

Scilab Textbook Companion for  
Electronic and Electrical Measuring  
Instruments & Machines  
by Bakshi And Bakshi<sup>1</sup>

Created by  
Vivek Goklani  
BE (EXTC)  
Others  
Mumbai University  
College Teacher  
Kanchan Chavan  
Cross-Checked by  
Chaitanya Potti

July 31, 2019

<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:** Electronic and Electrical Measuring Instruments & Machines

**Author:** Bakshi And Bakshi

**Publisher:** Technical Publications, Pune

**Edition:** 1

**Year:** 2009

**ISBN:** 978-81-8431-554-7

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 Electronic Voltmeters	5
2 Digital To Analog Converters	11
3 Analog To Digital Converters And Digital Voltmeters	14
4 Frequency Meters And Phase Meters	19
6 Oscilloscopes	21
7 Basic Measuring Instruments	24
8 Measurement Of Resistance Capacitance And Inductance	42
9 DC Motors	56
10 Three Phase Induction Motors	79

# List of Scilab Codes

Exa 1.1	calculate multiplier resistance . . . . .	5
Exa 1.2	calculate multiplier resistance . . . . .	5
Exa 1.3	calculate form factor and error . . . . .	6
Exa 1.4	calculate percentage error . . . . .	6
Exa 1.5	calculate series resistance . . . . .	7
Exa 1.6	calculate meter current . . . . .	7
Exa 1.7	calibrate meter . . . . .	7
Exa 1.8	design FET voltmeter . . . . .	8
Exa 1.9	calculate series resistance . . . . .	8
Exa 1.10	find calibration resistance . . . . .	9
Exa 1.11	design FET voltmeter . . . . .	9
Exa 2.1	design 4 bit DAC . . . . .	11
Exa 2.2	calculate resolution . . . . .	11
Exa 2.3	calculate final output voltage . . . . .	12
Exa 2.4	calculate full scale output . . . . .	12
Exa 2.5	find step size and analog output . . . . .	13
Exa 2.6	find output voltage . . . . .	13
Exa 3.1	find resolution and digital output . . . . .	14
Exa 3.2	calculate quantisation error . . . . .	14
Exa 3.3	calculate time period . . . . .	15
Exa 3.4	find digital output . . . . .	15
Exa 3.5	find conversion time . . . . .	16
Exa 3.6	find maximum input frequency . . . . .	16
Exa 3.7	find resolution . . . . .	16
Exa 3.8	find percentage error . . . . .	17
Exa 3.9	find sensivity of meter . . . . .	17
Exa 3.10	find resolution and display voltage . . . . .	18
Exa 4.1	plot graph between phase voltage and output	19

Exa 4.2	calculate output voltage . . . . .	20
Exa 6.1	calculate bandwidth of CRO . . . . .	21
Exa 6.2	find minimum rise time of pulse . . . . .	21
Exa 6.3	calculate amplitude and rms value . . . . .	22
Exa 6.4	calculate frequency and rms value . . . . .	22
Exa 6.5	find phase difference between two waves . . . . .	23
Exa 6.6	find frequency at vertical plate . . . . .	23
Exa 7.1	calculate deflection . . . . .	24
Exa 7.2	find deflection . . . . .	24
Exa 7.3	calculate value of shunt resistance . . . . .	25
Exa 7.4	calculate shunt current and meter resistance . . . . .	25
Exa 7.5	design multirange dc milliammeter . . . . .	26
Exa 7.6	design aryton shunt . . . . .	26
Exa 7.7	calculate multiplier resistance . . . . .	27
Exa 7.8	calculate shunt and multiplier resistance . . . . .	27
Exa 7.9	design D arsonval movement voltmeter . . . . .	28
Exa 7.10	sensitivity method design . . . . .	28
Exa 7.11	find multiplier resistance . . . . .	29
Exa 7.12	sensitivity of meter comparison . . . . .	29
Exa 7.13	accuracy of meter comparison . . . . .	30
Exa 7.14	error and accuracy measurement . . . . .	31
Exa 7.15	basic PMMC measurement . . . . .	32
Exa 7.16	find shunt current and resistance for fsd . . . . .	32
Exa 7.17	determine inductance of instrument . . . . .	33
Exa 7.18	find deflecting torque . . . . .	33
Exa 7.19	difference between dc and ac readings of voltmeter . . . . .	34
Exa 7.20	find revolutions and percentage error . . . . .	35
Exa 7.21	determine meter error at half load . . . . .	35
Exa 7.22	calculate power factor of load . . . . .	36
Exa 7.23	find speed of disc and error of meter . . . . .	36
Exa 7.24	find error at given power factor . . . . .	37
Exa 7.25	limits of error of wattmeter . . . . .	38
Exa 7.26	estimate line current . . . . .	39
Exa 7.27	estimate line voltage . . . . .	39
Exa 7.28	find percentage ratio error . . . . .	40
Exa 7.29	calculate actual primary current and ratio error . . . . .	40

Exa 8.1	calculate R1 and R2 of ohmmeter . . . . .	42
Exa 8.2	find unknown resistance . . . . .	42
Exa 8.3	find current through galvanometer . . . . .	43
Exa 8.4	find unknown resistance . . . . .	43
Exa 8.5	find constants of unknown impedance . . . . .	44
Exa 8.6	determine balance of bridge . . . . .	44
Exa 8.7	find equivalent series circuit . . . . .	45
Exa 8.8	find equivalent series circuit . . . . .	46
Exa 8.9	find components of branch BC . . . . .	46
Exa 8.10	find constants of unknown impedance . . . . .	46
Exa 8.11	calculate unknown capacitance and dissipation factor . . . . .	47
Exa 8.12	find deflection of galvanometer . . . . .	47
Exa 8.13	find deflection of galvanometer . . . . .	48
Exa 8.14	find current through galvanometer . . . . .	48
Exa 8.15	calculate smallest change in resistance . . . . .	49
Exa 8.16	calculate balance temperature and error . . . . .	49
Exa 8.17	find value of unknown resistance . . . . .	50
Exa 8.18	find unknown resistance and unbalance in bridge . . . . .	50
Exa 8.19	find series resistance . . . . .	51
Exa 8.20	find unknown resistance Rx . . . . .	51
Exa 8.21	find constants of arm CD . . . . .	52
Exa 8.22	find constants of Zx . . . . .	52
Exa 8.23	find unknown impedance . . . . .	53
Exa 8.24	find unknown impedance and dissipation factor . . . . .	53
Exa 8.25	determine unknown parameters of arm AB . . . . .	54
Exa 8.26	find resistance and inductance of coil . . . . .	54
Exa 8.27	find limiting values of unknown resistance . . . . .	55
Exa 9.1	calculate generated emf . . . . .	56
Exa 9.2	calculate speed and generated emf . . . . .	56
Exa 9.3	calculate induced emf . . . . .	57
Exa 9.4	calculate back emf and speed . . . . .	57
Exa 9.5	calculate gross torque . . . . .	58
Exa 9.6	calculate induced emf and lost torque . . . . .	58
Exa 9.7	calculate speed . . . . .	59
Exa 9.8	find armature current and back emf . . . . .	60
Exa 9.9	calculate speed on full load . . . . .	60

Exa 9.10	calculate speed on new load . . . . .	61
Exa 9.11	find new speed and armature current . . . . .	61
Exa 9.12	find external resistance . . . . .	62
Exa 9.13	calculate speed of motor . . . . .	63
Exa 9.14	find out speed of motor . . . . .	63
Exa 9.15	find speed and torque of motor . . . . .	64
Exa 9.16	find speed on full load . . . . .	65
Exa 9.17	determine armature current and speed of machine . . . . .	65
Exa 9.18	determine mechanical power on full load . . . . .	66
Exa 9.19	calculate full load speed . . . . .	67
Exa 9.20	determine full load speed and efficiency . . . . .	68
Exa 9.21	find speed for parallel field groups . . . . .	69
Exa 9.22	find new speed and armature current . . . . .	69
Exa 9.23	find external resistance . . . . .	70
Exa 9.24	estimate supply voltage . . . . .	70
Exa 9.25	find efficiency and power input . . . . .	71
Exa 9.26	calculate efficiency and armature current . . . . .	72
Exa 9.27	calculate back emf . . . . .	73
Exa 9.28	calculate torque and efficiency . . . . .	73
Exa 9.29	determine efficiency and speed of motor . . . . .	74
Exa 9.30	calculate efficiency of motor . . . . .	75
Exa 9.31	calculate speed of motor combination . . . . .	76
Exa 9.32	calculate efficiency and power output . . . . .	76
Exa 9.33	calculate speed on given load . . . . .	77
Exa 10.1	calculate full load slip . . . . .	79
Exa 10.2	calculate full load speed . . . . .	79
Exa 10.3	calculate rotor frequency . . . . .	80
Exa 10.4	find full load slip and speed . . . . .	80
Exa 10.5	calculate rotor frequency and induced emf . . . . .	81
Exa 10.6	find rotor current and rotor power factor . . . . .	81
Exa 10.7	calculate full load torque . . . . .	82
Exa 10.8	calculate starting torque and full load torque . . . . .	82
Exa 10.9	star connected induction motor . . . . .	83
Exa 10.10	calculate external resistance . . . . .	84
Exa 10.11	calculate rotor copper loss . . . . .	85
Exa 10.12	calculate full load efficiency . . . . .	85
Exa 10.13	calculate slip and rotor resistance per phase . . . . .	86



Exa 10.14	calculate gross mechanical power and efficiency	87
Exa 10.15	calculate shaft torque and full load efficiency	87
Exa 10.16	calculate tapping and supply start current .	89
Exa 10.17	determine ratios of torques . . . . .	89
Exa 10.18	calculate rotor current and external resistance	89
Exa 10.19	calculate starting torque and speed . . . . .	90
Exa 10.20	calculate efficiency on full load . . . . .	91
Exa 10.21	calculate new speed . . . . .	92
Exa 10.22	find rotor current and rotor emf per phase .	92
Exa 10.23	calculate starting torque and speed . . . . .	93
Exa 10.24	calculate full load torque and external resistance . . . . .	94
Exa 10.25	calculate slip and line current . . . . .	95
Exa 10.26	find power factor of rotor . . . . .	96
Exa 10.27	determine full load speed and speed at max torque . . . . .	96
Exa 10.28	calculate starting torque . . . . .	97
Exa 10.29	calculate speed torque and external resistance	98
Exa 10.30	calculate motor output and efficiency . . . . .	99
Exa 10.31	find ratio of torques . . . . .	99

# Chapter 1

## Electronic Voltmeters

Scilab code Exa 1.1 calculate multiplier resistance

```
1 //Chapter -1,Example1_1 ,pg 1_17
2 Erms=10
3 Ep=sqrt(2)*Erms
4 Eav=0.6*Ep
5 E=Eav/2
6 Edc=0.45*Erms
7 Idc=1*10^-3
8 Rm=200
9 Rs=(Edc/Idc)-Rm
10 printf("required multiplier resistance")
11 printf("Rs=%.2f ohm \n",Rs )
```

---

Scilab code Exa 1.2 calculate multiplier resistance

```
1 //Chapter -1,Example1_2 ,pg 1_18
2 Eav=9
3 Erms=10
4 Rm=500
```

```

5 Idc=2*10^-3
6 Edc=0.9*Erms
7 Rs=(Edc/Idc)-Rm
8 printf("required multiplier resistance")
9 printf("Rs=%0.2f ohm \n",Rs )

```

---

**Scilab code Exa 1.3** calculate form factor and error

```

1 //Chapter -1,Example1_3 ,pg 1_20
2 Kf=1//Erms=Em for 1 time period
3 Kf1=1.11//Kf(sine)/Kf(square)
4 pere=(Kf-Kf1)/Kf*100//percentage error
5 printf("percentage error ")
6 printf("pere=%0.2f ",pere)

```

---

**Scilab code Exa 1.4** calculate percentage error

```

1 //Chapter -1,Example1_4 ,pg 1_20
2 A=50
3 T=2
4 function E=f(t),E=(50*t)^2, endfunction //e=At(ramp
   function)
5 exact=-2.5432596188;
6 I=intg(0,T,f)
7 abs(exact-I)
8 Erms=sqrt((1/T)*I)
9 function e=f1(t),e=50*t, //e=At(ramp function)
10 endfunction
11 exact=-2.5432596188;
12 I1=intg(0,T,f)
13 Eav=(1/T)*I1
14 Kf=Erms/Eav
15 kf1=0.961//Kf(sine)/Kf(sawtooth)

```

```
16 pere=(1-kf1)/1*100//percentage error
17 printf("percentage error ")
18 printf("pere=%0.2 f ",pere)
```

---

#### Scilab code Exa 1.5 calculate series resistance

```
1 //Chapter-1,Example1_5 ,pg 1-27
2 Idc=25*10^-3
3 Erms=200
4 Rm=100
5 Rf=500
6 Rd=2*Rf
7 Rm1=Rm+Rd//total meter resistance
8 Rs=(0.9*Erms)/Idc-Rm1
9 printf("total meter resistance")
10 printf("Rs=%0.2 f ohm",Rs)
```

---

#### Scilab code Exa 1.6 calculate meter current

```
1 //Chapter-1,Example1_6 ,pg 1-38
2 V1=2
3 Rm=50
4 Rd=15*10^3
5 gm=0.006
6 rd=100*10^3
7 Im=(gm*rd*Rd/(rd+Rd)*V1)/((2*(rd*Rd/(rd+Rd))+Rm))
8 printf("meter current")
9 printf("Im=%0.4 f A",Im)
```

---

#### Scilab code Exa 1.7 calibrate meter

```

1 //Chapter -1,Example1_7 , pg 1_38
2 V1=1
3 Rm=50
4 Rd=15*10^3
5 gm=0.006
6 rd=100*10^3
7 Im=(gm*rd*Rd/(rd+Rd)*V1)/((2*(rd*Rd/(rd+Rd))+Rm))
8 printf("meter current")
9 printf("Im=%0.4 f A" ,Im)

```

---

#### Scilab code Exa 1.8 design FET voltmeter

```

1 //Chapter -1,Example1_8 , pg 1_39
2 V1=1
3 Vin=30
4 Rin=9*10^6
5 R4=Rin/100 //for Vin=100V
6 R3=(Rin-50*R4)/50 //for Vin=50V
7 R2=(Rin-30*R3-30*R4)/30 //for Vin=30V
8 R1=Rin-R2-R3-R4
9 printf("resistance values are\n")
10 printf("R1=%0.2 f ohm\n" ,R1)
11 printf("R2=%0.2 f ohm\n" ,R2)
12 printf("R3=%0.2 f ohm\n" ,R3)
13 printf("R4=%0.2 f ohm\n" ,R4)

```

---

#### Scilab code Exa 1.9 calculate series resistance

```

1 //Chapter -1,Example1_9 , pg 1_40
2 rd=10*10^3
3 gm=0.003
4 rdf=rd/(1+gm*rd) //actual rd
5 Rs=15*10^3

