Scilab Textbook Companion for Generation Distribution and Utilization Of Electrical of Energy by C. L. Wadhwa[1]

Created by
Deeraj A
B.Tech, Second Year
Electronics Engineering
NIT, Karnataka
College Teacher
NA
Cross-Checked by
Prof. Singla

July 17, 2017

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

Book Description

Title: 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

Chapter 2

Economics Of Generation

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

```

//To Determine the Demand and Supply Parameters for
    15 bulbs
//Page 73
clc;
clear;
W=60; //Wattage of the bulb
N=15; //No. of bulbs
CL=W*N; //Connected Load;
Wih=2*(10^3); //Wattage of immersion heater
Wh=2*(10^3); //Wattage of heater

//Usage of Bulbs at different time periods
N1=5;
N2=10;
N3=6

//Time periods for bulbs
T1=2; //6pm - 8pm
T2=2; //8pm - 10pm
T3=2; //10pm - 12pm
```

```
//Time Periods for heaters
T4=4; //1pm - 5pm
T5=3; //8pm - 11pm

//CASE 1
MD1=W*N2; //Maximum Demand
DF=MD1*100/CL; //Demand Factor
EC1=(N1*W*T1)+(N2*W*T2)+(N3*W*T3); //Energy Consumed
DLF1=EC1*100/(24*MD1); //Daily Load Factor

//CASE 2
MD2=(W*N2)+Wh; //From 8pm - 10pm
EC2=(T4*Wih)+(T5*Wh)+EC1; //Energy Consumed
DLF2=EC2*100/(24*MD2); //Daily Load Factor

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)
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

```

//To determine the Demand and supply parameter of
   four consumers
//Page 74
clc;
clear;

//Maximum Demands of various users
MD1=2*(10^3); //9pm
MD2=2*(10^3); //12 noon
MD3=8*(10^3); //5pm
```

```
MD4=4*(10^3); //8pm
MDT=MD1+MD2+MD3+MD4; //Sum of all Maximum Demands

//Demands of various users
D1=1.6*(10^3); //8pm
D2=1*(10^3); //8pm
D3=5*(10^3); //8pm

//The Number after the Alphabets represents the
   Consumer

//Maximum Demand of the System arises at 8.00 PM
MDS = D1+D2+D3+MD4;

TDF=MDT/MDS; //Diversity Factor
//Given Values
//Average Loads
AL2=500;
AL4=1000;
//Load Factors
LF1=15/100;
LF3=25/100;
//Calculated Values
//Average Loads
AL1=LF1*MD1;
AL3=LF3*MD3;
//Load Factors
LF2=AL2*100/MD2;
LF4=AL4*100/MD4;

ALS=AL1+AL2+AL3+AL4;  //Combined Average Loads
LFS=ALS*100/MDS; //Combined Load Factor

printf('i) The Diversity Factor is %g\n',TDF)
printf('ii) The Average load and Load factor of:\n')
printf(' Consumer 1 : %g W and %g percent\n',AL1,LF1
   *100)
printf(' Consumer 2 : %g W and %g percent\n',AL2,LF2
```

```
)
printf(' Consumer 3 : %g W and %g percent\n',AL3,LF3
    *100)
printf(' Consumer 4 : %g W and %g percent\n',AL4,LF4
    )
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

```

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

```
TotFCC=TotFCS+DFC; //Total

AMDS= DFT*MDGS/1000; //Aggregate of Maximum Demand
    at Supply
AMDC= DFD*AMDS; //Aggregate of Maximum Demand for
    Consumers

FCS=TotFCS/AMDS; //Fixed Charges Per KW at
    substation
CES=GCCU/Teff; //Cost of energy at the substation

FCC=TotFCC/AMDC; //Fixed Charges per KW at the
    consumer premises
CEC=CES/Deff; //Cost of Energy at the consumer
    premises

printf('The Yealy Cost per KW demand and the cost
    per KWhr at:\n')
printf('a) The substation is %g rupees per KW and %g
     paise per KWhr\n',FCS,CES*100)
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

```

//To determine the Load factor and suitable units
    for 24 hr operation of the plant
//Page 78
clc;
clear;

//Demands at Various Time Periods starting from 12PM
     to 12PM
D1=500*(10^3);
```

```
D2=800*(10^3);
D3=2000*(10^3);
D4=1000*(10^3);
D5=2500*(10^3);
D6=2000*(10^3);
D7=1500*(10^3);
D8=1000*(10^3);

MD=D5; //Maximum Demand
//Time Periods of demands ststing from 12PM
T1=5;
T2=5;
T3=2;
T4=2;
T5=3;
T6=3;
T7=2;
T8=2;

//Total Energy Demand in 24hrs
TED=(T1*D1)+(T2*D2)+(T3*D3)+(D4*T4)+(T5*D5)+(D6*T6)
   +(D7*T7)+(T8*D8);

LF=TED*100/(24*MD);

printf('Since Maximum Demand is 2500 kW, 2 units 0f
   1000W and one unit of 500W is required. Also for
   continuity of supply, A reserve of 1000W unit is
   required.\n')

C1000=3*1000*(10^3); //1000 unit
C500=1*500*(10^3); //500 Unit

TCP=C1000+C500; //Total capacity of the plant
PCF=TED*100/(24*TCP); //Plant Capacity Factor

//Operating Schedule, Units operated can be seen in
   the textbook
```

```
G1=500*(10^3);
G2=1000*(10^3);
G3=2000*(10^3);
G4=1000*(10^3);
G5=2500*(10^3);
G6=2000*(10^3);
G7=1500*(10^3);
G8=1000*(10^3);

TEG=(T1*G1)+(T2*G2)+(T3*G3)+(G4*T4)+(T5*G5)+(G6*T6)
   +(G7*T7)+(T8*G8);//Total Energy Generated
PUF=TED*100/(TEG); //Plant Use Factor

printf('a) The Reserve Capacity is a 1000kW Unit and
    Load Factor is %g percent\n',LF)
printf('b) The Plant Capacity Factor is %g percent\n
   ',PCF)
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

```

//To determine the Plant use factore of each unit
//Page 80
clc;
clear;

MDS=25*(10^6); //Maximum Demand on the System
U1=15*(10^6); //Load Supplied By Unit 1
U2=12.5*(10^6); //Load Supplied By Unit 2
//Running Time Factor of the Unit
T1=1;
T2=40/100;

```

```
//Energy generated by each unit
E1=1*(10^8);
E2=1*(10^7);
Et=E1+E2; //Total Energy

//Maximum Demands on Each Units
MD1=U1;
MD2=MDS-U1;

//Annual Load Factor for the Units
ALF1=E1*1000*100/(MD1*8760);
ALF2=E2*1000*100/(MD2*8760);

LF2=E2*1000*100/(MD2*0.4*8760); //Load Factor for
    the it is loaded

//Since Unit runs all through the year without any
    reserve, Load Factor ,Plant Use Factor and Plant
    Capacity Factor are the same
PUF1=ALF1; //Plant Use Factor
PCF1=ALF1; // Plant Capacity Factor

PCF2=E2*1000*100/(U2*8760); //Plant Capacity Factor
    for Unit 2
PUF2=E2*1000*100/(U2*0.4*8760);//Plant Use Factor
    for Unit 2

LFP=Et*100*1000/(MDS*8760); //Annual Load Factor of
    the Complete Plant

printf('The Load Factor, Plant Capacity Factor,
    Plant Use Factor of:\n')
printf('Unit 1 : %g percent, %g percent, %g percent\
    n',ALF1,PCF1,PUF1)
printf('Unit 2 : %g percent, %g percent, %g percent\
    n',ALF2,PCF2,PUF2)
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

```
//To determine the Overall cost per kWhr
//Page 82
clc;
clear;

// C1 =(100,000 rupees + 100 rupees/kW + 6 paise /
   kWhr) //Base Load Station
// C2 =(80,000 rupees + 60 rupees/kW + 8 paise /kWhr
   ) //Peak Load Station

MaxD=15*(10^6);
MinD=5*(10^6);

deff('a=BLS(b,c)','a=100000+(b*100)+((6/100)*c)');
   //Function to Find Annual Cost of Base Load
   Station.


deff('a=PLS(b,c)','a=80000+(b*60)+((8/100)*c)'); //
   Function to Find Annual Cost of Peak Load Station
   .

//Also from the Annual Cost characteristics of the
   base load and peak load station we find co-
   efficients which are helpful in computing the
   number of hours a peak load station must work for
    minimum cost

a1=100;
a2=60;
b1=6/100;
```

```
b2=8/100;

Tpeak=(a1-a2)/(b2-b1); //Number of hours the peak plant should operate
//From the straight line annual load duration curve, Maximum Demand at Peak Load Station can be calculated
MDP=Tpeak*(MaxD-MinD)/8760;

TotEG=(MaxD+MinD)*8760/(2*1000); //Total Energy Generated
EGP=MDP*Tpeak/(2*1000); //Energy Generated by Peak Load
EGB=TotEG-EGP; //Energy Generated by Base Load

MDB=MaxD-MDP; //Maximum Demand at the Base Load

//Total Annual Cost of Base Load and Peak Load Stations Respectively
C1=BLS((MDB/1000),EGB);
C2=PLS((MDP/1000),EGP);

TotC=C1+C2; //Total Cost of both the plants
CE=TotC*100/TotEG; //Cost of energy in paise per kWhr

printf('The Operating Scedule of Peak Load Station for Minimum Annual Cost is %g hours\n',Tpeak)
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

```

//To determine the amount saved to replace the
```

```
equipment
//Page 90
clc;
clear;

Pc=80000; //Plant Cost
UL= 15; //Useful Life of the Plant
SVE=5000; //Salvage Value of the Equipment
r=5/100; //Compound Interest Rate

A1=(Pc-SVE)/UL; //Annual Amount to be saved using straight line method
A2=(Pc-SVE)*r*100/(100*(((1+r)^UL)-1))// Annual Amount to be saaved using Sink Fund Method

printf('i) The Amount to be Saved Annually according to straight line method is %g Rupees\n',A1)
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`

```

//To Obatin a two part tariff for the consumers
//Page 90
clc;
clear;

EG=390*(10^6); //Energy Generated in kWhr
MD=130*(10^6); //Maximum Demand of the Supply
SCeff=90/100; //The Amount of energy transferred from Substation to Consumer
//Total Cost for Each Division
Fuel=5*(10^6);
Gen=2.4*(10^6);
```

```
Trans=5*(10^6);
Dist=3.4*(10^6);
Tot=Fuel+Gen+Trans+Dist; //Total Cost

deff('x=Runcost(y,z)','x=(y*z/100)') //Function to Find out the Running Costs
//Running Costs
Fuelr=Runcost(90,Fuel);
Genr=Runcost(10,Gen);
Transr=Runcost(5,Trans);
Distr=Runcost(7,Dist);
Totr=Fuelr+Genr+Transr+Distr; //Total Cost

FixCost=Tot-Totr;// Fixed Cost
FixChar=FixCost*1000/MD; //Fixed Charges per KW
EnChar=Totr*100/(EG*SCeff); //Energy Charges in Paise for Consumer
OverCost=Tot*100/(EG*SCeff); //Overall Energy Charges

LF=40/100; //Load Factor Raised to 40%
EG1=LF*MD*8760/1000;  //Energy Generated for Different Load Factor
Totr1=Totr*(EG1/EG); //Cost of Energy Generated
Tot1=FixCost+Totr1; //Total Cost for the New Load Factor
OverCost1=Tot1*100/(EG1*SCeff); //Overall Energy Charges

Saving=(OverCost-OverCost1)*100/OverCost; //Percentage Saving in the Overall Cost per kWhr

printf('The Fixed Charges is %g rupees per kW\n',FixChar)
printf(' The Energy Charges for the Consumer is %g paise per kWhr\n',EnChar)
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

```

//To determine the most economic power factor
//Page 91
clc;
clear;

P=200*(10^3); //Maximum Demand
pf=0.707; //Power Factor Lagging

a=100; //Tariff per kVA per year

b=200; //Power factor improvement cost Per kVA.
r=20; //Interest Depriciation , maintenance and cost
    of losses amount to  20% of capital cost per year

// Economic PF = sqrt(1-((b1/a)^2))

b1=r*b/100;// b' term accrding to the equation above

pfeco=sqrt(1-((b1/a)^2)); //Economic Power Factor

printf('The Economic Power Factor is %g \n',pfeco)
```

Chapter 3

Distribution

Scilab code Exa 3.1 To Determine the most economical current density

```

//To Determine the most economical current density
// Page 103
clc;
clear;

