

Scilab Textbook Companion for  
Electrical Engineering - Principles And  
Applications  
by A. R. Hambley<sup>1</sup>

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

May 24, 2016

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

# Book Description

**Title:** 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	4
1 Introduction	5
2 Resistive Circuits	10
3 Inductance and Capacitance	24
4 Transients	30
5 Steady state sinusoidal analysis	32
6 Frequency response bode plots and resonance	47
7 Logic circuits	58
9 Computer based instrumentation diodes	62
10 Diodes	63
11 Amplifiers Specifications and external characteristics	68
12 Field effect transistors	77
13 Bipolar junction transistors	81
14 Operational Amplifiers	91
15 Magnetic circuits and transformers	94

<b>16 DC Machines</b>	<b>104</b>
<b>17 AC Machines</b>	<b>111</b>

# List of Scilab Codes

Exa 1.1	example 1 . . . . .	5
Exa 1.2	example 2 . . . . .	5
Exa 1.3	example 3 . . . . .	7
Exa 1.4	example 4 . . . . .	7
Exa 1.5	example 5 . . . . .	8
Exa 1.6	example 6 . . . . .	8
Exa 1.7	example 7 . . . . .	9
Exa 2.1	example 1 . . . . .	10
Exa 2.2	example 2 . . . . .	10
Exa 2.3	example 3 . . . . .	12
Exa 2.4	example 4 . . . . .	12
Exa 2.5	example 5 . . . . .	13
Exa 2.6	example 6 . . . . .	14
Exa 2.7	example 7 . . . . .	14
Exa 2.8	example 8 . . . . .	15
Exa 2.9	example 9 . . . . .	15
Exa 2.11	example 11 . . . . .	15
Exa 2.12	example 12 . . . . .	16
Exa 2.13	example 13 . . . . .	16
Exa 2.14	example 14 . . . . .	17
Exa 2.15	example 15 . . . . .	17
Exa 2.16	example 16 . . . . .	18
Exa 2.17	example 17 . . . . .	18
Exa 2.18	example 18 . . . . .	19
Exa 2.19	example 19 . . . . .	20
Exa 2.20	example 20 . . . . .	20
Exa 2.21	example 21 . . . . .	21
Exa 2.22	example 22 . . . . .	22

Exa 2.23	example 23 . . . . .	23
Exa 3.1	example 1 . . . . .	24
Exa 3.2	example 2 . . . . .	25
Exa 3.3	example 3 . . . . .	26
Exa 3.4	example 4 . . . . .	27
Exa 3.5	example 5 . . . . .	27
Exa 3.6	example 6 . . . . .	28
Exa 4.1	example 1 . . . . .	30
Exa 4.2	example 2 . . . . .	30
Exa 5.1	example 1 . . . . .	32
Exa 5.2	example 2 . . . . .	33
Exa 5.3	example 3 . . . . .	33
Exa 5.4	example 4 . . . . .	34
Exa 5.5	example 5 . . . . .	35
Exa 5.6	example 6 . . . . .	36
Exa 5.7	example 7 . . . . .	37
Exa 5.8	example 8 . . . . .	39
Exa 5.9	example 9 . . . . .	40
Exa 5.10	example 10 . . . . .	40
Exa 5.11	example 11 . . . . .	42
Exa 5.12	example 12 . . . . .	43
Exa 5.13	example 13 . . . . .	44
Exa 5.14	example 14 . . . . .	45
Exa 6.1	example 1 . . . . .	47
Exa 6.2	example 2 . . . . .	48
Exa 6.3	example 3 . . . . .	50
Exa 6.4	example 4 . . . . .	52
Exa 6.5	example 5 . . . . .	53
Exa 6.6	example 6 . . . . .	55
Exa 6.7	example 7 . . . . .	56
Exa 7.1	example 1 . . . . .	58
Exa 7.2	example 2 . . . . .	58
Exa 7.3	example 3 . . . . .	59
Exa 7.4	example 4 . . . . .	59
Exa 7.5	example 5 . . . . .	60
Exa 7.6	example 6 . . . . .	60
Exa 9.1	example 1 . . . . .	62
Exa 10.1	example 1 . . . . .	63

Exa 10.2	example 2 . . . . .	64
Exa 10.3	example 3 . . . . .	65
Exa 10.4	example 4 . . . . .	65
Exa 10.5	example 5 . . . . .	66
Exa 10.7	example 7 . . . . .	67
Exa 11.1	example 1 . . . . .	68
Exa 11.2	example 2 . . . . .	69
Exa 11.3	example 3 . . . . .	70
Exa 11.4	example 4 . . . . .	70
Exa 11.5	example 5 . . . . .	71
Exa 11.6	example 6 . . . . .	72
Exa 11.7	example 7 . . . . .	72
Exa 11.8	example 8 . . . . .	73
Exa 11.9	example 9 . . . . .	73
Exa 11.10	example 10 . . . . .	74
Exa 11.11	example 11 . . . . .	75
Exa 11.12	example 12 . . . . .	76
Exa 12.1	example 1 . . . . .	77
Exa 12.2	example 2 . . . . .	77
Exa 12.3	example 3 . . . . .	78
Exa 12.5	example 5 . . . . .	79
Exa 13.1	example 1 . . . . .	81
Exa 13.2	example 2 . . . . .	81
Exa 13.4	example 4 . . . . .	82
Exa 13.5	example 5 . . . . .	84
Exa 13.6	example 6 . . . . .	85
Exa 13.7	example 7 . . . . .	86
Exa 13.8	example 8 . . . . .	87
Exa 13.9	example 9 . . . . .	89
Exa 14.5	example 5 . . . . .	91
Exa 14.6	example 6 . . . . .	91
Exa 14.7	example 7 . . . . .	92
Exa 15.3	example 3 . . . . .	94
Exa 15.5	example 5 . . . . .	95
Exa 15.6	example 6 . . . . .	96
Exa 15.7	example 7 . . . . .	97
Exa 15.8	example 8 . . . . .	98
Exa 15.9	example 9 . . . . .	98



Exa 15.10	example 10 . . . . .	99
Exa 15.11	example 11 . . . . .	100
Exa 15.12	example 12 . . . . .	101
Exa 15.13	example 13 . . . . .	102
Exa 16.1	example 1 . . . . .	104
Exa 16.2	example 2 . . . . .	105
Exa 16.3	example 3 . . . . .	107
Exa 16.4	example 4 . . . . .	107
Exa 16.5	example 5 . . . . .	108
Exa 16.6	example 6 . . . . .	109
Exa 17.1	example 1 . . . . .	111
Exa 17.2	example 2 . . . . .	112
Exa 17.3	example 3 . . . . .	113
Exa 17.4	example 4 . . . . .	114
Exa 17.5	example 6 . . . . .	116

# Chapter 1

## Introduction

Scilab code Exa 1.1 example 1

```
1 //ex1.1
2 //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
3 t=[0:0.000001:0.04];
4 q=2*(1-%e^(-100*t));
5 //current i=dq/dt=200*e^(-100*t)
6 i=200*%e^(-100*t);
7 subplot(121)
8 xtitle('charge vs time','time in ms','charge in
   coulombs') //ms-milli second(10^-3)
9 plot(t*10^3,q)
10 subplot(122)
11 xtitle('current vs time','time in ms','current in
   amperes') //ms-milli second(10^-3)
12 plot(t*10^3,i)
```

---

Scilab code Exa 1.2 example 2

```

1  clc
2  //ex1.2
3
4  //element A
5  disp('ELEMENT A :')
6  V_a=12;
7  i_a=2;
8  P_a=V_a*i_a;      //passive reference configuration
                      (current enters through +ve polarity)
9  if(P_a>0) then,    //absorption of power
10     disp(P_a,'Power for element A in watts is')
11     disp('As a battery , element A is being charged')
12 elseif(P_a<0) then,    //supplying of power
13     disp(P_a,'Power for element A in watts is')
14     disp('As a battery , element A is being
                      discharged')
15 end
16
17 //element B
18 disp('ELEMENT B')
19 V_b=12;
20 i_b=1;
21 P_b=-V_b*i_b;      //opposite to passive reference
                      configuration (current enters through -ve
                      polarity)
22 if(P_b>0) then,    //absorption of power
23     disp(P_b,'Power for element B in watts is')
24     disp('As a battery , element B is being charged')
25 elseif(P_b<0) then,    //supplying of power
26     disp(P_b,'Power for element B in watts is')
27     disp('As a battery , element B is being
                      discharged')
28 end
29
30 //element C
31 disp('ELEMENT C')
32 V_c=12;
33 i_c=-3;

```

```

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

```

---

#### Scilab code Exa 1.3 example 3

```

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

```

---

#### Scilab code Exa 1.4 example 4

```

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

```

```

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

```

---

#### Scilab code Exa 1.5 example 5

```

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

```

---

#### Scilab code Exa 1.6 example 6

```

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

```

```

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

```

---

#### Scilab code Exa 1.7 example 7

```

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

```

---

# Chapter 2

## Resistive Circuits

Scilab code Exa 2.1 example 1

```
1  clc
2  //ex2.1
3  R_1=10;
4  R_2=20;
5  R_3=5;
6  R_4=15;
7  //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.
8  R_eq_1=R_3+R_4;           //R_3 and R_4 in series
9  R_eq_2=1/((1/R_eq_1)+(1/R_2)); //R_eq_1 and R_2
   in parallel
10 R_eq=R_1+R_eq_2;         //R_1 and R_eq_2 in series
11 disp(R_eq, 'Equivalent resistance in ohms')
```

