Scilab Textbook Companion for Electrical Engineering - Principles And Applications

by A. R. Hambley[1]

Created by
Manoj Kumar
B.Tech
Mechanical Engineering
IIT Bombay
College Teacher
NA
Cross-Checked by
Mukul Kulkarni

July 30, 2019

---

[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: Electrical Engineering - Principles And Applications

Author: A. R. Hambley

Publisher: Pearson Education, New Jersey

Edition: 5

Year: 2011

ISBN: 0-13-213006-8

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 1

Introduction

Scilab code Exa 1.1 `example 1`

```
//ex1.1
//As both q and i are 0 when t<0, graph coincides with x-axis till t=0 and we here,show the part where t>0
t=[0:0.000001:0.04];
q=2*(1-%e^(-100*t));
//current i=dq/dt=200*e^(-100*t)
i=200*%e^(-100*t);
subplot(121)
xtitle('charge vs time','time in ms','charge in coulombs')        //ms-milli second(10^-3)
plot(t*10^3,q)
subplot(122)
xtitle('current vs time','time in ms','current in amperes')        //ms-milli second(10^-3)
plot(t*10^3,i)
```

Scilab code Exa 1.2 `example 2`

```
clc
//ex1.2

//element A
disp('ELEMENT A :')
V_a=12;
i_a=2;
P_a=V_a*i_a;        //passive reference configuration (current enters through +ve polarity)
if(P_a>0) then,       //absorption of power
    disp(P_a,'Power for element A in watts is')
    disp('As a battery, element A is being charged')
elseif(P_a<0) then,       //supplying of power
    disp(P_a,'Power for element A in watts is')
    disp('As a battery, element A is being discharged')
end

//element B
disp('ELEMENT B')
V_b=12;
i_b=1;
P_b=-V_b*i_b;       //opposite to passive reference configuration (current enters through -ve polarity)
if(P_b>0) then,       //absorption of power
    disp(P_b,'Power for element B in watts is')
    disp('As a battery, element B is being charged')
elseif(P_b<0) then,       //supplying of power
    disp(P_b,'Power for element B in watts is')
    disp('As a battery, element B is being discharged')
end

//element C
disp('ELEMENT C')
V_c=12;
i_c=-3;
```

```
P_c=V_c*i_c;        //passive reference configuration (current enters through +ve polarity)
if(P_c>0) then,       //absorption of power
    disp(P_c,'Power for element C in watts is')
    disp('As a battery, element C is being charged')
elseif(P_c<0) then,       //supplying of power
    disp(P_c,'Power for element C in watts is')
    disp('As a battery, element C is being discharged')
end
```

Scilab code Exa 1.3 example 3

```
clc
// initialisation of variables
G= 9200 // N/m^2
g1= 9.81 // m/sec^2
g2= 9.805 //m/sec^2
// Calculations
rho= G/g1
G2= rho*g2
// Results
printf ('Density of Fluid = %.1f N sec^2/m^4',rho)
printf ('\n New Specific Weight = %.f N/m^3',G2)
```

Scilab code Exa 1.4 example 4

```
clc
//ex1.4
d=2.05*10^-3;        //diameter of wire
l=10;        //length of wire
P=1.72*10^-8;        //resistivity of copper
A=%pi*d^2/4;        //area of wire
```

```
R=P*l/A;      //resistance of the copper wire
printf(" All the values in the textbook are approximated, hence the values in this code differ from those of textbook")
disp(R,'Resistance of copper wire in ohms')
```

Scilab code Exa 1.5 `example 5`

```
clc
//ex1.5
P=1500;      //power of heater
V=120;      //operating voltage
R=V^2/P;      //resistance of heater element
i=V/R;      //operating current
disp(R,'resistance of heater element in ohms')
disp(i,'operating current in amperes')
```

Scilab code Exa 1.6 `example 6`

```
clc
//ex1.6
V_s=10;      //source voltage
R=5;
V_x=-V_s;      //Voltage across R(applying KVL)
//the actual polarity is opposite to the reference, so we take polarity to be +ve at the top end of resistance
i_x=-V_x/R;      //ohm's law(-ve sign as V_x and i_x have references opposite to passive configuration)
i_y=-i_x;      //current through source
P_s=V_s*i_y;      //power for voltage source
```

```
P_R=-V_x*i_x;       //power for resistance(-ve sign as V_x and i_x have references opposite to passive configuration)
disp(V_x,'voltage across resistance in volts')
disp(i_x,'current through resistance in amperes')
disp(i_y,'current through source in amperes')
disp(P_s,'power for voltage source in watts')
disp(P_R,'power for resistance in watts')
if(V_x==-10&i_x==2&i_y==-2&P_s==-20&P_R==20) then,
    disp('Results are in agreement with those previously found in the textbook')
end
```

Scilab code Exa 1.7 `example 7`

```
clc
//ex1.7
R_1=10;
R_2=5;
V_R_2=15;       //voltage across R_2
a=0.5;
i_y=V_R_2/R_2;       //current across R_2
i_x=i_y*2/3;       //current across R_1, by applying KCL at the top end of the controlled source
V_x=i_x*R_1;       //ohm's law
V_s=V_x+V_R_2;       //KVL around the periphery of the circuit
disp(V_s,'Source voltage for given circuit in volts')
```

Chapter 2

Resistive Circuits

Scilab code Exa 2.1 `example 1`

```
clc
//ex2.1
R_1=10;
R_2=20;
R_3=5;
R_4=15;
//We proceed through various combinations of resistances in series or parallel while we replace them with equivalent resistances  We start with R_3 and R_4.
R_eq_1=R_3+R_4;      //R_3 and R_4 in series
R_eq_2=1/((1/R_eq_1)+(1/R_2));      //R_eq_1 and R_2 in parallel
R_eq=R_1+R_eq_2;      //R_1 and R_eq_2 in series
disp(R_eq,'Equivalent resistance in ohms')
```

Scilab code Exa 2.2 `example 2`

```
clc
//ex2.2
V_s=90;      //source voltage
R_1=10;
R_2=30;
R_3=60;
R_eq_1=1/((1/R_2)+(1/R_3));      //R_2 and R_3 in parallel
R_eq=R_1+R_eq_1;      //R_1 and R_eq_1 in series
i_1=V_s/R_eq;      //ohm's law
//i_1 flows clockwise through V_s,R_1 and R_eq_1
V_2=R_eq_1*i_1;      //voltage across R_eq_1
//As R_eq_1 is equivalent of parallel combination of R_2 and R_3, V_2 appears across both of them
i_2=V_2/R_2;      //ohm's law
i_3=V_2/R_3;      //ohm's law
//we can verify KCL, i_1=i_2+i_3
V_1=i_1*R_1;      //ohm's law
//we can verify KVL, V_s=V_1+V_2
P_s=-V_s*i_1;      //source power(-ve sign as V_s and i_1 have references opposite to passive configuration)
P_1=i_1^2*R_1;      //power for R_1
P_2=V_2^2/R_2;      //power for R_2
P_3=V_2^2/R_3;      //power for R_3
disp('FOR SOURCE')
disp(i_1,'current in amperes')
disp(P_s,'power in watts')
disp('FOR R1')
disp(i_1,'current in amperes')
disp(V_1,'voltage in volts')
disp(P_1,'power in watts')
disp('FOR R2')
disp(i_2,'current in amperes')
disp(V_2,'voltage in volts')
disp(P_2,'power in watts')
disp('FOR R3')
disp(i_3,'current in amperes')
```

```
disp(V_2,'voltage in volts')
disp(P_3,'power in watts')
//we may verify that P_s+P_1+P_2+P_3=0
```

Scilab code Exa 2.3 example 3

```
//ex2.3
V_total=15;
R_1=1*10^3;
R_2=1*10^3;
R_3=2*10^3;
R_4=6*10^3;
//By voltage-division priciple
V_1=R_1*V_total/(R_1+R_2+R_3+R_4);      //voltage
   across R_1
V_4=R_4*V_total/(R_1+R_2+R_3+R_4);      //voltage
   across R_4
disp(V_1,'voltage across R_1')
disp(V_4,'voltage across R_4')
```

Scilab code Exa 2.4 example 4

```
clc
//ex2.4
V_s=100;      //source current
R_1=60;
R_2=30;
R_3=60;
R_x=1/((1/R_2)+(1/R_3));      //R_2 and R_3 parallel
V_x=R_x*V_s/(R_1+R_x);      //voltage across R_x(
   voltage-division principle)
i_s=V_s/(R_1+R_x);      //ohm's law
```

```
i_3=R_2*i_s/(R_2+R_3);       //current through R_3(
   current-division principle)
printf(" All the values in the textbook are
   approximated, hence the values in this code
   differ from those of Textbook")
disp(V_x,'voltage across R2 or R3 in volts')
disp(i_s,'source current in amperes')
disp(i_3,'current through R3 in amperes')
```

Scilab code Exa 2.5 `example 5`

```
clc
//ex2.5
i_s=15;       //source current
R_1=10;
R_2=30;
R_3=60;
R_eq=1/((1/R_2)+(1/R_3));       //R_2 and R_3 in
   parallel
i_1=R_eq*i_s/(R_1+R_eq);       //current through R_1(
   current-division principle)
disp(i_1,'current through R1 in amperes from
   resistance method')
//we can also do the above calculations using
   conductances as shown below.
//Conductances of respective resistances
G_1=1/R_1;
G_2=1/R_2;
G_3=1/R_3;
i_1=G_1*i_s/(G_1+G_2+G_3);
disp(i_1,'current through R1 in amperes from
   conductance method')
disp('We get the same alue in both methods')
```

Scilab code Exa 2.6 `example 6`

```
clc
//ex2.6
//we display the equations in scilab as follows
disp('At node 1:')
disp('(V1/R1)+((V1-V2)/R2)+i_s=0')        //KCL at node 1
disp('At node 2:')
disp('((V2-V1)/R2)+(V2/R3)+((V2-V3)/R4)=0')        //KCL at node 2
disp('At node 3:')
disp('(V3/R5)+((V3-V2)/R4))=i_s')        //KCL at node 3
```

Scilab code Exa 2.7 `example 7`

```
clc
//ex2.7
disp('The matrix form is')
disp('G*V=I')
disp('where')
G=[0.45,-0.25,0;-0.25,0.85,-0.20;0,-0.20,0.30];
disp(G,'G=')
disp('V=')
disp('transpose of [V_1,V_2,V_3]')
disp('and')
I=[-3.5;3.5;2];
disp(I,'I=')
```

Scilab code Exa 2.8 `example 8`

```
clc
//ex2.8
R=20;
G=[0.35,-0.2,-0.05;-0.2,0.3,-0.1;-0.05,-0.1,0.35];      //coefficient matrix
I=[0;10;0]        //current matrix
V=G\I;       //voltage matrix(from G=V*I)
i_x=(V(1)-V(3))/R;
printf(" All the values in the textbook are approximated,hence the values in this code differ from those of textbook")
disp(V(1),'voltage at node1 in volts')
disp(V(2),'voltage at node2 in volts')
disp(V(3),'voltage at node3 in volts')
disp(i_x,'value of current ix in amperes')
```

Scilab code Exa 2.9 `example 9`

```
clc
//ex2.9
//we display the required equations as follows
disp('Current equations at node1 and node2:')
disp('((V1-V2)/5)+((V1-10)/2)=1')
disp('(V2/5)+((V2-10)/10)+((V2-V1)/5)=0')
disp('Writing the above equations in standard form')
disp('0.7V1-0.2V2=6')
disp('-0.2V1+0.5V2=1')
```

Scilab code Exa 2.11 `example 11`

```
clc
```

```
//ex2.11
disp('KCL for a supernode enclosing the conrolled
   voltage source')
disp('(V1/R2)+((V1-V3)/R1)+((V2-V3)/R3)=is')
disp('KCL at node 3')
disp('(V3/R4)+((V3-V2)/R3)+((V3-V1)/R1)=0')
disp('KCL at the reference node')
disp('(V1/R2)+(V3/R4)=is')
disp('From the closed loop with V1,Vx and V3')
disp('Vx=V3-V1')
disp('Applying KVL')
disp('V1=0.5(V3-V1)+V2')
disp('The last KVL equation along with any two of
   the first three KCL equations forms an
   independent set that can be solved for the node
   voltages.')
```

Scilab code Exa 2.12 `example 12`

```
clc
//ex2.12
//In all the below equations, mesh currents are
   taken to be flown in clockwise direction
disp('The required equations to solve for mesh
   currents are:')
disp('R2(i1-i3)+R3(i1-i2)-VA=0')        //KVL for
   mesh1
disp('R3(i2-i1)+R4(i2)+VB=0')        //KVL for mesh 2
disp('R2(i3-i1)+R1(i3)-VB=0')        //KVL for mesh 3
```

Scilab code Exa 2.13 `example 13`

```
clc
```

```
//ex2.13
R=[30 -10 -20;-10 22 -12;-20 -12 46];      //coefficient matrix
V=[70;-42;0]      //voltage matrix
I=R\V;      //current matrix(from R*I=V)
disp(I(1),'current in mesh1 in amperes, i1=')
disp(I(2),'current in mesh2 in amperes, i2=')
disp(I(3),'current in mesh3 in amperes, i3=')
```

Scilab code Exa 2.14 `example 14`

```
clc
//ex2.14
//taking mesh currents i1, i2 and i3 in clockwise direction
disp('The matrix form is')
disp('RI=V')
disp('where the matrices are defined as')
disp('R=[R2+R4+R5,-R2,-R5;-R2,R1+R2+R3,-R3;-R5,-R3,R3+R5+R6]')
disp('I=[i1;i2;i3]')
disp('V=[-VA+VB;VA;-VB]')
```

Scilab code Exa 2.15 `example 15`

```
clc
//ex2.15
//KVL over the supermesh, we get eqn-1   -20+4(i1)+8(i2)=0
//Vx=2(i2)   ohm's law
//writing an expression for the source current in terms of mesh currents and substituting Vx from above, we get eqn-2   (1/2)i2=i2-i1
```

```
//Putting eqn-1 and eqn-2 in standard form   4(i1)+8(i2)=20 and i1-(1/2)i2=0
//solving for currents in matrix method(Ax=b)
A=[4,8;1,-1/2];      //coeffcient matrix
b=[20;0];      //constant matrix
x=A\b;      //solution
disp(x(1),'Value of i1 in amperes')
disp(x(2),'Value of i2 in amperes')
```

Scilab code Exa 2.16 `example 16`

```
clc
//ex2.16
V_s=15;      //source voltage
R_1=100;
R_2=50;
//Analysis with an open circuit to find V_t
i_1=V_s/(R_1+R_2);      //closed circuit with R_1 and R_2 in series
V_oc=R_2*i_1;      //open-circuit voltage across R_2
V_t=V_oc;      //thevenin voltage
//Analysis with a short-circuit to find i_sc
i_sc=V_s/R_1;      //R_2 is short-circuited
R_t=V_oc/i_sc;      //thevenin resistance
printf(" All the values in the textbook are approximated, hence the values in this code differ from those of textbook")
disp(V_t,'Thevenin voltage for given circuit in volts')
disp(R_t,'Thevenin voltage for given circuit in ohms')
```