```

```

6 V1=1//input voltage
7 Vo=(gm*rd*Rs)*V1/(rdf+Rs)
8 Rth=(2*Rs*rdf/(Rs+rdf))
9 Rm=1800
10 Im=Vo/(Rth+Rm)
11 Img=0.1*10^-3//meter current given
12 Rf=(Vo/Img)-Rth-Rm//series resistance
13 printf("current Im=%0.5 f A\n",Im)
14 printf("seires resistance\n")
15 printf("Rf=%0.2 f ohm\n",Rf)

```

---

**Scilab code Exa 1.10** find calibration resistance

```

1 //Chapter -1,Example1_10 ,pg 1_41
2 rd=200*10^3
3 gm=0.004
4 Rs=40*10^3
5 Rm=1000
6 V1=1
7 rdf=rd/(1+gm*rd)//actual rd
8 Rth=(2*Rs*rdf/(Rs+rdf))
9 Vo=(gm*rd*Rs)*V1/(rdf+Rs)
10 Im=50*10^-6
11 Rcal=(Vo/Im)-Rth-Rm//caliberation resistance
12 printf("caliberation resistance\n")
13 printf("Rcal=%0.2 f ohm",Rcal)

```

---

**Scilab code Exa 1.11** design FET voltmeter

```

1 //Chapter -1,Example1_11 ,pg 1_42
2 Vin=3
3 V1=1
4 Rin=1*10^6//input resistance of FET

```

```
5 R4=Rin/100//for Vin=100V
6 R3=(Rin-30*R4)/30//for Vin=30V
7 R2=(Rin-3*R3-3*R4)/3//for Vin=3V
8 R1=Rin-R2-R3-R4
9 printf(" Resistances are\n")
10 printf("R1=%0.2 f ohm\n",R1)
11 printf("R2=%0.2 f ohm\n",R2)
12 printf("R3=%0.2 f ohm\n",R3)
13 printf("R4=%0.2 f ohm",R4)
```

---

## Chapter 2

# Digital To Analog Converters

Scilab code Exa 2.1 design 4 bit DAC

```
1
2 //Chapter -2,Example2_1 , pg 2_9
3 Vr=10
4 n=4
5 Res=0.5 // resolution
6 Rt=Vr/((2^n)*Res)
7 Rf=10*10^3
8 R=Rt*Rf
9 printf("input resistance\n")
10 printf("r=%0.2f ohm\n",R)
11 printf("feedback resistance\n")
12 printf("Rf=%0.f ohm",Rf)
```

---

Scilab code Exa 2.2 calculate resolution

```
1 //Chapter -2,Example2_2 , pg 2_11
2 n=8
3 Res1=2^n
```



```

4 Vofs=2.55//full scale output voltage
5 Res2=Vofs/(Res1-1)
6 printf("resolution through method-1\n")
7 printf("Res1=%0.2f \n",Res1)
8 printf("resolution through method-2\n")
9 printf("Res2=%0.2f \n",Res2)

```

---

**Scilab code Exa 2.3** calculate final output voltage

```

1 //Chapter-2,Example2_3 ,pg 2-12
2 n=4
3 Vofs=15
4 Res=Vofs/((2^n)-1)
5 D=bin2dec('0110')//decimal equivalent
6 Vo=Res*D
7 printf("output voltage\n")
8 printf("Vo=%0.2f V",Vo)

```

---

**Scilab code Exa 2.4** calculate full scale output

```

1 //Chapter-2,Example2_4 ,pg 2-12
2 Res=20*10^-3
3 n=8
4 Vofs=Res*((2^n)-1)
5 D=bin2dec('10000000')
6 Vo=Res*D
7 printf("output voltage\n")
8 printf("Vo=%0.2f V\n",Vo)
9 printf("full scale output voltage\n")
10 printf("Vofs=%0.2f V",Vofs)

```

---

**Scilab code Exa 2.5** find step size and analog output

```
1 //Chapter -2,Example2_5 , pg 2_12
2 n=4
3 Vofs=5
4 Res=Vofs/((2^n)-1)
5 D1=bin2dec('1000')
6 Vo1=Res*D1
7 D2=bin2dec('1111')
8 Vo2=Res*D2
9 printf("output voltage1\n")
10 printf("Vo1=%0.2f V\n",Vo1)
11 printf("output voltage2\n")
12 printf("Vo2=%0.2f V\n",Vo2)
```

---

**Scilab code Exa 2.6** find output voltage

```
1 //Chapter -2,Example2_6 , pg 2_13
2 n=12
3 Res=8*10^-3
4 Vofs=Res*((2^n)-1)
5 perR=Res/Vofs*100
6 Vo=Res*bin2dec('010101101101')
7 printf("percentage resolution\n")
8 printf("perR=%0.2f \n",perR)
9 printf("output voltage\n")
10 printf("Vo=%0.2f V",Vo)
```

---

## Chapter 3

# Analog To Digital Converters And Digital Voltmeters

Scilab code Exa 3.1 find resolution and digital output

```
1
2 //Chapter -3,Example3_1 , pg 3_5
3 n=8
4 Res1=2^n
5 Vifs=5.1
6 Res2=Vifs/((2^n)-1)
7 Res=Res2*1000//in mv/LSB
8 Vi=1.28
9 D=Vi/Res2
10 str=dec2bin(64)
11 printf("Resolution\n")
12 printf("Res2=%0. f mv/LSB\n",Res)
13 printf("digital output voltage \n")
14 printf("D=%0. f LSBs\n",D)
```

---

Scilab code Exa 3.2 calculate quantisation error

```

1 //Chapter -3,Example3_2 , pg 3_6
2 Vifs=4.095
3 n=12
4 Qe=Vifs/(((2^n)-1)*2)
5 printf("quantisation error\n")
6 printf("Qe=%0.5 f V" ,Qe)

```

---

**Scilab code Exa 3.3** calculate time period

```

1 //Chapter -3,Example3_3 , pg 3_10
2 V1=100*10^-3
3 Vr=100*10^-3
4 t1=83.33
5 t2=(V1/Vr)*t1
6 printf("t2=%0.5 f ms\n" ,t2)
7 Vi=200*10^-3 //input voltage
8 t2=(Vi/Vr)*t1
9 printf("t2=%0.5 f ms" ,t2)

```

---

**Scilab code Exa 3.4** find digital output

```

1 //Chapter -3,Example3_4 , pg 3_10
2 fclk=12*10^3 //clock frequency
3 t1=83.33*10^-3
4 V1=100*10^-3
5 Vr=100*10^-3
6 D=fclk*t1*(V1/Vr)
7 printf("digital output\n")
8 printf("D=%0.f counts" ,D)

```

---

**Scilab code Exa 3.5** find conversion time

```
1 //Chapter-3,Example3_5 ,pg 3_13
2 F=1*10^6
3 T=1/F
4 n=8
5 Tc=T*(n+1)
6 printf("conversion time\n")
7 printf("Tc=%0.7f sec",Tc)
```

---

**Scilab code Exa 3.6** find maximum input frequency

```
1 //Chapter-3,Example3_6 ,pg 3_15
2 Tc=9*10^-6
3 n=8
4 fmax=1/(2*%pi*Tc*(2^n))
5 printf("maximum input frequency\n")
6 printf("fmax=%0.2f Hz",fmax)
```

---

**Scilab code Exa 3.7** find resolution

```
1 //Chapter-3,Example3_7 ,pg 3_37
2 n=3//3 full digits
3 R=1/(10^n)
4 //for 1V range
5 Res1=1*R
6 //for 50V range
7 Res2=50*R
8 printf("least difference in readings for 50V range\n")
9 printf("Res=%0.2f V",Res2)
```

---

### Scilab code Exa 3.8 find percentage error

```
1
2 //Chapter -3,Example3-8 ,pg 3-38
3 n=3
4 R=1/(10^n)
5 //for 10V range
6 R=R*10
7 err1=R//1-digit error
8 //reading is 5V
9 err=(0.5/100)*5//error due to reading
10 errt=err1+err//total error
11 printf("error when reading is 5V\n")
12 printf("errt=%0.4f V\n",errt)
13 //reading is 0.1V
14 err=(0.5/100)*0.1//error due to reading
15 errt=err+err1//total error
16 errp=(errt/0.1)*100
17 printf("percent error when reading is 0.1V\n")
18 printf("errp=%0.1f ",errp)
```

---

### Scilab code Exa 3.9 find sensitivity of meter

```
1 //Chapter -3,Example3-9 ,pg 3-38
2 n=4
3 fsmin=10*10^-3//full scale value on min. range
4 R=1/(10^n)
5 S=fsmin*R
6 printf("sensitivity of meter\n ")
7 printf("s=%0.7f",S)
```

---

Scilab code Exa 3.10 find resolution and display voltage

```
1 //Chapter -3,Example3-10 ,pg 3-39
2 n=4
3 R=1/(10^n)
4 //for 10V range
5 R=10*R
6 printf("12.98 would be displayed as 12.980 for 10V
   range\n")
7 //for 1V range
8 R=1*R
9 printf("0.6973 would be displayed as 0.6973 for 1V
   range\n")
10 //for 10V range
11 printf("0.6973 would be displayed as 0.697 for 10V
   range\n")
```

---

# Chapter 4

## Frequency Meters And Phase Meters

Scilab code Exa 4.1 plot graph between phase voltage and output

```
1
2 //Chapter-4,Example4_1 , pg 4-22
3 E1mag=[0 3 5 7 9 12 15 18 21]
4 E1rms=E1mag/sqrt(2)
5 Erms=5//given
6 Einrms=(((E1rms)^2)+((Erms)^2))^(1/2)
7 Eab=(2*sqrt(2).*Einrms)/%pi
8 xlabel('E1(Volts)', 'fontsize',5)
9 ylabel('Eab(Volts)', 'fontsize',5)
10 title('Phase Meter', 'fontsize',5)
11 printf("E1 mag      E1 rms      Ein Rms      Eab output")
12 k=[0      0      5      4.501;
13    3      2.121    5.431    4.889;
14    5      3.53     6.123    5.513;
15    7      4.949    7.035    6.334;
16    9      6.363    8.093    7.286;
17   12     8.485    9.848    8.867;
18   15    10.606   11.726   10.557;
19   18    12.727   13.674   12.311;
```



```
20      21      14.849      15.668      14.106  ]
21  disp(k)
22  plot(E1mag, Eab)
```

---

Scilab code Exa 4.2 calculate output voltage

```
1  //Chapter -4, Example4_2 , pg 4-24
2  E1rms=10
3  E2rms=15
4  E1m=E1rms*sqrt(2)
5  E2m=E2rms*sqrt(2)
6  //voltage across AB is proportional to E1+E2 in
   positive half cycle
7  Ep=(1/(2*pi))*(2*E1m+E2m)//output in positive half
   cycle
8  //voltage across AB is proportional to E1-E2 in
   negative half cycle
9  En=(1/(2*pi))*(2*E1m-E2m)//output in negative half
   cycle
10 Eab=Ep-En
11 printf("output voltage\n")
12 printf("Eab=%.2f V", Eab)
```

---

# Chapter 6

## Oscilloscopes

Scilab code Exa 6.1 calculate bandwidth of CRO

```
1 //Chapter -6,Example6_1 , pg 6-26
2 Trs=17*10^-6
3 Trd=21*10^-6
4 Tro=sqrt((Trd^2)-(Trs^2))
5 BW=0.35/Tro
6 printf("bandwidth of CRO\n")
7 printf("BW=%0.2 f Hz" ,BW)
```

---

Scilab code Exa 6.2 find minimum rise time of plulse

```
1 //Chapter -6,Example6_2 , pg 6-53
2 SR=200*10^6//sampling rate
3 trmin=1/SR
4 printf("minimum rise time of pulse\n")
5 printf("trmin=%0.10 f s" ,trmin)
```

---

Scilab code Exa 6.3 calculate amplitude and rms value

```
1 //Chapter-6,Example6_3 ,pg 6-63
2 //from plot 1 subdivision=0.2 units
3 pp=2+3*0.2//positive peak
4 np=2+3*0.2//negative peak
5 Nd=pp+np//no. of divisions
6 Vd=2*10^-3//volts per division
7 Vpp=Nd*Vd
8 Vm=Vpp/2
9 Vrms=Vm/sqrt(2)
10 printf("peak value of voltage\n")
11 printf("Vm=%0.4f V\n",Vm)
12 printf("RMS value of voltage\n")
13 printf("Vrms=%0.4f V\n",Vrms)
```

---

Scilab code Exa 6.4 calculate frequency and rms value

```
1 //Chapter-6,Example6_4 ,pg 6-64
2 Vd=2
3 Tb=2*10^-3//time base
4 Vd=2
5 Nd=3
6 Vpp=Nd*Vd
7 Vm=Vpp/2
8 Vrms=Vm/sqrt(2)
9 Hd=2//horizontal occupancy
10 T=Tb*Hd
11 f=1/T
12 printf("RMS value of voltage\n")
13 printf("Vrms=%0.2f V\n",Vrms)
14 printf("frequency of voltage across resistor\n")
15 printf("f=%0.2f Hz",f)
```

---

**Scilab code Exa 6.5** find phase difference between two waves

```
1 //Chapter -6,Example6_5 , pg 6-67
2 y1=8
3 y2=10
4 phi=asin(y1/y2)//phase difference
5 phi=phi*(180/%pi)
6 printf(" phase difference\n")
7 printf("phi=%0.2 f deg",phi)
```

---

**Scilab code Exa 6.6** find frequency at vertical plate

```
1 //Chapter -6,Example6_6 , pg 6-69
2 Nv=2
3 Nh=5
4 fh=1*10^3
5 fv=(5/2)*fh// (fv/fh)=(Nh/Nv)=(5/2)
6 printf(" vertical signal frequency\n")
7 printf("fv=%0. f Hz",fv)
```

---

# Chapter 7

## Basic Measuring Instruments

Scilab code Exa 7.1 calculate deflection

```
1 //Chapter -7,Example7_1 , pg 7-13
2 N=100
3 B=0.15
4 A=10*8*10^-6
5 I=5*10^-3
6 Td=N*B*A*I//deflecting torque
7 K=0.2*10^-6//spring const.
8 theta=Td/K//deflecting angle
9 printf("deflection theta=%0.2f deg",theta)
```

---

Scilab code Exa 7.2 find deflection

```
1 //Chapter -7,Example7_2 , pg 7-21
2 x=poly(0,"x")
3 L=(12+6*x-(x^2))//x is deflection in rad from zero
4 dl=derivat(L)
5 K=12
6 I=8
```

```

7 x=6/(((2*K)/(I^2))+2)//x=((I^2)d1)/(2*k)
8 z=x*(180/%pi)
9 y=horner(L,x)
10 printf("deflection for given current\n")
11 printf("x=%0.2f deg\n",z)
12 printf("inductance for given deflection\n")
13 printf("L=%0.2f uH",y)

```

---

**Scilab code Exa 7.3** calculate value of shunt resistance

```

1 //Chapter -7,Example7_3 ,pg 7-23
2 Rm=100
3 Im=2*10^-3
4 I=150*10^-3
5 Rsh=(Im*Rm)/(I-Im)
6 printf("value of shunt resistance\n")
7 printf("Rsh=%0.2f ohm",Rsh)