OFC=15; //Cost of single phase overhead feeder per
    km per unit area
AIDC= 10*OFC/100; //Annual Interest and depriciation
     charges
R=1/58; //Resistance of 1m length and 1 sq.mm
CLPUL= 2*R*1000; // Copper Loss per unit length per
    unit area per unit square current
ACL= CLPUL*365*24*0.5; // Annual Copper Loss
GC= 5/100; // Genereating Cost per unit.
ACEL= ACL*GC/1000; // Annual cost of energy loss
CD= sqrt(AIDC/ACEL); //Current Density
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

```

//To determine the most economical cross sectional area
//Page 104
clc;
clear;

OFC=180; //Cost of single phase overhead feeder per km per unit area + 1200
AIDC= 10*OFC/100; //Annual Interest and depriciation charges + 120
R=1/58; //Resistance of 1m length and 1 sq.mm
I=200; // Maximum Current Flowing
CLPUL= 2*(I^2)*R*1000; // Copper Loss per unit length per unit area
ACL= CLPUL*365*24*(8/12); // Annual Copper Loss
GC= 5/100; // Genereating Cost per unit.
ACEL= ACL*GC/1000; // Annual cost of energy loss
A= sqrt(ACEL/AIDC); //Cross Sectional Area of the cable
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 suppl

```

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

```
clear;
deff('a=LLF(b)','a=(0.25*b)+(0.75*(b^2))'); //
    Function to determine the Loss Load Factor

OFC=0.20; //Cost of single phase overhead feeder per
    m per unit area + 10
AIDC= 10*OFC/100; //Annual Interest and depriciation
    charges + 1
TE= 2.5*(10^6); // Total energy to be supplied per
    annum
CEW=10/100; // Cost of energy wasted per unit
LFS= TE/(1000*365*24); // Load factor of supply
Llf=LLF(LFS); // Load Loss factor
R=1/58; // Resistance of the cable per unit length
PF=1; // Unity power factor
MD= 1*(10^6); // Maximum Demand
V=11*(10^3); // Voltage of the feeder
I=MD/(sqrt(3)*V*PF); // Full Load Current
FLCL= 3*(I^2)*R; // Full Load Copper Loss per Metre
ACL= Llf*FLCL; // Actual Copper Loss
CCL=ACL*(365*24*CEW/1000); // Cost of Copper Loss
A=sqrt(CCL/AIDC);
printf('The Most Economical Cross sectional area for
    this Case is %g A/sq.mm',A)
// 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 l

```

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

```
clear;

OFC=5000; //Cost of feeder per km per unit area + 6250
AIDC= 8*OFC/100; //Annual Interest and depriciation charges + 625
AIDCPD= AIDC/365; // Annual Interest and depriciation charges per day
R=1/58; //Resistance of 1m length and 1 sq.mm
CEU= 5/100; // Cost of energy per unit
V=33*(10^3); // Voltage of the feeder
L10=3000*(10^3); // Load for 10 hrs at unity power factor.
L8=1000*(10^3); // Load for 8 hrs at unity power factor.
L6=2000*(10^3); // Load for 6 hrs at unity power factor.

deff('a=LC(b)','a=b/(sqrt(3)*V)')// Funtion to determine the Load Current
I10=LC(L10);// Load Current for 10 hrs at unity power factor.
I8=LC(L8);// Load Current for 8 hrs at unity power factor.
I6=LC(L6);// Load Current for 6 hrs at unity power factor.
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)
CEL=ELPD*CEU/1000; // Cost of energy loass per km
A=sqrt(CEL/AIDCPD); // Cross Sectional Area
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`

```
// To Calculate Voltage between middle wire and
    outer at each load point
// Page 109
clc;
clear;

Vs=220; //Supply Voltage
//Resistances of the respective Sections
Rab=0.2;
Rbc=0.2;
Rdh=0.2;
Rhe=0.2;
Ref=0.3;
Rfl=0.1;
Rlg=0.3;
Rij=0.25;
Rjk=0.2;
Ra=0.2;
Rd=0.4;
Ri=0.3;

//Currents following through the respective Section,
    Found using KCL
Iad=5;
Ibe=10;
Icl=12;
Ihi=15;
Ifj=5;
Igk=15;
Ia=Iad+Ibe+Icl; // Current through the positive wire
Iab=Ia-Iad;
Ibc=Ia-Iad-Ibe;
Ii=Ihi+Ifj+Igk; // Current through the negative wire
Id=Ii-Ia; //Current through the Middle wire
Idh=Iad+Id;
Ihe=Ihi-Idh;
Ief=Ibe-Ihe;
```

```
Ifl=Ief-Ifj;
Ilg=Igk;
Iij=Ii-Ihi;
Ijk=Ii-Ihi-Ifj;

//Voltage drops across each section
Va= Ra*Ia;
Vab=Iab*Rab;
Vbc=Ibc*Rbc;
Vi=Ii*Ri;
Vd=Id*Rd;
Vdh=Idh*Rdh;
Vhe=Ihe*Rhe;
Vef=Ief*Ref;
Vfl=Ifl*Rfl;
Vlg=Ilg*Rlg;
Vij=Iij*Rij;
Vjk=Ijk*Rjk;

//Voltage across the middle wire and the outer load
   points Using KVL
Vad=Vs-Va+Vd;
Vbe=Vad-Vab-Vhe+Vdh;
Vcl=Vbe-Vbc+Vfl+Vef;
Vhi=Vs-Vd-Vdh-Vi;
Vfj=Vhi-Vef+Vhe-Vij;
Vgk=Vfj-Vfl-Vlg-Vjk;

printf('The Voltages between middle wire and outer
   wire at each load point are:\n')
printf(' 1. Vad = %g V\n',Vad)
printf(' 2. Vbe = %g V\n',Vbe)
printf(' 3. Vcl = %g V\n',Vcl)
printf(' 4. Vhi = %g V\n',Vhi)
printf(' 5. Vfj = %g V\n',Vfj)
printf(' 6. Vgk = %g V\n',Vgk)
```

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

```
//To obtain the voltages at the far end of the positive and negative wires
//Page 110
clc;
clear;

Vs=220; // Supply Voltage
//Resistances of the given sections
r1=0.015;
r2=0.035;
r3=0.02;
r4=0.01;
r5=0.025;
r6=0.015;
r7=0.03;
r8=0.01;
r9=0.02;
r10=0.03;
r11=0.04;

//Currents follwing in between the two wires
I1=10;
I2=20;
I3=25;
I4=5;
I5=15;
I6=15;
I7=15;
I8=15;
I9=18;
I10=30;
I11=15;
```

```
//Effective resistances for the above currents
R1=r1;
R2=r1+r2;
R3=r1+r2+r3;
R4=r1+r2+r3+r4;
R5=r1+r2+r3+r4+r5;
R6=r1+r2+r3+r4+r5+r6;
R7=r7;
R8=r7+r8;
R9=r7+r8+r9;
R10=r7+r8+r9+r10;
R11=r7+r8+r9+r10+r11;

//Voltage drop in the respective wires
//Outer positive wire
Vop=(I1*R1)+(I2*R2)+(I3*R3)+(I4*R4)+(I5*R5)+(I6*R6);
//Outer Negative wire
Vnp=(I7*R7)+(I8*R8)+(I9*R9)+(I10*R10)+(I11*R11);
// Net drop in neutral wire
Vn=Vop-Vnp;

//Effective drop in the respective wires
//Outer positive wire
Veop=Vop+Vn;
//Outer negative wire
Venp=Vnp-Vn;

//Voltage at far end of the wires
//Positive Wire
Vpf=Vs-Veop;
//Negaitve Wire
Vnf=Vs-Venp;

printf("The voltage drop at the far end of the wires
    are:\n")
printf(" Positive Wire: %g V\n",Vpf)
printf(" Negative Wire: %g V\n",Vnf)
```

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

```
//To determine the cross section of the conductor
   for a minimum consumer voltage
//Page 111
clc;
clear;

//Unknown Variable obtained in the equation
Ib=poly(0,"Ib");
r=poly(0,"r");

//Voltages at the respective ends
Va=230;
Vb=230;

//Minimum Consumers's Voltage
Vc=220;

//Lenghths of the segments
r1=200;
r2=200;
r3=100;
r4=300;
r5=300;
r6=100;

//Effective length of segments
R1=r1;
R2=r1+r2;
R3=r1+r2+r3;
R4=r1+r2+r3+r4;
R5=r1+r2+r3+r4+r5;
```

```
R6=r1+r2+r3+r4+r5+r6;

//Current drawn by different loads
I1=25;
I2=20;
I3=25;
I4=25;
I5=10;

//By the law of momemts of currents
Vnet=2*((I1*R1)+(I2*R2)+(I3*R3)+(I4*R4)+(I5*R5)-(Ib*
    R6)); //Since the equation is equated to zero , r
    vanishes
X=Vnet-(Va-Vb); //Polynomial Equation to Find Ib
Ib=roots(X); // Numerical Value of Ib
Ia=(I1+I2+I3+I4+I5)-Ib;

//From the given figure it is clear that point M is
    the point of Minimum Potential
Vd=Va-Vc;
X=(2*r*((Ia*r1)+((Ia-I1)*r2)+((Ia-I1-I2)*r3)))-Vd;
    //Polynomial Equation to Find r
r=roots(X); // Resistance per unit length (Numerical
     Value)

Rstd=(1/58); // Resistance for 1m and 1 sq.mm

A=Rstd/r; // Cross Section of Conductor required.

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

```
//To determine the cross section of the conductor
    for a minimum consumer voltage
//Page 111
clc;
clear;

//Unknown Variable obtained in the equation
x=poly(0,"x");
r=poly(0,"r");

//Voltages at the respective ends
Va=235;
Vb=230;

//Minimum Consumers's Voltage
Vc=220;

//Lenghths of the segments
r1=200;
r2=200;
r3=100;
r4=300;
r5=300;
r6=100;

//Effective length of segments
R1=r1;
R2=r1+r2;
R3=r1+r2+r3;
R4=r1+r2+r3+r4;
R5=r1+r2+r3+r4+r5;
R6=r1+r2+r3+r4+r5+r6;

//Current drawn by different loads
I1=25;
I2=20;
I3=25;
```

```
I4=25;
I5=10;

//The Minimum Point assumed is N, Hence the current
   following to Point N is given by 'x'

//The Effective Drops
Van=Va-Vc;
Vbn=Vb-Vc;
V=Van/Vbn;
A=2*(((x+I1)*r1)+(r2*x)); //Wrt to Van
B=2*(((I2-x)*r3)+((I2+I3-x)*r4)+((I2+I3+I4-x)*r5)+((
   I2+I3+I4+I5-x)*r6)); //Wrt to Vbn
C=A/B; // The 'r' term gets eliminated
X=C-V; // Polynomial Eqaution to find x
X=X(2); // We take only the numerator into
   consideration as the above equation is eaquated
   to zero
x=roots(X);

Ia=I1+x; //Current Supplied at end A;
//It is clear the above assumed Minimal Point is
   wrong and it has to Point M
//Therefore finding drop at Point M

X=(2*r*(((x+I1)*r1)+(r2*x)+((x-20)*r3)))-Van; //
   Polynomial Equation to find r
r=roots(X); // Numerical Value of r

Rstd=(1/58); // Resistance for 1m and 1 sq.mm

A=Rstd/r; // Cross Section of Conductor required.

printf('The Cross Section of the Conductor to
   provide minimal consumer''s voltage is %g sq.mm\n
   ',A)

// 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

```

//To Determine the location and magnitude of minimum
    voltage
//Page 114
clc;
clear;

Vs=220; //Supply Voltage at End A and B
//Different Conductor Lenghts
//From End A
L1=100;
L2=50;
L3=50;
L4=400; //Length of uniform loading

A=0.5; //Uniforming loading spread over 400m
r=0.05; // Resistance of Conductor per Km
//Different Currents drawn by various loads
I1=50;
I2=75;
I3=A*L4;

//Taking moments of all currents at A;
Ib=((I1*L1)+((L1+L2)*I2)+((L1+L2+L3+(L4/2))*I3))/(L1
   +L2+L3+L4);
Ia=I1+I2+I3-Ib;

//Minimum Potential Point in this case is the point
   where All the current from B is drawn
X=Ib/A; // Distance from B;
```

```
Y=(L1+L2+L3+L4)-X; //Distance from A;

//Minimum Potential Drop
Vmind=(2*r/1000)*((Ia*L1)+((Ia-I1)*L2)+((Ia-I1-I2)*
   L3)+(((Y-L1-L2-L3)*A)*((Y-L1-L2-L3)/2)));

