

Scilab Textbook Companion for
Generation Distribution and Utilization Of
Electrical of Energy
by C. L. Wadhwa¹

Created by
Deeraj A
B.Tech, Second Year
Electronics Engineering
NIT, Karnataka
College Teacher
NA

Cross-Checked by
Prof. Singla

July 31, 2019

¹Funded 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: Generation Distribution and Utilization Of Electrical of Energy

Author: C. L. Wadhwa

Publisher: New Age International, New Delhi

Edition: 3

Year: 2012

ISBN: 978-81-224-2821-6

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
2 Economics Of Generation	5
3 Distribution	18
4 Electric Drives and Industrial Applications	40
5 Electric Heating And Welding	62
6 Illumination Engineering	73
7 Electric Traction	85
8 Electrolytic Process	113

List of Scilab Codes

Exa 2.1	To Determine the Demand and Supply Parameters for 15 bulbs	5
Exa 2.2	To determine the Demand and supply parameter of four consumers	6
Exa 2.3	To Determine the Yearly Cost of the substation	8
Exa 2.4	To determine the Load factor and suitable units for 24 hr operation of the plant	9
Exa 2.5	To determine the Plant use factor of each unit	11
Exa 2.6	To determine the Overall cost per kWhr . . .	13
Exa 2.7	To determine the amount saved to replace the equipment	14
Exa 2.8	To Obtain a two part tariff for the consumers	15
Exa 2.9	To determine the most economic power factor	17
Exa 3.1	To Determine the most economical current density	18
Exa 3.2	To determine the most economical cross sectional area	19
Exa 3.3	To determine the Most Economical Cross Sectional Area to supply a 3 Phase Load	19
Exa 3.4	To determine the most economical cross section for a 3 Phase line 8 km long	20
Exa 3.5	To Calculate Voltage between middle wire and outer at each load point	21
Exa 3.6	To obtain the voltages at the far end of the positive and negative wires	24
Exa 3.7	To determine the cross section of the conductor for a minimum consumer voltage	26

Exa 3.8	To determine the cross section of the conductor for a minimum consumer voltage	27
Exa 3.9	To Determine the location and magnitude of minimum voltage	30
Exa 3.10	To determine the currents supplied to the ring main from A and B	31
Exa 3.11	To Compare the volume of copper required .	33
Exa 3.12	To Determine the Voltage at the far end . .	35
Exa 3.13	To determine line currents and neutral currents	36
Exa 3.14	To determine the best location of the substation for a given set of loads	38
Exa 4.1	To Determine Motor Parameters of Series Motor rated at 220V	40
Exa 4.2	To Determine the resistance to be inserted for various cases	42
Exa 4.3	To determine the speed for which the torque is maximum	43
Exa 4.4	To Determine the slip at maximum torque and at full load	44
Exa 4.5	To Determine the Maximum Torque as ratio of nominal value of voltage	46
Exa 4.6	To Determine the starting torque and starting current for various starters	47
Exa 4.7	To Detemine Minimum Starting Current drawn from the supply	49
Exa 4.8	To Determine the value of resistor connected in series to run the fan at a particular speed	50
Exa 4.9	To Determine the time taken to start the motor	51
Exa 4.10	To time taken and the number of revolutions made before the motor is stopped	52
Exa 4.11	To Determine the 30 mins rating of motor .	53
Exa 4.12	To Determine the Continous Rating of the Motor for Suitable Conditions	54
Exa 4.13	To estimate the final temperature of the machine and its time constant	55

Exa 4.14	To Determine the Moment of inertia for a slip of 8 percent	56
Exa 4.15	To determine the Moment of inertia of the flywheel	57
Exa 4.16	To determine the speed and torque of a single phase full converter	58
Exa 4.17	To Determine the motor current and torque for a particular firing angle	59
Exa 4.18	To determine the pulse width for a particular average value of back EMF	60
Exa 4.19	To determine the range of speed control and duty cycle	61
Exa 5.1	To determine the size and length of the wire	62
Exa 5.2	To determine the various temperature by changing the connection of the resistance elements	64
Exa 5.3	To Determine the average KW input to the furnace	66
Exa 5.4	To determine the efficiency of a high frequency induction furnace	67
Exa 5.5	To Determine the equivalent resistance of the charge and current	68
Exa 5.6	To Detemine power absorbed and the power factor	69
Exa 5.7	To determine the Voltage Required and Current Drawn	70
Exa 5.8	To estimate the voltage and Current during heating	71
Exa 6.1	To determine the value of resistance to be connected in the motor circuit	73
Exa 6.2	To determine the total radiation sent vertically downward from a lamp of 1500 cp	74
Exa 6.3	To calculate Spherical Candle Power and intensity of illumination	75
Exa 6.4	To determine average illumination and illumination at various points on the area	76
Exa 6.6	To determine illumination at a point when a mirror is used	77
Exa 6.7	To determine the illumination over a disc	79

Exa 6.8	To estimate the number rating and disposition of the lamps	80
Exa 6.9	To determine the number of lamps required	81
Exa 6.10	To determine the illumination on the ground	82
Exa 6.11	Determine the output voltage	83
Exa 7.1	To determine the Acceleration and Coasting Period	85
Exa 7.2	To determine the duration of acceleration and others	86
Exa 7.3	To determine the acceleration required to run the service	88
Exa 7.4	To determine the time taken by the train to attain a particular speed	89
Exa 7.5	To determine the Factors affeting the mechanics of the Train	90
Exa 7.6	To determine the Specific Energy Output Runs for Both the Cases	91
Exa 7.7	To determine the specific energy consumption	94
Exa 7.8	To determine the weight of the locomotive .	96
Exa 7.9	To deduce the speed current characteristic of the motor	97
Exa 7.10	To determine the speed torque curve for the series motor	97
Exa 7.11	To determine the horse power delivered by the locomotive	98
Exa 7.12	To Determine the tractive effort shared by the two locomotives	99
Exa 7.13	To determine the traction control of 2 motors rated at 1500V	102
Exa 7.14	To determine the speed of motors when connected in series	104
Exa 7.15	To determine the characteristics gear ratio and wheels are replaced	105
Exa 7.16	To Determine the Resistance of the Various Steps in a Series Motor	106
Exa 7.17	To detemine the characteristics for tapping of 30 percent of the turns	108

Exa 7.18	To determine the power fed into the supply system	109
Exa 7.19	To determine the value of resistance to be connected in the motor circuit	110
Exa 8.1	To determine the quantity of current	113
Exa 8.2	To determine the Electrochemical Equivalent of Silver	114
Exa 8.3	To Determine the Annual output of copper .	114
Exa 8.4	To Calculate the Energy usedd in producing chemical action developed in an electrolytic cell	115
Exa 8.5	To determine the weights of Nickel and Silver Deposited	116

Chapter 2

Economics Of Generation

Scilab code Exa 2.1 To Determine the Demand and Supply Parameters for 15 bulbs

```
1
2 //To Determine the Demand and Supply Parameters for
   15 bulbs
3 //Page 73
4 clc;
5 clear;
6 W=60; //Wattage of the bulb
7 N=15; //No. of bulbs
8 CL=W*N; //Connected Load;
9 Wih=2*(10^3); //Wattage of immersion heater
10 Wh=2*(10^3); //Wattage of heater
11
12 //Usage of Bulbs at different time periods
13 N1=5;
14 N2=10;
15 N3=6
16
17 //Time periods for bulbs
18 T1=2; //6pm - 8pm
19 T2=2; //8pm - 10pm
20 T3=2; //10pm - 12pm
```

```

21 //Time Periods for heaters
22 T4=4; //1pm - 5pm
23 T5=3; //8pm - 11pm
24
25 //CASE 1
26 MD1=W*N2; //Maximum Demand
27 DF=MD1*100/CL; //Demand Factor
28 EC1=(N1*W*T1)+(N2*W*T2)+(N3*W*T3); //Energy Consumed
29 DLF1=EC1*100/(24*MD1); //Daily Load Factor
30
31 //CASE 2
32 MD2=(W*N2)+Wh; //From 8pm - 10pm
33 EC2=(T4*Wh)+(T5*Wh)+EC1; //Energy Consumed
34 DLF2=EC2*100/(24*MD2); //Daily Load Factor
35
36 printf('i)a) Connected Load is %g W\nb) The Maximum
    Demand is %g W\nc) The Demand Factor is %g
    percent\nd) The Daily Load Factor is %g percent\n
    ',CL,MD1,DF,DLF1)
37 printf('ii) The Improved Daily Load Factor is %g
    percent\n',DLF2)

```

Scilab code Exa 2.2 To determine the Demand and supply parameter of four consumers

```

1
2 //To determine the Demand and supply parameter of
    four consumers
3 //Page 74
4 clc;
5 clear;
6
7 //Maximum Demands of various users
8 MD1=2*(10^3); //9pm
9 MD2=2*(10^3); //12 noon
10 MD3=8*(10^3); //5pm

```

```

11 MD4=4*(10^3); //8pm
12 MDT=MD1+MD2+MD3+MD4; //Sum of all Maximum Demands
13
14 //Demands of various users
15 D1=1.6*(10^3); //8pm
16 D2=1*(10^3); //8pm
17 D3=5*(10^3); //8pm
18
19 //The Number after the Alphabets represents the
    Consumer
20
21 //Maximum Demand of the System arises at 8.00 PM
22 MDS = D1+D2+D3+MD4;
23
24 TDF=MDT/MDS; //Diversity Factor
25 //Given Values
26 //Average Loads
27 AL2=500;
28 AL4=1000;
29 //Load Factors
30 LF1=15/100;
31 LF3=25/100;
32 //Calculated Values
33 //Average Loads
34 AL1=LF1*MD1;
35 AL3=LF3*MD3;
36 //Load Factors
37 LF2=AL2*100/MD2;
38 LF4=AL4*100/MD4;
39
40 ALS=AL1+AL2+AL3+AL4; //Combined Average Loads
41 LFS=ALS*100/MDS; //Combined Load Factor
42
43 printf('i) The Diversity Factor is %g\n',TDF)
44 printf('ii) The Average load and Load factor of:\n')
45 printf(' Consumer 1 : %g W and %g percent\n',AL1,LF1
    *100)
46 printf(' Consumer 2 : %g W and %g percent\n',AL2,LF2

```

```

)
47 printf(' Consumer 3 : %g W and %g percent\n',AL3,LF3
    *100)
48 printf(' Consumer 4 : %g W and %g percent\n',AL4,LF4
    )
49 printf('iii) The Combined Load Factor and The
    Combined Average Load is %g percent and %g W
    respectively\n',LFS,ALS )

```

Scilab code Exa 2.3 To Determine the Yearly Cost of the substation

```

1
2 //To Determine the Yearly Cost of the substation
3 //Page 75
4 clc;
5 clear;
6 Teff=95/100; //Transmission Efficiency
7 Deff=85/100; //Distribution Efficiency
8 DFT=1.2; //Diversity Factor For Transmission
9 DFD=1.3; //Diversity Factor For Distribution
10 MDGS=100*(10^6); //Maximum Demand of Generating
    Station
11 ALF=40/100; //Annual Load Factor
12 ACCT=2.5*(10^6); //Annual Capital Charge for
    Transmission
13 ACCD=2*(10^6); //Annual Capital Charge for
    Distribution
14 GCC=100; //Generating Cost per KW demand
15 GCCU=5/100; // Per Unit Cost
16 //Fixed Charges from Supply to Substation Annually
17 GFC=GCC*MDGS/1000; //Generating
18 TFC=ACCT; //Transmission
19 TotFCS=GFC+TFC //Total
20 //Fixed Charges for supply upto Consumer Annually
21 DFC=ACCD; //Distribution

```

```

22 TotFCC=TotFCS+DFC; //Total
23
24 AMDS= DFT*MDGS/1000; //Aggregate of Maximum Demand
    at Supply
25 AMDC= DFD*AMDS; //Aggregate of Maximum Demand for
    Consumers
26
27 FCS=TotFCS/AMDS; //Fixed Charges Per KW at
    substation
28 CES=GCCU/Teff; //Cost of energy at the substation
29
30 FCC=TotFCC/AMDC; //Fixed Charges per KW at the
    consumer premises
31 CEC=CES/Defeff; //Cost of Energy at the consumer
    premises
32
33 printf('The Yealy Cost per KW demand and the cost
    per KWhr at:\n')
34 printf('a) The substation is %g rupees per KW and %g
    paise per KWhr\n',FCS,CES*100)
35 printf('b) The consumer premises is %g rupees per KW
    and %g paise per KWhr\n',FCC,CEC*100)

```

Scilab code Exa 2.4 To determine the Load factor and suitable units for 24 hr oper

```

1
2 //To determine the Load factor and suitable units
    for 24 hr operation of the plant
3 //Page 78
4 clc;
5 clear;
6
7 //Demands at Various Time Periods starting from 12PM
    to 12PM
8 D1=500*(10^3);

```

```

9 D2=800*(10^3);
10 D3=2000*(10^3);
11 D4=1000*(10^3);
12 D5=2500*(10^3);
13 D6=2000*(10^3);
14 D7=1500*(10^3);
15 D8=1000*(10^3);
16
17 MD=D5; //Maximum Demand
18 //Time Periods of demands ststing from 12PM
19 T1=5;
20 T2=5;
21 T3=2;
22 T4=2;
23 T5=3;
24 T6=3;
25 T7=2;
26 T8=2;
27
28 //Total Energy Demand in 24hrs
29 TED=(T1*D1)+(T2*D2)+(T3*D3)+(D4*T4)+(T5*D5)+(D6*T6)
      +(D7*T7)+(T8*D8);
30
31 LF=TED*100/(24*MD);
32
33 printf('Since Maximum Demand is 2500 kW, 2 units of
      1000W and one unit of 500W is required. Also for
      continuity of supply, A reserve of 1000W unit is
      required.\n')
34
35 C1000=3*1000*(10^3); //1000 unit
36 C500=1*500*(10^3); //500 Unit
37
38 TCP=C1000+C500; //Total capacity of the plant
39 PCF=TED*100/(24*TCP); //Plant Capacity Factor
40
41 //Operating Schedule, Units operated can be seen in
      the textbook

```

```

42 G1=500*(10^3);
43 G2=1000*(10^3);
44 G3=2000*(10^3);
45 G4=1000*(10^3);
46 G5=2500*(10^3);
47 G6=2000*(10^3);
48 G7=1500*(10^3);
49 G8=1000*(10^3);
50
51 TEG=(T1*G1)+(T2*G2)+(T3*G3)+(G4*T4)+(T5*G5)+(G6*T6)
    +(G7*T7)+(T8*G8); //Total Energy Generated
52 PUF=TED*100/(TEG); //Plant Use Factor
53
54 printf('a) The Reserve Capacity is a 1000kW Unit and
    Load Factor is %g percent\n',LF)
55 printf('b) The Plant Capacity Factor is %g percent\n
    ',PCF)
56 printf('c) The Plant Use Factor is %g percent\n',PUF
    )

```

Scilab code Exa 2.5 To determine the Plant use factore of each unit

```

1
2 //To determine the Plant use factore of each unit
3 //Page 80
4 clc;
5 clear;
6
7 MDS=25*(10^6); //Maximum Demand on the System
8 U1=15*(10^6); //Load Supplied By Unit 1
9 U2=12.5*(10^6); //Load Supplied By Unit 2
10 //Running Time Factor of the Unit
11 T1=1;
12 T2=40/100;
13

```

```

14 //Energy generated by each unit
15 E1=1*(10^8);
16 E2=1*(10^7);
17 Et=E1+E2; //Total Energy
18
19 //Maximum Demands on Each Units
20 MD1=U1;
21 MD2=MDS-U1;
22
23 //Annual Load Factor for the Units
24 ALF1=E1*1000*100/(MD1*8760);
25 ALF2=E2*1000*100/(MD2*8760);
26
27 LFP=E2*1000*100/(MD2*0.4*8760); //Load Factor for
    the it is loaded
28
29 //Since Unit runs all through the year without any
    reserve, Load Factor, Plant Use Factor and Plant
    Capacity Factor are the same
30 PUF1=ALF1; //Plant Use Factor
31 PCF1=ALF1; // Plant Capacity Factor
32
33 PCF2=E2*1000*100/(U2*8760); //Plant Capacity Factor
    for Unit 2
34 PUF2=E2*1000*100/(U2*0.4*8760); //Plant Use Factor
    for Unit 2
35
36 LFP=Et*100*1000/(MDS*8760); //Annual Load Factor of
    the Complete Plant
37
38 printf('The Load Factor, Plant Capacity Factor,
    Plant Use Factor of:\n')
39 printf('Unit 1 : %g percent, %g percent, %g percent\n'
    ,ALF1,PCF1,PUF1)
40 printf('Unit 2 : %g percent, %g percent, %g percent\n'
    ,ALF2,PCF2,PUF2)
41 printf('The Annual Load Factor of the Entire Plant
    is %g percent',LFP)

