log(2));
C2=t_c1*fos/(R2*log(2));
if(C1>C2)
    printf("\nvalue of capacitor=%.4f uF",C1*10^6);
else
    printf("\nvalue of capacitor=%.4f uF",C2*10^6);
```

```
end
```

Scilab code Exa 5.4 to calculate 1 value of current 2 circuit turn off time for main and auxillery thyristor

```
clear;
clc;
V_s=230;
L=20*10^-6;
C=40*10^-6;
I_o=120;
I_p=V_s*sqrt(C/L);
printf("current through main thyristor=%.2f A",I_o+
   I_p);
printf("\ncurrent through auxillery thyristor=%.0f A
   ",I_o);

t_c=C*V_s/I_o;    printf("\ncircuit turn off time
   for main thyristor=%.2f us",t_c*10^6);
w_o=sqrt(1/(L*C));
t_c1=%pi/(2*w_o);    printf("\ncircuit turn off time
    for auxillery thyristor=%.2f us",t_c1*10^6);
```

Scilab code Exa 5.5 to compute min value of C

```
clear;
clc;
C_j=25*10^-12;
I_c=5*10^-3;     //charging current
V_s=200;
R=50;
C=(C_j*V_s)/(I_c*R);    printf("Value of C=%.2f uF",
   C*10^6);
```

Scilab code Exa 5.6 to find circuit turn off time

```
clear;
clc;
V_s=200;
R=5;
C=10*10^-6;
//for turn off V_s*(1-2*exp(-t/(R*C)))=0,     so after solving
t_c=R*C*log(2);     printf("circuit turn off time=%.4f us",t_c*10^6);
```

Scilab code Exa 5.7 to find conduction time of thyristor

```
clear;
clc;
R=1;
L=20*10^-6;
C=40*10^-6;
w_r=sqrt((1/(L*C))-(R/(2*L))^2);
t_1=%pi/w_r;     printf("conduction time of thyristor=%.3f us",t_1*10^6);
```

Scilab code Exa 5.8 1 to calculate value of Capacitor 2 determine value of Resistance

```
clear;
clc;
dv=400*10^-6;       //dv=dv_T/dt(V/s)
```

```
V_s=200;
R=20;
C=V_s/(R*dv);
C_j=.025*10^-12;
C_s=C-C_j;    printf("C_s=%.3f uF",C_s/10^6);

I_T=40;
R_s=1/((I_T/V_s)-(1/R));    printf("\nR_s=%.3f ohm",
    R_s);
//value of R_s in book is wrongly calculated
```

Scilab code Exa 5.9 calculate 1 time at which commutation of main thyristor gets initiated 2 ckt turn off time

```
clear;
clc;
V_s=200;
C=20*10^-6;
L=.2*10^-3;
i_c=10;
i=V_s*sqrt(C/L);
w_o=1/sqrt(L*C);
t_1=(1/w_o)*asin(i_c/i);    printf("reqd time=%.0f
    us",t_1*10^6);

t_o=%pi/w_o;
t_c=t_o-2*t_1;    printf("\nckt turn off time=%.1f
    us",t_c*10^6);
//solution in book wrong, as wrong values are
    selected while filling the formuleas
```

Scilab code Exa 5.11 to find current in R and L

```
clear;
clc;
L=1;
R=50;
V_s=200;
tau=L/R;
t=.01;
i=(V_s/R)*(1-exp(-t/tau));
Vd=.7;
t=8*10^-3;
i1=i-t*Vd;     printf("current through L=%.4f A",i1);
i_R=0;     //current in R at t=.008s
printf("\ncurrent through R=%.0f A",i_R);
```

Scilab code Exa 5.12 To find the current in R and L and voltage across C

```
clc
clear
L=1;//H
R=50;//ohm
V_s=200;//V
tau=L/R;
t=.01;//s
i=(V_s/R)*(1-exp(-t/tau));     printf("current in R,L
   =%.3f A",i);
C=1*10^-6;//F
V_c=sqrt(L/C)*i;
printf("\nvoltage across C=%.3f kV",V_c/1000);
```

Chapter 6

Phase Controlled Rectifiers

Scilab code Exa 6.1 to calculate the power absorbed in the heater element

```
clear;
clc;
V=230;
P=1000;
R=V^2/P;
disp("when firing angle delay is of 45deg");
a=%pi/4;
V_or=(sqrt(2)*V/(2*sqrt(%pi)))*sqrt((%pi-a)+.5*sin
    (2*a));
P=V_or^2/R;     printf("power absorbed=%.2f W",P);

disp("when firing angle delay is of 90deg");
a=%pi/2;
V_or=(sqrt(2)*V/(2*sqrt(%pi)))*sqrt((%pi-a)+.5*sin
    (2*a));
P=V_or^2/R;     printf("power absorbed=%.2f W",P);
```

Scilab code Exa 6.2 1 to find value of charging current 2 power supplied to battery and dissipated by resistor 3 calculate the supply pf

```
clear;
clc;
V=230;
E=150;
R=8;
th1=asind(E/(sqrt(2)*V));
I_o=(1/(2*%pi*R))*(2*sqrt(2)*230*cosd(th1)-E*(%pi-2*th1*%pi/180));
printf("avg charging curent=%.4f A",I_o);

P=E*I_o;    printf("\npower supplied to the battery=%.2f W",P);
I_or=sqrt((1/(2*%pi*R^2))*((V^2+E^2)*(%pi-2*th1*%pi/180)+V^2*sind(2*th1)-4*sqrt(2)*V*E*cosd(th1)));
P_r=I_or^2*R;    printf("\npower dissipated by the resistor=%.3f W",P_r);

pf=(P+P_r)/(V*I_or);    printf("\nsupply pf=%.3f",pf);
```

Scilab code Exa 6.3 1 to find value of charging current 2 power supplied to battery and dissipated by resistor 3 calculate the supply pf

```
clear;
clc;
V=230;
E=150;
R=8;
a=35;
th1=asind(E/(sqrt(2)*V));
th2=180-th1;
I_o=(1/(2*%pi*R))*(sqrt(2)*230*(cosd(a)-cosd(th2))-E*((th2-a)*%pi/180));
printf("avg charging curent=%.4f A",I_o);

```

```
P=E*I_o;    printf("\npower supplied to the battery=%.2f W",P);
I_or=sqrt((1/(2*%pi*R^2))*((V^2+E^2)*((th2-a)*%pi/180)-(V^2/2)*(sind(2*th2)-sind(2*a))-2*sqrt(2)*V*E*(cosd(a)-cosd(th2))));
P_r=I_or^2*R;    printf("\npower dissipated by the resistor=%.2f W",P_r);

pf=(P+P_r)/(V*I_or);    printf("\nsupply pf=%.4f",pf);
//Answers have small variations from that in the book due to difference in the rounding off of digits.
```

Scilab code Exa 6.4 to find ckt turn off time avg output voltage and avg load current

```
clear;
clc;
B=210;
f=50;    //Hz
w=2*%pi*f;
a=40;    //firing angle
V=230;
disp("for R=5ohm and L=2mH");
R=5;
L=2*10^-3;
t_c=(360-B)*%pi/(180*w);    printf("ckt turn off time=%.3f msec",t_c*1000);
V_o=(sqrt(2)*230/(2*%pi))*(cosd(a)-cosd(B));    printf("\navg output voltage=%.3f V",V_o);
I_o=V_o/R;    printf("\navg output current=%.4f A",I_o);

disp("for R=5ohm, L=2mH and E=110V");
```

```
E=110;
R=5;
L=2*10^-3;
th1=asind(E/(sqrt(2)*V));
t_c=(360-B+th1)*%pi/(180*w);    printf("ckt turn off time=%.3f msec",t_c*1000);
V_o=(sqrt(2)*230/(2*%pi))*(cosd(a)-cosd(B)); printf("\navg output voltage=%.3f V",V_o);
I_o=(1/(2*%pi*R))*(sqrt(2)*230*(cosd(a)-cosd(B))-E*((B-a)*%pi/180));    printf("\navg output current=%.4f A",I_o);
V_o=R*I_o+E;    printf("\navg output voltage=%.3f V",V_o);
```

Scilab code Exa 6.5 to determine 1 rectification efficiency 2 form factor 3 voltage ripple factor 4 transformer utilisation factor 5 PIV of thyristor

```
clear;
clc;
V_s=230;
f=50;
R=10;
a=60;
V_m=(sqrt(2)*V_s);
V_o=V_m/(2*%pi)*(1+cosd(a));
I_o=V_o/R;
V_or=(V_m/(2*sqrt(%pi)))*sqrt((%pi-a*%pi/180)+.5*sind(2*a));
I_or=V_or/R;
P_dc=V_o*I_o;
P_ac=V_or*I_or;
RE=P_dc/P_ac;    printf("rectification efficiency=%.4f",RE);
FF=V_or/V_o;    printf("\nform factor=%.3f",FF);
VRF=sqrt(FF^2-1);    printf("\nvoltage ripple factor
```

```
=%.4f",VRF);
TUF=P_dc/(V_s*I_or);     printf("\nt/f utilisation factor=%.4f",TUF);
PIV=V_m;     printf("\nPIV of thyristor=%.2f V",PIV);
```

Scilab code Exa 6.6 to find power handled by mid pt convertor and single phase bridge convertor

```
clear;
clc;
V=1000;
fos=2.5;     //factor of safety
I_TAV=40;
disp("for mid pt convertor");
V_m=V/(2*fos);
P=(2*V_m/%pi)*I_TAV;     printf("power handled=%.3f kW",P/1000);
disp("for bridge convertor");
V_m=V/(fos);
P=(2*V_m/%pi)*I_TAV;     printf("power handled=%.3f kW",P/1000);
```

Scilab code Exa 6.7 compute firing angle delay and pf

```
clear;
clc;
V_s=230;
V_m=sqrt(2)*V_s;
R=.4;
I_o=10;
I_or=I_o;
E=120;
```

```
a=acosd((E+I_o*R)*%pi/(2*V_m));    printf("firing angle delay=%.2f deg",a);
pf=(E*I_o+I_or^2*R)/(V_s*I_or);    printf("\npf=%.4f",pf);

E=-120;
a=acosd((E+I_o*R)*%pi/(2*V_m));    printf("\nfiring angle delay=%.2f deg",a);
pf=(-E*I_o-I_or^2*R)/(V_s*I_or);   printf("\npf=%.4f",pf);
```

Scilab code Exa 6.9 to find avg output current and power delivered

```
clear;
clc;
V_s=230;
f=50;
a=45;
R=5;
E=100;
V_o=((sqrt(2)*V_s)/(2*%pi))*(3+cosd(a));
I_o=(V_o-E)/R;    printf("avg o/p current=%.3f A",I_o);
P=E*I_o;    printf("\npower delivered to battery=%.4f kW",P/1000);
```

Scilab code Exa 6.10 to find avg value of load current and new value under given changed conditions

```
clear;
clc;
V_s=230;
f=50;
```

```
a=50;
R=6;
E=60;
V_o=((sqrt(2)*2*V_s)/(%pi))*(cosd(a));
I_o=(V_o-E)/R;    printf("avg o/p current=%.3f A",I_o);

//ATQ after applying the conditions
V_o=((sqrt(2)*V_s)/(%pi))*(cosd(a));
I_o=(V_o-E)/R;    printf("\navg o/p current after change=%.4f A",I_o);
```

Scilab code Exa 6.11 to calculate the input and output performance parameters for full conductor

```
clear;
clc;
V_s=230;
V_m=sqrt(2)*V_s;
a=45;
R=10;
V_o=(2*V_m/%pi)*cosd(a);
I_o=V_o/R;
V_or=V_m/sqrt(2);
I_or=I_o;
P_dc=V_o*I_o;
P_ac=V_or*I_or;
RE=P_dc/P_ac;    printf("rectification efficiency=%.4f",RE);
FF=V_or/V_o;    printf("\nform factor=%.4f",FF);
VRF=sqrt(FF^2-1);    printf("\nvoltage ripple factor=%.4f",VRF);
I_s1=2*sqrt(2)*I_o/%pi;
DF=cosd(a);
CDF=.90032;
```

```
pf=CDF*DF;     printf("\npf=%.5f",pf);
HF=sqrt((1/CDF^2)-1);     printf("\nHF=%.5f",HF);
printf("\nactive power=%.2f W",P_dc);
Q=2*V_m*I_o*sind(a)/%pi;     printf("\nreactive power
    =%.3f Var",Q);
```

Scilab code Exa 6.12 to calculate the input and output performance parameters for single phase semi conductor

```
clear;
clc;
V_s=230;
V_m=sqrt(2)*V_s;
a=45;
R=10;
V_o=(V_m/%pi)*(1+cosd(a));
I_o=V_o/R;
V_or=V_s*sqrt((1/%pi)*((%pi-a*%pi/180)+sind(2*a)/2))
    ;
I_or=I_o;
P_dc=V_o*I_o;
P_ac=V_or*I_or;
RE=P_dc/P_ac;     printf("rectification efficiency=%
    .4f",RE);
FF=V_or/V_o;     printf("\nform factor=%.3f",FF);
VRF=sqrt(FF^2-1);     printf("\nvoltage ripple factor
    =%.3f",VRF);
I_s1=2*sqrt(2)*I_o*cosd(a/2)/%pi;
DF=cosd(a/2);     printf("\nDF=%.4f",DF);
CDF=2*sqrt(2)*cosd(a/2)/sqrt(%pi*(%pi-a*%pi/180));
        printf("\nCDF=%.4f",CDF);
pf=CDF*DF;     printf("\npf=%.4f",pf);
HF=sqrt((1/CDF^2)-1);     printf("\nHF=%.4f",HF);
printf("\nactive power=%.3f W",P_dc);
Q=V_m*I_o*sind(a)/%pi;     printf("\nreactive power=%
```

```
.2f Var",Q);
//Answers have small variations from that in the book due to difference in the rounding off of digits.
```

Scilab code Exa 6.13 determine 1 firing angle 2 avg and rms values of load current 3 rectification efficiency

```
clear;
clc;
V_s=230;
R=10;
V_ml=sqrt(2)*V_s;
V_om=3*V_ml/(2*%pi);
V_o=V_om/2;
th=30;
a=acosd((2*%pi*sqrt(3)*V_o/(3*V_ml)-1))-th;
    printf("delay angle=%.1f deg",a);
I_o=V_o/R;     printf("\navg load current=%.3f A",I_o
    );
V_or=V_ml/(2*sqrt(%pi))*sqrt((5*%pi/6-a*%pi/180)+.5*
    sind(2*a+2*th));
I_or=V_or/R;     printf("\nrms load current=%.3f A",
    I_or);
RE=V_o*I_o/(V_or*I_or);     printf("\nrectification
    efficiency=%.4f",RE);
```

Scilab code Exa 6.15 to calculate 1 avg value of load volatage 2 avg and rms current and PIV 3 avg power dissipation in each thyristor

