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

February 8, 2026

Contents

List of Experiments

List of Figures

Experiment: 1

Air gap MMF calculation for magnetic circuits using SCILAB

Scilab code Solution 1.1 Experiment1

```
1  //Experiment-1
2  // windows 8.1 - 64-Bit
3  //Scilab - 6.0.0
4
5  //Aim : Air gap MMF calculation for magnetic
       circuits using SCILAB
6  // Data: Calculate mmf required for the air gap of
       dc achine having open slots
7
8  clc
9  clear all
10
11 // Following data are to be taken from user
12
13 ys=input('Enter Value of slot pitch in cm') // slot
       pitch of DC machine ( in range of 4 - 6 cm)
14 yo=input('Enter Value of slot opening in cm') //
```

```scilab
                slot  opening  in  DC  machine  (  in  range  of  2 − 4  cm
                   )
15  L=input('Enter  Value  of  gross  core  length  in  cm') //
                   length  of  armature  core  (  in  range  of  40  70  cm)
16  psi=input('Enter  Value  of  pole  arc  in  cm') // pole
                   arc  value  in  DC  machine  (  in  range  of  15  3  25  cm)
17  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)
18  phi=input('Enter  Value  of  flux  per  pole  in  Wb') //
                   airgap  flux  (  in  range  of  0.04 − 0.08  Wb)
19  nd=input('Enter  no.  of  ventilating  ducts') // radial
                   ventilating  ducts  (  in  range  of  4 − 10  ducts)
20  bd=input('Enter  opening  of  each  ventilating  duct')
                   // duct  opening  (  in  range  of  1 − 1.5  cm)
21
22   // Actual  calculations  begin
23
24  slot_ratio = yo/lg
25  if slot_ratio <= 1 then
26  kcs = 0.15 // carter's  coefficient  for  slots
27  elseif slot_ratio <= 2 then
28      kcs = 0.28 // carter's  coefficient  for  slots
29  elseif slot_ratio <= 3 then
30      kcs = 0.37 // carter's  coefficient  for  slots
31  elseif slot_ratio <= 3.6 then
32      kcs = 0.41 // carter's  coefficient  for  slots
33  else
34      kcs = 0.43 // carter's  coefficient  for  slots
35  end
36
37  duct_ratio = bd/lg
38  if duct_ratio <= 1 then
39  kcd = 0.15 // carter's  coefficient  for  ducts
40  elseif duct_ratio <= 2 then
41  kcd = 0.28 // carter's  coefficient  for  ducts
42  elseif duct_ratio <= 3 then
43  kcd = 0.37 // carter's  coefficient  for  ducts
```
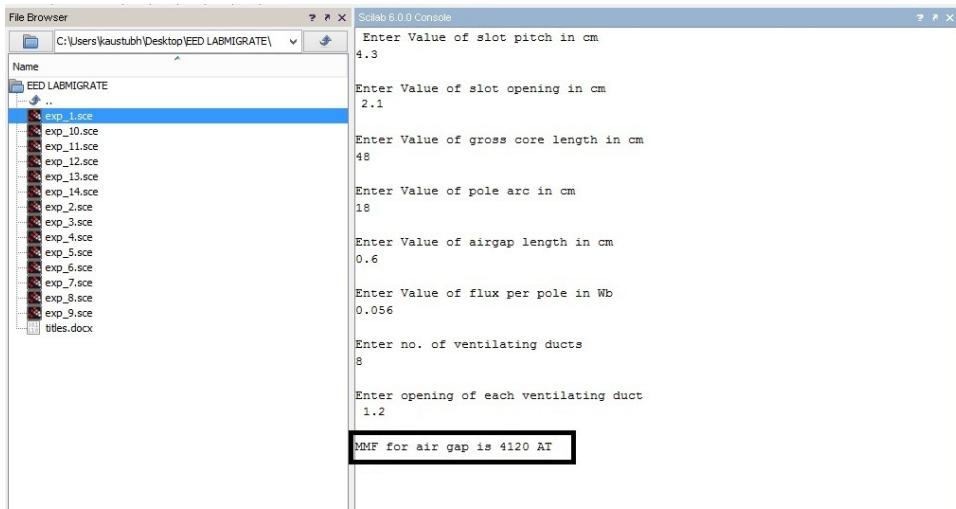
Figure 1.1: Experiment1

```
44  elseif duct_ratio <= 3.5 then
45  kcd = 0.41 // carter's coefficient for ducts
46  else
47      kcd = 0.43 // carter's coefficient for ducts
48  end
49
50  kgs = (ys)/(ys-kcs*yo) // gap contraction factor for
         slots
51  kgd = (L)/(L-kcd*nd*bd) // gap contraction factor
        for ducts
52  Bg  = (phi*1e4)/(psi*L) // air gap flux density
53  ATg = 8e3*Bg*kgs*kgd*lg
54
55  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

```
1  // Experiment-2
2  //  windows  8.1 - 64-Bit
3  // Scilab - 6.0.0
4
5  //Aim : A SCILAB Code to compute Apparent flux
       density in teeth of Armature in a DC machine
6  // Data: Calculate apparent flux density for teeth
      of dc achine having open slots
7
8  clc
9  clear all
10
11 // Following data are to be taken from user
12
13 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 )
```

```scilab
14  L=input('Enter Value of gross core length in cm') //
        length of armature core ( in range of 30 70 cm)
15  nd=input('Enter no. of ventilating ducts') // radial
        ventilating ducts ( in range of 4 - 10 ducts)
16  bd=input('Enter opening of each ventilating duct')
        // duct opening ( in range of 1 - 1.5 cm)
17  bt=input('Enter Value of tooth width in cm') //
        width of tooth section ( in range of 1 - 3 cm)
18  yo=input('Enter Value of slot opening in cm') //
        slot opening in DC machine ( in range of 1 - 4 cm
        )
19  mu=input('Enter Value of permeability') //
        permeability corresponding to real flux density(
        in range of 30 x 10^-6 - 40 x 10^-6)
20  Sf=input('stacking factor') // Stacking factor ( in
        the range of 0.85 - 0.95)
21
22  // Actual calculations begin
23
24  H  = Bm/mu // magnetization force in AT/m
25  Li = Sf*(L-nd*bd) // net iron length
26  ys = bt+yo // sloth pitch
27  Ks = (ys*L)/(bt*Li)
28
29  Bapp = Bm+4*%pi*1e-7*H*(Ks-1)// apparent flux
        density
30
31  mprintf('Apparent flux density for given case is %f,
        Wb/m^2',Bapp)
```
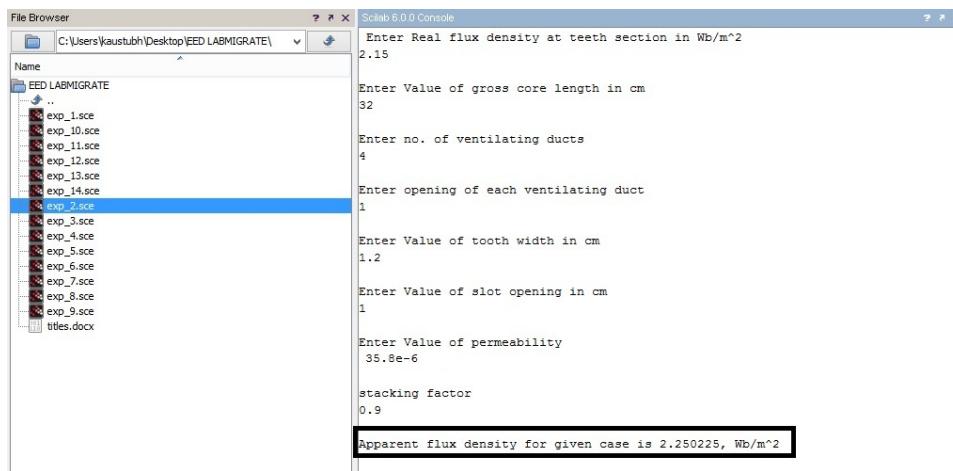
Figure 2.1: Experiment2

Experiment: 3

Design of starter for DC shunt motor through SCILAB code

Scilab code Solution 3.3 Experiment3