```

---

**Scilab code Exa 7.4** calculate shunt current and meter resistance

```

1 //Chapter -7,Example7_4 ,pg 7-23
2 Vsh1=400*10^-3
3 Rsh=0.01
4 Ish=Vsh1/Rsh
5 printf("current through shunt\n")
6 printf("Ish=%0.2f A\n",Ish)
7 Ish=50
8 Vsh=Ish*Rsh
9 printf("voltage through shunt\n")
10 printf("Vsh=%0.2f V\n",Vsh)
11 Rm=750//coil resistance
12 Im=Vsh1/Rm
13 Rm1=Vsh/Im//meter resistance

```

```
14 printf("meter resistance\n")
15 printf("Rm1=%0.2 f ohm\n", Rm1)
```

---

#### Scilab code Exa 7.5 design multirange dc milliammeter

```
1 //Chapter -7, Example7_5 , pg 7-25
2 I1=10*10^-3
3 Im=2*10^-3
4 Rm=75
5 R1=(Im*Rm)/(I1-Im)
6 I2=50*10^-3
7 R2=(Im*Rm)/(I2-Im)
8 I3=100*10^-3
9 R3=(Im*Rm)/(I3-Im)
10 printf("designed multi-range ammeter\n")
11 printf("full scale deflection Im=%0.5 f A\n", Im)
12 printf("meter resistance Rm=%0.2 f ohm\n", Rm)
13 printf("R1=%0.2 f ohm\n", R1)
14 printf("R2=%0.2 f ohm\n", R2)
15 printf("R3=%0.2 f ohm\n", R3)
```

---

#### Scilab code Exa 7.6 design alyton shunt

```
1 //Chapter -7, Example7_6 , pg 7-27
2 I1=10
3 Im=1*10^-3
4 Rm=50
5 //in position -1 R1 is in shunt with R2+R3+Rm
6 //R1=10^-4(R2+R3+50).....(1)
7 //in position -2 (R1+R2) is in shunt with R3+Rm
8 //R1+R2=2*10^-4(R3+50).....(2)
9 //in position -3 R1+R2+R3 is in shunt with Rm
10 //R1+R2+R3=0.05.....(3)
```

```

11 //from .....(3)
12 //R1+R2=0.05-R3
13 //substituting in .....(2)
14 R3=0.04/1.0002
15 //R2=0.01-R1 .....(4)
16 //substituting in (1)
17 R1=5.00139*10^-3/1.0001
18 R2=0.01-R1//from .....(4)
19 printf("various sections of aryton shunt are\n")
20 printf("full scale deflection Im=%0.4f A\n",Im)
21 printf("meter resistance Rm=%0.2f ohm\n",Rm)
22 printf("R1=%0.4f ohm\n",R1)
23 printf("R2=%0.4f ohm\n",R2)
24 printf("R3=%0.4f ohm\n",R3)

```

---

**Scilab code Exa 7.7** calculate multiplier resistance

```

1 //Chapter -7,Example7-7 , pg 7-30
2 Rm=500
3 Im=40*10^-6
4 V=10
5 Rs=(V/Im)-Rm
6 printf("multiplier resistance\n")
7 printf("Rs=%0.2f ohm",Rs)

```

---

**Scilab code Exa 7.8** calculate shunt and multiplier resistance

```

1 //Chapter -7,Example7-8 , pg 7-30
2 Im=20*10^-3
3 Vm=200*10^-3
4 Rm=(Vm/Im)
5 I=200
6 Rsh=(Im*Rm)/(I-Im)

```



```

7 printf("required shunt resistance\n")
8 printf("Rsh=%0.4 f ohm\n",Rsh)
9 V=500
10 Rs=(V/Im)-Rm
11 printf("required multiplier resistance\n")
12 printf("Rs=%0.2 f ohm",Rs)

```

---

#### Scilab code Exa 7.9 design D arsonoal movement voltmeter

```

1 //Chapter -7,Example7_9 ,pg 7-33
2 Rm=50
3 Im=2*10^-3
4 //for position V4 multiplier is R4
5 V4=10
6 R4=(V4/Im)-Rm//Rs=(V/Im)-RmV3 m
7 //for position V3 multiplier is R3+R4
8 V3=50
9 R3=(V3/Im)-Rm-R4
10 //for position V2 multiplier is R2+R3+R4
11 V2=100
12 R2=(V2/Im)-Rm-R3-R4
13 //for position V1 multiplier is R1+R2+R3+R4
14 V1=500
15 R1=(V1/Im)-Rm-R3-R4-R2
16 printf("series string of multipliers\n")
17 printf("R1=%0.2 f ohm\n",R1)
18 printf("R2=%0.2 f ohm\n",R2)
19 printf("R3=%0.2 f ohm\n",R3)
20 printf("R4=%0.2 f ohm\n",R4)

```

---

#### Scilab code Exa 7.10 sensitivity method design

```

1 //Chapter -7,Example7_10 ,pg 7-35

```

```

2 Rm=50
3 Im=2*10^-3
4 V1=500
5 V2=100
6 V3=50
7 V4=10
8 S=1/Im//sensitivity
9 R4=S*V4-Rm
10 R3=S*V3-(R4+Rm)
11 R2=S*V2-(R4+Rm+R3)
12 R1=S*V1-(R4+Rm+R3+R2)
13 printf("series string of multipliers\n")
14 printf("R1=%0.2 f ohm\n",R1)
15 printf("R2=%0.2 f ohm\n",R2)
16 printf("R3=%0.2 f ohm\n",R3)
17 printf("R4=%0.2 f ohm\n",R4)

```

---

**Scilab code Exa 7.11** find multiplier resistance

```

1 //Chapter-7,Example7_11 ,pg 7-36
2 Im=50*10^-6
3 S=1/Im
4 Rm=200
5 V=500//V is voltage range
6 Rs=S*V-Rm
7 printf("multiplier resistance\n")
8 printf("Rs=%0.2 f ohm",Rs)

```

---

**Scilab code Exa 7.12** sensitivity of meter comparison

```

1
2 //Chapter-7,Example7_12 ,pg 7-36
3 //for meter A

```

```

4 Rs=25*10^3
5 Rm=1*10^3
6 V=100
7 S=(Rs+Rm)/V
8 printf("sensivity of meter A\n")
9 printf("S=%0.2f ohm/volt\n",S)
10 //for meter B
11 Rs=150*10^3
12 Rm=1*10^3
13 V=1000
14 S=(Rs+Rm)/V
15 printf("sensivity of meter B\n")
16 printf("S=%0.2f ohm/volt",S)

```

---

#### Scilab code Exa 7.13 accuracy of meter comparison

```

1 //Chapter -7,Example7_13 ,pg 7-37
2 R1=20*10^3
3 R2=25*10^3
4 V=250 //voltage supply
5 VR2=R2*V/(R1+R2) //voltage across R2
6 //case -1
7 S=500
8 Vr=150 //voltage range of resistor
9 Rv=S*Vr
10 Req=R2*Rv/(R2+Rv)
11 VReq=Req*V/(Req+R1) //voltage across Req
12 printf("first voltmeter reading\n")
13 printf("VReq=%0.2f V\n",VReq)
14 //case -2
15 S=10*10^3
16 Rv=S*Vr
17 Req=R2*Rv/(R2+Rv)
18 VReq=Req*V/(Req+R1)
19 printf("second voltmeter reading\n")

```

```
20 printf("VReq=%0.2 f V",VReq)
```

---

### Scilab code Exa 7.14 error and accuracy measurement

```
1 //Chapter -7,Example7_14 ,pg 7-38
2 Rb=1*10^3
3 Ra=5*10^3
4 V=25
5 VRb=Rb*V/(Ra+Rb)//voltage across Rb
6 Vr=5
7 //case -1
8 S=1*10^3
9 Rv=S*Vr
10 Req=Rb*Rv/(Rb+Rv)
11 VReq=Req*V/(Req+Ra)
12 err=(VRb-VReq)*100/VRb
13 acc=100-err
14 printf("voltmeter reading case -1\n")
15 printf("VReq=%0.2 f V\n",VReq)
16 printf("percentage error\n")
17 printf("err=%0.2 f \n",err)
18 printf("percentage accuracy\n")
19 printf("acc=%0.2 f\n",acc)
20 //case -2
21 S=20*10^3
22 Rv=S*Vr
23 Req=Rb*Rv/(Rb+Rv)
24 VReq=Req*V/(Req+Ra)
25 err=(VRb-VReq)*100/VRb
26 acc=100-err
27 printf("voltmeter reading case -2\n")
28 printf("VReq=%0.2 f V\n",VReq)
29 printf("percentage error\n")
30 printf("err=%0.2 f \n",err)
31 printf("percentage accuracy\n")
```

```
32 printf("acc=%0.2f\n",acc)
```

---

### Scilab code Exa 7.15 basic PMMC measurement

```
1 //Chapter-7,Example7_15 ,pg 7-41
2 Rm=50
3 Im=20*10^-3
4 I=10
5 Rsh=(Im*Rm)/(I-Im)
6 printf("shunt resistance for I=10A\n")
7 printf("Rsh=%0.2f ohm\n",Rsh)
8 I=20
9 Rsh=(Im*Rm)/(I-Im)
10 printf("shunt resistance for I=20A\n")
11 printf("Rsh=%0.2f ohm\n",Rsh)
12 V=150
13 Rs=(V/Im)-Rm
14 printf("series resistance for V=150V\n")
15 printf("Rs=%0.2f ohm\n",Rs)
16 V=300
17 Rs=(V/Im)-Rm
18 printf("series resistance for V=300V\n")
19 printf("Rs=%0.2f ohm",Rs)
```

---

### Scilab code Exa 7.16 find shunt current and resistance for fsd

```
1 //Chapter-7,Example7_16 ,pg 7-42
2 Rsh=0.02
3 R=1000
4 Vm=500*10^-3
5 Im=Vm/R
6 Ish=Vm/Rsh
7 printf("shunt current\n")
```

```

8 printf("Ish=%0.2f A\n",Ish)
9 Ish1=10
10 V=Ish1*Rsh
11 R=V/Im
12 printf("resistance for Ish=10A\n")
13 printf("R=%0.2f ohm\n",R)
14 Ish2=75
15 V=Ish2*Rsh
16 R=V/Im
17 printf("resistance for Ish=75A\n")
18 printf("R=%0.2f ohm\n",R)

```

---

Scilab code Exa 7.17 determine inductance of instrument

```

1 //Chapter -7,Example7_17 ,pg 7-50
2 K=5.73*10^-6
3 I=20
4 theta=110*(%pi/180)//full scale deflection
5 dtheta=theta//change in theta
6 L=4*10^-6
7 dm=(theta*K/(I^2))*dtheta//change in inductance
8 Lf=L+dm
9 printf("final inductance\n")
10 printf("Lf=%0.8f H",Lf)

```

---

Scilab code Exa 7.18 find deflecting torque

```

1 //Chapter -7,Example7_18 ,pg 7-50
2 x=30//deflection
3 dM=5*sin((x+45)*(%pi/180))*10^-3//diffrentiate M w.r
  .t x
4 I=10*10^-3
5 Td=(I^2)*dM//deflecting torque

```

```

6 printf("deflecting torque\n")
7 printf("Td=%.8 f Nm" ,Td)

```

---

Scilab code Exa 7.19 difference between dc and ac readings of voltmeter

```

1 //Chapter -7,Example7_19 ,pg 7-51
2 I=100*10^-3
3 Td=0.8*10^-4
4 dtheta=90*%pi/180//in radians
5 theta=90//deflection
6 dM=Td*dtheta/(I^2)
7 Mo=0.5//original M
8 M=Mo+dM//total M
9 //case-1
10 Vdc=100
11 R=Vdc/I
12 w=2*%pi*50
13 Z=R+(%i*w*M)
14 Z=abs(Z)
15 Vac=R*Vdc/Z
16 dif=Vdc-Vac//difference between readings
17 //case-2
18 Vdc1=50
19 I1=Vdc1/R
20 theta1=theta*((I1/I)^2)//theta=kI^2
21 theta1=theta1*%pi/180//in radians
22 dM1=Td*theta1/(I^2)
23 M1=dM1+Mo
24 Z1=R+(%i*w*M1)
25 Z1=abs(Z1)
26 Vac1=R*Vdc1/Z1
27 dif1=Vdc1-Vac1
28 printf("difference in readings Vdc=100V\n")
29 printf("dif=%.2 f V\n",dif)
30 printf("difference in readings Vdc=50V\n")

```

```
31 printf(" dif1=%0.2 f V\n",dif1)
```

---

**Scilab code Exa 7.20** find revolutions and percentage error

```
1 //Chapter -7,Example7_20 ,pg 7-65
2 I=20
3 V=230
4 Pf=0.8//power factor
5 t=3600
6 K=100
7 Et=V*I*Pf*t
8 Et=Et/(3600*10^3)//in kWh
9 N=360
10 Er=3.6//in kWh
11 err=(Er-Et)/Et
12 err=err*100
13 printf("percentage error\n")
14 printf("err=%0.2 f\n",err)
15 printf("negative sign shows that meter is slow and
    Er<Et")
```

---

**Scilab code Exa 7.21** determine meter error at half load

```
1 //Chapter -7,Example7_21 ,pg 7-65
2 K=1800
3 V=230
4 I=10
5 Pf=1//half load
6 Ih1=I/2//half load current
7 t=138
8 Et=V*Ih1*Pf*t
9 Et=Et/(3600*10^3)
10 N=80//no. of revolutions
```



```

11 Er=N/K//in kWh
12 err=(Er-Et)/Et
13 err=err*100
14 printf("percentage error\n")
15 printf("err=%0.2f\n",err)
16 printf("positive sign shows that meter is fast and
    Er>Et")

```

---

**Scilab code Exa 7.22** calculate power factor of load

```

1 //Chapter -7,Example7_22 ,pg 7-66
2 V=230
3 I=4
4 t=6
5 Pf=1
6 N=2208
7 Et=V*I*Pf*t
8 K=N/Et
9 printf("meter constant\n")
10 printf("K=%0.2f rev/Wh\n",K)
11 V=230
12 I=5
13 t=4
14 N=1472
15 Et=V*I*Pf*t
16 Er=N/K
17 Pf=(Er/Et)
18 printf("power factor\n")
19 printf("Pf=%0.2f lagging",Pf)

```

---

**Scilab code Exa 7.23** find speed of disc and error of meter

```

1 //Chapter -7,Example7_23 ,pg 7-66

```

```

2 I=5
3 V=220
4 Pf=1
5 K=3275
6 t=1/60//in hr
7 E=V*I*Pf*t
8 E=E/10^3//in kWh
9 Rev=E*K//no. of revolutions
10 printf("speed of disc\n")
11 printf("s=%0.2f r.p.m\n",Rev)
12 //at half load
13 I=I/2
14 t=59.5
15 Et=V*I*Pf*t
16 Et=Et/(3600*10^3)//in kWh
17 N=30
18 Er=N/K
19 err=(Er-Et)/Et
20 err=err*100
21 printf("percentage error\n")
22 printf("err=%0.2f\n",err)
23 printf("Er>Et meter is fast")