```
clear;
clc;
V=400;
```

```
V_ml=sqrt(2)*V;
v_T=1.4;
disp("for firing angle = 30deg");
a=30;
V_o=3*V_ml/(2*%pi)*cosd(a)-v_T;    printf("avg output voltage=%.3f V",V_o);
disp("for firing angle = 60deg");
a=60;
V_o=3*V_ml/(2*%pi)*cosd(a)-v_T;    printf("avg output voltage=%.2f V",V_o);

I_o=36;
I_TA=I_o/3;    printf("\navg current rating=%.0f A",I_TA);
I_Tr=I_o/sqrt(3);    printf("\nrms current rating=%.3f A",I_Tr);
printf("\nPIV of SCR=%.1f V",V_ml);

P=I_TA*v_T;    printf("\npower dissipated=%.1f W",P);
```

Scilab code Exa 6.17 to compute firing angle delay and supply pf

```
clear;
clc;
E=200;
I_o=20;
R=.5;
V_o=E+I_o*R;
V_s=230;
V_ml=sqrt(2)*V_s;
a=acosd(V_o*%pi/(3*V_ml));    printf("firing angle delay=%.3f deg",a);
th=120;
I_s=sqrt((1/%pi)*I_o^2*th*%pi/180);
```

```
P=E*I_o+I_o^2*R;
pf=P/(sqrt(3)*V_s*I_s);     printf("\npf=%.3f",pf);

V_o=E-I_o*R;
a=acosd(-V_o*%pi/(3*V_ml));     printf("\nfiring angle delay=%.2f deg",a);
```

Scilab code Exa 6.18 to find commutation time and reverse voltage across SCR

```
clear;
clc;
V=230;
f=50;
w=2*%pi*f;
disp("for firing angle delay=0deg");
a=0;
t_c=(4*%pi/3-a*%pi/180)/w;     printf("commutation time=%.2f ms",t_c*1000);
printf("\npeak reverse voltage=%.2f V",sqrt(2)*V);

disp("for firing angle delay=30deg");
a=30;
t_c=(4*%pi/3-a*%pi/180)/w;     printf("commutation time=%.2f ms",t_c*1000);
printf("\npeak reverse voltage=%.2f V",sqrt(2)*V);
```

Scilab code Exa 6.19 to find the magnitude of per phase input supply voltage

```
clear;
clc;
a=30;
```

```
R=10;
P=5000;
V_s=sqrt(P*R*2*%pi/(2*3)/(%pi/3+sqrt(3)*cosd(2*a)/2)
   );
V_ph=V_s/sqrt(3);    printf("per phase voltage, V_ph
   =%.3f V",V_ph);
I_or=sqrt(P*R);
V_s=I_or*%pi/(sqrt(2)*3*cosd(a));
V_ph=V_s/sqrt(3);
printf("\nfor constant load current");
printf("\nV_ph=%.2f V",V_ph);
```

Scilab code Exa 6.20 to find magnitude of input per phase supply voltage

```
clear;
clc;
a=30;
R=10;
P=5000;
V_s=sqrt(P*R*4*%pi/(2*3)/(2*%pi/3+sqrt(3)*(1+cosd(2*
   a))/2));
V_ph=V_s/sqrt(3);    printf("per phase voltage, V_ph
   =%.3f V",V_ph);
I_or=sqrt(P*R);
V_s=I_or*2*%pi/(sqrt(2)*3*(1+cosd(a)));
V_ph=V_s/sqrt(3);
printf("\nfor constant load current");
printf("\nV_ph=%.2f V",V_ph);
```

Scilab code Exa 6.21 to find magnitude of input per phase supply voltage

```
clear;
clc;
```

```
a=90;
R=10;
P=5000;
V_s=sqrt(P*R*4*%pi/(2*3)/((%pi-%pi/2)+(sind(2*a))/2));
V_ph=V_s/sqrt(3);    printf("per phase voltage, V_ph=%.2f V",V_ph);
I_or=sqrt(P*R);
V_s=I_or*2*%pi/(sqrt(2)*3*(1+cosd(a)));
V_ph=V_s/sqrt(3);
printf("\nfor constant load current");
printf("\nV_ph=%.1f V",V_ph);
```

Scilab code Exa 6.22 to compute firing angle delay and supply pf

```
clear;
clc;
E=200;
I_o=20;
R=.5;
V_o=E+I_o*R;
V_s=230;
V_ml=sqrt(2)*V_s;
a=acosd(V_o*2*%pi/(3*V_ml)-1);    printf("firing angle delay=%.2f deg",a);
a1=180-a;
I_sr=sqrt((1/%pi)*I_o^2*(a1*%pi/180));
P=V_o*I_o;
pf=P/(sqrt(3)*V_s*I_sr);    printf("\npf=%.4f",pf);
```

Scilab code Exa 6.23 to calculate rectification efficiency TUF and input power factor

```
clear;
clc;
V_s=400;
f=50;
I_o=15;
a=45;
I_TA=I_o*120/360;
I_Tr=sqrt(I_o^2*120/360);
I_sr=sqrt(I_o^2*120/180);
V_ml=sqrt(2)*V_s;
V_o=3*V_ml*cosd(a)/%pi;
V_or=V_ml*sqrt((3/(2*%pi))*(%pi/3+sqrt(3/2)*cosd(2*a
   )));
I_or=I_o;
P_dc=V_o*I_o;
P_ac=V_or*I_or;
RE=P_dc/P_ac;     printf("rectification  efficiency=%
   .5f",RE);
VA=3*V_s/sqrt(3)*I_sr;
TUF=P_dc/VA;     printf("\nTUF=%.4f",TUF);
pf=P_ac/VA;     printf("\ninput  pf=%.3f",pf);
```

Scilab code Exa 6.24 to find DF CDF THD and pf and to calculate the active and reative input powers

```
clear;
clc;
I=10;
a=45;
V=400;
f=50;
DF=cosd(a);
printf("DF=%.3f",DF);
I_o=10;
I_s1=4*I_o/(sqrt(2)*%pi)*sin(%pi/3);
```

```
I_sr=I_o*sqrt(2/3);
I_o=1;      //suppose
CDF=I_s1/I_sr;      printf("\nCDF=%.3f",CDF);
THD=sqrt(1/CDF^2-1);      printf("\nTHD=%.5f",THD);
pf=CDF*DF;      printf("\nPF=%.4f",pf);
P=(3*sqrt(2)*V*cosd(a)/%pi)*I;      printf("\nactive
    power=%.2f W",P);
Q=(3*sqrt(2)*V*sind(a)/%pi)*I;      printf("\nreactive
     power=%.2f Var",Q);
```

Scilab code Exa 6.25 calculate the power delivered to load and input pf

```
//calculate the power delivered to load and i/p pf

clc;
disp("for firing angle=30deg");
a=30;
V=400;
V_ml=sqrt(2)*V;
V_o=3*V_ml*cosd(a)/%pi;
E=350;
R=10;
I_o=(V_o-E)/R;
I_or=I_o;
P=V_o*I_o;      printf("power delivered to load=%.2f W
    ",P);
I_sr=I_o*sqrt(2/3);
VA=3*V/sqrt(3)*I_sr;
pf=P/VA;      printf("\npf=%.4f",pf);

disp("for firing advance angle=60deg");
a=180-60;
V=400;
V_ml=sqrt(2)*V;
V_o=3*V_ml*cosd(a)/%pi;
```

```
E=-350;
R=10;
I_o=(V_o-E)/R;
I_or=I_o;
P=-V_o*I_o;     printf("power delivered to load=%.2f
   W",P);
I_sr=I_o*sqrt(2/3);
VA=3*V/sqrt(3)*I_sr;
pf=P/VA;     printf("\npf=%.4f",pf);
//Answers have small variations from that in the
   book due to difference in the rounding off of
   digits.
```

Scilab code Exa 6.26 calculate overlap angle for different firing angles

```
clear;
clc;
a=0;
u=15;
i=cosd(a)-cosd(a+u);
disp("for firing angle=30deg");
a=30;
u=acosd(cosd(a)-i)-a;     printf("overlap angle=%.1f
   deg",u);
disp("for firing angle=45deg");
a=45;
u=acosd(cosd(a)-i)-a;     printf("overlap angle=%.1f
   deg",u);
disp("for firing angle=60deg");
a=60;
u=acosd(cosd(a)-i)-a;     printf("overlap angle=%.2f
   deg",u);
```

Scilab code Exa 6.28 to calculate firing angle delay and overlap angle

```
clear;
clc;
E=400;
I_o=20;
R=1;
V_o=E+I_o*R;
f=50;
w=2*%pi*f
L=.004;
V=230;//per phase voltage
V_ml=sqrt(6)*V;
a=acosd(%pi/(3*V_ml)*(V_o+3*w*L*I_o/%pi));    printf("firing angle delay=%.3f deg",a);
u=acosd(%pi/(3*V_ml)*(V_o-3*w*L*I_o/%pi))-a;    printf("\noverlap angle=%.2f deg",u);
//Answers have small variations from that in the book due to difference in the rounding off of digits.
```

Scilab code Exa 6.29 to calculate firing angle firing angle delay and overlap angle

```
clear;
clc;
V=400;
f=50;
w=2*%pi*f;
R=1;
E=230;
I=15;
V_o=-E+I*R;
V_ml=sqrt(2)*V;
a=acosd(V_o*2*%pi/(3*V_ml));    printf("firing angle
```

```
=%.3f deg",a);
L=0.004;
a=acosd((2*%pi)/(3*V_ml)*(V_o+3*w*L*I/(2*%pi))); printf("\nfiring angle delay=%.3f deg",a);
u=acosd(cosd(a)-3*f*L*I/V_ml)-a;    printf("\noverlap angle=%.3f deg",u);
//Answers have small variations from that in the book due to difference in the rounding off of digits.
```

Scilab code Exa 6.31 calculate the peak value of circulating current

```
clear;
clc;
V=230;//per phase
V_ml=sqrt(3)*sqrt(2)*V;
f=50;
w=2*%pi*f;
a1=60;
L=0.015;
i_cp=(sqrt(3)*V_ml/(w*L))*(1-sind(a1));    printf("circulating current=%.4f A",i_cp);
```

Scilab code Exa 6.32 to determine 1 avg output voltage 2 avg output current 3 avg and rms values of thyristor currents 4 pf

```
clear;
clc;
V=230;
V_m=sqrt(2)*V;
a=30;
V_o=2*V_m*cosd(a)/%pi;    printf("avg o/p voltage=%.3f V",V_o);
```

```
R=10;
I_o=V_o/R;     printf("\navg o/p current=%.2f A",I_o);
I_TA=I_o*%pi/(2*%pi);     printf("\navg value of thyristor current=%.3f A",I_TA);
I_Tr=sqrt(I_o^2*%pi/(2*%pi));     printf("\nrms value of thyristor current=%.2f A",I_Tr);
I_s=sqrt(I_o^2*%pi/(%pi));
I_o=I_s;
pf=(V_o*I_o/(V*I_s));     printf("\npf=%.4f",pf);
//Answers have small variations from that in the book due to difference in the rounding off of digits.
```

Scilab code Exa 6.33 to determine 1 avg output voltage 2 angle of overlap 3 pf

```
clear;
clc;
V=230;
V_m=sqrt(2)*V;
a=30;
L=.0015;
V_o=2*V_m*cosd(a)/%pi;
R=10;
I_o=V_o/R;
f=50;
w=2*%pi*f;
V_ox=2*V_m*cosd(a)/%pi-w*L*I_o/%pi;     printf("avg o/p voltage=%.3f V",V_ox);
u=acosd(cosd(a)-I_o*w*L/V_m)-a;     printf("\nangle of overlap=%.3f deg",u);
I=I_o;
pf=V_o*I_o/(V*I);     printf("\npf=%.4f",pf);
//Answers have small variations from that in the
```

```
    book due to difference in the rounding off of
    digits.
```

Scilab code Exa 6.34 calculate the generator mean voltage

```
clear;
clc;
V=415;
V_ml=sqrt(2)*V;
a1=35;//firing angle advance
a=180-a1;
I_o=80;
r_s=0.04;
v_T=1.5;
X_l=.25;//reactance=w*L
E=-3*V_ml*cosd(a)/%pi+2*I_o*r_s+2*v_T+3*X_l*I_o/%pi;
    printf("mean generator voltage=%.3f V",E);
```

Scilab code Exa 6.35 find the mean value of E

```
clear;
clc;
V=415;
V_ml=sqrt(2)*V;
R=.2;
I_o=80;
r_s=0.04;
v_T=1.5;
X_l=.25;//reactance=w*L

disp("when firing angle=35deg");
a=35;
```

```
E=-(-3*V_ml*cosd(a)/%pi+I_o*R+2*I_o*r_s+2*v_T+3*X_l*
    I_o/%pi);    printf("mean generator voltage=%.3f
    V",E);
disp("when firing angle advance=35deg");
a1=35;
a=180-a1;
E=(-3*V_ml*cosd(a)/%pi+I_o*R+2*I_o*r_s+2*v_T+3*X_l*
    I_o/%pi);    printf("mean generator voltage=%.3f
    V",E);
```

Scilab code Exa 6.36 to find avg current through battery

```
clear;
clc;
R=5;
V=230;
V_mp=sqrt(2)*V;
a=30;
E=150;
B=180-asind(E/V_mp);
I_o=(3/(2*%pi*R))*(V_mp*(cosd(a+30)-cosd(B))-E*((B-a
    -30)*%pi/180));
printf("avg current flowing=%.2f A",I_o);
```

Scilab code Exa 6.37 to determine 1 avg outpunputt voltage 2 avg output current 3 avg and rms values of thyristor currents 4 avg and rms values of diodes currents 5 i pf 6 ckt turn off time

```
clear;
clc;
a=30;
V=230;
V_m=sqrt(2)*V;
```

```
V_o=V_m*(1+cosd(a))/%pi;     printf("avg o/p voltage=
    %.3f V",V_o);
E=100;
R=10;
I_o=(V_o-E)/R;     printf("\navg o/p current=%.2f A",
    I_o);
I_TA=I_o*%pi/(2*%pi);     printf("\navg value of
    thyristor current=%.2f A",I_TA);
I_Tr=sqrt(I_o^2*%pi/(2*%pi));     printf("\nrms value
     of thyristor current=%.3f A",I_Tr);
printf("\navg value of diode current=%.2f A",I_TA);
printf("\nrms value of diode current=%.3f A",I_Tr);
I_s=sqrt(I_o^2*(1-a/180)*%pi/(%pi));
I_or=I_o;
P=E*I_o+I_or^2*R;
pf=(P/(V*I_s));     printf("\npf=%.4f",pf);
f=50;
w=2*%pi*f;
t_c=(1-a/180)*%pi/w;     printf("\ncircuit turn off
    time=%.2f ms",t_c*1000);
```

Scilab code Exa 6.38 to calculate peak value of circulating currents and of both convertors