Vmin=Vs-Vmind; //Minimum Potential
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 an

```

//To determine the currents supplied to the ring
   main from A and B
//Page 115
clc;
clear;

//Currents in the ring scheme going clockwise
I1=50;
I2=15;
I3=25;
I4=10;
It = I1+I2+I3+I4; // Total Current
//Resistances of the respective segments going
   clockwise
R1=0.1;
R2=0.12;
R3=0.04;
R4=0.08;
R5=0.06;
R6=0.02;

Ra=0.04; //Resistance at A;
```

```
Rb=0.06; //Resistance at B;

// Va=Vb; Net Voltage is Zero
//Dividing the total current from A as x and y
//Taking voltage drops in clockwise direction and
    anticlockwise directions
// We get two simultaneous equations
// 3.2x + y = 120
// x + 3y = 114
R=[3.2,1;1,3];
V=[120;114];
I=inv(R)*V;

//To calculate and seperate the respective currents
    from the above matrix eqution
x=I(1);
y=I(2);
Ia=x+y;
Ib=It-Ia;

printf('a) When Va = Vb; Ia = %g A and Ib =%g\n',Ia,
    Ib)

// Va+5=Vb; Net Voltage is 5V
//Dividing the total current from A as x and y
//Taking voltage drops in clockwise direction and
    anticlockwise directions
// We get two simultaneous equations
// 3.2x + y = 120 -(5*NetVoltage) = 70
// x + 3y = 114 -(5*NetVoltage) = 64
R=[3.2,1;1,3];
V=[70;64];
I=inv(R)*V;

//To calculate and seperate the respective currents
    from the above matrix eqution
x=I(1);
y=I(2);
```

```
Ia=x+y;
Ib=It-Ia;

printf('b) When Va + 5 = Vb; Ia = %g A and Ib =%g\n'
    ,Ia,Ib)

//Highly Accurate Answers, Text Book answers are
    rounded off.
```

Scilab code Exa 3.11 To Compare the volume of copper required

```
//To Compare the volume of copper required
//Page 116
clc;
clear;
//Unknown Resistances
r=poly(0,'r');
r1=poly(0,'r1');
//Lengths of the segements of the ring scheme
L1=100;
L2=200;
L3=200;
L4=150;
L5=150;
//Currents taken by respective loads
I1=40;
I2=20;
I3=100;
I4=40;
It=I1+I2+I3+I4;//Total Current

//Without Interconnector
//Let x be the current flowing through the entire
    ring
```

```
x=poly(0,'x');
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

x=roots(Eq);
x1=It-x; //Current flowing in the other direction

Vac1=((x1*L5)+((x1-I4)*L4))*r; // Voltage across AC without the connector
MVac1=((x1*L5)+((x1-I4)*L4)); // Magnitude of Vac1;

//With Interconnector
//Considering x amount of current to flow clockwise through segment AE
//Considering y amount of current to flow anticlockwise through segment AB
//Considering 200-(x+y) amount if current to flow through the segment AC
// Mesh Analysis of ABCDE gives 5x - 3y = 140;
// Mesh Analysis of ABC gives 5x + 11y = 1120;
R=[5,-3;5,11];
V=[140;1120];
I=inv(R)*V;
x=I(1);
y=I(2);
Vac2=(It-(x+y))*250*r1; // Voltage across AC with connector
MVac2=(It-(x+y))*250; // Magnitude of Vac2;
printf('The Voltage drop across AC in both case is the same\n')
disp(Vac2,'Is Equal to',Vac1)
printf('\n \n')
//To Compute the Numerical Values of the Ratio of resistances
RatioA = MVac1/MVac2;

disp(RatioA,'is',r1,'divided by',r)
```

```
//Effective Length of both the cases
Leff=L1+L2+L3+L4+L5;
LeffC=Leff+250;
//Volume is Length * Area
RatioV=Leff*RatioA/LeffC;

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

```

//To Determine the Voltage at the far end
//Page 117
clc;
clear;
r=0.3; // Loop Resistance per km
x=0.15; // Loop Reactance per km

deff('x=reactive(y,z,v)','x=(sin(acos(y)))*z*v*x
   /1000') //To find the reactive drop of the
   current

deff('x=active(y,z,v)','x=y*z*v*r/1000') //To find
   the reactive drop of the current

//Power Factors of the loads from left to right
pf1=0.707;
pf2=1;
pf3=0.8;
//Currents delivering the respective loads from left
    to right
I1=50;
```

```
I2=60;
I3=40;
//Length of the conductors in metres
l1=200;
l2=300;
l3=300;
//Effective length of the conductors in metres
L1=l1;
L2=l1+l2;
L3=l1+l2+l3;
//Active component drops of the respective currents
Va1=active(pf1,I1,L1);
Va2=active(pf2,I2,L2);
Va3=active(pf3,I3,L3);
Vat=Va1+Va2+Va3; //Total Active Component Drop
//Reactive component drops of the respective
    currents
Vr1=reactive(pf1,I1,L1);
Vr2=reactive(pf2,I2,L2);
Vr3=reactive(pf3,I3,L3);
Vrt=Vr1+Vr2+Vr3; //Total Reactive Component Drop

Vt=Vrt+Vat; // Total voltage drop

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

```

//To determine line currents and neutral currents
//Page 118
clc;
clear;
V=400; // Supply Voltage
```

```
Vph=400/sqrt(3); //Phase Voltage
L=480*(10^3); //Balanced Load
//Loads at unit power factor
Lr=50*(10^3); //Load in R phase
Ly=150*(10^3); //Load in Y phase
Lb=200*(10^3); //Load in B phase
pf=0.8; //Power Factor lagging
theta=-1 * acos(pf); //Angle in radians of the
   balanced LAGGING current

I=L/(sqrt(3)*V*pf)*exp(%i*theta); //Balanced Current
    Magnitude
Ir=Lr/Vph; //Magnitude of R phase voltage
Iy=Ly/Vph; //Magnitude of Y phase voltage
Ib=Lb/Vph; //Magnitude of B phase voltage

//Vr is taken as reference
//Angles of the phase voltages wrt to the reference
Thetay=-120;
Thetab=120;
Thetar=0;
//Net Currents in the respective phases (RYB)
Irf=Ir+I;
Iyf=Iy+I;
Ibf=Ib+I;
//Angle of the above currents in degrees
r=atand(imag(Irf)/real(Irf));
y=atand(imag(Iyf)/real(Iyf));
b=atand(imag(Ibf)/real(Ibf));
//Angles of the above currents with respect to the
   reference
rf=Thetar+r;
yf=Thetay+y;
bf=Thetab+b;
//Effective Currents wrt to the reference voltage
Irn=abs(Irf)*exp(%i*%pi*rf/180)
Iyn=abs(Iyf)*exp(%i*%pi*yf/180)
Ibn=abs(Ibf)*exp(%i*%pi*bf/180)
```

```
In=Irn+Iyn+Ibn; // Neutral Current
//Note Take Vr as reference
printf('The Net Current in phase R is %g/_%g A\n',
    abs(Irf),rf);
printf('The Net Current in phase Y is %g/_%g A\n',
    abs(Iyf),yf)
printf('The Net Current in phase B is %g/_%g A\n',
    abs(Ibf),bf)
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

```

//To determine the best location of the substation
    for a given set of loads
//Page 123
clc;
clear;

//Various loads and how they are positioned on the
    corners of a square of length 25km
L1=5000; //(0,25)
L2=8000; //(25,25)
L3=3000; //(25,0)
L4=6000; //(0,0)

L=25;//Length of the square

TL=L1+L2+L3+L4; // Total load

X=((L1*0)+(L2*L)+(L3*L)+(L4*0))/TL // X coordinate
Y=((L1*25)+(L2*L)+(L3*0)+(L4*0))/TL// Y coordinate

printf('The Susbstation must be located at (%g km,%g
```

```
km)\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

```

//To Determine Motor Parameters of Series Motor
   rated at 220V
//Page 201
clc;
clear;
V=220; //Rated Voltage
N=1000; //Rated Speed
Ish=20; //Current without Armature being shunted
Ra=0.15; //Armature Resistance
Rf=0.2; //Field Resistance
Eb=V-(Ish*(Ra+Rf)); //Back EMF
K=Eb/(Ish*N); //Motor Constant
Rsh=20; //Shunt Resistance

//After Armature is shunted
x=poly(0,'x'); //Field Current Variable Value
Ia=(Ish^2)/x; //Armature Current //Since Torque is
   assumed constant
```

```
I20=(V-(x*Rf))/Rsh; //Current Through the Shunt resistance

X=Ia+I20-x; //Polynomial to find field Current
x=roots(X(2)); //Numerical Value of Field Current

//Conditon to find the Positive Root
if(abs(x(1))+x(1)==0)
    x=x(2);
else
    x=x(1);
end

Ia=(Ish^2)/x; //Armature when shunted
Ebsh=V-(Ia*Ra)-(x*Rf); //Back Emf when shunted
Nsh=Ebsh/(K*x); //Speed when armature is shunted

//When Load is not there, Ia = 0
If=V/(Rsh+Rf); //Field Current at No Load
Ebnl=V-(If*Rf); //Back Emf at No Load
Nnl=Ebnl/(K*If); //Speed at No Load

Rex=5; //External Resistance
//At Starting when Eb = 0
Is=V/(Rex+Rf+(Ra*Rsh/(Ra+Rsh))); //Supply Current
Ifs=Is; //Field Current
Ias=Ifs*Rsh/(Rsh+Ra); //Armature Current At the Start

//Torque is directly proportional to square of Current
T20 = (Ish^2);  //Torque at shunted condition
Tas= Ias*Ifs; //Torque at start

RT=Tas/T20; //Ratio of Torques

printf('a) The Speed of the Motor if the Load Torque
    is remains constant and the magnetic circuit is
```

```
printf('b) The No Load Speed of the Motor is %g RPM\n',Nnl)
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

```

//To Determine the resistance to be inserted for various cases
//Page 203
clc;
clear;

N1=1500; //Rated Speed
N2=1200; //Reduced Speed
rs=N2/N1; //Ratio of final speed to original speed
eff=80/100; //Efficiency
Pr=10*(10^3); //Power Rating
V=250; //Voltage supply
Il=Pr/(V*eff); //Full Load Current
Rf=110; //Field Resistance
Ra=0.25; //Armature Resistance
If=V/Rf; //Field Current
Ia1=Il-If; //Load Current
Eb=V-(Ia1*Ra); //Back EMF
Eb1200=Eb*rs; //Back EMF for 1200 rpm

deff('x=Res(y)','x=((V-Eb1200)/y)') //Function to calculate the Resistance of various cases
//Torque directly proportional to current
Ia2=(rs)*Ia1; // Torque directly propostional to speed
```

```
Ia3=(rs^2)*Ia1; //Torque directly proportional to
    square of the speed

//Resistances for Various Cases
R1=Res(Ia1)-Ra;
R2=Res(Ia2)-Ra;
R3=Res(Ia3)-Ra;

deff('x=Pow(y)','x=((V-Eb1200)/y)*(y^2)/1000') //
    Function to Find Power Loss
//Power Lost in kW for various cases
P1=Pow(Ia1);
P2=Pow(Ia2);
P3=Pow(Ia3);

printf('The External Resistances to be connected and
    the power loss for\n')
printf('a) The Load Torque is independent of the
    speed : %g ohm and %g kW respectively\n',R1,P1)
printf('b) The Load Torque is directly proportional
    to the speed : %g ohm and %g kW respectively\n',
    R2,P2)
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

```

//To determine the speed for which the torque is
    maximum
//Page 204
clc;
clear;

```

```
//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

//This Doesn't Affect the calculation of the speed but it does affect the Maximum Torque

//So if we consider 1 ohm we get 0.8333 instead of 2.5

//The Equation for the Torque is found out to be 2.5*K2*(V^2)*(1.1-K1*w)/((76.1-(50*K1*w)))

w=poly(0,'w'); //Variable Value of w;

//Lets assume the value of 2.5*K2*(V^2) = 1 and K1 = 1 to particularly to calculate the co-effcients

T=2.5*(1.1-w)/((76.1-(50*w))^2); //Torque

//Maximum Torque is derivative of the above equation
X=derivat(T); //Polynomial to find the value co-effcient of w;
w=roots(X(2));
w=w(2); //We Choose value less than 1 to suit the differenciation process

Tmax=2.5*(1.1-w)/((76.1-(50*w))^2); //Maximum Torque Co-Efficient

printf('The Speed at which the torque is Maximum is (%g/K1) rad/sec\n',w)
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

```
//To Determine the slip at maximum torque and at full load
//Page 205
clc;
clear;

Tl=1; //Assume that the Full Load torque is unity
Tst=125*Tl/100; //Starting Torque
Tmax=275*Tl/100; //Maximum Torque
si=1; //At the Start
RT=Tst/Tmax; //Ratio of Starting Torque and Maximum Torque
//R1 is negligible , elminating the terms with R1 and hence formulating the starting torque and Maximum Torque we get

// Tst/Tmax = 2*sTmax*s/((sTmax^2)+(s^2))
//Where sTmax is the Slip at Maximum Torque and s is the slip at the start

sTmx=poly(0,'sTmx'); //Variable Value of slip at Maximum Torque

X=RT-(2*sTmx*si/((sTmx^2)+(si^2))); //Polynomial to find sTmax

sTmx=roots(X(2));