---

Scilab code Exa 2.2 example 2

```

1  clc
2  //ex2.2
3  V_s=90;           //source voltage
4  R_1=10;
5  R_2=30;
6  R_3=60;
7  R_eq_1=1/((1/R_2)+(1/R_3));      //R_2 and R_3 in
    parallel
8  R_eq=R_1+R_eq_1;      //R_1 and R_eq_1 in series
9  i_1=V_s/R_eq;      //ohm's law
10 //i_1 flows clockwise through V_s,R_1 and R_eq_1
11 V_2=R_eq_1*i_1;      //voltage across R_eq_1
12 //As R_eq_1 is equivalent of parallel combination of
    R_2 and R_3, V_2 appears across both of them
13 i_2=V_2/R_2;      //ohm's law
14 i_3=V_2/R_3;      //ohm's law
15 //we can verify KCL, i_1=i_2+i_3
16 V_1=i_1*R_1;      //ohm's law
17 //we can verify KVL, V_s=V_1+V_2
18 P_s=-V_s*i_1;      //source power(-ve sign as V_s
    and i_1 have references opposite to passive
    configuration)
19 P_1=i_1^2*R_1;      //power for R_1
20 P_2=V_2^2/R_2;      //power for R_2
21 P_3=V_2^2/R_3;      //power for R_3
22 disp('FOR SOURCE')
23 disp(i_1,'current in amperes')
24 disp(P_s,'power in watts')
25 disp('FOR R1')
26 disp(i_1,'current in amperes')
27 disp(V_1,'voltage in volts')
28 disp(P_1,'power in watts')
29 disp('FOR R2')
30 disp(i_2,'current in amperes')
31 disp(V_2,'voltage in volts')
32 disp(P_2,'power in watts')
33 disp('FOR R3')
34 disp(i_3,'current in amperes')

```



```

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

```

---

### Scilab code Exa 2.3 example 3

```

1 //ex2.3
2 V_total=15;
3 R_1=1*10^3;
4 R_2=1*10^3;
5 R_3=2*10^3;
6 R_4=6*10^3;
7 //By voltage-division principle
8 V_1=R_1*V_total/(R_1+R_2+R_3+R_4); //voltage
   across R_1
9 V_4=R_4*V_total/(R_1+R_2+R_3+R_4); //voltage
   across R_4
10 disp(V_1, 'voltage across R_1')
11 disp(V_4, 'voltage across R_4')

```

---

### Scilab code Exa 2.4 example 4

```

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

```

```

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

```

---

#### Scilab code Exa 2.5 example 5

```

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

```

---

### Scilab code Exa 2.6 example 6

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

---

### Scilab code Exa 2.7 example 7

```
1 clc
2 //ex2.7
3 disp('The matrix form is')
4 disp('G*V=I')
5 disp('where')
6 G=[0.45,-0.25,0;-0.25,0.85,-0.20;0,-0.20,0.30];
7 disp(G,'G=')
8 disp('V=')
9 disp('transpose of [V_1,V_2,V_3]')
10 disp('and')
11 I=[-3.5;3.5;2];
12 disp(I,'I=')
```

---

### Scilab code Exa 2.8 example 8

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

---

### Scilab code Exa 2.9 example 9

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

---

### Scilab code Exa 2.11 example 11

```
1 clc
```

```

2 //ex2.11
3 disp('KCL for a supernode enclosing the controlled
      voltage source')
4 disp('(V1/R2)+((V1-V3)/R1)+((V2-V3)/R3)=is')
5 disp('KCL at node 3')
6 disp('(V3/R4)+((V3-V2)/R3)+((V3-V1)/R1)=0')
7 disp('KCL at the reference node')
8 disp('(V1/R2)+(V3/R4)=is')
9 disp('From the closed loop with V1,Vx and V3')
10 disp('Vx=V3-V1')
11 disp('Applying KVL')
12 disp('V1=0.5(V3-V1)+V2')
13 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

```

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

```

---

#### Scilab code Exa 2.13 example 13

```

1 clc

```

```

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

```

---

**Scilab code Exa 2.14** example 14

```

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

```

---

**Scilab code Exa 2.15** example 15

```

1 clc
2 //ex2.15
3 //KVL over the supermesh , we get eqn-1   -20+4(i1)
   +8(i2)=0
4 //Vx=2(i2) ohm's law
5 //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

```

```

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

```

---

#### Scilab code Exa 2.16 example 16

```

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

```

---

#### Scilab code Exa 2.17 example 17

```

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

```

---

### Scilab code Exa 2.18 example 18

```

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

```



```

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

```

---

#### Scilab code Exa 2.19 example 19

```

1  clc
2  //ex2.19
3  R1= 20 //Ohms
4  R2= 15 //ohms
5  vs= 15 //V
6  R3= 5 //Ohms
7  k= 0.25
8  ///CALCULATIONS
9  voc= (R2/R1)/((1/R1)+(1/(R2+R3))+(k/4))
10 isc= vs/R1
11 Rf= voc/isc
12 //RESULTS
13 printf ('Rf = %.2 f ohms ',Rf)

```

---

#### Scilab code Exa 2.20 example 20

```

1  clc
2  //ex2.20
3  V_s_1=20;          //voltage source

```

```

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

---

### Scilab code Exa 2.21 example 21

```

1 clc
2 //ex2.21
3 V_s=50;
```

```

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

```

---

#### Scilab code Exa 2.22 example 22

```

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

```

---

Scilab code Exa 2.23 example 23

```
1  clc
2  //ex2.23
3  R_1=1*10^3;
4  //case (a)
5  disp('case a:')
6  R_2=10*10^3;
7  R_3=732;
8  R_x=R_2*R_3/R_1;          //wheatstone bridge condition
9  disp(R_x,'Value of Rx in ohms')
10 //case (b)
11 disp('case b:')
12 //R_x is maximum when both R_2 and R_3 are maximum
13 R_2_max=1*10^6;
14 R_3_max=1100;
15 R_x_max=R_2_max*R_3_max/R_1;          //wheatstone
    bridge condition
16 disp(R_x_max,'Maximum value of Rx in ohms')
17 //case(c)
18 disp('case c:')
19 //increment in R_x is scale factor times increment
    in R_3
20 R_2=1*10^6;
21 R_3_inc=1;          //increment in R_3
22 R_x_inc=R_2*R_3_inc/R_1;          //increment in R_x
    from bride balance condition
23 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

```
1  clc
2  //ex3.1
3  C=1*10^-6;
4  //t in micro seconds
5  t_1=[0:0.001:2];
6  t_2=[2.001:0.001:4];
7  t_3=[4.001:0.001:5];
8  t=[t_1,t_2,t_3];
9  //corresponding voltage variations
10 V_1=5*t_1;
11 V_2=0*t_2+10;
12 V_3=-10*t_3+50;
13 //charge q=C*V
14 q_1=C*V_1;
15 q_2=C*V_2;
16 q_3=C*V_3;
17 q=[q_1,q_2,q_3];
18 subplot(121)
19 plot(t,q*10^6)
20 xtitle('charge vs time','time in Ms','charge in Mc')
    //M-micro(10^-6)
```

```

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

```

---

### Scilab code Exa 3.2 example 2

```

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

```

---

### Scilab code Exa 3.3 example 3

```
1  clc
2  //ex3.3
3  C=10*10^-6;
4  t_1=[0:0.001:1];
5  t_2=[1.001:0.001:3];
6  t_3=[3.001:0.001:5];
7  t=[t_1,t_2,t_3];
8  //voltage variations
9  V_1=1000*t_1;
10 V_2=0*t_2+1000;
11 V_3=500*(5-t_3);
12 //current i=C*dv/dt, for above equations we get
13 i_1=C*(0*t_1+1000);
14 i_2=C*(0*t_2);
15 i_3=C*(0*t_3-500);
16 i=[i_1,i_2,i_3];
17 //power delivered, P=V*i
18 P_1=C*(10^6*t_1);
19 P_2=C*(0*t_2+1000);
20 P_3=C*(-25*10^4*(5-t_3));
21 P=[P_1,P_2,P_3];
22 //energy stored, W=(1/2)*C*V^2
23 W_1=(1/2)*C*V_1^2;
24 W_2=(1/2)*C*V_2^2;
25 W_3=(1/2)*C*V_3^2;
26 W=[W_1,W_2,W_3];
27 subplot(221)
28 plot(t,i*10^3)
29 xtitle('current vs time','time in seconds','current
        in mA') //mA-milli amperes(10^-3)
30 subplot(222)
31 plot(t,P)
```

```

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

```

---

#### Scilab code Exa 3.4 example 4

```

1 clc
2 //ex3.4
3 L=10*10-2; //length
4 W=20*10-2; //width
5 d=0.1*10-3; //distance between plates
6 A=L*W; //area
7 E_o=8.85*10-12; //dielectric constant of
    vacuum
8 //dielectric is air
9 E_r=1; //relative dielectric constant of air
10 E=E_r*E_o; //dielectric constant
11 C=E*A/d; //capacitance
12 disp('When the dielectric is air, capacitance in pF
    is ') //pF-pico Farad(10-12)
13 disp(C*1012)
14 //dielectric is mica
15 E_r=7; //relative dielectric constant of mica
16 E=E_r*E_o; //dielectric constant
17 C=E*A/d; //capacitance
18 disp('When the dielectric is mica, capacitance in pF
    is ') //pF-pico Farad(10-12)
19 disp(C*1012)

```

---

#### Scilab code Exa 3.5 example 5



```

1  clc
2  //ex3.5
3  C_1=1*10^-6;
4  C_2=1*10^-6;
5  //Before the switch is closed
6  V_1=100;
7  V_2=0;
8  W_1=(1/2)*C_1*V_1^2;
9  W_2=0;          //V_2=0
10 W_t_1=W_1+W_2;    //total energy stored by both
    the capacitors before switch is closed
11 q_1=C_1*V_1;
12 q_2=0;
13 //After the switch is closed
14 q_eq=q_1+q_2;    //charge on equivalent
    capacitance
15 C_eq=C_1+C_2;    //C_1 and C_2 in parallel
16 V_eq=q_eq/C_eq;
17 V_1=V_eq;       //parallel combination
18 V_2=V_eq;       //parallel combination
19 W_1=(1/2)*C_1*V_eq^2;
20 W_2=(1/2)*C_2*V_eq^2;
21 W_t_2=W_1+W_2;    //total energy stored by both
    the capacitors after switch is closed
22 disp(W_t_1*10^3, 'Total energy stored by both the
    capacitors before switch is closed in mJ')
    //mJ-milli Joules(10^-3)
23 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

```

1  clc
2  //ex3.6

