Scilab Manual for
Elements of Electrical Design
by Prof Kaustubh Vyas
Electrical Engineering
Vishwakarma Government Engineering College[1]

Solutions provided by
Prof Kaustubh Vyas
Electrical Engineering
Vishwakarma Government Engineering College

December 5, 2025

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

Contents

List of Experiments

List of Figures

Experiment: 1

Air gap MMF calculation for magnetic circuits using SCILAB

Scilab code Solution 1.1 Experiment1

```
//Experiment-1
// windows 8.1 - 64-Bit
//Scilab - 6.0.0

//Aim : Air gap MMF calculation for magnetic
   circuits using SCILAB
// Data: Calculate mmf required for the air gap of
   dc achine having open slots

clc
clear all

// Following data are to be taken from user

ys=input('Enter Value of slot pitch in cm') // slot
   pitch of DC machine ( in range of 4 - 6 cm)
yo=input('Enter Value of slot opening in cm') //
```

```
    slot opening in DC machine ( in range of 2 - 4 cm
    )
L=input('Enter Value of gross core length in cm') //
    length of armature core ( in range of 40 70 cm)
psi=input('Enter Value of pole arc in cm') // pole
    arc value in DC machine ( in range of 15 3 25 cm)
lg=input('Enter Value of airgap length in cm') //
    length of airgap between armature and stator in
    DC machine ( in range of 0.4 - 0.7 cm)
phi=input('Enter Value of flux per pole in Wb') //
    airgap flux ( in range of 0.04 - 0.08 Wb)
nd=input('Enter no. of ventilating ducts') // radial
    ventilating ducts ( in range of 4 - 10 ducts)
bd=input('Enter opening of each ventilating duct')
    // duct opening ( in range of 1 - 1.5 cm)

 // Actual calculations begin

slot_ratio = yo/lg
if slot_ratio <= 1 then
kcs = 0.15 // carter's coefficient for slots
elseif slot_ratio <= 2 then
    kcs = 0.28 // carter's coefficient for slots
elseif slot_ratio <= 3 then
    kcs = 0.37 // carter's coefficient for slots
elseif slot_ratio <= 3.6 then
    kcs = 0.41 // carter's coefficient for slots
else
    kcs = 0.43 // carter's coefficient for slots
end

duct_ratio = bd/lg
if duct_ratio <= 1 then
kcd = 0.15 // carter's coefficient for ducts
elseif duct_ratio <= 2 then
kcd = 0.28 // carter's coefficient for ducts
elseif duct_ratio <= 3 then
kcd = 0.37 // carter's coefficient for ducts
```

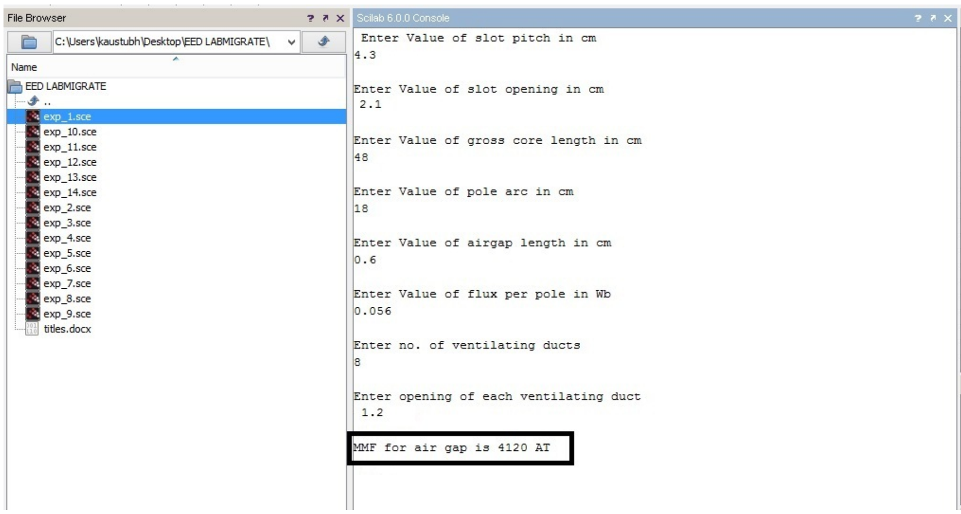

Figure 1.1: Experiment1

```
elseif duct_ratio <= 3.5 then
kcd = 0.41 // carter's coefficient for ducts
else
    kcd = 0.43 // carter's coefficient for ducts
end

kgs = (ys)/(ys-kcs*yo) // gap contraction factor for slots
kgd = (L)/(L-kcd*nd*bd) // gap contraction factor for ducts
Bg  = (phi*1e4)/(psi*L) // air gap flux density
ATg = 8e3*Bg*kgs*kgd*lg

mprintf('MMF for air gap is %d AT',ATg)
```

Experiment: 2

A SCILAB Code to compute Apparent flux density in teeth of Armature in a DC machine

Scilab code Solution 2.2 Experiment2

```
//Experiment-2
// windows 8.1 - 64-Bit
//Scilab - 6.0.0

//Aim : A SCILAB Code to compute Apparent flux
    density in teeth of Armature in a DC machine
// Data: Calculate apparent flux density for teeth
    of dc achine having open slots

clc
clear all

// Following data are to be taken from user

Bm=input('Enter Real flux density at teeth section
    in Wb/m^2'); // Real flux density ( in the range
    of 2 - 2.5 Wb/m^2 )
```

```
L=input('Enter Value of gross core length in cm') // length of armature core ( in range of 30 70 cm)
nd=input('Enter no. of ventilating ducts') // radial ventilating ducts ( in range of 4 - 10 ducts)
bd=input('Enter opening of each ventilating duct') // duct opening ( in range of 1 - 1.5 cm)
bt=input('Enter Value of tooth width in cm') // width of tooth section ( in range of 1 - 3 cm)
yo=input('Enter Value of slot opening in cm') // slot opening in DC machine ( in range of 1 - 4 cm )
mu=input('Enter Value of permeability') // permeability corresponding to real flux density( in range of 30 x 10^-6 - 40 x 10^-6)
Sf=input('stacking factor') // Stacking factor ( in the range of 0.85 - 0.95)

// Actual calculations begin

H  = Bm/mu // magnetization force in AT/m
Li = Sf*(L-nd*bd) // net iron length
ys = bt+yo // sloth pitch
Ks = (ys*L)/(bt*Li)

Bapp = Bm+4*%pi*1e-7*H*(Ks-1)// apparent flux density

mprintf('Apparent flux density for given case is %f, Wb/m^2',Bapp)
```