```

Scilab code Exa 2.6 To determine the Overall cost per kWhr

```
1
2 //To determine the Overall cost per kWhr
3 //Page 82
4 clc;
5 clear;
6
7 // C1 =(100,000 rupees + 100 rupees/kW + 6 paise /
   kWhr) //Base Load Station
8 // C2 =(80,000 rupees + 60 rupees/kW + 8 paise /kWhr
   ) //Peak Load Station
9
10 MaxD=15*(10^6);
11 MinD=5*(10^6);
12
13 deff ('a=BLS(b,c)', 'a=100000+(b*100)+((6/100)*c)');
   //Function to Find Annual Cost of Base Load
   Station.
14
15
16 deff ('a=PLS(b,c)', 'a=80000+(b*60)+((8/100)*c)'); //
   Function to Find Annual Cost of Peak Load Station
   .
17
18 //Also from the Annual Cost characteristics of the
   base load and peak load station we find co-
   efficient which are helpful in computing the
   number of hours a peak load station must work for
   minimum cost
19
20 a1=100;
21 a2=60;
22 b1=6/100;
```

```

23 b2=8/100;
24
25 Tpeak=(a1-a2)/(b2-b1); //Number of hours the peak
    plant should operate
26 //From the straight line annual load duration curve,
    Maximum Demand at Peak Load Station can be
    calculated
27 MDP=Tpeak*(MaxD-MinD)/8760;
28
29 TotEG=(MaxD+MinD)*8760/(2*1000); //Total Energy
    Generated
30 EGP=MDP*Tpeak/(2*1000); //Energy Generated by Peak
    Load
31 EGB=TotEG-EGP; //Energy Generated by Base Load
32
33 MDB=MaxD-MDP; //Maximum Demand at the Base Load
34
35 //Total Annual Cost of Base Load and Peak Load
    Stations Respectively
36 C1=BLS((MDB/1000),EGB);
37 C2=PLS((MDP/1000),EGP);
38
39 TotC=C1+C2; //Total Cost of both the plants
40 CE=TotC*100/TotEG; //Cost of energy in paise per
    kWhr
41
42 printf('The Operating Scedule of Peak Load Station
    for Minimum Annual Cost is %g hours\n',Tpeak)
43 printf(' The Overall Cost per kWhr is %g paise \n',
    CE)

```

Scilab code Exa 2.7 To determine the amount saved to replace the equipment

```

1
2 //To determine the amount saved to replace the

```

```

        equipment
3 //Page 90
4 clc;
5 clear;
6
7 Pc=80000; //Plant Cost
8 UL= 15; //Useful Life of the Plant
9 SVE=5000; //Salvage Value of the Equipment
10 r=5/100; //Compound Interest Rate
11
12 A1=(Pc-SVE)/UL; //Annual Amount to be saved using
        straight line method
13 A2=(Pc-SVE)*r*100/(100*((1+r)^UL)-1)// Annual
        Amount to be saaved using Sink Fund Method
14
15 printf('i) The Amount to be Saved Annually according
        to straight line method is %g Rupees\n',A1)
16 printf('ii) The Amount to be Saved Annually
        according to sink fund method is %g Rupees\n',A2)

```

Scilab code Exa 2.8 To Obatin a two part tariff for the consumers

```

1
2 //To Obatin a two part tariff for the consumers
3 //Page 90
4 clc;
5 clear;
6
7 EG=390*(10^6); //Energy Generated in kWhr
8 MD=130*(10^6); //Maximum Demand of the Supply
9 SCeff=90/100; //The Amount of energy transferred
        from Substation to Consumer
10 //Total Cost for Each Division
11 Fuel=5*(10^6);
12 Gen=2.4*(10^6);

```

```

13 Trans=5*(10^6);
14 Dist=3.4*(10^6);
15 Tot=Fuel+Gen+Trans+Dist; //Total Cost
16
17 deff('x=Runcost(y,z)', 'x=(y*z/100)') //Function to
    Find out the Running Costs
18 //Running Costs
19 Fuelr=Runcost(90,Fuel);
20 Genr=Runcost(10,Gen);
21 Transr=Runcost(5,Trans);
22 Distr=Runcost(7,Dist);
23 Totr=Fuelr+Genr+Transr+Distr; //Total Cost
24
25 FixCost=Tot-Totr;// Fixed Cost
26 FixChar=FixCost*1000/MD; //Fixed Charges per KW
27 EnChar=Totr*100/(EG*SCeff); //Energy Charges in
    Paise for Consumer
28 OverCost=Tot*100/(EG*SCeff); //Overall Energy
    Charges
29
30 LF=40/100; //Load Factor Raised to 40%
31 EG1=LF*MD*8760/1000; //Energy Generated for
    Different Load Factor
32 Totr1=Totr*(EG1/EG); //Cost of Energy Generated
33 Tot1=FixCost+Totr1; //Total Cost for the New Load
    Factor
34 OverCost1=Tot1*100/(EG1*SCeff); //Overall Energy
    Charges
35
36 Saving=(OverCost-OverCost1)*100/OverCost; //
    Percentage Saving in the Overall Cost per kWhr
37
38 printf('The Fixed Charges is %g rupees per kW\n',
    FixChar)
39 printf(' The Energy Charges for the Consumer is %g
    paise per kWhr\n',EnChar)
40 printf(' IF the Load Factor is raised to 40percent
    of the Same Maximum Demand, then the percentage

```

saving in the overall costs is %g percent\n',
Saving)

Scilab code Exa 2.9 To determine the most economic power factor

```
1
2 //To determine the most economic power factor
3 //Page 91
4 clc;
5 clear;
6
7 P=200*(10^3); //Maximum Demand
8 pf=0.707; //Power Factor Lagging
9
10 a=100; //Tariff per kVA per year
11
12 b=200; //Power factor improvement cost Per kVA.
13 r=20; //Interest Depreciation , maintenance and cost
    of losses amount to 20% of capital cost per year
14
15 // Economic PF = sqrt(1-((b1/a)^2))
16
17 b1=r*b/100; // b' term accrding to the equation above
18
19 pfeco=sqrt(1-((b1/a)^2)); //Economic Power Factor
20
21 printf('The Economic Power Factor is %g \n',pfeco)
```

Chapter 3

Distribution

Scilab code Exa 3.1 To Determine the most economical current density

```
1
2 //To Determine the most economical current density
3 // Page 103
4 clc;
5 clear;
6
7 OFC=15; //Cost of single phase overhead feeder per
          km per unit area
8 AIDC= 10*OFC/100; //Annual Interest and depreciation
          charges
9 R=1/58; //Resistance of 1m length and 1 sq.mm
10 CLPUL= 2*R*1000; // Copper Loss per unit length per
          unit area per unit square current
11 ACL= CLPUL*365*24*0.5; // Annual Copper Loss
12 GC= 5/100; // Generating Cost per unit.
13 ACEL= ACL*GC/1000; // Annual cost of energy loss
14 CD= sqrt(AIDC/ACEL); //Current Density
15 printf('The Most Economical Current Density for this
          Case is %g A/sq.mm',CD)
```

Scilab code Exa 3.2 To determine the most economical cross sectional area

```
1
2 //To determine the most economical cross sectional
   area
3 //Page 104
4 clc;
5 clear;
6
7 OFC=180; //Cost of single phase overhead feeder per
   km per unit area + 1200
8 AIDC= 10*OFC/100; //Annual Interest and depreciation
   charges + 120
9 R=1/58; //Resistance of 1m length and 1 sq.mm
10 I=200; // Maximum Current Flowing
11 CLPUL= 2*(I^2)*R*1000; // Copper Loss per unit
   length per unit area
12 ACL= CLPUL*365*24*(8/12); // Annual Copper Loss
13 GC= 5/100; // Generating Cost per unit.
14 ACEL= ACL*GC/1000; // Annual cost of energy loss
15 A= sqrt(ACEL/AIDC); //Cross Sectional Area of the
   cable
16 printf('The Most Economical cross sectional area of
   the cable for this Case is %g sq.mm',A)
```

Scilab code Exa 3.3 To determine the Most Economical Cross Sectional Area to supply

```
1
2 // To determine the Most Economical Cross Sectional
   Area to supply a 3 Phase Load
3 //Page 105
4 clc;
```

```

5 clear;
6 deff('a=LLF(b)', 'a=(0.25*b)+(0.75*(b^2))'); //
    Function to determine the Loss Load Factor
7
8 OFC=0.20; //Cost of single phase overhead feeder per
    m per unit area + 10
9 AIDC= 10*OFC/100; //Annual Interest and depreciation
    charges + 1
10 TE= 2.5*(10^6); // Total energy to be supplied per
    annum
11 CEW=10/100; // Cost of energy wasted per unit
12 LFS= TE/(1000*365*24); // Load factor of supply
13 Llf=LLF(LFS); // Load Loss factor
14 R=1/58; // Resistance of the cable per unit length
15 PF=1; // Unity power factor
16 MD= 1*(10^6); // Maximum Demand
17 V=11*(10^3); // Voltage of the feeder
18 I=MD/(sqrt(3)*V*PF); // Full Load Current
19 FLCL= 3*(I^2)*R; // Full Load Copper Loss per Metre
20 ACL= Llf*FLCL; // Actual Copper Loss
21 CCL=ACL*(365*24*CEW/1000); // Cost of Copper Loss
22 A=sqrt(CCL/AIDC);
23 printf('The Most Economical Cross sectional area for
    this Case is %g A/sq.mm',A)
24 // Calculation Mistake in the Book. Hence according
    to the concepts in the book the answer is as
    calculated. Please Note.

```

Scilab code Exa 3.4 To determine the most economical cross section for a 3 Phase 1

```

1
2 //To determine the most economical cross section for
    a 3 Phase line 8 km long
3 //Page 106
4 clc;

```

```

5 clear;
6
7 OFC=5000; //Cost of feeder per km per unit area +
  6250
8 AIDC= 8*OFC/100; //Annual Interest and deprecation
  charges + 625
9 AIDCPD= AIDC/365; // Annual Interest and
  deprecation charges per day
10 R=1/58; //Resistance of 1m length and 1 sq.mm
11 CEU= 5/100; // Cost of energy per unit
12 V=33*(10^3); // Voltage of the feeder
13 L10=3000*(10^3); // Load for 10 hrs at unity power
  factor.
14 L8=1000*(10^3); // Load for 8 hrs at unity power
  factor.
15 L6=2000*(10^3); // Load for 6 hrs at unity power
  factor.
16
17 def('a=LC(b)', 'a=b/(sqrt(3)*V)') // Funtion to
  determine the Load Current
18 I10=LC(L10); // Load Current for 10 hrs at unity
  power factor.
19 I8=LC(L8); // Load Current for 8 hrs at unity power
  factor.
20 I6=LC(L6); // Load Current for 6 hrs at unity power
  factor.
21 ELPD=3*(((I10^2)*10)+((I8^2)*8)+((I6^2)*6))
  *1000/(100*58); // Energy Loss Over the DAY per
  km (Area in sq.cm)
22 CEL=ELPD*CEU/1000; // Cost of energy loass per km
23 A=sqrt(CEL/AIDCPD); // Cross Sectional Area
24 printf('The Most Economical Cross Sectional Area for
  a 3 Phase line 8 km Long is %g sq..cm',A)

```

Scilab code Exa 3.5 To Calculate Voltage between middle wire and outer at each loa

```

1
2 // To Calculate Voltage between middle wire and
   outer at each load point
3 // Page 109
4 clc;
5 clear;
6
7 Vs=220; //Supply Voltage
8 //Resistances of the respective Sections
9 Rab=0.2;
10 Rbc=0.2;
11 Rdh=0.2;
12 Rhe=0.2;
13 Ref=0.3;
14 Rfl=0.1;
15 Rlg=0.3;
16 Rij=0.25;
17 Rjk=0.2;
18 Ra=0.2;
19 Rd=0.4;
20 Ri=0.3;
21
22 //Currents following through the respective Section ,
   Found using KCL
23 Iad=5;
24 Ibe=10;
25 Icl=12;
26 Ihi=15;
27 Ifj=5;
28 Igk=15;
29 Ia=Iad+Ibe+Icl; // Current through the positive wire
30 Iab=Ia-Iad;
31 Ibc=Ia-Iad-Ibe;
32 Ii=Ihi+Ifj+Igk; // Current through the negative wire
33 Id=Ii-Ia; //Current through the Middle wire
34 Idh=Iad+Id;
35 Ihe=Ihi-Idh;
36 Ief=Ibe-Ihe;

```

```

37 Ifl=Ief-I fj;
38 Ilg=I gk;
39 Iij=I i-I hi;
40 Ijk=I i-I hi-I fj;
41
42 //Voltage drops across each section
43 Va= Ra*I a;
44 Vab=I ab*R ab;
45 Vbc=I bc*R bc;
46 Vi=I i*R i;
47 Vd=I d*R d;
48 Vdh=I dh*R dh;
49 Vhe=I he*R he;
50 Vef=I ef*R ef;
51 Vf1=I f1*R f1;
52 Vlg=I lg*R lg;
53 Vij=I ij*R ij;
54 Vjk=I jk*R jk;
55
56 //Voltage across the middle wire and the outer load
    points Using KVL
57 Vad=Vs-V a+V d;
58 Vbe=V ad-V ab-V he+V dh;
59 Vcl=V be-V bc+V f1+V ef;
60 Vhi=Vs-V d-V dh-V i;
61 Vfj=V hi-V ef+V he-V ij;
62 Vgk=V fj-V f1-V lg-V jk;
63
64 printf('The Voltages between middle wire and outer
    wire at each load point are:\n')
65 printf(' 1. Vad = %g V\n',Vad)
66 printf(' 2. Vbe = %g V\n',Vbe)
67 printf(' 3. Vcl = %g V\n',Vcl)
68 printf(' 4. Vhi = %g V\n',Vhi)
69 printf(' 5. Vfj = %g V\n',Vfj)
70 printf(' 6. Vgk = %g V\n',Vgk)

```

Scilab code Exa 3.6 To obtain the voltages at the far end of the positive and negative

```
1 //To obtain the voltages at the far end of the
   positive and negative wires
2 //Page 110
3 clc;
4 clear;
5
6 Vs=220; // Supply Voltage
7 //Resistances of the given sections
8 r1=0.015;
9 r2=0.035;
10 r3=0.02;
11 r4=0.01;
12 r5=0.025;
13 r6=0.015;
14 r7=0.03;
15 r8=0.01;
16 r9=0.02;
17 r10=0.03;
18 r11=0.04;
19
20 //Currents following in between the two wires
21 I1=10;
22 I2=20;
23 I3=25;
24 I4=5;
25 I5=15;
26 I6=15;
27 I7=15;
28 I8=15;
29 I9=18;
30 I10=30;
31 I11=15;
```

```

32
33 // Effective resistances for the above currents
34 R1=r1;
35 R2=r1+r2;
36 R3=r1+r2+r3;
37 R4=r1+r2+r3+r4;
38 R5=r1+r2+r3+r4+r5;
39 R6=r1+r2+r3+r4+r5+r6;
40 R7=r7;
41 R8=r7+r8;
42 R9=r7+r8+r9;
43 R10=r7+r8+r9+r10;
44 R11=r7+r8+r9+r10+r11;
45
46 // Voltage drop in the respective wires
47 // Outer positive wire
48 Vop=(I1*R1)+(I2*R2)+(I3*R3)+(I4*R4)+(I5*R5)+(I6*R6);
49 // Outer Negative wire
50 Vnp=(I7*R7)+(I8*R8)+(I9*R9)+(I10*R10)+(I11*R11);
51 // Net drop in neutral wire
52 Vn=Vop-Vnp;
53
54 // Effective drop in the respective wires
55 // Outer positive wire
56 Veop=Vop+Vn;
57 // Outer negative wire
58 Venp=Vnp-Vn;
59
60 // Voltage at far end of the wires
61 // Positive Wire
62 Vpf=Vs-Veop;
63 // Negative Wire
64 Vnf=Vs-Venp;
65
66 printf("The voltage drop at the far end of the wires
        are:\n")
67 printf(" Positive Wire: %g V\n",Vpf)
68 printf(" Negative Wire: %g V\n",Vnf)

```

Scilab code Exa 3.7 To determine the cross section of the conductor for a minimum

```
1
2 //To determine the cross section of the conductor
   for a minimum consumer voltage
3 //Page 111
4 clc;
5 clear;
6
7 //Unknown Variable obtained in the equation
8 Ib=poly(0,"Ib");
9 r=poly(0,"r");
10
11 //Voltages at the respective ends
12 Va=230;
13 Vb=230;
14
15 //Minimum Consumers's Voltage
16 Vc=220;
17
18 //Lengths of the segments
19 r1=200;
20 r2=200;
21 r3=100;
22 r4=300;
23 r5=300;
24 r6=100;
25
26 //Effective length of segments
27 R1=r1;
28 R2=r1+r2;
29 R3=r1+r2+r3;
30 R4=r1+r2+r3+r4;
31 R5=r1+r2+r3+r4+r5;
```

```

32 R6=r1+r2+r3+r4+r5+r6;
33
34 //Current drawn by different loads
35 I1=25;
36 I2=20;
37 I3=25;
38 I4=25;
39 I5=10;
40
41 //By the law of momemts of currents
42 Vnet=2*((I1*R1)+(I2*R2)+(I3*R3)+(I4*R4)+(I5*R5)-(Ib*
    R6)); //Since the equation is equated to zero, r
    vanishes
43 X=Vnet-(Va-Vb); //Polynomial Equation to Find Ib
44 Ib=roots(X); // Numerical Value of Ib
45 Ia=(I1+I2+I3+I4+I5)-Ib;
46
47 //From the given figure it is clear that point M is
    the point of Minimum Potential
48 Vd=Va-Vc;
49 X=(2*r*((Ia*r1)+((Ia-I1)*r2)+((Ia-I1-I2)*r3)))-Vd;
    //Polynomial Equation to Find r
50 r=roots(X); // Resistance per unit length (Numerical
    Value)
51
52 Rstd=(1/58); // Resistance for 1m and 1 sq.mm
53
54 A=Rstd/r; // Cross Section of Conductor required.
55
56 printf('The Cross Section of the Conductor to
    provide minimal consumer''s voltage is %g sq.mm\n
    ',A)

```

Scilab code Exa 3.8 To determine the cross section of the conductor for a minimum

```

1
2 //To determine the cross section of the conductor
   for a minimum consumer voltage
3 //Page 111
4 clc;
5 clear;
6
7 //Unknown Variable obtained in the equation
8 x=poly(0,"x");
9 r=poly(0,"r");
10
11 //Voltages at the respective ends
12 Va=235;
13 Vb=230;
14
15 //Minimum Consumers's Voltage
16 Vc=220;
17
18 //Lengths of the segments
19 r1=200;
20 r2=200;
21 r3=100;
22 r4=300;
23 r5=300;
24 r6=100;
25
26 //Effective length of segments
27 R1=r1;
28 R2=r1+r2;
29 R3=r1+r2+r3;
30 R4=r1+r2+r3+r4;
31 R5=r1+r2+r3+r4+r5;
32 R6=r1+r2+r3+r4+r5+r6;
33
34 //Current drawn by different loads
35 I1=25;
36 I2=20;
37 I3=25;

```

```

38 I4=25;
39 I5=10;
40
41 //The Minimum Point assumed is N, Hence the current
    following to Point N is given by 'x'
42
43 //The Effective Drops
44 Van=Va-Vc;
45 Vbn=Vb-Vc;
46 V=Van/Vbn;
47 A=2*(((x+I1)*r1)+(r2*x)); //Wrt to Van
48 B=2*(((I2-x)*r3)+((I2+I3-x)*r4)+((I2+I3+I4-x)*r5)+((
    I2+I3+I4+I5-x)*r6)); //Wrt to Vbn
49 C=A/B; // The 'r' term gets eliminated
50 X=C-V; // Polynomial Equation to find x
51 X=X(2); // We take only the numerator into
    consideration as the above equation is equated
    to zero
52 x=roots(X);
53
54 Ia=I1+x; //Current Supplied at end A;
55 //It is clear the above assumed Minimal Point is
    wrong and it has to Point M
56 //Therefore finding drop at Point M
57
58 X=(2*r*(((x+I1)*r1)+(r2*x)+((x-20)*r3)))-Van; //
    Polynomial Equation to find r
59 r=roots(X); // Numerical Value of r
60
61 Rstd=(1/58); // Resistance for 1m and 1 sq.mm
62
63 A=Rstd/r; // Cross Section of Conductor required.
64
65 printf('The Cross Section of the Conductor to
    provide minimal consumer's voltage is %g sq.mm\n
    ',A)
66
67 // Please note the calculation mistake in the book.