```

```

3 L=5;          //inductance
4 t_1=[0:0.001:2];
5 t_2=[2.001:0.001:4];
6 t_3=[4.001:0.001:5];
7 t=[t_1,t_2,t_3];
8 //corresponding current variations
9 i_1=(1.5)*t_1;
10 i_2=0*t_2+3;
11 i_3=(-3*t_3)+15;
12 //voltage V=L*(di/dt)
13 V_1=L*(0*t_1+(1.5));
14 V_2=L*(0*t_2);
15 V_3=L*(0*t_3-3);
16 V=[V_1,V_2,V_3];
17 //stored energy W=1/2*L*i^2
18 W_1=(1/2)*L*i_1^2;
19 W_2=(1/2)*L*i_2^2;
20 W_3=(1/2)*L*i_3^2;
21 W=[W_1,W_2,W_3];
22 //power P=V*i
23 P_1=L*t_1*(1.5^2);
24 P_2=0*t_2;
25 P_3=-3*L*(-3*t_3+15);
26 P=[P_1,P_2,P_3];
27 subplot(221)
28 plot(t,V)
29 xtitle('voltage vs time','time in seconds','voltage
        in volts')
30 subplot(222)
31 plot(t,W)
32 xtitle('energy vs time','time in seconds','energy in
        joules')
33 subplot(223)
34 plot(t,P)
35 xtitle('power vs time','time in seconds','power in
        watts')

```

---

# Chapter 4

## Transients

Scilab code Exa 4.1 example 1

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

---

Scilab code Exa 4.2 example 2

```

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

```

---

# Chapter 5

## Steady state sinusoidal analysis

Scilab code Exa 5.1 example 1

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

```

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

```

---

### Scilab code Exa 5.2 example 2

```

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

```

---

### Scilab code Exa 5.3 example 3

```

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

```

```

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

```

---

#### Scilab code Exa 5.4 example 4

```

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

```

```

14 V_R=R*I;
15 V_L=Z_L*I;
16 V_C=Z_C*I;
17 printf(" All the values in the textbook are
           approximated hence the values in this code differ
           from those of Textbook")
18 //for resistance R
19 disp('For resistance R')
20 V_R_max=sqrt((real(V_R)^2)+(imag(V_R)^2));
21 V_R_phi=(atan(imag(V_R)/real(V_R)))*180/%pi;
22 disp(V_R_max,'peak value of voltage in volts')
23 disp(V_R_phi,'phase angle in degrees')
24 //result : V_R=Vcos(wt+phi)  V-peak voltage
25 //for inductance L
26 disp('For inductance L')
27 V_L_max=sqrt((real(V_L)^2)+(imag(V_L)^2));
28 V_L_phi=(atan(imag(V_L)/real(V_L)))*180/%pi;
29 disp(V_L_max,'peak value of voltage in volts')
30 disp(V_L_phi,'phase angle in degrees')
31 //result : V_L=Vcos(wt+phi)  V-peak voltage
32 //for capacitor C
33 disp('For capacitor C')
34 V_C_max=sqrt((real(V_C)^2)+(imag(V_C)^2));
35 V_C_phi=(atan(imag(V_C)/real(V_C)))*180/%pi;
36 disp(V_C_max,'peak value of voltage in volts')
37 disp(V_C_phi,'phase angle in degrees') //cos(t)
           =cos(t-180) (we get 75 instead of -105 given in
           textbook)
38 //result : V_C=Vcos(wt+phi)  V-peak voltage
39 disp('The phasor diagram cannot be plotted')

```

---

### Scilab code Exa 5.5 example 5

```

1 clc
2 //ex5.5

```



```

3 V_s_max=10;          //peak value of source voltage
4 phi=-%pi/2;         //phase of source voltage
5 V_s=complex(10*cos(%pi/2),10*sin(%pi/2));          //
   phasor of source voltage
6 W=1000;            //angular frequency
7 R=100;
8 L=0.1;
9 C=10*10^-6;
10 Z_L=%i*W*L;       //impedance of inductance
11 Z_C=-%i/(W*C);    //impedance of capacitance
12 Z_RC=1/((1/R)+(1/Z_C)); //R and Z_C in parallel
   combination
13 V_C=V_s*Z_RC/(Z_L+Z_RC); //voltage division
   principle
14 I=V_s/(Z_L+Z_RC); //current through source and
   inductor
15 I_R=V_C/R;        //current through resistance
16 I_C=V_C/Z_C;      //current through capacitor
17 //cos(t)=cos(180-t)
18 disp(sqrt((real(V_C)^2)+(imag(V_C)^2)), 'peak value
   of Vc in volts')
19 disp((atan(imag(V_C)/real(V_C)))*180/%pi, 'phase
   angle of Vc in degrees')
20 ////result : V_C=Vcos(wt+phi) V-peak voltage
21 disp(I, 'current through source and inductor in
   amperes')
22 disp(I_R, 'current through resistance in amperes')
23 disp(I_C, 'current through capacitance in amperes')
24 disp('phasor diagram cannot be plotted')

```

---

### Scilab code Exa 5.6 example 6

```

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

```

```

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

```

---

### Scilab code Exa 5.7 example 7

```

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

```

```

6 V_s_phi=phi_v;          //phase angle of voltage source
7 R=100;
8 V_s=complex(V_s_max*cos(V_s_phi),V_s_max*sin(V_s_phi
   ));          //phasor of voltage source
9 X_L=%i*100;
10 X_C=-%i*100;
11 I_max=0.1414;         //peak value of current
12 I_phi=phi_i;         //phase angle of current
13 I=complex(I_max*cos(I_phi),I_max*sin(I_phi));
   //phasor of current
14 V_s_rms=V_s_max/sqrt(2);      //rms value of voltage
15 I_rms=I_max/sqrt(2);         //rms value of current
16 I_R_max=0.1;                //peak value
17 I_R_phi=-2*pi;              //phase angle
18 I_R=complex(I_R_max*cos(I_R_phi),I_R_max*sin(I_R_phi
   ));          //phasor of current
19 I_R_rms=I_R_max/sqrt(2);     //rms value
20 I_C_max=0.1;                //peak value
21 I_C_phi=-pi/2;              //phase angle
22 I_C=complex(I_C_max*cos(I_C_phi),I_C_max*sin(I_C_phi
   ));          //phasor current in capacitor
23 I_C_rms=I_C_max/sqrt(2);     //rms value
24 P=V_s_rms*I_rms*cos(phi);    //power by source
25 Q=V_s_rms*I_rms*sin(phi);    //reactive power by
   source
26 printf(" All the values in the textbook are
   approximated hence the values in this code differ
   from those of Textbook")
27 disp(P,'power delivered by source in watts')
28 disp(Q,'reactive power delivered by source in VARs')
29 //using complex power method
30 disp('Using complex power method:')
31 S=(1/2)*V_s*(I');            //complex power
32 P=real(S);
33 Q=imag(S);
34 disp(P,'power delivered by source in watts')
35 disp(Q,'reactive power delivered by source in VARs')
36 disp('we see that, in both the methods answers are

```

```

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

```

---

#### Scilab code Exa 5.8 example 8

```

1
2 clc
3 //initialisation of variables
4 clear
5 Vrms = 10^2 //V
6 Irms= 10^2 //amp
7 pf= 0.5
8 pf1= 0.7
9 r= 1.41
10 //CALCULATIONS
11 PA= Vrms*Irms*pf
12 QA= -sqrt((Vrms*Irms)^2-PA^2)/1000
13 a= acosd(pf1)
14 QB= PA*tand(a)/1000
15 P= 2*PA/1000
16 Q= QA+QB
17 o= atand(Q/P)
18 pf2= cosd(o)
19 A= o+69.18
20 S= sqrt(P^2+Q^2)
21 I= S*r

```

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

---

### Scilab code Exa 5.9 example 9

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

---

### Scilab code Exa 5.10 example 10

```
1 clc
2 //ex5.10
3 R=100;
```

```

4 V_s_max=100;          //peak value of voltage
5 V_s_phi=0;          //phase angle of voltage
6 V_s=complex(V_s_max*cos(V_s_phi),V_s_max*sin(V_s_phi
   ));          //phasor of voltage
7 Z_C=-%i*100;        //impedance of capacitance
8 I_s_max=1;          //peak value of current
9 I_s_phi=%pi/2;      //phase angle of current
10 I_s=complex(I_s_max*cos(I_s_phi),I_s_max*sin(I_s_phi
   ));          //phasor of current
11 //zeroing sources to find Z_t i.e., thevenin
   impedance
12 Z_t=1/((1/R)+(1/Z_C)); //R and Z_C are in
   parallel combination
13 //apply short-circuit to find I_sc i.e., short-
   circuit current
14 I_R=abs(V_s)/R;     //ohm's law
15 I_sc=I_R-I_s;       //applying KCL
16 V_t=I_sc*Z_t;       //thevenin voltage
17 printf(" All the values in the textbook are
   approximated hence the values in this code differ
   from those of Textbook")
18 disp('FOR THEVENIN CIRCUIT:')
19 disp('thevenin voltage')
20 disp(abs(V_t),'peak value of voltage in volts')
21 //cos(t)=cos(t-180)
22 disp(atan(imag(V_t)/real(V_t))*180/%pi,'phase angle
   in degrees')
23 disp('thevenin resistance')
24 disp(abs(Z_t),'peak value of resistance in ohms')
25 disp(atan(imag(Z_t)/real(Z_t))*180/%pi,'phase angle
   in degrees')
26 disp('FOR NORTON CIRCUIT:')
27 disp('norton current')
28 disp(abs(I_sc),'peak value of norton current in
   amperes')
29 disp(atan(imag(I_sc)/real(I_sc))*180/%pi,'phase
   angle in degrees')
30 disp('resistance')

