Electrical Machinery
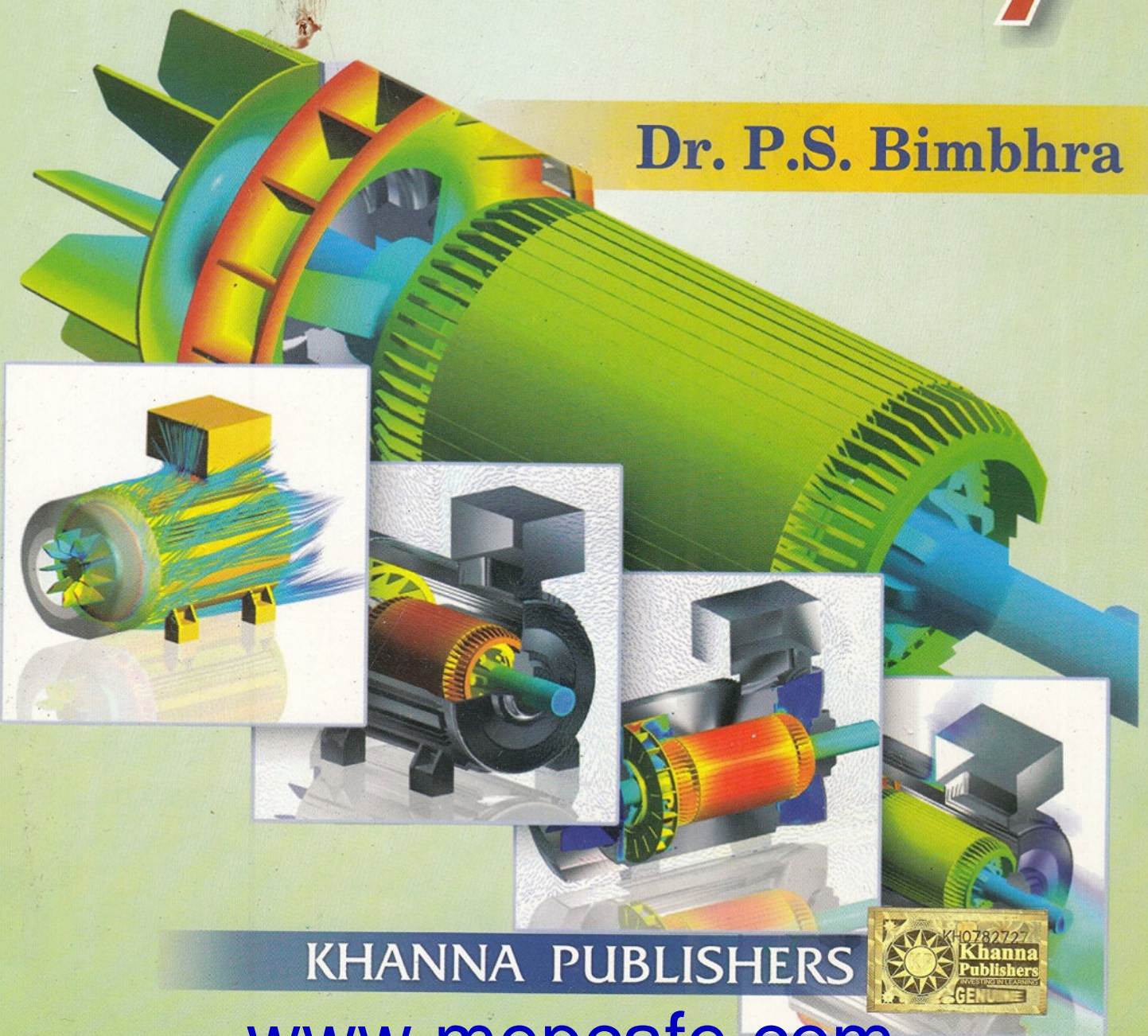
Dr. P.S. Bimbhra

KHANNA PUBLISHERS

Scilab Textbook Companion for Electrical Machinery by Dr. P S Bimbhra[1]

Created by
Tejash Ajay Sharma
B.Tech
Electrical Engineering
Visvesvaraya National Institute of Technology
College Teacher
None
Cross-Checked by
None

April 4, 2017

Book Description

Title: Electrical Machinery

Author: Dr. P S Bimbhra

Publisher: Khanna Publishers, New Delhi

Edition: 7

Year: 1995

ISBN: 81-7409-173-4

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

www.mepcafe.com

List of Scilab Codes

6

7

www.mepcafe.com

8

9

10

www.mepcafe.com

11

12

13

15

16

17

www.mepcafe.com

18

19

20

Chapter 1

Transformers

Scilab code Exa 1.1 finding number of primary and secondary turns and net cross section area of core

```
1
2 clc;
3 f=50; // frequency in Hz
4 Et=13; // emf per turn in volts
5 E1=2310; // primary voltage in volts
6 E2= 220; // secondary voltage in volts
7 B=1.4; // maximum flux density in Tesla
8 // calculating the number of turns in primary and
      secondary side
9 N2= round(E2/Et); //secondary side turns
10 printf('Number of secondary turns is %f\n',N2);
11 N1=round(N2*(E1/E2));// primary side turns
12 printf('Number of primary turns is %f\n',N1);
13 disp('The value of primary turns does not satisfy
      with the');
14 disp('value of secondary turns so taking value of N2
      =18(next nearest integer)');
15 N2=18; // new value of secondary turns
16 N1=18*(E1/E2);
17 printf('Number of primary turns is %f\n',N1);
```

```
18  printf('Number of secondary turns is %f\n',N2);
19  // calculating net core area
20  A=(220/(18*sqrt(2)*%pi*B*50))*10^4; // where N2=18
21  printf('Net area of core is %f cm^2',A);
```

Scilab code Exa 1.2 Determining number of turns in all three windings of transformer

```
 1  clc;
 2  f=50; // frequency in hertz
 3  B=1.2; // maximum flux density in Tesla
 4  A=75*10^-4; // net core area in m^2
 5  E1=220; // primary side voltage in volts
 6  E2=600; // secondary side voltage in volts
 7  E3=11; // tertiary side voltage in volts
 8  n3=round(E3/2); // number of turns in half of the
        tertiary winding
 9  Et=round(sqrt(2)*%pi*50*B*A); // calculating emf per
         turn
10  N3=Et*n3; // total number of turns in tertiary
        winding
11  printf('total number of turns in  tertiary winding
        is %f\n',N3);
12  N2=round(E2*(n3/E3)); // total number of turns in
        secondary winding
13  printf('total number of turns in  secondary winding
        is %f\n',N2);
14  N1=round(E1*(n3/E3)); //  total number of turns in
        secondary winding
15  printf('total number of turns in primary winding is
        %f',N1);
```

Scilab code Exa 1.3 the components of exciting current primary current and power factor

```
1 clc;
2 f=50; // frequency in hertz
3 E1=2200; // supply voltage in volts
4 E2=220; // secondary side voltage in volts
5 P=361; // core loss in watts
6 Io=0.6; // exciting current in Ampere
7 Is=60; // secondary load current in Ampere
8 pf=0.8; // power factor
9 Ic=P/E1; // core loss component of current
10 printf('core loss component of exciting current is
      %f A\n',Ic);
11 Im=sqrt(Io^2-Ic^2); // magnetising component of
      current
12 printf('magnetising component of exciting current is
      %f A\n',Im);
13 ip=Is*(E2/E1); // primary current required to
      neutralise the secondary current
14 Iv=ip*pf+Ic; // total vertical compartment of
      primary current
15 Ih=ip*0.6+Im; // total horizontal compartment of
      primary current,pf cos(theta)=0.8 so sin(theta)
      =0.6
16 Ip=sqrt(Iv^2+Ih^2); // toatl primary current
17 printf('Total primary current is %f A\n',Ip);
18 ppf=Iv/Ip; // primary power factor
19 printf('primary power factor is %f (lagging)',ppf);
```

Scilab code Exa 1.4 finding saving in weight of core and wire material

```
1 clc;
2 disp('weight of laminations is directly proportion
      to core volume density, which is directly
```

23

proportional to product of area and height of limbs and while taking the ratio of weight of CRGO laminations and hot rolled laminations, height of limbs gets cancelled out(height of limbs are assumed to be equal). So, in the end ratio of weights of laminations is equal to ratio of area of core.Now area of core is given by maximum flux/flux density.According to question maximum flux remain same so ,while taking ratio of areas the maximum flux gets cancelled')

```
3  B1=1.2; //flux density in hot rolled steel
       laminations
4  B2=1.6; //flux density in CRGO steel laminations
5  W1=100; // weight of H.R core in kg
6  W2=W1*(B1/B2); // calculating weight of CRGO
       laminations in kg
7  s=((W1-W2)/W1)*100; // calculating saving in core
       material
8  printf('percentage saving in core material is %f
       percent\n',s);
9  disp('weight of wire is directly proportional to
       product of length of turn around core and cross
       section of wire.(Wire cross section is assumed to
        be same in CRGO and HR laminations so gets
       cancelled out while taking  ratio) also the
       length of turn is inversely proportional to
       square root of flux density  ')
10 w1= 80 // weight of Hot rolled wire
11 w2=w1*(sqrt(1.2/1.6)); // weight of CRGO wire
12 s=((w1-w2)/w1)*100; //saving in weight of wire
13 printf('Percentage saving in weight of wire is %f
       percent',s);
```

Scilab code Exa 1.5 determination of no load voltage and frequency of secondary side

```
 1 clc;
 2 v1=240; // high voltage side voltage
 3 v2=120; // low voltage side voltage
 4 f1=50; // frequency in Hz
 5 disp('v1 is directly proportional to product of
       frequency and maximum flux. considering q1 be
       maximum flux for v1 and q2 be maximum flux for
       v11 then Q=q2/q1 can be calculated as follow ')
 6 disp('case a')
 7 v11=240; // new supply voltage
 8 f2=40; // new supply frequency
 9 Q=(v11*f1)/(v1*f2);
10 v22=(v2*f2*Q)/f1;
11 printf('secondary voltage for case a is %f v\n',v22)
       ;
12 disp('case b')
13 v11=120; // new supply voltage
14 f2=25; // new supply frequency
15 Q=(v11*f1)/(v1*f2);
16 v22=(v2*f2*Q)/f1;
17 printf('secondary voltage for case a is %f v\n',v22)
       ;
18 disp('case c')
19 v11=120; // new supply voltage
20 f2=50; // new supply frequency
21 Q=(v11*f1)/(v1*f2);
22 v22=(v2*f2*Q)/f1;
23 printf('secondary voltage for case a is %f v\n',v22)
       ;
24 disp('case d')
25 v11=480; // new supply voltage
26 f2=50; // new supply frequency
27 Q=(v11*f1)/(v1*f2);
28 v22=(v2*f2*Q)/f1;
29 printf('secondary voltage for case a is %f v\n',v22)
       ;
30 disp('case e')
31 v11=240; // new supply voltage
```

```
32  f2 =0; // new supply frequency
33  disp ( ' since frequency is zero. Source is a DC source
        so a very  high current will flow in primary
        side which will damage the transformer and the
        secondary induced emf is zero ')
```

Scilab code Exa 1.6 finding primary winding current and power factor

```
1  clc ;
2  N1 =100; // no. of primary turns
3  N2 =160; // No. of secondary turns
4  N3 =60; // No. of tertiary turns
5  I2 =10; // secondary side current
6  I3 =20; // tertiary side current
7  F2 = N2 * I2 ; // mmf of secondary winding
8  F3 = N3 * I3 ; // mmf of tertiary winding
9  disp ( ' load connected to secondary is purely
        resistive and load connected to tertiary is
        purely capacitive ' );
10  F23 = sqrt ( F2 ^2+ F3 ^2); // resultant load mmf
11  F1 = F23 ; // primary winding mmf balances this load
        mmf
12  I1 = F1 / N1 ;
13  printf ( ' primary current is %f A\n ' , I1 );
14  pf = F2 / F1 ;
15  printf ( ' primary side power factor is %f leading ' , pf )
        ;
```

Scilab code Exa 1.8 Determining 1 primary resistance and reactance referred to secondary 2 secondary resistance and reactance referred to primary 3 equivalent resistance and leakage reactance referred to primary and secondary 4 total ohmic losses 5 voltage applied to hv side