```
clear;
clc;
V=230;
V_m=sqrt(2)*V;
L=0.05;
f=50;
w=2*%pi*f;
a=30;
i_cp=2*V_m*(1-cosd(a))/(w*L);     printf("peak value
    of circulating current=%.3f A",i_cp);
R=30;
```

```
i_l=V_m/R;
i1=i_cp+i_l;    printf("\npeak value of current in
    convertor 1=%.3f A",i1);
i2=i_cp;    printf("\npeak value of current in
    convertor 2=%.3f A",i2);
```

Scilab code Exa 6.39 to estimate triggering angle for no current transients and for worst transients

```
clear;
clc;
f=50;
w=2*%pi*f;
R=5;
L=0.05;
disp("for no current transients");
phi=atand(w*L/R);    printf("triggering angle=%.2f
    deg",phi);
disp("for worst transients");
phi=90+atand(w*L/R);    printf("triggering angle=%.2
    f deg",phi);
```

Chapter 7

Choppers

Scilab code Exa 7.2 to calculate 1 avg and rms values of output voltage 2 chopper efficiency

```
clear;
clc;
a=.4;      //duty cycle ,a=T_on/T
V_s=230;
R=10;
V=a*(V_s-2);     printf("avg o/p voltage=%.1f V",V);
V_or=sqrt(a*(V_s-2)^2);     printf("\nrms value of o/
   p voltage=%.1f V",V_or);
P_o=V_or^2/R;
P_i=V_s*V/R;
n=P_o*100/P_i;     printf("\nchopper efficiency in
   percentage=%.2f",n);
```

Scilab code Exa 7.3 to compute pulse width of output voltage and avg value of new output voltage

```
clear;
```

```
clc;
V_i=220;
V_o=660;
a=1-V_i/V_o;
T_on=100;    //microsecond
T=T_on/a;
T_off=T-T_on;    printf("pulse width of o/p voltage=%.0f us",T_off);

T_off=T_off/2;
T_on=T-T_off;
a=T_on/T;
V_o=V_i/(1-a);    printf("\nnew o/p voltage=%.0f V",V_o);
```

Scilab code Exa 7.4 to find time ratio for chopper

```
clear;
clc;
I_1=12;
I_2=16;
I_0=(I_1+I_2)/2;
R=10;
V_0=I_0*R;
V_s=200;
a=V_0/V_s;
r=a/(1-a);    printf("time ratio(T_on/T_off)=%.3f",r);
```

Scilab code Exa 7.5 to compute pulse width of output voltage and avg value of new output voltage

```
clear;
```

```
clc;
V_o=660;
V_s=220;
a=(V_o/V_s)/(1+(V_o/V_s));
T_on=120;
T=T_on/a;
T_off=T-T_on;    printf("pulse width o/p voltage=%.0
    f us",T_off);

T_off=3*T_off;
T_on=T-T_off;
a=T_on/(T_on+T_off);
V_o=V_s*(a/(1-a));    printf("\nnew o/p voltage=%.2f
     V",V_o);
```

Scilab code Exa 7.11 1 to find whether load current is cont 2 calculate value of avg op current 3 to find its max and min values 4 find rms values of various harmonics 5 compute avg value of supply current 6 ip power power absorbed by load counter emf and power loss in res

```
clear;
clc;
R=1;
L=.005;
T_a=L/R;
T=2000*10^-6;
E=24;
V_s=220;
T_on=600*10^-6;
a=T_on/T;
a1=(T_a/T)*log(1+(E/V_s)*((exp(T/T_a))-1));
if(a1<a)
    disp("load current in continuous");
else
    disp("load current in discont.");
```

```
end
I_o=(a*V_s-E)/R;     printf("avg o/p current=%.0f A",I_o);
I_mx=(V_s/R)*((1-exp(-T_on/T_a))/(1-exp(-T/T_a)))-E/R;     printf("\nmax value of steady current=%.2f A",I_mx);
I_mn=(V_s/R)*((exp(T_on/T_a)-1)/(exp(T/T_a)-1))-E/R;     printf("\nmin value of steady current=%.3f A",I_mn);

f=1/T;
w=2*%pi*f;
I1=(2*V_s/(sqrt(2)*%pi)*sind(180*a))/(sqrt(R^2+(w*L)^2));     printf("\nfirst harmonic current=%.4f A",I1);
I2=(2*V_s/(2*sqrt(2)*%pi)*sind(2*180*a))/(sqrt(R^2+(w*L*2)^2));     printf("\nsecond harmonic current=%.4f A",I2);
I3=(2*V_s/(3*sqrt(2)*%pi)*sind(3*180*a))/(sqrt(R^2+(w*L*3)^2));     printf("\nthird harmonic current=%.5f A",I3);

I_TAV=a*(V_s-E)/R-L*(I_mx-I_mn)/(R*T);     printf("\navg supply current=%.4f A",I_TAV);

P1=I_TAV*V_s;
printf("\ni/p power=%.2f W",P1);
P2=E*I_o;
printf("\npower absorbed by load emf=%.0f W",P2);
printf("\npower loss in resistor=%.2f W",P1-P2);
I_or=sqrt(I_o^2+I1^2+I2^2+I3^2);
printf("\nrms value of load current=%.3f A",I_or);
```

Scilab code Exa 7.12 1 to find whether load current is cont or not 2 to calculate avg output voltage and avg current 3 to compute max and min values of output current

```
clear;
clc;
R=1;
L=.001;
V_s=220;
E=72;
f=500;
T_on=800*10^-6;
T_a=L/R;
T=1/f;
m=E/V_s;
a=T_on/T;
a1=(T_a/T)*log(1+m*(exp(-T/T_a)-1));
if(a1>a)
    disp("load current is continuous");
else
    disp("load current is discontinuous");
end
t_x=T_on+L*log(1+((V_s-E)/272)*(1-exp(-T_on/T_a)));
//Value of t_x wrongly calculated in the book so ans
    of V_o and I_o varies
V_o=a*V_s+(1-t_x/T)*E;    printf("avg o/p voltage=%
   .2f V",V_o);
I_o=(V_o-E)/R;    printf("\navg o/p current=%.2f A",
   I_o);
printf("\nmin value of load current=%.0f A",0);
I_mx=(V_s-E)/R*(1-exp(-T_on/T_a));    printf("\nmax
   value of load current=%.1f A",I_mx);
```

Scilab code Exa 7.13 to find the chopping freq

```
clear;
clc;
```

```
a=.2;
V_s=500;
E=a*V_s;
L=0.06;
I=10;
T_on=(L*I)/(V_s-E);
f=a/T_on;     printf("chopping freq=%.2f Hz",f);
```

Scilab code Exa 7.14 to find the value of external inductance to be added in series

```
clear;
clc;
a=.5;
pu=.1;//pu ripple
//x=T/T_a
//y=exp(-a*x)
y=(1-pu)/(1+pu);
//after solving
x=log(1/y)/a;
f=1000;
T=1/f;
T_a=T/x;
R=2;
L=R*T_a;
Li=.002;
Le=L-Li;     printf("external inductance=%.3f mH",Le
   *1000);
```

Scilab code Exa 7.15 1 to calculate min and and max value of load current 2 max value of ripple current 3 avg and rms values of load current 4 rms value of chopper current

```
clear;
clc;
R=10;
L=.015;
T_a=L/R;
f=1250;
T=1/f;
a=.5;
T_on=a*T;
V_s=220;
I_mx=(V_s/R)*((1-exp(-T_on/T_a))/(1-exp(-T/T_a)));     printf("max value of load current=%.3f A",I_mx);
I_mn=(V_s/R)*((exp(T_on/T_a)-1)/(exp(T/T_a)-1));  printf("\nmin value of load current=%.2f A",I_mn);
dI=I_mx-I_mn;    printf("\nmax value of ripple current=%.3f A",dI);
V_o=a*V_s;
I_o=V_o/R;    printf("\navg value of load current=%.2f A",I_o);
I_or=sqrt(I_mx^2+dI^2/3+I_mx*dI);    printf("\nrms value of load current=%.3f A",I_or);
I_chr=sqrt(a)*I_or;    printf("\nrms value of chopper current=%.3f A",I_chr);
//Answers have small variations from that in the book due to difference in the rounding off of digits.
```

Scilab code Exa 7.17 to find thetime for which current flows

```
clear;
clc;
L=0.0016;
C=4*10^-6;
```

```
w=1/sqrt(L*C);
t=%pi/w;    printf("time for which current flows=%.2
   f us",t*10^6);
```

Scilab code Exa 7.18 to calculate values of commutating capacitor and commutating inductor

```
clear;
clc;
t_q=20*10^-6;
dt=20*10^-6;
t_c=t_q+dt;
I_0=60;
V_s=60;
C=t_c*I_0/V_s;    printf("value of commutating
   capacitor=%.0f uF",C*10^6);
L1=(V_s/I_0)^2*C;
L2=(2*t_c/%pi)^2/C;
if(L1>L2)
    printf("\nvalue of commutating inductor=%.0f uH"
       ,L1*10^6);
else
    printf("\nvalue of commutating inductor=%.0f uH"
       ,L2*10^6);
end
```

Scilab code Exa 7.19 to calculate the value of comutating component C and comutating component L

```
clear;
clc;
t=100*10^-6;
R=10;
```

```
//V_s*(1-2*exp(-t/(R*C)))=0
C=-t/(R*log(1/2));    printf("Value of comutating
    component C=%.3f uF",C*10^6);
disp("max permissible current through SCR is 2.5
    times load current");
L=(4/9)*C*R^2;    printf("value of comutating
    component L=%.1f uH",L*10^6);
disp("max permissible current through SCR is 1.5
    times peak diode current");
L=(1/4)*C*R^2;    printf("value of comutating
    component L=%.2f uH",L*10^6);
```

Scilab code Exa 7.20 to compute 1 effective on period 2 peak current through main and auxillery thyristor 3 their turn off times 4 total commutation intervals 5 capacitor voltage 6 time needed to recharge the capacitor

```
clear;
clc;
T_on=800*10^-6;
V_s=220;
I_o=80;
C=50*10^-6;
T=T_on+2*V_s*C/I_o;    printf("effective on period=%
    .0f us",T*10^6);

L=20*10^-6;
C=50*10^-6;
i_T1=I_o+V_s*sqrt(C/L);    printf("\npeak current
    through main thyristor=%.2f A",i_T1);
i_TA=I_o;    printf("\npeak current through
    auxillery thyristor=%.0f A",i_TA);

t_c=C*V_s/I_o;    printf("\nturn off time for main
    thyristor=%.1f us",t_c*10^6);
t_c1=(%pi/2)*sqrt(L*C);    printf("\nturn off time
```

```
for auxillery thyristor=%.3f us",t_c1*10^6);

printf("\ntotal commutation interval=%.0f us",2*t_c*10^6);

t=150*10^-6;
v_c=I_o*t/C-V_s;    printf("\ncapacitor voltage=%.0f V",v_c);

printf("\ntime nedded to recharge the capacitor=%.0f us",2*V_s*C/I_o*10^6);
```

Scilab code Exa 7.21 to calculate values of 1 commutating components C and L and 2 min and max output voltages

```
clear;
clc;
I_o=260;
V_s=220;
fos=2;//factor of safety
t_off=18*10^-6;
t_c=2*t_off;
C=t_c*I_o/V_s;    printf("Value of C=%.3f uF",C*10^6);
L=(V_s/(.8*I_o))^2*C;    printf("\nvalue of L=%.3f uH",L*10^6);

f=400;
a_mn=%pi*f*sqrt(L*C);
V_omn=V_s*(a_mn+2*f*t_c);    printf("\nmin value of o/p voltage=%.3f V",V_omn);
V_omx=V_s;    printf("\nmax value of o/p voltage=%.0f V",V_omx);
```

Scilab code Exa 7.22 calculate 1 value of commutating inductor and capacitor 2 max capacitor voltage 3 peak commutating current

```
clear;
clc;
x=2;
t_q=30*10^-6;
dt=30*10^-6;
t_c=t_q+dt;
V_s=230;
I_o=200;
L=V_s*t_c/(x*I_o*(%pi-2*asin(1/x)));    printf("value of commutating inductor=%.3f uH",L*10^6);
C=x*I_o*t_c/(V_s*(%pi-2*asin(1/x)));    printf("\nvalue of commutating capacitor=%.3f uF",C*10^6);
V_cp=V_s+I_o*sqrt(L/C);    printf("\npeak capacitor voltage=%.0f V",V_cp);
I_cp=x*I_o;    printf("\npeak commutataing current=%.0f A",I_cp);

x=3;
L=V_s*t_c/(x*I_o*(%pi-2*asin(1/x)));    printf("\nvalue of commutating inductor=%.3f uH",L*10^6);
C=x*I_o*t_c/(V_s*(%pi-2*asin(1/x)));    printf("\nvalue of commutating capacitor=%.3f uF",C*10^6);
V_cp=V_s+I_o*sqrt(L/C);    printf("\npeak capacitor voltage=%.2f V",V_cp);
I_cp=x*I_o;    printf("\npeak commutataing current=%.0f A",I_cp);
```

Scilab code Exa 7.23 to compute 1 turn off time of main thyristor 2 total commutation interval 3 turn off time of auxillery thyristor

```
clear;
clc;
V_s=230;
C=50*10^-6;
L=20*10^-6;
I_cp=V_s*sqrt(C/L);
I_o=200;
x=I_cp/I_o;
t_c=(%pi-2*asin(1/x))*sqrt(C*L);     printf("turn off time of main thyristor=%.2f us",t_c*10^6);
th1=asind(1/x);
t=(5*%pi/2-th1*%pi/180)*sqrt(L*C)+C*V_s*(1-cosd(th1))/I_o;     printf("\ntotal commutation interval=%.3f us",t*10^6);
t=(%pi-th1*%pi/180)*sqrt(L*C);     printf("\nturn off time of auxillery thyristor=%.3f us",t*10^6);
```

Scilab code Exa 7.24 to calculate min and max value of load current

```
clear;
clc;
tc=.006;
R=10;
L=R*tc;
f=2000;
T=1/f;
V_o=50;
V_s=100;
a=V_o/V_s;
T_on=a*T;
T_off=T-T_on;
dI=V_o*T_off/L;
I_o=V_o/R;
I2=I_o+dI/2;     printf("max value of load current=%.3f A",I2);
```

```
I1=I_o-dI/2;     printf("\nmin value of load current=%.3f A",I1);
```

Scilab code Exa 7.27 calculate 1 range of speed control 2 range of duty cycle

```
clear;
clc;
I_a=30;
r_a=.5;
V_s=220;
a=I_a*r_a/V_s;     printf("min value of duty cycle=%.3f",a);
printf("\nmin Value of speed control=%.0f rpm",0);
a=1;
printf("\nmax value of duty cycle=%.0f",a);
k=.1;//V/rpm
N=(a*V_s-I_a*r_a)/k;     printf("\nmax value of speed control=%.0f rpm",N);
```

Scilab code Exa 7.28 determine chopping freq and duty cycle