```

```

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

```

---

### Scilab code Exa 5.11 example 11

```

1  clc
2  //ex5.11
3  //thevenin voltage
4  V_t_max=100;
5  V_t_phi=-%pi/2;
6  V_t=complex(V_t_max*cos(V_t_phi),V_t_max*sin(V_t_phi
    ));
7  //thevenin resistance
8  Z_t_max=70.71;
9  Z_t_phi=-%pi/4;
10 Z_t=complex(Z_t_max*cos(Z_t_phi),Z_t_max*sin(Z_t_phi
    ));
11 printf(" All the values in the textbook are
    approximated hence the values in this code differ
    from those of Textbook")
12 //a) Any complex load
13 disp('FOR ANY COMPLEX LOAD')
14 Z_load=Z_t';
15 I_a=V_t/(Z_t+Z_load); //ohm's law
16 I_a_rms=I_a/sqrt(2); //rms value
17 P_1=abs(I_a_rms)^2*real(Z_load); //power
18 disp(Z_load,'required complex load impedance')
19 disp(P_1,'power delivered to load in watts')
20 //b) purely resistive load
21 disp('FOR PURE RESISTIVE LOAD')
22 R_load=abs(Z_t);
23 I_b=V_t/(Z_t+R_load);
24 I_b_rms=I_b/sqrt(2);
25 P_2=abs(I_b_rms)^2*R_load;

```

```

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

```

---

### Scilab code Exa 5.12 example 12

```

1  clc
2  //ex5.12
3  V_Y=1000;      //line to neutral voltage
4  f=60;         //frequency
5  L=0.1;        //inductance
6  R=50;
7  W=2*%pi*f;    //angular frequency
8  Z=complex(R,W*L); //complex impedance
9  phi=atan(imag(Z)/real(Z));
10 //Balanced wye-wye calculations
11 V_an=complex(1000*cos(0),1000*sin(0));
12 V_bn=complex(1000*cos(-2*%pi/3),1000*sin(-2*%pi/3));
13 V_cn=complex(1000*cos(2*%pi/3),1000*sin(2*%pi/3));
14 I_aA=V_an/Z;
15 I_bB=V_bn/Z;
16 I_cC=V_cn/Z;
17 //line-line phasors
18 V_ab=V_an*sqrt(3)*complex(cos(%pi/6),sin(%pi/6));
19 V_bc=V_bn*sqrt(3)*complex(cos(%pi/6),sin(%pi/6));
20 V_ca=V_cn*sqrt(3)*complex(cos(%pi/6),sin(%pi/6));
21 I_L=abs(I_aA);
22 P=(3/2)*V_Y*I_L*cos(phi); //power
23 Q=(3/2)*V_Y*I_L*sin(phi); //reactive power
24 printf(" All the values in the textbook are
    approximated hence the values in this code differ
    from those of Textbook")
25 disp('LINE CURRENTS')
26 disp(I_aA, 'IaA=')
27 disp(I_bB, 'IbB=')
28 disp(I_cC, 'IcC=')

```



```

29 disp('LINE-LINE VOLTAGES')
30 disp(V_ab, 'Vab=')
31 disp(V_bc, 'Vbc=')
32 disp(V_ca, 'Vca=')
33 disp(P, 'POWER IN WATTS')
34 disp(Q, 'REACTIVE POWER IN VARs')
35 disp('the phasor diagram cannot be plotted')

```

---

### Scilab code Exa 5.13 example 13

```

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

```

```

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

```

---

#### Scilab code Exa 5.14 example 14

```

1 clc
2 //ex5.14
3 V_1=10^3*2.2*sqrt(2)*complex(cos(0), sin(0));
4 V_2=10^3*2*sqrt(2)*complex(cos(-%pi/18), sin(-%pi/18)
    );
5 //writing matrix form of mesh current equaions
    obtained by KVL
6 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
7 V=[2200*sqrt(2); -2000*sqrt(2)*complex(cos(-%pi/18),
    sin(-%pi/18))]; //voltage matrix
8 I=Z\V; //current matrix
9 S_1=(1/2)*V_1*((I(1,:))'); //complex power

```

```
10 P_1=real(S_1);          //power
11 Q_1=imag(S_1);         //reactive power
12 printf(" All the values in the textbook are
    approximated hence the values in this code differ
    from those of Textbook")
13 disp(P_1,'real power supplied by V1 in watts')
14 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

```
1  clc
2  //ex 6.1
3  // given  $V_{in}(t)=2\cos(2000*\%pi*t+A)$ ,  $A=40*\%pi/180$ 
4   $w=2000*\%pi$ ; //omega
5   $f=w/(2*\%pi)$ ; //frequency
6   $A=40*\%pi/180$ ; //40 degrees in radians
7  //equation of straight line of H_magnitude vs f is  $x$ 
    $+1000*y-4000=0$ 
8   $H_{max}=(4000-f)/1000$ ; //magnitude of H(transfer
   function)
9  //equation of straight line of H_phase angle vs f is
    $6000*y=\%pi*x$  (phase angle in radians)
10  $H_{phi}=\%pi*f/6000$ ; //phase angle of H
11  $H=H_{max}*\text{complex}(\cos(H_{phi}),\sin(H_{phi}))$ ;
12  $V_{in}=2*\text{complex}(\cos(A),\sin(A))$ ; //input voltage
   phasor
13  $V_{out}=H*V_{in}$ ; //output voltage phasor
14  $V_{out\_R}=\text{real}(V_{out})$ ; //real part
15  $V_{out\_I}=\text{imag}(V_{out})$ ; //imaginary part
```

```

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

```

---

### Scilab code Exa 6.2 example 2

```

1  clc
2  //ex6.2
3  //given V_in(t)=3+2*cos(2000*%pi*t)+cos(4000*%pi*t-A
    ), A=70*%pi/180
4  //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)
5
6  //first component V_1
7  V_in_1=3;
8  f_1=0;           //as omega is zero
9  //equation of straight line of H_magnitude vs f is x
    +1000*y-4000=0
10 H_1_max=(4000-f_1)/1000;           //magnitude of H(
    traansfer function)
11 //equation of straight line of H_phase angle vs f is
    6000*y=%pi*x (phase angle in radians)
12 H_1_phi=%pi*f_1/6000;           //phase angle of H
13 H_1=H_1_max*complex(cos(H_1_phi),sin(H_1_phi));
14 V_out_1=H_1*V_in_1;
15 V_out_1_R=real(V_out_1);           //real part
16 V_out_1_I=imag(V_out_1);           //imaginary part
17 V_out_1_max=sqrt((V_out_1_R^2)+(V_out_1_I^2));
    //peak value
18 V_out_1_phi=atan(V_out_1_I/V_out_1_R);           //phase
    angle

```

```

19
20 //second component V_in_2
21 V_in_2=2*complex(cos(0),sin(0)); //V_in_2
    phasor
22 w=2000*%pi; //omega
23 f_2=w/(2*%pi); //frequency
24 //equation of straight line of H_magnitude vs f is x
    +1000*y-4000=0
25 H_2_max=(4000-f_2)/1000; //magnitude of H(
    traansfer function)
26 //equation of straight line of H_phase angle vs f is
    6000*y=%pi*x (phase angle in radians)
27 H_2_phi=%pi*f_2/6000; //phase angle of H
28 H_2=H_2_max*complex(cos(H_2_phi),sin(H_2_phi));
29 V_out_2=H_2*V_in_2;
30 V_out_2_R=real(V_out_2); //real part
31 V_out_2_I=imag(V_out_2); //imaginary part
32 V_out_2_max=sqrt((V_out_2_R^2)+(V_out_2_I^2));
    //peak value
33 V_out_2_phi=atan(V_out_2_I/V_out_2_R); //phase
    angle
34
35 //third component
36 A=-70*%pi/180; //-70 degrees in radians
37 V_in_3=complex(cos(A),sin(A)); //V_in_3 phasor
38 w=4000*%pi; //omega
39 f_3=w/(2*%pi); //frequency
40 //equation of straight line of H_magnitude vs f is x
    +1000*y-4000=0
41 H_3_max=(4000-f_3)/1000; //magnitude of H(
    traansfer function)
42 //equation of straight line of H_phase angle vs f is
    6000*y=%pi*x (phase angle in radians)
43 H_3_phi=%pi*f_3/6000; //phase angle of H
44 H_3=H_3_max*complex(cos(H_3_phi),sin(H_3_phi));
45 V_out_3=H_3*V_in_3;
46 V_out_3_R=real(V_out_3); //real part
47 V_out_3_I=imag(V_out_3); //imaginary part

```

```

48 V_out_3_max=sqrt((V_out_3_R^2)+(V_out_3_I^2));
    //peak value
49 V_out_3_phi=atan(V_out_3_I/V_out_3_R);    //phase
    angle
50
51 disp('Output voltage is Vout1+Vout2+Vout3 where')
52 disp('')
53 disp('FOR Vout1:')
54 disp(V_out_1_max,'peak value in volts')
55 disp(V_out_1_phi*180/%pi,'phase angle in degrees')
56 disp(f_1,'with frequency in hertz')
57 disp('')
58 disp('FOR Vout2:')
59 disp(V_out_2_max,'peak value in volts')
60 disp(V_out_2_phi*180/%pi,'phase angle in degrees')
61 disp(f_2,'with frequency in hertz')
62 disp('')
63 disp('FOR Vout3:')
64 disp(V_out_3_max,'peak value in volts')
65 disp(V_out_3_phi*180/%pi,'phase angle in degrees')
66 disp(f_3,'with frequency in hertz')

```

---

### Scilab code Exa 6.3 example 3

```

1  clc
2  //ex6.3
3  R=1000/(2*%pi);    //resistance
4  C=10*10^-6;    //capacitance
5  f_B=1/(2*%pi*R*C);    //half-power frequency
6  //the three parts of V_in are V_1=5*cos(20*%pi*t)+5*
    cos(200*%pi*t)+5*cos(2000*%pi*t)
7
8  //first component V_in_1
9  V_in_1=5*complex(cos(0),sin(0));    //V_in_1
    phasor

```