26

```
1  clc;
2  P=33000; // rated power of transformer
3  E1=2200; // primary voltage
4  E2=220; // secondary voltage
5  k=E2/E1; // turn's ratio
6  r1=2.4; //primary winding resistance in ohm
7  x1=6; // primary winding reactance in ohm
8  r2=0.03; //secondary winding resistance in ohm
9  x2=0.07; //secondary winding reactance in ohm
10 r12=r1*k^2; //primary resistance referred to
       secondary
11 x12=x1*k^2; //primary reactance referred to
       secondary
12 printf('primary resistance and reactance referred to
        secondary are %f ohm and %f ohm\n',r12,x12);
13 r21=r2/k^2; //secondary resistance referred to
       primary
14 x21=x2/k^2; //secondary reactance referred to
       primary
15 printf('secondary resistance and reactance referred
       to primary are %f ohm and %f ohm\n',r21,x21);
16 re1=r1+r21;
17 xe1=x1+x21;
18 printf('equivalent resistance and reactance referred
        to primary are %f ohm and %f ohm\n',re1,xe1);
19 re2=r2+r12;
20 xe2=x2+x12;
21 printf('equivalent resistance and reactance referred
        to secondary are %f ohm and %f ohm\n',re2,xe2);
22 Ip=P/E1;
23 printf('primary full load current is %f A\n',Ip);
24 Is=P/E2;
25 printf('secondary full load current is %f A\n',Is);
26 O=Ip^2*re1;
27 printf('ohmic losses at full load is %f W\n',O);
28 Ils=160; // secondary side load current
29 Ilp=Ils*k; // primary side load current
30 Ze1=sqrt(re1^2+xe1^2);
```

27

```
31  V = Ilp * Ze1 ;
32  printf ( ' Voltage  applied  to  h . v  side  in  order  to
         obtain  160A  short  circuit  cirrent  in  low  voltage
         winding  is  %f  V\n ' ,V ) ;
33  Pi = Ilp ^2* re1 ;
34  printf ( ' power  input  is  %f W ' ,Pi ) ;
```

Scilab code Exa 1.9 finding secondary terminal voltage

```
 1  clc ;
 2  P =10000;  // rated  power  of  transformer
 3  E1 =2500;  //  rated  primary  side  voltage
 4  E2 =250;  //  rated  secondary  side  voltage
 5  //  initialising  primary  side  parameters
 6  r1 =4.8;  //  primary  resistance  in  ohm
 7  x1 =11.2;  //  primary  leakage  reactance  in  ohm
 8  // initialising  secondary  side  parameters
 9  r2 =0.048;  //  secondary  resistance  in  ohm
10  x2 =0.112;  //  secondary  leakage  reactance  in  ohm
11  k = E2 / E1 ;  //  turn ' s  ratio1
12  z =5+ %i *3.5;
13  re2 = r2 + r1 * k ^2;  // resistance  referred  to  secondary
14  xe2 = x2 + x1 * k ^2;  // reactance  referred  to  secondary
15  ze2 = re2 + %i * xe2 ;
16  zt = z + ze2 ;  //  total  load  on  secondary
17  Z = abs ( zt ) ;
18  I2 = E2 / Z ;  //  load  current  on  secondary
19  disp  ( ' case  a ' ) ;
20  V2 = round ( I2 * abs ( z ) ) ;
21  printf ( ' secondary  terminal  voltage  is  %f  V\n ' ,V2  ) ;
22  disp  ( ' case  b ' ) ;
23  pf =0.8;  //  power  factor
24  I2l = P / E2 ;  //  rated  current  of  secondary  side
25  VD = I2l *( re2 * pf + xe2 * sqrt (1 - pf ^2) ) ;  //  voltage  drop  in
         transformer  leakage  impedance
```

```
26  Vt=E2-VD;
27  printf('secondary terminal voltage is %f V',Vt)
```

Scilab code Exa 1.10 finding primary current secondary terminal voltage power factor power output and efficiency

```
1
2  clc;
3  E1=250;// voltage on low tension side
4  E2=2500; // voltage on high tension side
5  k=E2/E1; //turns ratio
6  Z=380+230*%i; // given load connected to high
        tension side
7  Zl=Z/k^2; // load referred to low tension side
8  zl=0.2+0.7*%i; // leakage impedance of transformer
9  zt=Zl+zl; // total series impedance
10 ztm=abs(zt); // magnitude of total series impedance
11 I1=E1/zt;
12 I1m=abs(I1); // magnitude of primary load current
13 I2=I1m/k; // secondary load current
14 vt=5*abs(Z);
15 printf('secondary terminal voltage is %f V\n',vt);
16 R=500; // shunt branch resistance
17 X=250; // shunt branch leakage reactance
18 Ic=E1/R; // core less component of current
19 Im=E1/(%i*X); // magnetizing component of current
20 Ie=Ic+Im;// total exciting current
21 It=I1+Ie;// total current on low tension side
22 Itm=abs(It);
23 printf('primary current is %f A\n',Itm);
24 pf=cos(atan(imag(It),real(It)));
25 printf('power factor is %f lagging\n',pf),
26 lpf=real(Z)/abs(Z);
27 op=vt*I2*lpf;
28 printf('output power is %f W\n',op);
```

29

```scilab
29 pc=Ic^2*R; // core less power
30 poh=I1m^2*real(zl); // ohmic losses
31 pin=E1*Itm*pf; // input power
32 n=(op/pin)*100; // efficiency
33 printf('efficiency of transformer is %f percent',n);
```

Scilab code Exa 1.11 determining parameters of equivalent circuit referred to l v and h v sides

```scilab
1 clc;
2 p=10000; // rated output of transformer
3 E1=2500; // primary side rated voltage
4 E2=250; // secondary side rated voltage
5 k=E2/E1; // turn's ratio
6 // initialising results of open circuit results on l
      .v side
7 Vo=250; //open circuit voltage
8 Io=1.4; // no load current
9 Po=105; // open circuit voltage
10 // initialising the results of short circuit results
       on h.v side
11 Vsc=104; // short circuit voltage
12 Isc=8; // short circuit current
13 Psc=320; // power dissipated
14 theta=Po/(Vo*Io); // no load power factor
15 Ic=Io*theta; // core less component of current
16 Im=Io*sqrt(1-theta^2); // magnetising component of
      current
17 Ro=round(Vo/Ic); // shunt branch resistance
18 Xo=round(Vo/Im); // shunt branch impedance
19 Zsc=Vsc/Isc; // short circuit impedance
20 reh=Psc/Isc^2; // total transformer resistance
21 xeh=sqrt(Zsc^2-reh^2); // total transformer leakage
      impedance
22 // equivalent circuit referred to l.v side
```

30

```
23  rel=reh*k^2;
24  xml=xeh*k^2;
25  printf('shunt branch resistance and reactance is %f
        ohm and %f ohm\n',Ro,Xo);
26  printf('value of transformer resistance and leakage
        reactance referred to l.v side is %f ohm and %f
        ohm\n',rel,xml);
27  // equivalent circuit referred to h.v side
28  Rch=Ro/k^2;
29  Xmh=Xo/k^2;
30  printf('shunt branch resistance and reactance
        referred to h.v side is %f ohm and %f ohm\n',Rch,
        Xmh);
31  printf('value of transformer resistance and leakage
        reactance referred to h.v side is %f ohm and %f
        ohm\n',reh,xeh);
```

Scilab code Exa 1.12 determining equivalent circuit parameters referred to h v side and its efficiency

```
1  clc;
2  P=200000; // rated power output of transformer
3  E1=11000; // primary side voltage
4  E2=400; // secondary side voltage
5  // initialising the results of the open circuit test
        performed on l v side
6  Vo=400; // open circuit voltage in V
7  Io=9; // no load current in A
8  Po=1500; // core loss in W
9  // initialising the results of short circuit test
        performed on h v side
10 Vsc=350; // voltage applied in short circuit test
11 Isc=P/(3*E1); // short circuit current
12 Psc=2100; // power dissipated in short circuit test
13 E2p=E2/sqrt(3); // per phase voltage
```

31

```
14  pop=Po/3; // per phase core loss
15  Ic=pop/E2p; // core loss current
16  Im=sqrt(Io^2-Ic^2); // magnetising component of
        current
17  R=E2p/Ic; // core loss resistance in ohm
18  X=E2p/Im; // magnetizing reactance
19  Rh=R*(E1/E2p)^2; // core loss resistance referred to
         h v side
20  Xh=floor(X*(E1/E2p)^2); // magnetizing component
        referred to h v side
21  printf('coreloss resistance and magnetizing
        reactance referred to h v side is %f ohm and %f
        ohm\n ',Rh,Xh);
22  Pscp=Psc/3; // ohmic loss per phase
23  Z=Vsc/Isc; // total impedance of transformer
24  Re=Pscp/Isc^2; // Total resistance of transformer
        refrred to high voltage side
25  Xe=sqrt(Z^2-Re^2); // total leakage impedance of
        transformer  referred to h v side
26  printf('transformer resistance and leakage impedance
         referred to h v side are %f ohm and %f ohm\n',Re
        ,Xe);
27  n=(1-(pop+Pscp/2^2)/(P/6+pop+Pscp/2^2))*100; //
        efficiency at half load
28  printf('efficiency at half load is %f percent',n);
```

Scilab code Exa 1.14 p u value of leakage impedance exciting current referred to l v and h v sides

```
1  clc;
2  p=20000; // rated power of transformer
3  vbh=2500; // base voltage in volts for h. v. side
4  vbl=250; // base voltage in volts for l. v. side
5  ibh=p/vbh; // base current in Ampere for h. v. side
6  zbh=vbh/ibh; // base impedance in ohm
```

32

```
 7  ze=2.6+4.3*%i; // equivalent leakage impedance
        referred to h. v. side in ohm
 8  zepu=ze/zbh; // per unit value in ohm
 9  disp('Per unit value of equivalent leakage impedance
        referred to h. v. side is');
10  disp(zepu);
11  k=vbl/vbh; // turn's ratio
12  zel=ze*k^2; // equivalent leakage impedance
        referred to l. v. side in ohm
13  ibl=p/vbl; // base current in Ampere for l. v. side
14  zbl=vbl/ibl; // base impedance for l. v. side
15  zelpu=zel/zbl; // per unit value in ohm
16  disp('Per unit value of equivalent leakage impedance
        referred to l. v. side is');
17  disp(zelpu);
18  zepum=abs(zepu); // magnitude of per unit impedance
19  vhl=zepum*vbh; // total leakage impedace drop on h.
        v. side
20  vbl=zepum*vbl; // total leakage impedace drop on l.
        v. side
21  printf('Total leakage impedance drop on h. v. side
        and l. v. side are %f V and %f V respectively\n',
        vhl,vbl);
22  Ieh=4.8; // exciting current in Ampere
23  Iepu=Ieh/ibh; // p u value of exciting current
        referred to h. v. side
24  printf('Per unit value of exciting current referred
        to h. v. side is %f p.u. \n',Iepu);
25  Iel=Ieh/k; // exciting current referred to l. v.
        side
26  Ielpu=Iel/ibl; // p u value of exciting current
        referred to l. v. side
27  printf('Per unit value of exciting current referred
        to l. v. side is %f p.u. \n',Ielpu);
```

Scilab code Exa 1.15 determining p u values of circuit parameters referred to both sides

```scilab
1  clc;
2  P=10000; // rated power of transformer
3  vbh=2000; // base voltage for h v side in volts
4  ibh=P/vbh; // base current for h v side in Ampere
5  vbl=200; // base voltage for l v side in volts
6  ibl=P/vbl; // base current for l v side in Ampere
7  k=vbl/vbh; // turns ratio
8  r1=3.6; // resistance of h v side of transformer in
      ohm
9  x1=5.2; //leakage reactace h v side of transformer
      in ohm
10 z=vbh/ibh; // base impedance for h v side'
11 r1pu=r1/z; // p u value for resistance of h v side
      of transformer in ohm
12 x1pu=x1/z; // p u value for leakage reactance of h v
       side of transformer in ohm
13 r2=0.04; //resistance of l v side of transformer in
      ohm
14 x2=0.056; //leakage reactace l v side of transformer
       in ohm
15 // total resistance referred to  h v side
16 re=r1+r2/k^2;
17 repu=re/z;
18 // total leakage impedance referred to h v side
19 xe=x1+x2/k^2;
20 xepu=xe/z;
21 printf('total  per unit resistance and per unit
      leakage impedance referred to  h v side  are %f
      and %f\n',repu,xepu);
22 Z=vbl/ibl; // base impedance for l v side
23 Re=r2+r1*k^2; // total resistance referred to  l v
      side
24 Repu=Re/Z;
25 Xe=x2+x1*k^2; //total leakage impedance referred to
      l v side
```

34

```
26  Xepu=Xe/Z;
27  printf('total  per unit resistance and per unit
       leakage impedance referred to  l v side are %f
       and %f ',Repu,Xepu);
```