//Condition to find the value of sTmax less than 1
if(floor(sTmx(1))==0)
    sTmx=sTmx(1);
else
    sTmx=sTmx(2);
end

```

```
s=poly(0,'s'); //Variable value of slip for Starting torque at full load

Y=(Tl/Tmax)-(2*sTmx*s/((sTmx^2)+(s^2))); //Polynomial to find s

s=roots(Y(2)); //Numerical Value of s
//Condition to find the value of s less than 1
if(floor(s(1))==0)
    s=s(1);
else
    s=s(2);
end

printf('i) The Slip at Maximum Torque is %g\n',sTmx)
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

```

//To Determine the Maximum Torque as ratio of nominal value of voltage
//Page 205
clc;
clear;

//Resistances and Reactance for Nomrmal Value of Volttage and Frequency
w=50; //Frequency of supply
V=1; //Assumed Voltage
R1=0.05;
R2=0.05;
x1=0.3;
x2=0.3;

```

```
//Resistance and reactance at half voltage and half frequency
wh=50/2; //Frequency of supply
Vh=1/2; //Assumed Voltage
R1h=0.05;
R2h=0.05;
x1h=0.3/2;
x2h=0.3/2;

//Function to Determine the Maximum Torque
deff('x=Max(a,b,c,d,e)','x=(3*(a^2))/(2*b*(c+(sqrt((c^2)+((d+e)^2)))))')
//Function to Determine to the Starting Torque
deff('x=Start(a,b,c,d,e,f)','x=(3*(a^2)*f)/(b*(((c+f)^2)+((d+e)^2)))')
//Maximum Torques and Starting Torques at
//Normal Voltage And Frequency
Tm=Max(V,w,R1,x1,x2);
Tst=Start(V,w,R1,x1,x2,R2);
//Half Voltage and Half Frequency
Tmh=Max(Vh,wh,R1h,x1h,x2h);
Tsth=Start(Vh,wh,R1h,x1h,x2h,R2h);

//Ratio of torques as fraction of its normal value
R1=Tmh/Tm;
R2=Tsth/Tst;

printf('i)The Maximum Torque at the reduced value of supply as a fraction of its normal value is %g\n',R1)
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

```

//To Determine the starting torque and starting
    current for various starters
//Page 206
clc;
clear;
V=400;//Rated Voltage //Phase Voltage //Delta
    Connected
Pr=50*735.5; //Rated Power
N=750; //Rated Speed
s=4.5/100; //Full load Slip
Il=50; //Full Load Current
Z=2.5; //Impedance per phase
w=2*%pi*N/60; //Angular Frequency
Tl=Pr/w; //Full Load Torque
Ist=V/Z; //Starting Current
Isl=sqrt(3)*Ist; //Line Current


//DOL Starter
Tstd=((Isl/Il)^2)*s*Tl; //Starting Torque

//Star - Delta Starter
Tsts=Tl*((1/sqrt(3))^2); //Starting Torque

//Autotransformer
tap=70/100; //Tapping
Tsta=(tap^2)*Tl; //Starting Torque

printf('The Starting Current for all the three
    starter are the same that is %g A\n',Isl)
printf('The Starting torque for:\n')
printf('i) D.O.L Starter : %g Nm\n',Tstd)
printf('ii) Star -Delta Starter : %g Nm\n',Tsts)
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

```

//To Detemine Minimum Starting Current drawn from
   the supply
//Page 207
clc;
clear;

TR=0.5; //Torque Ratio, Tst:Tl
V=400; //Supply Voltage
Pr=30*735; //Power Rating
N=500; //Rated Speed
eff=0.85; //Efficieny
pf=0.88; //Power Factor
Is=150; //Short Circuit Current
s=5/100; //Full load Slip
pfs=0.25; //Power Factor of the Short Cicuit Current
Il=Pr/(sqrt(3)*V*eff*pf); //Full Load Current

//Case 1
x=poly(0,'x'); //Tapping Percent
X=TR-(((Is/Il)^2)*(x^2)*s); //Polynomial to
   determine the 'x'
x=roots(X)
x=x(1); //Taking the Postive Root
Ist1=Is*x*x; //Starting Current numerical Value

//Case 2
Ist2=sqrt((TR*(Il^2)/s)); //Starting Current for
   Full Voltage

Z1=V/(sqrt(3)*Ist2); //impedance for the starting
   current at full voltage
```

```
Z2=(V/(sqrt(3)*Is))*exp((%i*%pi*acosd(pfs))/180); //Phasor form of the Impedance of for the short cicuit current

R=sqrt((Z1^2)-(imag(Z2)^2));
Rs=R-real(Z2); //Resistance to be connected the stator circuit

printf('i) The Tapping of the Transformer is %g percent and the starting Current is %g A\n',(x*100),Ist1)
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

```

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

```
ws=2*%pi*Ns/60; //Synchronous angular Frequency
s=(Ns-N)/Ns; //Slip
T=Pf/w; //Full Load Torque
x2=poly(0,'x2'); //Reactance of Rotor referred to
    stator, Variable Value

X=T-(3*(V1^2)*R2/(ws*s*(((R2/s)^2)+(x2^2)))); //
    Polynomial Expression to find X2
X2=roots(X(2)); //Numerical Value of Rotor
    Resistance
X2=X2(1);
N1=180; //Speed of the fan
s1=(Ns-N1)/Ns; //Slip
T1=T*((N1/N)^2); //Torque
R21=poly(0,'R21'); //Variable Value
Y=T1-(3*(V1^2)*R21/(ws*s1*(((R21/s1)^2)+(X2^2))));
    //Polynomial Expression to find R21
R21=roots(Y(2)); //Numerical Value

//R21 = 0.7545054 and 0.0100813
//We have to choose the value of R21 such that it is
     greater than R2

R21=R21(1);
Rex=(R21-R2)/(SR*SR); //External Resistance

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

```

//To Determine the time taken to start the motor
//Page 222
```

```
clc;
clear;
HP=746; //According to the TextBook
P=150*HP; //Power Rating
Lim=1.5; //Limited Current Factor
Eshp=5000; //energy stored per hp
N=750; //Rated Speed
w=N*2*%pi/60; //Angular Frequency
Es=Eshp*P/HP; //Total Energy Stored
Tfl=P/w; //Full Load Torque;
Ts=Tfl*Lim; //Starting Torque
Ta=Ts-Tfl; //Torque Available for acceleration
I=2*Es/(w^2); //Moment of Inertia

a=Ta/I; //Angular Accleration

//Intergrating wrt angular frequency
t=integrate('1/a','x',0,w); //Time Taken to start //
    Angular Acceleration is constant

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',t)
```

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

```
//To time taken and the number of revolutions made
    before the motor is stopped
//Page 223
clc;
clear;
P=50*735.5; //Power Rating
```

```
V=400; //Rated Voltage
N=750; //Rated Speed
w=N*%pi*2/60; //Angular Frequency
I=20; //Moment Inertia
pf=0.95; //Full Load power factor
eff=90/100; //Overall Efficiency
Rb=2; //Braking Resistor

Tm=P/w; //Full Load Torque
Ifl=P/(sqrt(3)*V*eff*pf); //Full Load Current
Ilb=V/(sqrt(3)*Rb); //The Line Current at the start
    of braking
Te=Tm*Ilb/Ifl; //Torque for electric braking

Tb=Te+Tm; //Total Braking torque

K=Te/w; //Motor constant
B=Tb/I; //Retardation

ts=integrate('I/(Tm+(K*x))','x',0,w); //Time Taken
    for the motor to stop

NoR=(1/(2*%pi*K))*integrate('((Tm+(K*w))*exp(-1*K*t/
    I))-Tm','t',0,ts); //Number of Revolutions

//Note The Value of time in the textbook is wrong,
    Calculation error

printf('The Time Taken for the Motor to stop is %g
    seconds\n',ts)
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

```
//To Determine the 30 mins rating of motor
//Page 229
clc;
clear;
Pr=100; //Power Rating in horse power
Tc=90*3600; //Time Constant
Tr=30*3600; //Time to determine the rating
theta=50; //Temperature Rise
theta1=50; //Temperature rise for the new case

P=poly(0,'P'); //Power rating for 30min variable value

//Loss Corresponding to 30 mins

P30=(P/Pr)^2; // Times the Power loss corresponding to the 100 hp motor

theta2=theta*P30; //Final Temperature rise if run continuously

X=theta1-(theta2*(1-exp(-1*Tr/Tc))); //Polynominal to find the 30min rating

P=roots(X); //Numerical Value
//We Consider the Positive Real Value of the root

P=P(1); //Power Rating of the 1/2 Hr

printf('The 1/2 Hr Power Rating is %g hp\n',P)
```

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

```
//To Determine the Continous Rating of the Motor for
    Suitable Conditions
//Page 229
clc;
clear;
T=120; //Total Time
t1=0; //Power Rise to 1050
t2=15; //Constant Power of 600
t3=85; //Regenerative Breaking at 200
t4=95; //Motor At Rest
t5=120; //End
//Note Power Rating are in Horse Power
HP1=1050; //First Rise
T1=(t2-t1); //Time for Rise of the Horse Power
HP2=600; //Constant Power
T2=(t3-t2); //Time For which the power is constant
HP3=200; //Braking Initial Point
T3=(t4-t3); //Time Period of Breaking

//Different Intergration
I1=integrate('((HP1/T1)*t)^2','t',t1,t2);
I2=integrate('(HP2^2)','t',t2,t3);
I3=integrate('((HP3/T3)*t)^2','t',0,T3);

HPrms=sqrt((I1+I2+I3)/120);

HPav=(((HP1*T1/2)+(HP2*T2)+(HP3*T3/2))/T);

printf('i) The H.P Rating according to the RMS Value
    of loading is %g hp\n',HPrms)
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

```
//To estimate the final temperature of the machine
    and its time constant
//Page 230
clc;
clear;

Eff=90/100; //Efficiency
P=15*735.5; //Power Rating in W
D=75*(10^-2); //Cylinder Diameter
L=120*(10^-2); //Length of the Cylinder
OSHD=12; //Outer Surface Heat Dissipation
A=%pi*L*D; //Curved Surface Area
M=450; //Motor Weight
SH=700; //Specific Heat

HTC=M*SH/(A*OSHD*3600); //The Heating Time Constant
Loss=(P/Eff)-P; //Loss in Motor

FT=Loss/(A*OSHD); //Final  Temperature
OSHD2=20; //Semi Closed Cooling Capacity
LossSC=FT*A*OSHD2; //Loss in Semi Closed System

L2=poly(0,'L2'); //Variable Load Value
X=(L2/(L2+LossSC)-Eff); //Polynomial to Calculate L2
L2=roots(X(2)); //Numerical Value of The Load

printf('The Final Temperature of the Machine and its
     Heating Time Constant is %g degree Celsius and
    %g Hrs respectively \n',FT,HTC)
printf('The Load which could be delivered is %g W\n'
    ,L2)
```

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

```
//To Determine the Moment of intertia for a slip of 8 percent
//Page 233
clc;
clear;
Tl=900; //load Torque
To=0;
s=8/100; //Slip
t=15;//time period for which the load torque was applied
Tm=675;// Limited motor Torque
N=600; //No Load Speed
Wo=2*%pi*N/60; //Angular Frequency
K=s/Tm; //Motor Constant
I=(t/(Wo*K*log((Tl-To)/(Tl-Tm)))); //Moment of Inertia

printf('The Moment of inertia of the flywheel is %g Kg m^2\n',I)
//Please Note the Calculation Mistake in the book. (Power of e)
```

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

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

```
Ta=3000; //Additional Torque
Tmin=600; //Constant Load Torque
Tm=Pi*60/(2*%pi*N); //Rated Torque
Tmax=2*Tm; //Maximum Torque
Tl=Tmin+Ta; //Total Torque
tp=15; //Time for which the additional torque is applied
//Assuming Slip Characteristic to Be a Straight line
s=0.05;
K=s/Tm; //Motor Load Constant

I=tp/(Wo*K*log((Tl-Tmin)/(Tl-Tmax))); //Moment Of intertia

Tm1=1500; //Trasistion Torque

t=I*Wo*K*log((Tmax-Tmin)/(Tm1-Tmin)); //Time Taken to become 1500 Nm

printf('a) The Moment of intertia of the flywheel for Maximum Torque to be twice the rated torque is %g Nm\n',I)
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

```

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

```
Vr=220; //Rated Voltage
N=1500; //Rated Speed

Ra=0.25; //Armature Resistance
Ia=20; //Rated Armature Current
Vs=250; //Supply Voltage
Kaphi=0.0278; //Motor Voltage Constant
a=30; //Firing Angle

Va=2*sqrt(2)*Vs*cosd(a)/%pi; //Armature Voltage
Eb=Va-(Ia*Ra); //Back EMF
w=Eb/(Kaphi*60); //Angular Frequency in Radians per second
Ta=Kaphi*Ia*60; //Torque