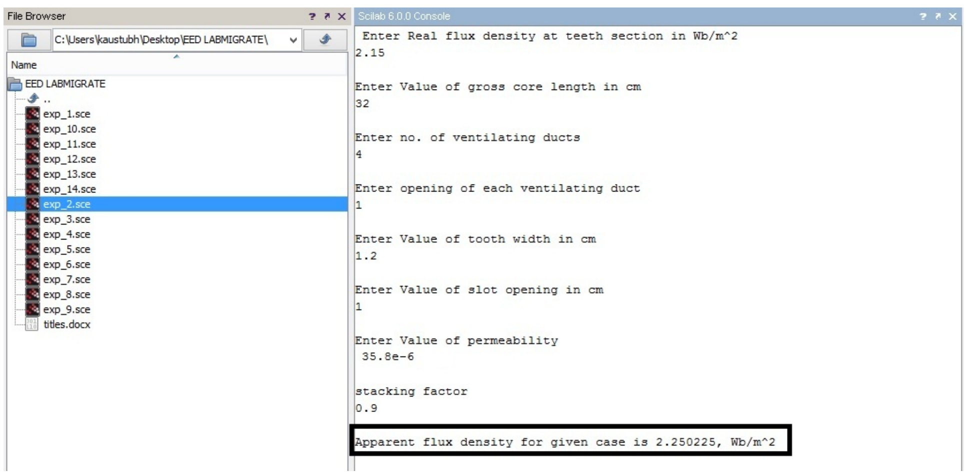

Figure 2.1: Experiment2

Experiment: 3

Design of starter for DC shunt motor through SCILAB code

Scilab code Solution 3.3 Experiment3

```
//Experiment-3
// windows 8.1 - 64-Bit
//Scilab - 6.0.0

//Aim : Design of starter for DC shunt motor through
    SCILAB code
// Data: Determine the resistance of each section of
    a strater to be used for DC shunt motor

clc
clear all

// Following data is to be taken form the user

N=input('Enter value of No. of studs in the starter'
  ) // no. of studs to determine the no. of element
    sections of starter ( in the range of 4 - 10)
V=input('Enter voltage rating of the motor')//
  voltage reating of the motor ( in the range of
```

```
220 - 400 V )
P=input('Power Rating of motor in kW') // Rating of motor in kW ( in the range of 15 - 50)
n=input('Enter Efficiency in percentage') // efficiency of motor ( in the range of 85 - 95 %)
Ra=input('Enter Value of Armature Resistance in Ohms') // Armature resistance ( in the range of 0.1 - 0.4 ohms)
Tm=input('Enter the ratio of maximum torque to full load torque') // ratio of torque in per unit ( in the range of 1.2 - 1.8)

// Actual Calculations begin

Ia = (P*1e5)/(V*n); // Armature current
Im = Tm*Ia // Maximum value of full load current
el = N-1 // no. of resistance elements
R1 = V/Im // Resistance of first section
K  = (R1/Ra)^(1/el);
Il = Im/K // lower reange of current
R(1) = R1
for i = 1:el
    R(i+1)=R(i)/K
    r(i)=R(i)-R(i+1)
    mprintf('Resistance value of section - %d is %f Ohms',i,r(i))
    mprintf('\n')
end

mprintf('Total Starter resistance is %f Ohms',sum(r))
    mprintf('\n')
mprintf('Resistance of motor is %f Ohms',Ra)
    mprintf('\n')
mprintf('Total Resistance at starting time is %f Ohms',Ra+sum(r))
    mprintf('\n')
mprintf('Upper range of Staring current is %f Amp',
```

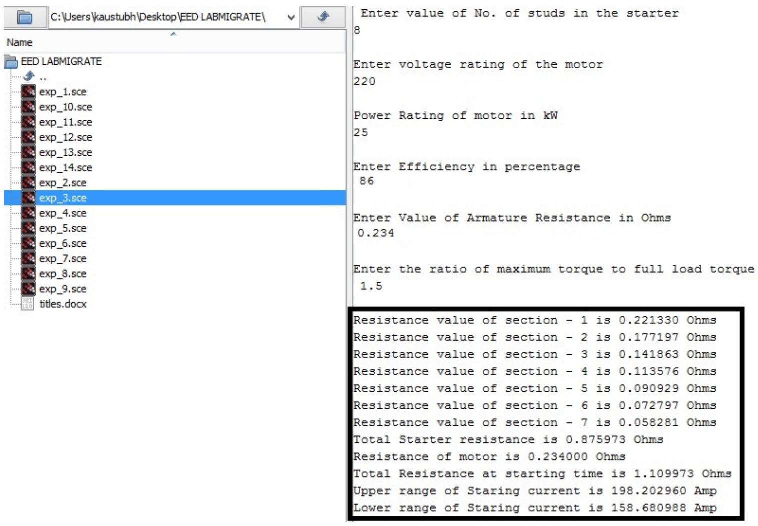

Figure 3.1: Experiment3

```
    Im)
  mprintf('\n')
mprintf('Lower range of Staring current is %f Amp',
    Il)
```

Experiment: 4

Design of starter for DC series motor through SCILAB code

Scilab code Solution 4.4 Experiment4

```
//Experiment-4
// windows 8.1 - 64-Bit
//Scilab - 6.0.0

//Aim : Design of starter for DC series motor
   through SCILAB code
// Data: Determine the resistance of each section of
    a strater to be used for DC series motor

clc
clear all

// Following data is to be taken form the user

N=input('Enter value of No. of studs in the starter'
   ) // no. of studs to determine the no. of element
    sections of starter ( in the range of 4 - 10)
V=input('Enter voltage rating of the motor')//
   voltage reating of the motor ( in the range of
```

```
    220 - 400 V )
Im=input('Maximum Starting current') // Maximum required value of starting current ( in the range of 150 -250 Amp)
Il=input('Minimum Starting current') // Minimum required value of starting current ( in the range of 100 -200 Amp)
Ra=input('Enter Value of Armature Resistance in Ohms') // Armature resistance ( in the range of 0.1 - 0.4 ohms)
Phi=input('Enter the ratio of maximum flux to minimum flux') // ratio of fluxes corresponing to minimum and maximum current (in per unit) ( in the range of 1.05 - 1.25)

// Actual Calculations begin

K = Im/Il // ratio of currents
b = Phi / K
el=N-1 // No. of element sections
R(1) = V/Im

for i = 1:el
    R(i+1)=b*R(i)+R(1)*(1-Phi)
    r(i)=R(i)-R(i+1)
    mprintf('Resistance value of section - %d is %f Ohms',i,r(i))
    mprintf('\n')
end
```