```
1  //Experiment-3
2  // windows 8.1 - 64-Bit
3  //Scilab - 6.0.0
4
5  //Aim : Design of starter for DC shunt motor through
        SCILAB code
6  // Data: Determine the resistance of each section of
        a strater to be used for DC shunt motor
7
8  clc
9  clear all
10
11 // Following data is to be taken form the user
12
13 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)
14 V=input('Enter voltage rating of the motor')//
        voltage reating of the motor ( in the range of
```

```
         220 - 400 V )
15  P= input ('Power Rating of motor in kW') // Rating of
        motor in kW ( in the range of 15 - 50)
16  n= input ('Enter Efficiency in percentage') //
        efficiency of motor ( in the range of 85 - 95 %)
17  Ra= input ('Enter Value of Armature Resistance in Ohms
        ') // Armature resistance ( in the range of 0.1 -
        0.4 ohms)
18  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)
19
20  // Actual Calculations begin
21
22  Ia = (P*1e5)/(V*n); // Armature current
23  Im = Tm*Ia // Maximum value of full load current
24  el = N-1 // no. of resistance elements
25  R1 = V/Im // Resistance of first section
26  K  = (R1/Ra)^(1/el);
27  Il = Im/K // lower reange of current
28  R(1) = R1
29  for i = 1:el
30      R(i+1)=R(i)/K
31      r(i)=R(i)-R(i+1)
32      mprintf('Resistance value of section - %d is %f
            Ohms',i,r(i))
33      mprintf('\n')
34  end
35
36  mprintf('Total Starter resistance is %f Ohms',sum(r)
        )
37      mprintf('\n')
38  mprintf('Resistance of motor is %f Ohms',Ra)
39      mprintf('\n')
40  mprintf('Total Resistance at starting time is %f
        Ohms',Ra+sum(r))
41      mprintf('\n')
42  mprintf('Upper range of Staring current is %f Amp',
```
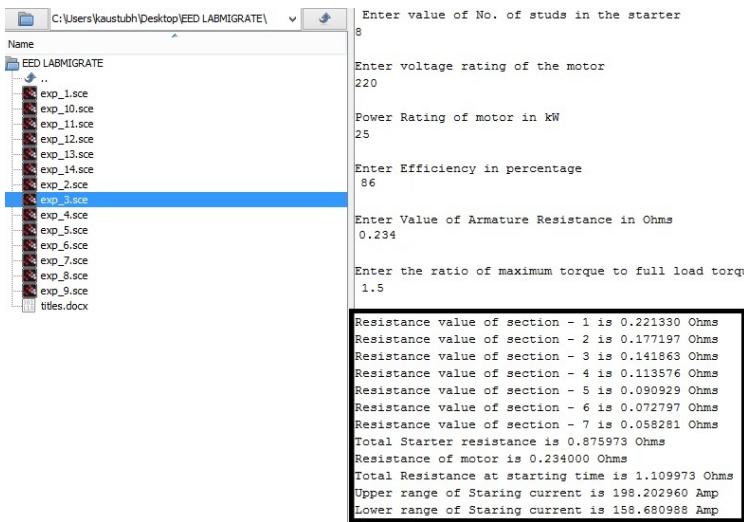
Figure 3.1: Experiment3

```
      Im )
43        mprintf ( '\n' )
44 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

```
1  //Experiment-4
2  // windows 8.1 - 64-Bit
3  //Scilab - 6.0.0
4
5  //Aim : Design of starter for DC series motor
         through SCILAB code
6  // Data: Determine the resistance of each section of
          a strater to be used for DC series motor
7
8  clc
9  clear all
10
11 // Following data is to be taken form the user
12
13 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)
14 V=input('Enter voltage rating of the motor')//
         voltage reating of the motor ( in the range of
```

```
            220 − 400 V )
15  Im=input('Maximum Starting current') // Maximum
        required value of starting current ( in the range
         of 150 −250 Amp)
16  Il=input('Minimum Starting current') // Minimum
        required value of starting current ( in the range
         of 100 −200 Amp)
17  Ra=input('Enter Value of Armature Resistance in Ohms
        ') // Armature resistance ( in the range of 0.1 −
        0.4 ohms)
18  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)
19
20  // Actual Calculations begin
21
22  K = Im/Il // ratio of currents
23  b = Phi / K
24  el=N-1 // No. of element sections
25  R(1) = V/Im
26
27  for i = 1:el
28      R(i+1)=b*R(i)+R(1)*(1-Phi)
29      r(i)=R(i)-R(i+1)
30      mprintf('Resistance value of section − %d is %f
           Ohms',i,r(i))
31      mprintf('\n')
32  end
```
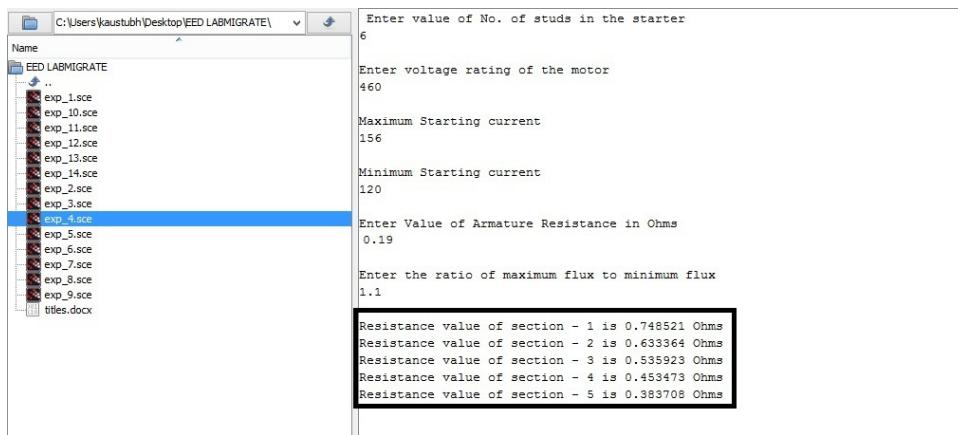
Figure 4.1: Experiment4