Scilab code Exa 1.16 Determining parameters of equivalent circuit diagram

```
1  clc;
2  P=200000; //rated power of transformer
3  E1=4000; // primary side rated voltage
4  E2=1000; // secondary side rated voltage
5  n=0.97; // efficiency
6  pfn=0.25; // power factor at no load
7  pff=0.8; // power factor at full load
8  vr=5; // percentage voltage regulation
9  Pl=((1/n)-1)*200000; // total losses at full load
10 Pf=Pl*0.6; // total losses at 60% of full load
11 Po=(Pl-Pf)/(1-0.36); // ohmic losses
12 Pc=Pl-Po; // core losses
13 re2=(Po/P)*100; // P U total resistance referred to
       l. v. side
14 xe2=(vr-re2*pff)/sqrt(1-pff^2); // P U total leakage
       reactance referred to l. v. side
15 re2=(re2*E2^2)/(100*P); // total resistance in ohms
16 disp('Total resistance referred to l. v. side is ');
17 printf('%f ohm',re2);
18 xe2=(xe2*E2^2)/(100*P); // total leakage reactance
       in ohms
19 disp('Total leakage reactance referred to l. v. side
       is ');
20 printf('%f ohm',xe2);
21 Rc=E2^2/Pc;
22 disp('Coreloss resistance is');
23 printf('%f ohm',Rc);
```

```
24 Ie2=Pc/(E2*pfn); // exciting current in Ampere
25 Ic=Pc/E2; // core loss current
26 Im=sqrt(Ie2^2-Ic^2); // magnetizing component of
       exciting current
27 Xm=E2/Im; // magnetizing reactance
28 disp('Magnetizing reactance is ');
29 printf('%f ohm',Xm);
30 disp('All parameters are known. So, equivalent
       circuit diagram referred to l. v. side can be
       drawn.');
```

Scilab code Exa 1.18 Determining voltage regulation and terminal voltage at different power factor

```
 1 clc;
 2 P=20000; // rated power of transformer
 3 E1=2500; // primary side voltage
 4 E2=500; // secondary side voltage
 5 r1=8; // primary resistance in ohm
 6 x1=17; // primary leakage reactance in ohm
 7 r2=0.3; // secondary resistance in ohm
 8 x2=0.7; // secondary leakage reactane in ohm
 9 k=E2/E1; // turns ratio
10 re2=r2+r1*k^2; // equivalent resistance referred to
       secondary winding
11 xe2=x2+x1*k^2; // equivalent leakage reactance
       referred to secondary winding
12 Il=P/E2; // full load secondary current
13 disp('case a');
14 pf=0.8; // lagging power factor
15 vd=Il*(re2*pf+xe2*sqrt(1-pf^2)); // Voltage drop in
       impedance in volts
16 vt=E2-vd; // secondary terminal voltage
17 printf('secondary terminal voltage for a lagging
       power factor is %f v\n',vt);
```

36

```
18 vr=((E2-vt)/E2)*100; // voltage regulation
19 printf('voltage regulation for a lagging power
      factor is %f percent\n',vr);
20 disp('case b');
21 pf=0.8; // leading power factor
22 vd=Il*(re2*pf-xe2*sqrt(1-pf^2)); // Voltage drop in
      impedance in volts
23 vt=E2-vd; // secondary terminal voltage
24 printf('secondary terminal voltage for a leading
      power factor is %f v\n',vt);
25 vr=((E2-vt)/E2)*100; // voltage regulation
26 printf('voltage regulation for a leading power
      factor is %f percent\n',vr);
```

Scilab code Exa 1.19 Determining voltage regulation and secondary terminal voltage

```
1 clc;
2 rpu=0.02; // P U equivalent resistance
3 xpu=0.05; // P U equivalent leakage reactance
4 E2=440; // Secondary full load voltage
5 pf=0.8; // lagging power factor
6 vr=rpu*pf+xpu*sqrt(1-pf^2); // P U voltage
      regulation
7 printf('Full load p.u. voltage regulation is %f or
      %f percent\n',vr,vr*100);
8 dv=E2*vr; // change in terminal voltage
9 V2=E2-dv; // secondary terminal voltage
10 printf('Secondary terminal voltage is %f V',V2);
```

Scilab code Exa 1.20 Determining voltage applied to high voltage side

```
1 clc;
```

```
2  P=10000; // rated power of transformer in VA
3  E1=2000; // full load primary voltage
4  E2=400; // full load secondary voltage
5  k=E2/E1; // turns ratio
6  pf=0.8; // lagging power factor
7  // initialising results of short circuit test
8  v=60; // voltage applied for short circuit test
9  i=4; // short circuit current
10 p=100; // power dissipated in short circuit;
11 reh=p/i^2; // total resistance referred to h v side
12 zeh=v/i; // total impedance referred to h v side
13 xeh=sqrt(zeh^2-reh^2); // total leakage reactance
      referred to h v side
14 rel=reh*k^2; // resistance referred to l v side
15 xel=xeh*k^2; // reactance referred to l v side
16 i2l=P/E2; // full load secondary current
17 vr=i2l*(rel*pf+xel*sqrt(1-pf^2)); // voltage
      regulation
18 v2=E2+vr; // total voltage of secondary when
      transformer is operating on full load
19 v1=v2/k; // voltage applied to primary to deliver
      full load
20 printf('voltage applied to primary to deliver full
      load is %f v',v1);
```

Scilab code Exa 1.21 voltage at sending end of feeder primary terminals
active and reactive power power factor

```
1  clc;
2  zf=30+120*%i; // feeder impedance
3  E1=33000; // primary side voltage
4  E2=3300; // secondary side voltage
5  k=E2/E1; // turns ratio
6  P=100000; // load power
7  pf=0.8;// leading power factor of load
```

38

```
 8  zl =0.3+1.4*%i; // leakage impedance referred to l v
        side
 9  zfl=zf*k^2; // feeder impedance referred to l v side
10  vt =3300; // terminal voltage
11  il=P/(vt*pf); // load current
12  R=real(zfl)+real(zl); // total resistance referred
        to l v side
13  X=imag(zfl)+imag(zl); // total impedance referred to
         l v side
14  vfl=vt+il*(R*pf-X*sqrt(1-pf^2)); // voltage at the
        sending end of feeder referred to l v side
15  vf=vfl/k; // voltage at the sending end of feeder
16  printf('Voltage at the sending end of feeder is %f v
        \n',vf);
17  v2=vt+il*(real(zl)*pf-imag(zl)*sqrt(1-pf^2)); //
        voltage induced in secondary windings
18  v1=round(v2/k);
19  printf('voltage at the primary terminals of
        transformer is %f v\n',v1);
20  ap=il^2*R;
21  printf('active power loss is %f W\n',ap);
22  ar=il^2*X;
23  printf('reactive power loss is %f W\n',ar);
24  cp=P-P*%i*tan(acosd(pf)*(%pi/180)); // complex power
         at load end in VA
25  cps=cp+((ap+ar*%i) ); // complex power at feeder end
        in VA
26  pfs=cos(atan(imag(cps),real(cps)));
27  printf('power factor at the sending end is %f
        leading ',pfs);
```

Scilab code Exa 1.22 input power and power factor

```
 1  clc;
 2  P=10000; // rated power of transformer
```

39

```scilab
 3 E1=2000; // primary side voltage
 4 E2=200; // secondary side voltage
 5 f=50; // frequency in hertz
 6 po=125; // no load power
 7 pfo=0.15; // no load power factor
 8 zbh=E1^2/P; // base impedance on h v side
 9 k=E2/E1; // turns ratio
10 zl=0.5+1*%i; // percent leakage impedance
11 zlh=zl*(zbh*k^2); // percent leakage impedance
      referred to h v side
12 Rc=E1^2/po; // coreloss resistance
13 Io=po/(E1*pfo); // No load current
14 Xm=E1/(Io*sqrt(1-pfo^2)); // magnetizing reactance
15 p=10000; // load power
16 pf=0.8; // power factor of load
17 il=p/(E2*pf); // secondary load current
18 ilp=il*k; // primary load current
19 vp=E1+ilp*(real(zlh)*pf+imag(zlh)*sqrt(1-pf^2));
20 ap=ilp^2*real(zlh); // active power loss in series
      resistance
21 ar=ilp^2*imag(zlh); // reactive power loss in series
       reactance
22 Ap=vp^2/Rc; // active power loss in coreloss
      resistance
23 Ar=vp^2/Xm; // reactive power loss in magnetizing
      reactance
24 cpl=p*(1+%i*tan(acos(0.8))); // complex power at
      load end in VA
25 cpi=(real(cpl)+ap+Ap)+%i*(imag(cpl)+ar+Ar); //
      complex power input to transformer VA
26 printf('real power input to transformer is %f W\n',
      real(cpi));
27 ipf=cos(atan(imag(cpi),real(cpi)));
28 printf('input power factor is %f lagging',ipf);
```

Scilab code Exa 1.24 hysteresis and eddy current losses

```
1  clc;
2  pc1=52; // core loss at f=40
3  f1=40; // frequency in hertz
4  pc2=90; // core loss at f=60
5  f2=60; // frequency in hertz
6  f=[f1 f1^2;f2 f2^2];
7  pc=[pc1;pc2];
8  k=inv(f)*pc;
9  // proportionality constants for hysteresis and eddy
       current losses are
10 kh=k(1);disp(kh) // proportionality constants for
      hysteresis losses
11 ke=k(2);disp(ke) //  proportionality constants for
      eddy current losses
12 // determining both losses at 50 hertz
13 f=50;
14 ph=kh*f;
15 printf('hysteresis losses at 50 hertz is %f W\n',ph)
      ;
16 pe=ke*f^2;
17 printf('eddy current losses at 50 hertz is %f W',pe)
      ;
18 // answer for eddy current losses is misprinted in
      book
```

Scilab code Exa 1.25 total core loss

```
1
2  clc;
3  // subscripts 1 and 2 are used the quantities
      referred to 60 hz and 50 hz frequency
      respectively
4  v1=220; // rated voltage at 60 hz
```

41

www.mepcafe.com

```
5  f1 =60; // operating frequency
6  ph1 =340; // hysteresis loss at 60 hz
7  pe1 =120; // eddy current loss at 60 hz
8  v2 =230; // rated voltage at 50 hz
9  f2 =50; // operating frequency
10 s =1.6; // Steinmetz's constant
11 B =(f1/f2)*(v2/v1); // ratio of flux densities Bm2/
      Bm1
12 ph2=ceil(ph1*(50/60)*B^s); // hysteresis loss at 50
      hz
13 pe2=pe1*(f2/f1)^2*(B)^2;// eddy current loss at 50
      hz
14 pc=ph2+pe2;
15 printf('Total core loss at 50 hz is %f W',pc);
```

Scilab code Exa 1.26 determining losses and output at 60 hz

```
1  clc;
2  // subscripts 1 and 2 are used to refer 50 hz and 60
       hz quantity respectively
3  // voltage and current is same for both the cases
4  s =1.6; // Steinmetz's coefficient
5  poh1 =1.6; // percentage ohmic losses
6  ph1 =0.9; // percentage hysteresis losses
7  pe1 =0.6; // percentage eddy current losses
8  f1 =50; // frequency in hertz
9  f2 =60; // frequency in hertz
10 B=f1/f2 // since voltage level are same for both
      cases ratio of flux densities i.e Bm2/Bm1=f1/f2
11 ph2=ph1*(f2/f1)*B^s; // percentage hysteresis losses
12 pe2=pe1*(f2/f1)^2*B^2; // percentage eddy current
      losses
13 poh2=poh1; // since the voltage and current levels
      are same therefore ohmic losses are same
14 // for the total losses to be remain same at both
```

42

www.mepcafe.com

```
        the frequencies only ohmic losses can be  varied
15  p=poh1+ph1+pe1; // total losses at 50 hz
16  pc=ph2+pe2; // total core losses at 60 hz
17  pnoh=p-pc; // permissible value for new ohmic losses
18  x=sqrt(pnoh/poh1); // factor by which output at 50
        hz should be multiplied to get the same output at
         60 hz
19  printf('ohmic losses at 60 hz is %f percent\n',poh2)
        ;
20  printf('hysteresis losses at 60 hz is %f percent\n',
        ph2);
21  printf('eddy current losses at 60 hz is %f percent\n
        ',pe2);
22  printf('factor by which output at 50 hz should be
        multiplied to get the same output at 60 hz is %f
        ',x);
```

Scilab code Exa 1.27 finding efficiency and losses

```
1  clc;
2  // subscripts 1 and 2 are used to indicate
       transformer of 11kv at 25hz and 22kv at 50 hz
       respectively
3  // for same current power is doubled therefore P2=2
       P1
4  poh1=1.8; // ohmic losses as a percentage of total
       power P1
5  ph1=0.8; // hysteresis losses as a percentage of
       total power P1
6  pe1=0.3; // eddy current losses as a percentage of
       total power P1
7  poh2=poh1/2; // ohmic losses do not change with
       frequency but changes with voltage since p1=2p1
       we get the result shown
8  // since frequency also gets doubled whwn voltage
```