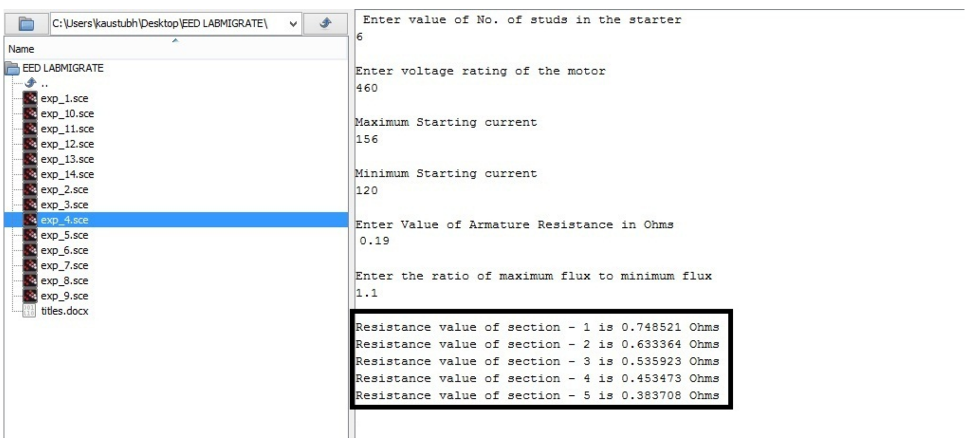

Figure 4.1: Experiment4

Experiment: 5

A SCILAB code to design rotor resistance starter of a slip-ring induction motor

Scilab code Solution 5.5 Experiment5

```
//Experiment-5
// windows 8.1 - 64-Bit
//Scilab - 6.0.0

//Aim : A SCILAB code to design rotor resistance
    starter of a slip-ring induction motor
// Data: Calculate the steps in 5 section rotor
    starter

clc
clear all

// Following data is to be taken form the user

P=input('Enter power rating of motor in kW') //
    power rating of motor in kW ( in the range of 1 -
     15 kW)
```

```
N=input('Enter no. of studes for starter to be designed') // No. of studs ( in range of 5 - 10)
s=input('Enter full load slip in percentage') // full load slip ( in the range of 1 - 5 %)
R=input('Enter value of rotor resistance per phase in Ohms') // rotor resistnace ( in the range of 0.01 - 0.1 Ohms)

// Actual Calculations begin

el = N-1    // no. of resistance elements
R(1) = R*100/s  // resistance at stud 1
K = (s/100)^(1/(N-1))

for i=1:el
R(i+1) = K*R(i)
r(i) = R(i)-R(i+1)
mprintf('Resistance of section - %d is %f Ohms',i,r(i))
mprintf('\n')
end
```

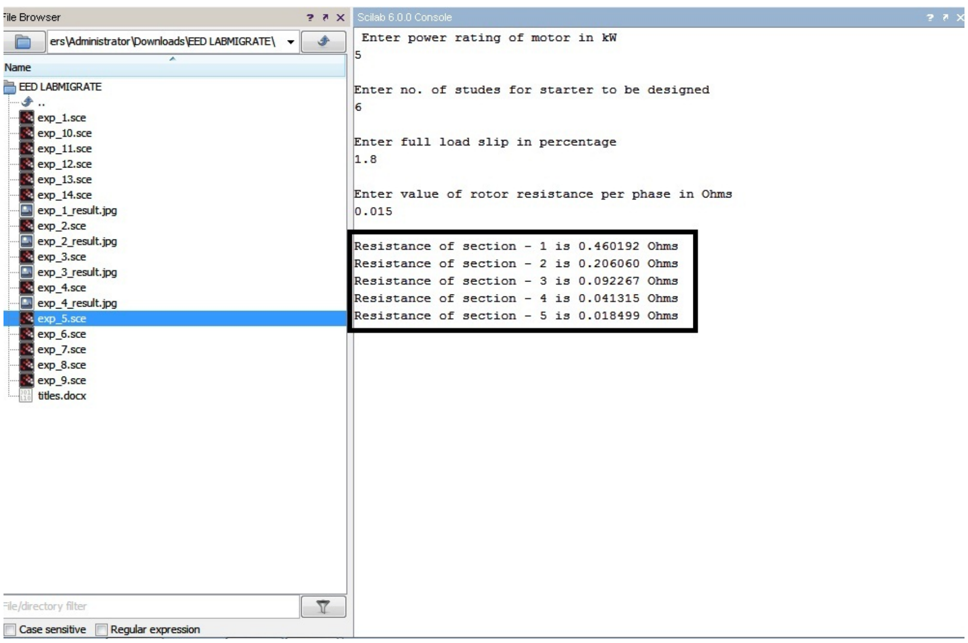

Figure 5.1: Experiment5

Experiment: 6

Design of a small single phase transformer using SCILAB coding

Scilab code Solution 6.6 Exp6

```
//Experiment-5
// windows 8.1 - 64-Bit
//Scilab - 6.0.0


//Aim : Design of a small single phase transformer
    using SCILAB coding
//Data: Design a small single phase transformer for
    given ratings

clc
clear all

// Following data is to be taken form the user
Vo=input('Enter value of output voltage of
    transformer') // output voltage of the
    transformer (in the range of 5 - 12 volts)
```

```
Io=input('Enter value of output current of
   transformer') // output current of the
   transformer (in the range of 2 - 5 Amp.)
Vi=input('Enter value of input voltage of
   transformer') // input voltage of the transformer
    (in the range of 110 - 230 volts)
f=input('Enter value of operating frequency in Hz')
   // operating frequency of the transformer (in the
    range of 50 - 60 Hz)

// Assuming following

n = 0.9   // assuming 90% eficiency
Et = 9 // emf per turn
Bm = 1 // maximum flux density
Ks = 0.9 // stacking factor
del = 2.3 // current density in conductor

// Actual Calculations begin

P = Vo*Io // output rating of transformer
phi_m = (1/(4.44*f*Et)) // maximum flux in the core
Ac = phi_m/Bm // net area of core
Ag = Ac/Ks // gross core area
A = sqrt(Ag) // width of central limb assuming
   square cross section
Np = Vi*Et // No.of turns in primary
Ns = ceil(1.05*Vo*Et) // No.of turns in secondary
Ip = P/(n*Vi) // Current in primary winding
Ap = Ip / del // corss sectional area of bare
   primary conductor
dp = sqrt(4*Ap/%pi) // diameter of bare primary
   conductor
dpi = dp+0.3 // diameter of insulated primary
   conductor
Api = (%pi*dpi^2)/4 // corss sectional area of
   insulated primary conductor
As = Io / del // corss sectional area of bare
```

```
secondary conductor
ds = 0.2+sqrt(4*As/%pi) // diameter of bare secondary conductor
dsi = ds+0.1 // diameter of insulated secondary conductor
Asi = (%pi*dsi^2)/4 // corss sectional area of insulated secondary conductor
sfp = 0.8*(dp/dpi)^2 // space factor of primary winding
Awp = ceil(Np*Api/sfp) // window area for primary winding
sfs = 0.8*(ds/dsi)^2 // space factor of secondary winding
Aws = ceil(Ns*Asi/sfs) // window area for secondary winding
Aw = 1.2*(Aws+Awp) // gross window area required

mprintf('No. of urns required in Primary is %d',Np)
mprintf('\n')
mprintf('No. of urns required in Secondary is %d',Ns)
mprintf('\n')
mprintf('Power Rating of the transformer is %d VA',P)
mprintf('\n')
mprintf('Diameter of insulated primary conductor is %f mm',dpi)
mprintf('\n')
mprintf('Diameter of insulated secondary conductor is %f mm',dsi)
mprintf('\n')
mprintf('Gross window area required is %d mm^2',Aw)
```