```

The value of r found out is wrong in the text book.

Scilab code Exa 3.9 To Determine the location and magnitude of minimum voltage

```
1
2 //To Determine the location and magnitude of minimum
   voltage
3 //Page 114
4 clc;
5 clear;
6
7 Vs=220; //Supply Voltage at End A and B
8 //Different Conductor Lengths
9 //From End A
10 L1=100;
11 L2=50;
12 L3=50;
13 L4=400; //Length of uniform loading
14
15 A=0.5; //Uniforming loading spread over 400m
16 r=0.05; // Resistance of Conductor per Km
17 //Different Currents drawn by various loads
18 I1=50;
19 I2=75;
20 I3=A*L4;
21
22 //Taking moments of all currents at A;
23 Ib=((I1*L1)+((L1+L2)*I2)+((L1+L2+L3+(L4/2))*I3))/(L1
   +L2+L3+L4);
24 Ia=I1+I2+I3-Ib;
25
26 //Minimum Potential Point in this case is the point
   where All the current from B is drawn
27 X=Ib/A; // Distance from B;
```

```

28 Y=(L1+L2+L3+L4)-X; //Distance from A;
29
30 //Minimum Potential Drop
31 Vmind=(2*r/1000)*((Ia*L1)+((Ia-I1)*L2)+((Ia-I1-I2)*
    L3)+(((Y-L1-L2-L3)*A)*((Y-L1-L2-L3)/2)));
32
33 Vmin=Vs-Vmind; //Minimum Potential
34 printf('The Location of The Minimum Voltage is %g m
    from side A and its magnitude is %g V\n',Y,Vmin)

```

Scilab code Exa 3.10 To determine the currents supplied to the ring main from A and B

```

1
2 //To determine the currents supplied to the ring
    main from A and B
3 //Page 115
4 clc;
5 clear;
6
7 //Currents in the ring scheme going clockwise
8 I1=50;
9 I2=15;
10 I3=25;
11 I4=10;
12 It = I1+I2+I3+I4; // Total Current
13 //Resistances of the respective segments going
    clockwise
14 R1=0.1;
15 R2=0.12;
16 R3=0.04;
17 R4=0.08;
18 R5=0.06;
19 R6=0.02;
20
21 Ra=0.04; //Resistance at A;

```

```

22 Rb=0.06; //Resistance at B;
23
24 // Va=Vb; Net Voltage is Zero
25 //Dividing the total current from A as x and y
26 //Taking voltage drops in clockwise direction and
    anticlockwise directions
27 // We get two simultaneous equations
28 //  $3.2x + y = 120$ 
29 //  $x + 3y = 114$ 
30 R=[3.2,1;1,3];
31 V=[120;114];
32 I=inv(R)*V;
33
34 //To calculate and seperate the respective currents
    from the above matrix equation
35 x=I(1);
36 y=I(2);
37 Ia=x+y;
38 Ib=It-Ia;
39
40 printf('a) When Va = Vb; Ia = %g A and Ib =%g\n',Ia,
    Ib)
41
42 // Va+5=Vb; Net Voltage is 5V
43 //Dividing the total current from A as x and y
44 //Taking voltage drops in clockwise direction and
    anticlockwise directions
45 // We get two simultaneous equations
46 //  $3.2x + y = 120 - (5*NetVoltage) = 70$ 
47 //  $x + 3y = 114 - (5*NetVoltage) = 64$ 
48 R=[3.2,1;1,3];
49 V=[70;64];
50 I=inv(R)*V;
51
52 //To calculate and seperate the respective currents
    from the above matrix equation
53 x=I(1);
54 y=I(2);

```

```

55 Ia=x+y;
56 Ib=It-Ia;
57
58 printf('b) When  $V_a + 5 = V_b$ ;  $I_a =$  %g A and  $I_b =$ %g\n',
        ,Ia,Ib)
59
60 //Highly Accurate Answers, Text Book answers are
    rounded off.

```

Scilab code Exa 3.11 To Compare the volume of copper required

```

1
2 //To Compare the volume of copper required
3 //Page 116
4 clc;
5 clear;
6 //Unknown Resistances
7 r=poly(0,'r');
8 r1=poly(0,'r1');
9 //Lengths of the segments of the ring scheme
10 L1=100;
11 L2=200;
12 L3=200;
13 L4=150;
14 L5=150;
15 //Currents taken by respective loads
16 I1=40;
17 I2=20;
18 I3=100;
19 I4=40;
20 It=I1+I2+I3+I4;//Total Current
21
22 //Without Interconnector
23 //Let x be the current flowing through the entire
    ring

```

```

24 x=poly(0, 'x');
25 Eq=(L1*x)+(L2*(x-I1))+(L3*(x-I1-I2))+(L4*(x-I1-I2-I3
    ))+(L4*(x-I1-I2-I3-I4)); //Polynomial Equation to
    find x
26
27 x=roots(Eq);
28 x1=It-x; //Current flowing in the other direction
29
30 Vac1=((x1*L5)+((x1-I4)*L4))*r; // Voltage across AC
    without the connector
31 MVac1=((x1*L5)+((x1-I4)*L4)); // Magnitude of Vac1;
32
33 //With Interconnector
34 //Considering x amount of current to flow clockwise
    through segment AE
35 //Considering y amount of current to flow
    anticlockwise through segment AB
36 //Considering 200-(x+y) amount if current to flow
    through the segment AC
37 // Mesh Analysis of ABCDE gives  $5x - 3y = 140$ ;
38 // Mesh Analysis of ABC gives  $5x + 11y = 1120$ ;
39 R=[5, -3;5, 11];
40 V=[140;1120];
41 I=inv(R)*V;
42 x=I(1);
43 y=I(2);
44 Vac2=(It-(x+y))*250*r1; // Voltage across AC with
    connector
45 MVac2=(It-(x+y))*250; // Magnitude of Vac2;
46 printf('The Voltage drop across AC in both case is
    the same\n')
47 disp(Vac2, 'Is Equal to', Vac1)
48 printf('\n \n')
49 //To Compute the Numerical Values of the Ratio of
    resistances
50 RatioA = MVac1/MVac2;
51
52 disp(RatioA, 'is ', r1, 'divided by', r)

```

```

53
54 //Effective Length of both the cases
55 Leff=L1+L2+L3+L4+L5;
56 LeffC=Leff+250;
57 //Volume is Length * Area
58 RatioV=Leff*RatioA/LeffC;
59
60 printf('\nThe Volume of copper without the connector
      is %g times the volume required with connector\n
      ',RatioV)

```

Scilab code Exa 3.12 To Determine the Voltage at the far end

```

1
2 //To Determine the Voltage at the far end
3 //Page 117
4 clc;
5 clear;
6 r=0.3; // Loop Resistance per km
7 x=0.15; // Loop Reactance per km
8
9 deff('x=reactive(y,z,v)', 'x=(sin(acos(y)))*z*v*x
    /1000') //To find the reactive drop of the
    current
10
11 deff('x=active(y,z,v)', 'x=y*z*v*r/1000') //To find
    the reactive drop of the current
12
13 //Power Factors of the loads from left to right
14 pf1=0.707;
15 pf2=1;
16 pf3=0.8;
17 //Currents delivering the respective loads from left
    to right
18 I1=50;

```

```

19 I2=60;
20 I3=40;
21 //Length of the conductors in metres
22 l1=200;
23 l2=300;
24 l3=300;
25 //Effective length of the conductors in metres
26 L1=l1;
27 L2=l1+l2;
28 L3=l1+l2+l3;
29 //Active component drops of the respective currents
30 Va1=active(pf1,I1,L1);
31 Va2=active(pf2,I2,L2);
32 Va3=active(pf3,I3,L3);
33 Vat=Va1+Va2+Va3; //Total Active Component Drop
34 //Reactive component drops of the respective
    currents
35 Vr1=reactive(pf1,I1,L1);
36 Vr2=reactive(pf2,I2,L2);
37 Vr3=reactive(pf3,I3,L3);
38 Vrt=Vr1+Vr2+Vr3; //Total Reactive Component Drop
39
40 Vt=Vrt+Vat; // Total voltage drop
41
42 printf('The voltage drop at the far end is %g V\n',
    Vt)

```

Scilab code Exa 3.13 To determine line currents and neutral currents

```

1
2 //To determine line currents and neutral currents
3 //Page 118
4 clc;
5 clear;
6 V=400; // Supply Voltage

```

```

7 Vph=400/sqrt(3); //Phase Voltage
8 L=480*(10^3); //Balanced Load
9 //Loads at unit power factor
10 Lr=50*(10^3); //Load in R phase
11 Ly=150*(10^3); //Load in Y phase
12 Lb=200*(10^3); //Load in B phase
13 pf=0.8; //Power Factor lagging
14 theta=-1 * acos(pf); //Angle in radians of the
    balanced LAGGING current
15
16 I=L/(sqrt(3)*V*pf)*exp(%i*theta); //Balanced Current
    Magnitude
17 Ir=Lr/Vph; //Magnitude of R phase voltage
18 Iy=Ly/Vph; //Magnitude of Y phase voltage
19 Ib=Lb/Vph; //Magnitude of B phase voltage
20
21 //Vr is taken as reference
22 //Angles of the phase voltages wrt to the reference
23 Thetay=-120;
24 Thetab=120;
25 Thetar=0;
26 //Net Currents in the respective phases (RYB)
27 Irf=Ir+I;
28 Iyf=Iy+I;
29 Ibf=Ib+I;
30 //Angle of the above currents in degrees
31 r=atand( imag(Irf)/real(Irf));
32 y=atand( imag(Iyf)/real(Iyf));
33 b=atand( imag(Ibf)/real(Ibf));
34 //Angles of the above currents with respect to the
    reference
35 rf=Thetar+r;
36 yf=Thetay+y;
37 bf=Thetab+b;
38 //Effective Currents wrt to the reference voltage
39 Irn=abs(Irf)*exp(%i*%pi*rf/180)
40 Iyn=abs(Iyf)*exp(%i*%pi*yf/180)
41 Ibn=abs(Ibf)*exp(%i*%pi*bf/180)

```

```

42 In=Irn+Iyn+Ibn; // Neutral Current
43 //Note Take Vr as reference
44 printf('The Net Current in phase R is %g/-%g A\n',
        abs(Irf),rf);
45 printf('The Net Current in phase Y is %g/-%g A\n',
        abs(Iyf),yf)
46 printf('The Net Current in phase B is %g/-%g A\n',
        abs(Ibf),bf)
47 printf('The Net Neutral Current is %g/-%g A\n',abs(
        In),atand(imag(In)/real(In)))

```

Scilab code Exa 3.14 To determine the best location of the substation for a given

```

1
2 //To determine the best location of the substation
   for a given set of loads
3 //Page 123
4 clc;
5 clear;
6
7 //Various loads and how they are positioned on the
   corners of a square of length 25km
8 L1=5000; //(0,25)
9 L2=8000; //(25,25)
10 L3=3000; //(25,0)
11 L4=6000; //(0,0)
12
13 L=25; //Length of the square
14
15 TL=L1+L2+L3+L4; // Total load
16
17 X=((L1*0)+(L2*L)+(L3*L)+(L4*0))/TL // X coordinate
18 Y=((L1*25)+(L2*L)+(L3*0)+(L4*0))/TL // Y coordinate
19
20 printf('The Substation must be located at (%g km,%g

```

$km) \setminus n', X, Y)$

Chapter 4

Electric Drives and Industrial Applications

Scilab code Exa 4.1 To Determine Motor Parameters of Series Motor rated at 220V

```
1
2 //To Determine Motor Parameters of Series Motor
   rated at 220V
3 //Page 201
4 clc;
5 clear;
6 V=220; //Rated Voltage
7 N=1000; //Rated Speed
8 Ish=20; //Current without Armature being shunted
9 Ra=0.15; //Armature Resistance
10 Rf=0.2; //Field Resistance
11 Eb=V-(Ish*(Ra+Rf)); //Back EMF
12 K=Eb/(Ish*N); //Motor Constant
13 Rsh=20; //Shunt Resistance
14
15 //After Armature is shunted
16 x=poly(0, 'x'); //Field Current Variable Value
17 Ia=(Ish^2)/x; //Armature Current //Since Torque is
   assumed constant
```

```

18 I20=(V-(x*Rf))/Rsh; //Current Through the Shunt
    resistance
19
20 X=Ia+I20-x; //Polynomial to find field Current
21 x=roots(X(2)); //Numerical Value of Field Current
22
23 //Conditon to find the Positive Root
24 if(abs(x(1))+x(1)==0)
25     x=x(2);
26 else
27     x=x(1);
28 end
29
30 Ia=(Ish^2)/x; //Armature when shunted
31 Ebsh=V-(Ia*Ra)-(x*Rf); //Back Emf when shunted
32 Nsh=Ebsh/(K*x); //Speed when armature is shunted
33
34 //When Load is not there , Ia = 0
35 If=V/(Rsh+Rf); //Field Current at No Load
36 Ebnl=V-(If*Rf); //Back Emf at No Load
37 Nnl=Ebnl/(K*If); //Speed at No Load
38
39 Rex=5; //External Resistance
40 //At Starting when Eb = 0
41 Is=V/(Rex+Rf+(Ra*Rsh/(Ra+Rsh))); //Supply Current
42 Ifs=Is; //Field Current
43 Ias=Ifs*Rsh/(Rsh+Ra); //Armature Current At the
    Start
44
45 //Torque is directly proportional to square of
    Current
46 T20 = (Ish^2); //Torque at shunted condition
47 Tas= Ias*Ifs; //Torque at start
48
49 RT=Tas/T20; //Ratio of Torques
50
51 printf('a) The Speed of the Motor if the Load Torque
    is remains constant and the magnetic circuit is

```

```

    assumed unsaturated is %g RPM\n',Nsh)
52 printf('b) The No Load Speed of the Motor is %g RPM\n
    n',Nnl)
53 printf('c) The Starting Torque represented in terms
    of torque corresponding 20A (without armature
    shunted) is %g times\n',RT)

```

Scilab code Exa 4.2 To Determine the resistance to be inserted for various cases

```

1
2 //To Determine the resistance to be inserted for
    various cases
3 //Page 203
4 clc;
5 clear;
6
7 N1=1500; //Rated Speed
8 N2=1200; //Reduced Speed
9 rs=N2/N1; //Ratio of final speed to original speed
10 eff=80/100; //Efficiency
11 Pr=10*(10^3); //Power Rating
12 V=250; //Voltage supply
13 I1=Pr/(V*eff); //Full Load Current
14 Rf=110; //Field Resistance
15 Ra=0.25; //Armature Resistance
16 If=V/Rf; //Field Current
17 Ia1=I1-If; //Load Current
18 Eb=V-(Ia1*Ra); //Back EMF
19 Eb1200=Eb*rs; //Back EMF for 1200 rpm
20
21 def('x=Res(y)', 'x=((V-Eb1200)/y)') //Function to
    calculate the Resistance of various cases
22 //Torque directly proportional to current
23 Ia2=(rs)*Ia1; // Torque directly propostional to
    speed