Scilab code Exa 2.17 `example 17`

```
clc
//ex2.17
V_s=20;       //source voltage
i_s=2;       //source current
R_1=5;
R_2=20;
//after zeroing the sources which includes replacing voltage source with short circuit and current source with open circuit , we get R_t
R_eq=1/((1/R_1)+(1/R_2));       //R_1 and R_2 are in parallel combination
R_t=R_eq;       //Thevenin resistance
//short-circuit analysis to find i_sc
i_2=0;       //voltage across R_2 is 0
i_1=V_s/R_1;
i_sc=i_1+2-i_2;       //short-circuit current(KCL at junction of R_2 and I_s)
V_t=R_t*i_sc;       //thevenin voltage
disp(i_sc,'short-circuit current in amperes')
disp(R_t,'thevenin resistance in ohms')
disp(V_t,'thevenin voltage in volts')
//thevenin equivalent can be made of V_t and R_t.
```

Scilab code Exa 2.18 example 18

```
clc
//ex2.18
V=10;
R_1=5;
R_2=10;
//Open-circuit anlaysis
//let V_oc be the open circuit voltage
//Current equation at node1    3(i_x)=(1/10)V_oc
//i_x=(10-V_oc)/5    ix in terms of V_oc
V_oc=2/((1/5)+(1/30));       //open-circuit voltage(
```

```
   from above two equations)
V_t=V_oc;        //thevenin voltage
//short-circuit analysis
i_x=V/R_1;
i_sc=3*i_x;        //short-circuit current
R_t=V_oc/i_sc;
printf(" All the values in the textbook are approximated, hence the values in this code differ from those of textbook")
disp(V_t,'Thevenin voltage in volts')
disp(R_t,'Thevenin resistance in ohms')
```

Scilab code Exa 2.19 `example 19`

```
clc
//ex2.19
R1= 20 //Ohms
R2= 15 //ohms
vs= 15 //V
R3= 5 //Ohms
k= 0.25
///CALCULATIONS
voc= (R2/R1)/((1/R1)+(1/(R2+R3))+(k/4))
isc= vs/R1
Rf= voc/isc
//RESULTS
printf ('Rf = %.2f ohms',Rf)
```

Scilab code Exa 2.20 `example 20`

```
clc
//ex2.20
V_s_1=20;        //voltage source
```

```
R_1=5;
R_2=10;
i_s_1=1;       //current source
//Method 1: To transform current source and R_2 into a voltage source in series with R_2
V_s_2=i_s_1*R_2;      //source transformation
i_1=(V_s_1-V_s_2)/(R_1+R_2);      //clockwise KVL
i_2=i_1+i_s_1;      //KCL at top node of original circuit
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp('By current source to voltage source transformation:')
disp(i_1,'current i1 in amperes')
disp(i_2,'current i2 in amperes')
//Method 2: To transform voltage source and R_1 into a current source in parallel with R_1
i_s_2=V_s_1/R_1;      //source transformation
i_t=i_s_2+i_s_1;      //total current
i_2=R_1*i_t/(R_1+R_2)      //current-division principle
i_1=i_2-i_s_1;      //KCL at top node of original circuit
disp('By voltage source to current source transformation:')
disp(i_1,'current i1 in amperes')
disp(i_2,'current i2 in amperes')
disp('In any method we get the same answers.')
```

Scilab code Exa 2.21 `example 21`

```
clc
//ex2.21
V_s=50;
```

```
R_1=20;
R_2=5;
//Zeroing the voltage source
R_eq=1/((1/R_1)+(1/R_2));       //R_1 and R_2 in parallel
R_t=R_eq;       //thevenin resistance
//open-circuit analysis
V_oc=V_s*R_2/(R_1+R_2);       //open-circuit voltage
V_t=V_oc;       //thevenin voltage
R_L=R_t;
P_L_max=V_t^2/(4*R_t)
disp(R_L,'load resistance for maximum power transfer in ohms')
disp(P_L_max,'maximum power in watts')
```

Scilab code Exa 2.22 example 22

```
clc
//ex2.22
V_s=15;       //voltage source
R_1=10;
R_2=5;
i_s=2;       //current source
//Analysis with only voltage source active
V_1=R_2*V_s/(R_1+R_2);       //voltage-division principle
//Analysis with only current source active
R_eq=1/((1/R_1)+(1/R_2));       //R_1 and R_2 in parallel
V_2=i_s*R_eq;       //ohm's law
V_T=V_1+V_2;       //total response
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp(V_T,'VT i.e., voltage across R2 in volts')
```

Scilab code Exa 2.23 `example 23`

```
clc
//ex2.23
R_1=1*10^3;
//case (a)
disp('case a:')
R_2=10*10^3;
R_3=732;
R_x=R_2*R_3/R_1;        //wheatstone bridge condition
disp(R_x,'Value of Rx in ohms')
//case (b)
disp('case b:')
//R_x is maximum when both R_2 and R_3 are maximum
R_2_max=1*10^6;
R_3_max=1100;
R_x_max=R_2_max*R_3_max/R_1;        //wheatstone bridge condition
disp(R_x_max,'Maximum value of Rx in ohms')
//case(c)
disp('case c:')
//increment in R_x is scale factor times increment in R_3
R_2=1*10^6;
R_3_inc=1;        //increment in R_3
R_x_inc=R_2*R_3_inc/R_1;        //increment in R_x from bride balance condition
disp(R_x_inc,'Increment between values of Rx in ohms for the bridge to be balanced')
```

Chapter 3

Inductance and Capacitance

Scilab code Exa 3.1 `example 1`

```
clc
//ex3.1
C=1*10^-6;
//t in micro seconds
t_1=[0:0.001:2];
t_2=[2.001:0.001:4];
t_3=[4.001:0.001:5];
t=[t_1,t_2,t_3];
//corresponding voltage variations
V_1=5*t_1;
V_2=0*t_2+10;
V_3=-10*t_3+50;
//charge q=C*V
q_1=C*V_1;
q_2=C*V_2;
q_3=C*V_3;
q=[q_1,q_2,q_3];
subplot(121)
plot(t,q*10^6)
xtitle('charge vs time','time in Ms','charge in Mc')
        //M—micro(10^-6)
```

```
//current i=C*dV/dt*10^6, for above equations we get
i_1=10^6*(0*t_1+C*(5));
i_2=10^6*0*t_2;
i_3=10^6*(0*t_3+C*(-10));
i=[i_1,i_2,i_3];
subplot(122)
plot(t,i)
xtitle('current vs time','time in Ms','current in amperes')        //M-micro(10^-6)
```

Scilab code Exa 3.2 `example 2`

```
clc
//ex3.2
C=0.1*10^-6;
//symbolic integration cannot be done in scilab
t=[0:0.001*10^-3:3*%pi*10^-4];
i=0.5*sin((10^4)*t);
//on integrating 'i' w.r.t t
q=0.5*10^-4*(1-cos(10^4*t));
C=10^-7;
V=q/C;
subplot(221)
plot(t,q*10^6)
xtitle('charge vs time','time in seconds','charge in Mc')        //Mc=micro coulombs(10^-6)
subplot(222)
plot(t,i)
xtitle('current vs time','time in seconds','current in amperes')        //Mc=micro coulombs(10^-6)
subplot(223)
xtitle('voltage vs time','time in seconds','voltage in volts')
plot(t,V)
```

Scilab code Exa 3.3 `example 3`

```
clc
//ex3.3
C=10*10^-6;
t_1=[0:0.001:1];
t_2=[1.001:0.001:3];
t_3=[3.001:0.001:5];
t=[t_1,t_2,t_3];
//voltage variations
V_1=1000*t_1;
V_2=0*t_2+1000;
V_3=500*(5-t_3);
//current i=C*dv/dt, for above equations we get
i_1=C*(0*t_1+1000);
i_2=C*(0*t_2);
i_3=C*(0*t_3-500);
i=[i_1,i_2,i_3];
//power delivered, P=V*i
P_1=C*(10^6*t_1);
P_2=C*(0*t_2+1000);
P_3=C*(-25*10^4*(5-t_3));
P=[P_1,P_2,P_3];
//energy stored, W=(1/2)*C*V^2
W_1=(1/2)*C*V_1^2;
W_2=(1/2)*C*V_2^2;
W_3=(1/2)*C*V_3^2;
W=[W_1,W_2,W_3];
subplot(221)
plot(t,i*10^3)
xtitle('current vs time','time in seconds','current in mA')      //mA-milli amperes(10^-3)
subplot(222)
plot(t,P)
```

```
xtitle('power delivered vs time','time in seconds','power in watts')
subplot(223)
plot(t,W)
xtitle('energy stored vs time','time in seconds','work in joules')
```

Scilab code Exa 3.4 `example 4`

```
clc
//ex3.4
L=10*10^-2;      //length
W=20*10^-2;      //width
d=0.1*10^-3;      //distance between plates
A=L*W;      //area
E_o=8.85*10^-12;      //dielectric constant of vacuum
//dielectric is air
E_r=1;      //relative dielectric constant of air
E=E_r*E_o;      //dielectric constant
C=E*A/d;      //capacitance
disp('When the dielectric is air, capacitance in pF is')      //pF-pico Farad(10^-12)
disp(C*10^12)
//dielectric is mica
E_r=7;      //relative dielectric constant of mica
E=E_r*E_o;      //dielectric constant
C=E*A/d;      //capacitance
disp('When the dielectric is mica, capacitance in pF is')      //pF-pico Farad(10^-12)
disp(C*10^12)
```

Scilab code Exa 3.5 `example 5`

```
clc
//ex3.5
C_1=1*10^-6;
C_2=1*10^-6;
//Before the switch is closed
V_1=100;
V_2=0;
W_1=(1/2)*C_1*V_1^2;
W_2=0;       //V_2=0
W_t_1=W_1+W_2;       //total energy stored by both the capacitors before switch is closed
q_1=C_1*V_1;
q_2=0;
//After the switch is closed
q_eq=q_1+q_2;       //charge on equivalent capacitance
C_eq=C_1+C_2;       //C_1 and C_2 in parallel
V_eq=q_eq/C_eq;
V_1=V_eq;       //parallel combination
V_2=V_eq;       //parallel combination
W_1=(1/2)*C_1*V_eq^2;
W_2=(1/2)*C_2*V_eq^2;
W_t_2=W_1+W_2;       //total energy stored by both the capacitors after switch is closed
disp(W_t_1*10^3,'Total energy stored by both the capacitors before switch is closed in mJ')       //mJ-milli Joules(10^-3)
disp(W_t_2*10^3,'Total energy stored by both the capacitors after switch is closed in mJ')       //mJ-milli Joules(10^-3)
```

Scilab code Exa 3.6 `example 6`

```
clc
//ex3.6
```

```
L=5;        //inductance
t_1=[0:0.001:2];
t_2=[2.001:0.001:4];
t_3=[4.001:0.001:5];
t=[t_1,t_2,t_3];
//corresponding current variations
i_1=(1.5)*t_1;
i_2=0*t_2+3;
i_3=(-3*t_3)+15;
//voltage V=L*(di/dt)
V_1=L*(0*t_1+(1.5));
V_2=L*(0*t_2);
V_3=L*(0*t_3-3);
V=[V_1,V_2,V_3];
//stored energy W=1/2*L*i^2
W_1=(1/2)*L*i_1^2;
W_2=(1/2)*L*i_2^2;
W_3=(1/2)*L*i_3^2;
W=[W_1,W_2,W_3];
//power P=V*i
P_1=L*t_1*(1.5^2);
P_2=0*t_2;
P_3=-3*L*(-3*t_3+15);
P=[P_1,P_2,P_3];
subplot(221)
plot(t,V)
xtitle('voltage vs time','time in seconds','voltage in volts')
subplot(222)
plot(t,W)
xtitle('energy vs time','time in seconds','energy in joules')
subplot(223)
plot(t,P)
xtitle('power vs time','time in seconds','power in watts')
```

Chapter 4

Transients

Scilab code Exa 4.1 `example 1`

```
clc
//ex4.1
V_s=10;       //source voltage
R_1=5;
R_2=5;
L=1;
C=1*10^-6;
//for t>>0, we apply steady state conditions i.e., inductor and capacitor are replaced by short and open circuits respectively
R_eq=R_1+R_2;      //R_1 and R_2 in series
i_x=V_s/R_eq;      //ohm's law
V_x=R_2*i_x;      //voltage across R_2
disp(i_x,'current ix in amperes')
disp(V_x,'voltage Vx in volts')
```

Scilab code Exa 4.2 `example 2`

```
clc
//ex4.3
//Vs is a direct source
//Circuit is in steady state prior to t=0
//Before t=0, the inductor behaves as a short circuit ==>V(t)=0 for t<0 and i(t)=Vs/Ri for t<0
//Before the switch opens, current circulates through Vs,R1 and the inductance and When it opens, nothing changes but the return path through R2
//Then, a voltage appears across R2 and the inductance, causing the current to decay
//There are no sources driving the circuit after the switch opens ==>the steady-state solution is zero for t>0
//Hence, the solution for i(t) is given by i(t)=K*e^(-t/T) for t>0 in time constant T=L/R2
//For current to be continuous i(0+)=(Vs/R1)=K*e^0=K ==> K=Vs/R1
//The voltage is given by V(t)=(L*d(i(t))/dt)=-(L*Vs*e^(-t/T))/(R1*T) for t>0
disp('Both current and voltage are 0 for t<0')
disp('')
disp('And for t>0:')
disp('The expression for the current is i(t)=(Vs/R1)*e^(-t/T)')
disp('The expression for the volatge is V(t)=-(L*Vs*e^(-t/T))/(R1*T)')
```

Chapter 5

Steady state sinusoidal analaysis

Scilab code Exa 5.1 `example 1`

```
clc
//ex5.1
R=50;
t=[0:0.000001:0.05];
V_t=100*cos(100*%pi*t);
V_m=100;        //peak value
V_rms=V_m/sqrt(2);
P_avg=(V_rms^2)/R;
P_t=V_t^2/R;
printf(" All the values in the textbook are
   approximated hence the values in this code differ
    from those of Textbook")
disp(V_rms,'RMS value of voltage in volts')
disp(P_avg,'average power in watts')
subplot(121)
plot(t*10^3,V_t);
xtitle('voltage vs time','time in ms','voltage in
   volts')       //ms-milli seconds(10^-3)
subplot(122)
```

```
plot(t*10^3,P_t)
xtitle('power vs time','time in ms','power in watts')       //ms-milli seconds(10^-3)
```

Scilab code Exa 5.2 `example 2`

```
clc
//ex5.2
//plot of V and t(already given with the question but to get clarity we plot it)
t_1=[0:0.001:1];
t_2=[1.001:0.001:2];
t=[t_1,t_2];
V_1=3*t_1;
V_2=6-3*t_2;
V=[V_1,V_2];
plot(t,V)
xtitle('voltage vs time','time in seconds','voltage in volts')
//now find V_rms
T=2;       //from the plot of V vs t
V_rms=sqrt((1/T)*((integrate('(3*t_1)^2','t_1',0,1))+(integrate('(6-3*t_2)^2','t_2',1,2))));
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp(V_rms,'RMS value in volts')
```