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 fi

```

//To Determine the motor current and torque for a particular firing angle
//Page 242
clc;
clear;
P=15*735.5; //Power Rating of the Motor
Raw=0.2; //Combined Armature and Field winding resistance
N=1000; //Speed
K=0.03; //Motor Constant
a=30; //Firing Angle
E=250; //Supply Voltage

```

```
//Semi Converter
w=2*%pi*N/60; //Angular Frequency
Vas=sqrt(2)*E*(1+cosd(a))/%pi;
Ias=Vas/(Raw+(K*w)); //Armature Current
Tas=K*(Ias^2); //Motor Torque

//FullConverter
Vaf=2*sqrt(2)*E*cosd(a)/%pi;
Iaf=Vaf/(Raw+(K*w)); //Armature Current
Taf=K*(Iaf^2); //Motor Torque

//Accurate Caclculation, Rounding Off not done.
printf('The Motor Current And Motor Torque for:\n')
printf('i) Semi Converter : %g A and %g Nm\n',Ias,Tas)
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

```

//To determine the pulse width for a particular average value of back EMF
//Page 244
clc;
clear;
E=220;//DC Supply
Raw=0.2; //Combined Armature and Field Winding Resistance
Ia=25;//Average Current
f=200;//Chopper Frequency
Eb=100; //Average Value of Back EMF

Eav=(Ia*Raw)+Eb; //Average Load Voltage
Ton=Eav*1000/(E*f); // Pulse Width in m sec
```

```
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

```

//To determine the range of speed control and duty
   cycle
//Page
clc;
clear;
E=220; //DC supply voltage
Ra=0.2;
Kaphi=0.08; // Motor Constant
Ia=25; //Average Current
Eav=Ia*Ra; //Voltage Drop

//Eav = lamda * E;
//For Maximum Speed, lamba =1

lamda=Eav/E;

lamda1=1; //Maximum Speed;
Eb= (E*lamda1)-(Ia*Ra); //Back EMF

Speed = Eb/Kaphi; // Maximum Speed

printf('i) The Range of speed control is 0<N<%g \n',
   Speed)
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

```
//To determine the size and length of the wire
//Page 282
clc;
clear;

r=poly(0,'r'); //Variable Value of radius
A=%pi*(r^2); //Area of cross section of the wire
V=220; //Supply Voltage
P=20*(10^3); //Power input
//Temperatures
T1=1127; //Wire
T2=427; //Charge

R=(V^2)/P; //Resistance of the wire
e=0.9; //emissivity constant
K=0.6; //Radiation Effciency

p=1.09*(10^-6); //Resistivity

l=R*A/p; //Length in term of 'r'
```

```
H=5.72*e*K*((((T1+273)/100)^4)-(((T2+273)/100)^4));
    //Heat dissipated per sq.m of the surface
CSA=%pi*2*r*l; //Curved surface area

CSAn=P/H; //Numerical Value of Curved suraface area

X=CSA-CSAn; //Polynomial to find 'r'

disp(X);
printf('The real roots of the above equation gives
    the value of the radius\n')
r=roots(X); //Numerical Value of radius
disp(r);
printf('Choosing the real roots from the list above\
    n')
r=r(3);
l=CSAn/(2*%pi*r); //Numerical Value of length

//For charge temperature to be cold
Ti=25; //Cold Temperature
T=poly(0,'T'); //Variable value of the element
    temperature
Hi=5.72*e*K*((((T+273)/100)^4)-(((Ti+273)/100)^4));
    //Heat dissipated per sq.m of the surface
CSA=%pi*2*r*l; //Curved surface area
Y=Hi-H; //polynomial to find the temperature of the
    element

//Roots of T must be real
T=roots(Y); //Numerical Value
disp(Y);
printf('The real roots of the above equation gives
    the value of the element temperature\n')
disp(T);
printf('Choosing the real roots from the list above\
    n')
T=T(4);
```

```
printf('\n\nThe length and radius of the wire element are %g m and %g mm respectively\n',l,(r*1000))
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

```

//To determine the various temperature by changing the connection of the resistance elements
//Page 284
clc;
clear;

//Note that the value in kelvin of the first case in the textbook is wrong

//P is directly proportion to V^2 and H is directly propostional to KT^4
//Different Temperatures for different configurations
T1=1125; //Temperature in First Case
T2=poly(0,'T2');
T3=poly(0,'T3');
T4=poly(0,'T4');

//Multiplying Factors to the square of voltages
V1=1; //Line to Line Voltage;
V2=V1/2; //when connected in series first and then delta
V3=V1/(2*sqrt(3)); //when connected in series and then in star
V4=V1/(sqrt(3)); //When connected in parallel and in
```

```
star

//To find the power loss in each case
deff('x=pow(y)','x=(y^2)');
P1=pow(V1);
P2=pow(V2);
P3=pow(V3);
P4=pow(V4);

//To find the heat dissipated from each case
deff('x=heatdiss(y)','x=(y^4)');
H1=heatdiss(T1+273);
H2=heatdiss(T2+273);
H3=heatdiss(T3+273);
H4=heatdiss(T4+273);

//Polynomials to find the temperature in degree celsius
deff('x=temp(y,z)','x=(P1/y)-(H1/z)');
X2=temp(P2,H2);
X3=temp(P3,H3);
X4=temp(P4,H4);

//Temperature Numerical Value
T2=roots(X2(2));
T3=roots(X3(2));
T4=roots(X4(2));

//Only to consider Real Roots
T2=T2(4);
T3=T3(4);
T4=T4(4);

printf('The Temperature for the following configurations are:\n')
printf('Two Groups connected in series first and then in delta : %g degree Celsius\n',T2)

```

```
printf('Two Groups connected in series first and
    then in star : %g degree Celsius\n',T3)

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

```

//To Determine the average KW input to the furnace
//Page 288
clc;
clear;
M=10*(10^3); //Mass of Steel Melted
t=2*3600; //Time Taken to Melt the steel
eff=50/100; //Overall Efficiency
I=9000; //Current Input
R=0.003; //Resistance
X=0.005; //Reactance
SH=0.12; //Specific Heat
LHF=8.89*(10^3); //Latent Heat of Fusion
Tm=1371; //Melting Point
Ti=20; //Room Temperature

Hm=M*LHF; //Heat Required for melting
Hr=M*SH*(Tm-Ti)*1000; //Heat Required to raise the
    temperature
Ht=Hm+Hr; //Total Amount of heat required

E=Ht*4.2/(3600); //Energy in Whr
P=E*3600/t; //Power

Pa=P/eff; //Actual Power Input to the Furnace

Vt=Pa/(3*I); //V Cos theta
```

```
//The Above voltage is the sum of arc drop and drop
    in resistance load

Va=Vt-(I*R); //Arc Drop
Vx=I*X; //Reactance Drop
Vs=sqrt((Vt^2)+(Vx^2)); //Supply Voltage
S=3*Vs*I/1000; //KVA input

printf('The Average kW input to the furnance is %g
   kW\n',Pa/1000)
printf(' The Arc Voltage is %g V\n',Va)
printf(' The kVA input is %g kVA\n',S)

//Accurate Answers, No rounding off.
```

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

```

//To determine the effciency of a high frequency
    induction furnance
//Page 294
clc;
clear;
t=10*60; //Time Taken to rise temperature in seconds
M=1.815; //Mass of aluminium melted
Pi=5*(10^3); //Power Input
Ti=15; //Initial Temperature
Tm=660; //Melting Point of Al
SHAl=0.212; //Specific heat of Al
LHFAl=76.8*(10^3); //Laten Heat of fusin in Cal/Kg

Hm=M*LHFAl; //Heat required to melt Al
Htr=SHAl*M*1000*(Tm-Ti); //Heat required to raise
    the temperature
HTot=Hm+Htr; //Total Heat Required
```

```
HToth=HTot*3600/t; //Heat required per hour

Po=HToth*4.2/3600; //Power Output

eff=Po*100/Pi; //Efficiency

printf('The Effciency 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 curre

```

//To Determine the equivalent resistance of the charge and current
//Page 294
clc;
clear;

f=960; //Frequency
N1=20; //Primary Turns
N2=1; //Secondary is Single Turn
Pi=325*(10^3); //Power Input
Di=45; //Internal Diameter
l=50; //Depth of the charge

//Assumptions
p=200*(10^-6); //Resistivity
M=1; //For Molten Steel

t=(1/(2*%pi))*sqrt(p*(10^9)/(M*f)); //Depth of penentration of the current
A=t*l; //Effective Area
Dm=Di+t; //Mean Diameter
Dmcf=%pi*Dm; //Mean Length of current flow
```

```
Rc=p*Dmcf/A; //Resistance of the Cylinder

Is=sqrt(Pi/Rc); //Current flowing through secondary
Ip=Is*N2/N1; //Primary Current

printf('The Equivalent Resistance of the cylinder is
    %g * 10^-6 ohm\n',Rc/(10^-6))
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

```

//To Detemine power absorbed and the power factor
//Page 295
clc;
clear;
Vs=15; //Secondary Voltage
P=500*(10^3) // Power Taken
pfs=0.6; //Power Factor

Is=P/(Vs*pfs); //Secondary Current

//Taking Current as Reference voltage will be
t=acosd(pfs); //Power Factor Angle
Vsp=Vs*(exp(%i*(t/180)*%pi)); //Phasor Secondary
    Voltage
R=Vsp/Is; //Impedance

//if the resistance is doubled, The Total impedance
    doubles, Considering Vs as reference
R2=real(R)+R;
I2=Vs/R2; //New Current
pfn=cosd(atand(imag(I2)/real(I2))); //power factor
    of new current
```

```
Pab=Vs*abs(I2)*pfn/1000; //Power Absorbed

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

```

//To determine the Voltage Required and Current
   Drawn
//Page 299
clc;
clear;
t=2*(10^-2); //Thickness
A=150*(10^-4); //Area of the slab
Er=4; //Relative Permittivity
pf=0.04; //Power Factor
f=30*(10^6); //Frequency of supply
w=2*%pi*f; //Angular Frequency
P=200; //Power Required
Eo=8.854*(10^-12); //Permittivity of free space

C=Er*Eo*A/t; //Capacitance
Xc=1/(C*w); //Capacitative Reactance
phi=acosd(pf); //power factor angle
R=tand(phi)*Xc; //Resistance
V=sqrt(P*R); //Voltage
I1=V/R; //Current
Ic=V/Xc; //Curent through the Capacitor
It=sqrt((I1^2)+(Ic^2)); //Total Current

Vn=600; //Limited Voltage
Rn=(Vn^2)/P; //New Resistance
wn=tand(phi)/(C*Rn); //New Angular Frequency
fn=wn/(2*%pi); //New Frequnency
```

```

printf('The Current And Voltage are %g A and %g V
    respectively\n',It,V)
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

```

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

```
del=(90-phi)*%pi/180;//In Radian
V=sqrt(AP/(w*C*del)); //Voltage
I=V*w*C; //Current

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

```
// To determine the distance of a 25 cp lamp for
    various illumination
//Page 327
clc;
clear;

//Candle power of the lamp
I=25;

// Various illumination levels
E1=5;        //Case1
E2=15;       //Case2
E3=8;        //Case3

// According to the law of illumination  E = I/(r^2)
    ;
// Using the above equation we find the distances
    for the above three illuminations

r1= sqrt(I/E1);
```

```
r2= sqrt(I/E2);
r3= sqrt(I/E3);

printf('a) The distance for %g flux illumination
   from the normally placed screen is %g m \n',E1,r1
   )
printf('b) The distance for %g flux illumination
   from the normally placed screen is %g m \n',E2,r2
   )
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

```

//To determine the total radiation sent vertically
   downward from a lamp of 1500 cp
//Page 327
clc;
clear;

//Candle power of the lamp
P=1500;

//Since the flux required corresponds to the one
   that lies in a plane passing through the lamp
   vertically downwards the angle is given by
Angle= %pi; // Angle in radians

Flux=P*Angle;

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 illuminat

```

//To calculate Spherical Candle Power and intensity
   of illumination
//Page 327
clc;
clear;

Tf= 1500;                                          //Total
   flux
D=3                                                //
   Distance from the source
CP=200;                                            //Candle
    power of the Uniform Source
E=CP/(D^2);                                        //
   Maximum Illumination Intensity
deff('y=IntIllumi(x)','y=E*sind(x)');              //
   Function to determine the intensity of
   iluumination for various conditions


MSCP=Tf/(4*%pi);                                   //The
   Mean spherical candle power

//Various Angles at which the rays falls on the
   surface
t1=90;
t2=60;
t3=0;

//Illumination Intensities at various angles
E1=IntIllumi(t1);
E2=IntIllumi(t2);
```