```

```

24 Ia3=(rs^2)*Ia1; //Torque directly proportional to
    square of the speed
25
26 //Resistances for Various Cases
27 R1=Res(Ia1)-Ra;
28 R2=Res(Ia2)-Ra;
29 R3=Res(Ia3)-Ra;
30
31 def('x=Pow(y)', 'x=((V-Eb1200)/y)*(y^2)/1000') //
    Function to Find Power Loss
32 //Power Lost in kW for various cases
33 P1=Pow(Ia1);
34 P2=Pow(Ia2);
35 P3=Pow(Ia3);
36
37 printf('The External Resistances to be connected and
    the power loss for\n')
38 printf('a) The Load Torque is independent of the
    speed : %g ohm and %g kW respectively\n',R1,P1)
39 printf('b) The Load Torque is directly proportional
    to the speed : %g ohm and %g kW respectively\n',
    R2,P2)
40 printf('c) The Load Torque is directly proportional
    to square of the speed : %g ohm and %g kW
    respectively\n',R3,P3)

```

Scilab code Exa 4.3 To determine the speed for which the torque is maximum

```

1
2 //To determine the speed for which the torque is
    maximum
3 //Page 204
4 clc;
5 clear;
6

```

```

7 //In The Figure , The author has taken the resistance
  in series with the motor as 0.2 ohm, but in the
  figure it is given as 1 ohm
8
9 //This Doesn't Affect the calculation of the speed
  but it does affect the Maximum Torque
10
11 //So if we consider 1 ohm we get 0.8333 instead of
  2.5
12
13 //The Equation for the Torque is found out to be
  
$$2.5 * K2 * (V^2) * (1.1 - K1 * w) / ((76.1 - (50 * K1 * w)))$$

14
15 w=poly(0, 'w'); //Variable Value of w;
16
17 //Lets assume the value of  $2.5 * K2 * (V^2) = 1$  and  $K1 =$ 
  1 to particularly to calculate the co-efficients
18
19 T=2.5*(1.1-w)/((76.1-(50*w))^2); //Torque
20
21 //Maximum Torque is derivative of the above equation
22 X=derivat(T); //Polynomial to find the value co -
  efficient of w;
23 w=roots(X(2));
24 w=w(2); //We Choose value less than 1 to suit the
  differentiation process
25
26 Tmax=2.5*(1.1-w)/((76.1-(50*w))^2); //Maximum Torque
  Co-Efficient
27
28 printf('The Speed at which the torque is Maximum is
  (%g/K1) rad/sec\n',w)
29 printf('The Maximum Torque is (K2*(V^2)*%g*10^-4) Nm
  \n', (Tmax/(10^-4)))

```

Scilab code Exa 4.4 To Determine the slip at maximum torque and at full load

```
1
2 //To Determine the slip at maximum torque and at
   full load
3 //Page 205
4 clc;
5 clear;
6
7 Tl=1; //Assume that the Full Load torque is unity
8 Tst=125*Tl/100; //Starting Torque
9 Tmax=275*Tl/100; //Maximum Torque
10 si=1; //At the Start
11 RT=Tst/Tmax; //Ratio of Starting Torque and Maximum
   Torque
12 //Rl is negligible , eliminating the terms with Rl and
   hence formulating the starting torque and
   Maximum Torque we get
13
14 //  $T_{st}/T_{max} = 2*sT_{max}*s/((sT_{max}^2)+(s^2))$ 
15 //Where sTmax is the Slip at Maximum Torque and s is
   the slip at the start
16
17 sTmx=poly(0,'sTmx'); //Variable Value of slip at
   Maximum Torque
18
19 X=RT-(2*sTmx*si/((sTmx^2)+(si^2))); //Polynomial to
   find sTmax
20
21 sTmx=roots(X(2));
22
23 //Condition to find the value of sTmax less than 1
24 if(floor(sTmx(1))==0)
25     sTmx=sTmx(1);
26 else
27     sTmx=sTmx(2);
28 end
29
```

```

30 s=poly(0,'s'); //Variable value of slip for Starting
    torque at full load
31
32 Y=(Tl/Tmax)-(2*sTmx*s/((sTmx^2)+(s^2))); //
    Polynomial to find s
33
34 s=roots(Y(2)); //Numerical Value of s
35 //Condition to find the value of s less than 1
36 if(floor(s(1))==0)
37     s=s(1);
38 else
39     s=s(2);
40 end
41
42 printf('i) The Slip at Maximum Torque is %g\n',sTmx)
43 printf('ii) The Slip at Full Load is %g\n',s)

```

Scilab code Exa 4.5 To Determine the Maximum Torque as ratio of nominal value of v

```

1
2 //To Determine the Maximum Torque as ratio of
    nominal value of voltage
3 //Page 205
4 clc;
5 clear;
6
7 //Resistances and Reactance for Nomrml Value of
    Volttage and Frequency
8 w=50; //Frequency of supply
9 V=1; //Assumed Voltage
10 R1=0.05;
11 R2=0.05;
12 x1=0.3;
13 x2=0.3;
14

```

```

15 //Resistance and reactance at half voltage and half
    frequency
16 wh=50/2; //Frequency of supply
17 Vh=1/2; //Assumed Voltage
18 R1h=0.05;
19 R2h=0.05;
20 x1h=0.3/2;
21 x2h=0.3/2;
22
23 //Function to Determine the Maximum Torque
24 deff('x=Max(a,b,c,d,e)', 'x=(3*(a^2))/(2*b*(c+(sqrt((
    c^2)+((d+e)^2))))))')
25 //Function to Determine to the Starting Torque
26 deff('x=Start(a,b,c,d,e,f)', 'x=(3*(a^2)*f)/(b*(((c+f
    )^2)+((d+e)^2)))')
27 //Maximum Torques and Starting Torques at
28 //Normal Voltage And Frequency
29 Tm=Max(V,w,R1,x1,x2);
30 Tst=Start(V,w,R1,x1,x2,R2);
31 //Half Voltage and Half Frequency
32 Tmh=Max(Vh,wh,R1h,x1h,x2h);
33 Tsth=Start(Vh,wh,R1h,x1h,x2h,R2h);
34
35 //Ratio of torques as fraction of its normal value
36 R1=Tmh/Tm;
37 R2=Tsth/Tst;
38
39 printf('i)The Maximum Torque at the reduced value of
    supply as a fraction of its normal value is %g\n
    ',R1)
40 printf('ii)The Starting Torque at the reduced value
    of supply as a fraction of its normal value is %g
    \n',R2)

```

Scilab code Exa 4.6 To Determine the starting torque and starting current for vari

```

1
2 //To Determine the starting torque and starting
   current for various starters
3 //Page 206
4 clc;
5 clear;
6 V=400; //Rated Voltage //Phase Voltage //Delta
   Connected
7 Pr=50*735.5; //Rated Power
8 N=750; //Rated Speed
9 s=4.5/100; //Full load Slip
10 I1=50; //Full Load Current
11 Z=2.5; //Impedance per phase
12 w=2*%pi*N/60; //Angular Frequency
13 T1=Pr/w; //Full Load Torque
14 Ist=V/Z; //Starting Current
15 Isl=sqrt(3)*Ist; //Line Current
16
17
18 //DOL Starter
19 Tstd=((Isl/I1)^2)*s*T1; //Starting Torque
20
21 //Star - Delta Starter
22 Tsts=T1*((1/sqrt(3))^2); //Starting Torque
23
24 //Autotransformer
25 tap=70/100; //Tapping
26 Tsta=(tap^2)*T1; //Starting Torque
27
28 printf('The Starting Current for all the three
   starter are the same that is %g A\n',Isl)
29 printf('The Starting torque for:\n')
30 printf('i) D.O.L Starter : %g Nm\n',Tstd)
31 printf('ii) Star -Delta Starter : %g Nm\n',Tsts)
32 printf('ii) Auto Transformer with 70 percent tapping
   : %g Nm\n',Tsta)

```

Scilab code Exa 4.7 To Detemine Minimum Starting Current drawn from the supply

```
1
2 //To Detemine Minimum Starting Current drawn from
   the supply
3 //Page 207
4 clc;
5 clear;
6
7 TR=0.5; //Torque Ratio , Tst:Tl
8 V=400; //Supply Voltage
9 Pr=30*735; //Power Rating
10 N=500; //Rated Speed
11 eff=0.85; //Efficieny
12 pf=0.88; //Power Factor
13 Is=150; //Short Circuit Current
14 s=5/100; //Full load Slip
15 pfs=0.25; //Power Factor of the Short Cicuit Current
16 I1=Pr/(sqrt(3)*V*eff*pf); //Full Load Current
17
18 //Case 1
19 x=poly(0, 'x'); //Tapping Percent
20 X=TR-(((Is/I1)^2)*(x^2)*s); //Polynomial to
   determine the 'x'
21 x=roots(X)
22 x=x(1); //Taking the Postive Root
23 Ist1=Is*x*x; //Starting Current numerical Value
24
25 //Case 2
26 Ist2=sqrt((TR*(I1^2)/s)); //Starting Current for
   Full Voltage
27
28 Z1=V/(sqrt(3)*Ist2); //impedance for the starting
   current at full voltage
```

```

29 Z2=(V/(sqrt(3)*Is))*exp((%i*%pi*acosd(pfs))/180); //
    Phasor form of the Impedance of for the short
    circuit current
30
31 R=sqrt((Z1^2)-(imag(Z2)^2));
32 Rs=R-real(Z2); //Resistance to be connected the
    stator circuit
33
34 printf('i) The Tapping of the Transformer is %g
    percent and the starting Current is %g A\n',(x
    *100),Ist1)
35 printf('ii) The Starting Current is %g A and the
    Resistance to be added to the stator circuit is
    %g ohm\n',Ist2,Rs)

```

Scilab code Exa 4.8 To Determine the value of resistor connected in series to run

```

1
2 //To Determine the value of resistor connected in
    series to run the fan at a particular speed
3 //Page 208
4 clc;
5 clear;
6 Pf=100*(10^3); //Power rating of the fan
7 f=50; //Supply Frequency
8 V=400; //Supply Line Voltage
9 V1=V/sqrt(3); //Phase Voltage
10 SR=1.4; //Ratio of stator to rotor turns
11 Rr=0.03; //Rotor Resistance per phase
12 R2=SR*SR*Rr; //Rotor Resistance referred to the
    stator
13 N=240; //Rated Speed
14 w=2*%pi*N/60; //Angular Frequency
15 P=24; //number of poles
16 Ns=120*f/P; //Synchronous Speed

```

```

17 ws=2*pi*Ns/60; //Synchronous angular Frequency
18 s=(Ns-N)/Ns; //Slip
19 T=Pf/w; //Full Load Torque
20 x2=poly(0, 'x2'); //Reactance of Rotor referred to
    stator, Variable Value
21
22 X=T-(3*(V1^2)*R2/(ws*s*((R2/s)^2)+(x2^2))); //
    Polynomial Expression to find X2
23 X2=roots(X(2)); //Numerical Value of Rotor
    Resistance
24 X2=X2(1);
25 N1=180; //Speed of the fan
26 s1=(Ns-N1)/Ns; //Slip
27 T1=T*((N1/N)^2); //Torque
28 R21=poly(0, 'R21'); //Variable Value
29 Y=T1-(3*(V1^2)*R21/(ws*s1*((R21/s1)^2)+(X2^2)));
    //Polynomial Expression to find R21
30 R21=roots(Y(2)); //Numerical Value
31
32 //R21 = 0.7545054 and 0.0100813
33 //We have to choose the value of R21 such that it is
    greater than R2
34
35 R21=R21(1);
36 Rex=(R21-R2)/(SR*SR); //External Resistance
37
38 printf('The Value of Resistance to be connected in
    series so that teh fan runs at 180 rpm is %g ohm\
    n',Rex)

```

Scilab code Exa 4.9 To Determine the time taken to start the motor

```

1
2 //To Determine the time taken to start the motor
3 //Page 222

```

```

4  clc;
5  clear;
6  HP=746; //According to the TextBook
7  P=150*HP; //Power Rating
8  Lim=1.5; //Limited Current Factor
9  Eshp=5000; //energy stored per hp
10 N=750; //Rated Speed
11 w=N*2*%pi/60; //Angular Frequency
12 Es=Eshp*P/HP; //Total Energy Stored
13 Tf1=P/w; //Full Load Torque;
14 Ts=Tf1*Lim; //Starting Torque
15 Ta=Ts-Tf1; //Torque Available for acceleration
16 I=2*Es/(w^2); //Moment of Inertia
17
18 a=Ta/I; //Angular Accleration
19
20 //Intergrating wrt angular frequency
21 t=integrate('1/a','x',0,w); //Time Taken to start //
    Angular Acceleration is constant
22
23 printf('The time taken to start the motor if the
    load torque is equal to full load torque\n during
    the starting period and the current is limited
    to 1.5 times the full load current is %g seconds\n
    n',t)

```

Scilab code Exa 4.10 To time taken and the number of revolutions made before the m

```

1
2 //To time taken and the number of revolutions made
    before the motor is stopped
3 //Page 223
4 clc;
5 clear;
6 P=50*735.5; //Power Rating

```

```

7 V=400; //Rated Voltage
8 N=750; //Rated Speed
9 w=N*%pi*2/60; //Angular Frequency
10 I=20; //Moment Inertia
11 pf=0.95; //Full Load power factor
12 eff=90/100; //Overall Efficiency
13 Rb=2; //Braking Resistor
14
15 Tm=P/w; //Full Load Torque
16 Ifl=P/(sqrt(3)*V*eff*pf); //Full Load Current
17 Ilb=V/(sqrt(3)*Rb); //The Line Current at the start
    of braking
18 Te=Tm*Ilb/Ifl; //Torque for electric braking
19
20 Tb=Te+Tm; //Total Braking torque
21
22 K=Te/w; //Motor constant
23 B=Tb/I; //Retardation
24
25 ts=integrate('I/(Tm+(K*x))','x',0,w); //Time Taken
    for the motor to stop
26
27 NoR=(1/(2*%pi*K))*integrate('((Tm+(K*w))*exp(-1*K*t/
    I))-Tm','t',0,ts); //Number of Revolutions
28
29 //Note The Value of time in the textbook is wrong,
    Calculation error
30
31 printf('The Time Taken for the Motor to stop is %g
    seconds\n',ts)
32 printf('The Number of revolutions undergone before
    it stops is %g revolutions which is approx %g
    revolutions\n',NoR,round(NoR))

```

Scilab code Exa 4.11 To Determine the 30 mins rating of motor

```

1
2 //To Determine the 30 mins rating of motor
3 //Page 229
4 clc;
5 clear;
6 Pr=100; //Power Rating in horse power
7 Tc=90*3600; //Time Constant
8 Tr=30*3600; //Time to determine the rating
9 theta=50; //Temperature Rise
10 theta1=50; //Temperature rise for the new case
11
12 P=poly(0, 'P'); //Power rating for 30min variable
    value
13
14 //Loss Corresponding to 30 mins
15
16 P30=(P/Pr)^2; // Times the Power loss corresponding
    to the 100 hp motor
17
18 theta2=theta*P30; //Final Temperature rise if run
    continuously
19
20 X=theta1-(theta2*(1-exp(-1*Tr/Tc))); //Polynomial
    to find the 30min rating
21
22 P=roots(X); //Numerical Value
23 //We Consider the Positive Real Value of the root
24
25 P=P(1); //Power Rating of the 1/2 Hr
26
27 printf('The 1/2 Hr Power Rating is %g hp\n',P)

```

Scilab code Exa 4.12 To Determine the Continuous Rating of the Motor for Suitable C

1

```

2 //To Determine the Continous Rating of the Motor for
   Suitable Conditions
3 //Page 229
4 clc;
5 clear;
6 T=120; //Total Time
7 t1=0; //Power Rise to 1050
8 t2=15; //Constant Power of 600
9 t3=85; //Regenerative Breaking at 200
10 t4=95; //Motor At Rest
11 t5=120; //End
12 //Note Power Rating are in Horse Power
13 HP1=1050; //First Rise
14 T1=(t2-t1); //Time for Rise of the Horse Power
15 HP2=600; //Constant Power
16 T2=(t3-t2); //Time For which the power is constant
17 HP3=200; //Braking Initial Point
18 T3=(t4-t3); //Time Period of Breaking
19
20 //Different Intergration
21 I1=integrate('((HP1/T1)*t)^2','t',t1,t2);
22 I2=integrate('(HP2^2)','t',t2,t3);
23 I3=integrate('((HP3/T3)*t)^2','t',0,T3);
24
25 HPrms=sqrt((I1+I2+I3)/120);
26
27 HPav=((HP1*T1/2)+(HP2*T2)+(HP3*T3/2))/T);
28
29 printf('i) The H.P Rating according to the RMS Value
   of loading is %g hp\n',HPrms)
30 printf('ii) The H.P Rating according to the Average
   Value of Loading is %g hp\n',HPav)