Scilab code Exa 5.3 `example 3`

```
clc
//ex5.3
//V_1 and V_2 are phasors of given voltages
```

```
theta_1=-%pi/4;       //for V_1
theta_2=-%pi/6;       //for V_2 (in cos form)
V_1=complex(20*cos(theta_1),20*sin(theta_1));
V_2=complex(10*cos(theta_2),10*sin(theta_2));
V_s=V_1+V_2;
V=sqrt((real(V_s)^2)+(imag(V_s)^2));       //peak voltage of resultant summation
phi=atan(imag(V_s)/real(V_s));       //phase angle of resultant sum voltage
printf(” All the values in the textbook are approximated hence the values in this code differ from those of Textbook”)
disp(V,'Peak value of resultant voltage in volts’)
disp(phi*180/%pi,'phase of resulting voltage in degrees’)       //converting phi in radians to degrees
//result : V_t=Vcos(wt+phi)
```

Scilab code Exa 5.4 example 4

```
clc
//ex5.4
L=0.3;
C=40*10^-6;
R=100;
V_s_max=100;       //peak value of given voltage
W=500;       //angular frequency
V_s_phi=%pi/6;       //phase angle in degrees
V_s=complex(V_s_max*cos(V_s_phi),V_s_max*sin(V_s_phi));       //phasor for voltage source
Z_L=%i*W*L;       //complex impedance of inductance
Z_C=-%i/(W*C);       //complex impedance of capacitance
Z_eq=R+Z_L+Z_C;       //R,Z_L and Z_C in series
I=V_s/Z_eq;       //phasor current
```

```
V_R=R*I;
V_L=Z_L*I;
V_C=Z_C*I;
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
//for resistance R
disp('For resistance R')
V_R_max=sqrt((real(V_R)^2)+(imag(V_R)^2));
V_R_phi=(atan(imag(V_R)/real(V_R)))*180/%pi;
disp(V_R_max,'peak value of voltage in volts')
disp(V_R_phi,'phase angle in degrees')
//result : V_R=Vcos(wt+phi)  V-peak voltage
//for inductance L
disp('For inductance L')
V_L_max=sqrt((real(V_L)^2)+(imag(V_L)^2));
V_L_phi=(atan(imag(V_L)/real(V_L)))*180/%pi;
disp(V_L_max,'peak value of voltage in volts')
disp(V_L_phi,'phase angle in degrees')
//result : V_L=Vcos(wt+phi)  V-peak voltage
//for capacitor C
disp('For capacitor C')
V_C_max=sqrt((real(V_C)^2)+(imag(V_C)^2));
V_C_phi=(atan(imag(V_C)/real(V_C)))*180/%pi;
disp(V_C_max,'peak value of voltage in volts')
disp(V_C_phi,'phase angle in degrees')      //cos(t)=cos(t-180)  (we get 75 instead of -105 given in textbook)
//result : V_C=Vcos(wt+phi)  V-peak voltage
disp('The phasor diagram cannot be plotted')
```

Scilab code Exa 5.5 `example 5`

```
clc
//ex5.5
```

```
V_s_max=10;        //peak value of source voltage
phi=-%pi/2;        //phase of source voltage
V_s=complex(10*cos(%pi/2),10*sin(%pi/2));        //phasor of source voltage
W=1000;        //angular frequency
R=100;
L=0.1;
C=10*10^-6;
Z_L=%i*W*L;        //impedance of inductance
Z_C=-%i/(W*C);        //impedance of capacitance
Z_RC=1/((1/R)+(1/Z_C));        //R and Z_C in parallel combination
V_C=V_s*Z_RC/(Z_L+Z_RC);        //voltage division principle
I=V_s/(Z_L+Z_RC);        //current through source and inductor
I_R=V_C/R;        //current through resistance
I_C=V_C/Z_C;        //current through capacitor
//cos(t)=cos(180-t)
disp(sqrt((real(V_C)^2)+(imag(V_C)^2)),'peak value of Vc in volts')
disp((atan(imag(V_C)/real(V_C)))*180/%pi,'phase angle of Vc in degrees')
////result : V_C=Vcos(wt+phi)  V-peak voltage
disp(I,'current through source and inductor in amperes')
disp(I_R,'current through resistance in amperes')
disp(I_C,'current through capacitance in amperes')
disp('phasor diagram cannot be plotted')
```

Scilab code Exa 5.6 `example 6`

```
clc
//ex5.6
V_s_max=2;        //peak value of source voltage
```

```
V_s_phi=-%pi/2;      //phase angle of source voltage
V_s=complex(V_s_max*cos(V_s_phi),V_s_max*sin(V_s_phi));
R=10;
Z_C=-%i*5;      //impedance of capacitance
Z_L=%i*10;      //impedance of inductance
I_s_max=1.5;      //peak value of current source
I_s_phi=0;      //phase angle of current source
I_s=complex(I_s_max*cos(I_s_phi),I_s_max*sin(I_s_phi));
//we write the standard equations of V_1 and V_2 in matrix form
//from node-voltage relation
A=[0.1+%i*0.2,-%i*0.2;-%i*0.2,%i*0.1];      //coefficient matrix
B=[-%i*2;1.5];      //constant matrix
//As in A*X=B form
V=inv(A)*B;
V_1=sqrt((real(V(1,:)))^2+(imag(V(1,:)))^2);      //peak value of V_1
V_1_phi=atan(imag(V(1,:))/real(V(1,:)));      //phase angle of V_1
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp(V_1,'peak value of V1 in volts')
disp(V_1_phi*180/%pi,'phase angle of V1 in degrees')
```

Scilab code Exa 5.7 example 7

```
//ex5.7
phi_v=-%pi/2;      //angle of voltage source
phi_i=-3*%pi/4;      //angle of current source
phi=phi_v-phi_i;      //power angle
V_s_max=10;      //peak value of voltage source
```

```
V_s_phi=phi_v;      //phase angle of voltage source
R=100;
V_s=complex(V_s_max*cos(V_s_phi),V_s_max*sin(V_s_phi));      //phasor of voltage source
X_L=%i*100;
X_C=-%i*100;
I_max=0.1414;      //peak value of current
I_phi=phi_i;      //phase angle of current
I=complex(I_max*cos(I_phi),I_max*sin(I_phi));      //phasor of current
V_s_rms=V_s_max/sqrt(2);      //rms value of voltage
I_rms=I_max/sqrt(2);      //rms value of current
I_R_max=0.1;      //peak value
I_R_phi=-2*%pi;      //phase angle
I_R=complex(I_R_max*cos(I_R_phi),I_R_max*sin(I_R_phi));      //phasor of current
I_R_rms=I_R_max/sqrt(2);      //rms value
I_C_max=0.1;      //peak value
I_C_phi=-%pi/2;      //phase angle
I_C=complex(I_C_max*cos(I_C_phi),I_C_max*sin(I_C_phi));      //phasor current in capacitor
I_C_rms=I_C_max/sqrt(2);      //rms value
P=V_s_rms*I_rms*cos(phi);      //power by source
Q=V_s_rms*I_rms*sin(phi);      //reactive power by source
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp(P,'power delivered by source in watts')
disp(Q,'reactive power delivered by source in VARs')
//using complex power method
disp('Using complex power method:')
S=(1/2)*V_s*(I');      //complex power
P=real(S);
Q=imag(S);
disp(P,'power delivered by source in watts')
disp(Q,'reactive power delivered by source in VARs')
disp('we see that, in both the methods answers are
```

```
the same')
Q_L=I_rms^2*X_L/%i;       //reactive power to inductance
Q_C=I_C_rms^2*X_C/%i;       //reactive power to capacitance
P_R=I_R_rms^2*R;       //power to resistance
disp(Q_L,'reactive power delivered to inductance in VARs')
disp(Q_C,'reactive power delivered to capacitance in VARs')
disp(P_R,'power delivered to resistance in watts')
```

Scilab code Exa 5.8 `example 8`

```

clc
//initialisation of variables
clear
Vrms = 10^2 //V
Irms= 10^2 //amp
pf= 0.5
pf1= 0.7
r= 1.41
//CALCULATIONS
PA= Vrms*Irms*pf
QA= -sqrt((Vrms*Irms)^2-PA^2)/1000
a= acosd(pf1)
QB= PA*tand(a)/1000
P= 2*PA/1000
Q= QA+QB
o= atand(Q/P)
pf2= cosd(o)
A= o+69.18
S= sqrt(P^2+Q^2)
I= S*r
```

```
//RESULTS
printf ('Phasor Current = %.f A',I)
printf ('\n Angle = %.2f degrees',A)
```

Scilab code Exa 5.9 `example 9`

```
clc
//ex5.9
//L is load
P_L=50*10^3;      //power of load
f=60;      //frequency
V_rms=10*10^3;      //rms voltage
PF_L=0.6;      //power factor
phi_L=acos(PF_L);      //power angle
Q_L=P_L*tan(phi_L);      //reactive power of load
//when capacitor is added, power angle changes
PF_L_new=0.9;
phi_L_new=acos(PF_L_new);
Q_new=P_L*tan(phi_L_new);
Q_C=Q_new-Q_L;      //reactive power of capacitance
X_C=-V_rms^2/Q_C;      //reactance of capacitor
W=2*%pi*f;      //angular frequency
C=1/(W*abs(X_C));      //capacitance
printf(" All the values in the textbook are
   approximated hence the values in this code differ
    from those of Textbook")
disp(C*10^6,'Required capacitance in micro-farads')
```

Scilab code Exa 5.10 `example 10`

```
clc
//ex5.10
R=100;
```

```
V_s_max=100;        //peak value of voltage
V_s_phi=0;        //phase angle of voltage
V_s=complex(V_s_max*cos(V_s_phi),V_s_max*sin(V_s_phi));        //phasor of voltage
Z_C=-%i*100;        //impedance of capacitance
I_s_max=1;        //peak value of current
I_s_phi=%pi/2;        //phase angle of current
I_s=complex(I_s_max*cos(I_s_phi),I_s_max*sin(I_s_phi));        //phasor of current
//zeroing sources to find Z_t i.e., thevenin impedance
Z_t=1/((1/R)+(1/Z_C));        //R and Z_C are in parallel combination
//apply short-circuit to find I_sc i.e., short-circuit current
I_R=abs(V_s)/R;        //ohm's law
I_sc=I_R-I_s;        //applying KCL
V_t=I_sc*Z_t;        //thevenin voltage
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp('FOR THEVENIN CIRCUIT:')
disp('thevenin voltage')
disp(abs(V_t),'peak value of voltage in volts')
//cos(t)=cos(t-180)
disp(atan(imag(V_t)/real(V_t))*180/%pi,'phase angle in degrees')
disp('thevenin resistance')
disp(abs(Z_t),'peak value of resistance in ohms')
disp(atan(imag(Z_t)/real(Z_t))*180/%pi,'phase angle in degrees')
disp('FOR NORTON CIRCUIT:')
disp('norton current')
disp(abs(I_sc),'peak value of norton current in amperes')
disp(atan(imag(I_sc)/real(I_sc))*180/%pi,'phase angle in degrees')
disp('resistance')
```

```
disp(abs(Z_t),'peak value of resistance in ohms')
disp(atan(imag(Z_t)/real(Z_t))*180/%pi,'phase angle in degrees')
```

Scilab code Exa 5.11 example 11

```
clc
//ex5.11
//thevenin voltage
V_t_max=100;
V_t_phi=-%pi/2;
V_t=complex(V_t_max*cos(V_t_phi),V_t_max*sin(V_t_phi));
//thevenin resistance
Z_t_max=70.71;
Z_t_phi=-%pi/4;
Z_t=complex(Z_t_max*cos(Z_t_phi),Z_t_max*sin(Z_t_phi));
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
//a) Any complex load
disp('FOR ANY COMPLEX LOAD')
Z_load=Z_t';
I_a=V_t/(Z_t+Z_load);       //ohm's law
I_a_rms=I_a/sqrt(2);       //rms value
P_1=abs(I_a_rms)^2*real(Z_load);       //power
disp(Z_load,'required complex load impedance')
disp(P_1,'power delivered to load in watts')
//b) purely resistive load
disp('FOR PURE RESISTIVE LOAD')
R_load=abs(Z_t);
I_b=V_t/(Z_t+R_load);
I_b_rms=I_b/sqrt(2);
P_2=abs(I_b_rms)^2*R_load;
```

```
disp(R_load,'required pure resistive load')
disp(P_2,'power delivered to load')
```

Scilab code Exa 5.12 `example 12`

```
clc
//ex5.12
V_Y=1000;        //line to neutral voltage
f=60;       //frequency
L=0.1;       //inductance
R=50;
W=2*%pi*f;        //angular frequency
Z=complex(R,W*L);        //complex impedance
phi=atan(imag(Z)/real(Z));
//Balanced wye−wye calculations
V_an=complex(1000*cos(0),1000*sin(0));
V_bn=complex(1000*cos(-2*%pi/3),1000*sin(-2*%pi/3));
V_cn=complex(1000*cos(2*%pi/3),1000*sin(2*%pi/3));
I_aA=V_an/Z;
I_bB=V_bn/Z;
I_cC=V_cn/Z;
//line−line phasors
V_ab=V_an*sqrt(3)*complex(cos(%pi/6),sin(%pi/6));
V_bc=V_bn*sqrt(3)*complex(cos(%pi/6),sin(%pi/6));
V_ca=V_cn*sqrt(3)*complex(cos(%pi/6),sin(%pi/6));
I_L=abs(I_aA);
P=(3/2)*V_Y*I_L*cos(phi);        //power
Q=(3/2)*V_Y*I_L*sin(phi);        //reactive power
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp('LINE CURRENTS')
disp(I_aA,'IaA=')
disp(I_bB,'IbB=')
disp(I_cC,'IcC=')
```

```
disp('LINE-LINE VOLTAGES')
disp(V_ab,'Vab=')
disp(V_bc,'Vbc=')
disp(V_ca,'Vca=')
disp(P,'POWER IN WATTS')
disp(Q,'REACTIVE POWER IN VARs')
disp('the phasor diagram cannot be plotted')
```

Scilab code Exa 5.13 `example 13`

```
clc
//ex5.13
Z_line=complex(0.3,0.4);        //impedance of wire
Z_d=complex(30,6);        //load impedance
R=real(Z_d);
R_line=real(Z_line);
//source voltages
V_ab=complex(1000*cos(%pi/6),1000*sin(%pi/6));
V_bc=complex(1000*cos(-%pi/2),1000*sin(-%pi/2));
V_ca=complex(1000*cos(5*%pi/6),1000*sin(5*%pi/6));
//choosing A phase of wye-equivalent circuit
V_an=V_ab/(sqrt(3)*complex(cos(%pi/6),sin(%pi/6)));
Z_Y=Z_d/3;
I_aA=V_an/(Z_line+Z_Y);        //line current
I_aA_rms=abs(I_aA)/sqrt(2);
V_An=I_aA*Z_Y;        //line to neutral voltage
V_AB=V_An*sqrt(3)*complex(cos(%pi/6),sin(%pi/6));        //line to line voltage at the load
I_AB=V_AB/Z_d;        //current through phase AB
I_AB_rms=abs(I_AB)/sqrt(2);        //rms value
P_AB=I_AB_rms^2*R;        //power delivered to phase AB
//power delivered in other two phases is same
P=3*P_AB;        //total power
P_A=I_aA_rms^2*R_line;        //power lost in line A
```

```
//power lost in other two lines is same
P_line=3*P_A;
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp('LINE CURRENTS')
disp(I_aA,'IaA=')
disp(I_aA*complex(cos(-2*%pi/3),sin(-2*%pi/3)),'IbB=')
disp(I_aA*complex(cos(2*%pi/3),sin(2*%pi/3)),'IcC=')
disp('LINE-LINE VOLTAGES')
disp(V_AB,'VAB=')
disp(V_AB*complex(cos(-2*%pi/3),sin(-2*%pi/3)),'VBB=')
disp(V_AB*complex(cos(2*%pi/3),sin(2*%pi/3)),'VCC=')
disp(P,'power delivered to load in watts')
disp(P_line,'total power dissipated in the line')
```