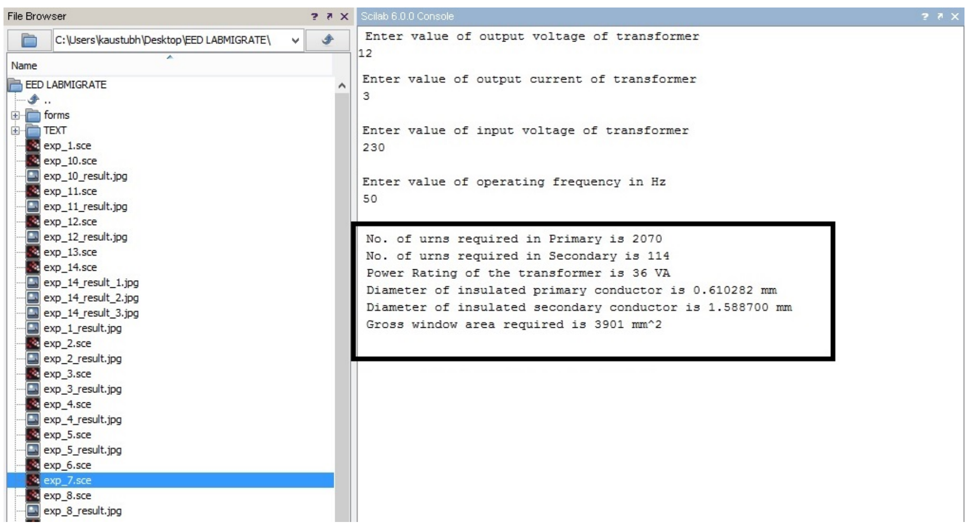

Figure 6.1: Exp6

Experiment: 7

Design of an iron cored choke coil using SCILAB coding

Scilab code Solution 7.7 Experiment7

```
//Experiment-7
//windows 8.1 - 64-Bit
//Scilab - 6.0.0

//Aim : Design of an iron cored choke coil using
    SCILAB coding
////Data: Design a single phase variable choke coil

clc
clear

//Following data are to be taken from user

V=input('Enter value of supply voltage') // supply
    voltage (in the range of 230 - 440 V)
f=input('Enter frequency of supply') // supply
    frequency = 50 Hz in India and 60 Hz in US
I=input('Enter value of current to be carried in Amp
    ') // currentn carrying capacity in Amp ( in the
```

```
range of 5 - 25 Amp)
lg=input('Enter maximum airgap length in cms') // airgap length varies between 0 and 10 cms being variable choke coil

//Actual Calculations begin

uo=4*%pi*1e-7 // permeability of free space
Sf=0.9 // stacking factor
K=(uo*V*I)/(2*%pi*f*2*lg/100)
i=1
for Bg=0.2:0.1:0.8
    A_root(i)=sqrt(K)/Bg
    i=i+1
end
Bgm=0.45
A_root=0.0225
A_i=A_root^2 // net iron area
Agi=A_i/Sf; // gross iron area
A  = sqrt(Agi) // width of limb
ATg = Bgm*2*lg/100/uo // airgap mmf
ATt = 1.1*ATg // total mmf
N  = ceil(ATt / I) // no. of turns
del=2.4 // current density
a  = I/del // conductor area in mm^2
d  = sqrt(4*a/%pi)+.05 // diameter of conductor in mm
d1 = 2.488 // diameter of insulated conductor
a1 = (%pi/4)*d1^2 // cross sectional area of insulated conductor
sf = 0.8*(d/d1)^2 // space factor
Aw = N*a1/sf // area of window
AW = 1.2*Aw // gross window area in mm^2
Ww = sqrt(AW/2) // width of window in mm
Hw = 2*Ww // height of window assuming H/W ratio is 2
hf = Hw-20 // height of winding
Nh = ceil(hf/d1) // no. of conductors in height
```

```
Nd = N/(2*Nh) // no. of conductors in depth
dc = ceil(Nd*d1) // depth of coil
dc1 = dc+5 // actual depth of coil
hf1 = ceil(hf+10) // actual height of winding
dw = ceil(Ww - 2*dc1) // distance betwen two coils
D = ceil(Ww + A*1000) // distance between limbs
WC = ceil(D+A*1000) // width of core in mm
HC = ceil(Hw+2*A*1000+lg*10) // height of core in mm
Z= V/I
mprintf('Diameter of insulated conductor is %f mm',d1)
mprintf('\n')
mprintf('Area of insulated conductor is %f mm^2', a1)
mprintf('\n')
mprintf('Height of coil is %d mm',hf1)
mprintf('\n')
mprintf('Depth of coil is %d mm',dc1)
mprintf('\n')
mprintf('Height of core is %d mm',HC)
mprintf('\n')
mprintf('Width of core is %d mm',WC)
mprintf('\n')
mprintf('Impedance of the coil is %d Ohms',Z)
```

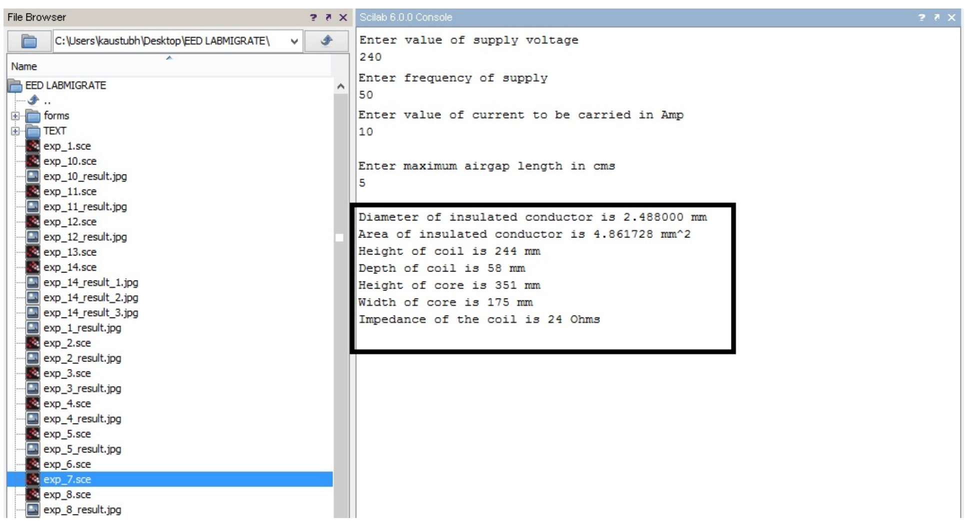

Figure 7.1: Experiment7

Experiment: 8

A SCILAB code to workout design of simplex lap winding in DC machines

Scilab code Solution 8.8 Exp8

```
//Experiment-8
// windows 8.1 - 64-Bit
//Scilab - 6.0.0

//Aim : A SCILAB code to workout design of simplex
    lap winding in DC machines
//Data: Workout details of simplex lap winding for
    DC generator

clc;
clear all;

// Following data is to be aken from user

p=input('Enter No. of poles') // no. poles (in the
    range of 2 - 12) always an even number
s=input('Enter No. of slots in armature') // no. of
```