```
clear;
clc;
V_t=72;
I_a=200;
r_a=0.045;
N=2500;
k=(V_t-I_a*r_a)/N;
E_a=k*1000;
L=.007;
Rm=.045;
Rb=0.065;
```

```
R=Rm+Rb;
T_a=L/R;
I_mx=230;
I_mn=180;
T_on=-T_a*log(-((V_t-E_a)/R-I_mx)/((I_mn)-(V_t-E_a)/R));
R=Rm;
T_a=L/R;
T_off=-T_a*log(-((-E_a)/R-I_mn)/((I_mx)-(-E_a)/R));
T=T_on+T_off;
f=1/T;    printf("chopping freq=%.1f Hz",f);
a=T_on/T;    printf("\nduty cycle ratio=%.3f",a);
```

Scilab code Exa 7.29 to determine the higher limit of current pulsation chopping freq and duty cycle ratio

```
clear;
clc;
I_mx=425;
I_lt=180;    //lower limit of current pulsation
I_mn=I_mx-I_lt;
T_on=.014;
T_off=.011;
T=T_on+T_off;
T_a=.0635;
a=T_on/T;
V=(I_mx-I_mn*exp(-T_on/T_a))/(1-exp(-T_on/T_a));
a=.5;
I_mn=(I_mx-V*(1-exp(-T_on/T_a)))/(exp(-T_on/T_a));
T=I_mx-I_mn;    printf("higher limit of current pulsation=%.0f A",T);
T=T_on/a;
f=1/T;    printf("\nchopping freq=%.3f Hz",f);
printf("\nduty cycle ratio=%.2f",a);
//Answers have small variations from that in the
```

book due to difference in the rounding off of digits.

Chapter 8

Inverters

Scilab code Exa 8.3 to find the value of C for having load commutation

```
clear;
clc;
T=.1*10^-3;
f=1/T;
k=15*10^-6;//k=th/w;
th=2*%pi*f*k;
X_l=10;
R=2;
X_c=R*tan(th)+X_l;
C=1/(2*%pi*f*X_c);     printf("value of C=%.3f uF",C
    *10^6);
```

Scilab code Exa 8.4 to find the power delivered

```
clear;
clc;
V_s=230;
V_01=2*V_s/(sqrt(2)*%pi);
```

```
R=2;
I_01=V_01/R;
P_d=I_01^2*R;     printf("power delivered to load=%.1
    f W",P_d);
V=V_s/2;
I_s=sqrt(2)*I_01/%pi;
P_s=V*I_s;
printf("\npower delivered by both sources=%.1f W",2*
    P_s);
```

Scilab code Exa 8.5 to find the power delivered to the load

```
clear;
clc;
V_s=230;
V_01=4*V_s/(%pi*sqrt(2));
R=1;
X_L=6;
X_c=7;
I_01=V_01/sqrt(R^2+(X_L-X_c)^2);
P=I_01^2*R;     printf("power delivered to the source
    =%.3f kW",P/1000);
I_s=sqrt(2)*I_01*(2*cosd(45))/%pi;
P_s=V_s*I_s;     printf("\npower from the source=%.3f
     kW",P_s/1000);
```

Scilab code Exa 8.6 to find rms value of thyristor and diode currents

```
clear;
clc;
disp("when load R=2 ohm");
V_01=230;
R=2;
```

```
I_01=V_01/R;
I_m=I_01*sqrt(2);
I_T1=I_m/2;    printf("rms value of thyristor current=%.2f A",I_T1);
I_D1=0;    printf("\nrms value of diode current=%.0f A",I_D1);

disp("when load R=2ohm, X_L=8ohm and X_C=6ohm");
X_L=8;
X_C=6;
I_01=V_01/sqrt(R^2+(X_L-X_C)^2);
phi1=atand((X_L-X_C)/R);
I_T1=I_T1*sqrt(2)*.47675;    printf("rms value of thyristor current=%.3f A",I_T1);
I_D1=.1507025*I_m/sqrt(2);    printf("\nrms value of diode current=%.3f A",I_D1);
```

Scilab code Exa 8.7 to find 1 rms value of fundamental load current 2 power absorbed by the load and fundamental power 3 rms and peak currents of each thyristor 4 conduction of thyristors and diodes

```
clear;
clc;
//v_o=4*V_s/%pi*(sind(wt)+sind(3*wt)/3+sind(5*wt)/5)
V_s=230;
R=4;
f=50;
w=2*%pi*f;
L=0.035;
C=155*10^-6;
X_L=w*L;
X_C=1/(w*C);
Z1=sqrt(R^2+(X_L-X_C)^2);
phi1=-atand((X_L-X_C)/R);
Z3=sqrt(R^2+(X_L*3-X_C/3)^2);
```

```
phi3=atand((X_L*3-X_C/3)/R);
Z5=sqrt(R^2+(X_L*5-X_C/5)^2);
phi5=atand((X_L*5-X_C/5)/R);

I_m1=4*V_s/(Z1*%pi);
I_01=I_m1/sqrt(2);    printf("rms value of fundamental load current=%.2f A",I_01);
I_m3=4*V_s/(3*Z3*%pi);
I_m5=4*V_s/(5*Z5*%pi);
I_m=sqrt(I_m1^2+I_m3^2+I_m5^2);
I_0=I_m/sqrt(2);
P_0=(I_0)^2*R;    printf("\nload power=%.1f W",P_0);
P_01=(I_01)^2*R;    printf("\nfundamental load power=%.1f W",P_01);
printf("\nrms value of thyristor current=%.3f A",I_m/2);

t1=(180-phi1)*%pi/(180*w);    printf("\nconduction time for thyristor=%.3f ms",t1*1000);
t1=(phi1)*%pi/(180*w);    printf("\nconduction time for diodes=%.3f ms",t1*1000);
```

Scilab code Exa 8.8 to determine 1 fundamental rms output voltage 2 total output power and fundamenetal freq power 3 avg and peak currents of each thyristor 4 input pf 5 distortion factor and THD and 6 harmonic factor foe lowest hamonic

```
clear;
clc;
V_s=230;
V_01=2*V_s/(sqrt(2)*%pi);    printf("fundamental rms o/p voltage=%.3f V",V_01);
R=10;
I_01=V_01/R;
P=I_01^2*R;    printf("\nfundamental power to load=%
```

```
.1f W",P);
V_or=sqrt((V_s/2)^2);
P=V_or^2/R;     printf("\ntotal o/p power to load=%.1f W",P);

I_TP=V_s/(2*R);
printf("\navg SCR current=%.2f A",I_TP*180/360);

I_or=I_TP;
pf=I_01^2*R/(V_or*I_or);     printf("\ni/p pf=%.3f",pf);

DF=V_01/V_or;     printf("\ndistortion factor=%.1f",DF);

V_oh=sqrt(V_or^2-V_01^2);
THD=V_oh/V_01;     printf("\nTHD=%.3f",THD);

V_03=V_01/3;
HF=V_03/V_01;     printf("\nharmonic factor=%.4f",HF);
```

Scilab code Exa 8.9 to calculate 1 rms value of output voltage and fundamental component of output voltage 2 output power 3 fundamental freq output power 4 avg and peak currents of each transistor 5 peak reverse blocking voltage 6 harmonic factor for third harmonic 7 THD

```
clear;
clc;
V_s=60;
R=3;
V_or=sqrt(V_s^2*%pi/%pi);     printf("rms value of o/p voltage=%.0f V",V_or);
V_01=4*V_s/(sqrt(2)*%pi);     printf("\nfundamental component of rms voltage=%.2f V",V_01);
```

```
P_o=V_or^2/R;     printf("\no/p power=%.0f W",P_o);
P_01=V_01^2/R;     printf("\nfundamental freq o/p power=%.2f W",P_01);

I_s=V_s/R;     printf("\npeak current=%.0f A",I_s);
I_avg=I_s*%pi/(2*%pi);     printf("\navg current of each transistor=%.0f A",I_avg);

printf("\npeak reverse blocking voltage=%.0f V",V_s);

V_03=V_01/3;
HF=V_03/V_01;     printf("\nharmonic factor=%.4f",HF);

V_oh=sqrt(V_or^2-V_01^2);
THD=V_oh/V_01;     printf("\nTHD=%.4f",THD);
```

Scilab code Exa 8.10 1 to calculate THD of output voltage and its distortion factor 2 to calculate THD of output current and its distortion factor 3 load power and avg dc source current 4 conduction time of each transistor and diode 5 peak and rms current of each transistor

```
clear;
clc;
V_s=220;
R=6;
f=50;
w=2*%pi*f;
L=0.03;
C=180*10^-6;
X_L=w*L;
X_C=1/(w*C);

V_or=sqrt(V_s^2*%pi/%pi);
```

```
V_01=4*V_s/(sqrt(2)*%pi);
V_oh=sqrt(V_or^2-V_01^2);
THD=V_oh/V_01;    printf("THD of voltage=%.4f",THD);
DF=V_01/V_or;    printf("\nDF=%.1f",DF);

Z1=sqrt(R^2+(X_L-X_C)^2);
phi1=-atand((X_L-X_C)/R);
Z3=sqrt(R^2+(X_L*3-X_C/3)^2);
phi3=atand((X_L*3-X_C/3)/R);
Z5=sqrt(R^2+(X_L*5-X_C/5)^2);
phi5=atand((X_L*5-X_C/5)/R);
Z7=sqrt(R^2+(X_L*7-X_C/7)^2);
phi7=atand((X_L*7-X_C/7)/R);

I_01=19.403;
I_m1=4*V_s/(Z1*%pi);
I_m3=4*V_s/(3*Z3*%pi);
I_m5=4*V_s/(5*Z5*%pi);
I_m7=4*V_s/(7*Z7*%pi);
I_m=sqrt(I_m1^2+I_m3^2+I_m5^2+I_m7^2);
I_or=I_m/sqrt(2);
I_oh=sqrt((I_m^2-I_m1^2)/2);
THD=I_oh/I_01;    printf("\nTHD of current=%.4f",THD);
DF=I_01/I_or;    printf("\nDF=%.3f",DF);

P_o=I_or^2*R;    printf("\nload power=%.1f W",P_o);
I_avg=P_o/V_s;    printf("\navg value of load current=%.2f A",I_avg);

t1=(180-phi1)*%pi/(180*w);    printf("\nconduction time for thyristor=%.0f ms",t1*1000);
t1=1/(2*f)-t1;    printf("\nconduction time for diodes=%.0f ms",t1*1000);

I_p=I_m1;    printf("\npeak transistor current=%.2f A",I_p);
I_t1=.46135*I_p;    printf("\nrms transistor current
```

```
=%.2f A",I_t1);
```

Scilab code Exa 8.11 to determine 1 rms value of load current 2 rms value of thyristor current 3 load power

```
clear;
clc;
V_s=450;
R=10;
disp("for 180deg mode");
I_or=sqrt((V_s/(3*R))^2*2/3+(2*V_s/(3*R))^2*1/3);     printf("rms value of load current=%.3f A",I_or);
I_T1=sqrt((1/(2*%pi))*((V_s/(3*R))^2*2*%pi/3+(2*V_s/(3*R))^2*%pi/3));     printf("\nrms value of load current=%.0f A",I_T1);
P=3*I_or^2*R;     printf("\npower delivered to load=%.1f kW",P/1000);

disp("for 120deg mode");
I_or=sqrt((1/(%pi))*((V_s/(2*R))^2*2*%pi/3));     printf("rms value of load current=%.3f A",I_or);
I_T1=sqrt((1/(2*%pi))*((V_s/(2*R))^2*2*%pi/3));     printf("\nrms value of load current=%.2f A",I_T1);
P=3*I_or^2*R;     printf("\npower delivered to load=%.3f kW",P/1000);
```

Scilab code Exa 8.12 to determine power delivered to load for 1 square wave output 2 quasi square wave ooutput 3 symmitrically placed pulses

```
clear;
clc;
```

```
V_s=230;
R=10;
f=50;
w=2*%pi*f;
L=0.03;
X_L=w*L;

V_or=sqrt(V_s^2*%pi/%pi);
V_01=4*V_s/(sqrt(2)*%pi);

Z1=sqrt(R^2+(X_L)^2);
phi1=-atand((X_L)/R);
Z3=sqrt(R^2+(X_L*3)^2);
phi3=atand((X_L*3)/R);
Z5=sqrt(R^2+(X_L*5)^2);
phi5=atand((X_L*5)/R);
Z7=sqrt(R^2+(X_L*7)^2);
phi7=atand((X_L*7)/R);

disp("using square wave o/p");
I_m1=4*V_s/(sqrt(2)*Z1*%pi);
I_m3=4*V_s/(sqrt(2)*3*Z3*%pi);
I_m5=4*V_s/(sqrt(2)*5*Z5*%pi);
I_m7=4*V_s/(sqrt(2)*7*Z7*%pi);
I_m=sqrt(I_m1^2+I_m3^2+I_m5^2+I_m7^2);
P=I_m^2*R;    printf("power delivered=%.2f W",P);

disp("using quasi-square wave o/p");
I_01=I_m1*sind(45);
I_03=I_m3*sind(3*45);
I_05=I_m5*sind(5*45);
I_07=I_m7*sind(7*45);
I_0=(I_01^2+I_03^2+I_05^2+I_07^2);
P=I_0*R;    printf("power delivered=%.2f W",P);

disp("using two symmitrical spaced pulses");
g=(180-90)/3+45/2;
I_01=2*I_m1*sind(g)*sind(45/2);
```

```
I_03=2*I_m3*sind(g*3)*sind(3*45/2);
I_05=2*I_m5*sind(g*5)*sind(5*45/2);
I_07=2*I_m7*sind(g*7)*sind(7*45/2);
I_0=(I_01^2+I_03^2+I_05^2+I_07^2);
P=I_0*R;    printf("power delivered=%.2f W",P);
```

Scilab code Exa 8.14 to determine the value of source inductance and also find value of commutating capacitors

```
clear;
clc;
f=50;
T=1/f;
I=.5;
di=I/T;    //di=di/dt
V_s=220;
L=V_s/di;    printf("source inductance=%.1f H",L);

t=20*10^-6;
fos=2;    //factor of safety
t_c=t*fos;
R=10;
C=t_c/(R*log(2));    printf("\ncommutating capacitor
    =%.2f uF",C*10^6);
```

Scilab code Exa 8.15 to check whether the ckt will commutate by itself or not and the voltage across the capacitor and inductor at the commutation