```

---

Scilab code Exa 7.24 find error at given power factor

```

1 //Chapter -7,Example7_24 ,pg 7-67
2 V=240
3 I=10
4 Pf=0.8
5 t=1/60
6 K=600
7 E=V*I*Pf*t
8 E=E/10^3//in kWh
9 Rev=E*K//no. of revolutions
10 printf("speed of disc\n")

```

```

11 printf("s=%0.2f r.p.m\n",Rev)
12 del=90//for correct lag adjustment
13 del1=86*%pi/180//given in radian
14 phi=0//case-1 unity power factor
15 err=(sin(del1-phi)-cos(phi))/cos(phi)
16 err=err*100
17 printf("percentage error in case-1\n")
18 printf("err=%0.2f \n",err)
19 Pf=0.5//case-2
20 phi=60*%pi/180//in radians
21 err=(sin(del1-phi)-cos(phi))/cos(phi)
22 err=err*100
23 printf("percentage error in case-2\n")
24 printf("err=%0.2f \n",err)

```

---

#### Scilab code Exa 7.25 limits of error of wattmeter

```

1 //Chapter-7,Example7-25 ,pg 7-67
2 V=240
3 I=5
4 K=1200
5 N=40
6 Er=N/K
7 W=V*I
8 t=99.8
9 Td=500//total divisions
10 div=K/Td//1 division
11 We=0.1*div//wattmeter error
12 Ce=0.05*K/100//construction wattmeter error
13 Te=We+Ce//total error
14 Wru=K+Te
15 Wrl=K-Te//wattmeter reading limits
16 He=0.05//human error
17 Se=0.01//stopwatch error
18 Tte=He+Se//total timing error

```

```

19 Sru=t+Tte//stopwatch reading limits
20 Srl=t-Tte
21 Eu=Wru*Sru*1/(3600*10^3)//energy obtained limits
22 El=Wrl*Srl*1/(3600*10^3)
23 errl=(Er-El)/El
24 errl=errl*100
25 erru=(Er-Eu)/Eu//error limits
26 erru=erru*100
27 printf("percentage error upperlimit\n")
28 printf("erru=%0.3f \n",erru)
29 printf("percentage error lowerlimit\n")
30 printf("errl=%0.3f \n",errl)

```

---

#### Scilab code Exa 7.26 estimate line current

```

1 //Chapter -7,Example7_26 ,pg 7-79
2 I1=250
3 I2=5
4 I=I1/I2
5 //as ammeter is in secondary I2=2.7
6 I1=I*2.7//line current
7 printf("line current\n")
8 printf("I1=%0.2f A",I1)

```

---

#### Scilab code Exa 7.27 estimate line voltage

```

1 //Chapter -7,Example7_27 ,pg 7-82
2 V1=11000
3 V2=110
4 V=V1/V2
5 V2=87.5
6 V1=87.5*V//line voltage
7 printf("line voltage\n")

```

```
8 printf("V1=%.2f V",V1)
```

---

**Scilab code Exa 7.28** find percentage ratio error

```
1 //Chapter-7,Example7_28 ,pg 7-88
2 Im=120
3 Ic=38
4 Kn=1000/5
5 //at full load
6 Is=5
7 Ns=1000
8 Np=5
9 n=Ns/Np//turns ratio
10 R=n+(Ic/Is)
11 err=(Kn-R)/R//ratio error
12 err=err*100
13 printf("percentage ratio error\n")
14 printf("err=%.2f ",err)
```

---

**Scilab code Exa 7.29** calculate actual primary current and ratio error

```
1 //Chapter-7,Example7_29 ,pg 7-88
2 Im=90
3 Ic=40
4 delta=28*(%pi/180)//in radians
5 Is=5
6 Ns=400
7 Np=1
8 n=Ns/Np
9 Kn=n
10 R=n+((Im*sin(delta)+Ic*cos(delta))/Is)
11 Ip=R*Is//actual primary current
12 err=(Kn-R)/R
```

```
13 err=err*100
14 printf("percentage ratio error\n")
15 printf("err=%0.2f ",err)
```

---

## Chapter 8

# Measurement Of Resistance Capacitance And Inductance

Scilab code Exa 8.1 calculate R1 and R2 of ohmmeter

```
1 //Chapter -8,Example8_1 ,pg 8_6
2 Rh=1000
3 Rm=50
4 V=3
5 Ifsd=1*10^-3
6 R1=Rh-(Ifsd*Rm*Rh)/V
7 R2=(Ifsd*Rm*Rh)/(V-Ifsd*Rh)
8 printf("R1=%0.2 f ohm\n",R1)
9 printf("R2=%0.2 f ohm\n",R2)
10 //due to 5% voltage drop
11 V=V-0.05*V
12 R2=(Ifsd*Rm*Rh)/(V-Ifsd*Rh)
13 printf("change in value R2 \n")
14 printf("R2=%0.2 f ohm",R2)
```

---

Scilab code Exa 8.2 find unknown resistance

```

1 //Chapter –8,Example8_2 , pg 8_18
2 R1=10*10^3
3 R2=2*10^3
4 R3=5*10^3
5 //R4=Rx
6 R4=(R1*R3)/R2
7 printf("unknown resistance\n")
8 printf("R4=%.2 f ohm" ,R4)

```

---

### Scilab code Exa 8.3 find current through galvanometer

```

1 //Chapter –8,Example8_3 , pg 8_18
2 R1=7*10^3
3 R2=2*10^3
4 R3=4*10^3
5 R4=20*10^3
6 E=8
7 Rg=300
8 Vth=(E*R4/(R3+R4))-(E*R1/(R1+R2))//voltage divider
  rule
9 Req=(R1*R2/(R1+R2))+(R3*R4/(R3+R4))
10 Ig=Vth/(Req+Rg)
11 printf("current through galvanometer\n")
12 printf("Ig=%.7 f A" ,Ig)

```

---

### Scilab code Exa 8.4 find unknown resisance

```

1 //Chapter –8,Example8_4 , pg 8_25
2 R3=100.03*10^-6
3 R2=100.24
4 R1=200
5 b=100.31
6 a=200

```



```

7 Ry=700*10^-6
8 Rx=R1*R3/R2
9 Rx=Rx+(b*Ry/(Ry+a+b))*((R1/R2)-(a/b))
10 printf("unknown resistance\n")
11 printf("Rx=%0.7f ohm",Rx)

```

---

**Scilab code Exa 8.5** find constants of unknown impedance

```

1 //Chapter-8,Example8_5,pg 8_35
2 Z2=250
3 Z3=200
4 Z1=50
5 Z4=Z2*Z3/Z1//magnitude condition
6 theta1=80
7 theta2=0
8 theta3=30
9 theta4=theta2+theta3-theta1//angle condition
10 theta4=theta4*pi/180//in radians
11 Rx=Z4*cos(theta4)//real part
12 Ry=Z4*sin(theta4)//imag. part
13 Z4=Rx+i*Ry
14 printf("unknown impedance\n")
15 disp(Z4)

```

---

**Scilab code Exa 8.6** determine balance of bridge

```

1 //Chapter-8,Example8_6,pg 8_35
2 Z1=sqrt(((50*cos(40*pi/180))^2)+(50*sin(40*pi/180))^2)//angle in radians
3 Z2=sqrt(((100*cos(-90*pi/180))^2)+(100*sin(-90*pi/180))^2)
4 Z3=sqrt(((15*cos(45*pi/180))^2)+(15*sin(45*pi/180))^2)

```

```

5 Z4=sqrt(((30*cos(30*pi/180))^2)+(30*sin(30*pi/180)
   )^2)
6 //mag(Z1*Z4)=mag(Z2*Z3)....magnitude condition
7 magl=Z1*Z4//lhs
8 magr=Z2*Z3//rhs
9 printf("magl=%f\n",magl)
10 printf("magr=%f\n",magr)
11 printf("lhs=rhs hence,magnitude condition is
   satisfied \n")
12 theta1=40
13 theta2=-90
14 theta3=45
15 theta4=30
16 //theta1+theta4=theta2+theta3.....angle condition
17 thetal=theta1+theta4//lhs
18 thetar=theta2+theta3//rhs
19 printf("thetal=%f\n",thetal)
20 printf("thetar=%f\n",thetar)
21 printf("angle condition is not satisfied \n")

```

---

#### Scilab code Exa 8.7 find equivalent series circuit

```

1 //Chapter -8,Example8_7 ,pg 8_37
2 C3=10*10^-6
3 R1=1.2*10^3
4 R2=100*10^3
5 R3=120*10^3
6 Rx=R2*R3/R1
7 Cx=R1*C3/R2
8 printf("equivalent series circuit\n")
9 printf("Rx=%f ohm\n",Rx)
10 printf("Cx=%f F",Cx)

```

---

**Scilab code Exa 8.8 find equivalent series circuit**

```
1 //Chapter-8,Example8_8 , pg 8_39
2 L3=8*10^-3
3 R1=1*10^3
4 R2=25*10^3
5 R3=50*10^3
6 Rx=R2*R3/R1
7 Lx=R2*L3/R1
8 printf("equivalent series circuit\n")
9 printf("Rx=%f ohm\n",Rx)
10 printf("Lx=%f H",Lx)
```

---

**Scilab code Exa 8.9 find components of branch BC**

```
1 //Chapter-8,Example8_9 , pg 8_44
2 //from the bridge
3 C1=0.5*10^-6
4 R1=1200
5 R2=700
6 R3=300
7 Rx=R2*R3/R1
8 Lx=R2*R3*C1
9 printf("components of branch RC\n")
10 printf("Rx=%f ohm\n",Rx)
11 printf("Lx=%f H\n",Lx)
```

---

**Scilab code Exa 8.10 find constants of unknown impedance**

```
1 //Chapter-8,Example8_10 , pg 8_49
2 //from hay's balance bridge
3 w=1000
4 R1=5.1*10^3
```

```

5 C1=2*10^-6
6 R2=7.9*10^3
7 R3=790
8 Rx=((w^2)*R1*(C1^2)*R2*R3)/(1+((w^2)*(R1^2)*(C1^2)))
9 Lx=R2*R3*C1/(1+((w^2)*(R1^2)*(C1^2)))
10 printf("unknown inductance and resistance\n")
11 printf("Rx=%g. f ohm\n",Rx)
12 printf("Lx=%g.5 f H",Lx)

```

---

**Scilab code Exa 8.11** calculate unknown capacitance and dissipation factor

```

1 //Chapter -8,Example8_11 ,pg 8_56
2 R1=1.2*10^3
3 R2=4.7*10^3
4 C1=1*10^-6
5 C3=1*10^-6
6 f=0.5*10^3
7 w=2*%pi*f
8 Rx=R2*C1/C3
9 Cx=R1*C3/R2
10 D=w*Cx*Rx
11 printf("unknown capacitance and resistance\n")
12 printf("Rx=%g. f ohm\n",Rx)
13 printf("Cx=%g.8 f F\n",Cx)
14 printf("dissipation factor\n")
15 printf("D=%g.3 f",D)

```

---

**Scilab code Exa 8.12** find deflection of galvanometer

```

1 //Chapter -8,Example8_12 ,pg 58
2 R1=200
3 R2=100
4 R3=1000

```

```

5 R4=2000
6 Rg=200
7 R41=2005 //changed by delR
8 Si=12 //sensitivity
9 E=10
10 Vth=E*((R41/(R3+R41))-(R1/(R1+R2)))
11 Req=(R1*R2/(R1+R2))+(R3*R41/(R3+R41))
12 Ig=Vth/(Rg+Req)
13 theta=Si*Ig*10^6 //deflection of galvanometer (mm)
14 printf(" deflection of galvanometer\n")
15 printf(" theta=%0.4 f mm",theta)

```

---

Scilab code Exa 8.13 find deflection of galvanometer

```

1 //Chapter -8,Example8_13 ,pg 59
2 R1=1000
3 R2=1000
4 R3=119
5 R4=121
6 Rg=200
7 S1=1
8 E=5
9 Vth=E*((R4/(R3+R4))-(R1/(R1+R2)))
10 Req=(R1*R2/(R1+R2))+(R3*R4/(R3+R4))
11 Ig=Vth/(Rg+Req)
12 theta=S1*Ig*10^6 //deflection of galvanometer (mm)
13 printf(" deflection of galvanometer\n")
14 printf(" theta=%0.4 f mm",theta)

```

---

Scilab code Exa 8.14 find current through galvanometer

```

1 //Chapter -8,Example8_14 ,pg 59
2 R=500

```

```

3 delR=20
4 E=10
5 Vth=E*delR/(4*R)
6 Req=R
7 Rg=125
8 Ig=Vth/(Req+Rg)
9 printf("current through galvanometer\n")
10 printf("Ig=%0.8f A",Ig)

```

---

**Scilab code Exa 8.15** calculate smallest change in resistance

```

1 //Chapter -8,Example8_15 ,pg 60
2 R=1000
3 E=20
4 Ig=1*10^-9
5 Req=R
6 //Ig=Vth/Req ..... Rg=0
7 delR=Ig*4*R^2/E
8 printf("change in resitance\n")
9 printf("delR=%0.8f ohm",delR)

```

---

**Scilab code Exa 8.16** calculate balance temperature and error

```

1 //Chapter -8,Example8_16 ,pg 61
2 //R4=Rv
3 R1=10*10^3
4 R2=10*10^3
5 R3=10*10^3
6 R4=R1*R3/R2
7 E=10
8 printf("bridge is balanced at 80deg. from graph when
      Rv=10k\n")
9 //at 60deg bridge is unbalanced

```

```

10 R4=9*10^3//from graph
11 e=E*((R4/(R3+R4))-(R1/(R1+R2)))//thevenin's voltage
12 printf("error voltage\n")
13 printf("e=%0.4f V\n",e)
14 printf("negative sign indicates opposite polarity of
      error voltage")

```

---

**Scilab code Exa 8.17** find value of unknown resistance

```

1 //Chapter-8,Example8_17 ,pg 8_62
2 R1=100
3 R2=10
4 R3=4
5 R4=50
6 E=10
7 Rg=20
8 Vth=E*((R4/(R3+R4))-(R1/(R1+R2)))
9 Req=(R1*R2/(R1+R2))+(R3*R4/(R3+R4))
10 Ig=Vth/(Rg+Req)
11 //for null deflection
12 R4=R3*R1/R2
13 printf("unbalanced current in galvanometer\n")
14 printf("Ig=%0.5f A\n",Ig)
15 printf("resistance for null deflection\n")
16 printf("R4=%0.f ohm",R4)

```

---

**Scilab code Exa 8.18** find unknown resistance and unbalance in bridge

```

1 //Chapter-8,Example8_18 ,pg 8_62
2 R1=1000
3 R2=100
4 R3=4*10^3
5 R4=40*10^3