Scilab code Exa 5.14 `example 14`

```
clc
//ex5.14
V_1=10^3*2.2*sqrt(2)*complex(cos(0),sin(0));
V_2=10^3*2*sqrt(2)*complex(cos(-%pi/18),sin(-%pi/18));
//writing matrix form of mesh current equaions obtained by KVL
Z=[5+3*%i+50*complex(cos(-%pi/18),sin(-%pi/18)),-50*complex(cos(-%pi/18),sin(-%pi/18));-50*complex(cos(-%pi/18),sin(-%pi/18)),4+%i+50*complex(cos(-%pi/18),sin(-%pi/18))];      //coefficient matrix
V=[2200*sqrt(2);-2000*sqrt(2)*complex(cos(-%pi/18),sin(-%pi/18))];      //voltage matrix
I=Z\V;      //current matrix
S_1=(1/2)*V_1*((I(1,:))');      //complex power
```

```
P_1=real(S_1);        //power
Q_1=imag(S_1);        //reactive power
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp(P_1,'real power supplied by V1 in watts')
disp(Q_1,'reactive power supplied by V1 in VARs')
```

Chapter 6

Frequency response bode plots and resonance

Scilab code Exa 6.1 `example 1`

```
clc
//ex 6.1
// given V_in(t)=2*cos(2000*%pi*t+A), A=40*%pi/180
w=2000*%pi;      //omega
f=w/(2*%pi);      //frequency
A=40*%pi/180;      //40 degrees in radians
//equation of straight line of H_magnitude vs f is x+1000*y-4000=0
H_max=(4000-f)/1000;      //magnitude of H(traansfer function)
//equation of straight line of H_phase angle vs f is 6000*y=%pi*x (phase angle in radians)
H_phi=%pi*f/6000;      //phase angle of H
H=H_max*complex(cos(H_phi),sin(H_phi));
V_in=2*complex(cos(A),sin(A));      //input voltage phasor
V_out=H*V_in;      //output voltage phasor
V_out_R=real(V_out);      //real part
V_out_I=imag(V_out);      //imaginary part
```

```
V_out_max=sqrt((V_out_R^2)+(V_out_I^2));        //peak value
V_out_phi=atan(V_out_I/V_out_R);
disp(V_out_max,'peak value of Vout in volts')
disp(V_out_phi*180/%pi,'phase angle of Vout in degrees')
disp(f,'with frequency equal to')
```

Scilab code Exa 6.2 `example 2`

```
clc
//ex6.2
//given V_in(t)=3+2*cos(2000*%pi*t)+cos(4000*%pi*t-A), A=70*%pi/180
//the three parts of V_in(t) are V_in_1=3, V_in_2=2*cos(2000*%pi*t),V_in_3=cos(4000*%pi*t-A)

//first component V_1
V_in_1=3;
f_1=0;        //as omega is zero
//equation of straight line of H_magnitude vs f is x+1000*y-4000=0
H_1_max=(4000-f_1)/1000;        //magnitude of H(traansfer function)
//equation of straight line of H_phase angle vs f is 6000*y=%pi*x (phase angle in radians)
H_1_phi=%pi*f_1/6000;        //phase angle of H
H_1=H_1_max*complex(cos(H_1_phi),sin(H_1_phi));
V_out_1=H_1*V_in_1;
V_out_1_R=real(V_out_1);        //real part
V_out_1_I=imag(V_out_1);        //imaginary part
V_out_1_max=sqrt((V_out_1_R^2)+(V_out_1_I^2));        //peak value
V_out_1_phi=atan(V_out_1_I/V_out_1_R);        //phase angle
```

```

//second component V_in_2
V_in_2=2*complex(cos(0),sin(0));      //V_in_2 phasor
w=2000*%pi;      //omega
f_2=w/(2*%pi);      //frequency
//equation of straight line of H_magnitude vs f is x+1000*y-4000=0
H_2_max=(4000-f_2)/1000;      //magnitude of H(traansfer function)
//equation of straight line of H_phase angle vs f is 6000*y=%pi*x (phase angle in radians)
H_2_phi=%pi*f_2/6000;      //phase angle of H
H_2=H_2_max*complex(cos(H_2_phi),sin(H_2_phi));
V_out_2=H_2*V_in_2;
V_out_2_R=real(V_out_2);      //real part
V_out_2_I=imag(V_out_2);      //imaginary part
V_out_2_max=sqrt((V_out_2_R^2)+(V_out_2_I^2));      //peak value
V_out_2_phi=atan(V_out_2_I/V_out_2_R);      //phase angle

//third component
A=-70*%pi/180;      //-70 degrees in radians
V_in_3=complex(cos(A),sin(A));      //V_in_3 phasor
w=4000*%pi;      //omega
f_3=w/(2*%pi);      //frequency
//equation of straight line of H_magnitude vs f is x+1000*y-4000=0
H_3_max=(4000-f_3)/1000;      //magnitude of H(traansfer function)
//equation of straight line of H_phase angle vs f is 6000*y=%pi*x (phase angle in radians)
H_3_phi=%pi*f_3/6000;      //phase angle of H
H_3=H_3_max*complex(cos(H_3_phi),sin(H_3_phi));
V_out_3=H_3*V_in_3;
V_out_3_R=real(V_out_3);      //real part
V_out_3_I=imag(V_out_3);      //imaginary part
```

```
V_out_3_max=sqrt((V_out_3_R^2)+(V_out_3_I^2));  //peak value
V_out_3_phi=atan(V_out_3_I/V_out_3_R);      //phase angle

disp('Output voltage is Vout1+Vout2+Vout3 where')
disp('')
disp('FOR Vout1:')
disp(V_out_1_max,'peak value in volts')
disp(V_out_1_phi*180/%pi,'phase angle in degrees')
disp(f_1,'with frequency in hertz')
disp('')
disp('FOR Vout2:')
disp(V_out_2_max,'peak value in volts')
disp(V_out_2_phi*180/%pi,'phase angle in degrees')
disp(f_2,'with frequency in hertz')
disp('')
disp('FOR Vout3:')
disp(V_out_3_max,'peak value in volts')
disp(V_out_3_phi*180/%pi,'phase angle in degrees')
disp(f_3,'with frequency in hertz')
```

Scilab code Exa 6.3 `example 3`

```
clc
//ex6.3
R=1000/(2*%pi);      //resistance
C=10*10^-6;      //capacitance
f_B=1/(2*%pi*R*C);      //half-power frequency
//the three parts of V_in are V_1=5*cos(20*%pi*t)+5*cos(200*%pi*t)+5*cos(2000*%pi*t)

//first component V_in_1
V_in_1=5*complex(cos(0),sin(0));      //V_in_1 phasor
```

```
w_1=20*%pi;      //omega
f_1=w_1/(2*%pi);      //frequency
H_1=1/(1+%i*(f_1/f_B));      //transfer function
V_out_1=H_1*V_in_1;
V_out_1_R=real(V_out_1);      //real part
V_out_1_I=imag(V_out_1);      //imaginary part
V_out_1_max=sqrt((V_out_1_R^2)+(V_out_1_I^2));      //peak value
V_out_1_phi=atan(V_out_1_I/V_out_1_R);      //phase angle

//second component V_in_2
V_in_2=5*complex(cos(0),sin(0));      //V_in_2 phasor
w_2=200*%pi;      //omega
f_2=w_2/(2*%pi);      //frequency
H_2=1/(1+%i*(f_2/f_B));      //transfer function
V_out_2=H_2*V_in_2;
V_out_2_R=real(V_out_2);      //real part
V_out_2_I=imag(V_out_2);      //imaginary part
V_out_2_max=sqrt((V_out_2_R^2)+(V_out_2_I^2));      //peak value
V_out_2_phi=atan(V_out_2_I/V_out_2_R);      //phase angle

//third component V_in_3
V_in_3=5*complex(cos(0),sin(0));      //V_in_3 phasor
w_3=2000*%pi;      //omega
f_3=w_3/(2*%pi);      //frequency
H_3=1/(1+%i*(f_3/f_B));      //transfer function
V_out_3=H_3*V_in_3;
V_out_3_R=real(V_out_3);      //real part
V_out_3_I=imag(V_out_3);      //imaginary part
V_out_3_max=sqrt((V_out_3_R^2)+(V_out_3_I^2));      //peak value
V_out_3_phi=atan(V_out_3_I/V_out_3_R);      //phase angle
```

```

printf(" All the values in the textbook are approximated, hence the values in this code differ from those of Textbook")
disp('Output voltage is Vout1+Vout2+Vout3 where')
disp('')
disp('FOR Vout1:')
disp(V_out_1_max,'peak value in volts')
disp(V_out_1_phi*180/%pi,'phase angle in degrees')
disp(f_1,'with frequency in hertz')
disp('')
disp('FOR Vout2:')
disp(V_out_2_max,'peak value in volts')
disp(V_out_2_phi*180/%pi,'phase angle in degrees')
disp(f_2,'with frequency in hertz')
disp('')
disp('FOR Vout3:')
disp(V_out_3_max,'peak value in volts')
disp(V_out_3_phi*180/%pi,'phase angle in degrees')
disp(f_3,'with frequency in hertz')
//we can observe that there is a clear discrimination in output signals based on frequencies i.e, lesser the frequency lesser the effect.
```

Scilab code Exa 6.4 `example 4`

```
clc
//ex6.4
H_max=-30;      //transfer function magnitude
f=60;
m=20;      //low-frequency asymptote slope rate in db/decade
//f_B must be K higher than f where K is
K=abs(H_max)/m;
```

```
//(base 10)log(f_B/60)=1.5 ==>
f_B=60*10^1.5;
printf(" All the values in the textbook are
   approximated, hence the values in this code
   differ from those of Textbook")
disp(f_B,'Break frequency in Hz')
```

Scilab code Exa 6.5 `example 5`

```
clc
//ex6.5
V_s=1*complex(cos(0),sin(0));
L=159.2*10^-3;
R=100;
C=0.1592*10^-6;
f_o=1/(2*%pi*sqrt(L*C));      //resonant frequency
Q_s=2*%pi*f_o*L/R;      //quality factor
B=f_o/Q_s;      //Bandwidth
//Approximate half-power frequencies are
f_H=f_o+(B/2);
f_L=f_o-(B/2);
//At resonance
Z_L=%i*2*%pi*f_o*L;      //impedance of inductance
Z_C=-%i/(2*%pi*f_o*C);      //impedance of
   capacitance
Z_s=R+Z_L+Z_C;
I=V_s/Z_s;      //phasor current
//voltages across diffrent elements are
//for resistance
V_R=R*I;
V_R_R=real(V_R);      //real part
V_R_I=imag(V_R);      //imaginary part
V_R_max=sqrt((V_R_R^2)+(V_R_I^2));      //peak value
V_R_phi=atan(V_R_I/V_R_R);      //phase angle
//for inductance
```

```
V_L=Z_L*I;
V_L_R=real(V_L);       //real part
V_L_I=imag(V_L);       //imaginary part
V_L_max=sqrt((V_L_R^2)+(V_L_I^2));       //peak value
//Z_L is pure imaginary ==> V_L is pure imaginary which means V_L_phi can be +or- %pi/2
if ((V_L/%i)==abs(V_L)) then
    V_L_phi=%pi/2
elseif ((V_L/%i)==-abs(V_L)) then
    V_L_phi=-%pi/2
end

//for capacitance
V_C=Z_C*I;
V_C_R=real(V_C);       //real part
V_C_I=imag(V_C);       //imaginary part
V_C_max=sqrt((V_C_R^2)+(V_C_I^2));       //peak value
//Z_C is pure imaginary ==> V_C is pure imaginary which means V_C_phi can be +or- %pi/2
if ((V_C/%i)==abs(V_C)) then
    V_C_phi=%pi/2
elseif ((V_C/%i)==-abs(V_C)) then
    V_C_phi=-%pi/2
end

disp('Phasor voltage across Resistance')
disp(V_R_max,'peak value in volts')
disp(V_R_phi*180/%pi,'phase angle in degrees')
disp('')
disp('Phasor voltage across Inductance')
disp(V_L_max,'peak value in volts')
disp(V_L_phi*180/%pi,'phase angle in degrees')
disp('')
disp('Phasor voltage across Capacitance')
disp(V_C_max,'peak value in volts')
disp(V_C_phi*180/%pi,'phase angle in degrees')
disp('Phasor diagram cannot be drawn here')
```

Scilab code Exa 6.6 `example 6`

```
clc
//ex6.6
R=10*10^3;
f_o=1*10^6;
B=100*10^3;
I=10^-3*complex(cos(0),sin(0));
Q_p=f_o/B;        //quality factor
L=R/(2*%pi*f_o*Q_p);
C=Q_p/(2*%pi*f_o*R);
//At resonance
V_out=I*R;
Z_L=%i*2*%pi*f_o*L;
Z_C=-%i/(2*%pi*f_o*C);

//across resistance
I_R=V_out/R;
I_R_R=real(I_R);        //real part
I_R_I=imag(I_R);        //imaginary part
I_R_max=sqrt((I_R_R^2)+(I_R_I^2));        //peak value
I_R_phi=atan(I_R_I/I_R_R);        //phase angle

//across inductance
I_L=V_out/Z_L;
I_L_R=real(I_L);        //real part
I_L_I=imag(I_L);        //imaginary part
I_L_max=sqrt((I_L_R^2)+(I_L_I^2));        //peak value
//Z_L is pure imaginary ==> V_L is pure imaginary which means V_L_phi can be +or- %pi/2
if ((I_L/%i)==abs(I_L)) then
    I_L_phi=%pi/2
elseif ((I_L/%i)==-abs(I_L)) then
    I_L_phi=-%pi/2
```

```
end

//across capacitor
I_C=V_out/Z_C;
I_C_R=real(I_C);        //real part
I_C_I=imag(I_C);        //imaginary part
I_C_max=sqrt((I_C_R^2)+(I_C_I^2));        //peak value
//Z_C is pure imaginary ==> V_C is pure imaginary which means V_C_phi can be +or- %pi/2
if ((I_C/%i)==abs(I_C)) then
    I_C_phi=%pi/2
elseif ((I_C/%i)==-abs(I_C)) then
    I_C_phi=-%pi/2
end

disp('Current phasor across Resistance')
disp(I_R_max,'peak value in amperes')
disp(I_R_phi*180/%pi,'phase angle in degrees')
disp('')
disp('Current phasor across Inductance')
disp(I_L_max,'peak value in amperes')
disp(I_L_phi*180/%pi,'phase angle in degrees')
disp('')
disp('current phasor across capacitance')
disp(I_C_max,'peak value in amperes')
disp(I_C_phi*180/%pi,'phase angle in degrees')
disp('Phasor diagram cannot be drawn here')
```