```
clear;
clc;
R=10;
L=.01;
C=10*10^-6;
```

```
if(R^2<4*L/C)
    disp("ckt will commutate on its own");
else
    disp("ckt will not commutate on its own");
end

xie=R/(2*L);
w_o=1/sqrt(L*C);
w_r=sqrt(w_o^2-xie^2);
phi=atand(xie/w_r);
t=%pi/w_r;
V_s=1;
v_L=V_s*(w_o/w_r)*exp(-xie*t)*cosd(180+phi);   printf("voltage across inductor(*V_s)=%.5f V",v_L);
v_c=V_s*(1-(w_o/w_r)*exp(-xie*t)*cosd(180-phi));   printf("\nvoltage across capacitor(*V_s)=%.5f V",v_c);
di=V_s/L;    printf("\ndi/dt*V_s (for t=0)=%.0f A/s",di);
```

Scilab code Exa 8.16 to calculate output freq

```
clear;
clc;
L=.006;
C=1.2*10^-6;
R=100;
T=%pi/sqrt(1/(L*C)-(R/(2*L))^2);
T_off=.2*10^-3;
f=1/(2*(T+T_off));    printf("o/p freq=%.2f Hz",f);

disp("for R=40ohm");
R=40;
T=%pi/sqrt(1/(L*C)-(R/(2*L))^2);
```

```
T_off=.2*10^-3;
f=1/(2*(T+T_off));    printf("upper limit o/p freq=%
    .1f Hz",f);

disp("for R=140ohm");
R=140;
T=%pi/sqrt(1/(L*C)-(R/(2*L))^2);
T_off=.2*10^-3;
f=1/(2*(T+T_off));    printf("lower limit o/p freq=%
    .1f Hz",f);
```

Scilab code Exa 8.17 To determine 1 ckt turn off time 2 max possible operating freq

```
clear;
clc;
f=5000;
w=2*%pi*f;
R=3;
L=60*10^-6;
xie=R/(2*L);
C=7.5*10^-6;
w_o=1/sqrt(L*C);
w_r=sqrt(w_o^2-xie^2);
t_c=%pi*(1/w-1/w_r);    printf("ckt turn off time=%
    .2f us",t_c*10^6);

fos=1.5;
t_q=10*10^-6;
f_max=1/(2*%pi*(t_q*fos/%pi+1/w_r));    printf("\
    nmax possible operating freq=%.1f Hz",f_max);
//Answers have small variations from that in the
    book due to difference in the rounding off of
    digits.
```

Scilab code Exa 8.18 to calculate power delivered to the load and avg and rms values of thyristor current

```
clear;
clc;
f=5000;
w=2*%pi*f;
R=3;
L=60*10^-6;
xie=R/(2*L);
C=7.5*10^-6;
w_o=1/sqrt(L*C);
w_r=sqrt(w_o^2-xie^2);
t1=%pi/(2*w_r);
V_s=220;
V_co=80;
I_omx=(V_s+V_co)*exp(-xie*t1)/(w_r*L);
I_rms=I_omx/sqrt(2);
P=I_rms^2*R;     printf("load power=%.2f W",P);
printf("\nrms value of thyristor current=%.3f A",I_omx/2);
I_SA=P/V_s;
printf("\navg thyristor current=%.3f A",I_SA/2);
//error in the book. wrong values are placed in the I_omx formulae. so all answer varies
```

Scilab code Exa 8.19 to find the value of capacitor C for given conditions

```
clear;
clc;
t=20;
fos=2;     //factor of safety
```

```
t_c=t*fos;
n=1/3;
R=20;
C=n^2*t_c/(4*R*log(2));     printf("value of capacitor=%.5f uF",C);
//printing mistake in the answer in book.
```

Scilab code Exa 8.20 to calculate 1 rms values of phase and line voltages 2 rms value of fundamental component of phase and line voltages 3 THD for voltages 4 load power and avg source current 5 avg value of thyristor current

```
clear;
clc;
V_s=220;
V_p=sqrt(2)*V_s/3;     printf("rms value of phasor voltages=%.2f V",V_p);
V_L=sqrt(3)*V_p;     printf("\nrms value of line voltages=%.2f V",V_L);

V_p1=sqrt(2)*V_s/%pi;     printf("\nfundamental component of phase voltage=%.3f V",V_p1);
V_L1=sqrt(3)*V_p1;     printf("\nfundamental component of line voltages=%.3f V",V_L1);

V_oh=sqrt(V_L^2-V_L1^2);
THD=V_oh/V_L1;     printf("\nTHD=%.7f",THD);

V_a1=2*V_s/%pi;
V_a5=2*V_s/(5*%pi);
V_a7=2*V_s/(7*%pi);
V_a11=2*V_s/(11*%pi);
R=4;
L=0.02;
f=50;
w=2*%pi*f;
```

```
Z1=sqrt(R^2+(w*L)^2);
Z5=sqrt(R^2+(5*w*L)^2);
Z7=sqrt(R^2+(7*w*L)^2);
Z11=sqrt(R^2+(11*w*L)^2);
I_a1=V_a1/Z1;
I_a5=V_a5/Z5;
I_a7=V_a7/Z7;
I_a11=V_a11/Z11;
I_or=sqrt((I_a1^2+I_a5^2+I_a7^2+I_a11^2)/2);
P=3*I_or^2*R;     printf("\nload power=%.1f W",P);
I_s=P/V_s;     printf("\navg value of source current=%.3f A",I_s);
I_TA=I_s/3;     printf("\navg value of thyristor current=%.3f A",I_TA);
```

Chapter 9

AC Voltage Controllers

Scilab code Exa 9.1 to determine 1 rms value of output voltage 2 power delivered to the load and input pf 3 avg input current

```
clear;
clc;
V_s=230;
V_m=sqrt(2)*V_s;
a=45;
V_or=(V_m/2)*sqrt(1/%pi*((2*%pi-a*%pi/180)+sind(2*a)
   /2));    printf("rms value of o/p voltage=%.3f V"
   ,V_or);
R=20;
I_or=V_or/R;
P_o=I_or^2*R;    printf("\nload power=%.1f W",P_o);
I_s=I_or;
VA=V_s*I_s;
pf=P_o/VA;    printf("\ni/p pf=%.4f",pf);

V_o=sqrt(2)*V_s/(2*%pi)*(cosd(a)-1);
I_ON=V_o/R;    printf("\navg i/p current=%.4f A",
   I_ON);
```

Scilab code Exa 9.2 to calculate 1 rms value of output voltage 2 load power and input pf 3 avg and rms currents of thyristors

```
clear;
clc;
V_s=230;
V_m=sqrt(2)*V_s;
a=45;
V_or=(V_s)*sqrt(1/%pi*((%pi-a*%pi/180)+sind(2*a)/2))
    ;    printf("rms value of o/p voltage=%.3f V",
    V_or);
R=20;
I_or=V_or/R;
P_o=I_or^2*R;    printf("\nload power=%.2f W",P_o);
I_s=I_or;
VA=V_s*I_s;
pf=P_o/VA;    printf("\ni/p pf=%.4f",pf);

I_TA=sqrt(2)*V_s/(2*%pi*R)*(cosd(a)+1);    printf("\
    navg thyristor current=%.3f A",I_TA);
I_Tr=sqrt(2)*V_s/(2*R)*sqrt(1/%pi*((%pi-a*%pi/180)+
    sind(2*a)/2));    printf("\nrms value of
    thyristor current=%.4f A",I_Tr);
```

Scilab code Exa 9.3 to determine 1 rms output voltage 2 input pf 3 avg and rms thyristor currents

```
clear;
clc;
V_s=230;
n=6;//on cycles
m=4;//off cycles
```

```
k=n/(n+m);
V_or=V_s*sqrt(k);    printf("rms value of o/ voltage=%.3f V",V_or);
pf=sqrt(k);    printf("\ni/p pf=%.4f",pf);
R=15;
I_m=V_s*sqrt(2)/R;
I_TA=k*I_m/%pi;    printf("\navg thyristor current=%.4f A",I_TA);
I_TR=I_m*sqrt(k)/2;    printf("\nrms value of thyristor current=%.3f A",I_TR);
```

Scilab code Exa 9.4 to calculate 1 max values of avg and rms thyristor currents 2 min ckt turn off time

```
clear;
clc;
V_s=230;
V_m=sqrt(2)*V_s;
R=3;
I_TAM=2*V_m/(2*%pi*R);    printf("max value of avg thyristor current=%.3f A",I_TAM);
I_TRM=V_m/(2*R);    printf("\nmax value of avg thyristor current=%.3f A",I_TRM);

f=50;
w=2*%pi*f;
t_c=%pi/w;    printf("\nckt turn off time=%.0f ms",t_c*1000);
```

Scilab code Exa 9.5 to calculate 1 control range of firing angle 2 max value of rms load current 3 max power and pf 4 max value of avg and rms thyristor currents 5 max possible value of current rating

```
clear;
clc;
R=3;
X_L=4;
phi=atand(X_L/R);    printf("min firing angle=%.2f
   deg",phi);
printf("\nmax firing angle=%.0f deg",180);
V_s=230;
Z=sqrt(R^2+X_L^2);
I_or=V_s/Z;    printf("\nmax value of rms load
   current=%.0f A",I_or);
P=I_or^2*R;    printf("\nmax power=%.0f W",P);
I_s=I_or;
pf=P/(V_s*I_s);    printf("\ni/p pf=%.1f",pf);

I_TAM=sqrt(2)*V_s/(%pi*Z);    printf("\nmax value of
    avg thyristor current=%.3f A",I_TAM);
I_Tm=sqrt(2)*V_s/(2*Z);    printf("\nmax value of
   rms thyristor current=%.3f A",I_Tm);

f=50;
w=2*%pi*f;
di=sqrt(2)*V_s*w/Z;    printf("\ndi/dt=%.0f A/s",di)
   ;
```

Scilab code Exa 9.6 to calculate extinction angle and rms value of output voltage

```
clc
clear
V=230;
R=3;//ohm
X_L=5;//ohm
a=120;//firing angle delay
phi=atand(X_L/R);
```

```
b=0;
i=1;
while i>0;
    LHS=sind(b-a);
    RHS=sind(a-phi)*exp(-(R/X_L)*(b-a)*%pi/180);
    if abs(LHS-RHS)<=.01;
        B=b;
        i=2;
        break;
    end
    b=b+.1
end
printf("extinction angle=%.1f deg",B);//answer in the book is wrong as formulae for RHS is wrongly employed
V_or=sqrt(2)*V*sqrt((1/(2*%pi))*((B-a)*%pi/180+(sind(2*a)-sind(2*B))/2));
printf("\nrms value of output voltage=%.2f V",V_or); //answer do not match due to wrong B in book
```

Scilab code Exa 9.8 to calculate 1 rms value of output voltage 2 rms value of current for upper thyristors 3 rms value of current for lower thyristors 4 transformer VA rating 5 input pf

```
clear;
clc;
V_s=230;
V_m=sqrt(2)*V_s;
a=60;
R=20;
V_or=sqrt((V_m^2/(2*%pi))*(a*%pi/180-sind(2*a)/2)+(2*V_m^2/(%pi))*(%pi-a*%pi/180+sind(2*a)/2)); printf("rms value of o/p voltage=%.2f V",V_or);
I_T1r=(V_m/R)*sqrt(1/%pi*((%pi-a*%pi/180)+sind(2*a)
```

```
/2));     printf("\nrms value of current for upper thyristors=%.3f A",I_T1r);
I_T3r=(V_m/(2*R))*sqrt(1/%pi*((a*%pi/180)-sind(2*a)/2));     printf("\nrms value of current for lower thyristors=%.3f A",I_T3r);
I1=sqrt(2)*I_T1r;
I3=sqrt((sqrt(2)*I_T1r)^2+(sqrt(2)*I_T3r)^2);
r=V_s*(I1+I3);     printf("\nt/f VA rating=%.2f VA",r);
P_o=V_or^2/R;
pf=P_o/r;     printf("\ni/p pf=%.4f",pf);
```

Chapter 10

Cycloconverters

Scilab code Exa 10.2 to calculate 1 rms value of output voltage 2 rms current of each converter 3 rms current of each thyristor and 4 input pf

```
clear;
clc;
V_s=230;
V_m=sqrt(2)*V_s;
R=10;
a=30;
V_or=(V_m/sqrt(2))*sqrt((1/%pi)*(%pi-a*%pi/180+sind(2*a)/2));
I_or=V_or/R;    printf("rms value of o/p current=%.2f A",I_or);
printf("\nrms value of o/p current for each convertor=%.2f A",I_or/sqrt(2));
printf("\nrms value of o/p current for each thyristor=%.3f A",I_or/2);
I_s=I_or;
pf=(I_or^2*R)/(V_s*I_s);    printf("\ni/p pf=%.4f",pf);
```

Scilab code Exa 10.4 to compute 1 value of fundamental rms output voltage 2 rms output current 3 output power

```
clear;
clc;
V_s=400;
V_ph=V_s/2;
a=160;
r=cosd(180-a);
m=3;
V_or=r*(V_ph*(m/%pi)*sin(%pi/m));     printf("rms o/p
     voltage=%.3f V",V_or);
R=2;
X_L=1.5;
th=atand(X_L/R);
Z=sqrt(R^2+X_L^2);
I_or=V_or/Z;     printf("\nrms o/p current=%.2f A",
    I_or);
printf("\nphase angle of o/p current=%.2f deg",-th)
P=I_or^2*R;     printf("\no/p power=%.2f W",P);
```

Scilab code Exa 10.5 to compute 1 value of fundamental rms output voltage 2 rms output current 3 output power Here 6 pulse bridge convertor is employed

```
clear;
clc;
V_s=400;
V_ph=V_s/2;
V_l=V_ph*sqrt(3);
a=160;
r=cosd(180-a);
m=6;
V_or=r*(V_l*(m/%pi)*sin(%pi/m));     printf("rms o/p
    voltage=%.2f V",V_or);
```

```
R=2;
X_L=1.5;
th=atand(X_L/R);
Z=sqrt(R^2+X_L^2);
I_or=V_or/Z;     printf("\nrms o/p current=%.2f A",
   I_or);
printf("\nphase angle of o/p current=%.2f deg",-th)
P=I_or^2*R;     printf("\no/p power=%.2f W",P);
```

Scilab code Exa 10.7 to calculata rms value of load voltage for various firing angle delays

```
clear;
clc;
V_l=400;
V_ml=sqrt(2)*V_l;
m=6;
f=50;
w=2*%pi*f;
L=.0012;
I=40;
disp("for firing angle=0deg");
a=0;
V_or=(V_ml*(m/%pi)*sin(%pi/m))*cosd(a);
V_omx=V_or-3*w*L*I/%pi;
V_rms=V_omx/sqrt(2);     printf("rms value of load
   voltage=%.2f V",V_rms);

disp("for firing angle=30deg");
a=30;
V_or=(V_ml*(m/%pi)*sin(%pi/m))*cosd(a);
V_omx=V_or-3*w*L*I/%pi;
V_rms=V_omx/sqrt(2);     printf("rms value of load
   voltage=%.2f V",V_rms);
```

Chapter 11

Some Applications

Scilab code Exa 11.1 to calculate 1 firing angle of the rectifier 2 output voltage 3 dc link voltage