```

```

6 Rth=(R1*R2/(R1+R2))+(R3*R4/(R3+R4))
7 Si=70
8 theta=3*10^-6 // deflection
9 E=10
10 Rg=80
11 delR=(theta*(Rth+Rg)*((R3+R4)^2))/(Si*E*R3)
12 printf("change in resistance\n")
13 printf("delR=%0.4 f ohm", delR)

```

---

#### Scilab code Exa 8.19 find series resistance

```

1 //Chapter -8, Example8_19 , pg 8_63
2 P=0.4
3 Rarm=150 //resistance in each arm
4 I=sqrt(P/Rarm) //P=(I^2)*R
5 //applying KVL to loop ABCEFA
6 r=1
7 E=25
8 R=(-I*Rarm-I*Rarm+E-2*I*r)/(2*I)
9 printf("series resistance\n")
10 printf("R=%0.4 f ohm", R)

```

---

#### Scilab code Exa 8.20 find unknown resistance Rx

```

1 //Chapter -8, Example8_20 , pg 8_63
2 R1=10
3 R2=R1/0.5 //given
4 Rba=1/1200 //Rb/Ra
5 Rx=R2*Rba
6 printf("unknown resistance\n")
7 printf("Rx=%0.4 f ohm", Rx)

```

---



Scilab code Exa 8.21 find constants of arm CD

```
1 //Chapter –8,Example8_21 ,pg 8_64
2 w=2*%pi*1000
3 C1=0.2*10^-6
4 R2=500
5 R3=300
6 C3=0.1*10^-6
7 Z4=(%i*w*C1*R2)/((1/R3)+(%i*w*C3))//from basic
   balance equaton
8 Zx=Z4//unknown impedance
9 Rx=real(Zx)
10 Xl=imag(Zx)
11 Lx=Xl/w//Xl=w*Lx
12 printf("unknown resistance\n")
13 printf("Rx=%0.2 f ohm\n",Rx)
14 printf("unknown inductance\n")
15 printf("Lx=%0.5 f H",Lx)
```

---

Scilab code Exa 8.22 find constants of Zx

```
1 //Chapter –8,Example8_22 ,pg 8_67
2 Z1=300
3 R2=200
4 w=2*%pi*10^3
5 C2=5*10^-6
6 Z2=R2-%i*(1/(w*C2))
7 R3=500
8 C3=0.2*10^-6
9 Z3=R3-%i*(1/(w*C3))
10 Z4=Z2*Z3/Z1//balance equation
11 Zx=Z4
```

```
12 printf("unknown impedance\n")
13 disp(Zx)
```

---

Scilab code Exa 8.23 find unknown impedance

```
1 //Chapter -8,Example8_23 ,pg 8_67
2 Z1=10*10^3
3 Z2=50*10^3
4 w=2*pi*2*10^3
5 C3=100*10^-6
6 R3=100*10^3
7 Z3=R3-%i*(1/(w*C3))
8 Z4=Z2*Z3/Z1
9 Zx=Z4
10 Rx=real(Zx)
11 Xc=-imag(Zx)
12 Cx=1/(Xc*w)
13 printf("unknown resistance\n")
14 printf("Rx=%f ohm\n",Rx)
15 printf("unknown capacitance\n")
16 printf("Cx=%f F",Cx)
```

---

Scilab code Exa 8.24 find unknown impedance and dissipation factor

```
1 //Chapter -8,Example8_24 ,pg 8_68
2 R2=4.8
3 r2=0.4
4 w=2*pi*450
5 C2=0.5*10^-6
6 Z2=R2+r2-%i*(1/(w*C2))
7 Z3=200
8 Z4=2850
9 //I1*Z1=I2*Z2..... null deflection detector
```

```

10 Z1=Z2*Z3/Z4
11 R1=real(Z1)
12 Xc1=-imag(Z1)
13 C1=1/(w*Xc1)
14 D=w*R1*C1//dissipation factor
15 printf("arm-1 resistance\n")
16 printf("R1=%0.4 f ohm\n",R1)
17 printf("arm-1 capacitance\n")
18 printf("C1=%0.6 f F\n",C1)
19 printf("dissipation factor\n")
20 printf("D=%0.6 f \n",D)

```

---

**Scilab code Exa 8.25** determine unknown parameters of arm AB

```

1 //Chapter -8,Example8_25 ,pg 8_70
2 R2=842
3 w=2*%pi*10^3
4 C2=0.135*10^-6
5 Z2=R2-%i*(1/(w*C2))
6 Z3=10
7 C4=10^-6
8 Z4=-%i*(1/(w*C4))
9 Z1=Z2*Z3/Z4
10 R1=real(Z1)
11 Xl1=imag(Z1)
12 L1=Xl1/w
13 printf("resistance of arm AB\n")
14 printf("R1=%0.3 f ohm\n",R1)
15 printf("inductance of arm AB\n")
16 printf("L1=%0.4 f H",L1)

```

---

**Scilab code Exa 8.26** find resistance and inductance of coil

```

1 //Chapter –8,Example8_26 ,pg 8_71
2 //balance is obtained when
3 L1=47.8*10^-3
4 R1=1.36
5 //at balance 100(r1+jwL1)=100((R2+r2)+jwL2)
6 L2=L1
7 r1=32.7
8 r2=r1-R1
9 printf("inductance of branch-CD\n")
10 printf("L2=%0.4f H\n",L2)
11 printf("resistance of branch-CD\n")
12 printf("r2=%0.2f ohm",r2)

```

---

**Scilab code Exa 8.27** find limiting values of unknown resistance

```

1 //Chapter –8,Example8_27 ,pg 8_72
2 R1=100
3 R2=100
4 R3=230
5 R4=R1*R3/R2
6 lerrR1=0.02/100
7 lerrR3=0.01/100
8 lerrR2=0.02/100//lerrR ..... limiting error in R
9 lerrR4=lerrR1+lerrR3+lerrR2
10 R4u=R4+lerrR4*R4
11 R4l=R4-lerrR4*R4//limiting ranges of R4
12 printf("limiting range of R4\n")
13 printf("upper limit=%0.3f ohm\n",R4u)
14 printf("lower limit=%0.3f ohm",R4l)

```

---

# Chapter 9

## DC Motors

Scilab code Exa 9.1 calculate generated emf

```
1 //Chapter -9,Example9_1 ,pg 9_14
2 P=4
3 Z=440
4 phi=0.07//flux(in Wb)
5 N=900
6 //for lap-wound
7 A=P
8 E=phi*P*N*Z/(60*A)
9 printf("e.m.f for lap wound\n")
10 printf("E=%f V\n",E)
11 //for wave wound
12 A=2
13 E=phi*P*N*Z/(60*A)
14 printf("e.m.f for wave wound\n")
15 printf("E=%f V\n",E)
```

---

Scilab code Exa 9.2 calculate speed and generated emf

```

1 //Chapter -9,Example9_2 , pg 9_15
2 P=4
3 phi=21*10^-3//flux (in Wb)
4 N=1120
5 C=42//coils
6 tpC=8//turns per coil
7 t=C*tpC//total turns
8 Z=2*t
9 //for lap wound
10 A=P
11 E=phi*P*N*Z/(60*A)
12 printf("e.m.f for lap wound\n")
13 printf("E=%f V\n",E)
14 //for wave wound
15 A=2
16 E=263.424
17 N=E*60*A/(phi*P*Z)
18 printf("speed of generator for wave wound\n")
19 printf("N=%f r.p.m\n",N)

```

---

### Scilab code Exa 9.3 calculate induced emf

```

1 //Chapter -9,Example9_3 , pg 9_20
2 V=220
3 Ia=30
4 Ra=0.75
5 Eb=V-Ia*Ra
6 printf("back e.m.f of motor\n")
7 printf("Ebb=%f V",Eb)

```

---

### Scilab code Exa 9.4 calculate back emf and speed

```

1 //Chapter -9,Example9_4 , pg 9_21

```

```

2 P=4
3 A=P
4 V=230
5 Ra=0.6
6 Z=250
7 phi=30*10^-3//flux (in Wb)
8 Ia=40
9 Eb=V-Ia*Ra
10 N=Eb*60*A/(phi*P*Z)
11 printf("back e.m. f\n")
12 printf("Eb=%f V\n",Eb)
13 printf("speed of motor\n")
14 printf("N=%f r.p.m",N)

```

---

#### Scilab code Exa 9.5 calculate gross torque

```

1 //Chapter -9,Example9_5 ,pg 9_24
2 P=4
3 A=P
4 Z=480
5 phi=20*10^-3//flux (in Wb)
6 Ia=50
7 Ta=0.159*phi*Ia*(P*Z/A)
8 printf("gross torque\n")
9 printf("Ta=%f N",Ta)

```

---

#### Scilab code Exa 9.6 calculate induced emf and lost torque

```

1 //Chapter -9,Example9_6 ,pg 9_25
2 P=4
3 A=P
4 No=1000//speed of motor
5 Z=540

```

```

6 V=230
7 phi=25*10^-3//flux(In Wb)
8 Ra=0.8
9 Ebo=phi*P*No*Z/(60*A)//induced e.m.f
10 Iao=(V-Ebo)/Ra//armature current
11 SL=Ebo*Iao//stray losses
12 wo=2*pi*No/60//angular velocity
13 Tf=Ebo*Iao/wo//loss torque
14 printf("induced e.m.f\n")
15 printf("Ebo=%f V\n",Ebo)
16 printf("armature current\n")
17 printf("Ia=%f A\n",Iao)
18 printf("stray losses\n")
19 printf("Sl=%f W\n",SL)
20 printf("loss torque\n")
21 printf("Tf=%f Nm",Tf)

```

---

#### Scilab code Exa 9.7 calculate speed

```

1 //Chapter -9,Example9_7 , pg 9_37
2 P=4
3 Z=200
4 V=250
5 A=2
6 phi=25*10^-3
7 Ia=60
8 Il=Ia
9 Ra=0.15
10 Rse=0.2
11 Eb=V-Ia*(Ra+Rse)
12 N=Eb*60*A/(phi*P*Z)
13 printf("speed of motor\n")
14 printf("N=%f r.p.m",N)

```

---



**Scilab code Exa 9.8** find armature current and back emf

```
1 //Chapter -9,Example9_8 , pg 9_38
2 V=250
3 I1=20
4 Ra=0.3
5 Rsh=200
6 Ish=V/Rsh
7 Ia=I1-Ish
8 Eb=V-Ia*Ra
9 printf("back e.m.f\n")
10 printf("Eb=%0.3 f V" ,Eb)
```

---

**Scilab code Exa 9.9** calculate speed on full load

```
1 //Chapter -9,Example9_9 , pg 9_38
2 No=1000
3 V=220
4 Rsh=110
5 Ra=0.3
6 Ish=V/Rsh
7 I1o=6
8 Iao=I1o-Ish
9 Rao=0.3
10 Ebo=V-Iao*Ra
11 //on full load
12 I1=50
13 IaFL=I1-Ish
14 EbFL=V-IaFL*Ra
15 //N=k*Eb/phi
16 NFL=No*EbFL/Ebo
17 printf("speed at full load\n")
```

```
18 printf("NFL=%0.3f r.p.m",NFL)
```

---

**Scilab code Exa 9.10** calculate speed on new load

```
1 //Chapter -9,Example9_10 ,pg 9_39
2 N1=800
3 I1=20
4 V=250
5 Ia1=I1
6 I2=50
7 Ia2=I2
8 Ra=0.2
9 Ise1=I1
10 Ise2=I2
11 Rse=0.3
12 Eb1=V-Ia1*Ra-Ise1*Rse
13 Eb2=V-Ia2*Ra-Ise2*Rse
14 //from speed equation
15 N2=N1*(Eb2/Eb1)*(Ia1/Ia2)
16 printf("speed of motor on new load\n")
17 printf("N2=%0.3f r.p.m",N2)
```

---

**Scilab code Exa 9.11** find new speed and armature current

```
1 //Chapter -9,Example9_11 ,pg 9_45
2 V=250
3 Rsh=250
4 Ra=0.25
5 Rx=Rsh
6 Ia1=20
7 Ish1=V/Rsh
8 Ish2=V/(Rsh+Rx)
9 N1=1500
```

```

10 Eb1=V-Ia1*Ra
11 //phi=k*Ish
12 //T1=T2
13 Ia2=Ish1*Ia1/Ish2//new current
14 Eb2=V-Ia2*Ra
15 //from speed equation
16 N2=N1*(((Eb1/Eb2)*(Ish2/Ish1))^-1)//new speed
17 printf("new current\n")
18 printf("Ia2=%0.1 f A\n",Ia2)
19 printf("new speed\n")
20 printf("N2=%0.3 f r.p.m",N2)

```

---

**Scilab code Exa 9.12 find external resistance**

```

1 //Chapter -9,Example9_12 ,pg 9_46
2 V=250
3 Ra=0.5
4 Rsh=250
5 Ia1=20
6 Ish1=V/Rsh
7 Eb1=V-Ia1*Ra
8 N1=600
9 N2=800
10 //T1=T2
11 //Ish1*Ia1=Ish2*Ia2
12 //Ish2*Ia2 = 20..... (1)
13 //(N1/N2)=(Eb1/Eb2)*(Ish2/Ish1)..... (2)
14 //using (1) and (2)
15 //240*(Ish2^2) -187.5*Ish2 + 7.5=0..... (3)
16 b=-187.5
17 a=240
18 c=7.5
19 Ish2=(-b+sqrt(((b^2)-4*a*c)))/(2*a)//neglecting
    lower value
20 Rx=(V/Ish2)-Rsh

```

```
21 printf("resistance in shunt feild\n")
22 printf("Rx=%0.3 f ohm",Rx)
```

---

Scilab code Exa 9.13 calculate speed of motor

```
1 //Chapter -9,Example9_13 ,pg 9_51
2 V=250
3 Ra=0.15
4 Rx=0.1
5 Rse=0.1
6 N1=800
7 Ise1=30
8 Ia1=30//Ia1=Ise1
9 I1=Ia1
10 //phi=k*Ise
11 //T2=T1+0.5*T1(increased by 50%) ..... (1)
12 //Ise2=Ia2*Rx/(Rx+Rse)
13 //putting values of Rx and Rse Ise2=0.5*Ia2
    ..... (2)
14 //putting (1) and (2) in torque equation
15 Ia2=sqrt(2700)
16 Ise2=0.5*Ia2//from (2)
17 Eb1=V-Ia1*Ra-Ise1*Rse
18 Eb2=V-Ia2*Ra-Ise2*Rse
19 //using speed equation
20 N2=N1*Eb2*Ise1/(Eb1*Ise2)
21 printf("speed of motor\n")
22 printf("N2=%0.3 f r.p.m",N2)
```