```

10 w_1=20*%pi;           //omega
11 f_1=w_1/(2*%pi);      //frequency
12 H_1=1/(1+%i*(f_1/f_B)); //transfer function
13 V_out_1=H_1*V_in_1;
14 V_out_1_R=real(V_out_1); //real part
15 V_out_1_I=imag(V_out_1); //imaginary part
16 V_out_1_max=sqrt((V_out_1_R^2)+(V_out_1_I^2));
    //peak value
17 V_out_1_phi=atan(V_out_1_I/V_out_1_R); //phase
    angle
18
19 //second component V_in_2
20 V_in_2=5*complex(cos(0),sin(0)); //V_in_2
    phasor
21 w_2=200*%pi;         //omega
22 f_2=w_2/(2*%pi);     //frequency
23 H_2=1/(1+%i*(f_2/f_B)); //transfer function
24 V_out_2=H_2*V_in_2;
25 V_out_2_R=real(V_out_2); //real part
26 V_out_2_I=imag(V_out_2); //imaginary part
27 V_out_2_max=sqrt((V_out_2_R^2)+(V_out_2_I^2));
    //peak value
28 V_out_2_phi=atan(V_out_2_I/V_out_2_R); //phase
    angle
29
30 //third component V_in_3
31 V_in_3=5*complex(cos(0),sin(0)); //V_in_3
    phasor
32 w_3=2000*%pi;        //omega
33 f_3=w_3/(2*%pi);     //frequency
34 H_3=1/(1+%i*(f_3/f_B)); //transfer function
35 V_out_3=H_3*V_in_3;
36 V_out_3_R=real(V_out_3); //real part
37 V_out_3_I=imag(V_out_3); //imaginary part
38 V_out_3_max=sqrt((V_out_3_R^2)+(V_out_3_I^2));
    //peak value
39 V_out_3_phi=atan(V_out_3_I/V_out_3_R); //phase
    angle

```



```

40
41 printf(" All the values in the textbook are
    approximated, hence the values in this code
    differ from those of Textbook")
42 disp('Output voltage is Vout1+Vout2+Vout3 where')
43 disp('')
44 disp('FOR Vout1:')
45 disp(V_out_1_max, 'peak value in volts')
46 disp(V_out_1_phi*180/%pi, 'phase angle in degrees')
47 disp(f_1, 'with frequency in hertz')
48 disp('')
49 disp('FOR Vout2:')
50 disp(V_out_2_max, 'peak value in volts')
51 disp(V_out_2_phi*180/%pi, 'phase angle in degrees')
52 disp(f_2, 'with frequency in hertz')
53 disp('')
54 disp('FOR Vout3:')
55 disp(V_out_3_max, 'peak value in volts')
56 disp(V_out_3_phi*180/%pi, 'phase angle in degrees')
57 disp(f_3, 'with frequency in hertz')
58 //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

```

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

```

```

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

```

---

### Scilab code Exa 6.5 example 5

```

1 clc
2 // ex6.5
3 V_s = 1 * complex(cos(0), sin(0));
4 L = 159.2 * 10^-3;
5 R = 100;
6 C = 0.1592 * 10^-6;
7 f_o = 1 / (2 * %pi * sqrt(L * C)); // resonant frequency
8 Q_s = 2 * %pi * f_o * L / R; // quality factor
9 B = f_o / Q_s; // Bandwidth
10 // Approximate half-power frequencies are
11 f_H = f_o + (B / 2);
12 f_L = f_o - (B / 2);
13 // At resonance
14 Z_L = %i * 2 * %pi * f_o * L; // impedance of inductance
15 Z_C = -%i / (2 * %pi * f_o * C); // impedance of
    capacitance
16 Z_s = R + Z_L + Z_C;
17 I = V_s / Z_s; // phasor current
18 // voltages across different elements are
19 // for resistance
20 V_R = R * I;
21 V_R_R = real(V_R); // real part
22 V_R_I = imag(V_R); // imaginary part
23 V_R_max = sqrt((V_R_R^2) + (V_R_I^2)); // peak value
24 V_R_phi = atan(V_R_I / V_R_R); // phase angle
25 // for inductance

```

```

26 V_L=Z_L*I;
27 V_L_R=real(V_L);           //real part
28 V_L_I=imag(V_L);          //imaginary part
29 V_L_max=sqrt((V_L_R^2)+(V_L_I^2)); //peak value
30 //Z_L is pure imaginary ==> V_L is pure imaginary
   which means V_L_phi can be +or- %pi/2
31 if ((V_L/%i)==abs(V_L)) then
32     V_L_phi=%pi/2
33 elseif ((V_L/%i)==-abs(V_L)) then
34     V_L_phi=-%pi/2
35 end
36
37 //for capacitance
38 V_C=Z_C*I;
39 V_C_R=real(V_C);           //real part
40 V_C_I=imag(V_C);          //imaginary part
41 V_C_max=sqrt((V_C_R^2)+(V_C_I^2)); //peak value
42 //Z_C is pure imaginary ==> V_C is pure imaginary
   which means V_C_phi can be +or- %pi/2
43 if ((V_C/%i)==abs(V_C)) then
44     V_C_phi=%pi/2
45 elseif ((V_C/%i)==-abs(V_C)) then
46     V_C_phi=-%pi/2
47 end
48
49 disp('Phasor voltage across Resistance')
50 disp(V_R_max,'peak value in volts')
51 disp(V_R_phi*180/%pi,'phase angle in degrees')
52 disp('')
53 disp('Phasor voltage across Inductance')
54 disp(V_L_max,'peak value in volts')
55 disp(V_L_phi*180/%pi,'phase angle in degrees')
56 disp('')
57 disp('Phasor voltage across Capacitance')
58 disp(V_C_max,'peak value in volts')
59 disp(V_C_phi*180/%pi,'phase angle in degrees')
60 disp('Phasor diagram cannot be drawn here')

```

---

## Scilab code Exa 6.6 example 6

```
1  clc
2  //ex6.6
3  R=10*10^3;
4  f_o=1*10^6;
5  B=100*10^3;
6  I=10^-3*complex(cos(0),sin(0));
7  Q_p=f_o/B;          //quality factor
8  L=R/(2*%pi*f_o*Q_p);
9  C=Q_p/(2*%pi*f_o*R);
10 //At resonance
11 V_out=I*R;
12 Z_L=%i*2*%pi*f_o*L;
13 Z_C=-%i/(2*%pi*f_o*C);
14
15 //across resistance
16 I_R=V_out/R;
17 I_R_R=real(I_R);    //real part
18 I_R_I=imag(I_R);   //imaginary part
19 I_R_max=sqrt((I_R_R^2)+(I_R_I^2)); //peak value
20 I_R_phi=atan(I_R_I/I_R_R); //phase angle
21
22 //across inductance
23 I_L=V_out/Z_L;
24 I_L_R=real(I_L);    //real part
25 I_L_I=imag(I_L);   //imaginary part
26 I_L_max=sqrt((I_L_R^2)+(I_L_I^2)); //peak value
27 //Z_L is pure imaginary ==> V_L is pure imaginary
   which means V_L_phi can be +or- %pi/2
28 if ((I_L/%i)==abs(I_L)) then
29     I_L_phi=%pi/2
30 elseif ((I_L/%i)==-abs(I_L)) then
31     I_L_phi=-%pi/2
```

```

32 end
33
34 //across capacitor
35 I_C=V_out/Z_C;
36 I_C_R=real(I_C);           //real part
37 I_C_I=imag(I_C);          //imaginary part
38 I_C_max=sqrt((I_C_R^2)+(I_C_I^2)); //peak value
39 //Z_C is pure imaginary ==> V_C is pure imaginary
   which means V_C_phi can be +or- %pi/2
40 if ((I_C/%i)==abs(I_C)) then
41     I_C_phi=%pi/2
42 elseif ((I_C/%i)==-abs(I_C)) then
43     I_C_phi=-%pi/2
44 end
45
46 disp('Current phasor across Resistance')
47 disp(I_R_max,'peak value in amperes')
48 disp(I_R_phi*180/%pi,'phase angle in degrees')
49 disp('')
50 disp('Current phasor across Inductance')
51 disp(I_L_max,'peak value in amperes')
52 disp(I_L_phi*180/%pi,'phase angle in degrees')
53 disp('')
54 disp('current phasor across capacitance')
55 disp(I_C_max,'peak value in amperes')
56 disp(I_C_phi*180/%pi,'phase angle in degrees')
57 disp('Phasor diagram cannot be drawn here')

```

---

#### Scilab code Exa 6.7 example 7

```

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

```

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

---

# Chapter 7

## Logic circuits

Scilab code Exa 7.1 example 1

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

---

Scilab code Exa 7.2 example 2

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

```

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

```

---

#### Scilab code Exa 7.3 example 3

```

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

```

---

#### Scilab code Exa 7.4 example 4

```

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

```



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

---

#### Scilab code Exa 7.5 example 5

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

---

#### Scilab code Exa 7.6 example 6

```
1 clc
2 //ex7.6
3 //Given 11110110.1(binary)
4 //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)
5 //And the corresponding numbers for 011,110,110 and
   100 in octal system are 3,6,6 and 4
6 disp('The octal representation of 11110110.1(binary)
   is 366.4')
```

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

---

# Chapter 9

## Computer based instrumentation diodes

Scilab code Exa 9.1 example 1

```
1 clc
2 //ex9.1
3 P=0.1; //system sensitivity change percent
4 R_th_U=15*10^3; //thevenin resistance upper
  limit
5 R_th_L=5*10^3; //thevenin resistance lower
  limit
6 //The required inequality is  $V_{\text{sensor}}*R_{\text{in}}/(R_{\text{th\_U}}+R_{\text{in}}) \geq (1-P/100)*V_{\text{sensor}}*R_{\text{in}}/(R_{\text{th\_L}}+R_{\text{in}})$ ,
  cancelling same terms on both sides of inequality
  and calculating  $R_{\text{in}}$  by taking equality we'll
  get minimum value of  $R_{\text{in}} \implies R_{\text{th\_L}}+R_{\text{in}}=(1-P/100)*(R_{\text{th\_U}}+R_{\text{in}})$  which gives
7 R_in=((1-P/100)*R_th_U)-R_th_L)*100/P;
8 disp(R_in/1000, 'The minimum value of Rin required in
  Kilo-ohms')
```