Experiment: 5

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

Scilab code Solution 5.5 Experiment5

```
1  //Experiment-5
2  // windows 8.1 - 64-Bit
3  //Scilab - 6.0.0
4
5  //Aim : A SCILAB code to design rotor resistance
       starter of a slip-ring induction motor
6  // Data: Calculate the steps in 5 section rotor
       starter
7
8  clc
9  clear all
10
11 // Following data is to be taken form the user
12
13 P=input('Enter power rating of motor in kW') //
       power rating of motor in kW ( in the range of 1 -
       15 kW)
```

```
14  N=input('Enter no. of studes for starter to be
        designed') // No. of studs ( in range of 5 - 10)
15  s=input('Enter full load slip in percentage') //
        full load slip ( in the range of 1 - 5 %)
16  R=input('Enter value of rotor resistance per phase
        in Ohms') // rotor resistnace ( in the range of
        0.01 - 0.1 Ohms)
17
18  // Actual Calculations begin
19
20  el = N-1      // no. of resistance elements
21  R(1) = R*100/s   // resistance at stud 1
22  K = (s/100)^(1/(N-1))
23
24  for i=1:el
25  R(i+1) = K*R(i)
26  r(i) = R(i)-R(i+1)
27  mprintf('Resistance of section - %d is %f Ohms',i,r(
        i))
28  mprintf('\n')
29  end
```
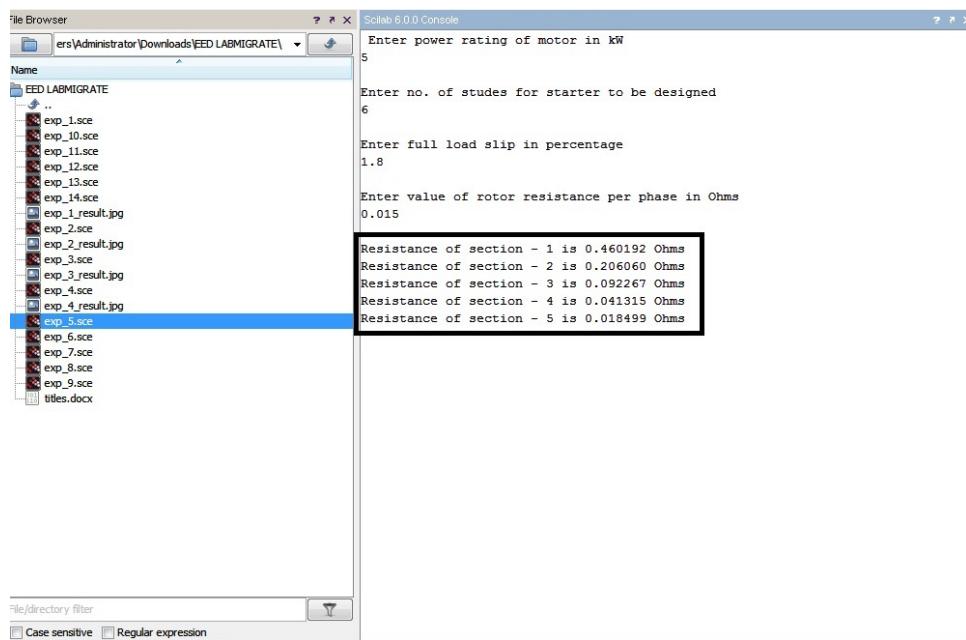
Figure 5.1: Experiment5

Experiment: 6

Design of a small single phase transformer using SCILAB coding

Scilab code Solution 6.6 Exp6

```
1  //Experiment−5
2  // windows  8.1 − 64−Bit
3  //Scilab − 6.0.0
4
5
6  //Aim :  Design  of  a  small  single  phase  transformer
         using  SCILAB  coding
7  //Data:  Design  a  small  single  phase  transformer  for
         given  ratings
8
9  clc
10 clear all
11
12 //  Following  data  is  to  be  taken  form  the  user
13 Vo=input ('Enter  value  of  output  voltage  of
         transformer ') //  output  voltage  of  the
         transformer  ( in  the  range  of  5 − 12  volts )
```

```scilab
14  Io=input('Enter value of output current of
        transformer') // output current of the
        transformer (in the range of 2 - 5 Amp.)
15  Vi=input('Enter value of input voltage of
        transformer') // input voltage of the transformer
         (in the range of 110 - 230 volts)
16  f=input('Enter value of operating frequency in Hz')
        // operating frequency of the transformer (in the
         range of 50 - 60 Hz)
17
18  // Assuming following
19
20  n = 0.9    // assuming 90% eficiency
21  Et = 9 // emf per turn
22  Bm = 1 // maximum flux density
23  Ks = 0.9 // stacking factor
24  del = 2.3 // current density in conductor
25
26  // Actual Calculations begin
27
28  P = Vo*Io // output rating of transformer
29  phi_m = (1/(4.44*f*Et)) // maximum flux in the core
30  Ac = phi_m/Bm // net area of core
31  Ag = Ac/Ks // gross core area
32  A = sqrt(Ag) // width of central limb assuming
        square cross section
33  Np = Vi*Et // No.of turns in primary
34  Ns = ceil(1.05*Vo*Et) // No.of turns in secondary
35  Ip = P/(n*Vi) // Current in primary winding
36  Ap = Ip / del // corss sectional area of bare
        primary conductor
37  dp = sqrt(4*Ap/%pi) // diameter of bare primary
        conductor
38  dpi = dp+0.3 // diameter of insulated primary
        conductor
39  Api = (%pi*dpi^2)/4 // corss sectional area of
        insulated primary conductor
40  As = Io / del // corss sectional area of bare
```

```
                secondary  conductor
41  ds = 0.2+sqrt(4*As/%pi) // diameter  of  bare
                secondary  conductor
42  dsi = ds+0.1 // diameter  of  insulated  secondary
                conductor
43  Asi = (%pi*dsi^2)/4 // corss  sectional  area  of
                insulated  secondary  conductor
44  sfp = 0.8*(dp/dpi)^2 // space  factor  of  primary
                winding
45  Awp = ceil(Np*Api/sfp) // window  area  for  primary
                winding
46  sfs = 0.8*(ds/dsi)^2 // space  factor  of  secondary
                winding
47  Aws = ceil(Ns*Asi/sfs) // window  area  for  secondary
                winding
48  Aw = 1.2*(Aws+Awp) // gross  window  area  required
49
50  mprintf('No.  of  urns  required  in  Primary  is  %d',Np)
51  mprintf('\n')
52  mprintf('No.  of  urns  required  in  Secondary  is  %d',Ns
        )
53  mprintf('\n')
54  mprintf('Power  Rating  of  the  transformer  is  %d VA',P
        )
55  mprintf('\n')
56  mprintf('Diameter  of  insulated  primary  conductor  is
        %f mm',dpi)
57  mprintf('\n')
58  mprintf('Diameter  of  insulated  secondary  conductor
        is  %f mm',dsi)
59  mprintf('\n')
60  mprintf('Gross  window  area  required  is  %d mm^2',Aw)
```
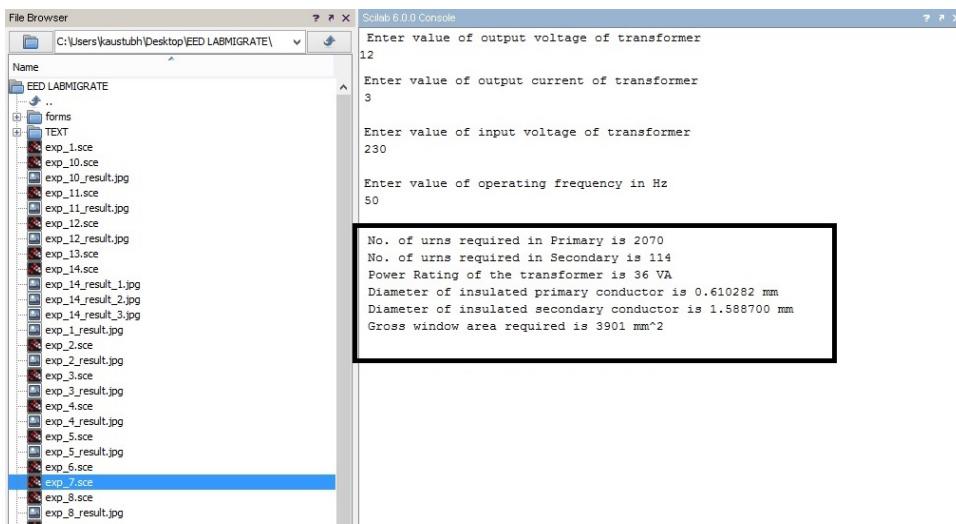
Figure 6.1: Exp6