43

```
       levels double therefore there is no change in
       flux density i.e is Bm1=Bm2
 9 f1=25; // frequency in hertz
10 f2=50; // frequency in hertz
11 ph2=(f2/f1)*ph1; // hysteresis losses are directly
       proportional to frequency
12 pe2=(f2/f1)^2*pe1; // eddy current losses are
       directly proportional to frequency
13 // we know p2=2p1
14 ph2p=ph2/2; //  hysteresis losses as a percentage of
        total power P2
15 pe2p=pe2/2; // eddy current losses as a percentage
       of total power P2
16 printf('ohmic losses as a percentage of total power
       at 50 hz is %f percent\n',poh2);
17 printf('hysteresis losses as a percentage of total
       power at 50 hz is %f percent\n',ph2p);
18 printf('eddy current losses as a percentage of total
        power at 50 hz is %f percent\n',pe2p);
19 // efficiency at f1,v1
20 n1=(1-((poh1+ph1+pe1)/100)/(1+((poh1+ph1+pe1)/100)))
       *100;
21 printf('efficiency at 25 hz is %f percent\n',n1);
22 // efficiency at f2,v2
23 n2=(1-((poh2+ph2p+pe2p)/100)/(1+((poh2+ph2p+pe2p)
       /100)))*100;
24 printf('efficiency at 50 hz is %f percent',n2);
```

Scilab code Exa 1.28 determining efficiency load voltage regulation and secondary terminal voltage

```
1 clc;
2 P=10000; // rated power of transformer in VA
3 E1=2500; // primary side voltage
4 E2=250; // secondary side voltage
```

44

www.mepcafe.com

```
 5  pf=0.8; // power factor
 6  //initialising the results of open circuit test
 7  vo=250; // open circuit voltage
 8  io=0.8; //no load current
 9  po=50; // open circuit voltage
10  // initialising the results of open circuit test
11  vsc=60; // short circuit voltage
12  isc=3; // short circuit current
13  psc=45; // power dissipated in test
14  ifl=P/E1; // full load current on primary side
15  poh=psc*(ifl/isc)^2; // ohmic losses at full load
        current
16  disp('case a(1)');
17  n=(1-(po+(poh/4^2))/(po+(poh/4^2)+(P*pf)/4))*100; //
        efficiency at 1/4 load
18  printf('efficiency at 1/4 load is %f percent\n',n);
19  disp('case a(2)');
20  n=(1-(po+(poh/2^2))/(po+(poh/2^2)+(P*pf)/2))*100; //
        efficiency at 1/2 load
21  printf('efficiency at 1/2 load is %f percent\n',n);
22  disp('case a(3)');
23  n=(1-(po+(poh/1^2))/(po+(poh/1^2)+(P*pf)/1))*100; //
        efficiency at full load
24  printf('efficiency at full load is %f percent\n',n);
25  disp('case a(4)');
26  n=(1-(po+((poh*5^2)/4^2))/(po+((poh*5^2)/4^2)+(P*pf
        *5)/4))*100; // efficiency at 1*1/4 load
27  printf('efficiency at 5/4 load is %f percent\n',n);
28  // let maximum efficiency occurs at x times the
        rated KVA
29  // maximum efficiency occurs when core loss becomes
        equal to ohmic losses
30  x=sqrt(po/poh);
31  nm=(x*P)/1000; // VA output at maximum
32  nmax=(1-(2*po)/(nm*1000*pf+2*po))*100;
33  printf('KVA load at which maximum efficiency occurs
        is %f KVA\n',nm);
34  printf('Maximum efficiency is %f percent\n',nmax);
```

45

```
35  // from  short  circuit  test
36  reh=psc/isc^2;  // total  resistance  referred  to  h  v
        side
37  zeh=vsc/isc;  // total  impedance  referred  to  h  v  side
38  xeh=sqrt(zeh^2-reh^2);  // total  leakage  reactance
        referred  to  h  v  side
39  er=(ifl*reh)/E1;  //p  u  resistance
40  ex=(ifl*xeh)/E1;  // p  u  reactance
41  vr=(er*pf+ex*sqrt(1-pf^2))*100;  // p  u  voltage
        regulation
42  printf(' p  u  voltage  regulation  for  lagging  power
        factor  is  %f  percent\n',vr);
43  dv=E2*(vr/100);  // voltage  drop  in  series  impedance
44  v2=E2-dv;
45  printf('secondary  terminal  voltage  for  lagging  power
        factor  of  0.8  is  %f  v\n',v2);
46  // voltaage  regulation  for  leading  power  factor
47  vr=(er*pf-ex*sqrt(1-pf^2))*100;  // p  u  voltage
        regulation
48  printf(' p  u  voltage  regulation  for  leading  power
        factor  is  %f  percent\n',vr);
49  dv=E2*(vr/100);  // voltage  drop  in  series  impedance
50  v2=E2-dv;
51  printf('secondary  terminal  voltage  for  leading  power
        factor  of  0.8  is  %f  v\n',v2);
```

Scilab code Exa 1.29 Determining transformer core loss ohmic loss and pu value of equivalent resistance

```
1  clc;
2  p=20000;  // rated  capacity  of  transformer
3  n=0.98;  // efficiency  of  transformer  at  full  load
        and  half  load
4  c=[ 1  1; 1  1/4];
5  o=[ ((1/n)-1)*p;  ((1/n)-1)*(p/2)];
```

46

```
6  l=inv(c)*o;
7  printf('Core losses are %f W\n',l(1));
8  printf('Ohmic losses are %f W\n',l(2));
9  re=l(2)/p;
10 printf('p.u. value of equivalent resistance is %f ',
       re);
```

Scilab code Exa 1.30 determining voltage regulation at given power factor

```
1  clc;
2  P=100000; // VA of transformer
3  nmax=0.98; // maximum efficiency of transformer
4  pf=0.8; // power factor at which maximum efficiency
       occurs
5  l=80; // percentage of full load at which maximum
       efficiency occurs
6  po=P*pf*(l/100); // output at maximum efficiency
7  pl=((1/nmax)-1)*po; // total losses
8  pc=pl/2; // core loss
9  poh=pc; // at maximum efficiency variable losses are
       equal to constant losses
10 pohl=poh*(100/l)^2; // ohmic losses at full load
11 z=0.05; // p u leakage impedance
12 r=pohl/P; // p u resistance
13 x=sqrt(z^2-r^2); // p u leakage reactance
14 vr=(r*pf+x*sqrt(1-pf^2))*100; // voltage regulation
15 printf('Voltage regulation at 0.8 p.f. lagging is %f
       percent ',vr);
```

Scilab code Exa 1.31 Determining efficiency on full load full load power factor and load power factor for zero voltage drop

47

```
1  clc;
2  vdr=2; // percentage full load voltage drop in
       resistance
3  vdx=4; // percentage full load voltage drop in
       leakage reactance
4  // full load ohmic losses are equal to 0.02*VA
       rating of transformer which is equal to iron
       losses
5  n=100/(1+(vdr/100)+(vdr/100));
6  printf('Efficiency on full load at unity p.f is %f
       percent\n',n);
7  // maximum voltage drop means voltage regulation is
       also maximum
8  pf=vdr/sqrt(vdr^2+vdx^2);
9  printf('Full load power factor at which voltage
       regulation is maximum is %f lagging\n',pf);
10 pf=vdx/sqrt(vdr^2+vdx^2);
11 printf(' load power factor at which voltage
       regulation is zero is %f leading',pf);
```

Scilab code Exa 1.32 Determine input power and power factor at a certain load

```
1  clc;
2  P=20000; // rated VA of transformer
3  E1=3300; // rated voltage of primary
4  E2=220; // rated voltage of secondary
5  v2=220; // voltage at which load is getting
       delivered
6  p=14960; // load power in Watts
7  pf=0.8; // power factor at on load
8  pc=160; // core loss
9  pfo=0.15; // power factor at no load
10 il=p/(v2*pf); // load current
11 is=P/E2; // rated current of secondary
```

48

www.mepcafe.com

```
12  vr =1 ; // percentage voltage drop of rated voltage
        in total resistance
13  vx =3 ; // percentage voltage drop of rated voltage
        in total leakage reactance
14  re2 =(E2*vr)/(is*100); // total resistance referred
        to secondary
15  xe2 =(E2*vx)/(is*100); // total leakage reactance
        referred to secondary
16  poh =il^2*re2; // ohmic losses
17  pi =poh+pc+p; // total input power
18  // E2 as a reference
19  i2 =il*(pf-%i*sqrt(1-pf^2));
20  E2n =v2+i2*(re2+%i*xe2); // secondary winding voltage
21  io =pc/(pfo*E2); // no load current
22  ic =pc/E2; // core loss current
23  im =sqrt(io^2-ic^2); // magnetizing current
24  I =i2+(ic-im*%i); // total input current , negative
        sign before im indicates that it lags behind E2
        by 90 degree
25  pfi =cos(atan(imag(I),real(I))); // input power
        factor
26  printf ('Total input power is %f W \n',pi);
27  printf ('Input power factor is %f lagging',pfi);
```

Scilab code Exa 1.33 determining load power factor at minimum secondary terminal voltage and minimum secondary terminal voltage

```
1  clc ;
2  P =500000; // VA rating of transformer
3  E2 =400; // rated secondary voltage
4  nmax =0.98; // maximum efficiency of transformer
5  l =80; // percentage of full load at which maximum
        efficiency occurs
6  ze2 =4.5; // percentage impedance
7  pt =((1/nmax)-1)*P*(l/100); // total losses
```

49

```
 8  pc=pt/2; // core loss = ohmic loss at maximum
        efficiency
 9  poh=pc; // ohmic loss
10  pohl=poh*(100/l)^2; // full load ohmic losses
11  re2=(pohl/P)*100; // percentage resistance
12  xe2=sqrt(ze2^2-re2^2); // percentage leakage
        reactance
13  pfl=re2/ze2; // load power factor
14  vr=re2*pfl+xe2*sqrt(1-pfl^2); // voltage regulation
15  dv=(E2*vr)/100; // change in terminal voltage
16  V2=E2-dv; // Secondary terminal voltage
17  printf('Load power factor at which secondary
        terminal voltage is minimum is %f\n',pfl);
18  printf('Secondary terminal voltage is %f v',V2);
19  // answer for total losses is given wrong in the
        book
```

Scilab code Exa 1.34 Determining full day efficiency

```
 1  clc;
 2  P=5000; // rated VA of transformer
 3  pc=40; // core loss , it remains fixed for whole day
 4  poh=100; // ohmic losses
 5  // data for duration 7 A.M to 1 P.M
 6  p1=3000; // power consumed
 7  pf1=0.6 // power factor of load
 8  pk1=p1/pf1; // VA load
 9  poh1=poh*(pk1/P)^2; // ohmic losses for given
        duration
10  // data for duration 1 P.M to 6 P.M
11  p2=2000; // power consumed
12  pf2=0.8 // power factor of load
13  pk2=p2/pf2; // VA load
14  poh2=poh*(pk2/P)^2; // ohmic losses for given
        duration
```

50

```
15  // data for duration 6 P.M to 1 A.M
16  p3=6000; // power consumed
17  pf3=0.9 // power factor of load
18  pk3=p3/pf3; // VA load
19  poh3=poh*(pk3/P)^2; // ohmic losses for given
        duration
20  // data for duration 1 A.M to 7 A.m =no load
21  poht=poh1*6+poh2*5+poh3*7; // energy lost in ohmic
        losses
22  pct=(pc*24); // daily energy lost as core loss
23  ptl=poht+pct; // total energy lost
24  po=p1*6+p2*5+p3*7; // output
25  n=(1-(ptl/(ptl+po)))*100;
26  printf('All day efficiency is %f percent',n);
```