---

# Chapter 10

## Diodes

Scilab code Exa 10.1 example 1

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

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

---

### Scilab code Exa 10.2 example 2

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

---

### Scilab code Exa 10.3 example 3

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

---

### Scilab code Exa 10.4 example 4

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

```

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

```

---

#### Scilab code Exa 10.5 example 5

```

1  clc
2  //ex10.5
3  V_1=10;
4  V_2=3;
5  R_1=4*10^3;
6  R_2=6*10^3;
7  //1) analysis by assuming D1 off and D2 on
8  I_D_2=V_2/R_2;    //ohm's law
9  //applying KVL
10 V_D_1=7;    //contradiction to 'D1 is off'
11 //this assumption is not correct
12
13 //2) analysis by assuming D1 on and D2 off
14 I_D_1=V_1/R_1;    //ohm's law
15 //applying KVL
16 V_D_2=-V_1+V_2+I_D_1*R_1;
17 //we get V_D_2 which is consistent
18 disp('correct assumption is D2 off and D1 on')

```

---

Scilab code Exa 10.7 example 7

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

---



# Chapter 11

## Amplifiers Specifications and external characteristics

Scilab code Exa 11.1 example 1

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

```

    approximated, hence the values in this code
    differ from those of Textbook”)
17 disp(A_v, 'Voltage gain Av')
18 disp(A_vs, 'Voltage gain Avs')
19 disp(A_i, 'Current gain')
20 disp(G, 'Power gain')

```

---

### Scilab code Exa 11.2 example 2

```

1  clc
2  //ex11.2
3  R_i_1=10^6;
4  R_o_1=500;
5  R_i_2=1500;
6  R_o_2=100;
7  R_L=100;
8  A_voc_1=200;
9  A_voc_2=100;
10 //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))
11 A_v_1=A_voc_1*(R_i_2/(R_i_2+R_o_1));
12 A_v_2=A_voc_2*(R_L/(R_L+R_o_2));
13 A_i_1=A_v_1*R_i_1/R_i_2;
14 A_i_2=A_v_2*R_i_2/R_L;
15 A_i=A_i_1*A_i_2;
16 G_1=A_v_1*A_i_1;
17 G_2=A_v_2*A_i_2;
18 G=G_1*G_2;
19 disp(A_i_1, 'Current gain of first stage')
20 disp(A_i_2, 'Current gain of second stage')
21 disp(A_v_1, 'Voltage gain of first stage')
22 disp(A_v_2, 'Voltage gain of second stage')
23 disp(G_1, 'Power gain of first stage')
24 disp(G_2, 'Power gain of second stage')
25 disp(G, 'Overall power gain')

```

---

Scilab code Exa 11.3 example 3

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

---

Scilab code Exa 11.4 example 4

```
1  clc
2  //ex11.4
3  V_AA=15;
```

```

4 V_BB=15;
5 V_i=1*10^-3;
6 I_A=1;
7 I_B=0.5;
8 R_L=8;
9 R_o=2;
10 R_i=100*10^3;
11 A_voc=10^4;
12 P_i=V_i^2/R_i;
13 V_o=A_voc*V_i*(R_L/(R_L+R_o));
14 P_o=V_o^2/R_L;
15 P_s=V_AA*I_A+V_BB*I_B;
16 P_d=P_s+P_i-P_o;
17 n=P_o*100/P_s;
18 printf(" All the values in the textbook are
        approximated, hence the values in this code
        differ from those of Textbook")
19 disp(P_i*10^12, 'Input power in picowatts')
20 disp(P_o, 'Output power in watts')
21 disp(P_s, 'Supply power in watts')
22 disp(P_d, 'Dissipated power in watts')
23 disp(n, 'Efficiency of the amplifier')

```

---

### Scilab code Exa 11.5 example 5

```

1 clc
2 //ex11.5
3 R_i=1*10^3;
4 R_o=100;
5 A_voc=100;
6 //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))
7 A_isc=A_voc*(R_i/R_o);
8 disp('The resulting current-amplifier is with an:')
9

```

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

---

#### Scilab code Exa 11.6 example 6

```
1 clc
2 //ex11.6
3 R_i=1*10^3;
4 R_o=100;
5 A_voc=100;
6 //i_osc=A_voc*V_i/R_o and G_msc=i_osc/V_i gives
   G_msc=A_voc/R_o
7 G_msc=A_voc/R_o;
8 disp('The resulting transconductance model is with
   an:')
9
10 disp(R_i, 'input resistance in ohms')
11 disp(R_o, 'output resistance in ohms')
12 disp(G_msc, 'and transconductance in siemens')
```

---

#### Scilab code Exa 11.7 example 7

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

```
9
10 disp(R_i, 'input resistance in ohms')
11 disp(R_o, 'output resistance in ohms')
12 disp(R_moc, 'and transresistance in ohms')
```

---

#### Scilab code Exa 11.8 example 8

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

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

```

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

```

---

#### Scilab code Exa 11.10 example 10

```

1 clc
2 //ex11.10
3 t=[0:0.000001:0.002];
4 V_i=3*cos(2000*%pi*t)-2*cos(6000*%pi*t);
5 //for A
6 A_1000_A_peak=10;
7 A_1000_A_phi=0;
8 A_3000_A_peak=10;
9 A_3000_A_phi=0;
10 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);
11 //for B
12 A_1000_B_peak=10;
13 A_1000_B_phi=-%pi/4;
14 A_3000_B_peak=10;
15 A_3000_B_phi=-3*%pi/4;
16 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);
17 //for C
18 A_1000_C_peak=10;

```

```

19 A_1000_C_phi=-%pi/4;
20 A_3000_C_peak=10;
21 A_3000_C_phi=-%pi/4;
22 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);
23 disp('VoA(t)=30cos(2000%pit)-10cos(6000%pit)')
24 disp('VoB(t)=30cos(2000%pit-%pi/4)-10cos(6000%pit-3
    %pi/4)')
25 disp('VoC(t)=30cos(2000%pit-%pi/4)-10cos(6000%pit-
    %pi/4)')
26 subplot(221)
27 xtitle('Output-voltage vs time for A','time in ms','
    Output-voltage for A in volts')
28 plot(t*10^3,V_o_A)
29 subplot(222)
30 xtitle('Output-voltage vs time for B','time in ms','
    Output voltage for B in volts')
31 plot(t*10^3,V_o_B)
32 subplot(223)
33 xtitle('Output-voltage vs time for C','time in ms','
    Output voltage for C in volts')
34 plot(t*10^3,V_o_C)

```

---

### Scilab code Exa 11.11 example 11

```

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

```



```

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

```

---

### Scilab code Exa 11.12 example 12

```

1  clc
2  //initialisation of variables
3  Rin= 1 //Mohms
4  Rs1= 100 //kohms
5  Rs2= 100 //kohms
6  Ioff= 84 //Amperes
7  Voff= 5 //mV
8  //CALCULARIONS
9  Vioff= Rin*Ioff*10-3*(Rs1+Rs2)/(2*(Rin+10-3*(Rs1+
    Rs2)))
10 Vvoff= Voff*Rin/(Rin+10-3*(Rs1+Rs2))
11 //RESULTS
12 printf ( ' Vioff = %.f mV ',Vioff)
13 printf ( '\n Vvoff = %.2f mV ',Vvoff)

```

---

# Chapter 12

## Field effect transistors

Scilab code Exa 12.1 example 1

```
1  clc
2  //initialisation of variables
3  K= 2
4  VGS1= 5 //V
5  VGS2= 4 //V
6  VGS3= 3 //V
7  VGS4= 2 //V
8  //CALCULATIONS
9  id1= K*(VGS1-2)^2
10 id2= K*(VGS2-2)^2
11 id3= K*(VGS3-2)^2
12 id4= K*(VGS4-2)^2
13 //RESULTS
14 printf ('iD = %.f V ',id1)
15 printf ('\n iD = %.f V ',id2)
16 printf ('\n iD = %.f V ',id3)
17 printf ('\n iD = %.f V ',id4)
```

---

Scilab code Exa 12.2 example 2

```

1  clc
2  //initialisation of variables
3  KP= 50 //uA/V62
4  Vto= 2 //V
5  L= 10 //um
6  W= 400 //um
7  Vdd= 20 //mV
8  R2= 1 //kohms
9  R1= 3 //ohms
10 Rd= 11.5 //Mohms
11 Rs= 1 //kohms
12 V= 4 //mV
13 //CALCULATIONS
14 K= W*KP/(2*L*10^3)
15 Vg= Vdd*R2/(R1+R2)
16 clc
17 x=poly(0,"x")
18 vec=roots(x^2-3.630*x+2.148)
19 VGSQ= vec(2)
20 IDQ= K*(VGSQ-Vto)^2
21 VDSQ= Vdd+V+L-(Rd+Rs)*IDQ
22 //RESULTS
23 printf ( 'VDSQ = %.1 f V ',VDSQ)

```

---

### Scilab code Exa 12.3 example 3

```

1  clc
2  //initialisation of variables
3  VGSQ= 3.5 //V
4  VDSQ= 10 //V
5  id1= 10.7 //mA
6  id2= 4.7 //mA
7  dvgs= 1 //V
8  id3= 8 //mA
9  id4= 6.7 //mA

```

```

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

```

---

### Scilab code Exa 12.5 example 5

```

1 clc
2 //initialisation of variables
3 RL= 1 //kohms
4 R1= 2 //Mohms
5 R2= 2 //Mohms
6 KP= 50 //uA/V^2
7 L= 2 //um
8 W= 160 //um
9 Vto= 1 //V
10 IDQ= 10 //mA
11 VG= 7.5 //V
12 //CALCULATIONS
13 K= W*KP/(2*L*10^3)
14 VGSQ= sqrt(IDQ/K)+Vto
15 VS= VG-VGSQ
16 RS= VS*10^3/IDQ
17 gm= sqrt(2*KP/10^3)*sqrt(W/L)*sqrt(IDQ)
18 RL1= 1/(1/(RS)+(1/(RL*10^3)))
19 Av= gm*RL1*10^-3/(1+gm*RL1*10^-3)
20 Rin= 1/((1/R1)+(1/R2))
21 Ro= 1/(gm*10^-3+(1/RS))
22 Ai= Av*Rin/RL
23 G= Av*Ai*10^3
24 //RESULTS
25 printf ('G = %.1f ',G)