```
E3=IntIllumi(t3);

printf('a)The Mean Spherical Candle Power is %g c.p\
   n',MSCP)
printf('b The illumination intensities for the
   following cases are: \n')
printf('i)   Normal : %g lux\n',E1)
printf('ii)  Inclined at 60 degrees : %g lux\n',E2)
printf('iii) Parallel : %g lux\n',E3)
```

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

```

//To determine average illumination and illumination
    at various points on the area
//Page 328
clc;
clear;

CP=200;                    //Candle power of the lamp
D=4;                       //Vertical Distance from the
    centre of the area to the lamp
R=5/2;                     //Radius of the given area
De=sqrt((D^2)+((R)^2));    //Distance from the edge
   of the area
Reff=80/100;               //Reflector Efficiency

Ec= CP/(D^2);              //Illumination at the centre
    of the area
theta= acosd(D/De);        //Angle made between
   perpendicular and edge distances
w=2*%pi*(1-cosd(theta));//Solid angle subtended by
   the area

//Assuming uniform intensity in all directions
```

```
//Illumination at the edge is
Eue=(CP/(De^2))*cosd(theta);
//Flux is given by Iw
flux=CP*w;
//Average illumination
Euavg= flux/(%pi*(R^2));
//Average illumination with reflector efficiency
Euavgr= Euavg*Reff;

//When candle power is only vertically downwards
Eve=Eue*cosd(theta);    //Illumination at the edge of the area

//Total flux equation for non uniform intensity is derived and given by 2*pi*I*(sec(theta)-1)
fluxn=2*%pi*CP*(secd(theta)-1);
//Average illumination
Enuavg= fluxn/(%pi*(R^2));
//Average illumination with reflector efficiency
Enuavgr= Enuavg*Reff;

printf('The illumination at the \n')
printf('i) Centre of area = %g lux\n',Ec)
printf('ii)Periphery of the area \n')
printf('a) Uniform illumination = %g lux\n',Eue)
printf('b) Candle power is vertically downward = %g lux\n',Eve)
printf('iii) Average illumination and Average illumination with reflector efficiency\n')
printf('a) Uniform intensity = %g lux and %g lux respectively.\n',Euavg,Euavgr)
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`

```
//To determine illumination at a point when a mirror is used
//Page 330
clc;
clear;

CP=500;                      //Candle power of the lamp
Vd=5;                        //Vertical distance from the point to the source
Hd=5;                        //Horizontal distance fron the point to the source
D=sqrt((Vd^2)+(Hd^2));       //Distance between the source and the point
Md=2;                        //Distance of the source from the mirror
theta=atand(Hd/Vd);          //Angle made between the source to point and the horizontal

//Lets consider the mirror reflection
R=80/100;                    //Reflected rays ratio
Vdm=Vd+(2*Md)                //Vertical Distance from the point to the source reflection
Hdm=Hd;                      //Horizontal distance from the point to the source reflection
Dm=sqrt((Vdm^2)+(Hdm^2));//Distance between the source reflection and the point
thetam=atand(Hdm/Vdm);       //Angle made between the source reflection to point and the horizontal

//Illumination at A due to source
Es=(CP/(D^2))*cosd(theta);

//Illumination at A due to source refelction
Er=(R*CP/(Dm^2))*cosd(thetam);

```

```
//Total Illumination at point A
E=Er+Es;

printf('The total illumination at point A is %g lux
    .\n',E)
```

Scilab code Exa 6.7 To determine the illumination over a disc

```

//To determine the illumination over a disc
//Page 330
clc;
clear;

MSCP=300;                              //Uniform
    intensity of the lamp or Mean Spherical candle
    power
Vd=6;                                  //Vertical
    distance of the lamp from the disc
R=6/2;                                 //Radius of the
    disc
Reff=60/100;                           //Reflector
    efficiency
D=sqrt((Vd^2)+(R^2));                  //Distance from
    the source to the edge of the disc
theta=atand(R/Vd);                     //Angle made
    between D and Vd

Ecwr=MSCP/(Vd^2);                      //Illumination at
    the centre without reflector
Eewr=(MSCP/(D^2))*cosd(theta);         //Illumination at
    the edge without reflector

//Illumination at the centre is equal to the
    illumination at the egde with a reflector
```

```
Eer=MSCP*Reff*(4*%pi/(%pi*(R^2)));
Ecr=Eer;

w=2*%pi*(1-cosd(theta));            //Solid angle made by the surface
flux=MSCP*w;                        //Total flux produced by the source
Eavg=flux/(%pi*(R^2));              //Average illumination

printf('The illumination at:\n')
printf('a) At the centre:\n')
printf('i) With reflector : %g lux\n',Ecr)
printf('ii)Without reflector : %g lux\n',Ecwr)
printf('b) At the edge of the surface:\n')
printf('i) With reflector : %g lux\n',Eer)
printf('ii)Without reflector : %g lux\n',Eewr)
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

```

//To estimate the number rating and disposition of the lamps
//Page 338
clc;
clear;

E=32;                     //Illumination required for the working plane
A=80*15;                  //Area of the work bench
UF=0.5;                   //Utilization Factor
MF=0.8;                   //Maintenance Factor
SHR=1.5;                  //Maximum permissible value
```

```
   of spacing to height ratio
h=4.5;                   //Required height for the
   lamps to be hung above the work bench
Leff=14;                 //Lamp efficacy
Tlumen=E*A;              //Total lumens required
Llumen=Tlumen/(UF*MF);   //Lamp lumens required
l=(A/15);                //Length of the workspace

Nc=ceil(l/(1.5*4.5));     //Minimum number of lamps
   in a single row (Number of columns)

W=200;                   //Assumed wattage of the
   bulb

NoL=Llumen/(W*Leff);     //Number of bulbs required,
   Calculated value

Nr=ceil(NoL/12);         //Number of rows calculated
   for the required criteria

N=Nc*Nr;                 //Number of lamps necessary.

Sp=l/Nc;                 //Length wise spacing
   between the lamps
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

```

//To determine the number of lamps required
//Page 344
clc;
```

```
clear;

A=30*20;                  //Area of the building
B=25;                     //Brightness in lumen/sq.metre
CoR=0.25;                 //Co - efficient of reflection
Lwatt=500;                //Lamp Wattage
Ll=8000;                  //Lamp lumens output
E=B/CoR;                  //Illumination required
BF=0.6;                   //Beam factor
WF=1.2;                   //Waste light factor
MF=0.75;                  //Mainteanance factor

Tlumen= E*A*WF/MF;        //Total lumens required

Llumen=Ll*BF;             //Lumens provided by one lamp

NoL=Tlumen/Llumen;        //Number of lamps

printf('The number of lamps required are %g.\n',round(NoL))
```

Scilab code Exa 6.10 To determine the illumination on the ground

```

//To determine the illumination on the ground
//Page 344
clc;
clear;

CP=300;                   //Uniform luminous intensity of a lamp
Vd=6;                     //Vertical distance from the
```

```
   ground to the lamp
Hd=6;                    //Horizontal distance from
   the ground to the lamp
D=sqrt((Vd^2)+(Hd^2));   //Distance from the lamp to
   point 6 metres away from the vertical line
theta=atand(Hd/Vd);      //Angle made between D and
   Vd.

// Illumination
Ec=CP/(Vd^2);            //Vertically beneath the
   lamp
E6=CP*cosd(theta)/(D^2);//^=6 metres away from the
   centre

printf('The illumination on the ground :\n')
printf('a)Vertically beneath the lamp : %g lux.\n',
   Ec)
printf('b)6 metres away from condition a) : %g lux.\
   n',E6)
```

Scilab code Exa 6.11 Determine the output voltage

```

//Determine the output voltage
//Page 344
clc;
clear;

Area=(5.08*3.75)*(10^-4);        //Area projected by
   the cathode
Is=12*(10^-6);                   //Sensitivity
Rl=1.5*(10^6);                   // Load of operation

//Function to calculate the output voltage for each
   case
```

```
deff('y=volt(a,b)','y=Area*a*Is*Rl/(b^2)');

//Case 1
CP=60;                                  //Lamp intensity
D=1.8;                                  //Vertical distance of the lamp from the cell
V1=volt(CP,D);
//Case 2
CP=6;                                   //Lamp intensity
D=0.5;                                  //Vertical distance of the lamp from the cell
V2=volt(CP,D);
//Case 3
W=100;                                  //Wattage of the lamp
eff=20;                                 //eficacy of the lamp
CP=W*eff/(4*%pi);                       //Lamp intensity
D=2;                                    //Vertical distance of the lamp from the cell
V3=volt(CP,D);

printf('The voltage output of the cells are :\n')
printf('a) 60 CP lamp at 1.8m : %g V\n',V1)
printf('b) 6 CP lamp at 0.5m : %g V\n',V2)
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`

```

//To determine the Acceleration and Coasting Period
//Page 362
clc;
clear;
D=1.92
Ts=20; //Duration of stops
Vsch=40;
V1=60.8;
//Retardation
B=3.2;
Bc=0.16;

SchTime=D*3600/Vsch; // Schedule time in seconds

T=SchTime-Ts; //Time of Travel

a=poly(0,'a'); // Acceleration Unknown Value

V2=(V1-Bc*(T-(V1/a)))/(1-(Bc/B)); // From the Speed Time Curve
```

```
X=((V1+V2)*T-(V1*V2*((1/B)+(1/a))))-(7200*D); //
    Polynomial Equation to find a

disp(X(2))
printf('\nThe Above Equation is Equated to zero to
    get a\n\n\n')
a=roots(X(2)); //Roots of the Characteristic
    Equation
//To Determine the positive roots
Y=abs(a)+a;
if(Y(1)==0)
    a=a(2);
else
    a=a(1);
end

Ta=V1/a; //Duration of acceleration
V2=(V1-Bc*(T-(V1/a)))/(1-(Bc/B)); // From the Speed
    Time Curve
Tcs=(V1-V2)/Bc; //Coasting Time Period

printf('The Acceleration of the Train is %g Kmphps\n
    ',a)
printf(' The Coasting Period is %g seconds\n',Tcs)
```

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

```

//To determine the duration of acceleration and
    others
//Page 363
clc;
clear;

```

```
D=1.6;
Vav=40;
Bc=0.16;
B=3.2;
a=2; //Acceleration

T=D*3600/Vav; //Time of travel

V1=poly(0,'V1'); //Variable Value

V2=(V1-Bc*(T-(V1/a)))/(1-(Bc/B)); // From the Speed Time Curve
X=((V1+V2)*T-(V1*V2*((1/B)+(1/a))))-(7200*D); // Polynomial Equation to find V1
V1=roots(X); //Numerical Values for V1
disp(X);
printf('\nThe Roots of the above eqaution are\n');
disp(V1)

V1=V1(2); //Train Considered have speed close to this value
V2=(V1-Bc*(T-(V1/a)))/(1-(Bc/B)); // From the Speed Time Curve

Ta=V1/a; //Acceleration Time
Tc=(V1-V2)/Bc; //Coasting Time
Tb=V2/B; //Braking time
//Distance are calculated according to the area under their respective curves
Da=(Ta*V1/2)/3600; //Distance during acceleration
Dc=(((V1-V2)*Tc/2)+(V2*Tc))/3600; //Distance during coasting
Db=(Tb*V2/2)/3600; //Distance during braking

printf('\n\nThe Time periods and the distance covered for:\n')
printf('Acceleration : %g seconds and %g km\n',Ta,Da)
```

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

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

```

//To determine the acceleration required to run the service
//Page 364
clc;
clear;

D=1;
Ts=20; //Stopping Time
B=3;
Vsh=30;
ShT=D*3600/Vsh; //Schedule time
T=ShT-Ts; //Actual Run Time
Vav=D*3600/T; //Average Speed
Vm=1.25*Vav;
a=poly(0,'a'); //Acceleration Variable

X=((2*Vm*T)-((Vm^2)*((1/a)+(1/B))))-(7200*D); //Polynomial Equation to find 'a'

disp(X(2));
printf('\n The above equation is equated to zero to find the Acceleration\n\n')

a=roots(X(2)); //Numerical Value of the Acceleration

printf('The Accleration 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 particula

```
//To determine the time taken by the train to attain a particular speed
//Page 366
clc;
clear;
M=200; //Mass of train
g=9.81; //Acceleration due to gravity
W=M*g;
RI=10/100; //Rotational Inertia
We=W*(1+RI);
motor=4; //Number of Motors
Tm=5000; //Torque per motor
T=Tm*motor;
V=40; //Speed to be Attained
N=0.9; //Gear Efficiency
Y=3.6/1; //Gear Ratio
R=91.5*(10^-2)/2;
Ft=N*T*Y/R;
r=40;
G=(1/200)*100; //Gradient in Percentage
a=poly(0,'a'); //Acceleration

X=((28.3*We*a)+(10*W*G)+(M*r))-Ft; //Polynomial Equation to find acceleration

a=roots(X); //Numerical Value of Acceleration
Time=V/a; //Time Taken to attain the required the necessary speed

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