Scilab code Exa 1.35 Determining hysteresis and eddy current losses at given frequencies

```
1  clc;
2
3  //V/f ratio is same for every case hence hysteresis
        losses and eddy current losses can be calculated
        separately
4  // data for column 1
5  vt1=214; // terminal voltage
6  f1=50; // frequency in hz
7  p1=100; // power input in Watts
8  vp1=vt1; // per phase voltage
9  pv1=p1/3; // per phase power
10 pc1=pv1/f1; // core loss per cycle
11 // data for column 2
12 vt2=171; // terminal voltage
13 f2=40; // frequency in hz
14 p2=72.5; // power input in Watts
15 vp2=vt2; // per phase voltage
```

```
16  pv2=p2/3; // per phase power
17  pc2=pv2/f2; // core loss per cycle
18  // data for column 3
19  vt3=128; // terminal voltage
20  f3=30; // frequency in hz
21  p3=50; // power input in Watts
22  vp3=vt3; // per phase voltage
23  pv3=p3/3; // per phase power
24  pc3=pv3/f3; // core loss per cycle
25  // data for column 4
26  vt4=85.6; // terminal voltage
27  f4=20; // frequency in hz
28  p4=30; // power input in Watts
29  vp4=vt4; // per phase voltage
30  pv4=p4/3; // per phase power
31  pc4=pv4/f4; // core loss per cycle
32  // Values of k1 and k2 have been obtained from graph
33  k1=0.39;
34  k2=(pc1-k1)/50;
35  F1=60; //frequency at which losses has to be
        calculated
36  ph1=k1*F1; //per phase hysteresis loss at 60 hz
37  pe1=k2*F1^2; // per phase eddy curent loss at 60 hz
38  pht=3*ph1; // total hysteresis loss
39  pet=3*pe1; // total eddy current loss
40  printf('Total hysteresis and eddy current losses at
        60 hz are %f W and %f W respectively\n',pht,pet);
41  F2=40; //frequency at which losses has to be
        calculated
42  ph2=k1*F2; //per phase hysteresis loss at 40 hz
43  pe2=k2*F2^2; // per phase eddy curent loss at 40 hz
44  pht=3*ph2; // total hysteresis loss
45  pet=3*pe2; // total eddy current loss
46  printf('Total hysteresis and eddy current losses at
        40 hz are %f W and %f W respectively',pht,pet);
```

Scilab code Exa 1.36 Determining reading of wattmeter connected on l v side

```scilab
1  clc;
2  E1=230; // primary rating of transformer 1 and
       transformer 2
3  E2=400; // secondary rating of transformer 1
4  e2=410; // secondary rating of transformer 2
5  iv=25; // current feeded by voltage regulator to h v
        series winding
6  pc=200; // core loss in each transformer
7  r=1 // assuming resistance of transformer to be 1
8  x=3*r // as per question leakage reactance is thrice
        of resistance
9  il1=(iv*E2)/E1; // primary current of transformer 1
10 il2=(iv*e2)/E1; // primary current of transformer 2
11 pf=r/sqrt(r^2+x^2); // power factor during short
       circuit
12 // As per the circuit diagram given in question, by
       Kirchoffs current law current through current
       coil of wattmeter W1 is given by
13 I=il2-il1;
14 // 2*core loss is the reading of wattmeter 2
15 W=E1*I*pf; // reading of wattmeter 1 connected on l
       v side
16 // in circuit diagram if terminal a is connected to
       c and terminal b is connected to d the current I
       and Io (no load current) flow in the same
       direction of current coil of Wattmeter.Hence its
       reading is increased to
17 Wt=2*pc+W;
18 printf('reading of wattmeter as per the connection
       described is %f W\n',Wt);
19 // in circuit diagram if terminal c is connected to
```

53

www.mepcafe.com

b and terminal d is connected to a the current I and Io (no load current) flow in the opposite direction through current coil of Wattmeter.Hence its reading is decreased to

```
20  Wt=2*pc-W;
21  printf('reading of wattmeter as per the connection
        described is %f W',Wt);
```

Scilab code Exa 1.37 Determine leakage impedance per phase in ohm and p u system

```
1  clc;
2  E1=3300; // rated phase voltage of primary of a
       three phase transformer
3  v=360; // voltage injected in open delta h v winding
        to circulate full load current
4  vph=v/3; // voltage across each phase
5  P=300; // rated KVA of transformer
6  Pph=P/3; // KVA per phase
7  Iph=(Pph*1000)/E1; // per phase current
8  z=vph/Iph;
9  printf('Per Phase leakage impedance is %f ohms\n',z)
       ;
10 zb=E1/Iph; // base impedance
11 zpu=z/zb;
12 printf('leakage impedance per phase in per unit
        system is %f p.u',zpu);
```

Scilab code Exa 1.38 Determine power transformed power conducted power output and auto transformer efficiency at same power factor

```
1  clc;
2  P=20000; // rated VA of transformer
```

```
3  E1=2300; // rated voltage of primary
4  E2=230; // rated voltage of secondary
5  pf=0.6; // power factor
6  n=0.96; // efficiency
7  ih=P/E1; // rated current of h v winding
8  il=P/E2; // rated current of l v winding
9  // As per the connections given in fig 14.1(a), two
       voltages are in series aiding
10 Et=E1+E2; // output voltage of autotransformer
11 disp('case a');
12 // By Kirchoffs law at point b , supply current is
       given by
13 I=il+ih;
14 Pa1=Et*il; // VA rating of autotransformer
15 Po1=(Pa1/1000); // power output at full load unity
       power factor
16 Pt1=(E2*il)/1000; // power transformed
17 Pc1=(Po1-Pt1); // power conducted
18 printf('For the given connection , output power is %f
       kW\n',Po1);
19 printf('For the given connection , transformed power
       is %f kW\n',Pt1);
20 printf('For the given connection , conducted power is
       %f kW\n',Pc1);
21 disp('case b');
22 // As per the connections given in fig 14.1(b), two
       voltages are in series opposition
23 Et=E1-E2; // output voltage of autotransformer
24 // By Kirchoffs law at point b , supply current is
       given by
25 I=il-ih;
26 Pa2=E1*I; // VA rating of autotransformer
27 Po2=Pa2/1000; // power output at full load unity
       power factor
28 Pt2=(E2*il)/1000; // power transformed
29 Pc2=(Po2-Pt2); // power conducted
30 printf('For the given connection , output power is %f
       kW\n',Po2);
```

```
31  printf('For the given connection, transformed power
        is %f kW\n',Pt2);
32  printf('For the given connection, conducted power is
         %f kW\n',Pc2);
33  pl=((1/n)-1)*P*pf; // losses in 2-winding
        transformer
34  // autotransformer operates at rated current and
        rated voltage so efficiency and losses remain
        constant
35  disp('Efficiency for case a');
36  n1=(1-(pl/(Po1*1000*pf+pl)))*100;
37  printf('Efficiency of autotransformer for %f VA is
        %f percent\n',Po1,n1);
38  disp('Efficiency for case b');
39  n2=(1-(pl/(Po2*1000*pf+pl)))*100;
40  printf('Efficiency of autotransformer for %f VA is
        %f percent',Po2,n2);
```

Scilab code Exa 1.39 Determine current in various parts of circuit

```
1  clc;
2  // connections have been made in fig 1.42 in book to
        suit voltage requirement of 3000V, 3500V and
        1000V.
3  E1=1000; // primary winding  of transformer
4  E2=2000; // secondary winding of transformer
5  E3=500; // tertiary winding of transformer
6  l1=1050; // load in KVA across 3500 V
7  l2=180; // load in KVA across 1000 V
8  i1=(l1*1000)/(E1+E2+E3); // current through load of
        1050 KVA
9  i2=(l2*1000)/(E1); // current through load of 180
        KVA
10  kt=l1+l2; // Total KVA load supplied
11  I=(kt*1000)/(E1+E2);
```

56

```
12 printf('current through %f KVA load is %f A\n',l1,i1
     );
13 printf('current through %f KVA load is %f A\n',l2,i2
     );
14 printf('current drawn from supply is %f A',I);
```

Scilab code Exa 1.40 Determine KVA output KVA transformed and KVA conducted of auto transformer

```
1 clc;
2 E1=2500; // primary side voltage
3 E2=250; // secondary side voltage
4 P=10000; // rated VA of transformer
5 // to achieve a voltage level of 2625, two equal
     parts of 125 V each of secondary winding are
     connected in parallel with each other and in
     series with primary winding
6 Eo=E1+E2/2; // desired output of autotransformer
7 il=P/E2; // rated current of l v winding
8 i=2*il; // Total output current
9 K=(i*Eo)/1000; // Auto transsformer KVA rating
10 ip=P/E1; // rated current of h v winding
11 I=i+ip; // current drawn from supply
12 Pt=(i*(E2/2))/1000; // KVA transformed
13 Pc=K-Pt; // KVA conducted
14 printf('KVA output of autotransformer is %f KVA\n',K
     );
15 printf('KVA transformed is %f KVA\n',Pt);
16 printf('KVA conducted is %f KVA',Pc);
```

Scilab code Exa 1.41 determining currents in various branch of autotransformer

```
 1 clc ;
 2 E1=440; // primary supply voltage
 3 E2=380; // voltage at which load at secondary
      terminal is being supplied
 4 l1=40000; // power rating of load in watts
 5 pf=0.8; // lagging power factor
 6 I2=l1/(sqrt(3)*E2*pf);
 7 // per phase KVA input=per phase KVA output
 8 I1=(E2/E1)*I2;
 9 In=I2-I1;
10 printf('Current in primary branch is %f A\n',I1);
11 printf('current in secondary branch is %f A\n',I2);
12 printf('current between neutral and tapping points
      is %f A',In);
```

Scilab code Exa 1.42 Finding primary current and primary input

```
 1 clc ;
 2 // From fig 1.45
 3 N1=1000; // no of turns on primary
 4 N2=400; // no. of turns on secondary
 5 n2=300; // no. of turns across points A and B
 6 l1=600; // a load of 600 KW connected between points
      A and C
 7 l2=60+60*%i; // load connected between points A and
      B
 8 E=30000; // primary supply voltage
 9 vac=E*(N2/N1); // secondary side voltage
10 I1=(l1*1000)/vac; // current through load of 600 KW
11 vab=(vac/N2)*n2; // volatge across pints A and B
12 I2=vab/l2 ; // load current through load of 60+60i
13 iba=I1+I2; // current through section Ab of winding
14 mfs=iba*n2+I1*(N2-n2); // seconadry mmf
15 ip=mfs/N1;
16 printf('primary current is %f%fi A\n',real(ip),imag(
```

```
       ip));
17 Pi=(E*abs(ip)*cos(atan(imag(ip),real(ip))))/1000;
18 printf('primary power input is %f KW\n',Pi);
19 pf=cos(atan(imag(ip),real(ip)))
20 printf('power factor at primary terminal is %f
       lagging',pf)
```

Scilab code Exa 1.43 Finding current in three section of autotransformer

```
1 clc;
2 E=400; // supply voltage
3 l1=200; // load connected across 75% tapping
4 l2=400; // load connected between 25% and 100%
       tapping
5 t1=25; // 25% tapping point
6 t2=50; // 50% tapping point
7 t3=75; // 75% tapping point
8 V2=(t3/100)*E; // voltage across 200 ohm load
9 I2=V2/l1; // current through 200 ohm load
10 I1=(V2*I2)/E;
11 // from fig.(1.46 b), KCL at point d gives
12 idb=I2-I1;
13 // same secondary voltage is applied against load of
        400 ohm
14 I2=V2/l2; // current through 400 ohm load
15 I1=(V2*I2')/E;
16 // from fig (1.46 c), KCL at point c gives
17 ica=I2-I1;
18 // superimposing the currents of above results
       current in three portion of winding can be known
19 icd=ica;
20 disp('current in section cd of winding is ')
21 printf('%f A\n',icd);
22 ibc=I1;
23 disp('current in section bc of winding is ')
```

```
24 printf('%f A\n',ibc);
25 iab=idb+I1;
26 disp('current in section ab of winding is ')
27 printf('%f A\n',iab);
```

Scilab code Exa 1.44 voltage and current rating KVA rating efficiency percentage impedance regulation and short circuit current on each side

```
1 clc;
2 P=100000; // VA rating of two winding transformer
3 E1=2000; // rated voltage of h v side
4 E2=200; // rated voltage of l v side
5 l=2.5; // percentage of loss in two winding
      transformer
6 vr=3; // percentage of voltage regulation in two
      winding transformer
7 z=4; // percentage of leakage impedance in two
      winding transformer
8 ih=P/E1; // full load current of h v side
9 il=P/E2; // full load current of l v side
10 V1=E1; // rated voltage on l v side of
      autotransformer
11 V2=E1+E2; // rated voltage on h v side of
      autotransformer
12 Il=il+ih; // rated current on l v side of
      autotransformer
13 printf('Rated voltage on l v and h v side of
      autotransformer are %f v and %f v respectively\n,
      ',V1,V2);
14 printf('Rated current on h v and l v side of
      autotransformer are %f A and %f A respectively\n,
      ',il,Il);
15 k=E1/V2; // turns ratio for auto transformer
16 K=((1/(1-k))*P)/1000;
17 printf('Rated KVA of autotransformer is %f KVA\n',K)
```

60

www.mepcafe.com

```
   ;
18 pl=(1-k)*l; //percent full load losses in
      autotransformer
19 n=100-pl;
20 printf('Efficiency of auto transformer is %f percent
      \n',n);
21 Z=(1-k)*z;
22 printf('Percentage impedance as an auto transformer
      is %f \n',Z);
23 VR=(1-k)*vr;
24 printf('percentage voltage regulation as an auto
      transformer is %f \n',VR);
25 Is=(1/(1-k))*(100/z); // short circuit p u current
26 Ish=(Is*il)/1000;
27 printf('Short circuit of auto transformer on h v
      side is %f KA \n',Ish);
28 Isl=(Is*Il)/1000;
29 printf('Short circuit of auto transformer on l v
      side is %f KA \n',Isl);
```