```
clear;
clc;
V_s=11000;
V_ml=sqrt(2)*V_s;
f=50;
w=2*%pi*f;
I_d=300;
R_d=1;
g=20;//g=gamma
a=acosd(cosd(g)+%pi/(3*V_ml)*I_d*R_d);    printf("
   firing angle=%.3f deg",a);
L_s=.01;
V_d=(3/%pi)*((V_ml*cosd(a))-w*L_s*I_d);    printf("\
   nrectifier o/p voltage=%.1f V",V_d);
printf("\ndc link voltage=%.3f V",2*V_d/1000);
```

Scilab code Exa 11.2 To calculate rms current and peak reverse voltage ratings for each of thyristor valves

```
clear;
clc;
V_d=(200+200)*10^3;
P=1000*10^6;
I_d=P/V_d;
//each thristor conducts for 120deg for a periodicity of 360deg
printf("rms current rating of thyristor=%.2f A",I_d*sqrt(120/360));
a=0;
V_d=200*10^3;
V_ml=V_d*%pi/(3*cosd(a));
printf("\npeak reverse voltage across each thyristor=%.2f kV",V_ml/2/1000);
```

Scilab code Exa 11.3 to determine voltage and current rating of 1 thyristor and 2 diodes of the bridge

```
clear;
clc;
V_m=230;
V_s=230/sqrt(2);
pf=.8;
P=2000;
I_m=P/(V_s*pf);
I_Tr=I_m/sqrt(2);
I_TA=2*I_m/%pi;
fos=2;//factor of safety
printf("rms value of thyristor current=%.2f A",fos*I_Tr);
printf("\navg value of thyristor current=%.3f A",fos*I_TA);
PIV=V_m*sqrt(2);
printf("\nvoltage rating of thyristor=%.2f V",PIV);

```

```
I_Tr=I_m/(2);
I_TA=I_m/%pi;
printf("\nrms value of diode current=%.3f A",fos*I_Tr);
printf("\navg value of diode current=%.3f A",fos*I_TA);
printf("\nvoltage rating of diode=%.2f V",PIV);
```

Scilab code Exa 11.4 to find the value of parameters of R2 C and load resistance

```
clear;
clc;
V=200;
I=10;
R_L=V/I;    printf("value of load resistance=%.0f ohm",R_L);
I_h=.005;    //holding current
R2=V/I_h;    printf("\nvalue of R2=%.0f kilo-ohm",R2/1000);
t_c=20*10^-6;
fos=2;//factor of safety
C=t_c*fos/(R_L*log(2));    printf("\nvalue of C=%.3f uF",C*10^6);
```

Scilab code Exa 11.5 to calculate 1 depth of heat of penetration 2 heat generated per unit cylinder surface area and 3 heat generated per unit cylinder volume

```
clear;
clc;
u_r=10;
f=10000;//Hz
```

```
p=4*10^-8;//ohm-m
dl=(1/(2*%pi))*sqrt(p*10^7/(u_r*f));    printf("depth of heat of penetration=%.5f mm",dl*1000);
l=.12;//length of cylinder
t=20;//no of turns
I=100;
H=t*I/l;
P_s=2*%pi*H^2*sqrt(u_r*f*p*10^-7);    printf("\nheat generated per unit cylinder surface area=%.3f W/m^2",P_s);
d=.02;//diameter
P_v=4*H^2*p/(d*dl);    printf("\nheat generated per unit cylinder volume=%.0f W/m^3",P_v);
//answer of P_v varies as given in book as value of d is not taken as in formulae.
```

Scilab code Exa 11.6 to calculate reqd capacitor size

```
clear;
clc;
f=3000;
t_qmin=30*10^-6;
f_r=f/(1-2*t_qmin*f);
R=.06;
L=20*10^-6;
C=1/(L*((2*%pi*f_r)^2+(R/(2*L))^2));    printf("required capacitor size=%.4f F",C*10^6);
//Answers have small variations from that in the book due to difference in the rounding off of digits.
```

Chapter 12

Electic Drives

Scilab code Exa 12.1 To determine 1 firing angle delay of armature converter 2 rms value of thyristor and freewheeling diode currents 3 input pf of armature converter

```
clear;
clc;
T_e=15;//Nm
K_m=.5;//V-s/rad
I_a=T_e/K_m;
n_m=1000;
w_m=2*%pi*n_m/60;
E_a=K_m*w_m;
r_a=.7;
V_t=E_a+I_a*r_a;
V_s=230;
V_m=sqrt(2)*V_s;
a=acosd(2*%pi*V_t/V_m-1);    printf("firing angle
   delay=%.3f deg",a);
I_Tr=I_a*sqrt((180-a)/360);    printf("\nrms value
   of thyristor current=%.3f A",I_Tr);
I_fdr=I_a*sqrt((180+a)/360);    printf("\nrms value
   of freewheeling diode current=%.3f A",I_fdr);
pf=V_t*I_a/(V_s*I_Tr);    printf("\ninput power
```

```
    factor=%.4f",pf);
```

Scilab code Exa 12.2 calculate the avg armature current and the motor torque

```
clear;
clc;
V_m=330;
E_a=80;
r_a=4;
a=30;
I_a=((V_m/%pi*(1+cosd(a)))-E_a)/r_a;    printf("avg
    armature current=%.3f A",I_a);
n_s=1400;
w_m=2*%pi*n_s/60;
K_m=E_a/w_m;
T_e=K_m*I_a;    printf("\nmotor torque=%.3f Nm",T_e)
    ;
```

Scilab code Exa 12.3 to determine 1 motor current 2 motor torque 3 input pf

```
clear;
clc;
V_s=250;
V_m=sqrt(2)*V_s;
a=30;
k=0.03;//Nm/A^2
n_m=1000;
w_m=2*%pi*n_m/60;
r=.2;//r_a+r_s
V_t=V_m/%pi*(1+cosd(a));
```

```
I_a=V_t/(k*w_m+r);    printf("motor armature current =%.2f A",I_a);
T_e=k*I_a^2;    printf("\nmotor torque=%.3f Nm",T_e);
I_sr=I_a*sqrt((180-a)/180);
pf=(V_t*I_a)/(V_s*I_sr);    printf("\ninput power factor=%.4f",pf);
```

Scilab code Exa 12.4 to determine 1 rated armature current 2 firing angle delay of armature convertor 3 speed regulation at full load 4 input pf of armature convertor and drive

```
clear;
clc;
V_s=400;
V_m=sqrt(2)*V_s;
V_f=2*V_m/%pi;
r_f=200;
I_f=V_f/r_f;
T_e=85;
K_a=.8;
I_a=T_e/(I_f*K_a);    printf("rated armature current =%.2f A",I_a);
n_m=1200;
w_m=2*%pi*n_m/60;
r_a=.2;
V_t=K_a*I_f*w_m+I_a*r_a;
a=acosd(V_t*%pi/(2*V_m));    printf("\nfiring angle delay=%.2f deg",a);
E_a=V_t;
w_mo=E_a/(K_a*I_f);
N=60*w_mo/(2*%pi);
reg=((N-n_m)/n_m)*100;    printf("\nspeed regulation at full load=%.2f",reg);
I_ar=I_a;
```

```
pf=(V_t*I_a)/(V_s*I_ar);    printf("\ninput power factor of armature convertor=%.4f",pf);
I_fr=I_f;
I_sr=sqrt(I_fr^2+I_ar^2);
VA=I_sr*V_s;
P=V_t*I_a+V_f*I_f;
printf("\ninput power factor of drive=%.4f",P/VA);
//Answers have small variations from that in the book due to difference in the rounding off of digits.
```

Scilab code Exa 12.5 to calculate 1 delay angle of field converter 2 delay angle of armature converter 3 power fed back to the supply

```
clear;
clc;
V_s=400;
V_m=sqrt(2)*V_s;
V_f=2*V_m/%pi;
a1=acosd(-V_f*%pi/(2*V_m));    printf("delay angle of field converter=%.0f deg",a1);
r_f=200;
I_f=V_f/r_f;
T_e=85;
K_a=.8;
I_a=T_e/(I_f*K_a);
n_m=1200;
w_m=2*%pi*n_m/60;
r_a=.1;
I_a=50;
V_t=-K_a*I_f*w_m+I_a*r_a;
a=acosd(V_t*%pi/(2*V_m));    printf("\nfiring angle delay of armature converter=%.3f deg",a);
printf("\npower fed back to ac supply=%.0f W",-V_t*I_a);
```

Scilab code Exa 12.6 To compute 1 motor speed 2 torque developed

```
clear;
clc;
V_t=220;
n_m=1500;
w_m=2*%pi*n_m/60;
I_a=10;
r_a=1;
K_m=(V_t-I_a*r_a)/(w_m);
T=5;
I_a=T/K_m;
V_s=230;
V_m=sqrt(2)*V_s;
a=30;
V_t=2*V_m*cosd(a)/%pi;
w_m=(V_t-I_a*r_a)/K_m;
N=w_m*60/(2*%pi);    printf("motor speed=%.2f rpm",N
    );
a=45;
n_m=1000;
w_m=2*%pi*n_m/60;
V_t=2*V_m*cosd(a)/%pi;
I_a=(V_t-K_m*w_m)/r_a;
T_e=K_m*I_a;    printf("\ntorque developed=%.3f Nm",
    T_e);
//Answers have small variations from that in the
    book due to difference in the rounding off of
    digits.
```

Scilab code Exa 12.7 to compute firing angle and motor speed

```
clear;
clc;
V_t=220;
n_m=1000;
w_m=2*%pi*n_m/60;
I_a=60;
r_a=.1;
K_m=(V_t-I_a*r_a)/(w_m);
V_s=230;
V_m=sqrt(2)*V_s;
disp("for 600rpm speed");
n_m=600;
w_m=2*%pi*n_m/60;
a=acosd((K_m*w_m+I_a*r_a)*%pi/(2*V_m));    printf("
    firing angle=%.3f deg",a);

disp("for -500rpm speed");
n_m=-500;
w_m=2*%pi*n_m/60;
a=acosd((K_m*w_m+I_a*r_a)*%pi/(2*V_m));    printf("
    firing angle=%.3f deg",a);

I_a=I_a/2;
a=150;
V_t=2*V_m*cosd(a)/%pi;
w_m=(V_t-I_a*r_a)/K_m;
N=w_m*60/(2*%pi);    printf("\nmotor speed=%.3f rpm"
    ,N);
```

Scilab code Exa 12.8 to calculate the speed of the motor

```
clear;
clc;
K_m=1.5;
T_e=50;
```

```
I_a=T_e/K_m;
r_a=0.9;
a=45;
V_s=415;
V_ml=sqrt(2)*V_s;
w_m=((3*V_ml*(1+cosd(a))/(2*%pi))-I_a*r_a)/K_m;
N=w_m*60/(2*%pi);     printf("motor speed=%.2f rpm",N
    );
```

Scilab code Exa 12.9 to compute rms value of source and thyristor currents avg value of thyristor currents and input supply pf

```
clear;
clc;
V_t=600;
n_m=1500;
w_m=2*%pi*n_m/60;
I_a=80;
r_a=1;
K_m=(V_t-I_a*r_a)/(w_m);
V_s=400;
V_m=sqrt(2)*V_s;
disp("for firing angle=45deg and speed=1200rpm");
a=45;
n_m=1200;
w_m=2*%pi*n_m/60;
I_a=(3*V_m*(1+cosd(a))/(2*%pi)-K_m*w_m)/r_a;
I_sr=I_a*sqrt(2/3);     printf("rms value of source
    current=%.3f A",I_sr);
printf("\nrms value of thyristor current=%.3f A",I_a
    *sqrt(1/3));
printf("\navg value of thyristor current=%.2f A",I_a
    *(1/3));
pf=(3/(2*%pi)*(1+cosd(a)));     printf("\ninput power
     factor=%.3f",pf);
```

```

disp("for firing angle=90deg and speed=700rpm");
a=90;
n_m=700;
w_m=2*%pi*n_m/60;
I_a=(3*V_m*(1+cosd(a))/(2*%pi)-K_m*w_m)/r_a;
I_sr=I_a*sqrt(90/180);    printf("rms value of
    source current=%.3f A",I_sr);
printf("\nrms value of thyristor current=%.3f A",I_a
    *sqrt(90/360));
printf("\navg value of thyristor current=%.3f A",I_a
    *(1/3));
pf=(sqrt(6)/(2*%pi)*(1+cosd(a)))*sqrt(180/(180-a));
      printf("\ninput power factor=%.4f",pf);
```

Scilab code Exa 12.10 To find 1 no load speed 2 firing angle and supply pf 3 speed regulation

```
clear;
clc;
V_s=400;
V_m=sqrt(2)*V_s;
a=30;
V_t=3*V_m*cosd(a)/%pi;
I_a=21;
r_a=.1;
V_d=2;
K_m=1.6;
w_m=(V_t-I_a*r_a-V_d)/K_m;
N=w_m*60/(2*%pi);    printf("speed of motor=%.1f rpm
    ",N);

N=2000;
w_m=2*%pi*N/60;
I_a=210;
```

```
V_t=K_m*w_m+I_a*r_a+V_d;
a=acosd(V_t*%pi/(3*V_m));     printf("\nfiring angle=%.2f deg",a);
I_sr=I_a*sqrt(2/3);
pf=V_t*I_a/(sqrt(3)*V_s*I_sr);     printf("\nsupply power factor=%.3f",pf);

I_a=21;
w_m=(V_t-I_a*r_a-V_d)/K_m;
n=w_m*60/(2*%pi);
reg=(n-N)/N*100;     printf("\nspeed regulation(percent)=%.2f",reg);
```

Scilab code Exa 12.11 to calculate 1 transformer phase turn ratio 2 firing angle delay

```
clear;
clc;
V_t=230;
V_l=V_t*%pi/(3*sqrt(2));
V_ph=V_l/sqrt(3);
V_in=400;     //per phase voltage input
printf("transformer phase turns ratio=%.3f",V_in/V_ph);

N=1500;
I_a=20;
r_a=.6;
disp("for motor running at 1000rpm at rated torque");
E_a1=V_t-I_a*r_a;
n=1000;
E_a=E_a1/1500*1000;
V_t=E_a+I_a*r_a;
a=acosd(V_t*%pi/(3*sqrt(2)*V_l));     printf("firing
```

```
angle delay=%.2f deg",a);

disp("for motor running at -900rpm at half of rated torque");
I_a=.5*I_a;
n=-900;
V_t=n*E_a1/N+I_a*r_a;
a=acosd(V_t*%pi/(3*sqrt(2)*V_l));   printf("firing angle delay=%.3f deg",a);
```

Scilab code Exa 12.12 to calculate firing angle for different given conditions

```
clear;
clc;
V_s=400;
V_ml=sqrt(2)*V_s;
V_f=3*V_ml/%pi;
R_f=300;
I_f=V_f/R_f;
T_e=60;
k=1.1;
I_a=T_e/(k*I_f);
N=1000;
w_m=2*%pi*N/60;
r_a=.3;
V_t=k*I_f*w_m+I_a*r_a;
a=acosd(V_t*%pi/(3*V_ml));   printf("firing angle=%.3f deg",a);