```
//To determine the Factors affeting the mechanics of
    the Train
//Page 373
clc;
clear;
D=1.92
Ts=20; //Duration of stops
Vsch=40;
V1=60.8;
//Retardation
B=3.2;
Bc=0.16;
SchTime=D*3600/Vsch; // Schedule time in seconds
T=SchTime-Ts; //Time of Travel
a=poly(0,'a'); // Acceleration Unknown Value
V2=(V1-Bc*(T-(V1/a)))/(1-(Bc/B)); // From the Speed
    Time Curve
X=((V1+V2)*T-(V1*V2*((1/B)+(1/a))))-(7200*D); //
    Polynomial Equation to find a
a=roots(X(2)); //Roots of the Characteristic
    Equation
//To Determine the positive roots
Y=abs(a)+a;
if(Y(1)==0)
    a=a(2);
else
    a=a(1);
end
Ta=V1/a; //Duration of acceleration
V2=(V1-Bc*(T-(V1/a)))/(1-(Bc/B)); // From the Speed
    Time Curve
```

```
Tc=(V1-V2)/Bc; //Coasting Time
Tb=V2/B; //Braking time
//Distance are calculated according to the area
   under their respective curves
Da=(Ta*V1/2)/3600; //Distance during acceleration
Dc=(((V1-V2)*Tc/2)+(V2*Tc))/3600; //Distance during
   coasting
Db=(Tb*V2/2)/3600; //Distance during braking

r=4.53*9.81; //Train Resistance in N per tonne
M=200; //Mass of Train
AWF=1.1; //Accelerating Weight Factor
SEOA=0.010726*(V1^2)*(AWF)/D; //Specific Energy
   Output during acceleration
SEOAr=0.2778*r*Da/D; //Specific Energy Output during
    acceleration against train resistance
SEO=SEOA+SEOAr; //Specific Energy for the run
SEOB=0.010726*(V2^2)*(AWF)/D; //Specific Energy
   Output during braking
SEOC=SEOA-SEOB; //energy utilized during coasting
rc=SEOC*D/(0.2778*Dc)//Mean Train Resistance during
   coasting

printf('\ni)The Specific Energy Output for the Run
   is %g Whr/Tonne-km\n',SEO)
printf('ii) The Energy Dissipated by the brakes is
   %g Whr/Tonne-km\n',SEOB)
printf('iii)The Energy Utilized during coasting is
   %g Whr/Tonne-km\n',SEOC)
printf('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

```
//To determine the Specific Energy Output Runs for
    Both the Cases
//Page 374
clc;
clear;
G=(1/80)*100;
W=210;
We=1.1*W;
Vm=poly(0,'Vm'); //Variable Value of The Max
    Velocity
D=1.6; //Distance between Stations
Ftg=10*W*G; //Tractive Effort due to Gradient
Fta=13050; //Total Tractive Effort for Acceleration
Ftb=20445; //Total Tractive Effort for Retardation
ron=5.43; //Train Resistance when power on
rc=6.525; //Train Resistance when coasting
//Level Track
al=2;
Bl=3.2;

//Up Gradient
Vavu=40; //Average Speed
Tu=D*3600/Vavu; //Total Time Taken
au=(Fta-Ftg)/(28.3*We); //Acceleration
Bu=(Ftb+Ftg)/(28.3*We); //Retardation
Xu=((2*Vm*Tu)-((Vm^2)*((1/au)+(1/Bu))))-(7200*D); //
    Polynomial Equation to Find Vm
Vmu=roots(Xu); //Numerical Value of Vm
Zu=Vmu(1)-Vmu(2); //To detemine which root to take
if(abs(Zu)+Zu==0)
    Vmu=Vmu(1);
else
    Vmu=Vmu(2);
end
Tua=Vmu/au; //Accelerating Period
Tub=Vmu/Bu; //Braking Period
Tucs=Tu-Tua-Tub; //Constant Speed Duration
Dua=Vmu*Tua/(2*3600); //Distance Travelled During
```

```
    Accleration in km
Dub=Vmu*Tub/(2*3600); //Distance Travelled During
    Braking in km
Duon=D-Dub; //Distance run with power on
SEOua=0.010726*(Vmu^2)*(We/W)/D; //Specific Energy
    Output during Acceleration
SEOug=27.25*G*Duon/D; //Specific Energy Output for
    Gradient
SEOur=0.2778*ron*9.81*Duon/D; //Specific Energy
    Output for resitance
TSEOu=SEOua+SEOug+SEOur; //Total Specific Output

//Down Gradient
Vavd=44; //Average Speed
Td=D*3600/Vavd; //Total Time Taken
ad=(Fta+Ftg)/(28.3*We); //Acceleration
Bd=(Ftb-Ftg)/(28.3*We); //Retardation
Xd=((2*Vm*Td)-((Vm^2)*((1/ad)+(1/Bd))))-(7200*D); //
    Polynomial Equation to Find Vm
Vmd=roots(Xd); //Numerical Value of Vm
Zd=Vmd(1)-Vmd(2); //To detemine which root to take
if(abs(Zd)+Zd==0)
    Vmd=Vmd(1);
else
    Vmd=Vmd(2);
end
Tda=Vmd/ad; //Accelerating Period
Tdb=Vmd/Bd; //Braking Period
Tdcs=Td-Tda-Tdb; //Constant Speed Duration
Dda=Vmd*Tda/(2*3600); //Distance Travelled During
    Accleration in km
Ddb=Vmd*Tdb/(2*3600); //Distance Travelled During
    Braking in km
Ddon=D-Ddb; //Distance run with power on
SEOda=0.010726*(Vmd^2)*(We/W)/D; //Specific Energy
    Output during Acceleration
//Net Force Acting Downward due to gradient and
    resistance
```

```
Fnet=W*((10*G)-rc);
//Since Fnet is Postive , To run the train at
    constant speed brakes will have to applied
    therefore they cannot supply electric energy
TSEOd=SEOda; //Total Sepcific Enegy Consumption

printf('The Total Specific Energy:\n')
printf('Up Gradient : %g Whr/Tonne km\n',TSEOu)
printf('Down Gradient : %g Whr/Tonne km\n',TSEOd)
```

Scilab code Exa 7.7 To determine the specific energy consumption

```

//To determine the specific energy consumption
//Page 376
clc;
clear;
W=400;
G=1;
a=1.5; //Acceleration
Ta=30; //Acceleration Time
Tf=36; //Free running Period
Tc=25; //Coasting Period
B=2.6;
r=45;
RI=10/100; //Rotational Inertia Effect
Eff=75/100; // Overall Efficieny
g=9.81; //Accleration due to gravity

Vm=Ta*a;
We=W*(1+RI);

//Distance Covered is equal to the area under the
    speed time curve
Da=Vm*Ta/(2*3600); //Acceleration
```

```
Df=Vm*Tf/3600; //Free Run

Ftf=W*(r+(10*g*G)); //Tractive effort during free run period

Fta=We*277.8; //Tractive effort due to acceleration

//During Coasting, The Accelrating force is equal to Tractive effort during free run
//Retardation due to coasting is
Bc=Ftf/Fta;

V2=Vm-(Bc*Tc); //Speed of train after the coasting period
Tb=V2/B; // Braking period

Dc=((((Vm-V2)*Tc)/2)+(V2*Tc))/3600; //Distance covered during coasting
Db=V2*Tb/(2*3600); //Distance covered during braking

D=Da+Df+Dc+Db; //Total Distance

D1=Da+Df; //Distance for which the energy is being spent

SEO=(0.010726*(Vm^2)*We/(W*D))+(27.25*G*D1/D)+(0.2778*r*D1/D); //Specific Energy Output
//Note the Calculated Specific Energy Output during accleration is wrong.

SEOa=SEO/Eff; //Actual Specific Energy Output with 75% efficieny

printf('The Specific Energy Consumption is %g Whr/Tonne Km\n',SEOa)
```

Scilab code Exa 7.8 To determine the weight of the locomotive

```
//To determine the weight of the locomotive
//Page 378
clc;
clear;
Wg=300; //Weight of the train to be hauled
RI=10/100; //Rotation inertia
Ma=20/100; //Co-Efficient of adhesion
Wa=20; //Permissible Weight of axle load
r=45;
G=2;
a=1;  //Acceleration
Wl=poly(0,'Wl'); //Variable  Weight of locmotive
W=Wg+Wl; //Total Weight of the train
We=W*(1+RI);
Ft=((277.8*a*We/W)+(98.1*G)+r)*W;

Fmax=(9.81*1000*Ma*Wl);

X=Ft-Fmax; //Polynomial Equation to find Wl

Wl=roots(X(2)); //Numerical Value of the Weight of
    the locomotive

NoA=Wl/Wa; //Number of axles

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

```
//To deduce the speed current characteristic of the
   motor
//Page 380
clc;
clear;
Ia=[10,20,30,40,50,60,70]; //Current
T=[45,130,230,350,470,610,765]; //Torque
V=440; //operating voltage
r=0.5; //Armature Circuit Resistance

N=9.55*(V-(Ia*r))./(T./Ia); //Speed

printf('For the Given Current:\n')
disp(Ia);
printf('\nThe Speeds tabulated are:\n')
disp(N);
```

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

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

```
P=4; //Poles of the machine
N1=600; //Series Motor Speed
Vc=600; //Constant Operating Voltage

// 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

Eb2=Vc-(I.*(Rt));
N2=N1.*Eb2./Eb1; //Speed

T=9.55.*Eb2.*I./N2; //Torque

printf('The Tabulated Speeds for 600V are:\n')
disp(N2)
printf('\nAnd their corresponding torques are:\n')
disp(T)

plot(T,N2)
xlabel('Torque')
ylabel('Speed')
```

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

```

//To determine the horse power delivered by the locomotive
//Page 388
clc;
clear;

Ftl=35000; //Tractive Effort on a level surface
Ftg=55000; //Tractive Effort on a gradient
```

```
V=50; //Speed of the train
HP=735.5; //One Horse Power
Pl=Ftl*V*1000/(3600*HP); //Power Output(in HP) of
    the Locomotive on the level track

//CASE 1:
//D.C Series Motor, Power is directly proportional
    to the root of the Trative Effort
HP1=Pl*sqrt(Ftg/Ftl);

//CASE 2:
//3 Phase Induction Motor, Power is directly
    proportional to the Trative Effort
HP2=Pl*(Ftg/Ftl);

printf('The Horse Power delivered by the locmotive
    when the motors used are:\n')
printf('i) D.C Series Motor is %g H.P \n',HP1)
printf('ii) 3 Phase Induction Motor is %g H.P \n',
    HP2)
//Note the Calculation Mistake in TextBook for case
    2
```

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

```

//To Determine the tractive effort shared by the two
     locomotives
//Page 388
clc;
clear;

Da=1.27; //Loco A wheel diameter
Db=1.244; //Loco B wheel diameter
S=5; //Slip of both the Locomotives in Case 1
```

```
Sa=S; //Slip of Loco A in Case 2
Sb=4; //Slip of Loco B in Case 2
Ftta=11325; //Total Tractive Effort in Case 1
Fttb=1309; //Total Tractive Effort in Case 2
Ft=5227; //Full Load Tractive Effort of Loco

x=poly(0,'x'); //Variable for Tractive Effort Exerted by Loco A

//Slips are same
//For 11325 kg
//% Speed
NA=100-(S*x/Ft);
NB=100-(S*(Ftta-x)/Ft);
Rna=NB/NA; //Speed Ratio

Y1=Rna-(Da/Db); //Polynomial to find out 'x'
x=roots(Y1(2)); //Numerical Value of Tractive Effort shared by A

//Tractive Efforts Shared by A Loco and B Loco
FtAas=x;
FtBas=Ftta-x;

//For 1309 kg
//% Speed
x=poly(0,'x'); //Variable for Tractive Effort Exerted by Loco A
NA=100-(S*x/Ft);
NB=100-(S*(Fttb-x)/Ft);
Rn1=NB/NA; //Speed Ratio

Y2=Rn1-(Da/Db); //Polynomial to find out 'x'
x=roots(Y2(2)); //Numerical Value of Tractive Effort shared by A

//Tractive Efforts Shared by A Loco and B Loco
FtAbs=x;
```

```
FtBbs=Fttb-x;

//Different Slips
//For 11325 kg
//% Speed
x=poly(0,'x'); //Variable for Tractive Effort Exerted by Loco A
NA=100-(Sa*x/Ft);
NB=100-(Sb*(Ftta-x)/Ft);
Rna=NB/NA; //Speed Ratio

Y3=Rna-(Da/Db); //Polynomial to find out 'x'
x=roots(Y3(2)); //Numerical Value of Tractive Effort shared by A

//Tractive Efforts Shared by A Loco and B Loco
FtAad=x;
FtBad=Ftta-x;