Scilab code Exa 6.7 `example 7`

```
clc
//ex6.7
//We need a high-pass filter
L=50*10^-3;
//for the transfer function to be approximately
```

```
   constant in passband area(from graph given in the
    text), we choose
Q_s=1;
f_o=1*10^3;
C=1/(((2*%pi)^2)*f_o^2*L);
R=2*%pi*f_o*L/Q_s;
printf(" All the values in the textbook are
   approximated, hence the values in this code
   differ from those of Textbook")
disp('')
disp('The required second order circuit
   configuration is')
disp(L*10^3,'Inductance in KH')
disp(C*10^6,'Capacitance in mF(micro Farads)')
disp(R,'Resistance in ohms')
```

Chapter 7

Logic circuits

Scilab code Exa 7.1 `example 1`

```
clc
//ex7.1
N=343;        //decimal integer
N2=dec2bin(N);        //binary equivalent of N
disp(N2,'Binary equivalent of 343 is')
```

Scilab code Exa 7.2 `example 2`

```
clc
//ex7.2
N=0.392;        //decimal
DP=N;       //decimal part(no integer part)
i=1;
x=1;
//Each decimal digit is stored in D(x)
while (x<=9)
DP=DP*2;
D(x)= floor (DP);
```

```
x=x+1;
DP= modulo (DP ,1);
end
DP=0;
for j=1: length (D)
//bits of decimal part are multiplied with their position values and adding them
  DP=DP+(10^(-1*j)*D(j));
end
disp(DP,'Binary form of 0.392 is')
```

Scilab code Exa 7.3 `example 3`

```
clc
//ex7.3
N=343.392;
//convert the integer and decimal parts into binary form separately
B_1='101010111';      //for 343 from ex7.1
B_2='0.011001';       //for 0.392 from ex7.2
//combining these two
B='101010111.011001';       //for N, given number
disp(B,'binary form of 343.392')
```

Scilab code Exa 7.4 `example 4`

```
//ex7.4
N_1=1000.111;
N_2=1100.011;
//Adding these two according to the rules of binary addition in fig7.6, we get
disp("The result of addition of given two binary numbers is")
```

```
disp("10101.010")
```

Scilab code Exa 7.5 `example 5`

```
clc
//ex7.5
//Given 317.2 (octal)and F3A.2 (hexadecimal)
//From table 7.1 in text, corresponding octal forms of 3,1,7 and 2 are 011,001,111 and 010
disp('The binary representation of 317.2(octal) is ')
disp('011001111.010')
disp('')
//From table 7.1 in text, corresponding hexadecimal forms of F,3,A and 2 are 1111,0011,1010 and 0010
disp('The binary representation of F3A.2(hexadecimal) is ')
disp('111100111010.0010')
```

Scilab code Exa 7.6 `example 6`

```
clc
//ex7.6
//Given 11110110.1(binary)
//Working outward from the binary point, we form three-bit groups ==> 11110110.1=011 110 110. 100(we have appended leading and trailing zeros so that each group contains 3 bits)
//And the corresponding numbers for 011,110,110 and 100 in octal system are 3,6,6 and 4
disp('The octal representation of 11110110.1(binary) is 366.4')
```

```
//Now we form four-bit groups appending leading and trailing zeros as needed ==> 11110110.1=1111 0110. 1000
//The corresponding numbers for 1111,0110 and 1000 in hexadecimal system are F,6 and 8
disp('The hexadecimal representation of 11110110.1(binary) is F6.8')
```

Chapter 9

Computer based instrumentation diodes

Scilab code Exa 9.1 `example 1`

```
clc
//ex9.1
P=0.1;        //system sensitivity change percent
R_th_U=15*10^3;       //thevenin resistance upper limit
R_th_L=5*10^3;       //thevenin resistance lower limit
//The required inequality is V_sensor*R_in/(R_th_U+R_in)>=(1-P/100)*V_sensor*R_in/(R_th_L+R_in), cancelling same terms on both sides of inequality and calculating R_in by taking equality we'll get minimum value of R_in ==>R_th_L+R_in=(1-P/100)*(R_th_U+R_in) which gives
R_in=(((1-P/100)*R_th_U)-R_th_L)*100/P;
disp(R_in/1000,'The minimum value of Rin required in Kilo-ohms')
```

Chapter 10

Diodes

Scilab code Exa 10.1 `example 1`

```
clc
//ex10.1
V_ss=2;
R=1*10^3;
V_D=[0:0.001:2];
plot(V_D,10^3*(V_ss-V_D)/R)
xtitle('load line plot','voltage in volts','current in milli-amperes')      //milli -10^-3
//we use the equation V_ss=R*i_D+V_D
//at point B
i_D=V_ss/R;       //as V_D=0
//at point A
V_D=V_ss;       //as i_D=0
//now we see intersection of load line with characteristic and we get following at operating point
V_DQ=0.7;       //voltage
I_DQ=1.3*10^-3;       //current
//diode characteristic cannot be plotted
disp(V_DQ,'diode voltage at operating point in volts')
```

```
disp(I_DQ*10^3,'current at opeating point in milli-amperes')        //milli-10^-3
```

Scilab code Exa 10.2 `example 2`

```
clc
//ex10.2
V_ss=10;
R=10*10^3;
V_D=[0:0.001:2];
plot(V_D,10^3*(V_ss-V_D)/R)
xtitle('load line plot','voltage in volts','current in milli-amperes')        //milli-10^-3
//we use the equation V_ss=R*i_D+V_D
//at point C
i_D=V_ss/R;        //as V_D=0
//now if we take i_D=0, we get V_D=10 which plots at a point far off the page
//so we take the value on the right-hand edge of V-axis i.e.,V_D=2
//at point D
V_D=2;
i_D=(V_ss-V_D)/R;
//from the intersection of load line with characteristic
V_DQ=0.68;
I_DQ=0.93*10^-3;
//diode characteristic cannot be plotted
disp(V_DQ,'diode voltage at operating point in volts')
disp(I_DQ*10^3,'current at opeating point in milli-amperes')        //milli-10^-3
```

Scilab code Exa 10.3 `example 3`

```
clc
//ex10.3
R=1*10^3;
//diode characteristic cannot be plotted
//case a)V_ss=15
V_ss=15;
V_D=[-15:0.001:0];
//from the intersection of load line and diode characteristic
V_o=10;
disp(V_o,'output voltage for Vss=15 in volts')
//case b)V_ss=20
V_ss=20;
V_D=[-20:0.001:0];
//from the intersection of load line and diode characteristic
V_o=10.5;
disp(V_o,'output voltage for Vss=20 in volts')
```

Scilab code Exa 10.4 `example 4`

```

clc
//ex10.4
V_ss=24;
R=1.2*10^3;
R_L=6*10^3;
//by grouping linear elements together on left side of diode
V_T=V_ss*R_L/(R+R_L);       //thevenin voltage
//zeroing sources
R_T=1/((1/R)+(1/R_L));       //thevenin resistance
//load-line equation is V_T+R_T*i_D+V_D=0
```

```
//locating the operating point
V_D=-10;
V_L=-V_D;       //load voltage
I_s=(V_ss-V_L)/R;       //source current
//diode characteristic cannot be plotted
printf(" All the values in the textbook are
   approximated hence the values in this code differ
    from those of Textbook")
disp(V_L,'load voltage in volts')
disp(I_s,'source current in amperes')       //milli
   -10^-3
```

Scilab code Exa 10.5 `example 5`

```
clc
//ex10.5
V_1=10;
V_2=3;
R_1=4*10^3;
R_2=6*10^3;
//1)analysis by assuming D1 off and D2 on
I_D_2=V_2/R_2;       //ohm's law
//applying KVL
V_D_1=7;       //contradiction to 'D1 is off'
//this assumption is not correct

//2)analysis by assuming D1 on and D2 off
I_D_1=V_1/R_1;       //ohm's law
//applying KVL
V_D_2=-V_1+V_2+I_D_1*R_1;
//we get V_D_2 which is consistent
disp('correct assumption is D2 off and D1 on')
```

Scilab code Exa 10.7 `example 7`

```
clc
//ex10.7
V_1=3;
R_1=20;
//As given voltage source results in forward bias, we assume operating point is on line segment A
//replacing diode with the equivalent circuit
V_2=0.6;
R_2=10;
i_D=(V_1-V_2)/(R_1+R_2);        //KVL around the circuit
disp(i_D*10^3,'current in the circuit in milli-amperes')        //milli-10^-3
```

Chapter 11

Amplifiers Specifications and eternal characterstics

Scilab code Exa 11.1 `example 1`

```
clc
//ex11.1
V_s=1*10^-3;
R_s=1*10^6;
A_voc=10^4;      //open-circuit voltage gain
R_i=2*10^6;      //input resistance
R_o=2;      //output resistance
R_L=8;      //load resistance
V_i=V_s*(R_i/(R_i+R_s));      //input voltage(voltage-divider principle)
V_vcs=A_voc*V_i;      //voltage controlled source voltage
V_o=V_vcs*(R_L/(R_L+R_o));      //output voltage(voltage-divider principle)
A_v=V_o/V_i;
A_vs=V_o/V_s;
A_i=A_v*R_i/R_L;      //current gain
G=A_v*A_i;      //power gain
printf(" All the values in the textbook are
```

```
approximated, hence the values in this code differ from those of Textbook")
disp(A_v,'Voltage gain Av')
disp(A_vs,'Voltage gain Avs')
disp(A_i,'Current gain')
disp(G,'Power gain')
```

Scilab code Exa 11.2 `example 2`

```
clc
//ex11.2
R_i_1=10^6;
R_o_1=500;
R_i_2=1500;
R_o_2=100;
R_L=100;
A_voc_1=200;
A_voc_2=100;
//voltage gain of the first stage...A_v_1=(V_o_1/V_i_1)=(V_i_2/V_i_2)=A_voc_1(R_i_2/(R_i_2+R_o_1))
A_v_1=A_voc_1*(R_i_2/(R_i_2+R_o_1));
A_v_2=A_voc_2*(R_L/(R_L+R_o_2));
A_i_1=A_v_1*R_i_1/R_i_2;
A_i_2=A_v_2*R_i_2/R_L;
A_i=A_i_1*A_i_2;
G_1=A_v_1*A_i_1;
G_2=A_v_2*A_i_2;
G=G_1*G_2;
disp(A_i_1,'Current gain of first stage')
disp(A_i_2,'Current gain of second stage')
disp(A_v_1,'Voltage gain of first stage')
disp(A_v_2,'Voltage gain of second stage')
disp(G_1,'Power gain of first stage')
disp(G_2,'Power gain of second stage')
disp(G,'Overall power gain')
```

Scilab code Exa 11.3 `example 3`

```
clc
//ex11.3
R_i_1=10^6;
R_o_1=500;
R_i_2=1500;
R_o_2=100;
R_L=100;
A_voc_1=200;
A_voc_2=100;
A_v_1=A_voc_1*(R_i_2/(R_i_2+R_o_1));       //Voltage
    gain of first stage
A_v_2=A_voc_2;       //Voltage gain of second stage
    with open-circuit load
A_voc=A_v_1*A_v_2;       //overall open-circuit
    voltage gain
R_i=R_i_1;       //input resistance of cascading
    amplifier
R_o=R_o_2;       //output resistance
disp('Hence the simplified model for the cascade is
    with an:')
disp(R_i,'Input resistance in ohms')
disp(R_o,'Input resistance in ohms')
disp(A_voc,'Overall open-circuit voltage gain')
```

Scilab code Exa 11.4 `example 4`

```
clc
//ex11.4
V_AA=15;
```

```
V_BB=15;
V_i=1*10^-3;
I_A=1;
I_B=0.5;
R_L=8;
R_o=2;
R_i=100*10^3;
A_voc=10^4;
P_i=V_i^2/R_i;
V_o=A_voc*V_i*(R_L/(R_L+R_o));
P_o=V_o^2/R_L;
P_s=V_AA*I_A+V_BB*I_B;
P_d=P_s+P_i-P_o;
n=P_o*100/P_s;
printf(" All the values in the textbook are approximated, hence the values in this code differ from those of Textbook")
disp(P_i*10^12,'Input power in picowatts')
disp(P_o,'Output power in watts')
disp(P_s,'Supply power in watts')
disp(P_d,'Dissipated power in watts')
disp(n,'Efficiency of the amplifier')
```

Scilab code Exa 11.5 `example 5`

```
clc
//ex11.5
R_i=1*10^3;
R_o=100;
A_voc=100;
//I_i=V_i/R_i, I_osc=A_voc*V_i/R_o    from these two we get A_isc=(i_osc/I_i)=(A_voc(R_i/R_o))
A_isc=A_voc*(R_i/R_o);
disp('The resulting current-amplifier is with an:')

```

```
disp(R_i,'input resitance in ohms')
disp(R_o,'output resistance in ohms')
disp(A_isc,'and a short-cut current gain of:')
```

Scilab code Exa 11.6 example 6

```
clc
//ex11.6
R_i=1*10^3;
R_o=100;
A_voc=100;
//i_osc=A_voc*V_i/R_o and G_msc=i_osc/V_i gives G_msc=A_voc/R_o
G_msc=A_voc/R_o;
disp('The resulting transconductance model is with an:')

disp(R_i,'input resitance in ohms')
disp(R_o,'output resistance in ohms')
disp(G_msc,'and transconductance in siemens')
```

Scilab code Exa 11.7 example 7

```
clc
//ex11.7
R_i=1*10^3;
R_o=100;
A_voc=100;
//V_ooc=A_voc*V_i and I_i=V_i/R_i gives R_moc=V_ooc/I_i
R_moc=A_voc*R_i;
disp('The resulting transconductance model is with an:')
```

```

disp(R_i,'input resitance in ohms')
disp(R_o,'output resistance in ohms')
disp(R_moc,'and transresistance in ohms')
```

Scilab code Exa 11.8 `example 8`

```
clc
//ex11.8
V_i=complex(0.1*cos(-%pi/6),0.1*sin(-%pi/6));
V_o=complex(10*cos(%pi/12),10*sin(%pi/12));
A_v=V_o/V_i;
A_v_max=sqrt((real(A_v)^2)+(imag(A_v)^2))
phi=atan(imag(A_v)/real(A_v));
printf(" All the values in the textbook are
   approximated, hence the values in this code
   differ from those of Textbook")
disp('The complex voltage gain is with')
disp(A_v_max,'a peak value of')
disp(phi,'a phase angle in degrees')
disp(20*log(A_v_max)/2.30258,'and the decibel gain
   is')       //2.30258 is for base 10
```

Scilab code Exa 11.9 `example 9`

```
clc
//ex11.9
t=[0:0.000001:0.002];
V_i=3*cos(2000*%pi*t)-2*cos(6000*%pi*t);
//let A_1000 and A_3000 be the gains
A_1000_peak=10;
A_1000_phi=0;
A_3000_peak=2.5;
```

```
A_3000_phi=0;
//multiplying by respective gains
V_o=A_1000_peak*3*cos(2000*%pi*t+A_1000_phi)-A_3000_peak*2*cos(6000*%pi*t+A_3000_phi);
subplot(121)
xtitle('Input-voltage vs time','time in ms','Internal-voltage in volts')
plot(t*10^3,V_i)
subplot(122)
xtitle('Output-voltage vs time','time in ms','Output voltage in volts')
plot(t*10^3,V_o)
```