```

Scilab code Exa 4.13 To estimate the final temperature of the machine and its time

```

1
2 //To estimate the final temperature of the machine
   and its time constant
3 //Page 230
4 clc;
5 clear;
6
7 Eff=90/100; //Efficiency
8 P=15*735.5; //Power Rating in W
9 D=75*(10^-2); //Cylinder Diameter
10 L=120*(10^-2); //Length of the Cylinder
11 OSHD=12; //Outer Surface Heat Dissipation
12 A=%pi*L*D; //Curved Surface Area
13 M=450; //Motor Weight
14 SH=700; //Specific Heat
15
16 HTC=M*SH/(A*OSHD*3600); //The Heating Time Constant
17 Loss=(P/Eff)-P; //Loss in Motor
18
19 FT=Loss/(A*OSHD); //Final Temperature
20 OSHD2=20; //Semi Closed Cooling Capacity
21 LossSC=FT*A*OSHD2; //Loss in Semi Closed System
22
23 L2=poly(0, 'L2'); //Variable Load Value
24 X=(L2/(L2+LossSC)-Eff); //Polynomial to Calculate L2
25 L2=roots(X(2)); //Numerical Value of The Load
26
27 printf('The Final Temperature of the Machine and its
   Heating Time Constant is %g degree Celsius and
   %g Hrs respectively \n',FT,HTC)
28 printf('The Load which could be delivered is %g W\n'
   ,L2)

```

Scilab code Exa 4.14 To Determine the Moment of inertia for a slip of 8 percent

```

1
2 //To Determine the Moment of inertia for a slip of
   8 percent
3 //Page 233
4 clc;
5 clear;
6 Tl=900; //load Torque
7 To=0;
8 s=8/100; //Slip
9 t=15; //time period for which the load torque was
   applied
10 Tm=675; // Limited motor Torque
11 N=600; //No Load Speed
12 Wo=2*%pi*N/60; //Angular Frequency
13 K=s/Tm; //Motor Constant
14 I=(t/(Wo*K*log((Tl-To)/(Tl-Tm)))); //Moment of
   Inertia
15
16 printf('The Moment of inertia of the flywheel is %g
   Kg m^2\n',I)
17 //Please Note the Calculation Mistake in the book. (
   Power of e)

```

Scilab code Exa 4.15 To determine the Moment of inertia of the flywheel

```

1
2 //To determine the Moment of inertia of the
   flywheel
3 //Page 234
4 clc;
5 clear;
6 Pi=100*(10^3); //Power Rating
7 P=6; //Poles of the Machine
8 N=950; //Rated Speed
9 Wo=2*%pi*N/60; //Angular Frequency

```

```

10 Ta=3000; //Additional Torque
11 Tmin=600; //Constant Load Torque
12 Tm=Pi*60/(2*%pi*N); //Rated Torque
13 Tmax=2*Tm; //Maximum Torque
14 Tl=Tmin+Ta; //Total Torque
15 tp=15; //Time for which the additional torque is
    applied
16 //Assuming Slip Characteristic to Be a Straight line
17 s=0.05;
18 K=s/Tm; //Motor Load Constant
19
20 I=tp/(Wo*K*log((Tl-Tmin)/(Tl-Tmax))); //Moment Of
    inertia
21
22 Tm1=1500; //Trasistion Torque
23
24 t=I*Wo*K*log((Tmax-Tmin)/(Tm1-Tmin)); //Time Taken
    to become 1500 Nm
25
26 printf('a) The Moment of inertia of the flywheel
    for Maximum Torque to be twice the rated torque
    is %g Nm\n',I)
27 printf('b) The Time Taken after the removal of
    additional load before the motor torque becomes
    1500 Nm is %g seconds\n',t)

```

Scilab code Exa 4.16 To determine the speed and torque of a single phase full conv

```

1
2 //To determine the speed and torque of a single
    phase full converter
3 //Page 241
4 clc;
5 clear;
6 P=5*735.5; //Power Rating

```

```

7 Vr=220; //Rated Voltage
8 N=1500; //Rated Speed
9
10 Ra=0.25; //Armature Resistance
11 Ia=20; //Rated Armature Current
12 Vs=250; //Supply Voltage
13 Kaphi=0.0278; //Motor Voltage Constant
14 a=30; //Firing Angle
15
16 Va=2*sqrt(2)*Vs*cosd(a)/%pi; //Armature Voltage
17 Eb=Va-(Ia*Ra); //Back EMF
18 w=Eb/(Kaphi*60); //Angular Frequency in Radians per
    second
19 Ta=Kaphi*Ia*60; //Torque
20
21 printf('The Speed and Torque of the Single Phase
    Full Converter are %g rad/s and %g Nm
    respectively\n',w,Ta)

```

Scilab code Exa 4.17 To Determine the motor current and torque for a particular firing angle

```

1
2 //To Determine the motor current and torque for a
    particular firing angle
3 //Page 242
4 clc;
5 clear;
6 P=15*735.5; //Power Rating of the Motor
7 Raw=0.2; //Combined Armature and Field winding
    resistance
8 N=1000; //Speed
9 K=0.03; //Motor Constant
10 a=30; //Firing Angle
11 E=250; //Supply Voltage
12

```

```

13 //Semi Converter
14 w=2*pi*N/60; //Angular Frequency
15 Vas=sqrt(2)*E*(1+cosd(a))/pi;
16 Ias=Vas/(Raw+(K*w)); //Armature Current
17 Tas=K*(Ias^2); //Motor Torque
18
19 //FullConverter
20 Vaf=2*sqrt(2)*E*cosd(a)/pi;
21 Iaf=Vaf/(Raw+(K*w)); //Armature Current
22 Taf=K*(Iaf^2); //Motor Torque
23
24 //Accurate Caclulation , Rounding Off not done.
25 printf('The Motor Current And Motor Torque for:\n')
26 printf('i) Semi Converter : %g A and %g Nm\n',Ias,
    Tas)
27 printf('ii) Full Converter : %g A and %g Nm\n',Iaf,
    Taf)

```

Scilab code Exa 4.18 To determine the pulse width for a particular average value o

```

1
2 //To determine the pulse width for a particular
    average value of back EMF
3 //Page 244
4 clc;
5 clear;
6 E=220; //DC Supply
7 Raw=0.2; //Combined Armature and Field Winding
    Resistance
8 Ia=25; //Average Current
9 f=200; //Chopper Frequency
10 Eb=100; //Average Value of Back EMF
11
12 Eav=(Ia*Raw)+Eb; //Average Load Voltage
13 Ton=Eav*1000/(E*f); // Pulse Width in m sec

```

```
14
15 printf('The Pulse Width for the 100V back EMF is %g
    m seconds\n',Ton)
```

Scilab code Exa 4.19 To determine the range of speed control and duty cycle

```
1
2 //To determine the range of speed control and duty
  cycle
3 //Page
4 clc;
5 clear;
6 E=220; //DC supply voltage
7 Ra=0.2;
8 Kaphi=0.08; // Motor Constant
9 Ia=25; //Average Current
10 Eav=Ia*Ra; //Voltage Drop
11
12 //Eav = lamda * E;
13 //For Maximum Speed, lambda =1
14
15 lamda=Eav/E;
16
17 lamda1=1; //Maximum Speed;
18 Eb= (E*lamda1)-(Ia*Ra); //Back EMF
19
20 Speed = Eb/Kaphi; // Maximum Speed
21
22 printf('i) The Range of speed control is 0<N<%g \n',
    Speed)
23 printf('ii) The Range of duty cycle is %g<lamda<%g\n
    ',lamda,lamda1)
```

Chapter 5

Electric Heating And Welding

Scilab code Exa 5.1 To determine the size and length of the wire

```
1
2 //To determine the size and length of the wire
3 //Page 282
4 clc;
5 clear;
6
7 r=poly(0, 'r'); //Variable Value of radius
8 A=%pi*(r^2); //Area of cross section of the wire
9 V=220; //Supply Voltage
10 P=20*(10^3); //Power input
11 //Temperatures
12 T1=1127; //Wire
13 T2=427; //Charge
14
15 R=(V^2)/P; //Resistance of the wire
16 e=0.9; //emissivity constant
17 K=0.6; //Radiation Efficiency
18
19 p=1.09*(10^-6); //Resistivity
20
21 l=R*A/p; //Length in term of 'r'
```

```

22
23 H=5.72*e*K*(((T1+273)/100)^4)-(((T2+273)/100)^4));
    //Heat dissipated per sq.m of the surface
24 CSA=%pi*2*r*l; //Curved surface area
25
26 CSAn=P/H; //Numerical Value of Curved surface area
27
28 X=CSA-CSAn; //Polynomial to find 'r'
29
30 disp(X);
31 printf('The real roots of the above equation gives
    the value of the radius\n')
32 r=roots(X); //Numerical Value of radius
33 disp(r);
34 printf('Choosing the real roots from the list above\
    n')
35 r=r(3);
36 l=CSAn/(2*pi*r); //Numerical Value of length
37
38 //For charge temperature to be cold
39 Ti=25; //Cold Temperature
40 T=poly(0,'T'); //Variable value of the element
    temperature
41 Hi=5.72*e*K*(((T+273)/100)^4)-(((Ti+273)/100)^4));
    //Heat dissipated per sq.m of the surface
42 CSA=%pi*2*r*l; //Curved surface area
43 Y=Hi-H; //polynomial to find the temperature of the
    element
44
45 //Roots of T must be real
46 T=roots(Y); //Numerical Value
47 disp(Y);
48 printf('The real roots of the above equation gives
    the value of the element temperature\n')
49 disp(T);
50 printf('Choosing the real roots from the list above\
    n')
51 T=T(4);

```

```

52
53 printf('\n\nThe length and radius of the wire
    element are %g m and %g mm respectively\n',l,(r
    *1000))
54 printf('The Temperature of the element when the
    charge is cold is %g degree celsius\n',T)

```

Scilab code Exa 5.2 To determine the various temperature by changing the connectio

```

1
2 //To determine the various temperature by changing
    the connection of the resistance elements
3 //Page 284
4 clc;
5 clear;
6
7 //Note that the value in kelvin of the first case in
    the textbook is wrong
8
9 //P is directly proportion to V^2 and H is directly
    propostional to KT^4
10 //Different Temperatures for different
    configurations
11 T1=1125; //Temperature in First Case
12 T2=poly(0, 'T2');
13 T3=poly(0, 'T3');
14 T4=poly(0, 'T4');
15
16 //Multiplying Factors to the square of voltages
17 V1=1; //Line to Line Voltage;
18 V2=V1/2; //when connected in series first and then
    delta
19 V3=V1/(2*sqrt(3)); //when connected in series and
    then in star
20 V4=V1/(sqrt(3)); //When connected in parallel and in

```

```

        star
21
22 //To find the power loss in each case
23 deff('x=pow(y)', 'x=(y^2)');
24 P1=pow(V1);
25 P2=pow(V2);
26 P3=pow(V3);
27 P4=pow(V4);
28
29 //To find the heat dissipated from each case
30 deff('x=heatdiss(y)', 'x=(y^4)');
31 H1=heatdiss(T1+273);
32 H2=heatdiss(T2+273);
33 H3=heatdiss(T3+273);
34 H4=heatdiss(T4+273);
35
36 //Polynomials to find the temperature in degree
    celsius
37 deff('x=temp(y, z)', 'x=(P1/y)-(H1/z)');
38 X2=temp(P2, H2);
39 X3=temp(P3, H3);
40 X4=temp(P4, H4);
41
42 //Temperature Numerical Value
43 T2=roots(X2(2));
44 T3=roots(X3(2));
45 T4=roots(X4(2));
46
47 //Only to consider Real Roots
48 T2=T2(4);
49 T3=T3(4);
50 T4=T4(4);
51
52 printf('The Temperature for the following
    configurations are:\n')
53 printf('Two Groups connected in series first and
    then in delta : %g degree Celsius\n', T2)
54

```

```

55 printf('Two Groups connected in series first and
        then in star : %g degree Celsius\n',T3)
56
57 printf('Two Groups connected in parallel first and
        then in star : %g degree Celsius\n',T4)

```

Scilab code Exa 5.3 To Determine the average KW input to the furnace

```

1
2 //To Determine the average KW input to the furnace
3 //Page 288
4 clc;
5 clear;
6 M=10*(10^3); //Mass of Steel Melted
7 t=2*3600; //Time Taken to Melt the steel
8 eff=50/100; //Overall Efficiency
9 I=9000; //Current Input
10 R=0.003; //Resistance
11 X=0.005; //Reactance
12 SH=0.12; //Specific Heat
13 LHF=8.89*(10^3); //Latent Heat of Fusion
14 Tm=1371; //Melting Point
15 Ti=20; //Room Temperature
16
17 Hm=M*LHF; //Heat Required for melting
18 Hr=M*SH*(Tm-Ti)*1000; //Heat Required to raise the
    temperature
19 Ht=Hm+Hr; //Total Amount of heat required
20
21 E=Ht*4.2/(3600); //Energy in Whr
22 P=E*3600/t; //Power
23
24 Pa=P/eff; //Actual Power Input to the Furnace
25
26 Vt=Pa/(3*I); //V Cos theta

```

```

27 //The Above voltage is the sum of arc drop and drop
    in resistance load
28
29 Va=Vt-(I*R); //Arc Drop
30 Vx=I*X; //Reactance Drop
31 Vs=sqrt((Vt^2)+(Vx^2)); //Supply Voltage
32 S=3*Vs*I/1000; //KVA input
33
34 printf('The Average kW input to the furnance is %g
    kW\n',Pa/1000)
35 printf(' The Arc Voltage is %g V\n',Va)
36 printf(' The kVA input is %g kVA\n',S)
37
38 //Accurate Answers, No rounding off.

```

Scilab code Exa 5.4 To determine the efficiency of a high frequency induction furna

```

1
2 //To determine the efficiency of a high frequency
    induction furnance
3 //Page 294
4 clc;
5 clear;
6 t=10*60; //Time Taken to rise temperature in seconds
7 M=1.815; //Mass of aluminium melted
8 Pi=5*(10^3); //Power Input
9 Ti=15; //Initial Temperature
10 Tm=660; //Melting Point of Al
11 SHAl=0.212; //Specific heat of Al
12 LHFA1=76.8*(10^3); //Laten Heat of fusin in Cal/Kg
13
14 Hm=M*LHFA1; //Heat required to melt Al
15 Htr=SHAl*M*1000*(Tm-Ti); //Heat required to raise
    the temperature
16 HTot=Hm+Htr; //Total Heat Required

```

```

17
18 HToth=HTot*3600/t; //Heat required per hour
19
20 Po=HToth*4.2/3600; //Power Output
21
22 eff=Po*100/Pi; //Efficiency
23
24 printf('The Efficiency of the High Frequency
    Induction Furnace is %g percent\n',eff)

```

Scilab code Exa 5.5 To Determine the equivalent resistance of the charge and current

```

1
2 //To Determine the equivalent resistance of the
   charge and current
3 //Page 294
4 clc;
5 clear;
6
7 f=960; //Frequency
8 N1=20; //Primary Turns
9 N2=1; //Secondary is Single Turn
10 Pi=325*(10^3); //Power Input
11 Di=45; //Internal Diameter
12 l=50; //Depth of the charge
13
14 //Assumptions
15 p=200*(10^-6); //Resistivity
16 M=1; //For Molten Steel
17
18 t=(1/(2*%pi))*sqrt(p*(10^9)/(M*f)); //Depth of
   penetration of the current
19 A=t*l; //Effective Area
20 Dm=Di+t; //Mean Diameter
21 Dmcf=%pi*Dm; //Mean Length of current flow

```

```

22 Rc=p*Dmcf/A; //Resistance of the Cylinder
23
24 Is=sqrt(Pi/Rc); //Current flowing through secondary
25 Ip=Is*N2/N1; //Primary Current
26
27 printf('The Equivalent Resistance of the cylinder is
        %g * 10-6 ohm\n',Rc/(10-6))
28 printf('The Required Current in the primary is %g A\
        n',Ip)

```

Scilab code Exa 5.6 To Detemine power absorbed and the power factor

```

1
2 //To Detemine power absorbed and the power factor
3 //Page 295
4 clc;
5 clear;
6 Vs=15; //Secondary Voltage
7 P=500*(103) // Power Taken
8 pfs=0.6; //Power Factor
9
10 Is=P/(Vs*pfs); //Secondary Current
11
12 //Taking Current as Reference voltage will be
13 t=acosd(pfs); //Power Factor Angle
14 Vsp=Vs*(exp(%i*(t/180)*%pi)); //Phasor Secondary
    Voltage
15 R=Vsp/Is; //Impedance
16
17 //if the resistance is doubled, The Total impedance
    doubles, Considering Vs as reference
18 R2=real(R)+R;
19 I2=Vs/R2; //New Current
20 pfn=cosd(atan2(imag(I2),real(I2))); //power factor
    of new current

```

```

21 Pab=Vs*abs(I2)*pfn/1000; //Power Absorbed
22
23 printf('The Power Factor and The Absorbed power are
    %g lagging and %g kW respectively.\n',pfn,Pab)

```

Scilab code Exa 5.7 To determine the Voltage Required and Current Drawn

```

1
2 //To determine the Voltage Required and Current
  Drawn
3 //Page 299
4 clc;
5 clear;
6 t=2*(10^-2); //Thickness
7 A=150*(10^-4); //Area of the slab
8 Er=4; //Relative Permittivity
9 pf=0.04; //Power Factor
10 f=30*(10^6); //Frequency of supply
11 w=2*pi*f; //Angular Frequency
12 P=200; //Power Required
13 Eo=8.854*(10^-12); //Permittivity of free space
14
15 C=Er*Eo*A/t; //Capacitance
16 Xc=1/(C*w); //Capacitive Reactance
17 phi=acosd(pf); //power factor angle
18 R=tand(phi)*Xc; //Resistance
19 V=sqrt(P*R); //Voltage
20 I1=V/R; //Current
21 Ic=V/Xc; //Curent through the Capacitor
22 It=sqrt((I1^2)+(Ic^2)); //Total Current
23
24 Vn=600; //Limited Voltage
25 Rn=(Vn^2)/P; //New Resistance
26 wn=tand(phi)/(C*Rn); //New Angular Frequency
27 fn=wn/(2*pi); //New Frequency