Experiment: 7

Design of an iron cored choke coil using SCILAB coding

Scilab code Solution 7.7 Experiment7

```
1  //Experiment −7
2  //windows  8.1  −  64−Bit
3  //Scilab  −  6.0.0
4
5  //Aim  :  Design  of  an  iron  cored  choke  coil  using
       SCILAB  coding
6  ////Data:  Design  a  single  phase  variable  choke  coil
7
8  clc
9  clear
10
11  //Following  data  are  to  be  taken  from  user
12
13  V=input('Enter  value  of  supply  voltage ')  //  supply
        voltage  (in  the  range  of  230  −  440  V)
14  f=input('Enter  frequency  of  supply ')  //  supply
        frequency  =  50  Hz  in  India  and  60  Hz  in  US
15  I=input('Enter  value  of  current  to  be  carried  in  Amp
        ')  //  currentn  carrying  capacity  in  Amp  (  in  the
```

```
      range of 5 − 25 Amp)
16  lg=input('Enter maximum airgap length in cms') //
      airgap length varies between 0 and 10 cms being
      variable choke coil
17
18  //Actual Calculations begin
19
20  uo=4*%pi*1e-7 // permeability of free space
21  Sf=0.9 // stacking factor
22  K=(uo*V*I)/(2*%pi*f*2*lg/100)
23  i=1
24  for Bg=0.2:0.1:0.8
25      A_root(i)=sqrt(K)/Bg
26      i=i+1
27  end
28  Bgm=0.45
29  A_root=0.0225
30  A_i=A_root^2 // net iron area
31  Agi=A_i/Sf; // gross iron area
32  A  = sqrt(Agi) // width of limb
33  ATg = Bgm*2*lg/100/uo // airgap mmf
34  ATt = 1.1*ATg // total mmf
35  N  = ceil(ATt / I) // no. of turns
36  del=2.4 // current density
37  a  = I/del // conductor area in mm^2
38  d  = sqrt(4*a/%pi)+.05 // diameter of conductor in
      mm
39  d1 = 2.488 // diameter of insulated conductor
40  a1 = (%pi/4)*d1^2 // cross sectional area of
      insulated conductor
41  sf = 0.8*(d/d1)^2 // space factor
42  Aw = N*a1/sf // area of window
43  AW = 1.2*Aw // gross window area in mm^2
44  Ww = sqrt(AW/2) // width of window in mm
45  Hw = 2*Ww // height of window assuming H/W ratio is
      2
46  hf = Hw-20 // height of winding
47  Nh = ceil(hf/d1) // no. of conductors in height
```

```scilab
48  Nd = N/(2*Nh) // no. of conductors in depth
49  dc = ceil(Nd*d1) // depth of coil
50  dc1 = dc+5 // actual depth of coil
51  hf1 = ceil(hf+10) // actual height of winding
52  dw = ceil(Ww - 2*dc1) // distance betwen two coils
53  D = ceil(Ww + A*1000) // distance between limbs
54  WC = ceil(D+A*1000) // width of core in mm
55  HC = ceil(Hw+2*A*1000+lg*10) // height of core in mm
56  Z= V/I
57  mprintf('Diameter of insulated conductor is %f mm',
        d1)
58  mprintf('\n')
59  mprintf('Area of insulated conductor is %f mm^2', a1
        )
60  mprintf('\n')
61  mprintf('Height of coil is %d mm',hf1)
62  mprintf('\n')
63  mprintf('Depth of coil is %d mm',dc1)
64  mprintf('\n')
65  mprintf('Height of core is %d mm',HC)
66  mprintf('\n')
67  mprintf('Width of core is %d mm',WC)
68  mprintf('\n')
69  mprintf('Impedance of the coil is %d Ohms',Z)
```
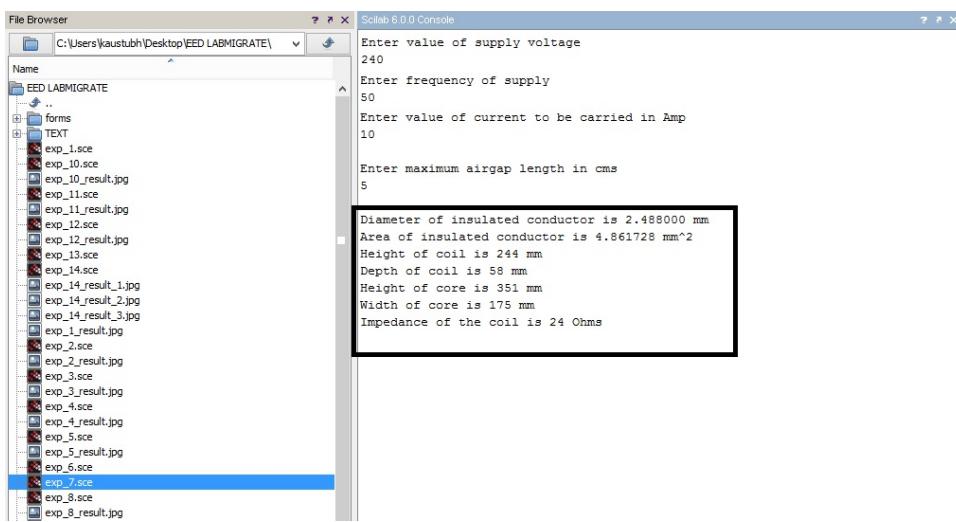
Figure 7.1: Experiment7