Scilab code Exa 11.10 `example 10`

```
clc
//ex11.10
t=[0:0.000001:0.002];
V_i=3*cos(2000*%pi*t)-2*cos(6000*%pi*t);
//for A
A_1000_A_peak=10;
A_1000_A_phi=0;
A_3000_A_peak=10;
A_3000_A_phi=0;
V_o_A=A_1000_A_peak*3*cos(2000*%pi*t+A_1000_A_phi)-A_3000_A_peak*2*cos(6000*%pi*t+A_3000_A_phi);
//for B
A_1000_B_peak=10;
A_1000_B_phi=-%pi/4;
A_3000_B_peak=10;
A_3000_B_phi=-3*%pi/4;
V_o_B=A_1000_B_peak*3*cos(2000*%pi*t+A_1000_B_phi)-A_3000_B_peak*2*cos(6000*%pi*t+A_3000_B_phi);
//for C
A_1000_C_peak=10;
```

```
A_1000_C_phi=-%pi/4;
A_3000_C_peak=10;
A_3000_C_phi=-%pi/4;
V_o_C=A_1000_C_peak*3*cos(2000*%pi*t+A_1000_C_phi)-A_3000_C_peak*2*cos(6000*%pi*t+A_3000_C_phi);
disp('VoA(t)=30cos(2000%pit)-10cos(6000%pit)')
disp('VoB(t)=30cos(2000%pit-%pi/4)-10cos(6000%pit-3%pi/4)')
disp('VoC(t)=30cos(2000%pit-%pi/4)-10cos(6000%pit-%pi/4)')
subplot(221)
xtitle('Output-voltage vs time for A','time in ms','Output-voltage for A in volts')
plot(t*10^3,V_o_A)
subplot(222)
xtitle('Output-voltage vs time for B','time in ms','Output voltage for B in volts')
plot(t*10^3,V_o_B)
subplot(223)
xtitle('Output-voltage vs time for C','time in ms','Output voltage for C in volts')
plot(t*10^3,V_o_C)
```

Scilab code Exa 11.11 `example 11`

```
clc
//ex11.11
A_d=1000;       //differential gain
V_d_peak=1*10^-3;       //peak value of differential input signal
V_o_peak=A_d*V_d_peak;       //peak output signal
V_cm=100;
V_o_cm=0.01*V_o_peak;       //common mode contribution is 1% or less
A_cm=V_o_cm/V_cm;       //common mode gain
```

```
CMRR=20*log(A_d/A_cm)/2.30258;
printf(" All the values in the textbook are approximated, hence the values in this code differ from those of Textbook")
disp(CMRR,'The minimum CMRR is')
```

Scilab code Exa 11.12 `example 12`

```
clc
//initialisation of variables
Rin= 1 //Mohms
Rs1= 100 //kohms
Rs2= 100 //kohms
Ioff= 84 //Amperes
Voff= 5 //mV
//CALCULARIONS
Vioff= Rin*Ioff*10^-3*(Rs1+Rs2)/(2*(Rin+10^-3*(Rs1+Rs2)))
Vvoff= Voff*Rin/(Rin+10^-3*(Rs1+Rs2))
//RESULTS
printf ('Vioff = %.f mV ',Vioff)
printf ('\n Vvoff = %.2f mV ',Vvoff)
```

Chapter 12

Field effect transistors

Scilab code Exa 12.1 `example 1`

```
clc
//initialisation of variables
K= 2
VGS1= 5 //V
VGS2= 4 //V
VGS3= 3 //V
VGS4= 2 //V
//CALCULATIONS
id1= K*(VGS1-2)^2
id2= K*(VGS2-2)^2
id3= K*(VGS3-2)^2
id4= K*(VGS4-2)^2
//RESULTS
printf ('iD = %.f V ',id1)
printf ('\n iD = %.f V ',id2)
printf ('\n iD = %.f V ',id3)
printf ('\n iD = %.f V ',id4)
```

Scilab code Exa 12.2 `example 2`

```
clc
//initialisation of variables
KP= 50 //uA/V62
Vto= 2 //V
L= 10 //um
W= 400 //um
Vdd= 20 //mV
R2= 1 //kohms
R1= 3 //ohms
Rd= 11.5 //Mohms
Rs= 1 //kohms
V= 4 //mV
//CALCULATIONS
K= W*KP/(2*L*10^3)
Vg= Vdd*R2/(R1+R2)
clc
x=poly(0,"x")
vec=roots(x^2-3.630*x+2.148)
VGSQ= vec(2)
IDQ= K*(VGSQ-Vto)^2
VDSQ= Vdd+V+L-(Rd+Rs)*IDQ
//RESULTS
printf ('VDSQ = %.1f V ',VDSQ)
```

Scilab code Exa 12.3 example 3

```
clc
//initialisation of variables
VGSQ= 3.5 //V
VDSQ= 10 //V
id1= 10.7 //mA
id2= 4.7 //mA
dvgs= 1 //V
id3= 8 //mA
id4= 6.7 //mA
```

```
vds1= 14 //V
vds2= 4 //V
//CALCULATIONS
gm= (id1-id2)/dvgs
rd= (vds1-vds2)*10^3/(id3-id4)
//RESULTS
printf ('rd = %.1e ohms',rd)
```

Scilab code Exa 12.5 `example 5`

```
clc
//initialisation of variables
RL= 1 //kohms
R1= 2 //Mohms
R2= 2 //Mohms
KP= 50 //uA/V^2
L= 2 //um
W= 160 //um
Vto= 1 //V
IDQ= 10 //mA
VG= 7.5 //V
//CALCULATIONS
K= W*KP/(2*L*10^3)
VGSQ= sqrt(IDQ/K)+Vto
VS= VG-VGSQ
RS= VS*10^3/IDQ
gm= sqrt(2*KP/10^3)*sqrt(W/L)*sqrt(IDQ)
RL1= 1/(1/(RS)+(1/(RL*10^3)))
Av= gm*RL1*10^-3/(1+gm*RL1*10^-3)
Rin= 1/((1/R1)+(1/R2))
Ro= 1/(gm*10^-3+(1/RS))
Ai= Av*Rin/RL
G= Av*Ai*10^3
//RESULTS
printf ('G = %.1f ',G)
```

Chapter 13

Bipolar junction transistors

Scilab code Exa 13.1 `example 1`

```
clc
//ex13.1
V_CE=4;        //It should be high enough so that collector base junction is reverse-biased
i_B=30*10^-6;        //base current, a value is selected from the graph
i_C=3*10^-3;        //collector current corresponding to values of i_B and V_CE
B=i_C/i_B;        //beta value
disp(B,'The value of beta B is')
```

Scilab code Exa 13.2 `example 2`

```
clc
//ex13.2
V_CC=10;
V_BB=1.6;
R_B=40*10^3;
```

```
R_C=2*10^3;
V_in_Q=0;      //Q point
V_in_max=0.4;
V_in_min=-0.4;
//the following values are found from the intersection of input loadlines with the input characteristic
i_B_Q=25*10^-3;      //for V_in_Q
i_B_max=35*10^-3;      //for V_in_max
i_B_min=15*10^-3;      //for V_in_min
//the following values are found from the intersection of output loadlines with the output characteristic
V_CE_Q=5;      //corresponding to i_B_Q
V_CE_max=7;      //corresponding to i_B_min
V_CE_min=3;      //corresponding to i_B_max
disp('graphs cannot be shown but the required values are')
disp(V_CE_max,'maximum value of V_CE')
disp(V_CE_min,'minimum value of V_CE')
disp(V_CE_Q,'Q-point value of V_CE')
```

Scilab code Exa 13.4 `example 4`

```
clc
//ex13.4
V_CC=15;
B=100;      //beta value
R_B=200*10^3;
R_C=1*10^3;
//we proceed in such a way that the required values will be displayed according to the satisfied condition of the below three cases

//a)cut-off region
```

```
V_BE=15;      //no voltage drop across R_B in cut-off state
V_CE=15;      //no voltage drop across R_C in cut-off state
i_C=0;      //no collector current flows as there is no voltage drop
i_B=0;      //no base current flows as there is no voltage drop
if(V_BE<0.5) then,      //cut-off condition
    disp(i_C,'collector current in amperes')
    disp(V_CE,'collector to emitter voltage in volts')
    end

//b)saturation region
V_BE=0.7;      //base to emitter voltage in saturation state
V_CE=0.2;      //collector to emitter voltage in saturation state
i_C=(V_CC-V_CE)/R_C;      //collector current
i_B=(V_CC-V_BE)/R_B;      //base current
if((B*i_B>i_C)&(i_B>0)) then,      //saturation state conditions
        disp(i_C,'collector current in amperes')
        disp(V_CE,'collector to emitter voltage in volts')
    end

//c)active region
V_BE=0.7;      //base to emitter voltage in active state
i_B=(V_CC-V_BE)/R_B;      //base current
i_C=B*i_B;      //collector current in active state
V_CE=V_CC-i_C*R_C;      //collector to emitter voltage
if((V_CE>0.2)&(i_B>0)) then,      //active state conditions
        disp(i_C,'collector current in amperes')
```

```
        disp(V_CE,'collector to emitter voltage in volts')
    end
```

Scilab code Exa 13.5 `example 5`

```
clc
//ex13.5
R_B=200*10^3;
R_C=1*10^3;
V_CC=15;
B=300;      //beta value
//we proceed in such a way that the required values will be displayed according to the satisfied condition of the below three cases

//a)active region
V_BE=0.7;      //base to emitter voltage in active state
i_B=(V_CC-V_BE)/R_B;      //base current
i_C=B*i_B;      //collector current in active state
V_CE=V_CC-i_C*R_C;      //collector to emitter voltage
if((V_CE>0.2)&(i_B>0)) then,      //active state conditions
        disp(i_C,'collector current in amperes')
        disp(V_CE,'collector to emitter voltage in volts')
    end

//b)saturation region
V_BE=0.7;      //base to emitter voltage in saturation state
V_CE=0.2;      //collector to emitter voltage in saturation state
```

```
i_C=(V_CC-V_CE)/R_C;      //collector current
i_B=(V_CC-V_BE)/R_B;      //base current
if((B*i_B>i_C)&(i_B>0)) then,      //saturation state conditions
        disp(i_C,'collector current in amperes')
        disp(V_CE,'collector to emitter voltage in volts')
    end

//c)cut-off region
V_BE=15;      //no voltage drop across R_B in cut-off state
V_CE=15;      //no voltage drop across R_C in cut-off state
i_C=0;      //no collector current flows as there is no voltage drop
i_B=0;      //no base current flows as there is no voltage drop
if(V_BE<0.5) then,      //cut-off condition
    disp(i_C,'collector current in amperes')
    disp(V_CE,'collector to emitter voltage in volts')
    end
```

Scilab code Exa 13.6 `example 6`

```
clc
//ex13.6
V_CC=15;
V_BB=5;
V_BE=0.7;      //assuming the device is in the active state
R_C=2*10^3;
R_E=2*10^3;
i_E=(V_BB-V_BE)/R_E;      //emitter current
```

```
printf(" All the values in the textbook are Approximated hence the values in this code differ from those of Textbook")

//a)B=100
disp('For beta B=100:')
B=100;      //beta value
i_B=i_E/(B+1);      //base current
i_C=B*i_B;      //collector current
V_CE=V_CC-i_C*R_C-i_E*R_E;      //collector to emitter voltage
disp(i_C,'collector current in amperes')
disp(V_CE,'collector to emitter voltage in volts')

//b)B=300
disp('For beta B=300:')
B=300;      //beta value
i_B=i_E/(B+1);      //base current
i_C=B*i_B;      //collector current
V_CE=V_CC-i_C*R_C-i_E*R_E;      //collector to emitter voltage
disp(i_C,'collector current in amperes')
disp(V_CE,'collector to emitter voltage in volts')
```

Scilab code Exa 13.7 `example 7`

```
clc
//ex13.7
V_CC=15;
R_1=10*10^3;
R_2=5*10^3;
R_C=1*10^3;
R_E=1*10^3;
V_BE=0.7;
R_B=1/((1/R_1)+(1/R_2));      //thevenin resistance
```

```
V_B=V_CC*R_2/(R_1+R_2);       //thevenin voltage
printf(" All the values in the textbook are
   Approximated hence the values in this code differ
    from those of Textbook")

//a)B=100
disp('For beta B=100:')
B=100;      //beta value
i_B=(V_B-V_BE)/(R_B+(B+1)*R_E);      //base current
i_C=B*i_B;      //collector current
i_E=i_B+i_C;      //emitter current
V_CE=V_CC-i_C*R_C-i_E*R_E;      //collector to
   emitter voltage
disp(i_C,'collector current in amperes')
disp(V_CE,'collector to emitter voltage in volts')

//b)B=300
disp('For beta B=300:')
B=300;      //beta value
i_B=(V_B-V_BE)/(R_B+(B+1)*R_E);      //base current
i_C=B*i_B;      //collector current
i_E=i_B+i_C;      //emitter current
V_CE=V_CC-i_C*R_C-i_E*R_E;      //collector to
   emitter voltage
disp(i_C,'collector current in amperes')
disp(V_CE,'collector to emitter voltage in volts')
```

Scilab code Exa 13.8 `example 8`

```
clc
//ex13.8
V_CC=15;
V_BE=0.7;
B=100;      //beta value
R_1=10*10^3;
```

```
R_2=5*10^3;
R_L_1=2*10^3;        //R_L is taken as R_L_1
R_C=1*10^3;
R_E=1*10^3;
V_T=26*10^-3;        //thermal voltage
//from the analysis of the previous example we have the values of i_C_Q and V_CE
i_C_Q=4.12*10^-3;
V_CE=6.72;
r_pi=(B*V_T)/i_C_Q;
R_B=1/((1/R_1)+(1/R_2));        //thevenin resistance
R_L_2=1/((1/R_L_1)+(1/R_C));        //R_L' is taken as R_L_2
A_v=-(R_L_2*B)/r_pi;        //voltage gain
A_voc=-(R_C*B)/r_pi;        //open circuit voltage gain
Z_in=1/((1/R_B)+(1/r_pi));        //input impedance
A_i=(A_v*Z_in)/R_L_1;        //current gain
G=A_i*A_v;        //power gain
Z_o=R_C        //output impedance
//assume f=1hz
f=1;
t=0:0.0005:3;
V_in=0.001*sin(2*%pi*f*t);
V_o=-(V_in*R_L_2*B)/r_pi;
subplot(121)
xtitle('Input voltage vs time','time','input voltage')
plot(t,V_in)
subplot(122)
xtitle('output voltage vs time','time','output voltage')
plot(t,V_o)
//In the graph, notice the phase inversion between input and output voltages
printf(" All the values in the textbook are Approximated hence the values in this code differ from those of Textbook")
```

```
disp(A_v,'voltage gain')
disp(A_voc,'open circuit voltage gain')
disp(Z_in,'input impedance in ohms')
disp(A_i,'current gain')
disp(G,'power gain')
disp(Z_o,'output impedance in ohms')
```