```

```

28
29 printf('The Current And Voltage are %g A and %g V
    respectively\n',It,V)
30 printf('For the New Voltage the frequency is %g MHz\
    n',(fn/(10^6)))

```

Scilab code Exa 5.8 To estimate the voltage and Current during heating

```

1
2 //To estimate the voltage and Current during heating
3 //Page 300
4 clc;
5 clear;
6 l=30; //Length
7 b=15; //Breadth
8 t=2; //Thickness
9 t1=20; //Initial Temperature
10 t2=180; //Final Temperature
11 T=10*60; //Time Period in Seconds
12 f=40*(10^6); //Frequency of supply
13 w=2*pi*f; //Angular Frequency
14 SH=0.35; //Specific Heat Of Wood
15 Er=5; //Relative Permittivity
16 Eo=8.854*(10^-12); // Permittivity of free space
17 pf=0.05; //Power Factor
18 Eff=90/100; //Efficiency
19 p=0.55; //Density
20 A=l*b; //Area of the wooden board
21 W=(A*t)*p/1000; //Weight of wood in kilograms
22 H=W*SH*(t2-t1); //Heat required to raise temperature
23 E=H*4.2/3600; //Energy in kWhr
24 P=E*3600/T; //In kilowatts
25 AP=P*1000/Eff; //Actual Power
26 C=Eo*Er*A*(10^-2)/t; //Capacitance
27 phi=acosd(pf); //Power Factor Angle

```

```
28 del=(90-phi)*%pi/180; //In Radian
29 V=sqrt(AP/(w*C*del)); //Voltage
30 I=V*w*C; //Current
31
32 printf('The Voltage and Current are %g V and %g A
        respectively\n',V,I)
```

Chapter 6

Illumination Engineering

Scilab code Exa 6.1 To determine the value of resistance to be connected in the mo

```
1
2 // To determine the distance of a 25 cp lamp for
   various illumination
3 //Page 327
4 clc;
5 clear;
6
7 //Candle power of the lamp
8 I=25;
9
10 // Various illumination levels
11 E1=5;      //Case1
12 E2=15;     //Case2
13 E3=8;      //Case3
14
15 // According to the law of illumination  $E = I/(r^2)$ 
   ;
16 // Using the above equation we find the distances
   for the above three illuminations
17
18 r1= sqrt(I/E1);
```

```

19 r2= sqrt(I/E2);
20 r3= sqrt(I/E3);
21
22 printf('a) The distance for %g flux illumination
        from the normally placed screen is %g m \n',E1,r1
        )
23 printf('b) The distance for %g flux illumination
        from the normally placed screen is %g m \n',E2,r2
        )
24 printf('c) The distance for %g f.c illumination from
        the normally placed screen is %g ft \n',E3,r3)

```

Scilab code Exa 6.2 To determine the total radiation sent vertically downward from

```

1
2 //To determine the total radiation sent vertically
  downward from a lamp of 1500 cp
3 //Page 327
4 clc;
5 clear;
6
7 //Candle power of the lamp
8 P=1500;
9
10 //Since the flux required corresponds to the one
    that lies in a plane passing through the lamp
    vertically downwards the angle is given by
11 Angle= %pi; // Angle in radians
12
13 Flux=P*Angle;
14
15 printf('The total radiation sent vertically downward
        is %g lumens \n',Flux)

```

Scilab code Exa 6.3 To calculate Spherical Candle Power and intensity of illumination

```
1
2 //To calculate Spherical Candle Power and intensity
  of illumination
3 //Page 327
4 clc;
5 clear;
6
7 Tf= 1500; //Total
  flux
8 D=3 //
  Distance from the source
9 CP=200; //Candle
  power of the Uniform Source
10 E=CP/(D^2); //
  Maximum Illumination Intensity
11 deff('y=IntIllumi(x)', 'y=E*sind(x)'); //
  Function to determine the intensity of
  illumination for various conditions
12
13
14 MSCP=Tf/(4*%pi); //The
  Mean spherical candle power
15
16 //Various Angles at which the rays falls on the
  surface
17 t1=90;
18 t2=60;
19 t3=0;
20
21 //Illumination Intensities at various angles
22 E1=IntIllumi(t1);
23 E2=IntIllumi(t2);
```

```

24 E3=IntIllumi(t3);
25
26 printf('a)The Mean Spherical Candle Power is %g c.p\
    n',MSCP)
27 printf('b The illumination intensities for the
    following cases are: \n')
28 printf('i) Normal : %g lux\n',E1)
29 printf('ii) Inclined at 60 degrees : %g lux\n',E2)
30 printf('iii) Parallel : %g lux\n',E3)

```

Scilab code Exa 6.4 To determine average illumination and illumination at various

```

1
2 //To determine average illumination and illumination
    at various points on the area
3 //Page 328
4 clc;
5 clear;
6
7 CP=200; //Candle power of the lamp
8 D=4; //Vertical Distance from the
    centre of the area to the lamp
9 R=5/2; //Radius of the given area
10 De=sqrt((D^2)+((R)^2)); //Distance from the edge
    of the area
11 Reff=80/100; //Reflector Efficiency
12
13 Ec= CP/(D^2); //Illumination at the centre
    of the area
14 theta= acosd(D/De); //Angle made between
    perpendicular and edge distances
15 w=2*%pi*(1-cosd(theta)); //Solid angle subtended by
    the area
16
17 //Assuming uniform intensity in all directions

```

```

18
19 //Illumination at the edge is
20 Eue=(CP/(De^2))*cosd(theta);
21 //Flux is given by Iw
22 flux=CP*w;
23 //Average illumination
24 Euavg= flux/(%pi*(R^2));
25 //Average illumination with reflector efficiency
26 Euavgr= Euavg*Reff;
27
28 //When candle power is only vertically downwards
29 Eve=Eue*cosd(theta); //Illumination at the edge
    of the area
30
31 //Total flux equation for non uniform intensity is
    derived and given by 2*pi*I*(sec(theta)-1)
32 fluxn=2*%pi*CP*(secd(theta)-1);
33 //Average illumination
34 Enuavg= fluxn/(%pi*(R^2));
35 //Average illumination with reflector efficiency
36 Enuavgr= Enuavg*Reff;
37
38 printf('The illumination at the \n')
39 printf('i) Centre of area = %g lux\n',Ec)
40 printf('ii) Periphery of the area \n')
41 printf('a) Uniform illumination = %g lux\n',Eue)
42 printf('b) Candle power is vertically downward = %g
    lux\n',Eve)
43 printf('iii) Average illumination and Average
    illumination with reflector efficiency\n')
44 printf('a) Uniform intensity = %g lux and %g lux
    respectively.\n',Euavg, Euavgr)
45 printf('b) Non Uniform intensity = %g lux and %g lux
    respectively.\n',Enuavg, Enuavgr)

```

Scilab code Exa 6.6 To determine illumination at a point when a mirror is used

```
1
2 //To determine illumination at a point when a mirror
   is used
3 //Page 330
4 clc;
5 clear;
6
7 CP=500;           //Candle power of the lamp
8 Vd=5;            //Vertical distance from
   the point to the source
9 Hd=5;           //Horizontal distance from
   the point to the source
10 D=sqrt((Vd^2)+(Hd^2)); //Distance between the
   source and the point
11 Md=2;          //Distance of the source
   from the mirror
12 theta=atand(Hd/Vd); //Angle made between the
   source to point and the horizontal
13
14 //Lets consider the mirror reflection
15 R=80/100;      //Reflected rays ratio
16 Vdm=Vd+(2*Md) //Vertical Distance from
   the point to the source reflection
17 Hdm=Hd;       //Horizontal distance from
   the point to the source reflection
18 Dm=sqrt((Vdm^2)+(Hdm^2)); //Distance between the
   source reflection and the point
19 thetam=atand(Hdm/Vdm); //Angle made between the
   source reflection to point and the horizontal
20
21 //Illumination at A due to source
22 Es=(CP/(D^2))*cosd(theta);
23
24 //Illumination at A due to source refelction
25 Er=(R*CP/(Dm^2))*cosd(thetam);
26
```

```

27 //Total Illumination at point A
28 E=Er+Es;
29
30 printf('The total illumination at point A is %g lux
        .\n',E)

```

Scilab code Exa 6.7 To determine the illumination over a disc

```

1
2 //To determine the illumination over a disc
3 //Page 330
4 clc;
5 clear;
6
7 MSCP=300; //Uniform
            intensity of the lamp or Mean Spherical candle
            power
8 Vd=6; //Vertical
        distance of the lamp from the disc
9 R=6/2; //Radius of the
        disc
10 Reff=60/100; //Reflector
            efficiency
11 D=sqrt((Vd^2)+(R^2)); //Distance from
            the source to the edge of the disc
12 theta=atand(R/Vd); //Angle made
            between D and Vd
13
14 Ecwr=MSCP/(Vd^2); //Illumination at
            the centre without reflector
15 Eewr=(MSCP/(D^2))*cosd(theta); //Illumination at
            the edge without reflector
16
17 //Illumination at the centre is equal to the
            illumination at the egde with a reflector

```

```

18 Eer=MSCP*Reff*(4*pi/(pi*(R^2)));
19 Ecr=Eer;
20
21 w=2*pi*(1-cosd(theta));           //Solid angle made
    by the surface
22 flux=MSCP*w;                       //Total flux
    produced by the source
23 Eavg=flux/(pi*(R^2));              //Average
    illumination
24
25 printf('The illumination at:\n')
26 printf('a) At the centre:\n')
27 printf('i) With reflector : %g lux\n',Ecr)
28 printf('ii) Without reflector : %g lux\n',Eewr)
29 printf('b) At the edge of the surface:\n')
30 printf('i) With reflector : %g lux\n',Eer)
31 printf('ii) Without reflector : %g lux\n',Eewr)
32 printf('c) Average illumination over the disc
    without the reflector : %g lux\n',Eavg)

```

Scilab code Exa 6.8 To estimate the number rating and disposition of the lamps

```

1
2 //To estimate the number rating and disposition of
  the lamps
3 //Page 338
4 clc;
5 clear;
6
7 E=32;                               //Illumination required for
  the working plane
8 A=80*15;                            //Area of the work bench
9 UF=0.5;                              //Utilization Factor
10 MF=0.8;                             //Maintenance Factor
11 SHR=1.5;                             //Maximum permissible value

```

```

    of spacing to height ratio
12 h=4.5; //Required height for the
    lamps to be hung above the work bench
13 Leff=14; //Lamp efficacy
14 Tlumen=E*A; //Total lumens required
15 Llumen=Tlumen/(UF*MF); //Lamp lumens required
16 l=(A/15); //Length of the workspace
17
18 Nc=ceil(1/(1.5*4.5)); //Minimum number of lamps
    in a single row (Number of columns)
19
20 W=200; //Assumed wattage of the
    bulb
21
22 NoL=Llumen/(W*Leff); //Number of bulbs required ,
    Calculated value
23
24 Nr=ceil(NoL/12); //Number of rows calculated
    for the required criteria
25
26 N=Nc*Nr; //Number of lamps necessary.
27
28 Sp=l/Nc; //Length wise spacing
    between the lamps
29 printf('Assuming 200W bulbs , in a rectangular
    workspace of 80m*15m, we require %g bulbs
    arranged in %g rows and %g columns having a
    spacing of %gm between them.\n',N,Nr,Nc,Sp)

```

Scilab code Exa 6.9 To determine the number of lamps required

```

1
2 //To determine the number of lamps required
3 //Page 344
4 clc;

```

```

5  clear;
6
7  A=30*20;           //Area of the building
8  B=25;             //Brightness in lumen/sq.
    metre
9  CoR=0.25;        //Co - efficient of
    reflection
10 Lwatt=500;        //Lamp Wattage
11 Ll=8000;         //Lamp lumens output
12 E=B/CoR;         //Illumination required
13 BF=0.6;          //Beam factor
14 WF=1.2;          //Waste light factor
15 MF=0.75;         //Mainteanance factor
16
17 Tlumen= E*A*WF/MF; //Total lumens required
18
19 Llumen=Ll*BF;    //Lumens provided by one
    lamp
20
21 NoL=Tlumen/Llumen; //Number of lamps
22
23 printf('The number of lamps required are %g.\n',
    round(NoL))

```

Scilab code Exa 6.10 To determine the illumination on the ground

```

1
2 //To determine the illumination on the ground
3 //Page 344
4 clc;
5 clear;
6
7 CP=300;           //Uniform luminous intensity
    of a lamp
8 Vd=6;            //Vertical distance from the

```

```

    ground to the lamp
9 Hd=6; //Horizontal distance from
    the ground to the lamp
10 D=sqrt((Vd^2)+(Hd^2)); //Distance from the lamp to
    point 6 metres away from the vertical line
11 theta=atand(Hd/Vd); //Angle made between D and
    Vd.
12
13 // Illumination
14 Ec=CP/(Vd^2); //Vertically beneath the
    lamp
15 E6=CP*cosd(theta)/(D^2); //^=6 metres away from the
    centre
16
17 printf('The illumination on the ground :\n')
18 printf('a)Vertically beneath the lamp : %g lux.\n',
    Ec)
19 printf('b)6 metres away from condition a) : %g lux.\n
    n',E6)

```

Scilab code Exa 6.11 Determine the output voltage

```

1
2 //Determine the output voltage
3 //Page 344
4 clc;
5 clear;
6
7 Area=(5.08*3.75)*(10^-4); //Area projected by
    the cathode
8 Is=12*(10^-6); //Sensitivity
9 Rl=1.5*(10^6); // Load of operation
10
11 //Function to calculate the output voltage for each
    case

```

```

12 deff('y=volt(a,b)', 'y=Area*a*Is*Rl/(b^2)');
13
14 //Case 1
15 CP=60; //Lamp intensity
16 D=1.8; //Vertical distance
    of the lamp from the cell
17 V1=volt(CP,D);
18 //Case 2
19 CP=6; //Lamp intensity
20 D=0.5; //Vertical distance
    of the lamp from the cell
21 V2=volt(CP,D);
22 //Case 3
23 W=100; //Wattage of the
    lamp
24 eff=20; //efficacy of the
    lamp
25 CP=W*eff/(4*pi); //Lamp intensity
26 D=2; //Vertical distance
    of the lamp from the cell
27 V3=volt(CP,D);
28
29 printf('The voltage output of the cells are :\n')
30 printf('a) 60 CP lamp at 1.8m : %g V\n',V1)
31 printf('b) 6 CP lamp at 0.5m : %g V\n',V2)
32 printf('c) A 100W lamp having a efficacy of 20
    lumens/watt at 2m : %g V\n',V3)

```

Chapter 7

Electric Traction

Scilab code Exa 7.1 To determine the Acceleration and Coasting Period

```
1
2 //To determine the Acceleration and Coasting Period
3 //Page 362
4 clc;
5 clear;
6 D=1.92
7 Ts=20; //Duration of stops
8 Vsch=40;
9 V1=60.8;
10 //Retardation
11 B=3.2;
12 Bc=0.16;
13
14 SchTime=D*3600/Vsch; // Schedule time in seconds
15
16 T=SchTime-Ts; //Time of Travel
17
18 a=poly(0, 'a'); // Acceleration Unknown Value
19
20 V2=(V1-Bc*(T-(V1/a)))/(1-(Bc/B)); // From the Speed
    Time Curve
```

```

21
22 X=((V1+V2)*T-(V1*V2*((1/B)+(1/a))))-(7200*D); //
    Polynomial Equation to find a
23
24 disp(X(2))
25 printf('\n\nThe Above Equation is Equated to zero to
    get a\n\n\n')
26 a=roots(X(2)); //Roots of the Characteristic
    Equation
27 //To Determine the positive roots
28 Y=abs(a)+a;
29 if(Y(1)==0)
30     a=a(2);
31 else
32     a=a(1);
33 end
34
35 Ta=V1/a; //Duration of acceleration
36 V2=(V1-Bc*(T-(V1/a)))/(1-(Bc/B)); // From the Speed
    Time Curve
37 Tcs=(V1-V2)/Bc; //Coasting Time Period
38
39 printf('The Acceleration of the Train is %g Kmphps\n
    ',a)
40 printf(' The Coasting Period is %g seconds\n',Tcs)

```

Scilab code Exa 7.2 To determine the duration of acceleration and others

```

1
2 //To determine the duration of acceleration and
    others
3 //Page 363
4 clc;
5 clear;
6

```

```

7 D=1.6;
8 Vav=40;
9 Bc=0.16;
10 B=3.2;
11 a=2; //Acceleration
12
13 T=D*3600/Vav; //Time of travel
14
15 V1=poly(0, 'V1'); //Variable Value
16
17 V2=(V1-Bc*(T-(V1/a)))/(1-(Bc/B)); // From the Speed
    Time Curve
18 X=((V1+V2)*T-(V1*V2*((1/B)+(1/a))))-(7200*D); //
    Polynomial Equation to find V1
19 V1=roots(X); //Numerical Values for V1
20 disp(X);
21 printf('\n\nThe Roots of the above equation are\n\n');
22 disp(V1)
23
24 V1=V1(2); //Train Considered have speed close to
    this value
25 V2=(V1-Bc*(T-(V1/a)))/(1-(Bc/B)); // From the Speed
    Time Curve
26
27 Ta=V1/a; //Acceleration Time
28 Tc=(V1-V2)/Bc; //Coasting Time
29 Tb=V2/B; //Braking time
30 //Distance are calculated according to the area
    under their respective curves
31 Da=(Ta*V1/2)/3600; //Distance during acceleration
32 Dc=((V1-V2)*Tc/2)+(V2*Tc)/3600; //Distance during
    coasting
33 Db=(Tb*V2/2)/3600; //Distance during braking
34
35 printf('\n\nThe Time periods and the distance
    covered for:\n\n')
36 printf('Acceleration : %g seconds and %g km\n\n',Ta, Da
    )