Experiment: 8

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

Scilab code Solution 8.8 Exp8

```
1  //Experiment -8
2  // windows 8.1 - 64-Bit
3  //Scilab - 6.0.0
4
5  //Aim : A SCILAB code to workout design of simplex
       lap winding in DC machines
6  //Data: Workout details of simplex lap winding for
      DC generator
7
8  clc;
9  clear all;
10
11  // Following data is to be aken from user
12
13  p=input('Enter No. of poles') // no. poles (in the
       range of 2 - 12) always an even number
14  s=input('Enter No. of slots in armature') // no. of
```
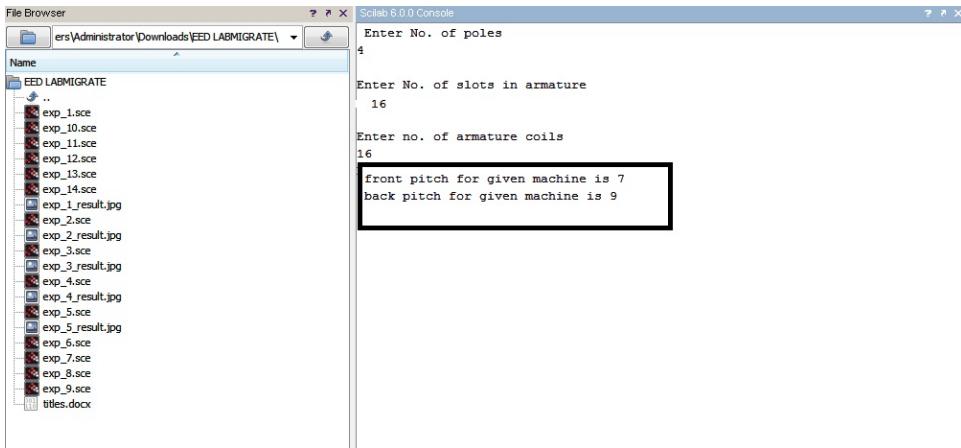
Figure 8.1: Exp8

```
         armature  slots  (in  the  range  of  16  −48)
15  c=input('Enter  no.  of  armature  coils')  //  armature
         coils  in  multiple  of  armature  slots  (in  the  range
          of  16 −  48)
16
17  // Actual  calculations  begin
18
19  cs=c*2  //  no.  of  coil  sides
20  csps=cs/s  //  no.  of  coils  sides  per  slot
21  cspp=cs/p  //  no.  of  coils  sides  per  pole
22
23  yb =  cspp+1  //  back  pitch
24  yf =  cspp-1  //front  pitch
25
26  mprintf('front  pitch  for  given  machine  is  %d',yf )
27  mprintf('\n')
28  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

```scilab
1  //Experiment-8
2  //  windows  8.1 - 64-Bit
3  //Scilab -  6.0.0
4
5  //Aim : A SCILAB code to workout design of simplex
        Wave winding in DC machines
6  //Data: Workout details of simplex wave winding for
       DC generator
7
8  clc;
9  clear all;
10
11 // Following data is to be taken from user
12
13 p=input('Enter No. of poles') // no. poles (in the
       range of 2 - 12) always an even number
14 s=input('Enter No. of slots in armature') // no. of
```
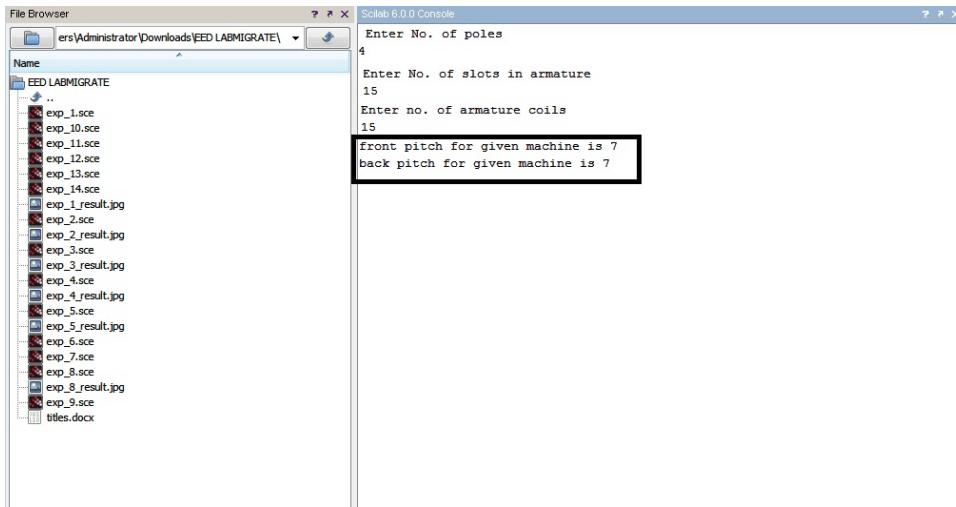
Figure 9.1: Exp9

```
      armature  slots  ( in  the  range  of  10 − 48)
15  c=input ( 'Enter no.  of  armature  coils ')  // armature
      coils  in  multiple  of  armature  slots  ( in  the  range
       of  10 − 48)
16
17  // Actual  calculations  begin
18
19  cs=c*2  // no.  of  coil  sides
20  csps=cs/s  // no.  of  coils  sides  per  slot
21  cspp=cs/p  // no.  of  coils  sides  per  pole
22
23  yb = floor(cs/p)  // back  pitch
24  y = (cs-2)/(p/2)  // winding  pitch
25  yf=y-yb
26
27  mprintf ( 'front  pitch  for  given  machine  is  %d ',yf )
28  mprintf ( '\n ')
29  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

```
1  //Experiment-10
2  // windows 8.1 - 64-Bit
3  //Scilab - 6.0.0
4
5  //Aim : Core Loss Calculations in magnetic materials
        using SCILAB Programming
6  //Data: CAlculate core loss per kg in a specimen of
      ally sheet using user defined data
7
8  clear;
9  clc;
10
11
12  // Following data is to be taken from user
13
14  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)
15  f=input('Enter value of frequency in Hz') //
        frequency of flux reversal (genrally 50 Hz)
16  t=input('Enter thickness of laminated plates in mm')
         // tickness of laminations (in range of 0.2 -
        0.6 mm )
17  r=input('Enter Resistivity of material in Ohm*m')//
        resistivity in Ohm*m (in the range of 0.2 - 0.6
        micro Ohm*m)
18  sg=input('Enter specific gravity of material in kg/m
        ^3') // specific gravity (in the range of 7 - 10
        kg/m^3)
19  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)
20
21  // Actual calculations begin
22
23  Pe=((%pi^2)*(f^2)*(Bm^2)*((t/1000)^2))/((sg*1000)
        *(6*r)) // eddy current loss per kg
24  Ph=hl*f/(sg*1e3)
25  Pi=Pe+Ph
26  mprintf('Eddy Current Loss per kg for given material
         is %f W',Pe)
27  mprintf('\n')
28  mprintf('Hysterisis Loss per kg for given material
        is %f W',Ph)
29  mprintf('\n')
30  mprintf('Total Iron Loss per kg for given material
        is %f W',Pi)
```
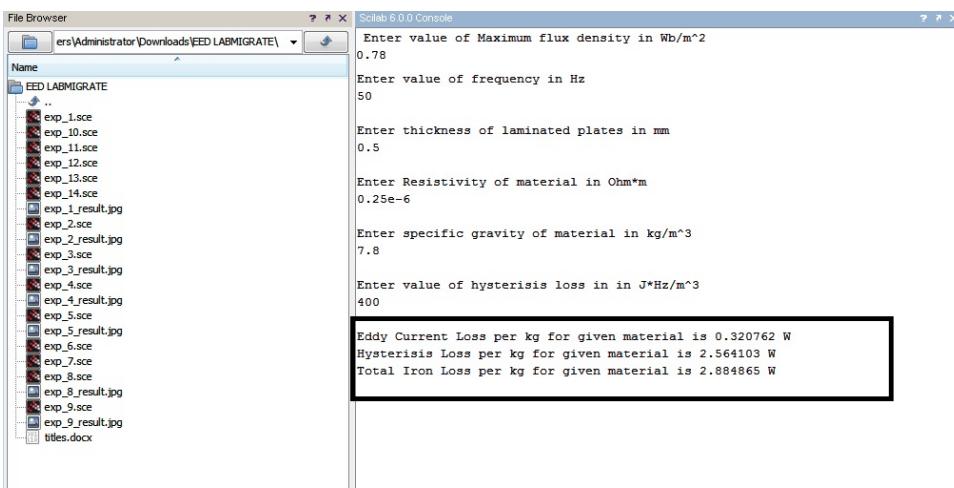
Figure 10.1: Experiment10