Scilab code Exa 1.45 Determining load voltage at given power factors

```
1 clc;
2 v1=10; // voltage applied to primary when secondary
      is short circuited
3 ip=60; // primary current when secondary is short
      circuited
4 k=0.8; // turns ratio
5 E1=250; // input voltage for load voltage has to be
      calculated
6 E2=200; // rated voltage of secondary
7 il=100; // load current
8 pfo=0.24; // power factor during short circuit test
9 f=(1-k)^2/k^2; // factor by which secondary
      impedance has to be multiplied for referring it
```

```
        to  primary
10  // ze1=z1+f*z2  therefore  by  ohm  s  law
11  ze1=v1/ip;  // total  impedance  referred  to  primary
12  re1=ze1*pfo;  // total  resistance  referred  to  primary
13  xe1=ze1*sqrt(1-pfo^2);  // total  leakage  reactance
        referred  to  primary
14  disp('case  a');
15  pf=0.8;  //  lagging  power  factor  of  load
16  Ip=(E2*il)/E1;  // current  in  primary  due  to  load
        current
17  v2=(E1-Ip*(re1*pf+xe1*sqrt(1-pf^2)))*k;
18  printf('Secondary  terminal  voltage  at  %f  lagging
        power  factor  is  %f  v\n',pf,v2);
19  disp('case  b')
20  pf=1;  // unity  power  factor
21  v2=(E1-Ip*(re1*pf+xe1*sqrt(1-pf^2)))*k;
22  printf('Secondary  terminal  voltage  at  unity  power
        factor  is  %f  v',v2);
```

Scilab code Exa 1.46 Finding ratio of full load KVA delivered to sum of individual KVA ratings

```
1  clc;
2  r1=9 ;  // ratio  of  reactance  to  resistance  for
        transformer  1
3  r2=3 ;  // ratio  of  reactance  to  resistance  for
        transformer  2
4  d=atand(r1)-atand(r2);  // differene  between  angles
        of  transformer's  leakage  impedance
5  // leakage  impedance  of  both  transformers  are  equal
        z1=z2,  threefore  currents  in  both  transformers
        are  equal  that  is  i1=i2;
6  I=1/cos((d/2)*(%pi/180));  // ratio  of  numerical  sum
        of  i1  and  i2  to  phasor  sum  of  i1  and  i2
7  k=cos((d/2)*(%pi/180));
```

62

```
8  printf('ratio of full load KVA delivered to sum of
       both transformers KVA ratings is %f',k);
```

Scilab code Exa 1.48 Determining greatest load that can be put across parallel combination of transformers and secondary terminal voltage

```
1  clc;
2  P=400000; // rated KVA of transformer
3  P1=11000; // rated primary voltage
4  S2=6600; // rated secondary voltage
5  v1=360; // voltage recorded during short circuit of
       l v winding for first transformer
6  p1=3025; // power dissipated during short circuit of
        l v winding for first transformer
7  v2=400; // voltage recorded during short circuit of
       l v winding for second transformer
8  p2=3200; // power dissipated during short circuit of
        l v winding for second transformer
9  v3=480; // voltage recorded during short circuit
       test of l v winding third transformer
10 p3=3250; // power dissipated during short circuit of
        l v winding for third transformer
11 l1=(P+(v1/v2)*P+(v1/v3)*P)/1000;
12 printf('The greatest load that can be put on the
       transformers is %f KVA\n',l1);
13 is=P/S2; // secondary rated current
14 // transformer 1 is fully loaded , its carries full
       load current
15 re2=p1/is^2; // total resistance referred to
       secondary side
16 vd=is*re2; // voltage drop for transformer 1
17 E2=S2-vd;
18 printf('Secondary terminal voltage is %f v',E2);
```

63

Scilab code Exa 1.49 1 How transformers share a given load 2 Determine the greatest load that can be supplied by parallel combination of transformers and 3 Determining magnitude of the equivalent leakage impedance under normal loading

```scilab
1  clc ;
2  disp ( ' case  b ' ) ;
3  // KVA  ratings  and  leakage  impedances  for  the
        transformers  are
4  k1 =100;  // KVA  rating  for  transformer  1;
5  z1 =0.02;  // p  u  impedance  for  transformer  1;
6  k2 =75;  // KVA  rating  for  transformer  2;
7  z2 =0.03;  // p  u  impedance  for  transformer  2;
8  k3 =50;  // KVA  rating  for  transformer  3;
9  z3 =0.025;  // p  u  impedance  for  transformer  3;
10 disp ( ' case  b (1) ' ) ;
11 // assumng  k1  as  a  base  KVA
12 S =225;  // load  which  has  to  be  shared  by  three
        transformers
13 ze1 = z1 *100;  // percentage  impedance  for  transformer
        1
14 ze2 =( k1 / k2 ) * z2 *100;  // percentage  impedance  for
        transformer  2
15 ze3 =( k1 / k3 ) * z3 *100;  // percentage  impedance  for
        transformer  3
16 zt =(1/ ze1 ) +(1/ ze2 ) +1/( ze3 ) ;  // total  percentage
        leakage  impedance
17 s1 = S /( ze1 * zt ) ;
18 s2 = S /( ze2 * zt ) ;
19 s3 = S /( ze3 * zt ) ;
20 printf ( ' load  shared  by  transformer  1 ,2  and  3  are  %f
        KVA,  %f  KVA  and  %f  KVA  respectively \n ' , s1 , s2 , s3 ) ;
21 disp ( ' case  b (2) ' ) ;
22 // since  transformer  1  has  lowest  leakage  impedance
```

64

```
         among  three ,  it  will  be  loaded  to  its  rated
            capacity
23  S=k1*ze1*zt ; // total KVA shared
24  printf ('greatest load that can be shared by
            transformers is %f KVA\n',S);
25  disp('case b(3)');
26  // for successful parallel operation of transformer
            all the three leakage impedances based on their
            KVA rating  should be equal.Since magnitude of
            leakage impedance of transformer1 is fixed that
            is 2 percent z2=z3=2 percent
27  ze1 =2;
28  ze2=ze1*(k1/k2);
29  ze3=ze1*(k1/k3);
30  zt=(1/ze1)+(1/ze2)+(1/ze3); // Total leakage
            impedance
31  printf ('magnitude of equivalent leakage impedance is
             %f percent\n',zt);
```

Scilab code Exa 1.50 1 How transformers share a given load 2 Determine the secondary terminal voltage for part1

```
1  clc ;
2  //  shorts circuits test on two transformers gave
        the following results
3  P1 =200000; // KVA of transformer 1
4  V1 =3; // percentage rated voltage
5  pf1 =0.25; // lagging power factor for Xmer1
6  P2 =500000; // KVA of transformer 2
7  V2 =4; // percentage rated voltage
8  pf2 =0.3 // lagging power factor for Xmer2
9  l=560000; // load connected across parallel
        combination of transformers in KW
10  pf =0.8; // power factor of load
11  E1 =11000; // Rated primary voltage
```

65

```
12  E2=400; // Rated secondary voltage
13  ib=1; // base current
14  vb=1; // base voltage
15  z1=(V1/100)*1; // base impedance
16  Ze1=z1*(pf1+%i*sqrt(1-pf1^2)); // p u impedance
17  z2=(V2/100)*1; // base imedance
18  Ze2=z2*(pf2+%i*sqrt(1-pf2^2)); // p u impedance
19  pb=P2; // base for p u conversion
20  ze1=(pb/P1)*Ze1;
21  ze2=(pb/P2)*Ze2;
22  zt=ze1+ze2; // total impedance
23  s=l/pf; // KVA rating of transformer
24  S=s*(pf-%i*sqrt(1-pf^2)); // complex form of KVA
        rating
25  s1=(S*ze2)/(zt); // KVA shared by first transformer
26  PF1=cos(atand(imag(s1),real(s1))*(%pi/180));
27  s1w=round((abs(s1)*PF1)/1000);
28  printf('KW load shared by transformer 1 is %f at %f
        power factor lagging\n',s1w,PF1);
29  s2=(S*ze1)/(zt); // KVA shared by first transformer
30  PF2=cos(atand(imag(s2),real(s2))*(%pi/180));
31  s2w=ceil((abs(s2)*PF2)/1000);
32  printf('KW load shared by transformer 2 is %f at %f
        power factor lagging\n',s2w,PF2);
33  i1=abs(s1)/P1; // p u current shared by transformer
        1
34  i2=abs(s2)/P2; // p u current shared by transformer
        2
35  vr=i1*(real(Ze1)*PF1+imag(Ze1)*sqrt(1-PF1^2)); //
        voltag regulation
36  dv=E2*vr; // change in terminal voltage
37  Vt=E2-dv; // terminal voltage
38  printf('Secondary terminal voltage is %f v',Vt);
```

Scilab code Exa 1.51 Determining largest KVA that can be shared by parallel combination of transformer without overloading

```
1  clc;
2  k1=1000; // Rated KVA of transformer1
3  k2=500; // Rated KVA of transformer2
4  ze1=0.02+%i*0.06; // p u leakage impedance of
       transformer 1
5  ze2=0.025+%i*0.08; // p u leakage impedance of
       transformer 2
6  zb=1000; // base impedance
7  Z1=(zb/k1)*ze1; // impedance of transformer 1
8  Z2=(zb/k2)*ze2; // impedance of transformer 2
9  zt=Z1+Z2; // total impedance
10 S=k1*(abs(zt)/abs(Z2)); // since ze1<ze2 transformer
       1 reaches its rated KVA threrfore load shared by
        transformer 2 is given by
11 l2=S-k1; // load shared by transformer 2
12 printf('Largest KVA load that can be loaded on
       parallel combination of transformer is given by
       %f KVA',S);
```

Scilab code Exa 1.52 Determining no load circulating current and ohmic loss caused by it and no load terminal voltage

```
1  clc;
2  // two transformers are connected in parallel and
       has following data
3  P1=100; // rated KVA of transformer 1
4  E11=6600; // rated primary voltage for transformer 1
5  E21=230; // rated secondary voltage for transformer
       1
6  z1=1.5+4*%i // percentage leakage impedance for
       transformer 1
7  P2=200; // rated KVA of transformer 2
```

```
8  E12=6600; // rated primary voltage for transformer 2
9  E22=220; // rated secondary voltage for transformer
       2
10 z2=1+5*%i // percentage leakage impedance for
       transformer 2
11 i1=(P1*1000)/E21; // full load current for
       transformer 1
12 ze1=(z1/100)*(E21^2/(P1*1000)); // leakage impedance
        for transformer 1 in ohm
13 i2=(P2*1000)/E22; // full load current for
       transformer 2
14 ze2=(z2/100)*(E22^2/(P2*1000)); // leakage impedance
        for transformer 2 in ohm
15 io=(E21-E22)/abs(ze1+ze2); // circulating current at
        no load
16 printf('No load circulating current is %f A\n',real(
       io));
17 poh=abs(io)^2*(real(ze1)+real(ze2));
18 printf(' Ohmic losses due to circulting current is
       %f W\n',poh);
19 vd=abs(io)*abs(ze1); // voltage drop in leakage
       impedance
20 vt=E21-vd; // secondary terminal voltage
21 printf('secondary terminal voltage is %f v',vt);
```

Scilab code Exa 1.53 Finding no load circulating current current supplied by each transformer and KVA KW and power factor of each transformer

```
1 clc;
2 k1=500000; // rated VA of transformer 1
3 r1=0.015; // p u resistance of transformer 1
4 x1=0.05; // p u reactance of transformer 1
5 s1=405; // secondary no load voltage for transformer
       1
6 k2=250000; // rated VA of transformer 2
```