```

```
37 printf('Coasting : %g seconds and %g km\n',Tc,Dc)
38 printf('Braking : %g seconds and %g km\n',Tb,Db)
```

Scilab code Exa 7.3 To determine the acceleration required to run the service

```
1
2 //To determine the acceleration required to run the
   service
3 //Page 364
4 clc;
5 clear;
6
7 D=1;
8 Ts=20; //Stopping Time
9 B=3;
10 Vsh=30;
11 ShT=D*3600/Vsh; //Schedule time
12 T=ShT-Ts; //Actual Run Time
13 Vav=D*3600/T; //Average Speed
14 Vm=1.25*Vav;
15 a=poly(0,'a'); //Acceleration Variable
16
17 X=((2*Vm*T)-((Vm^2)*((1/a)+(1/B))))-(7200*D); //
   Polynomial Equation to find 'a'
18
19 disp(X(2));
20 printf('\n The above equation is equated to zero to
   find the Acceleration\n\n')
21
22 a=roots(X(2)); //Numerical Value of the Acceleration
23
24 printf('The Acceleration required to run the service
   is %g Kmphps\n',a)
```

Scilab code Exa 7.4 To determine the time taken by the train to attain a particular

```
1
2 //To determine the time taken by the train to attain
   a particular speed
3 //Page 366
4 clc;
5 clear;
6 M=200; //Mass of train
7 g=9.81; //Acceleration due to gravity
8 W=M*g;
9 RI=10/100; //Rotational Inertia
10 We=W*(1+RI);
11 motor=4; //Number of Motors
12 Tm=5000; //Torque per motor
13 T=Tm*motor;
14 V=40; //Speed to be Attained
15 N=0.9; //Gear Efficiency
16 Y=3.6/1; //Gear Ratio
17 R=91.5*(10^-2)/2;
18 Ft=N*T*Y/R;
19 r=40;
20 G=(1/200)*100; //Gradient in Percentage
21 a=poly(0, 'a'); //Acceleration
22
23 X=((28.3*We*a)+(10*W*G)+(M*r))-Ft; //Polynomial
   Equation to find acceleration
24
25 a=roots(X); //Numerical Value of Acceleration
26 Time=V/a; //Time Taken to attain the required the
   necessary speed
27
28 printf('The Time Taken by the Train to attain 40
   Kmphs is %g seconds\n',Time)
```

Scilab code Exa 7.5 To determine the Factors affeting the mechanics of the Train

```
1
2 //To determine the Factors affeting the mechanics of
   the Train
3 //Page 373
4 clc;
5 clear;
6 D=1.92
7 Ts=20; //Duration of stops
8 Vsch=40;
9 V1=60.8;
10 //Retardation
11 B=3.2;
12 Bc=0.16;
13 SchTime=D*3600/Vsch; // Schedule time in seconds
14 T=SchTime-Ts; //Time of Travel
15 a=poly(0, 'a'); // Acceleration Unknown Value
16 V2=(V1-Bc*(T-(V1/a)))/(1-(Bc/B)); // From the Speed
   Time Curve
17 X=((V1+V2)*T-(V1*V2*((1/B)+(1/a))))-(7200*D); //
   Polynomial Equation to find a
18 a=roots(X(2)); //Roots of the Characteristic
   Equation
19 //To Determine the positive roots
20 Y=abs(a)+a;
21 if(Y(1)==0)
22     a=a(2);
23 else
24     a=a(1);
25 end
26 Ta=V1/a; //Duration of acceleration
27 V2=(V1-Bc*(T-(V1/a)))/(1-(Bc/B)); // From the Speed
   Time Curve
```

```

28 Tc=(V1-V2)/Bc; //Coasting Time
29 Tb=V2/B; //Braking time
30 //Distance are calculated according to the area
    under their respective curves
31 Da=(Ta*V1/2)/3600; //Distance during acceleration
32 Dc=((V1-V2)*Tc/2)+(V2*Tc)/3600; //Distance during
    coasting
33 Db=(Tb*V2/2)/3600; //Distance during braking
34
35 r=4.53*9.81; //Train Resistance in N per tonne
36 M=200; //Mass of Train
37 AWF=1.1; //Accelerating Weight Factor
38 SEOA=0.010726*(V1^2)*(AWF)/D; //Specific Energy
    Output during acceleration
39 SEOAr=0.2778*r*Da/D; //Specific Energy Output during
    acceleration against train resistance
40 SEO=SEOA+SEOAr; //Specific Energy for the run
41 SEOB=0.010726*(V2^2)*(AWF)/D; //Specific Energy
    Output during braking
42 SEOC=SEOA-SEOB; //energy utilized during coasting
43 rc=SEOC*D/(0.2778*Dc)//Mean Train Resistance during
    coasting
44
45 printf('\n i) The Specific Energy Output for the Run
    is %g Whr/Tonne-km\n',SEO)
46 printf('\n ii) The Energy Dissipated by the brakes is
    %g Whr/Tonne-km\n',SEOB)
47 printf('\n iii) The Energy Utilized during coasting is
    %g Whr/Tonne-km\n',SEOC)
48 printf('\n iv) The Mean Train Resistance during
    coasting is %g N/tonne\n',rc)

```

Scilab code Exa 7.6 To determine the Specific Energy Output Runs for Both the Case

```

2 //To determine the Specific Energy Output Runs for
   Both the Cases
3 //Page 374
4 clc;
5 clear;
6 G=(1/80)*100;
7 W=210;
8 We=1.1*W;
9 Vm=poly(0, 'Vm'); //Variable Value of The Max
   Velocity
10 D=1.6; //Distance between Stations
11 Ftg=10*W*G; //Tractive Effort due to Gradient
12 Fta=13050; //Total Tractive Effort for Acceleration
13 Ftb=20445; //Total Tractive Effort for Retardation
14 ron=5.43; //Train Resistance when power on
15 rc=6.525; //Train Resistance when coasting
16 //Level Track
17 al=2;
18 Bl=3.2;
19
20 //Up Gradient
21 Vavu=40; //Average Speed
22 Tu=D*3600/Vavu; //Total Time Taken
23 au=(Fta-Ftg)/(28.3*We); //Acceleration
24 Bu=(Ftb+Ftg)/(28.3*We); //Retardation
25 Xu=((2*Vm*Tu)-((Vm^2)*((1/au)+(1/Bu))))-(7200*D); //
   Polynomial Equation to Find Vm
26 Vmu=roots(Xu); //Numerical Value of Vm
27 Zu=Vmu(1)-Vmu(2); //To determine which root to take
28 if(abs(Zu)+Zu==0)
29     Vmu=Vmu(1);
30 else
31     Vmu=Vmu(2);
32 end
33 Tua=Vmu/au; //Accelerating Period
34 Tub=Vmu/Bu; //Braking Period
35 Tucs=Tu-Tua-Tub; //Constant Speed Duration
36 Dua=Vmu*Tua/(2*3600); //Distance Travelled During

```

```

    Accleration in km
37 Dub=Vmu*Tub/(2*3600); //Distance Travelled During
    Braking in km
38 Duon=D-Dub; //Distance run with power on
39 SE0ua=0.010726*(Vmu^2)*(We/W)/D; //Specific Energy
    Output during Acceleration
40 SE0ug=27.25*G*Duon/D; //Specific Energy Output for
    Gradient
41 SE0ur=0.2778*ron*9.81*Duon/D; //Specific Energy
    Output for resitance
42 TSE0u=SE0ua+SE0ug+SE0ur; //Total Specific Output
43
44 //Down Gradient
45 Vavd=44; //Average Speed
46 Td=D*3600/Vavd; //Total Time Taken
47 ad=(Fta+Ftg)/(28.3*We); //Acceleration
48 Bd=(Ftb-Ftg)/(28.3*We); //Retardation
49 Xd=((2*Vm*Td)-((Vm^2)*((1/ad)+(1/Bd))))-(7200*D); //
    Polynomial Equation to Find Vm
50 Vmd=roots(Xd); //Numerical Value of Vm
51 Zd=Vmd(1)-Vmd(2); //To detemine which root to take
52 if(abs(Zd)+Zd==0)
53     Vmd=Vmd(1);
54 else
55     Vmd=Vmd(2);
56 end
57 Tda=Vmd/ad; //Accelerating Period
58 Tdb=Vmd/Bd; //Braking Period
59 Tdcs=Td-Tda-Tdb; //Constant Speed Duration
60 Dda=Vmd*Tda/(2*3600); //Distance Travelled During
    Accleration in km
61 Ddb=Vmd*Tdb/(2*3600); //Distance Travelled During
    Braking in km
62 Ddon=D-Ddb; //Distance run with power on
63 SE0da=0.010726*(Vmd^2)*(We/W)/D; //Specific Energy
    Output during Acceleration
64 //Net Force Acting Downward due to gradient and
    resistance

```

```

65 Fnet=W*((10*G)-rc);
66 //Since Fnet is Postive , To run the train at
    constant speed brakes will have to applied
    therefore they cannot supply electric energy
67 TSE0d=SE0da; //Total Sepcific Enegy Consumption
68
69 printf('The Total Specific Energy:\n')
70 printf('Up Gradient : %g Whr/Tonne km\n',TSE0u)
71 printf('Down Gradient : %g Whr/Tonne km\n',TSE0d)

```

Scilab code Exa 7.7 To determine the specific energy consumption

```

1
2 //To determine the specific energy consumption
3 //Page 376
4 clc;
5 clear;
6 W=400;
7 G=1;
8 a=1.5; //Acceleration
9 Ta=30; //Acceleration Time
10 Tf=36; //Free running Period
11 Tc=25; //Coasting Period
12 B=2.6;
13 r=45;
14 RI=10/100; //Rotational Inertia Effect
15 Eff=75/100; // Overall Efficieny
16 g=9.81; //Accleration due to gravity
17
18 Vm=Ta*a;
19 We=W*(1+RI);
20
21 //Distance Covered is equal to the area under the
    speed time curve
22 Da=Vm*Ta/(2*3600); //Acceleration

```

```

23 Df=Vm*Tf/3600; //Free Run
24
25 Ftf=W*(r+(10*g*G)); //Tractive effort during free
    run period
26
27 Fta=We*277.8; //Tractive effort due to acceleration
28
29 //During Coasting, The Accelrating force is equal to
    Tractive effort during free run
30 //Retardation due to coasting is
31 Bc=Ftf/Fta;
32
33 V2=Vm-(Bc*Tc); //Speed of train after the coasting
    period
34 Tb=V2/B; // Braking period
35
36 Dc=(((Vm-V2)*Tc)/2)+(V2*Tc))/3600; //Distance
    covered during coasting
37 Db=V2*Tb/(2*3600); //Distance covered during braking
38
39 D=Da+Df+Dc+Db; //Total Distance
40
41 D1=Da+Df; //Distance for which the energy is being
    spent
42
43 SEO=(0.010726*(Vm^2)*We/(W*D))+(27.25*G*D1/D)
    +(0.2778*r*D1/D); //Specific Energy Output
44 //Note the Calculated Specific Energy Output during
    accelration is wrong.
45
46 SEOa=SEO/Eff; //Actual Specific Energy Output with
    75% efficieny
47
48 printf('The Specific Energy Consumption is %g Whr/
    Tonne Km\n',SEOa)

```

Scilab code Exa 7.8 To determine the weight of the locomotive

```
1
2 //To determine the weight of the locomotive
3 //Page 378
4 clc;
5 clear;
6 Wg=300; //Weight of the train to be hauled
7 RI=10/100; //Rotation inertia
8 Ma=20/100; //Co-Efficient of adhesion
9 Wa=20; //Permissible Weight of axle load
10 r=45;
11 G=2;
12 a=1; //Acceleration
13 Wl=poly(0, 'Wl'); //Variable Weight of locomotive
14 W=Wg+Wl; //Total Weight of the train
15 We=W*(1+RI);
16 Ft=((277.8*a*We/W)+(98.1*G)+r)*W;
17
18 Fmax=(9.81*1000*Ma*Wl);
19
20 X=Ft-Fmax; //Polynomial Equation to find Wl
21
22 Wl=roots(X(2)); //Numerical Value of the Weight of
    the locomotive
23
24 NoA=Wl/Wa; //Number of axles
25
26 printf('The weight of the locomotive and the number
    of axles is %g tonnes and %g axles respectively\n
    ',Wl,ceil(NoA))
```

Scilab code Exa 7.9 To deduce the speed current characteristic of the motor

```
1
2 //To deduce the speed current characteristic of the
   motor
3 //Page 380
4 clc;
5 clear;
6 Ia=[10,20,30,40,50,60,70]; //Current
7 T=[45,130,230,350,470,610,765]; //Torque
8 V=440; //operating voltage
9 r=0.5; //Armature Circuit Resistance
10
11 N=9.55*(V-(Ia*r))./(T./Ia); //Speed
12
13 printf('For the Given Current:\n')
14 disp(Ia);
15 printf(' \n\nThe Speeds tabulated are:\n')
16 disp(N);
```

Scilab code Exa 7.10 To determine the speed torque curve for the series motor

```
1
2 //To determine the speed torque curve for the series
   motor
3 //Page 381
4 clc;
5 clear;
6 I=[50,100,150,200,250,300]; //Field Current
7 Eb1=[230,360,440,500,530,580]; //Armature Volts
8 Ra=0.07; //Armature Winding Resistance
9 Rf=0.05; //Field Resistance
10 Rt=Rf+Ra; //Total Resistance
```

```

11 P=4; //Poles of the machine
12 N1=600; //Series Motor Speed
13 Vc=600; //Constant Operating Voltage
14
15 // Note that the Suffix 1 and 2 have given according
    the question; 1 stands for the case where the
    magnetisation curve has been given and 2 stands
    for the case where we have to find the speed
    torque curve
16
17 Eb2=Vc-(I.*(Rt));
18 N2=N1.*Eb2./Eb1; //Speed
19
20 T=9.55.*Eb2.*I./N2; //Torque
21
22 printf('The Tabulated Speeds for 600V are:\n')
23 disp(N2)
24 printf('\nAnd their corresponding torques are:\n')
25 disp(T)
26
27 plot(T,N2)
28 xlabel('Torque')
29 ylabel('Speed')

```

Scilab code Exa 7.11 To determine the horse power delivered by the locomotive

```

1
2 //To determine the horse power delivered by the
    locomotive
3 //Page 388
4 clc;
5 clear;
6
7 Ft1=35000; //Tractive Effort on a level surface
8 Ftg=55000; //Tractive Effort on a gradient

```

```

 9 V=50; //Speed of the train
10 HP=735.5; //One Horse Power
11 P1=Ft1*V*1000/(3600*HP); //Power Output(in HP) of
    the Locomotive on the level track
12
13 //CASE 1:
14 //D.C Series Motor, Power is directly proportional
    to the root of the Tractive Effort
15 HP1=P1*sqrt(Ftg/Ft1);
16
17 //CASE 2:
18 //3 Phase Induction Motor, Power is directly
    proportional to the Tractive Effort
19 HP2=P1*(Ftg/Ft1);
20
21 printf('The Horse Power delivered by the locomotive
    when the motors used are:\n')
22 printf('i) D.C Series Motor is %g H.P \n',HP1)
23 printf('ii) 3 Phase Induction Motor is %g H.P \n',
    HP2)
24 //Note the Calculation Mistake in TextBook for case
    2

```

Scilab code Exa 7.12 To Determine the tractive effort shared by the two locomotive

```

1
2 //To Determine the tractive effort shared by the two
    locomotives
3 //Page 388
4 clc;
5 clear;
6
7 Da=1.27; //Loco A wheel diameter
8 Db=1.244; //Loco B wheel diameter
9 S=5; //Slip of both the Locomotives in Case 1

```

```

10 Sa=S; //Slip of Loco A in Case 2
11 Sb=4; //Slip of Loco B in Case 2
12 Ftta=11325; //Total Tractive Effort in Case 1
13 Fttb=1309; //Total Tractive Effort in Case 2
14 Ft=5227; //Full Load Tractive Effort of Loco
15
16 x=poly(0,'x'); //Variable for Tractive Effort
    Exerted by Loco A
17
18 //Slips are same
19 //For 11325 kg
20 //% Speed
21 NA=100-(S*x/Ft);
22 NB=100-(S*(Ftta-x)/Ft);
23 Rna=NB/NA; //Speed Ratio
24
25 Y1=Rna-(Da/Db); //Polynomial to find out 'x'
26 x=roots(Y1(2)); //Numerical Value of Tractive Effort
    shared by A
27
28 //Tractive Efforts Shared by A Loco and B Loco
29 FtAas=x;
30 FtBas=Ftta-x;
31
32 //For 1309 kg
33 //% Speed
34 x=poly(0,'x'); //Variable for Tractive Effort
    Exerted by Loco A
35 NA=100-(S*x/Ft);
36 NB=100-(S*(Fttb-x)/Ft);
37 Rn1=NB/NA; //Speed Ratio
38
39 Y2=Rn1-(Da/Db); //Polynomial to find out 'x'
40 x=roots(Y2(2)); //Numerical Value of Tractive Effort
    shared by A
41
42 //Tractive Efforts Shared by A Loco and B Loco
43 FtAbs=x;