---

Scilab code Exa 9.14 find out speed of motor

```
1 //Chapter -9,Example9_14 ,pg 9_52
2 V=220
```

```

3 Ise1=15
4 Ia1=Ise1
5 Ia2=10
6 Ise2=Ia2
7 I2=Ia2
8 N1=900
9 Ra=0.5
10 Rse=0.5
11 Rx=4
12 Eb1=V-Ia1*Ra-Ise1*Rse
13 Eb2=V-Ia2*Ra-Ise2*Rse-I2*Rx
14 N2=N1*Eb2*Ise1/(Eb1*Ise2)
15 printf("speed of motor\n")
16 printf("N2=%0.3 f r.p.m",N2)

```

---

**Scilab code Exa 9.15** find speed and torque of motor

```

1 //Chapter -9,Example9_15 ,pg 9_64
2 P=6
3 V=500
4 A=2//wave wound
5 Z=1200
6 phi=20*10^-3//flux
7 Ra=0.5
8 Rsh=250
9 I1=20
10 Ish=V/Rsh
11 Ia=I1-Ish
12 Eb=V-Ia*Ra
13 N=Eb*60*A/(phi*P*Z)
14 Pm=Eb*Ia//mechanical power
15 w=2*%pi*N/60//angular velocity
16 Tg=Pm/w
17 ML=900//mechanical losses
18 Pout=Pm-ML

```

```

19 Tsh=Pout/w//usefull torque
20 Pin=V*I1
21 n=Pout*100/Pin//efficiency at load
22 printf("usefull torque\n")
23 printf("Tsh=%0.2 f Nm\n",Tsh)
24 printf("efficiency at load\n")
25 printf("n=%0.2 f",n)

```

---

**Scilab code Exa 9.16** find speed on full load

```

1 //Chapter -9,Example9_16 ,pg 9_65
2 V=120
3 Ra=0.2
4 Rsh=60
5 //for full load
6 I11=40
7 N1=1800
8 //for shunt motor
9 Ish=V/Rsh
10 Ia1=I11-Ish
11 Eb1=V-Ia1*Ra
12 //for half load T2=T1/2
13 Ia2=Ia1*0.5//T=k*Ia
14 Eb2=V-Ia2*Ra
15 N2=N1*Eb2/Eb1//from torque equation
16 printf("speed of motor\n")
17 printf("N2=%0.2 f r.p.m",N2)

```

---

**Scilab code Exa 9.17** determine armature current and speed of machine

```

1 //Chapter -9,Example9_17 ,pg 9_66
2 Ra=0.08
3 Eb1=242

```

```

4 V=250
5 Ia=87
6 Vt=V//generator supply
7 Nm=1500
8 Ia1=(V-Eb1)/Ra
9 //at start N=0, Eb=0
10 Ias=V/Ra//Ia(start)
11 Ia2=120
12 Eb2=V-Ia2*Ra
13 Eg=Vt+Ia*Ra//generator e.m.f
14 Ng=Nm*Eg/Eb1//speed as generator
15 printf("speed as generator\n")
16 printf("Ng=%.2 f r.p.m",Ng)

```

---

Scilab code Exa 9.18 determine mechanical power on full load

```

1 //Chapter -9,Example9_18 ,pg 9_67
2 V=250
3 Po=59680
4 Rsh=250
5 Ra=0.04
6 n=80//efficiency
7 N1=1200
8 I1=Po*100/(V*n)//Pi=V*I1
9 Ish=V/Rsh
10 Ia=I1-Ish
11 Eb=V-Ia*Ra
12 Pm=Eb*Ia//gross mechanical power
13 SL=Pm-Po//stray losses
14 printf("gross mechanical power\n")
15 printf("Pm=%.3 f W\n",Pm)
16 printf("stray losses\n")
17 printf("SL=%.2 f W\n",SL)
18 //on no load
19 //Pg=S, Ebo*Iao=SL.....(1)

```

```

20 //Ebo=V-Iao*Ra..... (2)
21 //putting (2) in (1)
22 //((Iao^2)-6250*Iao+278303.24=0
23 b=-6250
24 a=1
25 c=278303.24
26 Iao=(-b-sqrt((b^2)-4*a*c))/(2*a)
27 I=Iao-Ish//current drawn on no load
28 Ebo=V-Iao*Ra
29 No=N1*Ebo/Eb
30 printf("no load speed\n")
31 printf("No=%0.3 f r.p.m",No)

```

---

**Scilab code Exa 9.19** calculate full load speed

```

1 //Chapter -9,Example9_19 ,pg 9_69
2 V=250
3 P=4
4 Ra=0.1
5 Rsh=125
6 Vbr=2//brush drop
7 //no load condition
8 Ilo=4
9 No=1200
10 I11=61
11 Ish=V/Rsh
12 Iao=Ilo-Ish
13 Ebo=V-Iao*Ra-Vbr
14 //full load condition
15 //phi1=phio-0.05*phio (weakened by 5%)
16 //phi=phi1/phi0
17 phi=0.95
18 Ia1=I11-Ish
19 Eb1=V-Ia1*Ra-Vbr
20 N1=No*Eb1/(Ebo*phi)

```



```
21 printf(" full load speed\n")
22 printf("N1=%0.3 f r.p.m", N1)
```

---

Scilab code Exa 9.20 determine full load speed and efficiency

```
1 //Chapter -9, Example9_20 , pg 9_70
2 V=250
3 Ra=0.15
4 Rsh=166.67
5 No=1280
6 I11=67
7 Ish=V/Rsh
8 Ia1=I11-Ish
9 Eb1=V-Ia1*Ra
10 //on no load
11 Ilo=6.5
12 Ish=1.5
13 Iao=Ilo-Ish
14 Ebo=V-Iao*Ra
15 N1=Eb1*No/Ebo
16 Sr=(No-N1)*100/No//speed regulation
17 SL=Ebo*Iao
18 Po=Eb1*Ia1-SL//full load shaft output
19 hp=Po/746//horse power rating
20 Pi=V*I11
21 n=Po*100/Pi
22 printf(" full load speed\n")
23 printf("N1=%0.3 f r.p.m\n", N1)
24 printf(" speed regulation\n")
25 printf(" Sr=%0.2 f \n", Sr)
26 printf(" hp rating of machine\n")
27 printf(" hp=%0.2 f hp\n", hp)
28 printf(" full load efficiency\n")
29 printf(" n=%0.2 f ", n)
```

---

**Scilab code Exa 9.21** find speed for parallel field groups

```
1 //Chapter –9,Example9_21 ,pg 9_71
2 Ra=0.1
3 V=110
4 P=4
5 Ia1=50
6 I1=Ia1
7 Rse=0.02
8 N1=700
9 Eb1=V-Ia1*Ra-Ia1*Rse
10 //using torque equation  $T=k*\phi*Ia$ 
11 Ia2=sqrt(2)*Ia1
12 Eb2=V-Ia2*Ra-Ia2*Rse/4//parallel speed groups
13 //using speed equation  $N=k*Eb/\phi$ 
14 N2=N1*Eb2*2*Ia1/(Eb1*Ia2)
15 printf("speed of motor\n")
16 printf("N2=%0.3 f r.p.m",N2)
```

---

**Scilab code Exa 9.22** find new speed and armature current

```
1 //Chapter –9,Example9_22 ,pg 9_73
2 P=4
3 Ia1=50
4 N1=2000
5 V=230
6 //coils connected in series
7 //phi1=k*Ia1*(4*n)=k*200*n
8 //coils connected in parallel groups of series coils
9 //phi2=k*((Ia2*2*n/2)+(Ia2*2*n/2))=k*2*n*Ia2
10 //phi1/phi2=100/Ia2 ..... (1)
11 //N1/N2=phi2/phi1 ..... (2)
```

```

12 //T=kN ^ 2.....(3)
13 Ia2=(Ia1*(100^3))^(1/4)//using (1) in (3)
14 N2=(((N1^3)*Ia2)/Ia1)^(1/3)
15 printf("new speed of motor\n")
16 printf("N2=%0.3 f r.p.m",N2)

```

---

**Scilab code Exa 9.23** find external resistance

```

1 //Chapter -9,Example9_23 ,pg 9_76
2 V=200
3 Ia1=30
4 Ra=0.75
5 Rse=0.75
6 R=Ra+Rse
7 Eb1=V-Ia1*R
8 //N2=0.6*N1
9 N=0.6//N=N2/N1
10 //using T=k*Ia^2 and T=k*N^3
11 Ia2=sqrt(((0.6^3)*30^2))
12 //using speed equation N=k*Eb/Ia
13 Eb2=N*Eb1*Ia2/Ia1
14 //Eb2=V-Ia2*(R+Rx)
15 Rx=-(Eb2-V+Ia2*R)/Ia2
16 printf("extra resistance to reduce speed\n")
17 printf("Rx=%0.3 f ohm",Rx)

```

---

**Scilab code Exa 9.24** estimate supply voltage

```

1 //Chapter -9,Example9_24 ,pg 9_77
2 R=1
3 V1=230
4 N1=300
5 Ia1=15

```

```

6 N2=375
7 //using torque equation  $T=k*N^2$ 
8 Ia2=N2*Ia1/N1
9 //using speed equation  $N=k*Eb/Ia$  .....(1)
10 Eb1=V1-Ia1*R
11 //case-2
12 //Eb2=V2-Ia2*R=V2-18.75.....(2)
13 //putting (2) in (1)
14 V2=(N2*Eb1*Ia2/(N1*Ia1))+18.75
15 printf("new supply voltage\n")
16 printf("V2=%0.3 f V",V2)

```

---

**Scilab code Exa 9.25** find efficiency and power input

```

1 //Chapter-9,Example9_25 ,pg 9_78
2 V=400
3 Po1=18.5*10^3
4 Pi1=22.5*10^3
5 Rsh=200
6 Ra=0.4
7 Po2=9*10^3
8 I1=Pi1/V
9 Ish=V/Rsh
10 Ia1=I1-Ish
11 Ac1=(Ia1^2)*Ra//armature copper loss
12 Sc1=(Ish^2)*Rsh//shunt feild copper loss
13 TL=Pi1-Po1//total losses
14 SF1=TL-(Ac1+Sc1)//stray and friction loss
15 //case-2
16 Pm=Po2+SF1//mechanical power
17 //Pm=Eb2*Ia2 ..... (1)
18 //Eb2=V-Ia2*Ra ..... (2)
19 //using (1) and (2)
20 //0.4*(Ia2^2)-400*Ia2+11022.75=0
21 a=0.4

```

```

22 b=-400
23 c=11022.775
24 Ia2=(-b-sqrt((b^2)-4*a*c))/(2*a)//neglecting higher
    value
25 Pi2=Po2+(Ia2^2)*Ra+(Ish^2)*Rsh+SF1
26 n=Po2*100/Pi2//efficiency
27 printf("power input in case -2\n")
28 printf("Pi2=%0.3f W\n",Pi2)
29 printf("efficiency of motor\n")
30 printf("n=%0.2f ",n)

```

---

Scilab code Exa 9.26 calculate efficiency and armature current

```

1 //Chapter -9,Example9-26 ,pg 9-79
2 V=250
3 Ilo=4
4 Ra=1
5 Rsh=250
6 Ish=V/Rsh
7 Il1=20
8 Iao=Ilo-Ish
9 Ia1=Il1-Ish
10 Ebo=V-Iao*Ra
11 Po=Ebo*Iao
12 Eb1=V-Ia1*Ra
13 P1=Eb1*Ia1
14 Pout=P1-Po
15 Pi=V*Il1
16 n=Pout*100/Pi
17 //fro max. efficiency
18 //const. losses=variable losses
19 Ia=sqrt(Po+(Ish^2)*Rsh)
20 Ebm=V-Ia*Ra
21 Pm=Ebm*Ia
22 Pout=Pm-Po

```

```
23 Pi=V*(Ia+Ish)
24 nm=Pout*100/Pi
25 printf("maximum efficiency\n")
26 printf("nm=%0.2 f",nm)
```

---

#### Scilab code Exa 9.27 calculate back emf

```
1 //Chapter -9,Example9_27 ,pg 9_81
2 V=250
3 FLo=16*10^3//full scale output
4 n=80
5 I=FLo*100/n//input
6 I1=I/V
7 I1=I1
8 Ia=1.5*I1
9 //at start
10 Ra=V/Ia
11 Rac=0.18//Ra actual
12 Ras=Ra-Rac//Ra starter
13 Ia=I1//Ia drops as motor starts
14 Eb=V-Ia*(Ra)
15 printf("back e.m. f\n")
16 printf("Eb=%0.2 f V",Eb)
```

---

#### Scilab code Exa 9.28 calculate torque and efficiency

```
1 //Chapter -9,Example9_28 ,pg 9_82
2 Po=20*735.5//(in W)
3 V=230
4 N=1150
5 P=4
6 A=P
7 Z=882
```

```

8 Ia=73
9 Ish=1.6
10 T=60*Po/(2*%pi*N)
11 phi=T*A/(0.159*Ia*P*Z)//flux per pole
12 Il=Ia+Ish
13 Pin=V*Il
14 n=Po*100/Pin
15 printf("electromagnetic torque\n")
16 printf("T=%0.3f Nm\n",T)
17 printf("flux per pole\n")
18 printf("phi=%0.3f Wb\n",phi)
19 printf("efficiency of motor\n")
20 printf("n=%0.3f",n)

```

---

**Scilab code Exa 9.29** determine efficiency and speed of motor

```

1 //Chapter -9,Example9_29 ,pg 9_83
2 Pr=12*10^3//rated output
3 V=200
4 Rsh=80
5 N1=800
6 n=0.9//efficiency
7 Out=0.8*Pr//output is 80% of rated
8 In=Out/n//input
9 TL=In-Out
10 //for max. efficiency
11 Iln=70//new current
12 //TL=Wc+(Ia1^2)*Ra
13 //bur Wc=(Ia1^2)*Ra
14 Wc=TL/2
15 Il=In/V
16 Ish=V/Rsh
17 Ia1=Il-Ish
18 Ra=Wc/(Ia1^2)
19 Ia2=Iln-Ish

```

```

20 Wcn=Wc//const. losses remain same
21 TL=(Ia2^2)*Ra+Wcn
22 Pi=V*Iln
23 n=(Pi-TL)*100/Pi
24 Eb1=V-Ia1*Ra
25 Eb2=V-Ia2*Ra
26 N2=N1*Eb2/Eb1
27 printf("speed of motor\n")
28 printf("N2=%0.3 f r.p.m",N2)

```

---

**Scilab code Exa 9.30** calculate efficiency of motor

```

1 //Chapter -9,Example9_30 ,pg 9_85
2 Po=8.952*10^3
3 V=440
4 Ra=1.1
5 Rsh=650
6 Rint=0.4
7 Rreg=50
8 Ml=450
9 Vbr=2//brush drop
10 Il=24
11 Rat=Ra+Rint//series connection
12 Rsht=Rsh+Rreg//series connection
13 Ish=V/Rsht
14 Ia=Il-Ish
15 Acl=(Ia^2)*Rat//armature copper loss
16 Fcl=(Ish^2)*Rsht//feild copper loss
17 Bdl=Vbr*Ia//brush drop loss
18 TL=Acl+Fcl+Bdl+Ml
19 n=Po*100/(Po+TL)
20 printf("efficiency of motor\n")
21 printf("n=%0.2 f ",n)