N=3000;
w_m=2*%pi*N/60;
a=0;
V_t=3*V_ml*cosd(a)/%pi;
I_f=(V_t-I_a*r_a)/(w_m*k);
```

```
V_f=I_f*R_f;
a=acosd(V_f*%pi/(3*V_ml));    printf("\nfiring angle =%.3f deg",a);
```

Scilab code Exa 12.13 to evaluate the time taken for the speed to reach 1000rpm

```
clear;
clc;
//after calculating
//t=w_m/6000-%pi/360

N=1000;
w_m=2*%pi*N/60;
t=w_m/6000-%pi/360;    printf("time reqd=%.5f s",t);
//printing mistake in the answer in book
```

Scilab code Exa 12.14 to determine the 3rd and 5th harmonic components of line current

```
clear;
clc;
I_a=1;//supposition
a=60;
I_s1=2*sqrt(2)/%pi*I_a*sind(a);
I_s3=2*sqrt(2)/(3*%pi)*I_a*sind(3*a);
I_s5=2*sqrt(2)/(5*%pi)*I_a*sind(5*a);
per3=I_s3/I_s1*100;    printf("percent of 3rd harmonic current in fundamental=%.0f",per3);
per5=I_s5/I_s1*100;    printf("\npercent of 5th harmonic current in fundamental=%.0f",per5);
```

Scilab code Exa 12.15 to calculate 1 rms and avg value of thyristor current 2 pf of ac source 3 motor speed

```
clear;
clear;
clc;
I_a=60;
I_TA=I_a/3;     printf("avg thyristor current=%.0f A"
    ,I_TA);
I_Tr=I_a/sqrt(3);     printf("\nrms thyristor current
    =%.3f A",I_Tr);

V_s=400;
V_m=sqrt(2)*V_s;
I_sr=I_a*sqrt(2/3);
a=150;
V_t=3*V_m*cosd(a)/%pi;
pf=V_t*I_a/(sqrt(3)*V_s*I_sr);     printf("\npower
    factor of ac source=%.3f",pf);

r_a=0.5;
K_m=2.4;
w_m=(V_t-I_a*r_a)/K_m;
N=w_m*60/(2*%pi);     printf("\nspeed of motor=%.2f
    rpm",N);
//Answers have small variations from that in the
    book due to difference in the rounding off of
    digits.
```

Scilab code Exa 12.16 to determine 1 input power from source 2 input resistance of chopper drive 3 motor speed and 4 motor torque

```
clear;
clc;
I_a=300;
V_s=600;
a=.6;
V_t=a*V_s;
P=V_t*I_a;     printf("input power from source=%.0f
    kW",P/1000);
R_eq=V_s/(a*I_a);     printf("\nequivalent input
    resistance=%.3f ohm",R_eq);
k=.004;
R=.04+.06;
w_m=(a*V_s-I_a*R)/(k*I_a);
N=w_m*60/(2*%pi);     printf("\nmotor speed=%.1f rpm"
    ,N);
T_e=k*I_a^2;     printf("\nmotor torque=%.0f Nm",T_e)
    ;
```

Scilab code Exa 12.17 to calculate avg load current

```
clear;
clc;
T_on=10;
T_off=15;
a=T_on/(T_on+T_off);
V_s=230;
V_t=a*V_s;
r_a=3;
K_m=.5;
N=1500;
w_m=2*%pi*N/60;
I_a=(V_t-K_m*w_m)/r_a;     printf("motor load current
    =%.3f A",I_a);
```

Scilab code Exa 12.18 to determine 1 range of speed control 2 range of duty cycle

```
clear;
clc;
w_m=0;    printf("lower limit of speed control=%.0f rpm",w_m);
I_a=25;
r_a=.2;
V_s=220;
K_m=0.08;
a=(K_m*w_m+I_a*r_a)/V_s;    printf("\nlower limit of duty cycle=%.3f",a);

a=1;    printf("\nupper limit of duty cycle=%.0f",a);
w_m=(a*V_s-I_a*r_a)/K_m;    printf("\nupper limit of speed control=%.1f rpm",w_m);
```

Scilab code Exa 12.19 To calculate the min and max values of armature current and armaure current extrusion

```
clear;
clc;
clear
T_e=30;
K_m=1.5;
I_a=T_e/K_m;
N=1000;
w_m=2*%pi*N/60;
E_a=K_m*w_m;
r_a=0;
```

```
V_t=E_a+I_a*r_a;
V_s=220;
a=V_t/V_s;
f=400;
T=1/f;
T_on=a*T;
T_off=T-T_on;
L=0.02;
di=(V_s-E_a)/L;     //di=di_a/dt, during on period
dii=(-E_a)/L;     //di=di_a/dt, during off period
//I_mx=I_mn+di*T_on;
//I_a=(I_mx+I_mn)/2;
//after solving
I_mx=22.808;     printf("maximum armature current=%.3f A",I_mx);
I_mn=2*I_a-I_mx;     printf("\nminimum armature current=%.3f A",I_mn);
printf("\narmature current extrusion=%.3f A",I_mx-I_mn);
t=poly(0,'t');
i_a=addf('I_mn',mulf('t','di'));
printf("\narmature current expression during turn-on");
disp(eval(i_a));
i_a=addf('I_mx',mulf('t','dii'));
printf("\narmature current expression during turn-off");
disp(eval(i_a));
```

Scilab code Exa 12.21 to determine 1 power returned to the dc supply 2 equivalent load resistance of motor acting as generator 3 min and max breaking speeds 4 speed during regenerative braking

```
clear;
clc;
```

```
a=.6;
V_s=400;
V_t=(1-a)*V_s;
I_a=300;
P=V_t*I_a;    printf("power returned=%.0f kW",P
    /1000);

r_a=.2;
K_m=1.2;
R_eq=(1-a)*V_s/I_a+r_a;    printf("\nequivalent load
     resistance=%.4f ohm",R_eq);

w_mn=I_a*r_a/K_m;
N=w_mn*60/(2*%pi);    printf("\nmin braking speed=%
    .2f rpm",N);
w_mx=(V_s+I_a*r_a)/K_m;
N=w_mx*60/(2*%pi);    printf("\nmax braking speed=%
    .1f rpm",N);

w_m=(V_t+I_a*r_a)/K_m;
N=w_m*60/(2*%pi);    printf("\nmax braking speed=%.1
    f rpm",N);
```

Scilab code Exa 12.22 to determine max current in terms of rated currents at given speeds

```
clear;
clc;
N=1500;
disp("when speed=1455rpm");
n=1455;
s1=(N-n)/N;
r=sqrt(1/3)*(2/3)/(sqrt(s1)*(1-s1));    printf("
    I_2mx/I_2r=%.3f",r);

```

```
disp("when speed=1350rpm");
n=1350;
s1=(N-n)/N;
r=sqrt(1/3)*(2/3)/(sqrt(s1)*(1-s1));     printf("I_2mx/I_2r=%.3f",r);
```

Scilab code Exa 12.23 to calculate 1 motor speed at rated load 2 slip at which max torque occurs 3 max torque

```
clear;
clc;
Po=20000;
N=1440;
w_m=2*%pi*N/60;
T_e=Po/w_m;
f1=120;
P=4;
w_s=4*%pi*f1/P;
r2=.4;
x2=1.6;
f2=50;
Z1=r2+%i*x2*f1/f2;
Z=abs(Z1);
ph=3;
V_s=400;
s=(ph/w_s)*(V_s/(Z*sqrt(3)))^2*(r2/T_e);
N=w_s*f1/(4*%pi)*(1-s);     printf("motor speed at rated laod=%.2f rpm",N);
s_m=r2/imag(Z1);     printf("\nslip at which max torque occurs=%.4f",s_m);
T_em=(3/w_s)*(V_s/sqrt(3))^2/(2*imag(Z1));     printf("\nmax torque=%.3f Nm",T_em);
```

Scilab code Exa 12.24 To calculate 1 current and pf at the instant of starting and under max torque conditions 2 starting and max torques

```
clear;
clc;
V1=400;
r1=.6;
r2=.4;
s=1;
x1=1.6;
x2=1.6;
disp("at starting in normal conditions");
I_n=V1/sqrt((r1+r2/s)^2+(x1+x2)^2);    printf("current=%.2f A",I_n);
pf=(r1+r2)/sqrt((r1+r2/s)^2+(x1+x2)^2);    printf("\npf=%.4f",pf);
f1=50;
w_s=4*%pi*f1/4;
T_en=(3/w_s)*I_n^2*(r2/s);    printf("\nTorque developed=%.2f Nm",T_en);
disp("motor is operated with DOL starting");
I_d=V1/2/sqrt((r1+r2/s)^2+((x1+x2)/2)^2);    printf("current=%.0f A",I_d);
pf=(r1+r2)/sqrt((r1+r2/s)^2+((x1+x2)/2)^2);    printf("\npf=%.2f",pf);
f1=25;
w_s=4*%pi*f1/4;
T_ed=(3/w_s)*I_d^2*(r2/s);    printf("\nTorque developed=%.3f Nm",T_ed);

disp("at max torque conditions");
s_mn=r2/sqrt((r1)^2+((x1+x2))^2);
I_n=V1/sqrt((r1+r2/s_mn)^2+(x1+x2)^2);    printf("current=%.3f A",I_n);
pf=(r1+r2/s_mn)/sqrt((r1+r2/s_mn)^2+(x1+x2)^2);    printf("\npf=%.4f",pf);
f1=50;
w_s=4*%pi*f1/4;
```

```
T_en=(3/w_s)*I_n^2*(r2/s_mn);    printf("\nTorque developed=%.2f Nm",T_en);
disp("motor is operated with DOL starting");
s_mn=r2/sqrt((r1)^2+((x1+x2)/2)^2);
I_d=V1/2/sqrt((r1+r2/s_mn)^2+((x1+x2)/2)^2);  printf("current=%.3f A",I_d);
pf=(r1+r2/s_mn)/sqrt((r1+r2/s_mn)^2+((x1+x2)/2)^2);    printf("\npf=%.3f",pf);
f1=25;
w_s=4*%pi*f1/4;
T_en=(3/w_s)*I_d^2*(r2/s_mn);    printf("\nTorque developed=%.3f Nm",T_en);
```

Scilab code Exa 12.25 calculate 1 slip for max torque 2 starting and max torques 3 supply voltage reqd

```
clear;
clc;
x1=1;
X_m=50;
X_e=x1*X_m/(x1+X_m);
V=231;
V_e=V*X_m/(x1+X_m);
x2=1;
r2=.4;
r1=0;
s_m=r2/(x2+X_e);    printf("slip at max torque=%.2f",s_m);
s_mT=r2/(x2+X_m);    printf("\nslip at max torque=%.5f",s_mT);
f1=50;
w_s=4*%pi*f1/4;
disp("for constant voltage input");
T_est=(3/w_s)*(V_e/sqrt(r2^2+(x2+X_e)^2))^2*(r2);    printf("starting torque=%.3f Nm",T_est);
```

```
T_em=(3/w_s)*V_e^2/(2*(x2+X_e));    printf("\nmaximum torque developed=%.2f Nm",T_em);
disp("for constant current input");
I1=28;
T_est=(3/w_s)*(I1*X_m)^2/(r2^2+(x2+X_m)^2)*r2;    printf("starting torque=%.3f Nm",T_est);
T_em=(3/w_s)*(I1*X_m)^2/(2*(x2+X_m));    printf("\nmaximum torque developed=%.3f Nm",T_em);
s=s_mT;
I_m=I1*(r2/s+%i*x2)/(r2/s+%i*(x2+X_m));
I_m=abs(I_m);
V1=sqrt(3)*I_m*X_m;    printf("\nsupply voltage reqd=%.1f V",V1);
```

Scilab code Exa 12.27 to calculate 1 the value of chopper resistance R 2 inductor current 3 duty cycle 4 rectified output voltage 5 efficiency

```
clear;
clc;
V=420;
V1=V/sqrt(3);
T_e=450;
N=1440;
n=1000;
T_L=T_e*(n/N)^2;
n1=1500;
w_s=2*%pi*n1/60;
w_m=2*%pi*n/60;
a=.8;
I_d=T_L*w_s/(2.339*a*V1);
k=0;
R=(1-w_m/w_s)*(2.339*a*V1)/(I_d*(1-k));    printf("value of chopper resistance=%.4f ohm",R);

n=1320;
```

```
T_L=T_e*(n/N)^2;
I_d=T_L*w_s/(2.339*a*V1);    printf("\ninductor
   current=%.3f A",I_d);

w_m=2*%pi*n/60;
k=1-((1-w_m/w_s)*(2.339*a*V1)/(I_d*R));    printf("\
   nvalue of duty cycle=%.4f",k);

s=(n1-n)/n1;
V_d=2.339*s*a*V1;    printf("\nrectifed o/p voltage=
   %.3f V",V_d);

P=V_d*I_d;
I2=sqrt(2/3)*I_d;
r2=0.02;
Pr=3*I2^2*r2;
I1=a*I2;
r1=0.015;
Ps=3*I1^2*r1;
Po=T_L*w_m;
Pi=Po+Ps+Pr+P;
eff=Po/Pi*100;    printf("\nefficiency(in percent)=%
   .2f",eff);
```

Scilab code Exa 12.28 To calculate 1 value of chopper duty cycle 2 efficiency for power input 3 input power factor

```
clear;
clc;
V=400;
V_ph=V/sqrt(3);
N_s=1000;
N=800;
a=.7;
I_d=110;
```

```
R=2;
k=1-((1-N/N_s)*(2.339*a*V_ph)/(I_d*R));    printf("value of duty cycle=%.3f",k);
P=I_d^2*R*(1-k);
I1=a*I_d*sqrt(2/3);
r1=.1;
r2=.08;
Pr=3*I1^2*(r1+r2);
P_o=20000;
P_i=P_o+Pr+P;
eff=P_o/P_i*100;    printf("\nefficiency=%.2f",eff);

I11=sqrt(6)/%pi*a*I_d
th=43;
P_ip=sqrt(3)*V*I11*cosd(th);
pf=P_ip/(sqrt(3)*V*I11);    printf("\ninput power factor=%.4f",pf);
```

Scilab code Exa 12.29 to calculate 1 rotor rectified voltage 2 inductor angle 3 delay angle of invertor 4 efficiency 5 motor speed

```
clear;
clc;
V=420;
V1=V/sqrt(3);
N=1000;
w_m=2*%pi*N/60;
N_s=1500;
s=(N_s-N)/N_s;
a=.8;
V_d=2.339*a*s*V1;    printf("rectified voltage=%.2f V",V_d);
T=450;
N1=1200;
T_L=T*(N/N1)^2;
```