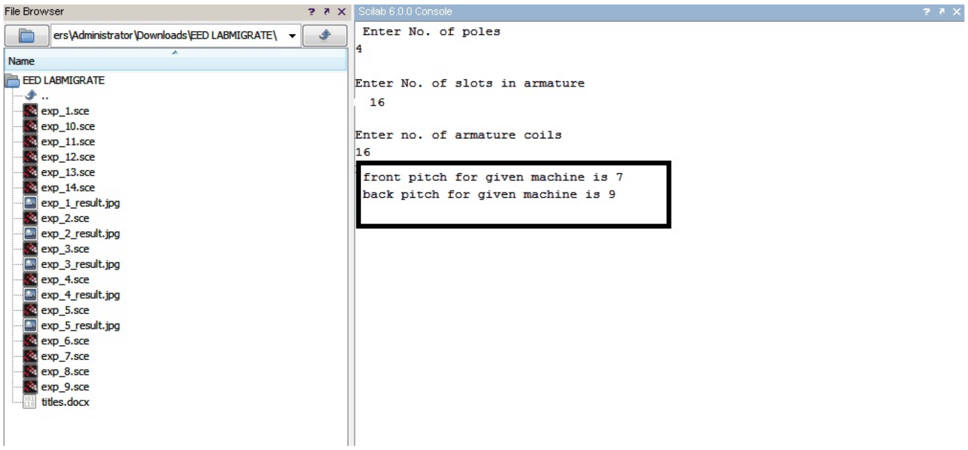

Figure 8.1: Exp8

```
   armature slots (in the range of 16 -48)
c=input('Enter no. of armature coils') // armature
   coils in multiple of armature slots (in the range
    of 16 - 48)

// Actual calculations begin

cs=c*2 // no. of coil sides
csps=cs/s // no. of coils sides per slot
cspp=cs/p // no. of coils sides per pole

yb = cspp+1 // back pitch
yf = cspp-1 //front pitch

mprintf('front pitch for given machine is %d',yf )
mprintf('\n')
mprintf('back pitch for given machine is %d',yb )
```

Experiment: 9

A SCILAB code to workout design of simplex Wave winding in DC machines

Scilab code Solution 9.9 Exp9

```
//Experiment-8
// windows 8.1 - 64-Bit
//Scilab - 6.0.0

//Aim : A SCILAB code to workout design of simplex
    Wave winding in DC machines
//Data: Workout details of simplex wave winding for
    DC generator

clc;
clear all;

// Following data is to be taken from user

p=input('Enter No. of poles') // no. poles (in the
    range of 2 - 12) always an even number
s=input('Enter No. of slots in armature') // no. of
```

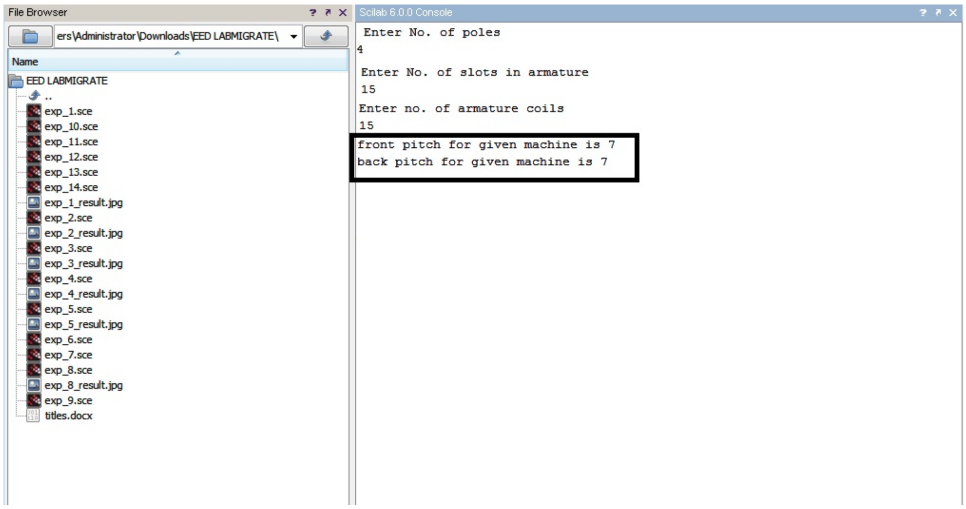

Figure 9.1: Exp9

```
    armature slots (in the range of 10 - 48)
c=input('Enter no. of armature coils') // armature
    coils in multiple of armature slots (in the range
     of 10 - 48)

// Actual calculations begin

cs=c*2 // no. of coil sides
csps=cs/s // no. of coils sides per slot
cspp=cs/p // no. of coils sides per pole

yb = floor(cs/p) // back pitch
y = (cs-2)/(p/2) //winding pitch
yf=y-yb

mprintf('front pitch for given machine is %d',yf )
mprintf('\n')
mprintf('back pitch for given machine is %d',yb )
```

Experiment: 10

Core Loss Calculations in magnetic materials using SCILAB Programming

Scilab code Solution 10.10 Experiment10

```
//Experiment-10
// windows 8.1 - 64-Bit
//Scilab - 6.0.0

//Aim : Core Loss Calculations in magnetic materials
    using SCILAB Programming
//Data: CAlculate core loss per kg in a specimen of
    ally sheet using user defined data

clear;
clc;


// Following data is to be taken from user

Bm=input('Enter value of Maximum flux density in Wb/
    m^2') // maxium flux density (in the range of 0.5
```

```
 - 2.5 Wb/m^2)
f=input('Enter value of frequency in Hz') // frequency of flux reversal (genrally 50 Hz)
t=input('Enter thickness of laminated plates in mm') // tickness of laminations (in range of 0.2 - 0.6 mm )
r=input('Enter Resistivity of material in Ohm*m')// resistivity in Ohm*m (in the range of 0.2 - 0.6 micro Ohm*m)
sg=input('Enter specific gravity of material in kg/m^3') // specific gravity (in the range of 7 - 10 kg/m^3)
hl=input('Enter value of hysterisis loss in in J*Hz/m^3') // specific hysterisis loss (in range of 400 - 800 J*Hz/m^3)

// Actual calculations begin

Pe=((%pi^2)*(f^2)*(Bm^2)*((t/1000)^2))/((sg*1000)*(6*r)) // eddy current loss per kg
Ph=hl*f/(sg*1e3)
Pi=Pe+Ph
mprintf('Eddy Current Loss per kg for given material is %f W',Pe)
mprintf('\n')
mprintf('Hysterisis Loss per kg for given material is %f W',Ph)
mprintf('\n')
mprintf('Total Iron Loss per kg for given material is %f W',Pi)
```