Scilab code Exa 13.9 `example 9`

```
clc
//ex_13.9
V_CC=20;
V_BE_Q=0.7;
V_T=26*10^-3;       //thermal voltage
B=200;       //beta value
R_S_1=10*10^3;       //R_S is taken as R_S_1
R_1=100*10^3;
R_2=100*10^3;
R_L_1=1*10^3;       //R_L is taken as R_L_1
R_E=2*10^3;
V_B=V_CC*R_2/(R_1+R_2);       //thevenin voltage
R_B=1/((1/R_1)+(1/R_2));       //thevenin resistance
R_L_2=1/((1/R_L_1)+(1/R_E));       //R_L' is taken as R_L_2
i_B_Q=(V_B-V_BE_Q)/(R_B+R_E*(1+B))
i_C_Q=B*i_B_Q;
i_E_Q=i_B_Q+i_C_Q;
V_CE_Q=V_CC-i_E_Q*R_E;
//we can verify that the device is in active region as we get V_CE>0.2 and i_BQ>0
r_pi=B*V_T/i_C_Q;
A_v=(1+B)*R_L_2/(r_pi+(1+B)*R_L_2);       //voltage gain
Z_it=r_pi+(1+B)*R_L_2;       //input impedance of base of transistor
```

```
Z_i=1/((1/R_B)+(1/Z_it));        //input impedance of emitter-follower
R_S_2=1/((1/R_S_1)+(1/R_1)+(1/R_2));        //R_S' is taken as R_S_2
Z_o=1/(((1+B)/(R_S_2+r_pi))+(1/R_E));        //output impedance
A_i=A_v*Z_i/R_L_1;        //current gain
G=A_v*A_i;        //power gain
printf(" All the values in the textbook are Approximated hence the values in this code differ from those of Textbook")
disp(A_v,'voltage gain')
disp(Z_i,'input impedance in ohms')
disp(A_i,'current gain')
disp(G,'power gain')
disp(Z_o,'output impedance in ohms')
```

Chapter 14

Operational Amlifiers

Scilab code Exa 14.5 `example 5`

```
clc
//initialisation of variables
ADOL= 10^5
ADOL1= 10
dc= 20
dc1= 10
f= 40 //kHz
//CALCULATIONS
ADOL2= dc*log(ADOL)
ADOL3= dc*log10(ADOL1)
f1= ADOL1*f
//RESULTS
printf ('A0CL = %.f dB ',ADOL3)
printf ('\n frequency = %.f kHz ',f1)
```

Scilab code Exa 14.6 `example 6`

```
clc
```

```
//initialisation of variables
SR= 0.5 //V/us
Vcon= 12 //V
//CALCULATIONS
f= SR*1000/(2*%pi*Vcon)
//RESULTS
printf ('full power = %.2f kHz ',f)
```

Scilab code Exa 14.7 `example 7`

```
//ex14.7
V_in=0;
I_B_max=100*10^-9;       //maximum bias current
I_os_max=40*10^-9;       //maximum offset current magnitude
V_os_max=2*10^-3;       //maximum offset voltage
R_1=10*10^3;
R_2=100*10^3;
//we approach in such a way to calculate output voltage due to each of dc sources and using superposition
//1)OFFSET–VOLTAGE
//As we place offset voltage at noninverting input
V_o_osV_max=-(1+(R_2/R_1))*(-V_os_max);
V_o_osV_min=-(1+(R_2/R_1))*V_os_max;
//2)BIAS–CURRENT SOURCES
//assuming ideal opamp conditions
V_i=0;
I_1=0;
I_2=-I_B_max;
V_o_bias_max=-R_2*I_2-R_1*I_1;
V_o_bias_min=0;       //no minimum value of I_B is specified
//3)OFFSET–CURRENT SOURCE
//by analysis as in bias−current sources
```

```
V_o_osI_max=R_2*I_os_max/2;
V_o_osI_min=-R_2*I_os_max/2;

V_o_max=V_o_osV_max+V_o_bias_max+V_o_osI_max; //maximum output volage
V_o_min=V_o_osV_min+V_o_bias_min+V_o_osI_min; //minimum output voltage
disp(V_o_max*10^3,'Maximum output voltage in milli-volts')
disp(V_o_min*10^3,'Minimum output voltage in milli-volts')
```

Chapter 15

Magnetic circuits and transformers

Scilab code Exa 15.3 `example 3`

```
clc
//ex15.3
M_r=5000;       //relative permeability
R=10*10^-2;
r=2*10^-2;
N=100;      //number of turns
//complex number 'i' is used as a symbol here
I=2*%i;      //here 'i' represents sin(200*%pi*t), not as a complex number
M_o=4*%pi*10^-7;      //permeability of free space
M=M_r*M_o;      //permeability of the core material
phi=M*N*I*r^2/(2*R);      //flux
FL=N*phi;      //flux linkages
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp('In the below two values,i represents sin(200*%pi*t)')      //t-time
disp(phi,'flux in webers')
```

```
disp(FL,'flux linkages in weber turns')
//differentiating '  ' with respect to t
disp('In the below answer, i represents cos(200*%pi*t)')
disp(FL*200*%pi,'Voltage induced in the coil in volts')
```

Scilab code Exa 15.5 `example 5`

```
clc
//ex15.5
M_r=6000;       //relative permeability
M_o=4*%pi*10^-7;       //permeability of free space
w_r=3*10^-2;       //width of rectangular cross-section
d_r=2*10^-2;       //depth of rectangular cross-section
N=500;       //number of turns of coil
B_gap=0.25;       //flux density
gap=0.5*10^-2;       //air gap
//centerline of the flux path is a square of side 6 cm
l_s=6*10^-2;       //side of square
l_core=4*l_s-gap;       //mean length of the iron core
A_core=w_r*d_r;       //cross-sectional area of the core
M_core=M_r*M_o;       //permeability of core
R!_core=l_core/(M_core*A_core);       //reluctance of the core
A_gap=(d_r+gap)*(w_r+gap);       //effective area of gap
M_gap=M_o;       //permeability of air(gap)
R!_gap=gap/(M_gap*A_gap);       //reluctance of gap
R!=R!_gap+R!_core;       //total reluctance
```

```
phi=B_gap*A_gap;      //flux
F=phi*R!;      //magnetomotive force
i=F/N;      //current
printf(" All the values in the textbook are
   approximated hence the values in this code differ
    from those of Textbook")
disp(i,'Current value in amperes')
```

Scilab code Exa 15.6 `example 6`

```
clc
//ex15.6
w_core=2*10^-2;      //width
d_core=2*10^-2;      //depth
A_core=w_core*d_core;      //area of core
M_r=1000;      //relative permeability
M_o=4*%pi*10^-7;      //permeability of free space
gap_a=1*10^-2;
gap_b=0.5*10^-2;
N=500;      //number of turns of coil
i=2;      //current in the coil
l_c=10*10^-2;      //length for center path
R!_c=l_c/(M_r*M_o*A_core);      //reluctance of
   center path
//For left side
//taking fringing ino account
A_gap_a=(w_core+gap_a)*(d_core+gap_a);      //area
   of gap a
R!_gap_a=gap_a/(M_o*A_gap_a);      //reluctance of
   gap a
l_s=10*10^-2;      //side of square
l_core_l=3*l_s-gap_a;      //mean length on left
   side
R!_core_l=l_core_l/(M_r*M_o*A_core);      //
   reluctance of core
```

```
R!_L=R!_core_l+R!_gap_a;        //total reluctance on left side
//For right side
//taking fringing ino account
A_gap_b=(w_core+gap_b)*(d_core+gap_b);        //area of gap b
R!_gap_b=gap_b/(M_o*A_gap_b);        //reluctance of gap b
l_s=10*10^-2;        //side of square
l_core_r=3*l_s-gap_b;        //mean length on right side
R!_core_r=l_core_r/(M_r*M_o*A_core);        //reluctance of core
R!_R=R!_core_r+R!_gap_b;        //total reluctance on right side
R!_T=R!_c+1/((1/R!_L)+(1/(R!_R)));        //total reluctance
phi_c=N*i/(R!_T);        //flux in the center leg of coil
//by current-division principle
phi_L=phi_c*R!_R/(R!_L+R!_R);        //left side
phi_R=phi_c*R!_L/(R!_L+R!_R);        //right side
B_L=phi_L/A_gap_a;        //flux density in gap a
B_R=phi_R/A_gap_b;        //flux density in gap b
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp(B_L,'flux density in gap a in tesla')
disp(B_R,'flux density in gap b in tesla')
```

Scilab code Exa 15.7 `example 7`

```
clc
//ex15.7
N=500;        //number of turns of coil
```

```
R!=4.6*10^6;       //reluctance of the magnetic path
   from ex15.5
L=N^2/R!;       //inductance
printf(" All the values in the textbook are
   approximated hence the values in this code differ
    from those of Textbook")
disp(L*10^3,'Inductance of the given coil in milli-
   henry')       //milli-10^-3
```

Scilab code Exa 15.8 `example 8`

```
clc
//ex15.8
R!=10^7;       //reluctance of core
N_1=100;       //turns for coil 1
N_2=200;       //turns for coil 2
L_1=N_1^2/R!;       //self-inductance of coil 1
L_2=N_2^2/R!;       //self-inductance of coil 2
//here, complex number i represents i_1 in textbook
phi_1=N_1*%i/R!;       //flux produced by i(i_1)
L_21=N_2*phi_1;       //flux linkages of coil 2 from
   current in coil 1
M=L_21/%i;       //mutual inductance
//milli-(10^-3)
disp(L_1*10^3,'self-inductance of coil 1 in milli
   henry')
disp(L_2*10^3,'self-inductance of coil 2 in milli
   henry')
disp(M*10^3,'mutual inductance of the coils in milli
    henry')
```

Scilab code Exa 15.9 `example 9`

```
clc
//ex15.9
V_s_rms=4700;       //for source
V_L_rms=220;       //load voltage
tr=V_s_rms/V_L_rms;       //turns ratio
printf(" All the values in the textbook are
   approximated hence the values in this code differ
    from those of Textbook")
disp('The required turns ratio N1/N2=')
disp(tr)
```

Scilab code Exa 15.10 `example 10`

```
clc
//ex15.10
V_1_rms=110;
R_L=10;
tr=5;       //turns ratio(N1/N2)
V_2_rms=V_1_rms/tr;       //primary and secondary
   voltage relation
//a)open switch
disp('OPEN switch')
disp(V_1_rms,'Primary voltage in volts')
disp(V_2_rms,'Secondary voltage in volts')
//As switch is open, current in second winding is 0
   which implies the current in primary coil to be 0
    (ideal transformer condition)
disp(0,'Current in primary winding in amperes')
disp(0,'Current in secondary winding in amperes')
//b)closed switch
disp('CLOSED switch')
I_2_rms=V_2_rms/R_L;       //ohm's law
I_1_rms=I_2_rms/tr;       //ideal transformer
   condition
disp(V_1_rms,'Primary voltage in volts')
```

```
disp(V_2_rms,'Secondary voltage in volts')
disp(I_1_rms,'Current in primary winding in amperes')
disp(I_2_rms,'Current in secondary winding in amperes')
```

Scilab code Exa 15.11 `example 11`

```
clc
//ex15.11
V_s=1000*complex(cos(0),sin(0));      //source voltage phasor
R_1=10^3;
R_L=10;
Z_L_1=R_L+%i*20;      //impedance
tr=10;      //turns ratio(N1/N2)
Z_L_2=(tr^2)*Z_L_1;      //reflecting Z_L_1 onto primary side
Z_s=R_1+Z_L_2;      //total impedance seen by the source
[Z_s_max,Z_s_phi]=polar(Z_s);
//primary quantities
I_1=V_s/Z_s;
[I_1_max,I_1_phi]=polar(I_1);
V_1=I_1*Z_L_2;
[V_1_max,V_1_phi]=polar(V_1);
//using turns ratio to find secondary quantities
I_2=tr*I_1;
[I_2_max,I_2_phi]=polar(I_2);
V_2=V_1/tr;
[V_2_max,V_2_phi]=polar(V_2);
I_2_rms=I_2_max/sqrt(2);
P_L=(I_2_rms^2)*R_L;      //power to load
printf(" All the values in the textbook are approximated hence the values in this code differ
```

```
    from those of Textbook")
//we take real parts of angles to take out
   neglegible and unnecessary imaginary parts(if any
    are there)
disp('PRIMARY CURRENT: ')
disp(I_1_max,'peak value in amperes')
disp(real(I_1_phi*180/%pi),'phase angle in degrees')
disp('PRIMARY VOLTAGE: ')
disp(V_1_max,'peak value in amperes')
disp(real(V_1_phi*180/%pi),'phase angle in degrees')
disp('SECONDARY CURRENT')
disp(I_2_max,'peak value in amperes')
disp(real(I_2_phi*180/%pi),'phase angle in degrees')
disp('SECONDARY VOLTAGE')
disp(V_2_max,'peak value in amperes')
disp(real(V_2_phi*180/%pi),'phase angle in degrees')
disp(P_L,'power delivered to load in watts')
```

Scilab code Exa 15.12 `example 12`

```
clc
//ex15.12
V_s=1000*complex(cos(0),sin(0));      //source
   voltage phasor
R_1=10^3;
tr=10;      //turns ratio(N1/N2)
V_S=V_s/tr;      //reflected voltage
[V_S_max,V_S_phi]=polar(V_S);
R1=R_1/(tr^2);      //reflected resistance
//we take real parts of angles to take out
   neglegible and unnecessary imaginary parts(if any
    are there)
disp('Reflected voltage:')
disp(V_S_max,'Peak value in volts')
disp(V_S_phi*180/%pi,'phase angle in degrees')
```

```
disp(R1,'Reflected resistance in ohms')
```

Scilab code Exa 15.13 `example 13`

```
clc
//ex15.13
V_L_max=240;
V_L=V_L_max*complex(cos(0),sin(0));       //load voltage
R_1=3;
R_2=0.03;
R_c=100*10^3;       //core-loss resistance
tr=10;       //turns ratio(N1/N2)
//leakage reactances
Z_1=%i*6.5;
Z_2=%i*0.07;
Z_m=%i*15*10^3;
P_R=20*10^3;       //rated power
I_2_max=P_R/real(V_L);
PF=0.8;       //power factor
phi=-acos(PF);       //-ve for lagging power
I_2=complex(I_2_max*cos(phi),I_2_max*sin(phi));       //phasor
I_1=I_2/tr;       //primary current
[I_1_max,I_1_phi]=polar(I_1);
V_2=V_L+(R_2+Z_2)*I_2;       //KVL equation
V_1=tr*V_2;
V_s=V_1+(R_1+Z_1)*I_1;       //KVL equation
[V_s_max,V_s_phi]=polar(V_s);
P_loss=((V_s_max^2)/R_c)+((I_1_max^2)*R_1)+((I_2_max^2)*R_2);       //power loss in transformer
P_L=V_L*I_2*PF;       //power to load
P_in=P_L+P_loss;       //input power
P_eff=(1-(P_loss/P_in))*100;
//under no-load condtions
```

```
I_1=0;
I_2=0;
V_1=V_s_max;
V_no_load=V_1/tr;
PR=((V_no_load-V_L_max)/V_L_max)*100;
disp(PR,'Percent regulation')
```