```
f1=50;
w_s=4*%pi*f1/4;
I_d=w_s*T_L/(2.339*a*V1);     printf("\ninductor current=%.2f A",I_d);
a_T=-.4;
a1=acosd(s*a/a_T);     printf("\ndelay angle of inverter=%.2f deg",a1);

P_s=V_d*I_d;
P_o=T_L*w_m;
R_d=0.01;
P_i=I_d^2*R_d;
I2=sqrt(2/3)*I_d;
r2=0.02;
r1=0.015;
P_rol=3*I2^2*r2;
I1=a*I2;
P_sol=3*I1^2*r1;
P_i=P_o+P_rol+P_sol+P_i;
eff=P_o/P_i*100;     printf("\nefficiency=%.2f",eff);
w_m=w_s*(1+(-a_T/a)*cosd(a1)-w_s*R_d*T_L/(2.339*a*V1)^2);
N=w_m*60/(2*%pi);     printf("\nmotor speed=%.1f rpm",N);
//Answers have small variations from that in the book due to difference in the rounding off of digits.
```

Scilab code Exa 12.30 to find firing angle advance of inveter

```
clear;
clc;
V=700;
E2=V/sqrt(3);
N_s=1500;
```

```
N=1200;
s=(N_s-N)/N_s;
V_dd=.7;
V_dt=1.5;
V_d=3*sqrt(6)*s*E2/%pi-2*V_dd;
V1=415;
a=acosd((3*sqrt(2)*E2/%pi)^-1*(-V_d+2*V_dt));
printf("firing angle advance=%.2f deg",180-a);
```

Scilab code Exa 12.31 to find firing angle advance of inveter

```
clear;
clc;
V=700;
E2=V/sqrt(3);
N_s=1500;
N=1200;
s=(N_s-N)/N_s;
V_dd=.7;
V_dt=1.5;
a=0;
u=18;//overlap angle in case of rectifier
V_d=3*sqrt(6)*s*E2*(cosd(a)+cosd(a+u))/(2*%pi)-2*V_dd;

V1=415;
V_ml=sqrt(2)*V1;
u=4;//overlap anglein the inverter
//V_dc=-(3*V_ml*(cosd(a)+cosd(a+u))/(2*%pi)-2*V_dt);
//V_dc=V_d;
//after solving, (1+cosd(u))*cosd(a)-sind(u)*sind(a)=-.6425
a=acosd(-.6425/(sqrt((1+cosd(u))^2+sind(u)^2))-atand(sind(u)/(1+cosd(u)));
printf("firing angle advance=%.2f deg",180-a);
```

Scilab code Exa 12.32 to find the voltage ratio of the transformer

```
clear;
clc;
V=700;
E2=V;
N_s=1500;
N=1200;
s=(N_s-N)/N_s;
V1=415;
a_T=s*E2/V1;    printf("voltage ratio of the
    transformer=%.4f",a_T);
```

Scilab code Exa 12.33 to calculate 1 supply voltage 2 armature current 3 excitation voltage 4 load angle 5 pull out torque

```
clear;
clc;
P=6;
N_s=600;
f1=P*N_s/120;
V=400;
f=50;
V_t=f1*V/f;    printf("supply freq=%.0f Hz",V_t);
T=340;
N=1000;
T_L=T*(N_s/N)^2;
w_s=2*%pi*N_s/60;
P=T_L*w_s;
I_a=P/(sqrt(3)*V_t);    printf("\narmature current=%
    .2f A",I_a);
```

```
Z_s=%i*2;
X_s=f1/f*abs(Z_s);
V_t=V_t/sqrt(3);
Ef=sqrt(V_t^2+(I_a*X_s)^2);
printf("\nexcitation voltage=%.2f V",sqrt(3)*Ef);
dl=atand(I_a*X_s/V_t);    printf("\nload angle=%.2f deg",dl);
T_em=(3/w_s)*(Ef*V_t/X_s);    printf("\npull out torque=%.2f Nm",T_em);
```

Scilab code Exa 12.34 to calculate 1 load angle 2 line current 3 input pf

```
clear;
clc;
P=4;
f=50;
w_s=4*%pi*f/P;
X_d=8;
X_q=2;
T_e=80;
V=400;
V_t=V/sqrt(3);
dl=(1/2)*asind(T_e*w_s/((3/2)*(V_t)^2*(1/X_q-1/X_d)));    printf("load angle=%.3f deg",dl);
I_d=V_t*cosd(dl)/X_d;
I_q=V_t*sind(dl)/X_q;
I_a=sqrt(I_d^2+I_q^2);    printf("\narmature current=%.2f A",I_a);
pf=T_e*w_s/(sqrt(3)*V*I_a);    printf("\ninput power factor=%.4f",pf);
```

Scilab code Exa 12.35 To calculate motor speed

```
clear;
clc;
T_e=3;
K_m=1.2;
I_a=T_e/K_m;
r_a=2;
V=230;
E_a=(.263*sqrt(2)*V-I_a*r_a)/(1-55/180);
w_m=E_a/K_m;
N=w_m*60/(2*%pi);     printf("motor speed=%.2f rpm",N
    );
```

Scilab code Exa 12.36 to calculate avg motor torque

```
clear;
clc;
K_m=1;
N=1360;
w_m=2*%pi*N/60;
E_a=K_m*w_m;
//after calculations V_t, calculated
V_t=163.45;
r_a=4;
I_a=(V_t-E_a)/r_a;
T_e=K_m*I_a;     printf("motor torque=%.4f Nm",T_e);
```

Scilab code Exa 12.37 to calculate avg motor torque

```
clear;
clc;
K_m=1;
N=2100;
w_m=2*%pi*N/60;
```

```
E_a=K_m*w_m;
//after calculations V_t, calculated
V_t=227.66;
r_a=4;
I_a=(V_t-E_a)/r_a;
T_e=K_m*I_a;     printf("motor torque=%.4f Nm",T_e);
```

Scilab code Exa 12.38 to calculate avg motor torque

```
clear;
clc;
K_m=1;
N=840;
w_m=2*%pi*N/60;
E_a=K_m*w_m;
V=230;
a=75;
V_t=sqrt(2)*V/%pi*(1+cosd(a));
r_a=4;
I_a=(V_t-E_a)/r_a;
T_e=K_m*I_a;     printf("motor torque=%.4f Nm",T_e);
//Answers have small variations from that in the
   book due to difference in the rounding off of
   digits.
```

Scilab code Exa 12.39 to calculate avg motor torque

```
clear;
clc;
K_m=1;
N=1400;
w_m=2*%pi*N/60;
E_a=K_m*w_m;
```

```
V=230;
a=60;
a1=212;
V_t=sqrt(2)*V/%pi*(cosd(a)-cosd(a1))+E_a*(180+a-a1)
    /180;
r_a=3;
I_a=(V_t-E_a)/r_a;
T_e=K_m*I_a;     printf("motor torque=%.3f Nm",T_e);
```

Scilab code Exa 12.40 to calculate avg motor torque

```
clear;
clc;
K_m=1;
N=600;
w_m=2*%pi*N/60;
E_a=K_m*w_m;
V=230;
a=60;
V_t=2*sqrt(2)*V/%pi*(cosd(a));
r_a=3;
I_a=(V_t-E_a)/r_a;
T_e=K_m*I_a;     printf("motor torque=%.3f Nm",T_e);
```

Scilab code Exa 12.41 to calculate the value of load current and source current

```
clear;
clc;
r1=.6;
r2=.4;
s=0.04;
x1=1.6;
```

```
x2=1.6;
Z=(r1+r2/s)+%i*(x1+x2);
V=400;
I1=V/Z;     printf("source current=%.3f A and with %.1f deg phase",atand(imag(I1)/real(I1)),abs(I1));
I2=V/Z;
N=1500;
w_s=2*%pi*N/60;
T_e=(3/w_s)*abs(I2)^2*r2/s;     printf("\nmotor torque=%.2f Nm",T_e);
N_r=N*(1-s);

f=45;
N_s1=120*f/4;
w_s=2*%pi*N_s1/60;
s1=(N_s1-N_r)/N_s1;
Z=(r1+r2/s1)+%i*(x1+x2)*f/50;
V=360;
I1=V/Z;     printf("\nsource current=%.3f A and with %.1f deg phase",atand(imag(I1)/real(I1)),abs(I1));
I2=V/Z;
T_e=(3/w_s)*abs(I2)^2*r2/s1;     printf("\nmotor torque=%.2f Nm",T_e);
```

Chapter 13

Power Factor Improvement

Scilab code Exa 13.1 To calculate load voltage voltage regulation system utilisation and energy consumed

```
clear;
clc;
V_s=250;
R_l=5;
I_l=20;
disp("for pf=1");
V_l=sqrt(V_s^2-(R_l*I_l)^2);    printf("load voltage=%.2f V",V_l);
reg=(V_s-V_l)/V_s*100;    printf("\nvoltage regulation=%.2f",reg);
pf=1;
P_l=V_l*I_l*pf;    //load power
P_r=V_s*I_l*pf;    //max powwible system rating
utf=P_l*100/P_r;    printf("\nsystem utilisation factor=%.3f",utf);
printf("\nenergy consumed(in units)=%.1f",P_l/1000);
disp("for pf=.5");
pf=.5;
//(.5*V_l)^2+(.866*V_l+R_l*I_l)^2=V_s^2
//after solving
```

```
V_l=158.35;    printf("load voltage=%.2f V",V_l);
reg=(V_s-V_l)/V_s*100;    printf("\nvoltage
  regulation=%.2f",reg);
P_l=V_l*I_l*pf;    //load power
P_r=V_s*I_l;    //max powwible system rating
utf=P_l*100/P_r;    printf("\nsystem utilisation
  factor=%.3f",utf);
printf("\nenergy consumed(in units)=%.2f",P_l/1000);
```

Scilab code Exa 13.2 to calculate the capacitance reqd

```
clear;
clc;
f=50;
V_s=230;
disp("at no load");
I_m=2;
pf=.3;
I_c=I_m*sind(acosd(pf));
C=I_c/(2*%pi*f*V_s);    printf("value of capacitance
  =%.3f uF",C*10^6);
disp("at half full load");
I_m=5;
pf=.5;
I_c=I_m*sind(acosd(pf));
C=I_c/(2*%pi*f*V_s);    printf("value of capacitance
  =%.3f uF",C*10^6);
disp("at full load");
I_m=10;
pf=.7;
I_c=I_m*sind(acosd(pf));
C=I_c/(2*%pi*f*V_s);    printf("value of capacitance
  =%.3f uF",C*10^6);
```

Scilab code Exa 13.3 to find reqd values of capacitor and inductor

```
clear;
clc;
I_c=10;
f=50;
V_s=230;
C=I_c/(2*%pi*f*V_s);     printf("value of capacitance
   =%.3f uF",C*10^6);
I_l=10;
L=V_s/(2*%pi*f*I_l);     printf("\nvalue of inductor=
   %.3f mH",L*1000);
```

Scilab code Exa 13.4 to find the firing angle of the TCR

```
clc
clear
V_s=230;
I_L=10;
X_L=V_s/I_L;
I_f1=6;
//B=2*a-sin(2*a)
B=2*%pi-I_f1*%pi*X_L/V_s;
a=0;
i=1;
for a= 0:.01:360
    b=2*a*%pi/180-sind(2*a);
    if abs(B-b)<=.001;          //by hit and trial
        i=2;
        break;
    end
end
```

```
printf("firing angle of TCR = %.1f deg",a);
//(a-.01)*180/%pi);
```

Scilab code Exa 13.5 to calculate the effective inductance at different firing angles

```
clear;
clc;
L=.01;
disp("for firing angle=90deg");
a=90*%pi/180;
L_eff=%pi*L/(2*%pi-2*a+sin(2*a));    printf("effective inductance=%.0f mH",L_eff*1000);
disp("for firing angle=120deg");
a=120*%pi/180;
L_eff=%pi*L/(2*%pi-2*a+sin(2*a));    printf("effective inductance=%.3f mH",L_eff*1000);
disp("for firing angle=150deg");
a=150*%pi/180;
L_eff=%pi*L/(2*%pi-2*a+sin(2*a));    printf("effective inductance=%.2f mH",L_eff*1000);
disp("for firing angle=170deg");
a=170*%pi/180;
L_eff=%pi*L/(2*%pi-2*a+sin(2*a));    printf("effective inductance=%.3f H",L_eff);
disp("for firing angle=175deg");
a=175*%pi/180;
L_eff=%pi*L/(2*%pi-2*a+sin(2*a));    printf("effective inductance=%.2f H",L_eff);
disp("for firing angle=180deg");
a=180*%pi/180;
L_eff=%pi*L/(2*%pi-2*a+sin(2*a));    printf("effective inductance=%.3f H",L_eff);
//random value at firing angle =180 is equivalent to infinity as in answer in book
```

Scilab code Exa 13.6 to find value of inductance

```
clear;
clc;
Q=100*10^3;
V_s=11*10^3;
f=50;
L=V_s^2/(2*%pi*f*Q);      printf("effective inductance
   =%.4f H",L);
```

Chapter 14

Miscellaneous Topics

Scilab code Exa 14.1 to calculate dc output voltage rms value of output voltage DF PF and HF

```
clear;
clc;
V_s=230;
V_m=sqrt(2)*V_s;
a1=0;
a2=45;
printf("for two single phase series semiconvertors")
   ;
V_0=V_m/%pi*(2+cosd(a1)+cosd(a2));    printf("\navg
   o/p voltage=%.2f V",V_0);
V_or=V_s*sqrt((1/%pi)*(4*%pi-3*a2*%pi/180+(3/2)*sind
   (2*a2)));    printf("\nrms value of o/p voltage=%
   .2f V",V_or);
DF=(3+cosd(a2))/(sqrt(2)*sqrt(5+3*cosd(a2)));
   printf("\nDF=%.4f",DF);
PF=sqrt(2/%pi)*(3+cosd(a2))/sqrt(4*%pi-3*a2*%pi/180)
   ;    printf("\nPF=%.5f",PF);
HF=sqrt((%pi*(%pi-(3/4)*a2*%pi/180)/(5+3*cosd(a2)))
   -1);    printf("\nHF=%.5f",HF);

```

```
printf("\n\nfor two single phase series full convertors");
a=45;
V_0=2*V_m/%pi*(1+cosd(a));     printf("\navg o/p voltage=%.2f V",V_0);
V_or=2*V_s*sqrt((1/%pi)*(%pi-a2*%pi/180+(1/2)*sind(2*a2)));     printf("\nrms value of o/p voltage=%.2f V",V_or);
DF=cosd(a2/2);     printf("\nDF=%.4f",DF);
PF=sqrt(2/(%pi*(%pi-a2*%pi/180)))*(1+cosd(a2));     printf("\nPF=%.4f",PF);
HF=sqrt((%pi*(%pi-a2*%pi/180)/(4+4*cosd(a2)))-1);     printf("\nHF=%.2f",HF);
```