//For 1309 kg
//% Speed
x=poly(0,'x'); //Variable for Tractive Effort Exerted by Loco A
NA=100-(Sa*x/Ft);
NB=100-(Sb*(Fttb-x)/Ft);
Rn2=NB/NA; //Speed Ratio

Y4=Rn2-(Da/Db); //Polynomial to find out 'x'
x=roots(Y4(2)); //Numerical Value of Tractive Effort shared by A

//Tractive Efforts Shared by A Loco and B Loco
FtAbd=x;
FtBbd=Fttb-x;

printf('The Tractive Effort shared A and B for:\n \n')
printf('i) Slips are %g percent for both the
```

```
Locomotives for a Tractive effort of:\n',S)
printf('a) %g kg : %g kg by A and %g kg by B respectively\n',Ftta,FtAas,FtBas)
printf('b) %g kg : %g kg by A and %g kg by B respectively\n\n',Fttb,FtAbs,FtBbs)
printf('i) Slips are %g percent for A and %g percent for B for a Tractive effort of:\n',Sa,Sb)
printf('a) %g kg : %g kg by A and %g kg by B respectively\n',Ftta,FtAad,FtBad)
printf('b) %g kg : %g kg by A and %g kg by B respectively\n\n',Fttb,FtAbd,FtBbd)

//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

```

//To determine the traction control of 2 motors rated at 1500V
//Page 396
clc;
clear;

NoM=2; //Number of motors
V=1500; //Rated Voltage
I=500; //Starting Current
R=0.15; //Armature Resistance
r=50; //Specific Resistance
W=120;
We=140;
Ft=38000; // Tractive Effort Per motor

Vs=40; //Speed at the end of starting period
//Note the question is asked for 50 Kmph, But the
```

```
answer is calculated for 40Kmph
//Therefore it is calculated for 40 Kmph
//By Changing the value of Vs to 50 Kmph, The Specific Parameters can also be obtained

a=poly(0,'a'); //Accleration Variable

X=(2*Ft)-((277.8*We*a)+(W*r)); //Polynomial To find 'a'

a=roots(X); //To find the Numerical Value of the Acceleration

t=Vs/a; //Starting Time

Vd=I*R; //Resistance drop per motor

ts=t*(V-(2*Vd))/(2*(V-Vd)); //Starting Series Time Period
tp=t-ts; //Starting Parallel Time Period

Vt=a*ts; //Speed at Transition
//Loss can be found out by the computing the area under the of the given figure in the text book for different period
//In series, the Voltage reduces, Hence 675;
Rhel=((((V/2)-Vd)*I*ts)+((V/2)*I*tp))/(2*3600*1000); //Loss per motor in kWhr
Rhe=NoM*Rhel; //Total Rheostatic Loss

printf('i) The Duration of starting period is %g seconds\n',t)
printf('ii)The Speed of Train at Transition is %g Kmph\n',Vt)
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

```
//To determine the speed of motors when connected in
    series
//Page 397
clc;
clear;
V=650; //Rating of motor and Supply Voltage when
   connected in parallel
IR=10*650/100; //Armature Drop
N1p=1000; //Speed of the first motor in parallel
   operation
D1=88; //Motor 1 wheels diameter
D2=86; //Motor 2 wheels diameter

//Current Remains Constant during the start

r=D1/D2; //Ratio of the first motor wheel diameter
   to the second motor wheel diameter

//N1/N2 = 1/r

V1=((V-IR)/(1+(r)))+((IR)/(1+(1/r))); //Voltage of
   Motor 1 in series operation
V2=V-V1; //Voltage of Motor 2 in series operation

N1=N1p*(V1-IR)/(V-IR); //Speed of motor 1 in series
   configuration
N2=N1*r; //Speed of motor 2 in series configuration

printf('The Speed of Motor 1 in series Configuration
    is %g rpm\n',N1 )
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`

```

//To determine the characteristics gear ratio and wheels are replaced
//Page 397
clc;
clear;

D1=86; //Diameter of wheels in Case 1
Y1=71/21; //Gear Ratio in Case 1
D2=88; // Diameter of wheels in Case 2
Y2=74/19; //Gear Ratio in case 2
I=[50,100,150,200,250,300]; //Current
L1=[80,50,45,40,36,32]; //Speed in Kmph
Ft1=[2000,6000,12000,16000,18000,22000]; //Tractive Effort

// V = pi*D*N*60/(100*100*Y)
//According to the above equation, V is directly proportional to D*N/Y

//Angular Frequency And Torque remains the same irrespective of the gear ratio or change in diameter

//Hence the V is directly proportional to D/Y

//V id directly proportional to N
// N directly proportional to Linear speed
//Hence relating the above terms
//We get linear speed directly proportional to D/Y

```

```
L2=L1.*D2*Y1/(D1*Y2); //Linear Speed in Case two

//Similarly T = Ft * D/(200)
//Dividing by Y ; (T/Y) = Ft*D/(200*Y)
//Taking two cases where we find Torque per gear ratio of one case
// And multiply the Torque to Gear ratio to the other gear ratio to get Torque
// From this process we T1*Y2/Y1 =T2
// We Get Ft is directly proportional to Y/D

Ft2=Ft1.*D1*Y2/(D2*Y1); //Tractive Effort in Second Case

printf('The New Characteristics are\n Current in Amperes\n')
disp(I)
printf('\nSpeed in Kmph\n')
disp(L2)
printf('\nTractive Effort in N\n')
disp(Ft2)
```

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

```

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

```

I=P/(V*Eff); //Maximum Current

//VAriation in Current
Imin=1.5*I;
Imax=2*I;
Rs=[]; //Total Resistance after each step
Rg=[]; //Step Resistance
FluxRatio=1+(Iflux); //Flux raises by 10% for every
    step

//Condition to make sure Imax doesnt go less than
    the Rated Value and also to find the number of
    steps and Step Resistances
for i= 1:1000
                Is=V/Raw;
    M=Imax-Is;
    if((abs(M)+M)==0)
        c=i; //Number of steps + 1
        R=poly(0,'R'); //Variable Resitance
    X=((V-(Imax*Raw))/(V-(Imin*R)))-FluxRatio; //
        Polynomial To Find The Next Resistance
    Rs(i)=roots(X(2)); //Total Next Resistance
    Rg(i)=Rs(i)-Raw; //Resistance of the ith Step
    Raw=Rs(i);
    else
        break;
    end
end

S=c-1; //Number Of steps
printf('\nThe Number of steps is %g\n\n\n',S)
for j=1:S
    printf('The Resistance of ''%g'' step is %g ohm\
        n',j,Rg(j))
end
```

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

```
//To detemine the characteristics for tapping of 30
   percent of the turns
//Page 400
clc;
clear;
I=[50,100,150,200,250]; //Current
N1=[73.5,48,41,37,35]; //Speed
Ft1=[131,457,810,1163,1525]; //Tractive Effort in Kg
Fl=70/100; //Effective Tapping of the Turns (30%
   reduction)

//Consider flux corresponding to 250A as 100%, that
   is for 35 Kmph

Flux=(35*100)./N1; //Flux Percent for Current Value

//Reduction by 30% of turns is same as reducing the
   Ampere Turns by 30% which is same as reducing the
    Current by 30%

RI=Fl.*I; //Reduced Current
RFlux=[32,57,74,83,90]; //Reduced flux determined
   From the Graph of %Flux and Current in the
   TextBook

N2=(N1.*Flux)./RFlux; //Speed for the new case

//Tractive Effort is directly proportional to the
   product of flux and current

Ft2=RFlux.*Ft1./Flux; //Tractive Effort of the
```

```
   Second Case

printf('The Current in Amperes:\n')
disp(I)
printf('\nThe Speed in Kmph\n')
disp(N2)
printf('\nThe Tractive Effort in Kg\n')
disp(Ft2)
```

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

```

//To determine the power fed into the supply system
//Page 410
clc;
clear;
W=200;
RI=10/100; //Rotational Intertia
Eff=75/100; //Effeciency of Conversion
G=2;
V1=60;
V2=20;
D=4000; //Distance covered during the speed change
   in m
r=40;
We=(1+RI)*W;

Ftb=W*((98.1*G)-r); //Tractive Effort during
   Retardation

KE=0.01072*(We/W)*200*((V1^2)-(V2^2))/1000; //K.E(in
    kWhr) Available due to change in speed

EquiEE=Ftb*D/(3600*1000); //Equivalent Electrical
   Energy
```

```

TE=KE+EquiEE; //Total Energy Regenarated

ERS=Eff*TE; //Energy Returned to the Supply

B=((V1^2)-(V2^2))/(2*D*3600); //Retardation
Tb=V1-V2/B; //time taken during retardation

PRS=ERS*3600/Tb; //Power Returned due to Retardation

//If there is no change in speed, Net Reactive force is Tractive Effort due to retardation

Vm=V1; //Maximum Speed

//Power that Can be fed into the system
PFS=0.278*Ftb*Vm/1000; //In Kilowatts

APFS=Eff*PFS; //Actual Power Fed Back into the supply system

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

```

//To determine the value of resistance to be connected in the motor circuit
//Page 411
clc;
clear;

N1=600; //Intial Speed
E1=[252,522,750,900,951]; // Voltages At 600 RPM
```

```
Ia=[20,40,60,80,100]; // Field Current
N2=400; //Changed Speed
E2=E1.*(N2/N1); // Voltages at 400 RPM
T=350;
Raw = 0.05; //Armature and Winding Resistance
printf('Field Current in A:\n')
disp(Ia)
printf('\nThe Respective Voltages at 400 rpm (V):\n'
    )
disp(E2)

P=E2.*Ia; //To find the value of E and Ia for the
    Required Speed

PI=2*%pi*N2*T/60; //Power Input

//To find the Portion of the Curve EIa vs Ia , under
     the currrent limits where our Power Input is
    located
for i = 1:5
    X=P(i)-PI; //Difference of Powers
    if(abs(X)+X==0)
        continue;
    else
        L=i-1; //Lower Limit
        H=i; // Upper limit
        break
    end
end

Ip=Ia(H)-Ia(L); //To find the Current Period
Im=Ia(L); //Starting Value of Current in the
    Particular Portion of the Curve

I=Im+(((PI-P(L))/(P(H)-P(L)))*Ip); //Current
    Required for 400 rpm
R=PI/(I^2); //Total Resistance required in the
    circuit
```

```
Re=R-Raw; //External Resistance

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

```
//To determine the quantity of current
//Page 450
clc;
clear;

//Parameters of the Copper Plating
TSA=0.3716; //Total Surface Area
t=0.0254*(10^-3); //Thickness of the plating
density=8.883*(10^3); //Density of Copper

W=TSA*t*density*1000; //Weight of copper deposited
    in gms
ECEcu=0.329; //Electrochemical Equivalent of copper
    in gm per coulomb
C=W*1000/ECEcu; //Quantity of Current required.

F=96500; //One Farad Charge

Q=C/F; //Quantity of Current in Faradays

```

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

Scilab code Exa 8.2 To detemine the Electrochemical Equivalent of Silver

```

//To detemine the Electrochemical Equivalent of Silver
//Page 456
clc;
clear;

I=20; //Current Passed
W=26.84; //amount of silver deposited in gms
t=20*60; //Time Period in seconds

TC=I*t; //Total Charge in the given time period

ECEAg=W*1000/TC; //Electrochemical Equivalent of Silver

printf('The Electrochemical Equivalent of Silver is %g mg/C\n',ECEAg)
```

Scilab code Exa 8.3 To Determine the Annual output of copper

```

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

```
TC=NW*T*I; //Total Charge supplied all over the year
    .
ECu=31.8; //Equivalent Weight of Copper in grams
F=96500; //One Farad of Charge

// 1 F of charge gives 31.8 gms of copper

W=(TC/F)*ECu/(1000*1000); //Weight of copper in
    tonnes

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

```

//To Calculate the Energy usedd in producing
    chemical action developed in an electrolytic cell
//Page 456
clc;
clear;

T=15/60; //Time Period of Operation in Hours
I=100; //Current in Amperes
V=15; //Potential Difference
R=0.05; //Resistance of Solution

NetEP=V-(I*R); //Net Electrode Potential

//Energy Equation, E = VIt.
Energy=NetEP*I*T/1000; // Energy in KiloWatts
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`

```

//To determine the weights of Nickel and Silver
    Deposited
//Page 456
clc;
clear;

F=96500; //One Farad of Charge
//Equivalent Weights of the Following metals
EWAg= 108; //silver
EWNi= 58.6/2; //Nickel

I=20; //Current Passed
T=1*60*60; //Time for which the current is passed
TC=T*I; //Total Charge produced

//One Equivalent of Metal requires 1 F of charge

Q=TC/F; // Total Charge in Farad

//Amount of metal deposited
WAg=Q*EWAg; //Silver
WNi=Q*EWNi; //Nickel

printf('The Weight of Nickel And Silver deposited by
     20A for an hour is %g gm and %g gm respectively
    .\n',WNi,WAg)
```