Chapter 16

DC Machines

Scilab code Exa 16.1 `example 1`

```
clc
//ex16.1
V_rms=440;
P_o_fl=5*746;       //full-load rated output power
I_rms_fl=6.8;       //full-load line current
PF_fl=0.78;       //full-load power factor
n_fl=1150;       //full-load speed in rpm
I_rms_nl=1.2;       //no-load line current
PF_nl=0.3;       //no-load power factor
n_nl=1195;       //no-load speed in rpm
P_in_fl=sqrt(3)*V_rms*I_rms_fl*PF_fl;       //full-load input power
P_loss_fl=P_in_fl-P_o_fl;       //full-load power loss
eff_fl=(P_o_fl/P_in_fl)*100;       //full-load efficiency
P_in_nl=sqrt(3)*V_rms*I_rms_nl*PF_nl;       //no-load input power
P_o_nl=0;       //no-load output power
eff_nl=0;       //no-load efficiency('0' as P_o_nl=0)
SR=(n_nl-n_fl)*100/n_fl;       //speed regulation
```

```
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp(P_loss_fl,'Power loss with full-load in watts')
disp(eff_fl,'Efficiency with full-load')
disp(P_in_nl,'Input power with no-load in watts')
disp(SR,'speed regulation percentage for the motor')
```

Scilab code Exa 16.2 `example 2`

```
clc
//ex16.2
B=1;        //magnetic flux density
l=0.3;
V_T=2;
R_A=0.05;
//CASE a
//bar is stationary at t=0
u_ini=0;        //initial velocity of bar is 0
e_A=B*l*u_ini;        //induced voltage
i_A_ini=(V_T-e_A)/R_A;        //initial current
F_ini=B*l*i_A_ini;        //initial force on the bar
//steady state condition with no-load e_A=B*l*u=V_T
u=V_T/(B*l);        //from steady state condition with no-load
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp('CASE a:')
disp(i_A_ini,'initial current in amperes')
disp(F_ini,'initial force on the bar in newtons')
disp(u,'steady-state final speed in m/s')
//CASE b
F_load=4;        //mechanical load
//steady state condition F=B*l*i_A=F_load
```

```
i_A=F_load/(B*l);       //from steady state condition
e_A=V_T-R_A*i_A;       //induced voltage
u=e_A/(B*l);       //steady-state speed
P_m=F_load*u;       //mechanical power
P_t=V_T*i_A;       //power taken from battery
P_R=i_A^2*R_A;       //power dissipated in the resistance
eff=P_m*100/P_t;       //efficiency
disp('CASE b:')
disp(u,'steady-state speed in m/s')
disp(P_t,'power delivered by V_t in watts')
disp(P_m,'power delivered to mechanical load in watts')
disp(P_R,'power lost to heat in the resistance in watts')
disp(eff,'effciency of converting electrical power to mechanical power')
//CASE c
//with the pulling force acting to the right, machine operates as a generator
F_pull=2;       //pulling force
//steady-state condition F=B*l*i_A=F_pull
i_A=F_pull/(B*l);       //from steady-state condition
e_A=V_T+R_A*i_A;       //induced voltage
u=e_A/(B*l);       //steady-state speed
P_m=F_pull*u;       //mechanical power
P_t=V_T*i_A;       //power taken by battery
P_R=i_A^2*R_A;       //power dissipated in the resistance
eff=P_t*100/P_m;       //efficiency
disp('CASE c:')
disp(u,'steady-state speed in m/s')
disp(P_m,'power taken from mechanical source in watts')
disp(P_t,'power delivered to the battery in watts')
disp(P_R,'power lost to heat in the resistance')
disp(eff,'efficiency of converting mechanical power to electrical power')
```

Scilab code Exa 16.3 `example 3`

```
clc
//ex16.3
n_2=800;        //speed in rpm
I_A=30;        //armature current
I_F=2.5;        //field current
R_A=0.3;        //armature resistance
R_F=50;        //field resistance
V_F=I_F*R_F;        //field coil voltage
//E_A1 and n_1 from magnetization curve
E_A1=145;        //induced voltage
n_1=1200;        //speed in rpm
E_A2=n_2*E_A1/n_1;
W_m=n_2*2*%pi/60;        //speed in radians per second
K=E_A2/W_m;        //K*phi is taken as K, machine
   constant
T_dev=K*I_A;        //developed torque
P_dev=W_m*T_dev;        //developed power
V_T=R_A*I_A+E_A2;        //voltage applied to armature
printf(" All the values in the textbook are
   approximated hence the values in this code differ
    from those of Textbook")
disp(V_F,'Voltage applied to field circuit in volts'
   )
disp(V_T,'Voltage applied to armature in volts')
disp(T_dev,'Developed torque in Nm')        //Nm-
   newton meter
disp(P_dev,'Developed power in watts')
```

Scilab code Exa 16.4 `example 4`

```
clc
//ex16.4
V_T=240;        //dc supply voltage
R_A=0.065;        //armature resistance
R_F=10;        //field resistance
R_adj=14;        //adjustable resistance
n=1200;        //speed in rpm
P_rot=1450;        //rotational power loss
T_out=250;        //hoist torque
I_F=V_T/(R_F+R_adj);        //field current
//E_A at I_F and n from magnetization curve
E_A_1=280;        //armature voltage
W_m_1=n*2*%pi/60;        //speed in radians per second
K=E_A_1/W_m_1;        //machine constant
T_rot=P_rot/W_m_1;        //rotational loss-torque
T_dev=T_rot+T_out;        //developed torque
I_A=T_dev/K;        //armature current
E_A_2=V_T-R_A*I_A;        //applying KVL
W_m_2=E_A_2/K;        //speed in radians per second
n_m=W_m_2*60/(2*%pi);        //speed in rpm
P_out=T_out*W_m_2;        //output power
I_L=I_F+I_A;        //line current
P_in=V_T*I_L;        //input power
eff=P_out*100/P_in;        //efficiency
printf(" All the values in the textbook are
    approximated hence the values in this code differ
     from those of Textbook")
disp(n_m,'Motor speed in rpm')
disp(eff,'Efficiency of the motor')
```

Scilab code Exa 16.5 `example 5`

```
clc
//ex16.5
n_m_1=1200;        //speed in rpm
```

```
T_out_1=12;        //motor torque
W_m_1=n_m_1*2*%pi/60;        //angular speed
//As we are neglecting losses, the output torque and power are equal to the developed torque and power respectively
P_out_1=W_m_1*T_out_1;        //output power
//For Torque=24
T_out_2=24;
T_dev_2=T_out_2;
//T_dev=K*K_F*V_T^2/(R_A+R_F+K*K_F*W_m^2)
//neglecting resistances and with the above equation for T_dev, we get inverse relation between torque and square of speed
W_m_2=W_m_1*sqrt(T_out_1)/sqrt(T_dev_2);
n_m_2=W_m_2*60/(2*%pi);
P_out_2=T_dev_2*W_m_2;
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp(P_out_1,'Output power for load torque=12 in watts')
disp(n_m_2,'speed for torque=24 in rpm')
disp(P_out_2,'Output power for load torque=24 in watts')
```

Scilab code Exa 16.6 example 6

```
clc
//ex16.6
V_F=140;        //field voltage
R_F=10;        //field resistance
R_adj=4;        //adjusting resistance
R_A=0.065;        //armature resistance
n_A=1000;        //armature speed in rpm
I_fl=200;        //full-load current
```

```
eff=0.85;       //efficiency not including power supplied to field circuit
I_F=V_F/(R_adj+R_F);       //field current
//E, voltage from magnetization curve for speed of n=1200
n=1200;
E=280;       //voltage of armature
//E_A is no-load voltage
E_A=E*n_A/n;       //E_A is proportional to speed
V_FL=E_A-R_A*I_fl;       //full-load voltage
VR=(E_A-V_FL)*100/V_FL;       //voltage regulation
P_out=I_fl*V_FL;       //output power
P_dev=P_out+(I_fl^2)*R_A;       //developed power
W_m=n_A*2*%pi/60;       //angular speed
P_in=P_out/eff;       //input power
P_loss=P_in-P_dev;       //all power losses combined
T_in=P_in/W_m;       //input torque
T_dev=P_dev/W_m;       //developed torque
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp(I_F,'Field current in amperes')
disp(E_A,'no-load voltage in volts')
disp(V_FL,'full-load voltage in volts')
disp(VR,'percentage voltage regulation')
disp(T_in,'input torque in Nm')
disp(T_dev,'developed torque')
disp(P_loss,'all types of power losses combined in watts')
```

Chapter 17

AC Machines

Scilab code Exa 17.1 `example 1`

```
clc
//ex17.1
P_rot=900;      //rotational losses
V_L=440*complex(cos(0),sin(0));
R_s=1.2;
X_s=%i*2;
X_m=%i*50;
R_r_1=0.6;
R_r_2=19.4;
X_r=%i*0.8;
n_m=1746;      //machine operating speed in rpm
W_m=n_m*2*%pi/60;      //speed in radians per second
n_s=1800;      //synchronous speed for a four-pole monitor
s=(n_s-n_m)/n_s;      //slip
Z_s=R_s+X_s+(X_m*(R_r_1+R_r_2+X_r))/(X_m+R_r_1+R_r_2+X_r);      //impedance seen by the source
[Z_s_max,phi]=polar(Z_s);
Z_s_phi=real(phi);      //removing negligible imaginary part(if any is there)
PF=cos(Z_s_phi);      //power factor
```

```
V_s=V_L;      //phase voltage
I_s=V_s/Z_s;      //phase current
[I_s_max,I_s_phi]=polar(I_s);
I_L=I_s_max*sqrt(3);      //line current
P_in=3*I_s*V_s*PF;      //input power
V_x=I_s*(X_m*(R_r_1+R_r_2+X_r))/(X_m+R_r_1+R_r_2+X_r);
I_r=V_x/(X_r+R_r_1+R_r_2);
[I_r_max,I_r_phi]=polar(I_r);
P_s=3*R_s*I_s_max^2;      //copper loss in stator
P_r=3*R_r_1*I_r_max^2;      //copper loss in rotor
P_dev=3*(1-s)*R_r_1*I_r_max^2/s;      //developed power
//we may verify that P_in=P_dev+P_s+P_r to within rounding error
P_in=P_dev+P_s+P_r;      //input power
P_o=P_dev-P_rot;      //output power
T_o=P_o/W_m;      //output torque
eff=P_o*100/P_in;      //efficiency
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp(PF,'Power factor')
disp(I_L,'line current in amperes')
disp(P_o,'output power in watts')
disp(T_o,'output torque in Nm')
disp(eff,'efficiency percentage is')
```

Scilab code Exa 17.2 `example 2`

```
clc
//ex17.2
s=1;      //slip for starting
V_L=440*complex(cos(0),sin(0));
f=60;
```

```
R_s=1.2;
X_s=%i*2;
X_m=%i*50;
R_r_1=0.6;
R_r_2=19.4;
X_r=%i*0.8;
Z_eq=X_m*(R_r_1+X_r)/(X_m+R_r_1+X_r);      //equivalent impedance to the right in the figure in textbook
Z_s=R_s+X_s+Z_eq;
I_s=V_L/Z_s;      //starting phase current
[I_s_max,phi]=polar(I_s);
I_L=sqrt(3)*I_s_max;      //starting line current
//I_L here is almost six times larger than in previous example. It is a typical characteristic of induction motors.
P_ag=3*real(Z_eq)*I_s_max^2;      //power crossing air gap
W_s=2*%pi*(60);
T_dev=P_ag/(W_s/2);
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp(I_L,'Starting line current')
disp(T_dev,'Torque in Nm')
```

Scilab code Exa 17.3 example 3

```
clc
//ex17.3
V_L=220;
V_s=V_L/sqrt(3);      //phase voltage
I_s=31.87;
P_s=400;      //total stator copper losses
P_r=150;      //total rotoe copper losses
```

```
P_rot=500;      //rotational losses
PF=0.75;      //power factor
P_in=3*V_s*I_s*PF;      //input power
P_ag=P_in-P_s;      //air-gap power
P_dev=P_in-P_s-P_r;      //developed power
P_o=P_dev-P_rot;      //output power
eff=P_o*100/P_in;      //efficiency
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp(P_ag,'Power crossing the air gap in watts')
disp(P_dev,'developed power in watts')
disp(P_o,'output power in watts')
disp(eff,'effciency percentage')      //this value is given wrong in the textbook
```

Scilab code Exa 17.4 `example 4`

```
//ex17.4
P_dev_1=50*746;      //developed power
V_L=480;      //line voltage
PF=0.9;      //power factor
f=60;      //frequency
P=8;      //number of poles
X_s=1.4;      //synchronous reactance
//CASE a
n_s=120*f/P;      //speed of machine in rpm
W_s=n_s*2*%pi/60;      //speed in radians per second
T_dev=P_dev_1/W_s;      //developed torque
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp('CASE a:')
disp(n_s,'speed in rpm')
disp(T_dev,'developed torque in Nm')
```

```
//CASE b
V_a=V_L;      //phase voltage
I_a_max=P_dev_1/(3*V_a*PF);      //phase current
phi=acos(PF);
I_a=I_a_max*complex(cos(phi),sin(phi));
E_r=V_a-%i*X_s*I_a;      //voltage induced by rotor
E_r_max=sqrt((real(E_r)^2)+(imag(E_r)^2));
E_r_phi=atan(imag(E_r)/real(E_r));
TA=-E_r_phi;      //torque angle
disp('CASE b:')
disp('Phase current:')
disp(I_a_max,'peak value in amperes')
disp(phi*180/%pi,'phase angle in degrees')
disp('Voltage induced by rotor:')
disp(E_r_max,'peak value in volts')
disp(E_r_phi*180/%pi,'phase angle in degrees')
disp(TA*180/%pi,'torque angle in degrees')
//CASE c
//excitation constant means the values of I_f, B_r and E_r are constant
P_dev_2=100*746;
sin_t=P_dev_2*sin(TA)/P_dev_1;      //developed power is proportional to sin_t
t=asin(sin_t);
E_r=E_r_max*complex(cos(-t),sin(-t));      //E_r is constant in magnitude
I_a=(V_a-E_r)/(%i*X_s);      //new phase current
I_a_max=sqrt((real(I_a)^2)+(imag(I_a)^2));
I_a_phi=atan(imag(I_a)/real(I_a));
PF=cos(I_a_phi);
disp('CASE c:')
disp('Phase current:')
disp(I_a_max,'peak value in amperes')
disp(I_a_phi*180/%pi,'phase angle in degrees')
disp('Voltage induced by rotor:')
disp(E_r_max,'peak value in volts')
disp(-t*180/%pi,'phase angle in degrees')
disp(t*180/%pi,'torque angle in degrees')
```

```
disp(PF,'power factor is')
```

Scilab code Exa 17.5 `example 6`

```
clc
//ex17.5
V_a=480;       //phase voltage
f=60;       //frequency
P_dev=200*746;       //developed power
PF=0.85;       //power factor
I_f_1=10;       //field current
X_s=1.4;       //synchronous resistance
phi=acos(PF);
I_a_1_max=P_dev/(3*V_a*PF);       //phase current
I_a_1_phi=-phi;
I_a_1=I_a_1_max*complex(cos(-phi),sin(-phi));
E_r_1=V_a-%i*X_s*I_a_1;       //rotor induced voltage
[E_r_1_max,E_r_1_phi]=polar(E_r_1);
//to achieve 100 percent power factor, increase I_a until it is in phase with V_a
I_a_2=P_dev/(3*V_a*cos(0));
E_r_2=V_a-%i*X_s*I_a_2;
[E_r_2_max,E_r_2_phi]=polar(E_r_2);
I_f_2=I_f_1*E_r_2_max/E_r_1_max;       //magnitude of E_r proportional to field current
printf(" All the values in the textbook are approximated hence the values in this code differ from those of Textbook")
disp(I_f_2,'The new field current to achieve 100% power factor in amperes')
```