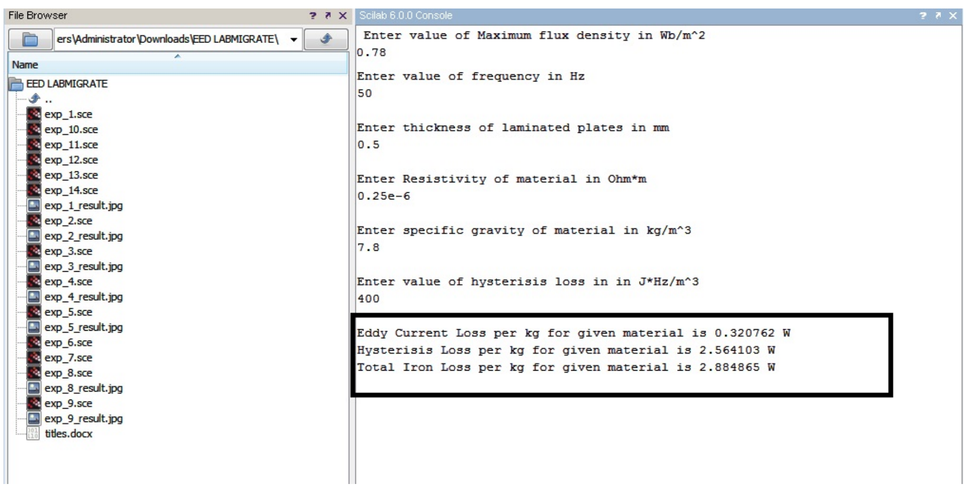

Figure 10.1: Experiment10

Experiment: 11

A SCILAB program for Computation of slot leakage reactance in induction motor

Scilab code Solution 11.11 Experiment11

```
//Experiment-11
// windows 8.1 - 64-Bit
//Scilab - 6.0.0

//Aim : A SCILAB program for Computation of slot
    leakage reactance in induction motor
// Data: Find out slot leakage reactance for
    induction motor from given slot dimensions

clc;
clear all;

// Data to be taken from user

T=input('Enter no. of turns per phase') // turns per
     phase for induction motor (in the range of 200 -
     300)
```

```
s=input('Enter no. of slots per phase') // slots per
    phase for induction motor (in the range of 12 -
   24)
bs=input('Enter slot width in mm') // slot opening
   in mm (in the range of 10 -30 mm)
L=input('Enter core / slot length in mm') // slot
   length in mm (in the range of 150 -650 mm)
wo=input('Enter lip opening in mm') // width of lip
   in mm (in the range of 2 - 6 mm)
h1=input('Enter heigth of conductor in slot in mm')
   // heigth of conductor portion in mm (in the
   range of 20 - 80 mm)
h2=input('Enter Value of clearance between
   conductors and wedge') // clearnace in mm (in the
    range of 1 - 4 mm)
h3=input('Enter heigth of wedge in mm') // wedge
   height in mm (in the range of 2 - 6 mm)
h4=input('Enter height of lip in mm') // lip height
   in mm (in the range of 1 - 4 mm)

// Actual calculations begin

f=50; // frequency of supply
uo=4*%pi*1e-7; // pearmeabilty of free space
Ls=(h1/(3*bs))+(h2/(bs))+(2*h3/(bs+wo))+(h4/wo)
xs=8*%pi*f*uo*T^2*(L/1000)*Ls/s

mprintf('Leakage reactance for given machine is %f
   Ohms',xs)
```

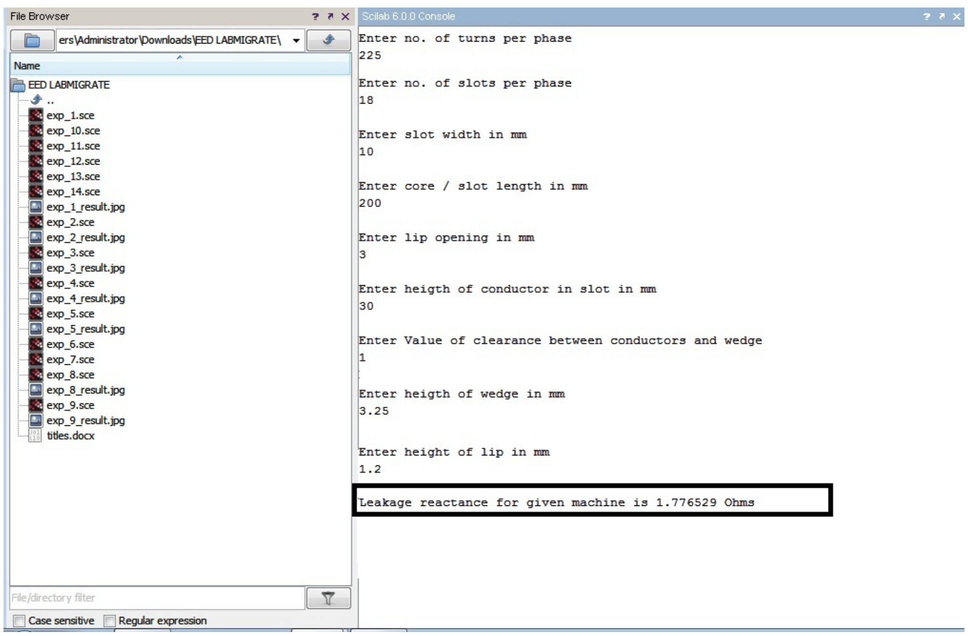

Figure 11.1: Experiment11

Experiment: 12

A SCILAB program for Design of Electromagnet

Scilab code Solution 12.12 Experiment12

```
//Experiment-12
//windows 8.1 - 64-Bit
//Scilab - 6.0.0

//Aim :A SCILAB program for Design of Electromagnet
//Data:Design a flat faced armature type of circular electromagnet

clc
clear

//Following data is to be taken form the user

F=input('Enter Amount of force in kg') // force to be exerted by electromagnet in kg(in the range of 100 - 250 kg)
s=input('Enter value of stroke in mm') // stroke in mm (in the range of 0.5 - 1.5 mm)
v=input('Enter supply volage') // in the range of 5
```

```
- 15 volts
ta=input('Enter ambient temperature in degree celcius') // general value is 20 oC
t=input('Enter permissible temperature rise above ambient temperature') // 50 - 80 oC

//Actual Calculations begin

Bm = 1.1 // maximum flux density in Wb/m^2
uo = 4*%pi*1e-7
ro_0=0.01734 // resistib=vity at 0 degree
alp_0=0.00393 // temperature coefficient of resistance
h_by_d=3 // heigth to depth ratio of coil
C=0.085 // constant
Sf=0.5 // assumed space factor

A  = F*uo/(0.102*Bm^2) // are of central limb
r1 = sqrt(A/%pi) // radius of central limb
//disp(r1)
AT=(1600000*Bm*s*1e-3)/0.85 // total mmf required
ro_70=ro_0*(1+(alp_0*t))
hc=((3*ro_70*C*AT^2)/(2*Sf*t)*1e-6)^(1/3) // heigth of coil
//disp(hc)
dc=hc/3 //depth of coil
//disp(dc)
r2=dc+r1
//disp(r2)
t1=r1/2
//disp(t1)
t2=(r1^2)/(2*r2)
//disp(t2)
r3=sqrt(r1^2+r2^2)
//disp(r3)
a=(AT*ro_70*%pi*(r1+r2))*1e-3/v // cross sectional area of conductor
//disp(a)
```

```
d=sqrt(4*a*1000/%pi) // diameter of conductor
//disp(d)
dc1=dc*1000-2 // depth of coil considering clearance
//disp(dc1)
nd=ceil(dc1/d) // no. of layers in depth
//disp(nd)
hc1=hc*1000-3 // height of coil considering
    clearance
nh=ceil(hc1/d) // no o flayers in height
//disp(nh)
T=nd*nh // no. of turns in coil
//disp(T)
ab=(%pi/4*d^2) // diameter of bare conductor
R=T*ro_70*(%pi*(r1+r2))/ab // resistance of coil
//disp(R)
I=v/R; // current in coil
//disp(I)
mmf=I*T // actual mmf developed by coil
//disp(mmf)
sf=T*ab/(hc*dc) // actual space factor
//disp(sf)
theta = (ro_70*C*mmf^2*1e-6)*1e6/(2*sf*dc*hc^2)
//disp(theta  )

mprintf('heigth of coil is %f mm',hc*1000)
mprintf('\n')
mprintf('depth of coil is %f mm',dc*1000)
mprintf('\n')
mprintf('total no. of turns in the coil are %d',T)
mprintf('\n')
mprintf('Current flowing through the coil is  %f Amp
    ',I)
mprintf('\n')
mprintf('MMF developed by the coil is %d AT',mmf)
mprintf('\n')
mprintf('Temperature rise of the coil is %f oC',
    theta)
mprintf('\n')
```