68

```
 7  r2=0.01; // p u resistance of transformer 2
 8  x2=0.05; // p u reactance of transformer 2
 9  s2=415; // secondary no load voltage for transformer
         2
10  l=750000; // KVA rating of load
11  pf=0.8; // power factor of load
12  v=400; // voltage at which load is being delivered
13  z1=(r1+%i*x1)*(v^2/k1); // impedance for transformer
         1 in ohms
14  z2= (r2+%i*x2)*(v^2/k2); // impedance for
         transformer 2 in ohms
15  il=l/v; // load current
16  zl=v/il; // load impedance
17  zl=zl*(pf+%i*sqrt(1-pf^2)); // complex form of load
         impedance
18  zt=z1+z2; // equivalent impedance of both
         transformer
19  io=(s2-s1)/abs(zt); // circulating current at no
         load
20  aio=cos(atand(imag(zt),real(zt))*(%pi/180)); //
         power factor
21  printf('Circulating current at no load is %f A at a
         power factor of %f lag\n',io,aio);
22  Ia=((s1*z2)+(s1-s2)*zl)/((z1*z2)+(zl*zt));
23  ia=abs(Ia);
24  printf('Current shared by transformer 1 is %f A\n',
         ia);
25  Ib=((s2*z1)-(s1-s2)*zl)/((z1*z2)+(zl*zt));
26  ib=abs(Ib);
27  printf('Current shared by transformer 2 is %f A\n',
         ib);
28  kv1=(ia*v)/1000;
29  pf1=cos(atand(imag(Ia),real(Ia))*(%pi/180));
30  kw1=kv1*pf1;
31  printf('KVA shared by transformer 1 is %f KVA at %f
         lagging power factor\n',kv1,pf1);
32  printf('KW shared by transformer 1 is %f KW\n',kw1);
33  kv2=(ib*v)/1000;
```

69

```
34  pf2=cos(atand(imag(Ib),real(Ib))*(%pi/180));
35  kw2=kv2*pf2;
36  printf('KVA shared by transformer 2 is %f KVA at %f
        lagging power factor\n',kv2,pf2);
37  printf('KW shared by transformer 2 is %f KW\n',kw2);
```

Scilab code Exa 1.54 Determining value of reactance connected in series with transformer B so that load current is equally shared

```
1  clc;
2  z1=0.4+2.2*%i; // leakage impedance referred to
        secondary for transformer 1
3  s1=510; // secondary no load voltage for transformer
        1
4  z2=0.6+1.7*%i; // leakage impedance referred to
        secondary for transformer 2
5  s2=500; // secondary no load voltage for transformer
        2
6  zl=5+3*%i; // load impedance
7  // current shared by both transformers should be
        equal i.e. I1=I2
8  t=(s2*(z1)-(s1-s2)*zl-(s1-s2)*zl)/s1;
9  xn=sqrt(abs(t)^2-real(z2)^2);
10 xe=xn-imag(z2); // external reactance to be added in
         series
11 printf('External reactance connected in series with
        leakage impedance of transformer 2 so that the
        current shared by both transformers are equal is
        %f ohm',xe);
```

Scilab code Exa 1.55 Determining tap setting to maintain rated voltage on secondary side for given loads

70

```
1  clc;
2  k=100; // Kva rating of transformer
3  disp('case(a)');
4  l1=90; // KVA rating of load
5  pf=0.8; // lagging pf of load
6  z=0.0075+0.09*%i; // p u leakage impedance of
       transformer
7  l1=l1/k; // load expressed in p u with base KVA=100
8  vd=l1*(real(z)*pf+imag(z)*sqrt(1-pf^2)); // p u
       voltage regulation
9  ts=vd*100; // tap setting
10 printf('Number of turns in the primary winding
       should be tapped down by %f percent\n',ts);
11 disp('case(b)');
12 lk=100; // KW rating of load
13 pf=0.8; // lagging pf of load
14 l1=lk/pf; // KVA rating of load
15 z=0.0075+0.09*%i; // p u leakage impedance of
       transformer
16 l1=l1/k; // load expressed in p u with base KVA=100
17 vd=l1*(real(z)*pf+imag(z)*sqrt(1-pf^2)); // p u
       voltage regulation
18 ts=vd*100; // tap setting
19 printf('Number of turns in the primary winding
       should be tapped down by %f percent\n',ts);
```

Scilab code Exa 1.56 Determining tap setting to maintain given voltage at load terminal

```
1  clc;
2  p=11000; // per phase h v side voltage
3  s=433; // l v side voltage
4  sp=s/sqrt(3); // per phase l v side voltage
5  K=10000; // VA rating of star connected load
6  pf=0.8; // power factor of load
```

```
7  z=0.5+1*%i; // Impedance per lead
8  zh=300+1500*%i; // per phase h v side leakage
      impedance
9  zl=0.2+1*%i; // per phase l v side leakage impedance
10 vl=400; // load voltage
11 k=p/sp; // turns ratio
12 zhl=zh/k^2; // h v side leakage impedance referred
      to l v side
13 zt=zhl+zl+z; // total impedance per phase between
      transformer and seconday load
14 il=K/(sqrt(3)*vl); // per phase load current
15 vd=round(il*(real(zt)*pf+imag(zt)*sqrt(1-pf^2))); //
       voltage drop per phase
16 vlp=vl/sqrt(3); // per phase load terminal voltage
17 E2=vlp+vd; // per phase voltage that must maintained
       l v terminals
18 vb=E2-sp; // voltage boost that tap changer must
      provide
19 ts=round((vb/sp)*100);
20 printf('tap down if the tapped coils are on h v side
       or tap up if the tapped coils are on l v side by
      %f percent',ts);
```

Scilab code Exa 1.57 Calculating primary winding current and line current

```
1 clc;
2 k=5; // Effective turns ratio
3 E1=400; // supply voltage for primary
4 il=10; // load current
5 E2=E1/k; // magnitude of maximum secondary induced
6 E1o=E1+E2; // maximum possibe value of output
      voltage
7 P=(il*E2)/1000; // Rating of secondary winding
8 ip=(il*E2)/E1; // neglecting noload current, primary
```

72

```
          winding  current
 9  I = ip + il ;
10  printf ( ' Total  primary  line  current  is  %f A ' ,I );
```

Scilab code Exa 1.58 Calculating KVA required to maintain the given voltage

```
 1  clc ;
 2  E1 =430;  // supply  voltage
 3  k =100000;  // VA rating  of  load
 4  il = k /( sqrt (3) * E1 );  // load  current
 5  v1 =380;  v2 =460;  // range  in  which  voltage  at  feeder
       end  varies
 6  vr = E1 - v1 ;  // maximum  variation  in  output  side  of
       regulator
 7  P =( sqrt (3) * vr * il )/1000;
 8  printf ( 'KVA  rating  of  induction  regulator  is  %f KVA '
       ,P );
```

Scilab code Exa 1.59 Calculating the rating of the primary and secondary winding

```
 1  clc ;
 2  V1 =400;  // supply  voltage
 3  E2 =50;  // induced  secondary  voltage
 4  l =8;  // KW rating  of  load
 5  pf =0.8;  // power  factor  of  load
 6  n =0.85;  // efficiency  of  induction  regulator
 7  ip =( l *1000)/( sqrt (3) * pf * n * V1 );  // input  current
 8  vmo = V1 - E2 ;  // minimum  output  voltage
 9  imo =( l *1000)/( sqrt (3) * pf * vmo );  // maximum  output
       current
```

```
10  ks=(sqrt(3)*E2*imo)/1000; // KVA rating of secondary
        winding
11  Vmo=V1+E2; // maximum output voltage
12  Imo=(l*1000)/(sqrt(3)*pf*Vmo); // minimum output
        current
13  //primary winding has to carry the current induced
        in it by secondary current due to transformer
        action and the difference of input current and
        output current
14  ipm=((Imo*E2)/V1)+ip-Imo; // maximum primary winding
        current
15  kp=(sqrt(3)*V1*ipm)/1000;
16  printf('Rating of primary winding is %f KVA\n',kp);
17  printf('Rating of secondary winding is %f KVA\n',ks)
        ;
```

Scilab code Exa 1.60 Determining the magnitude of maximum induced emf and For supply voltage finding the limit of output voltage and finding the rating of regulator for given load current

```
1  clc;
2  p=10; // number of full pitch coil in stator
3  t=120; // phase spread of coil
4  q=10.5*10^-3; // flux per pole in Weber
5  f=50; // frequency  of induction regulator
6  n=2; // number of turns in each coil;
7  E1=230; // supply voltage
8  il=30; // load current
9  N=p*n; // Total number of turns
10 kw=sin((t/2)*(%pi/180))/((t/2)*(%pi/180)); //
        distribution factor
11 E2=sqrt(2)*%pi*f*N*q*kw; // maximum secondary
        induced EMF
12 Vo1=E1+E2; Vo2=E1-E2; // range of output voltage
13 k=(il*E2)/1000;
```

74

```
14  printf('Maximum induced voltage is %f v\n',E2);
15  printf('range of output voltage is %f v-%f v\n',Vo2,
        Vo1);
16  printf('KVA rating of induction regulator is %f KVA'
        ,k);
```

Scilab code Exa 1.61 1 Determining self impedances of primary and secondary windings 2 Finding the value of equivalent circuit parameter referred to both sides 3 Determining secondary terminal voltage for given load

```
1  clc;
2  // subscript 1 and 2 indicates h v and l v winding
3  P=10000; // rated VA of transformer
4  E1=2300; // rated voltage
5  E2=230; // rated voltage
6  r1=10; // total resistance
7  r2=0.10; // total resistance
8  l1=40*10^-3 ; l2=4*10^-4; // self-inductances
9  m=10; // mutual inductance
10  k=E2/E1; // turns ratio
11  f=50; // frequency of supply;
12  disp('case 1');
13  L1=(m/k)+l1;
14  printf('Primary self inductance is %f H\n',L1);
15  L2=(m*k)+l2;
16  printf('Secondary self inductance is %f H\n',L2);
17  disp('case b');
18  r21=r2/k^2; // l v side resistance referred to h v
        side
19  l21=l2/k^2; // l v side self inductance referred to
        h v side
20  M1=m/k; // mutual inductance referred to h v side
21  printf('circuit parameters referred to primary
        winding are R1=%f ohm,R2=%f ohm,L1=%f H,L2=%f H
        and Lm1=%f H\n',r1,r21,l1,l21,M1);
```

75

```
22  r12=r1*k^2; // h v side resistance referred to l v
        side
23  l12=l1*k^2; // h v side self inductance referred to
        l v side
24  M2=m*k; // mutual inductance referred to l v side
25  printf('circuit parameters referred to secondary
        winding are R1=%f ohm,R2=%f ohm,L1=%f H,L2=%f H
        and Lm2=%f H\n',r12,r2,l12,l2,M2);
26  disp('case c');
27  lo=5+5*%i; // load connected to secondary
28  x1=2*%pi*f*l12; // leakage reactance
29  x2=2*%pi*f*l2; // leakage reactance
30  re2=real(lo)+r2+r12; // total resistance after
        referring to l v side
31  xe2=imag(lo)+x1+x2; // total reactance after
        referring to l v side
32  Z=re2+%i*xe2; // total impedance
33  vt=(E2/abs(Z))*abs(lo);
34  printf('Secondary terminal voltage is %f v',vt);
```

Scilab code Exa 1.62 1 Determining self inductances of hv and lv windings 2 Calculating mutual inductance between h v and l v windings 3 Finding coupling factors k1 and k2 for both windings and coefficient of coupling

```
1  clc;
2  n1=590; // primary side turns
3  n2=295; // secondary side turns
4  V1=230; // voltage source from which h v side was
        energised during test
5  io1=0.35; // no load current for when h v side is
        energised
6  V2=110; // induced voltage across open circuited l
        v winding due energised h v side
7  v2=115; // voltage source from which l v side was
        energised during test
```

```
8   io2 =0.72;  // no load current for when l v side is
        energised
9   v1 =226;  // induced voltage across open circuited  h
        v winding due energised l v side
10  f =50;  // frequency of supply;
11  w1=V1/( sqrt (2)*%pi*50);  // Maximum value of flux
        linkage with h v winding
12  L1=w1*(1/( sqrt (2)*io1));
13  printf ('self inductance of h v winding i %f H\n',L1)
        ;
14  w2=v2/( sqrt (2)*%pi*50);  // Maximum value of flux
        linkage with l v winding
15  L2=w2*(1/( sqrt (2)*io2));
16  printf ('self inductance of l v winding i %f H\n',L2)
        ;
17  M=(V2/( sqrt (2)*%pi*f))*(1/( sqrt (2)*io1));
18  printf ('mutual inductance between h v and l v
        winding is %f H\n',M);
19  k1=(n1/n2)*(M/L1);  // coupling factor for h v side
20  k2=(n2/n1)*(M/L2);  // coupling factor for l v side
21  k= sqrt (k1*k2);  // coefficient of coupling
22  printf ('coupling factor for h v side is %f\n',k1);
23  printf ('coupling factor for l v side is %f\n',k2);
24  printf ('coefficient of coupling is %f\n',k);
```

Scilab code Exa 1.63 Calculating the current that would flow in the winding 1 when the winding is connected to given supply and given load

```
1   clc ;
2   L1=4*10^-3;  // self indutance of winding 1
3   L2=6*10^-3;  // self indutance of winding 2
4   M=1.8*10^-3;  // mutual inductance of two windings
5   E1 =130;  // supply voltage for winding 1
6   f =500/%pi;  // frequency of supply
7   l =0.2*10^-3;  // load connected to winding 2
```