Experiment: 11

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

Scilab code Solution 11.11 Experiment11

```
1  //Experiment-11
2  // windows 8.1 - 64-Bit
3  //Scilab - 6.0.0
4
5  //Aim : A SCILAB program for Computation of slot
       leakage reactance in induction motor
6  // Data: Find out slot leakage reactance for
       induction motor from given slot dimensions
7
8  clc;
9  clear all;
10
11 // Data to be taken from user
12
13 T=input('Enter no. of turns per phase') // turns per
        phase for induction motor (in the range of 200 -
        300)
```

```
14  s=input('Enter no. of slots per phase') // slots per
        phase for induction motor (in the range of 12 −
       24)
15  bs=input('Enter slot width in mm') // slot opening
       in mm (in the range of 10 −30 mm)
16  L=input('Enter core / slot length in mm') // slot
       length in mm (in the range of 150 −650 mm)
17  wo=input('Enter lip opening in mm') // width of lip
       in mm (in the range of 2 − 6 mm)
18  h1=input('Enter heigth of conductor in slot in mm')
        // heigth of conductor portion in mm (in the
       range of 20 − 80 mm)
19  h2=input('Enter Value of clearance between
       conductors and wedge') // clearnace in mm (in the
        range of 1 − 4 mm)
20  h3=input('Enter heigth of wedge in mm') // wedge
       height in mm (in the range of 2 − 6 mm)
21  h4=input('Enter height of lip in mm') // lip height
       in mm (in the range of 1 − 4 mm)
22
23  // Actual calculations begin
24
25  f=50; // frequency of supply
26  uo=4*%pi*1e-7; // pearmeabilty of free space
27  Ls=(h1/(3*bs))+(h2/(bs))+(2*h3/(bs+wo))+(h4/wo)
28  xs=8*%pi*f*uo*T^2*(L/1000)*Ls/s
29
30  mprintf('Leakage reactance for given machine is %f
       Ohms',xs)
```
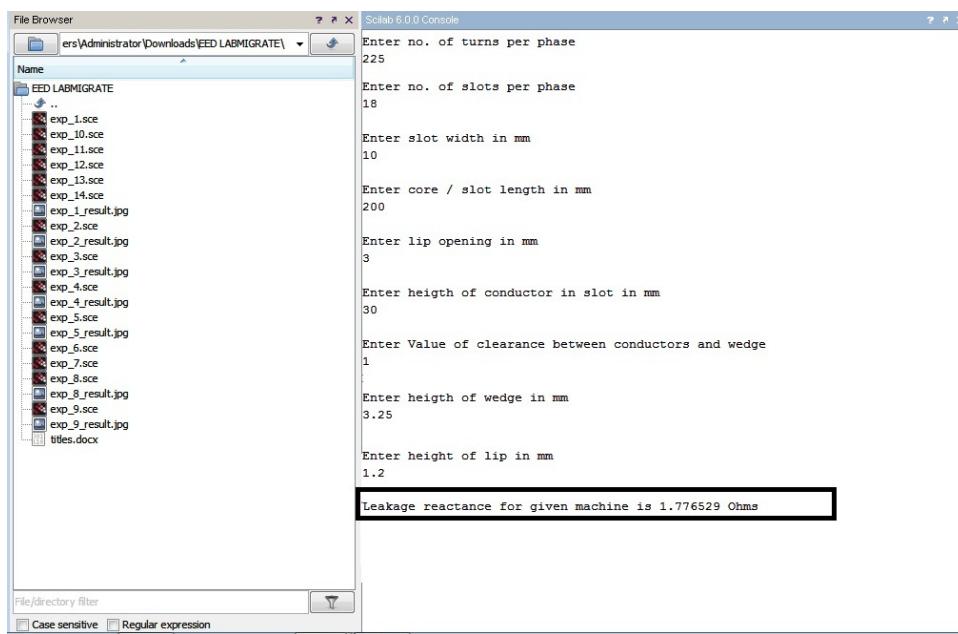
Figure 11.1: Experiment11