```

---



**Scilab code Exa 9.31** calculate speed of motor combination

```
1 //Chapter -9,Example9_31 ,pg 9_85
2 //for first motor
3 N1=700
4 R=0.5 //Ra+Rse
5 I1=70
6 V=500
7 Eb1=V-I1*R
8 K1=Eb1/(N1*I1)
9 //for second motor
10 N2=750
11 R=0.5
12 I2=70
13 V=500
14 Eb2=V-I2*R
15 K2=Eb2/(N2*I2)
16 //motors in series
17 It=70
18 Rt=2*R
19 Eb=V-It*Rt
20 N=Eb/(K1*It+K2*It)
21 printf("speed of motors\n")
22 printf("N=%0.3 f r.p.m",N)
```

---

**Scilab code Exa 9.32** calculate efficiency and power output

```
1 //Chapter -9,Example9_32 ,pg 9_86
2 Po=7.46*10^3
3 V=250
4 Ilo=5
5 Ra=0.5
```

```

6 Rsh=250
7 Ish=V/Rsh
8 Iao=Ilo-Ish
9 Acl=(Iao^2)*Ra
10 Fcl=(Ish^2)*Rsh
11 Pi=V*Ilo
12 FWl=Pi-Acl-Fcl//friction and windage loss
13 //Pin=Eb*Ia=(V-Ia*Ra)*Ia
14 //0.5*(Ia^2)-250*Ia+8452=0
15 b=-250
16 a=0.5
17 c=8452
18 Ia=(-b-sqrt((b^2)-4*a*c))/(2*a)//neglecting higher
    value
19 TL=(Ia^2)*Ra+(Ish^2)*Rsh+FWl
20 n=Po*100/(Po+TL)
21 //for max. efficiency
22 Ia=sqrt((FWl+Fcl)/Ra)
23 Eb=V-Ia*Ra
24 Pm=Eb*Ia
25 //Po at nmax
26 Po=Pm-FWl
27 printf("maximum efficiency output\n")
28 printf("Po=%0.3f W",Po)

```

---

**Scilab code Exa 9.33** calculate speed on given load

```

1 //Chapter -9,Example9_33 ,pg 9_87
2 V=500
3 Ra=1.2
4 Rsh=500
5 Ish=V/Rsh
6 Ilo=4
7 Iao=Ilo-Ish
8 Ebo=V-Iao*Ra

```

```
9 I11=26
10 Ish1=1
11 Ia1=I11-Ish1
12 Eb1=V-Ia1*Ra
13 No=1000
14 N1=No*Eb1/Ebo
15 Rx=2.3//connected in series with armature
16 Eb2=V-Ia1*(Ra+Rx)
17 N2=N1*Eb2/Eb1
18 printf("speed of motor case-1\n")
19 printf("N2=%0.3f r.p.m\n",N2)
20 Ish3=Ish1-0.15*Ish1//reduced by 15%
21 Ia3=Ish1*Ia1/(Ish3)
22 Eb3=V-Ia3*Ra
23 N3=N1*Eb3*Ish1/(Eb1*Ish3)
24 printf("speed of motor case-2\n")
25 printf("N3=%0.3f r.p.m\n",N3)
```

---

# Chapter 10

## Three Phase Induction Motors

Scilab code Exa 10.1 calculate full load slip

```
1 //Chapter -10,Example10.1 ,pg10_14
2 P=4
3 f=50
4 N=1410
5 Ns=120*f/P
6 s=(Ns-N)/Ns
7 s=s*100 // %s
8 printf("full load slip\n")
9 printf("s=%0.1f ",s)
```

---

Scilab code Exa 10.2 calculate full load speed

```
1 //Chapter -10,Example10.2 ,pg10_14
2 P=4
3 f=50
4 sfl=4/100
5 Ns=120*f/P
6 Nfl=Ns-sfl*Ns
```

```
7 printf("full load speed of motor\n")
8 printf("Nfl=%f r.p.m",Nfl)
```

---

**Scilab code Exa 10.3** calculate rotor frequency

```
1 //Chapter -10,Example10_3 ,pg10-16
2 P=4
3 f=50
4 N=1470
5 Ns=120*f/P
6 s=(Ns-N)/Ns
7 fr=s*f
8 printf("frequency of induced e.m.f\n")
9 printf("fr=%f Hz",fr)
```

---

**Scilab code Exa 10.4** find full load slip and speed

```
1 //Chapter -10,Example10_4 ,pg10-20
2 P=8
3 f=50
4 fr=2
5 s=fr/f
6 s=s*100
7 printf("full load slip\n")
8 printf("s=%f \n",s)
9 s=s/100
10 Ns=120*f/P
11 N=Ns*(1-s)
12 printf("speed of motor\n")
13 printf("N=%f r.p.m",N)
```

---

Scilab code Exa 10.5 calculate rotor frequency and induced emf

```
1 //Chapter -10,Example10_5 , pg10_20
2 P=4
3 f=50
4 N=1455
5 E1line=415
6 Ns=120*f/P
7 s=(Ns-N)/Ns
8 fr=s*f
9 E1ph=E1line/sqrt(3)
10 E2ph=0.5*E1ph//K=2
11 E2r=s*E2ph
12 printf("frequency of rotor e.m.f\n")
13 printf("fr=%.2f Hz\n",fr)
14 printf("magnitude of induced e.m.f standstill\n")
15 printf("E2ph=%.2f V\n",E2ph)
16 printf("magnitude of induced e.m.f running\n")
17 printf("E2r=%.3f V",E2r)
```

---

Scilab code Exa 10.6 find rotor current and rotor power factor

```
1 //Chapter -10,Example10_6 , pg10_21
2 P=4
3 f=50
4 R2=0.2
5 X2=1
6 E2line=120
7 E2ph=E2line/sqrt(3)
8 Ns=120*f/P
9 //at start
10 pf=R2/sqrt((R2^2)+(X2^2))//power factor
11 I2=E2ph/sqrt((R2^2)+(X2^2))
12 printf(" at start\n")
13 printf("pf=%.3f lagging\n",pf)
```

```

14 printf(" I2=%0.2 f A\n", I2)
15 //on full load
16 N=1440
17 s=(Ns-N)/Ns
18 pf=R2/sqrt((R2^2)+((s*X2)^2))
19 I2=E2ph*s/sqrt((R2^2)+((s*X2)^2))
20 printf(" on full load\n")
21 printf(" pf=%0.3 f lagging\n", pf)
22 printf(" I2=%0.2 f A", I2)

```

---

Scilab code Exa 10.7 calculate full load torque

```

1 //Chapter -10, Example10_7 , pg10_24
2 P=4
3 f=50
4 R2=0.1
5 X2=1
6 N=1440
7 K=0.5
8 Ns=120*f/P
9 E1line=400
10 E1ph=E1line/sqrt(3)
11 E2ph=0.5*E1ph
12 s=(Ns-N)/Ns
13 ns=Ns/60//synchronous speed (r.p.s)
14 T=(3/(2*pi*ns))*(s*(E2ph^2)*R2/((R2^2)+((s*X2)^2)))
15 printf("torque on full load\n")
16 printf("T=%0.2 f N-m", T)

```

---

Scilab code Exa 10.8 calculate starting torque and full load torque

```

1 //Chapter -10, Example10_8 , pg10_27
2 P=4

```

```

3 f=50
4 K=1/4
5 R2=0.01
6 X2=0.1
7 E1line=400
8 E1ph=E1line/sqrt(3)
9 E2=E1ph/4
10 Ns=120*f/P
11 //at start
12 s=1
13 ns=Ns/60
14 k=3/(2*pi*ns)
15 Tst=k*(E2^2)*R2/((R2^2)+(X2^2))
16 printf("starting torque\n")
17 printf("Tst=%0.3 f N-m\n",Tst)
18 //slip at max torque
19 sm=R2/X2
20 sm=sm*100
21 printf("slip at which max torque occurs\n")
22 printf("sm=%0. f \n",sm)
23 //speed at max torque
24 sm=sm/100
25 N=Ns*(1-sm)
26 printf("speed at which max torque occurs\n")
27 printf("N=%0. f r.p.m\n",N)
28 //max. torque
29 Tm=k*(E2^2)/(2*X2)
30 sf=0.04
31 Tfl=k*sf*(E2^2)*R2/((R2^2)+((sf*X2)^2))
32 printf("max torque\n")
33 printf("Tm=%0.2 f N-m\n",Tm)
34 printf("full load torque\n")
35 printf("Tfl=%0.2 f N-m",Tfl)

```

---

Scilab code Exa 10.9 star connected induction motor



```

1 //Chapter -10,Example10_9 ,pg10_33
2 P=24
3 f=50
4 R2=0.016
5 X2=0.265
6 N=247
7 Ns=120*f/P
8 sf=(Ns-N)/Ns
9 sm=R2/X2
10 Tfm=2*sm*sf/((sm^2)+(sf^2))
11 Tsm=2*sm/(1+(sm^2))
12 printf("full load torque to max torque\n")
13 printf("Tfm=%0.4f \n",Tfm)
14 printf("starting torque to max torque\n")
15 printf("Tsm=%0.4f \n",Tsm)

```

---

Scilab code Exa 10.10 calculate external resistance

```

1 //Chapter -10,Example10_10 ,pg10_36
2 R2=0.04
3 X2=0.2
4 //for Tm=Tst , sm=1
5 R21=X2
6 Rex=R2-R21
7 //for Tst=Tm/2.....(1)
8 //Tst=k*(E2^2)*R21/((R21^2)+(X2^2)) .....(2) with
   added resistance
9 //from (1) and (2)
10 //(R21^2)-0.8*R21+0.04=0
11 a=1
12 b=-0.8
13 c=0.04
14 R21=(-b-sqrt((b^2)-4*a*c))/(2*a)//neglecting higher
   value
15 Rex=R21-R2

```

```
16 printf(" external resistance \n")
17 printf("Rex=%0.4 f ohm per phase",Rex)
```

---

**Scilab code Exa 10.11** calculate rotor copper loss

```
1 //Chapter –10,Example10_11 , pg10_42
2 Tsh=190
3 P=8
4 f=50
5 fr=1.5
6 ML=700
7 s=fr/f
8 Ns=120*f/P
9 N=Ns*(1-s)
10 Po=Tsh*(2*pi*N/60)
11 Pm=Po+ML
12 Pc=Pm*s/(1-s)
13 printf(" rotor copper loss \n")
14 printf("Pc=%0.3 f W",Pc)
```

---

**Scilab code Exa 10.12** calculate full load efficiency

```
1 //Chapter –10,Example10_12 , pg10_43
2 P=4
3 f=50
4 Pi=50*10^3
5 N=1440
6 S1=1000
7 F1=650
8 Ns=120*f/P
9 s=(Ns-N)/Ns
10 P2=Pi-S1
11 Pc=s*P2
```

```

12 Pm=P2-Pc
13 Po=Pm-F1
14 n=Po*100/Pi
15 printf("full load efficiency\n")
16 printf("n=%0.2 f",n)

```

---

Scilab code Exa 10.13 calculate slip and rotor resistance per phase

```

1 //Chapter-10,Example10_13 ,pg10_44
2 P=4
3 f=50
4 Tsh=300
5 Tlost=50
6 fr=120/60//Hz
7 s=fr/f
8 s=s*100
9 printf("slip s=%0.f \n",s)
10 Ns=120*f/P
11 s=s/100
12 N=Ns*(1-s)
13 Po=Tsh*2*%pi*N/60
14 F1=Tlost*2*%pi*N/60
15 Pm=Po+F1
16 Pc=Pm*s/(1-s)
17 Rcl=Pc/3//rotor copper loss per phase
18 P2=Pc/s
19 n=Pm*100/P2
20 I2r=60
21 R2=Rcl/(I2r^2)
22 printf("net output power\n")
23 printf("Po=%0.3 f W\n",Po)
24 printf("rotor copper loss per phase\n")
25 printf("Rcl=%0.3 f W\n",Rcl)
26 printf("rotor efficiency\n")
27 printf("n=%0.2 f \n",n)

```

```
28 printf("rotor resistance per phase\n")
29 printf("R2=%0.4 f ohm/ph" ,R2)
```

---

**Scilab code Exa 10.14** calculate gross mechanical power and efficiency

```
1 //Chapter –10,Example10_14 , pg10_45
2 Po=25*10^3
3 f=50
4 P=4
5 Ns=120*f/P
6 N=1410
7 s=(Ns -N)/Ns
8 Ml=850
9 Pm=Po+Ml
10 Pc=Pm*s/(1-s)
11 I2r=65
12 R2=Pc/(3*(I2r^2))
13 S1=1.7*Pc
14 P2=Pc/s
15 Pin=P2+S1
16 n=Po*100/Pin
17 printf("gross mechanical power\n")
18 printf("Pm=%0. f W\n" ,Pm)
19 printf("rotor copper losses\n")
20 printf("Pc=%0. f W\n" ,Pc)
21 printf("rotor resistance per phase\n")
22 printf("R2=%0.3 f ohm/ph\n" ,R2)
23 printf("full load efficiency\n")
24 printf("n=%0.2 f" ,n)
```

---

**Scilab code Exa 10.15** calculate shaft torque and full load efficiency

```
1 //Chapter –10,Example10_15 , pg10_47
```

```

2 Po=24*10^3
3 I1=57
4 Is=I1
5 P=8
6 N=720
7 f=50
8 V1=415
9 pf=0.707
10 Ns=120*f/P
11 s=(Ns-N)/Ns
12 Ml=1000
13 Pm=Po+Ml
14 Pc=Pm*s/(1-s)
15 Tsh=Po*60/(2*pi*N)
16 T=Pm*60/(2*pi*N)
17 Rcl=1041.66//rotor copper loss
18 P2=Pc/s
19 Pi=sqrt(3)*V1*I1*pf
20 Rs=0.1
21 Scl=3*(Is^2)*Rs//stator copper loss
22 Sl=Pi-P2
23 Sil=Sl-Scl//stator iron loss
24 n=Po*100/Pi
25 printf("shaft torque\n")
26 printf("Tsh=%0.3f N-m\n",Tsh)
27 printf("gross torque \n")
28 printf("T=%0.3f N-m\n",T)
29 printf("rotor copper losses\n")
30 printf("Pc=%0.2f W\n",Pc)
31 printf("stator copper losses\n")
32 printf("Scl=%0.2f W\n",Scl)
33 printf("stator iron losses\n")
34 printf("Sil=%0.2f W\n",Sil)
35 printf("overallefficiency\n")
36 printf("n=%0.2f",n)

```

---

Scilab code Exa 10.16 calculate tapping and supply start current

```
1 //Chapter -10,Example10_16 , pg10_52
2 sf=0.05
3 //Tst=Tfl
4 Ifs=1/6//Isc/Ifl=6
5 x=sqrt((Ifs^2)/sf)//tapping on transformer
6 t=x*100
7 Ist=(x^2)*6
8 printf("supply current\n")
9 printf("Ist=%0.2f times Ifl",Ist)
```