```

```

44 FtBbs=Fttb-x;
45
46 //Different Slips
47 //For 11325 kg
48 //% Speed
49 x=poly(0,'x'); //Variable for Tractive Effort
    Exerted by Loco A
50 NA=100-(Sa*x/Ft);
51 NB=100-(Sb*(Ftta-x)/Ft);
52 Rna=NB/NA; //Speed Ratio
53
54 Y3=Rna-(Da/Db); //Polynomial to find out 'x'
55 x=roots(Y3(2)); //Numerical Value of Tractive Effort
    shared by A
56
57 //Tractive Efforts Shared by A Loco and B Loco
58 FtAad=x;
59 FtBad=Ftta-x;
60
61 //For 1309 kg
62 //% Speed
63 x=poly(0,'x'); //Variable for Tractive Effort
    Exerted by Loco A
64 NA=100-(Sa*x/Ft);
65 NB=100-(Sb*(Fttb-x)/Ft);
66 Rn2=NB/NA; //Speed Ratio
67
68 Y4=Rn2-(Da/Db); //Polynomial to find out 'x'
69 x=roots(Y4(2)); //Numerical Value of Tractive Effort
    shared by A
70
71 //Tractive Efforts Shared by A Loco and B Loco
72 FtAbd=x;
73 FtBbd=Fttb-x;
74
75 printf('The Tractive Effort shared A and B for:\n \n
    ')
76 printf('i) Slips are %g percent for both the

```

```

    Locomotives for a Tractive effort of:\n',S)
77 printf('a) %g kg : %g kg by A and %g kg by B
    respectively\n',Ftta,FtAas,FtBas)
78 printf('b) %g kg : %g kg by A and %g kg by B
    respectively\n\n',Fttb,FtAbs,FtBbs)
79 printf('i) Slips are %g percent for A and %g percent
    for B for a Tractive effort of:\n',Sa,Sb)
80 printf('a) %g kg : %g kg by A and %g kg by B
    respectively\n',Ftta,FtAad,FtBad)
81 printf('b) %g kg : %g kg by A and %g kg by B
    respectively\n\n',Fttb,FtAbd,FtBbd)
82
83 //Please Note there is caluculation mistake
    calculation in the TextBook for the First Case

```

Scilab code Exa 7.13 To determine the traction control of 2 motors rated at 1500V

```

1
2 //To determine the traction control of 2 motors
    rated at 1500V
3 //Page 396
4 clc;
5 clear;
6
7 NoM=2; //Number of motors
8 V=1500; //Rated Voltage
9 I=500; //Starting Current
10 R=0.15; //Armature Resistance
11 r=50; //Specific Resistance
12 W=120;
13 We=140;
14 Ft=38000; // Tractive Effort Per motor
15
16 Vs=40; //Speed at the end of starting period
17 //Note the question is asked for 50 Kmph, But the

```

```

    answer is calculated for 40Kmph
18 //Therefore it is calculated for 40 Kmph
19 //By Changing the value of Vs to 50 Kmph, The
    Specific Parameters can also be obtained
20
21 a=poly(0, 'a'); //Accleration Variable
22
23 X=(2*Ft) - ((277.8*We*a)+(W*r)); //Polynomial To find
    'a'
24
25 a=roots(X); //To find the Numerical Value of the
    Acceleration
26
27 t=Vs/a; //Starting Time
28
29 Vd=I*R; //Resistance drop per motor
30
31 ts=t*(V-(2*Vd))/(2*(V-Vd)); //Starting Series Time
    Period
32 tp=t-ts; //Starting Parallel Time Period
33
34 Vt=a*ts; //Speed at Transition
35 //Loss can be found out by the computing the area
    under the of the given figure in the text book
    for different period
36 //In series , the Voltage reduces , Hence 675;
37 Rhe1=(((V/2)-Vd)*I*ts)+((V/2)*I*tp))/(2*3600*1000);
    //Loss per motor in kWhr
38 Rhe=NoM*Rhe1; //Total Rheostatic Loss
39
40 printf('i) The Duration of starting period is %g
    seconds\n',t)
41 printf('ii)The Speed of Train at Transition is %g
    Kmph\n',Vt)
42 printf('iii) The Rheostatic Loss is %g kWhr\n',Rhe)

```

Scilab code Exa 7.14 To determine the speed of motors when connected in series

```
1
2 //To determine the speed of motors when connected in
   series
3 //Page 397
4 clc;
5 clear;
6 V=650; //Rating of motor and Supply Voltage when
   connected in parallel
7 IR=10*650/100; //Armature Drop
8 N1p=1000; //Speed of the first motor in parallel
   operation
9 D1=88; //Motor 1 wheels diameter
10 D2=86; //Motor 2 wheels diameter
11
12 //Current Remains Constant during the start
13
14 r=D1/D2; //Ratio of the first motor wheel diameter
   to the second motor wheel diameter
15
16 //N1/N2 = 1/r
17
18 V1=((V-IR)/(1+(r)))+(IR)/(1+(1/r)); //Voltage of
   Motor 1 in series operation
19 V2=V-V1; //Voltage of Motor 2 in series operation
20
21 N1=N1p*(V1-IR)/(V-IR); //Speed of motor 1 in series
   configuration
22 N2=N1*r; //Speed of motor 2 in series configuration
23
24 printf('The Speed of Motor 1 in series Configuration
   is %g rpm\n',N1 )
25 printf('The Speed of Motor 2 in series Configuration
```

is %g rpm\n',N2)

Scilab code Exa 7.15 To determine the characteristics gear ratio and wheels are re

```
1
2 //To determine the characteristics gear ratio and
   wheels are replaced
3 //Page 397
4 clc;
5 clear;
6
7 D1=86; //Diameter of wheels in Case 1
8 Y1=71/21; //Gear Ratio in Case 1
9 D2=88; // Diameter of wheels in Case 2
10 Y2=74/19; //Gear Ratio in case 2
11 I=[50,100,150,200,250,300]; //Current
12 L1=[80,50,45,40,36,32]; //Speed in Kmph
13 Ft1=[2000,6000,12000,16000,18000,22000]; //Tractive
   Effort
14
15 //  $V = \pi * D * N * 60 / (100 * 100 * Y)$ 
16 //According to the above equation, V is directly
   proportional to  $D * N / Y$ 
17
18 //Angular Frequency And Torque remains the same
   irrespective of the gear ratio or change in
   diameter
19
20 //Hence the V is directly proportional to  $D / Y$ 
21
22 //V is directly proportional to N
23 // N directly proportional to Linear speed
24 //Hence relating the above terms
25 //We get linear speed directly proportional to  $D / Y$ 
26
```

```

27 L2=L1.*D2*Y1/(D1*Y2); //Linear Speed in Case two
28
29 //Similarly T = Ft * D/(200)
30 //Dividing by Y ; (T/Y) = Ft*D/(200*Y)
31 //Taking two cases where we find Torque per gear
    ratio of one case
32 // And multiply the Torque to Gear ratio to the
    other gear ratio to get Torque
33 // From this process we T1*Y2/Y1 =T2
34 // We Get Ft is directly proportional to Y/D
35
36 Ft2=Ft1.*D1*Y2/(D2*Y1); //Tractive Effort in Second
    Case
37
38 printf('The New Characteristics are\n Current in
    Amperes\n')
39 disp(I)
40 printf('\nSpeed in Kmph\n')
41 disp(L2)
42 printf('\nTractive Effort in N\n')
43 disp(Ft2)

```

Scilab code Exa 7.16 To Determine the Resistance of the Various Steps in a Series

```

1
2 //To Determine the Resistance of the Various Steps
    in a Series Motor
3 //Page 398
4 clc;
5 clear;
6 P=20*735.5; //Power Rating
7 V=500; //Rated Voltage
8 Eff=80/100; //Efficiency
9 Raw=1; //Resistance of Armature and Windings
10 Iflux=10/100; //Increase in Flux

```

```

11
12 I=P/(V*Eff); //Maximum Current
13
14 //VAriation in Current
15 Imin=1.5*I;
16 Imax=2*I;
17 Rs=[]; //Total Resistance after each step
18 Rg=[]; //Step Resistance
19 FluxRatio=1+(Iflux); //Flux raises by 10% for every
    step
20
21 //Condition to make sure Imax doesnt go less than
    the Rated Value and also to find the number of
    steps and Step Resistances
22 for i= 1:1000
23     Is=V/Raw;
24     M=Imax-Is;
25     if((abs(M)+M)==0)
26         c=i; //Number of steps + 1
27         R=poly(0,'R'); //Variable Resitance
28         X=((V-(Imax*Raw))/(V-(Imin*R)))-FluxRatio; //
            Polynomial To Find The Next Resistance
29         Rs(i)=roots(X(2)); //Total Next Resistance
30         Rg(i)=Rs(i)-Raw; //Resistance of the ith Step
31         Raw=Rs(i);
32     else
33         break;
34     end
35 end
36
37 S=c-1; //Number Of steps
38 printf('\n\nThe Number of steps is %g\n\n\n',S)
39 for j=1:S
40     printf('The Resistance of ''%g'' step is %g ohm\
        n',j,Rg(j))
41 end

```

Scilab code Exa 7.17 To determine the characteristics for tapping of 30 percent of

```
1
2 //To determine the characteristics for tapping of 30
   percent of the turns
3 //Page 400
4 clc;
5 clear;
6 I=[50,100,150,200,250]; //Current
7 N1=[73.5,48,41,37,35]; //Speed
8 Ft1=[131,457,810,1163,1525]; //Tractive Effort in Kg
9 F1=70/100; //Effective Tapping of the Turns (30%
   reduction)
10
11 //Consider flux corresponding to 250A as 100%, that
   is for 35 Kmph
12
13 Flux=(35*100)./N1; //Flux Percent for Current Value
14
15 //Reduction by 30% of turns is same as reducing the
   Ampere Turns by 30% which is same as reducing the
   Current by 30%
16
17 RI=F1.*I; //Reduced Current
18 RFlux=[32,57,74,83,90]; //Reduced flux determined
   From the Graph of %Flux and Current in the
   TextBook
19
20 N2=(N1.*Flux)./RFlux; //Speed for the new case
21
22 //Tractive Effort is directly proportional to the
   product of flux and current
23
24 Ft2=RFlux.*Ft1./Flux; //Tractive Effort of the
```

Second Case

```
25
26 printf('The Current in Amperes:\n')
27 disp(I)
28 printf('\nThe Speed in Kmph\n')
29 disp(N2)
30 printf('\nThe Tractive Effort in Kg\n')
31 disp(Ft2)
```

Scilab code Exa 7.18 To determine the power fed into the supply system

```
1
2 //To determine the power fed into the supply system
3 //Page 410
4 clc;
5 clear;
6 W=200;
7 RI=10/100; //Rotational Intertia
8 Eff=75/100; //Effeciency of Conversion
9 G=2;
10 V1=60;
11 V2=20;
12 D=4000; //Distance covered during the speed change
    in m
13 r=40;
14 We=(1+RI)*W;
15
16 Ftb=W*((98.1*G)-r); //Tractive Effort during
    Retardation
17
18 KE=0.01072*(We/W)*200*((V1^2)-(V2^2))/1000; //K.E(in
    kWhr) Available due to change in speed
19
20 EquiEE=Ftb*D/(3600*1000); //Equivalent Electrical
    Energy
```

```

21
22 TE=KE+EquiEE; //Total Energy Regenarated
23
24 ERS=Eff*TE; //Energy Returned to the Supply
25
26 B=((V1^2)-(V2^2))/(2*D*3600); //Retardation
27 Tb=V1-V2/B; //time taken during retardation
28
29 PRS=ERS*3600/Tb; //Power Returned due to Retardation
30
31 //If there is no change in speed, Net Reactive force
    is Tractive Effort due to retardation
32
33 Vm=V1; //Maximum Speed
34
35 //Power that Can be fed into the system
36 PFS=0.278*Ftb*Vm/1000; //In Kilowatts
37
38 APFS=Eff*PFS; //Actual Power Fed Back into the
    supply system
39
40 printf('The Power Fed Back into the supply system is
    %g kW\n',APFS)

```

Scilab code Exa 7.19 To determine the value of resistance to be connected in the m

```

1
2 //To determine the value of resistance to be
    connected in the motor circuit
3 //Page 411
4 clc;
5 clear;
6
7 N1=600; //Intial Speed
8 E1=[252,522,750,900,951]; // Voltages At 600 RPM

```

```

 9 Ia=[20,40,60,80,100]; // Field Current
10 N2=400; //Changed Speed
11 E2=E1.*(N2/N1); // Voltages at 400 RPM
12 T=350;
13 Raw = 0.05; //Armature and Winding Resistance
14 printf('Field Current in A:\n')
15 disp(Ia)
16 printf('\nThe Respective Voltages at 400 rpm (V):\n'
    )
17 disp(E2)
18
19 P=E2.*Ia; //To find the value of E and Ia for the
    Required Speed
20
21 PI=2*%pi*N2*T/60; //Power Input
22
23 //To find the Portion of the Curve EIa vs Ia , under
    the current limits where our Power Input is
    located
24 for i = 1:5
25     X=P(i)-PI; //Difference of Powers
26     if(abs(X)+X==0)
27         continue;
28     else
29         L=i-1; //Lower Limit
30         H=i; // Upper limit
31         break
32     end
33 end
34
35 Ip=Ia(H)-Ia(L); //To find the Current Period
36 Im=Ia(L); //Starting Value of Current in the
    Particular Portion of the Curve
37
38 I=Im+(((PI-P(L))/(P(H)-P(L)))*Ip); //Current
    Required for 400 rpm
39 R=PI/(I^2); //Total Resistance required in the
    circuit

```

```
40 Re=R-Raw; //External Resistance
41
42 printf('\n\nThe External Circuit that needs to
        connected in the motor circuit to limit the speed
        to 400 rpm is %g Ohm\n',Re)
```

Chapter 8

Electrolytic Process

Scilab code Exa 8.1 To determine the quantity of current

```
1
2 //To determine the quantity of current
3 //Page 450
4 clc;
5 clear;
6
7 //Parameters of the Copper Plating
8 TSA=0.3716; //Total Surface Area
9 t=0.0254*(10^-3); //Thickness of the plating
10 density=8.883*(10^3); //Density of Copper
11
12 W=TSA*t*density*1000; //Weight of copper deposited
   in gms
13 ECEcu=0.329; //Electrochemical Equivalent of copper
   in gm per coulomb
14 C=W*1000/ECEcu; //Quantity of Current required.
15
16 F=96500; //One Farad Charge
17
18 Q=C/F; //Quantity of Current in Faradays
19
```

```
20 printf('The Quantity of Current Required is %g F',Q)
```

Scilab code Exa 8.2 To determine the Electrochemical Equivalent of Silver

```
1
2 //To determine the Electrochemical Equivalent of
  Silver
3 //Page 456
4 clc;
5 clear;
6
7 I=20; //Current Passed
8 W=26.84; //amount of silver deposited in gms
9 t=20*60; //Time Period in seconds
10
11 TC=I*t; //Total Charge in the given time period
12
13 ECEAg=W*1000/TC; //Electrochemical Equivalent of
  Silver
14
15 printf('The Electrochemical Equivalent of Silver is
  %g mg/C\n',ECEAg)
```

Scilab code Exa 8.3 To Determine the Annual output of copper

```
1
2 //To Determine the Annual output of copper
3 //Page 456
4 clc;
5 clear;
6 I=2000; //Current Passed
7 NW=52; //Number of weeks in a Year
8 T=100*3600; //Number of seconds per week
```

```

9 TC=NW*T*I; //Total Charge supplied all over the year
10 ECu=31.8; //Equivalent Weight of Copper in grams
11 F=96500; //One Farad of Charge
12
13 // 1 F of charge gives 31.8 gms of copper
14
15 W=(TC/F)*ECu/(1000*1000); //Weight of copper in
    tonnes
16
17 printf('The Annual Output of Copper is %g Tonnes\n',
    W)

```

Scilab code Exa 8.4 To Calculate the Energy usedd in producing chemical action dev

```

1
2 //To Calculate the Energy usedd in producing
    chemical action developed in an electrolytic cell
3 //Page 456
4 clc;
5 clear;
6
7 T=15/60; //Time Period of Operation in Hours
8 I=100; //Current in Amperes
9 V=15; //Potential Difference
10 R=0.05; //Resistance of Solution
11
12 NetEP=V-(I*R); //Net Electrode Potential
13
14 //Energy Equation, E = VI t.
15 Energy=NetEP*I*T/1000; // Energy in KiloWatts
16 printf('The Energy used in producing the chemical
    action developed in an electrolytic Cell for 15
    mins is %g kWhr\n',Energy)

```

Scilab code Exa 8.5 To determine the weights of Nickel and Silver Deposited

```
1
2 //To determine the weights of Nickel and Silver
   Deposited
3 //Page 456
4 clc;
5 clear;
6
7 F=96500; //One Farad of Charge
8 //Equivalent Weights of the Following metals
9 EWAg= 108; //silver
10 EWNi= 58.6/2; //Nickel
11
12 I=20; //Current Passed
13 T=1*60*60; //Time for which the current is passed
14 TC=T*I; //Total Charge produced
15
16 //One Equivalent of Metal requires 1 F of charge
17
18 Q=TC/F; // Total Charge in Farad
19
20 //Amount of metal deposited
21 WAg=Q*EWAg; //Silver
22 WNi=Q*EWNi; //Nickel
23
24 printf('The Weight of Nickel And Silver deposited by
   20A for an hour is %g gm and %g gm respectively
   .\n',WNi,WAg)
```