Experiment: 12

A SCILAB program for Design of Electromagnet

Scilab code Solution 12.12 Experiment12

```
1  //Experiment −12
2  //windows  8.1  −  64−Bit
3  //Scilab  −  6.0.0
4
5  //Aim :A SCILAB program for Design of Electromagnet
6  //Data:Design a flat faced armature type of circular
         electromagnet
7
8  clc
9  clear
10
11  //Following data is to be taken form the user
12
13  F=input('Enter Amount of force in kg') // force to
       be exerted by electromagnet in kg(in the range of
       100 − 250 kg)
14  s=input('Enter value of stroke in mm') // stroke in
       mm (in the range of 0.5 − 1.5 mm)
15  v=input('Enter supply volage') // in the range of 5
```

```scilab
            − 15  volts
16  ta=input('Enter  ambient  temperature  in  degree
        celcius ')  // general  value  is  20 oC
17  t=input('Enter  permissible  temperature  rise  above
        ambient  temperature ')  // 50 − 80 oC
18
19  //Actual  Calculations  begin
20
21  Bm = 1.1  // maximum  flux  density  in  Wb/m^2
22  uo = 4*%pi*1e-7
23  ro_0=0.01734  // resistib=vity  at  0  degree
24  alp_0=0.00393  // temperature  coefficient  of
        resistance
25  h_by_d=3  // heigth  to  depth  ratio  of  coil
26  C=0.085  // constant
27  Sf=0.5  // assumed  space  factor
28
29  A   = F*uo/(0.102*Bm^2)  // are  of  central  limb
30  r1 = sqrt(A/%pi)  // radius  of  central  limb
31  //disp(r1)
32  AT=(1600000*Bm*s*1e-3)/0.85  // total  mmf  required
33  ro_70=ro_0*(1+(alp_0*t))
34  hc=((3*ro_70*C*AT^2)/(2*Sf*t)*1e-6)^(1/3)  // heigth
        of  coil
35  //disp(hc)
36  dc=hc/3  //depth  of  coil
37  //disp(dc)
38  r2=dc+r1
39  //disp(r2)
40  t1=r1/2
41  //disp(t1)
42  t2=(r1^2)/(2*r2)
43  //disp(t2)
44  r3=sqrt(r1^2+r2^2)
45  //disp(r3)
46  a=(AT*ro_70*%pi*(r1+r2))*1e-3/v  // cross  sectional
        area  of  conductor
47  //disp(a)
```

```
48  d=sqrt(4*a*1000/%pi) // diameter of conductor
49  //disp(d)
50  dc1=dc*1000-2 // depth of coil considering clearance
51  //disp(dc1)
52  nd=ceil(dc1/d) // no. of layers in depth
53  //disp(nd)
54  hc1=hc*1000-3 // height of coil considering
         clearance
55  nh=ceil(hc1/d) // no o flayers in height
56  //disp(nh)
57  T=nd*nh // no. of turns in coil
58  //disp(T)
59  ab=(%pi/4*d^2) // diameter of bare conductor
60  R=T*ro_70*(%pi*(r1+r2))/ab // resistance of coil
61  //disp(R)
62  I=v/R; // current in coil
63  //disp(I)
64  mmf=I*T // actual mmf developed by coil
65  //disp(mmf)
66  sf=T*ab/(hc*dc) // actual space factor
67  //disp(sf)
68  theta = (ro_70*C*mmf^2*1e-6)*1e6/(2*sf*dc*hc^2)
69  //disp(theta  )
70
71  mprintf('heigth of coil is %f mm',hc*1000)
72  mprintf('\n')
73  mprintf('depth of coil is %f mm',dc*1000)
74  mprintf('\n')
75  mprintf('total no. of turns in the coil are %d',T)
76  mprintf('\n')
77  mprintf('Current flowing through the coil is  %f Amp
         ',I)
78  mprintf('\n')
79  mprintf('MMF developed by the coil is %d AT',mmf)
80  mprintf('\n')
81  mprintf('Temperature rise of the coil is %f oC',
         theta)
82  mprintf('\n')
```
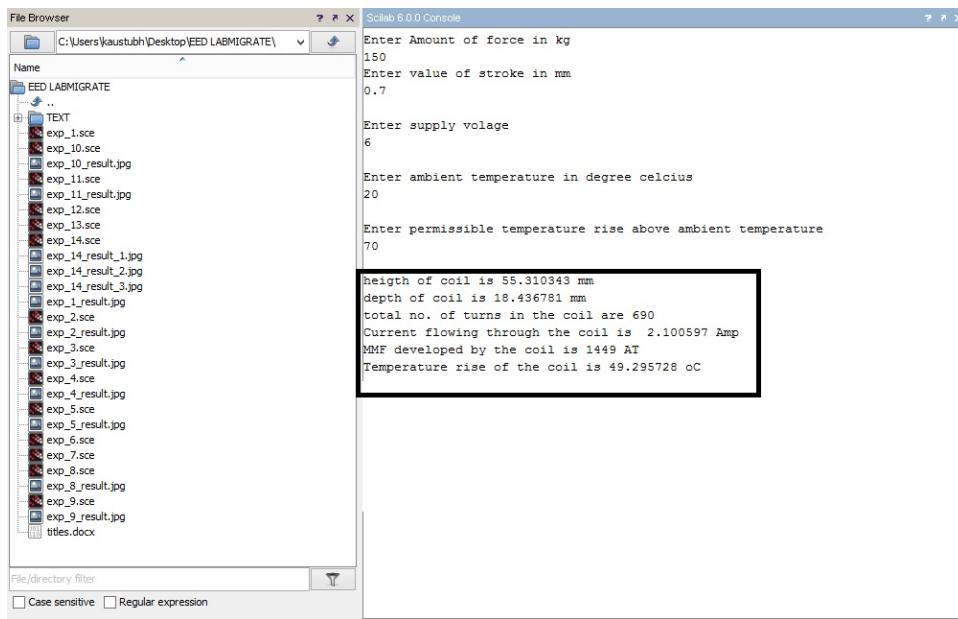
Figure 12.1: Experiment12

Experiment: 13

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

Scilab code Solution 13.13 Exp13

```
1  //Experiment −13
2  // windows 8.1 − 64−Bit
3  //Scilab − 6.0.0
4
5  //Aim : Computation of winding factor and
       distribution factor in armature winding using
       SCILAB programming
6  //Data: Calculate winding factor and distribution
       factor for armature winding of synchronous
       machine
7
8  clc;
9  clear all;
10
11  // Data to be taken from the user
```

```scilab
12
13  p=input('Enter no. of poles') // no. of poles in
        synchronous machine (in the range of 2 - 12)
14  s=input('Enter no. of slots') // no. of slots in
        synchronous machine ( in the range of 16 - 96)
15  pitch=input('Enter No. of short pitch slots') // no.
         of slots by which short pitching is required (
        in the range of 1 to 4)
16
17  // Actual calculations start
18
19  sp = s/p // slots per pole
20  spp = sp/3 // slots per pole per phase
21  dist = %pi/spp //
22  spread = %pi/sp // phase spread angle
23  alpha = spread*pitch // angle of short pitch
24  bet = %pi/p // distribution angle
25
26  kp1=cos(alpha/2) // pitch factor for fundamental
27  kp5=cos(5*alpha/2) // pitch factor for 5th harmonic
28
29  kd1=(sin(spp*bet/2))/(spp*sin(bet/2)) //
        distribution factor for fundamental
30  kd5=(sin(spp*5*bet/2))/(spp*sin(5*bet/2)) //
        distribution factor for 5th harmonic
31
32  mprintf('Pitch factor for fundamental is %f',kp1)
33  mprintf('\n')
34  mprintf('Pitch factor for 5th harmonic is %f',kp5)
35  mprintf('\n')
36  mprintf('Distribution factor for fundamental is %f',
        kd1)
37  mprintf('\n')
38  mprintf('Distribution factor for 5th harmonic is %f'
        ,kd5)
39  mprintf('\n')
```
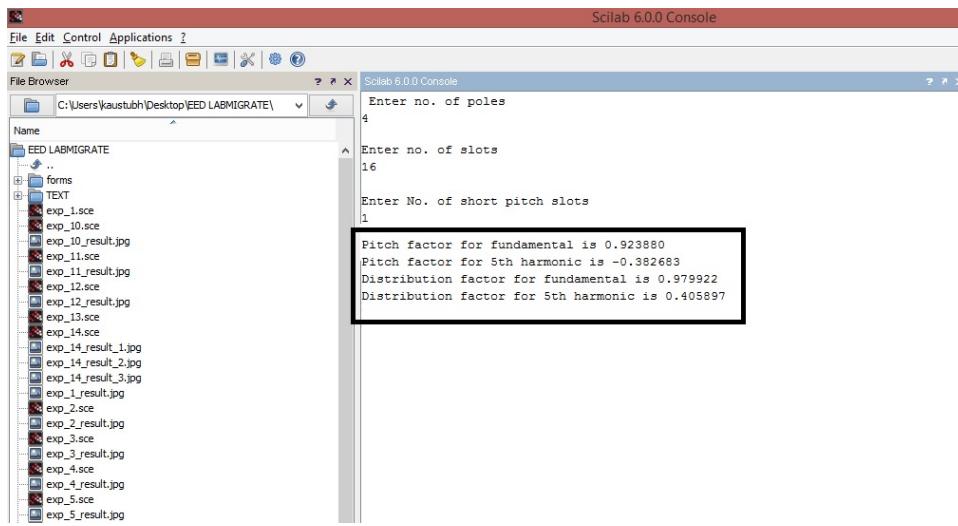
Figure 13.1: Exp13