---

Scilab code Exa 10.17 determine ratios of torques

```
1 //Chapter -10,Example10_17 , pg10_54
2 R2=0.4
3 X2=4
4 //Tm=k*(E2^2)/(2*X2)
5 // Tfl=Tm/2.5
6 // Tfl=k*(E2^2)/20
7 // Tst=k*(E2^2)*R2/((R2^2)+(X2^2))
8 //E2=E2/sqrt(3)
9 T=20*R2/(3*((R2^2)+(X2^2)))
10 printf("ratio of starting torque to full load torque
    \n")
11 printf("T=%0.3f ",T)
```

---

Scilab code Exa 10.18 calculate rotor current and external resistance

```

1
2 //Chapter -10,Example10_18 , pg10_57
3 V1=1000
4 f=50
5 K=3.6
6 R2=0.01
7 X2=0.2
8 E1line=1000
9 E1=E1line/sqrt(3)
10 E2=E1/K
11 //at start ,s=1
12 I2=160.37/sqrt((R2^2)+(X2^2))
13 pf=R2/sqrt((R2^2)+(X2^2))
14 printf("rotor current at start\n")
15 printf("I2=%0.2f A\n",I2)
16 printf("rotor power factor\n")
17 printf("pf=%0.3f lagging (answer in book is wrong)\n"
    ,pf)
18 //at s=0.03
19 s=0.03
20 I2r=s*160.37/sqrt((R2^2)+((s*X2)^2))
21 printf("rotor current at slip 0.03\n")
22 printf("I2r=%0.2f A\n",I2r)
23 I2=200
24 R21=sqrt(((E2/I2)^2)-(X2^2))
25 Rex=R21-R2
26 printf("external resistance \n")
27 printf("Rex=%0.4f ohm/ph (answer in book is wrong)",
    Rex)

```

---

**Scilab code Exa 10.19** calculate starting torque and speed

```

1 //Chapter -10,Example10_19 , pg10_58
2 P=12
3 f=50

```

```

4 R2=0.15
5 X2=0.25
6 E2=32
7 Ns=120*f/P
8 ns=Ns/60
9 Tst=3*(E2^2)*R2/((2*pi*ns)*((R2^2)+(X2^2)))
10 N=480
11 s=(Ns-N)/Ns
12 Tfl=3*s*(E2^2)*R2/((2*pi*ns)*((R2^2)+((s*X2)^2)))
13 Tm=3*(E2^2)/(2*pi*ns*2*X2)
14 sm=R2/X2
15 N=Ns*(1-sm)
16 printf("starting torque\n")
17 printf("Tst=%0.2 f Nm\n",Tst)
18 printf("full load torque\n")
19 printf("Tfl=%0.3 f Nm\n",Tfl)
20 printf("maximum torque\n")
21 printf("Tm=%0.3 f Nm\n",Tm)
22 printf("speed at max torque\n")
23 printf("N=%0. f r.p.m",N)

```

---

**Scilab code Exa 10.20** calculate efficiency on full load

```

1 //Chapter-10,Example10_20 ,pg10_59
2 Po=50*735.5//(in W)
3 s=0.04
4 //Rcl=X.....rotor copper loss
5 //Sil=1.25X.....stator iron loss
6 //Ml=Y, Y=(Y+1.25X)/3, Y=0.625X
7 //TL=Sil+Rcl+Scl+Ml, TL=3.875X.....( a)
8 //Pm=Po+Y, 36775+625X.....( 1)
9 //Pc=Pm*s/(1-s) .....( 2)
10 //Pc=X, from (1) and (2)
11 X=(s*Po)/(1-s-s*0.625)
12 TL=3.875*X//from (a)

```



```

13 n=Po*100/(Po+TL)
14 printf("efficiency on full load\n")
15 printf("n=%0.2f ",n)

```

---

Scilab code Exa 10.21 calculate new speed

```

1 //Chapter -10,Example10_21 , pg10_61
2 P=4
3 f=50
4 R2=0.25
5 X2=0.55
6 Ns=120*f/P
7 N1=1440
8 s1=(Ns-N1)/Ns
9 Rex=0.2
10 R21=R2+Rex
11 //T1 at s1=T2 at s2
12 //0.3025*s2^2-2.8342*s2+0.2025=0, s1=0.04
13 a=0.3025
14 b=-2.8342
15 c=0.2025
16 s2=(-b-sqrt((b^2)-4*a*c))/(2*a)//neglecting higher
    value
17 N2=Ns*(1-s2)
18 printf("new speed of motor\n")
19 printf("N2=%0.f r.p.m",N2)

```

---

Scilab code Exa 10.22 find rotor current and rotor emf per phase

```

1 //Chapter -10,Example10_22 , pg10_62
2 E2line=50
3 R2=0.5
4 X2=3

```

```

5 E2=E2line/sqrt(3)
6 //at start
7 s=1
8 I2r=s*E2/(sqrt((R2^2)+((s*X2)^2)))
9 printf("rotor current atstart\n")
10 printf(" I2r=%0.3 f A\n", I2r)
11 Rx=6
12 I2r=s*E2/(sqrt(((R2+Rx)^2)+((s*X2)^2)))
13 printf("rotor current for rheostat of 6 ohm\n")
14 printf(" I2r=%0.3 f A\n", I2r)
15 //at full load
16 s=0.04
17 I2r=s*E2/(sqrt((R2^2)+((s*X2)^2)))
18 pf=R2/(sqrt((R2^2)+((s*X2)^2)))
19 printf("full load rotor current\n")
20 printf(" I2r=%0.3 f A\n", I2r)
21 printf("full load power factor\n")
22 printf(" pf=%0.3 f lagging\n", pf)
23 E2r=s*E2
24 printf("rotor e.m.f on full load\n")
25 printf(" E2r=%0.3 f V", E2r)

```

---

**Scilab code Exa 10.23** calculate starting torque and speed

```

1 //Chapter –10, Example10.23 , pg10.63
2 P=12
3 f=50
4 R2=0.15
5 X2=0.25
6 E2=32
7 Ns=120*f/P
8 ns=Ns/60
9 k=3
10 Tst=k*(E2^2)*R2/((2*pi*ns)*((R2^2)+(X2^2)))
11 N=480

```

```

12 s=(Ns-N)/Ns
13 Tfl=k*s*(E2^2)*R2/((2*pi*ns)*((R2^2)+((s*X2)^2)))
14 Tm=k*(E2^2)/(2*pi*ns*2*X2)
15 sm=R2/X2
16 N=Ns*(1-sm)
17 printf("starting torque\n")
18 printf("Tst=%0.2 f Nm\n",Tst)
19 printf("full load torque\n")
20 printf("Tfl=%0.3 f Nm\n",Tfl)
21 printf("maximum torque\n")
22 printf("Tm=%0.3 f Nm\n",Tm)
23 printf("speed at max torque\n")
24 printf("N=%0. f r.p.m",N)

```

---

Scilab code Exa 10.24 calculate full load torque and external resistance

```

1 //Chapter -10,Example10_24 , pg10_64
2 P=4
3 f=50
4 R2=0.4
5 X2=2
6 E2b=520//between slip rings
7 E2ph=E2b/sqrt(3)
8 Ns=120*f/P
9 N=1425
10 sf=(Ns-N)/Ns
11 ns=Ns/60
12 Tfl=3*sf*(E2ph^2)*R2/((2*pi*ns)*((R2^2)+((sf*X2)^2)
    ))
13 Tst=3*(E2ph^2)*R2/((2*pi*ns)*((R2^2)+((X2)^2)))
14 T=Tst/Tfl
15 Tm=3*(E2ph^2)/((2*pi*ns)*((R2^2)+((X2)*2)))
16 T1=Tm/Tfl
17 //at start
18 sm=1

```

```

19 R21=X2
20 Rex=R21-R2
21 printf(" full load torque\n")
22 printf(" Tfl=%0.2 f Nm\n",Tf1)
23 printf(" ratio of Tst to Tf1\n")
24 printf("T=%0.4 f \n",T)
25 printf(" ratio of Tm to Tf1\n")
26 printf(" T1=%0.4 f \n",T1)
27 printf(" external resistance required\n")
28 printf(" Rex=%0.2 f ohm/ph",Rex)

```

---

**Scilab code Exa 10.25** calculate slip and line current

```

1 //Chapter -10,Example10_25 , pg10_65
2 Po=33.73*10^3
3 P=4
4 V1=400
5 f=50
6 Nf1=1440
7 pf=0.8
8 Ml=1.3*10^3
9 Ns=120*f/P
10 s=(Ns-Nf1)/Ns
11 fr=s*f
12 Pm=Po+Ml
13 Pc=Pm*s/(1-s)
14 Pcp=Pc/3//copper loss per phase
15 P2=Pc/s
16 S1=1.4*10^3
17 Pi=P2+S1
18 n=Po*100/Pi
19 I1=Pi/(sqrt(3)*V1*pf)
20 printf(" slip at full load\n")
21 printf(" s=%0.3 f \n",s)
22 printf(" rotor frequency\n")

```

```

23 printf(" fr=%0. f Hz\n",fr)
24 printf(" rotor copper loss per phase\n")
25 printf(" Pcp=%0.2 f W\n",Pcp)
26 printf(" total copper loss\n")
27 printf(" Pc=%0.2 f W\n",Pc)
28 printf(" efficiency at full load\n")
29 printf(" n=%0.2 f \n",n)
30 printf(" line current drawn\n")
31 printf(" I1=%0.3 f A\n",I1)

```

---

**Scilab code Exa 10.26** find power factor of rotor

```

1 //Chapter –10,Example10_26 , pg10_66
2 R2=0.04
3 X2=0.2
4 sf1=0.03
5 //at Tst , s=1
6 // Tfl=Tst
7 // (R21^2) –1.3633*R21+0.04=0
8 a=1
9 b=-1.3633
10 c=0.04
11 R21=(-b+sqrt((b^2)-4*a*c))/(2*a)
12 Rex=R21-R2
13 pf=R21/sqrt((R21^2)+(X2^2))
14 printf(" power factor of rotor\n")
15 printf(" pf=%0.3 f lagging",pf)

```

---

**Scilab code Exa 10.27** determine full load speed and speed at max torque

```

1 //Chapter –10,Example10_27 , pg10_67
2 P=4
3 f=50

```

```

4 Po=8*10^3
5 //Tst=1.5*Tfl and Tm=2*Tfl
6 //(R2^2)+((sfl*X2)^2)=1.5*sfl*((R2^2)+(X2^2))
   ..... (1)
7 //(R2^2)+((sfl*X2)^2)=2*(sfl/sm)*((R2^2)+((sm*X2)^2)
   ) ..... (2)
8 //dividing (1) and (2) by (X2^2) on both sides and
   R2/X2=sm
9 //(sm^2)+(sfl^2)=5*(1+(sm^2))*sfl ..... (3)
10 //(sm^2)+(sfl^2)=2*(2*(sm^2))*(sfl/sm)=4*sm*sfl
   ..... (4)
11 //dividing (3) by (4)
12 //(sm^2)-2.667*sm+1=0
13 a=1
14 b=-2.667
15 c=1
16 sm=(-b-sqrt((b^2)-4*a*c))/(2*a)
17 Ns=120*f/P
18 //substituting sm in (4)
19 //(sfl^2)-1.8052*sfl+0.2036=0
20 a=1
21 b=-1.8052
22 c=0.2036
23 sfl=(-b-sqrt((b^2)-4*a*c))/(2*a)
24 N=Ns*(1-sfl)
25 Nm=Ns*(1-sm)
26 printf("full load speed\n")
27 printf("N=%0.2f r.p.m\n",N)
28 printf("speed at max. torque\n")
29 printf("Nm=%0.2f r.p.m\n",Nm)

```

---

Scilab code Exa 10.28 calculate starting torque

```

1 //Chapter-10,Example10_28 ,pg10_68
2 Po=10*735.5//(in W)

```

```

3 Nf1=1410
4 P=4
5 f=50
6 Ns=120*f/P
7 sfl=(Ns-Nf1)/Ns
8 Nm=1200
9 sm=(Ns-Nm)/Ns
10 T=2*sfl*sm/((sm^2)+(sfl^2)) // Tfl/Tm
11 T1=(1+(sm^2))/(2*sm) //Tm/Tst
12 T2=T1*T // Tfl/Tst
13 Tfl=Po*60/(2*pi*Nf1)
14 Tst=Tfl/T2
15 printf(" starting torque\n")
16 printf(" Tst=%0.2 f Nm" ,Tst)

```

---

**Scilab code Exa 10.29** calculate speed torque and external resistance

```

1 //Chapter –10,Example10_29 , pg10_70
2 P=4
3 f=50
4 R2=0.025
5 X2=0.15
6 sfl=0.04
7 Tfl=150
8 sm=R2/X2
9 Tm=Tfl*((R2^2)+((sfl*X2)^2))*sm/(sfl*((R2^2)+((sm*X2
    )^2)))
10 Ns=120*f/P
11 N=Ns*(1-sm)
12 //at start
13 R21=X2
14 Rex=R21-R2
15 printf("maximum torque\n")
16 printf("Tm=%0.2 f Nm\n" ,Tm)
17 printf("speed N=%0.f r.p.m\n" ,N)

```

```
18 printf(" external resistance\n")
19 printf(" Rex=%0.3 f ohm/ph" ,Rex)
```

---

**Scilab code Exa 10.30** calculate motor output and efficiency

```
1 //Chapter –10,Example10_30 , pg10_70
2 Tsh=162.84
3 P=6
4 f=50
5 Tlost=20.36
6 fr=1.5
7 s=fr/f
8 Ns=120*f/P
9 N=Ns*(1-s)
10 Po=Tsh*(2*pi*N)/60
11 Fl=Tlost*(2*pi*N)/60
12 Pm=Po+Fl
13 Pc=Pm*s/(1-s)
14 P2=Pc/s
15 S1=830
16 Pi=P2+S1
17 n=Po*100/Pi
18 printf("motor output\n")
19 printf("Po=%0.4 f W\n" ,Po)
20 printf("copper loss in rotor\n")
21 printf("Pc=%0.3 f W\n" ,Pc)
22 printf("motor input\n")
23 printf("Pi=%0.3 f W\n" ,Pi)
24 printf("efficiency of motor\n")
25 printf("n=%0.2 f " ,n)
```

---

**Scilab code Exa 10.31** find ratio of torques



```

1 //Chapter -10,Example10_31 ,pg10_71
2 f=50
3 P=8
4 R2=0.01
5 X2=0.1
6 sfl=0.04
7 //for Tmax
8 sm=R2/X2
9 //for Tfl
10 s=sfl
11 T=sm*R2*((R2^2)+((sfl*X2)^2))/((sfl*R2)*((R2^2)+((sm
    *X2)^2)))/Tmax/Tfl
12 Ns=120*f/P
13 sm=0.1
14 N=Ns*(1-sm)
15 printf("ratio of max to full load torque\n")
16 printf("T=%0.2 f\n",T)
17 printf("speed at max torque\n")
18 printf("N=%0. f r.p.m",N)

```

---