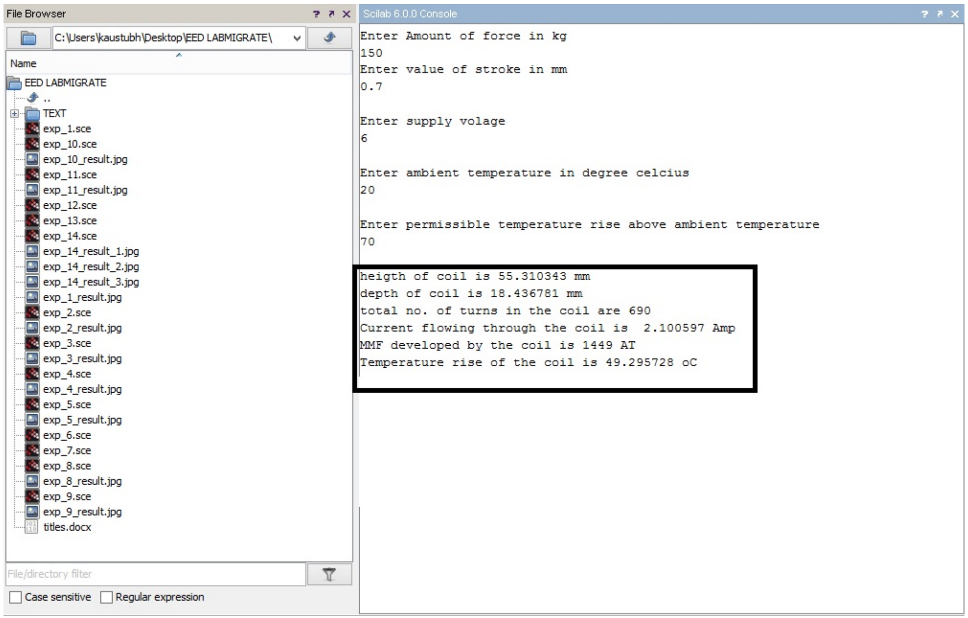

Figure 12.1: Experiment12

Experiment: 13

Computation of winding factor and distribution factor in armature winding using SCILAB programming

Scilab code Solution 13.13 Exp13

```
//Experiment-13
// windows 8.1 - 64-Bit
//Scilab - 6.0.0

//Aim :Computation of winding factor and
    distribution factor in armature winding using
    SCILAB programming
//Data:Calculate winding factor and distribution
    factor for armature winding of synchronous
    machine

clc;
clear all;

// Data to be taken from the user
```

```
p=input('Enter no. of poles') // no. of poles in synchronous machine (in the range of 2 - 12)
s=input('Enter no. of slots') // no. of slots in synchronous machine ( in the range of 16 - 96)
pitch=input('Enter No. of short pitch slots') // no. of slots by which short pitching is required ( in the range of 1 to 4)

// Actual calculations start

sp = s/p // slots per pole
spp = sp/3 // slots per pole per phase
dist = %pi/spp //
spread = %pi/sp // phase spread angle
alpha = spread*pitch // angle of short pitch
bet = %pi/p // distribution angle

kp1=cos(alpha/2) // pitch factor for fundamental
kp5=cos(5*alpha/2) // pitch factor for 5th harmonic

kd1=(sin(spp*bet/2))/(spp*sin(bet/2)) // distribution factor for fundamental
kd5=(sin(spp*5*bet/2))/(spp*sin(5*bet/2)) // distribution factor for 5th harmonic

mprintf('Pitch factor for fundamental is %f',kp1)
mprintf('\n')
mprintf('Pitch factor for 5th harmonic is %f',kp5)
mprintf('\n')
mprintf('Distribution factor for fundamental is %f',kd1)
mprintf('\n')
mprintf('Distribution factor for 5th harmonic is %f',kd5)
mprintf('\n')
```

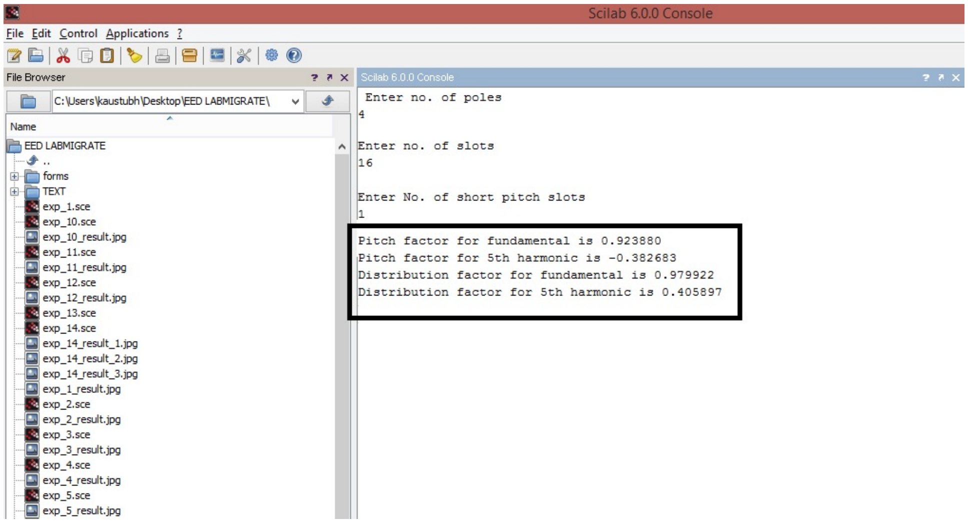

Figure 13.1: Exp13

Experiment: 14

Analyzing variation of slot leakage reactance in induction motor using SCILAB code