```



# Chapter 13

## Bipolar junction transistors

Scilab code Exa 13.1 example 1

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

---

Scilab code Exa 13.2 example 2

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

```

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

```

---

#### Scilab code Exa 13.4 example 4

```

1 clc
2 //ex13.4
3 V_CC=15;
4 B=100;      //beta value
5 R_B=200*10^3;
6 R_C=1*10^3;
7 //we proceed in such a way that the required values
    will be displayed according to the satisfied
    condition of the below three cases
8
9 //a)cut-off region

```

```

10 V_BE=15;          //no voltage drop across R_B in cut-
    off state
11 V_CE=15;          //no voltage drop across R_C in cut-
    off state
12 i_C=0;           //no collector current flows as there is
    no voltage drop
13 i_B=0;           //no base current flows as there is no
    voltage drop
14 if(V_BE<0.5) then,      //cut-off condition
15     disp(i_C,'collector current in amperes')
16     disp(V_CE,'collector to emitter voltage in volts
    ')
17     end
18
19 //b)saturation region
20 V_BE=0.7;         //base to emitter voltage in
    saturation state
21 V_CE=0.2;         //collector to emitter voltage in
    saturation state
22 i_C=(V_CC-V_CE)/R_C;      //collector current
23 i_B=(V_CC-V_BE)/R_B;      //base current
24 if((B*i_B>i_C)&(i_B>0)) then,      //saturation
    state conditions
25     disp(i_C,'collector current in amperes')
26     disp(V_CE,'collector to emitter voltage in
    volts')
27     end
28
29 //c)active region
30 V_BE=0.7;         //base to emitter voltage in active
    state
31 i_B=(V_CC-V_BE)/R_B;      //base current
32 i_C=B*i_B;          //collector current in active state
33 V_CE=V_CC-i_C*R_C;      //collector to emitter
    voltage
34 if((V_CE>0.2)&(i_B>0)) then,      //active state
    conditions
35     disp(i_C,'collector current in amperes')

```



```

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

```

---

### Scilab code Exa 13.5 example 5

```

1  clc
2  //ex13.5
3  R_B=200*10^3;
4  R_C=1*10^3;
5  V_CC=15;
6  B=300;      //beta value
7  //we proceed in such a way that the required values
   will be displayed according to the satisfied
   condition of the below three cases
8
9  //a) active region
10 V_BE=0.7;    //base to emitter voltage in active
   state
11 i_B=(V_CC-V_BE)/R_B;    //base current
12 i_C=B*i_B;    //collector current in active state
13 V_CE=V_CC-i_C*R_C;    //collector to emitter
   voltage
14 if((V_CE>0.2)&(i_B>0)) then,    //active state
   conditions
15     disp(i_C,'collector current in amperes')
16     disp(V_CE,'collector to emitter voltage in
           volts')
17     end
18
19 //b) saturation region
20 V_BE=0.7;    //base to emitter voltage in
   saturation state
21 V_CE=0.2;    //collector to emitter voltage in
   saturation state

```

```

22 i_C=(V_CC-V_CE)/R_C;      //collector current
23 i_B=(V_CC-V_BE)/R_B;      //base current
24 if((B*i_B>i_C)&(i_B>0)) then,      //saturation
    state conditions
25     disp(i_C,'collector current in amperes')
26     disp(V_CE,'collector to emitter voltage in
        volts')
27     end
28
29 //c)cut-off region
30 V_BE=15;      //no voltage drop across R_B in cut-
    off state
31 V_CE=15;      //no voltage drop across R_C in cut-
    off state
32 i_C=0;      //no collector current flows as there is
    no voltage drop
33 i_B=0;      //no base current flows as there is no
    voltage drop
34 if(V_BE<0.5) then,      //cut-off condition
35     disp(i_C,'collector current in amperes')
36     disp(V_CE,'collector to emitter voltage in volts
        ')
37     end

```

---

### Scilab code Exa 13.6 example 6

```

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

```

```

9  printf(" All the values in the textbook are
    Approximated hence the values in this code differ
    from those of Textbook")
10
11 //a)B=100
12 disp('For beta B=100:')
13 B=100;          //beta value
14 i_B=i_E/(B+1); //base current
15 i_C=B*i_B;     //collector current
16 V_CE=V_CC-i_C*R_C-i_E*R_E; //collector to
    emitter voltage
17 disp(i_C,'collector current in amperes')
18 disp(V_CE,'collector to emitter voltage in volts')
19
20 //b)B=300
21 disp('For beta B=300:')
22 B=300;          //beta value
23 i_B=i_E/(B+1); //base current
24 i_C=B*i_B;     //collector current
25 V_CE=V_CC-i_C*R_C-i_E*R_E; //collector to
    emitter voltage
26 disp(i_C,'collector current in amperes')
27 disp(V_CE,'collector to emitter voltage in volts')

```

---

### Scilab code Exa 13.7 example 7

```

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

```

```

10 V_B=V_CC*R_2/(R_1+R_2);          //thevenin voltage
11 printf(" All the values in the textbook are
    Approximated hence the values in this code differ
    from those of Textbook")
12
13 //a)B=100
14 disp('For beta B=100:')
15 B=100;          //beta value
16 i_B=(V_B-V_BE)/(R_B+(B+1)*R_E);    //base current
17 i_C=B*i_B;      //collector current
18 i_E=i_B+i_C;    //emitter current
19 V_CE=V_CC-i_C*R_C-i_E*R_E;        //collector to
    emitter voltage
20 disp(i_C,'collector current in amperes')
21 disp(V_CE,'collector to emitter voltage in volts')
22
23 //b)B=300
24 disp('For beta B=300:')
25 B=300;          //beta value
26 i_B=(V_B-V_BE)/(R_B+(B+1)*R_E);    //base current
27 i_C=B*i_B;      //collector current
28 i_E=i_B+i_C;    //emitter current
29 V_CE=V_CC-i_C*R_C-i_E*R_E;        //collector to
    emitter voltage
30 disp(i_C,'collector current in amperes')
31 disp(V_CE,'collector to emitter voltage in volts')

```

---

### Scilab code Exa 13.8 example 8

```

1 clc
2 //ex13.8
3 V_CC=15;
4 V_BE=0.7;
5 B=100;          //beta value
6 R_1=10*10^3;

```

```

7 R_2=5*10^3;
8 R_L_1=2*10^3;           //R_L is taken as R_L_1
9 R_C=1*10^3;
10 R_E=1*10^3;
11 V_T=26*10^-3;         //thermal voltage
12 //from the analysis of the previous example we have
    the the values of i_C_Q and V_CE
13 i_C_Q=4.12*10^-3;
14 V_CE=6.72;
15 r_pi=(B*V_T)/i_C_Q;
16 R_B=1/((1/R_1)+(1/R_2));           //thevenin resistance
17 R_L_2=1/((1/R_L_1)+(1/R_C));       //R_L' is taken as
    R_L_2
18 A_v=-(R_L_2*B)/r_pi;           //voltage gain
19 A_voc=-(R_C*B)/r_pi;           //open circuit voltage
    gain
20 Z_in=1/((1/R_B)+(1/r_pi));       //input impedance
21 A_i=(A_v*Z_in)/R_L_1;           //current gain
22 G=A_i*A_v;                       //power gain
23 Z_o=R_C                           //output impedance
24 //assume f=1hz
25 f=1;
26 t=0:0.0005:3;
27 V_in=0.001*sin(2*%pi*f*t);
28 V_o=-(V_in*R_L_2*B)/r_pi;
29 subplot(121)
30 xtitle('Input voltage vs time','time','input voltage
    ')
31 plot(t,V_in)
32 subplot(122)
33 xtitle('output voltage vs time','time','output
    voltage')
34 plot(t,V_o)
35 //In the graph, notice the phase inversion between
    input and output voltages
36 printf(" All the values in the textbook are
    Approximated hence the values in this code differ
    from those of Textbook")

```

```

37 disp(A_v, 'voltage gain')
38 disp(A_voc, 'open circuit voltage gain')
39 disp(Z_in, 'input impedance in ohms')
40 disp(A_i, 'current gain')
41 disp(G, 'power gain')
42 disp(Z_o, 'output impedance in ohms')

```

---

### Scilab code Exa 13.9 example 9

```

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

```

```

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

```

---

# Chapter 14

## Operational Amplifiers

Scilab code Exa 14.5 example 5

```
1  clc
2  //initialisation of variables
3  ADOL= 10^5
4  ADOL1= 10
5  dc= 20
6  dc1= 10
7  f= 40 //kHz
8  //CALCULATIONS
9  ADOL2= dc*log(ADOL)
10 ADOL3= dc*log10(ADOL1)
11 f1= ADOL1*f
12 //RESULTS
13 printf ('A0CL = %.f dB ',ADOL3)
14 printf ('\n frequency = %.f kHz ',f1)
```

---

Scilab code Exa 14.6 example 6

```
1  clc
```



```

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

```

---

#### Scilab code Exa 14.7 example 7

```

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

```

```
22 V_o_osI_max=R_2*I_os_max/2;
23 V_o_osI_min=-R_2*I_os_max/2;
24
25 V_o_max=V_o_osV_max+V_o_bias_max+V_o_osI_max;
    //maximum output volage
26 V_o_min=V_o_osV_min+V_o_bias_min+V_o_osI_min;
    //minimum output voltage
27 disp(V_o_max*10^3,'Maximum output voltage in milli-
    volts ')
28 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

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

```

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

```

---

### Scilab code Exa 15.5 example 5

```

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

```

```

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

```

---

### Scilab code Exa 15.6 example 6

```

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

```

```

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

```

---

### Scilab code Exa 15.7 example 7

```

1 clc
2 //ex15.7
3 N=500;      //number of turns of coil

```

```

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

```

---

#### Scilab code Exa 15.8 example 8

```

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

```

---

#### Scilab code Exa 15.9 example 9

```

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

```

---

#### Scilab code Exa 15.10 example 10