Experiment: 14

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

Scilab code Solution 14.14 Experiment14

```
1  //Experiment −14
2  //  windows  8.1  − 64−Bit
3  //Scilab  −  6.0.0
4  //Aim :Analyzing  variation  of  slot  leakage  reactance
        in  induction  motor  using  SCILAB  code
5  //  Data:  Plot  graphs  showing  effect  of  no.  of  slots ,
        length  of  armature  core  and  no.  of  turns  on  slot
        leakage  reactance
6
7  clc ;
8  clear ;
9
10 //  Assumed  data
11
12
13
14 bs =10  //  slot  opening  in  mm ( in  the  range  of  10  −30
```

```scilab
     mm)
15  wo=3 // width of lip in mm (in the range of 2 - 6 mm
        )
16  h1=30 // heigth of conductor portion in mm (in the
        range of 20 - 80 mm)
17  h2=1 // clearnace in mm (in the range of 1 - 4 mm)
18  h3=3.25 // wedge height in mm (in the range of 2 - 6
        mm)
19  h4=1.2 // lip height in mm (in the range of 1 - 4 mm
        )
20
21  f=50; // frequency of supply
22  uo=4*%pi*1e-7; // pearmeabilty of free space
23  Ls=(h1/(3*bs))+(h2/(bs))+(2*h3/(bs+wo))+(h4/wo)
24
25  // Evaluating variation in leakage reactance with
        change in no. of slots
26  L=200
27  T=225
28  for s = 1:25
29      xs(s)=8*%pi*f*uo*T^2*(L/1000)*Ls/s
30  end
31
32  x=1:25
33  figure(1)
34  plot(x,xs)
35  xlabel('No. of slots')
36  ylabel('Leakage reactance in Ohms')
37  title('Variation of leakage reactance with change in
        no. of slots')
38
39  clear xs s T L
40  // Evaluating variation in leakage reactance with
        change in length of core
41  L = 100:10:2000
42  s = 20
43  T = 225
44  for i = 1:length(L)
```

```
45      xs(i)=8*%pi*f*uo*T^2*(L(i)/1000)*Ls/s
46  end
47  figure(2)
48  plot(L,xs)
49  xlabel('length of core in mm')
50  ylabel('Leakage reactance in Ohms')
51  title('Variation of leakage reactance with change in
        length of core')
52
53  clear xs s T L
54  // Evaluating variation in leakage reactance with
        change in no. of turns per phase
55  L = 200
56  s=20
57  T = 100:400
58  for i = 1:length(T)
59      xs(i)=8*%pi*f*uo*T(i)^2*(L/1000)*Ls/s
60  end
61  figure(3)
62  plot(T,xs)
63  xlabel('No. of turns per phase')
64  ylabel('Leakage reactance in Ohms')
65  title('Variation of leakage reactance with change in
        no. of turns per phase')
```
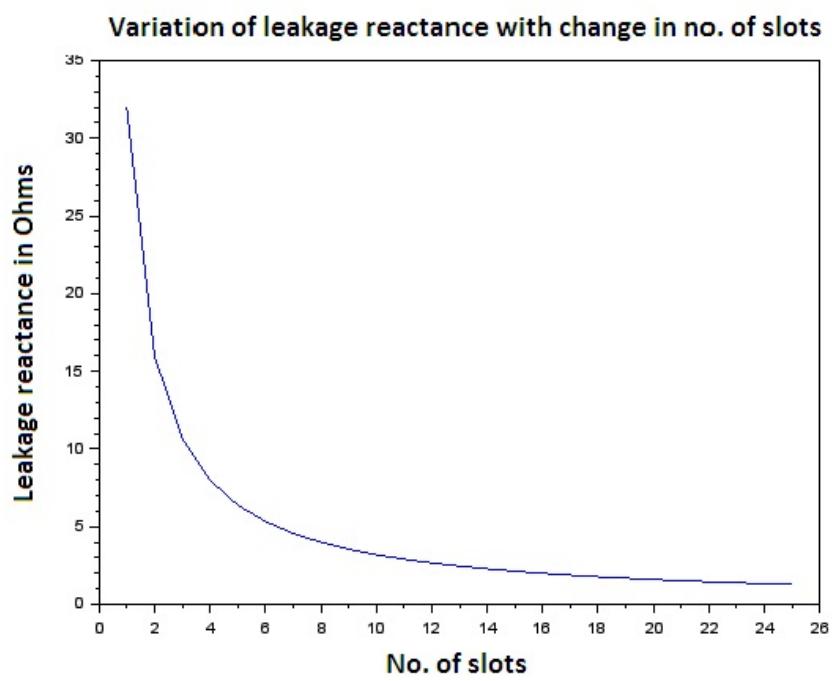
Figure 14.1: Experiment14

Variation of leakage reactance with change in length of core

Variation of leakage reactance with change in no. of turns per phase
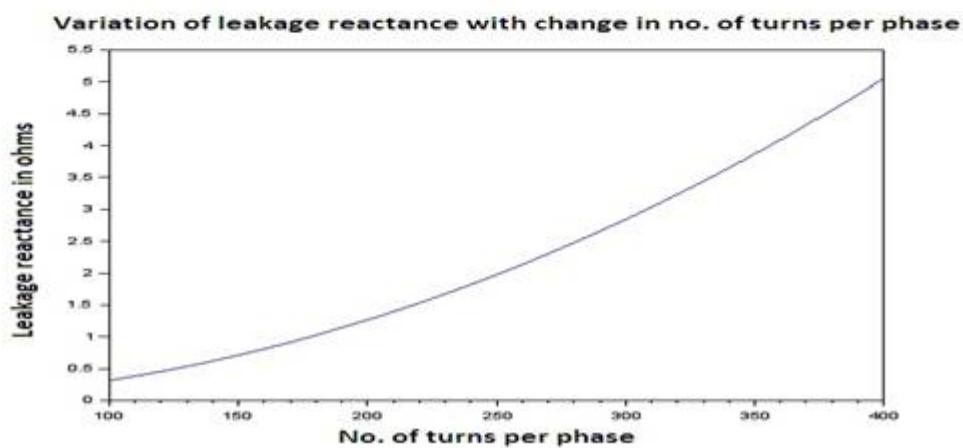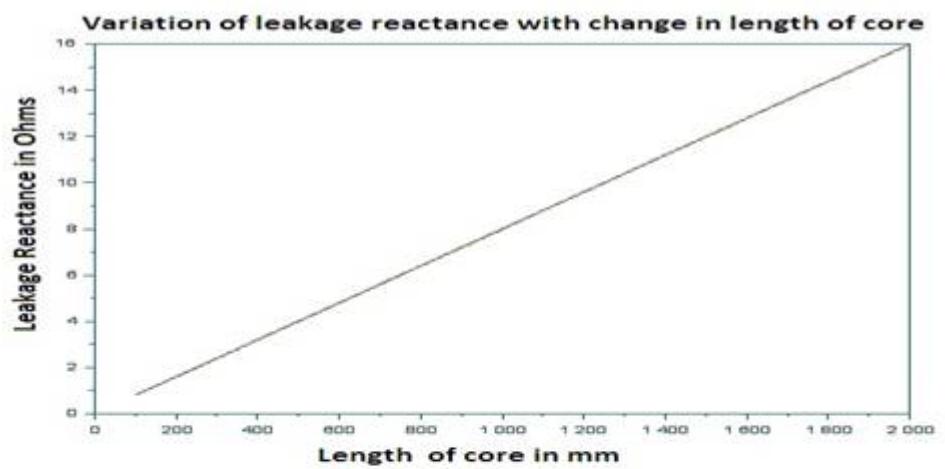
Figure 14.2: Experiment14