Scilab code Solution 14.14 Experiment14

```
//Experiment-14
// windows 8.1 - 64-Bit
//Scilab - 6.0.0
//Aim :Analyzing variation of slot leakage reactance
    in induction motor using SCILAB code
// Data: Plot graphs showing effect of no. of slots,
    length of armature core and no. of turns on slot
    leakage reactance

clc;
clear;

// Assumed data



bs=10 // slot opening in mm (in the range of 10 -30
```

```
mm)
wo=3 // width of lip in mm (in the range of 2 - 6 mm )
h1=30 // heigth of conductor portion in mm (in the range of 20 - 80 mm)
h2=1 // clearnace in mm (in the range of 1 - 4 mm)
h3=3.25 // wedge height in mm (in the range of 2 - 6 mm)
h4=1.2 // lip height in mm (in the range of 1 - 4 mm )

f=50; // frequency of supply
uo=4*%pi*1e-7; // pearmeabilty of free space
Ls=(h1/(3*bs))+(h2/(bs))+(2*h3/(bs+wo))+(h4/wo)

// Evaluating variation in leakage reactance with change in no. of slots
L=200
T=225
for s = 1:25
    xs(s)=8*%pi*f*uo*T^2*(L/1000)*Ls/s
end

x=1:25
figure(1)
plot(x,xs)
xlabel('No. of slots')
ylabel('Leakage reactance in Ohms')
title('Variation of leakage reactance with change in no. of slots')

clear xs s T L
// Evaluating variation in leakage reactance with change in length of core
L = 100:10:2000
s = 20
T = 225
for i = 1:length(L)
```

```
    xs(i)=8*%pi*f*uo*T^2*(L(i)/1000)*Ls/s
end
figure(2)
plot(L,xs)
xlabel('length of core in mm')
ylabel('Leakage reactance in Ohms')
title('Variation of leakage reactance with change in length of core')

clear xs s T L
// Evaluating variation in leakage reactance with change in no. of turns per phase
L = 200
s=20
T = 100:400
for i = 1:length(T)
    xs(i)=8*%pi*f*uo*T(i)^2*(L/1000)*Ls/s
end
figure(3)
plot(T,xs)
xlabel('No. of turns per phase')
ylabel('Leakage reactance in Ohms')
title('Variation of leakage reactance with change in no. of turns per phase')
```

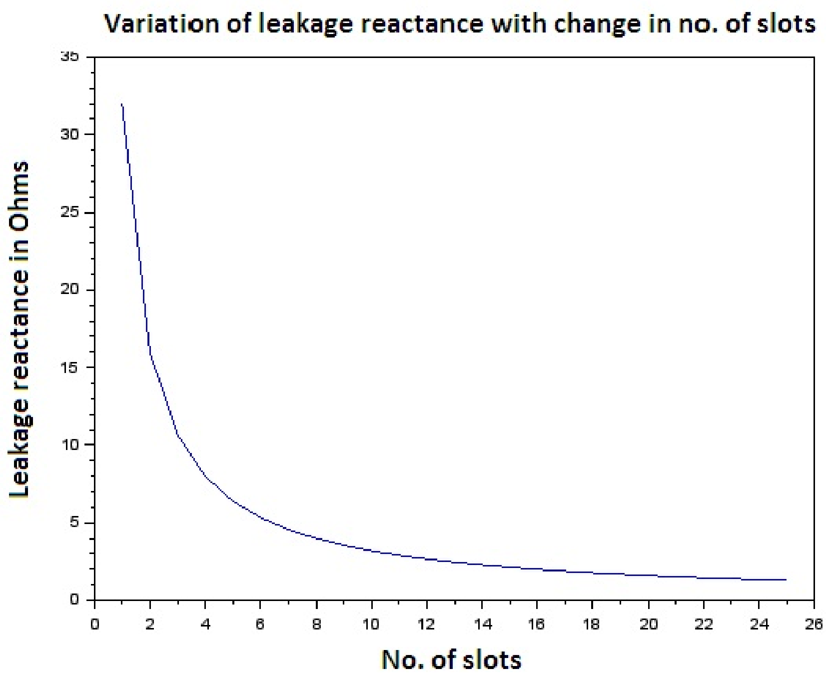

Figure 14.1: Experiment14

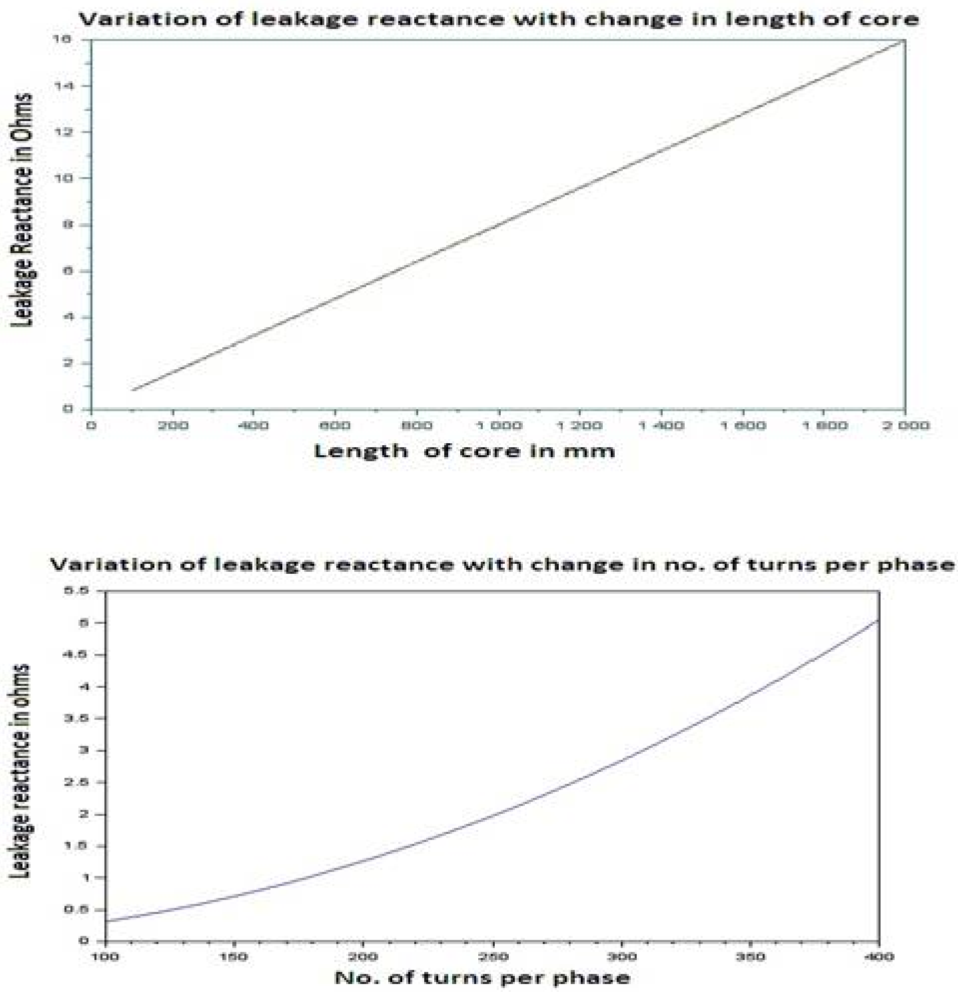

Figure 14.2: Experiment14