```

1 clc
2 //ex15.10
3 V_1_rms=110;
4 R_L=10;
5 tr=5;      //turns ratio (N1/N2)
6 V_2_rms=V_1_rms/tr; //primary and secondary
    voltage relation
7 //a)open switch
8 disp('OPEN switch')
9 disp(V_1_rms,'Primary voltage in volts')
10 disp(V_2_rms,'Secondary voltage in volts')
11 //As switch is open, current in second winding is 0
    which implies the current in primary coil to be 0
    (ideal transformer condition)
12 disp(0,'Current in primary winding in amperes')
13 disp(0,'Current in secondary winding in amperes')
14 //b)closed switch
15 disp('CLOSED switch')
16 I_2_rms=V_2_rms/R_L; //ohm's law
17 I_1_rms=I_2_rms/tr; //ideal transformer
    condition
18 disp(V_1_rms,'Primary voltage in volts')

```



```

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

```

---

### Scilab code Exa 15.11 example 11

```

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

```

```

    from those of Textbook”)
24 //we take real parts of angles to take out
    neglegible and unnecessary imaginary parts(if any
    are there)
25 disp('PRIMARY CURRENT:')
26 disp(I_1_max,'peak value in amperes')
27 disp(real(I_1_phi*180/%pi),'phase angle in degrees')
28 disp('PRIMARY VOLTAGE:')
29 disp(V_1_max,'peak value in amperes')
30 disp(real(V_1_phi*180/%pi),'phase angle in degrees')
31 disp('SECONDARY CURRENT')
32 disp(I_2_max,'peak value in amperes')
33 disp(real(I_2_phi*180/%pi),'phase angle in degrees')
34 disp('SECONDARY VOLTAGE')
35 disp(V_2_max,'peak value in amperes')
36 disp(real(V_2_phi*180/%pi),'phase angle in degrees')
37 disp(P_L,'power delivered to load in watts')

```

---

#### Scilab code Exa 15.12 example 12

```

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

```

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

---

### Scilab code Exa 15.13 example 13

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

```
29 I_1=0;
30 I_2=0;
31 V_1=V_s_max;
32 V_no_load=V_1/tr;
33 PR=((V_no_load-V_L_max)/V_L_max)*100;
34 disp(PR, 'Percent regulation')
```

---

# Chapter 16

## DC Machines

Scilab code Exa 16.1 example 1

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

```

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

```

---

### Scilab code Exa 16.2 example 2

```

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

```

```

23 i_A=F_load/(B*l);           //from steady state condition
24 e_A=V_T-R_A*i_A;           //induced voltage
25 u=e_A/(B*l);               //steady-state speed
26 P_m=F_load*u;              //mechanical power
27 P_t=V_T*i_A;               //power taken from battery
28 P_R=i_A^2*R_A;             //power dissipated in the
    resistance
29 eff=P_m*100/P_t;           //efficiency
30 disp('CASE b:')
31 disp(u,'steady-state speed in m/s')
32 disp(P_t,'power delivered by V_t in watts')
33 disp(P_m,'power delivered to mechanical load in
    watts')
34 disp(P_R,'power lost to heat in the resistance in
    watts')
35 disp(eff,'efficiency of converting electrical power
    to mechanical power')
36 //CASE c
37 //with the pulling force acting to the right,
    machine operates as a generator
38 F_pull=2;                  //pulling force
39 //steady-state condition F=B*l*i_A=F_pull
40 i_A=F_pull/(B*l);          //from steady-state condition
41 e_A=V_T+R_A*i_A;           //induced voltage
42 u=e_A/(B*l);               //steady-state speed
43 P_m=F_pull*u;              //mechanical power
44 P_t=V_T*i_A;               //power taken by battery
45 P_R=i_A^2*R_A;             //power dissipated in the
    resistance
46 eff=P_t*100/P_m;           //efficiency
47 disp('CASE c:')
48 disp(u,'steady-state speed in m/s')
49 disp(P_m,'power taken from mechanical source in
    watts')
50 disp(P_t,'power delivered to the battery in watts')
51 disp(P_R,'power lost to heat in the resistance')
52 disp(eff,'efficiency of converting mechanical power
    to electrical power')

```

---

Scilab code Exa 16.3 example 3

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

---

Scilab code Exa 16.4 example 4



```

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

```

---

### Scilab code Exa 16.5 example 5

```

1  clc
2  //ex16.5
3  n_m_1=1200;       //speed in rpm

```

```

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

```

---

#### Scilab code Exa 16.6 example 6

```

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

```

```

9  eff=0.85;          //efficiency not including power
    supplied to field circuit
10  I_F=V_F/(R_adj+R_F);      //field current
11  //E, voltage from magnetization curve for speed of n
    =1200
12  n=1200;
13  E=280;            //voltage of armature
14  //E_A is no-load voltage
15  E_A=E*n_A/n;      //E_A is proportional to speed
16  V_FL=E_A-R_A*I_fl;    //full-load voltage
17  VR=(E_A-V_FL)*100/V_FL; //voltage regulation
18  P_out=I_fl*V_FL;     //output power
19  P_dev=P_out+(I_fl^2)*R_A; //developed power
20  W_m=n_A*2*%pi/60;    //angular speed
21  P_in=P_out/eff;      //input power
22  P_loss=P_in-P_dev;   //all power losses combined
23  T_in=P_in/W_m;      //input torque
24  T_dev=P_dev/W_m;    //developed torque
25  printf(" All the values in the textbook are
    approximated hence the values in this code differ
    from those of Textbook")
26  disp(I_F, 'Field current in amperes')
27  disp(E_A, 'no-load voltage in volts')
28  disp(V_FL, 'full-load voltage in volts')
29  disp(VR, 'percentage voltage regulation')
30  disp(T_in, 'input torque in Nm')
31  disp(T_dev, 'developed torque')
32  disp(P_loss, 'all types of power losses combined in
    watts')

```

---

# Chapter 17

## AC Machines

Scilab code Exa 17.1 example 1

```
1  clc
2  //ex17.1
3  P_rot=900;          //rotational losses
4  V_L=440*complex(cos(0),sin(0));
5  R_s=1.2;
6  X_s=%i*2;
7  X_m=%i*50;
8  R_r_1=0.6;
9  R_r_2=19.4;
10 X_r=%i*0.8;
11 n_m=1746;          //machine operating speed in rpm
12 W_m=n_m*2*pi/60;  //speed in radians per second
13 n_s=1800;          //synchronous speed for a four-pole
    monitor
14 s=(n_s-n_m)/n_s;  //slip
15 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
16 [Z_s_max,phi]=polar(Z_s);
17 Z_s_phi=real(phi); //removing negligible
    imaginary part(if any is there)
18 PF=cos(Z_s_phi); //power factor
```

```

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

```

---

### Scilab code Exa 17.2 example 2

```

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

```

```

6 R_s=1.2;
7 X_s=%i*2;
8 X_m=%i*50;
9 R_r_1=0.6;
10 R_r_2=19.4;
11 X_r=%i*0.8;
12 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
13 Z_s=R_s+X_s+Z_eq;
14 I_s=V_s/Z_s; //starting phase current
15 [I_s_max, phi]=polar(I_s);
16 I_L=sqrt(3)*I_s_max; //starting line current
17 //I_L here is almost six times larger than in
    previous example. It is a typical characteristic
    of induction motors.
18 P_ag=3*real(Z_eq)*I_s_max^2; //power crossing
    air gap
19 W_s=2*pi*(60);
20 T_dev=P_ag/(W_s/2);
21 printf(" All the values in the textbook are
    approximated hence the values in this code differ
    from those of Textbook")
22 disp(I_L, 'Starting line current')
23 disp(T_dev, 'Torque in Nm')

```

---

### Scilab code Exa 17.3 example 3

```

1 clc
2 //ex17.3
3 V_L=220;
4 V_s=V_L/sqrt(3); //phase voltage
5 I_s=31.87;
6 P_s=400; //total stator copper losses
7 P_r=150; //total rotor copper losses

```

```

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

```

---

#### Scilab code Exa 17.4 example 4

```

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

```

```

16 //CASE b
17 V_a=V_L;           //phase voltage
18 I_a_max=P_dev_1/(3*V_a*PF);           //phase current
19 phi=acos(PF);
20 I_a=I_a_max*complex(cos(phi),sin(phi));
21 E_r=V_a-%i*X_s*I_a;           //voltage induced by rotor
22 E_r_max=sqrt((real(E_r)^2)+(imag(E_r)^2));
23 E_r_phi=atan(imag(E_r)/real(E_r));
24 TA=-E_r_phi;           //torque angle
25 disp('CASE b:')
26 disp('Phase current:')
27 disp(I_a_max,'peak value in amperes')
28 disp(phi*180/%pi,'phase angle in degrees')
29 disp('Voltage induced by rotor:')
30 disp(E_r_max,'peak value in volts')
31 disp(E_r_phi*180/%pi,'phase angle in degrees')
32 disp(TA*180/%pi,'torque angle in degrees')
33 //CASE c
34 //excitation constant means the values of I_f , B_r
   and E_r are constant
35 P_dev_2=100*746;
36 sin_t=P_dev_2*sin(TA)/P_dev_1;           //developed
   power is proportional to sin_t
37 t=asin(sin_t);
38 E_r=E_r_max*complex(cos(-t),sin(-t));           //E_r is
   constant in magnitude
39 I_a=(V_a-E_r)/(%i*X_s);           //new phase current
40 I_a_max=sqrt((real(I_a)^2)+(imag(I_a)^2));
41 I_a_phi=atan(imag(I_a)/real(I_a));
42 PF=cos(I_a_phi);
43 disp('CASE c:')
44 disp('Phase current:')
45 disp(I_a_max,'peak value in amperes')
46 disp(I_a_phi*180/%pi,'phase angle in degrees')
47 disp('Voltage induced by rotor:')
48 disp(E_r_max,'peak value in volts')
49 disp(-t*180/%pi,'phase angle in degrees')
50 disp(t*180/%pi,'torque angle in degrees')

```



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

---

### Scilab code Exa 17.5 example 6

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

---