77

www.mepcafe.com

```
8  // writing voltage in rms form in matrix form V1=r1*
      I1+jw*L1*I1-jw*m*I2, V2=-r2*I2-jw*L2*I2+jw*m*I1
      to determine I1 and I2
9  t1=%i*2*%pi*f*L1;
10 t2=-%i*2*%pi*f*M;
11 t3=%i*2*%pi*f*M;
12 t4=-%i*2*%pi*f*l-%i*2*%pi*f*L2; // writing different
       terms of matrix
13 s=[t1 t2;t3 t4];
14 v=[130;0];
15 i=inv(s)*v; // calculating current in two windings
16 p=-imag(i(1));
17 printf('magnitude of current in primary winding is
      %f A',p);
```

Scilab code Exa 1.64 1 Determining the transformers turns ratio for maximum transfer of power 2 Determining load current voltage and power under maximum transfer of power

```
1  clc;
2  r=60; // resistive load which is coupled to
       electronic circuit
3  v=5; R=3000;  // electronic circuit represented by a
        voltage source in series with a internal
       resistance
4  // for maximum transfer of power, load resistance
       referred to primary must be equal to internal
       resitance of source
5  k=sqrt(R/r);
6  printf('turns ratio for maximum transfer of power is
       %f\n',k);
7  // referrig all quantities to load side
8  vl=v/k; // source voltage referred to load side
9  Rl=R/k^2; // source referred to load side
10 il=(vl/(Rl+r))*1000 ; // load current
```

```
11  vld=(il*r)/1000; // load voltage
12  p=(il^2*r)/1000; // load power
13  printf('load voltage is %f v\n',vld);
14  printf('load current is %f mA\n',il);
15  printf('load power is %f mW\n',p);
```

Scilab code Exa 1.65 Finding turns ratio for maximum transfer of power 2 Computing the load voltage at given frequencies

```
1  clc;
2  r1=20; // resistance of primary side
3  l1=1*10^-3; // leakage inductance of primary side
4  r2=0.5; // resistance of secondary side
5  l2=0.025*10^-3; // leakage inductance of secondary
       side
6  m=0.2; //  mutual inductance
7  l=50; // load in ohm connected to transformer
8  v=5; // voltage source
9  R=2000; // internal resistance of source
10 k=sqrt(R/l); // turns ratio for maximum power
       transfer
11 printf('Turns ratio is %f\n',k);
12 r21=0.5*k^2; // secondary resistance referred to
       primary
13 l21=l2*k^2; // secondary inductance referred to
       primary
14 lp=l*k^2; // load resistance referred to primary
15 rs=r1+r21+lp+R; // total series resistance
16 rp=((R+r1)*(R+r1))/rs; // equivalent resistance
17 leq=l1+l21; // equivalent inductance
18 f1=100; // frequency in hertz at which load voltage
       has to be calculated
19 Vl=(1/k)*(R/rs)*v*(1/(sqrt(1+(rp/(2*%pi*f1*m))^2)));
20 printf('load voltage at %f hz is %f v\n',f1,Vl);
21 f2=5000; // frequency in hertz at which load voltage
```

79

```
        has to be calculated
22  Vl=(1/k)*(R/rs)*v;
23  printf('load voltage at %f hz is %f v\n',f2,Vl);
24  f3=15000; // frequency in hertz at which load
        voltage has to be calculated
25  Vl=(1/k)*(R/rs)*(1/(sqrt(1+((2*%pi*f3*l1)/rs)^2)))*v
        ;
26  printf('load voltage at %f hz is %f v\n',f3,Vl);
```

Scilab code Exa 1.66 Computing secondary line voltage line current and output KVA for the given connection

```
1  clc;
2  // Answer for case c , secondary line voltage is
        given wrong in book
3  k=12; // per phase turns ratio
4  E1=11000; // supply voltage from feeder line
5  ip=20; // primary line current
6  disp('case a:star-delta');
7  vph=E1/sqrt(3); // primary phase voltage
8  iph=ip; // phase current on primary
9  sph=vph/k; // secondary phase voltage
10 vls=sph;
11 printf('Line voltage on secondary is %f v\n',vls);
12 isph=k*iph; // phase current on secondary
13 isl=sqrt(3)*isph;
14 printf('line current on secondary is %f A\n',isl);
15 Kv=(3*sph*isph)/1000;
16 printf('Output KVA is %f KVA\n',Kv);
17 disp('case b:delta-star');
18 vph=E1; // primary phase voltage
19 iph=ip/sqrt(3); // phase current on primary
20 sph=vph/k; // secondary phase voltage
21 vls=sqrt(3)*sph;
22 printf('Line voltage on secondary is %f v\n',vls);
```

80

```
23  isph=k*iph; // phase current on secondary
24  isl=isph;
25  printf('line current on secondary is %f A\n',isl);
26  Kv=(3*sph*isph)/1000;
27  printf('Output KVA is %f KVA\n',Kv);
28  disp('case c:delta-delta');
29  vph=E1; // primary phase voltage
30  iph=ip/sqrt(3); // phase current on primary
31  sph=vph/k; // secondary phase voltage
32  vls=sph;
33  printf('Line voltage on secondary is %f v\n',vls);
34  isph=k*iph; // phase current on secondary
35  isl=sqrt(3)*isph;
36  printf('line current on secondary is %f A\n',isl);
37  Kv=(3*sph*isph)/1000;
38  printf('Output KVA is %f KVA\n',Kv);
39  disp('case d:star-star');
40  vph=E1/sqrt(3); // primary phase voltage
41  iph=ip; // phase current on primary
42  sph=vph/k; // secondary phase voltage
43  vls=sqrt(3)*sph;
44  printf('Line voltage on secondary is %f v\n',vls);
45  isph=k*iph; // phase current on secondary
46  isl=isph;
47  printf('line current on secondary is %f A\n',isl);
48  Kv=(3*sph*isph)/1000;
49  printf('Output KVA is %f KVA\n',Kv);
```

Scilab code Exa 1.67 Determining transformer rating and phase and line current on both high and low voltage sides

```
1  clc;
2  E1=11000; // line voltage of primary
3  E2=415; // line voltage of secondary
4  kw=30; // KW rating of 3 phase induction motor
```

81

```
 5  n =0.9;  // efficieny
 6  pf =0.833;  // power factor at which load is operating
 7  Kv= kw /( n* pf );  // KVA rating of transformer
 8  printf ('KVA rating of transformer is %f KVA\n',Kv);
 9  ilv =( Kv *1000)/( sqrt (3)* E2 );  // line current on l v
        side
10  // secondary is star connected therefore line current
        = phase current
11  printf ('Line current on l v side is %f A\n',ilv);
12  printf ('Phase current on l v side is %f A\n',ilv);
13  ilp =( Kv *1000)/( sqrt (3)* E1 );  // line current on h v
        side
14  ipp = ilp / sqrt (3);
15  printf ('Line current on h v side is %f A\n',ilp);
16  printf ('Phase current on h v side is %f A\n',ipp);
```

Scilab code Exa 1.68 1 Finding power consumed by load 2 KVA rating of transformer 3 Phase and line currents on both h v and l v side

```
 1  clc ;
 2  il =100;  // load current
 3  pf =0.8;
 4  E1 =11000;  // primary line voltage
 5  E2 =400;  // secondary line voltage
 6  p =( sqrt (3)* E2 * il * pf )/1000;
 7  printf ('power consumed by load is %f KW\n',p);
 8  k =( sqrt (3)* E2 * il )/1000;
 9  printf ('KVA rating of load is %f KVA\n',k);
10  iph =( k *1000)/( sqrt (3)*11000);  // phase current on h
        v side
11  // primary is star connected therefore line current=
        phase current
12  printf ('Line current on h v side is %f A\n',iph);
13  printf ('Phase current on h v side is %f A\n',iph);
14  ipl = il / sqrt (3);
```

```
15  printf('Line current on l v side is %f A\n',il);
16  printf('Phase current on l v side is %f A\n',ipl);
```

Scilab code Exa 1.69 1 Determining number of primary turns 2 current power factor and power on primary side under no load

```
1   clc;
2   B=1.2; // maximum flux density in core
3   H=600; // magnetic field intensity in A/m
4   d=7.8; // gram density in g/cm^3
5   cl=2; // core loss per kg
6   e=1200; // supply voltage
7   f=50; // frequency of supply voltage
8   t=10; // thickness of core in cm
9   w=40;
10  L=30; // dimensions of core
11  s=0.9; // stacking factor
12  ga=t^2*10^-4; // gross core area in m^2
13  nga=ga*s; // net gross core area in m^2
14  q=nga*B; // maximum flux in core
15  N=(e)/(sqrt(2)*%pi*f*q);
16  printf('Number of turns in primary is %f\n',N);
17  ml=((w+t)+(L+t))*2; // mean flux path in cm
18  mmf=H*(ml/100); // mmf of the core
19  mi=mmf/N; // maximum value of magnetizing current
20  irm=mi/sqrt(2); // rms value of magnetizing current
21  cv=(ml/100)*nga; // core volume
22  wc=cv*d*10^3; // weight of core
23  pc=wc*cl; // total core loss
24  Ic=pc/e; // core loss current
25  io=sqrt(Ic^2+irm^2);
26  printf('No load current on primary side is %f A\n',
        io);
27  pf=Ic/io;
28  printf('No load power factor is %f\n',pf);
```

83

```
29  pf = e * io * pf ;
30  printf ( ' power  input  at  no  load  is  %f W ' , pf ) ;
```
_____

Scilab code Exa 1.70 Determining magnetizing current for the given figs

```
1  clc ;
2  // Three  core  type  transformers  are  given  in  fig
       1.80
3  // For  first  core  type  transformer
4  im1 =4;  // magnetizing  core
5  e2 =100;  // emf  induced  in  secondary  winding
6  B =1;  // maximum  flux  density  in  Tesla
7  // mmf is  directly  proportional  maximum  flux  in  core
        i.e im*N(no.  of  turns )=kq( flux ) , k  is
       proportionality  constant
8  // for  fig (80( b )) ,qm2( flux  for  core  transformer  1)=
       qm1( flux  for  core  transformer  2) , that  is  flux  in
        both  coils  in  core  transformer  2 is  qm1 /2;
9  // for  upper  coil  im2*N  is  directly  proportional  to
       qm1 /2
10  // for  lower  coil  Im2*N  is  directly  proportional  to
       qm2 /2
11  // adding  above  relation  we  get  im2+Im2 =4( magnetizing
        current )
12  Im2 = im1 /2;
13  im2 = Im2 ;  // magnetizing  current  of  each  coil  is  2 A
14  imt = Im2 + im2 ;  // total  magnetizing  current  for
       transformer  2
15  // since  flux  is  same  for  both  transformer , emf
       induced  is  also  same
16  // since  flux  is  same  for  both  transformer , area  is
       same  ,  therefore  magnetic  flux  density  is  also
       same
17  printf ( ' Magnetizing  current  for  transformer  2 is  %f
       A\n ' , imt ) ;
```

84

```
18 printf('emf induced in secondary for transformer 2
      is %f v\n',e2);
19 printf('Magnetic flux density in transformer 2 is %f
       T\n',B);
20 // for fig (80(c)), qm3=qm1/2
21 // qm1~im1*N,qm3~im3*N; ~-sign of directly
      proportional (assumption)
22 // from above two relations, we get
23 im3=im1/4;
24 B3=B/2;
25 E2=e2/2;
26 printf('Magnetizing current for transformer 3 is %f
      A\n',im3);
27 printf('emf induced in secondary for transformer 3
      is %f v\n',E2);
28 printf('Magnetic flux density in transformer 3 is %f
       T\n',B3);
```

Scilab code Exa 1.71 Determining load voltage load power factor and load power

```
1 clc;
2 P=20; // Rated KVA of transformer
3 E1=250; // rated primary voltage
4 E2=125; // rated secondary voltage
5 r1=0.15; // resistance of primary side
6 x1=0.25; // leakage reactance of primary side
7 r2=0.03; // resistance of secondary side
8 x2=0.04; // leakage reactance of secondary side
9 // given E1=V1(primary terminal voltage)
10 k=E2/E1; // turns ratio
11 ip=(P*1000)/E1; // full load primary current
12 // voltage regulation=0, because E1=V1 therefore
13 pf=-atand(r1,x1); // phase angle between E1 and ip
14 // negative sign indicates that current leads
```

```
              voltage
15 re2=r2/k^2; // secondary resistance referred to
       primary
16 xe2=x2/k^2; // secondary leakage reactance referred
       to primary
17 ip=ip*(cos(pf*(%pi/180))-%i*sin(pf*(%pi/180))); //
       complex form of primary current
18 V2=E1-ip*(re2+xe2*%i);
19 pfl=-atand(imag(V2),real(V2))-pf; // phase angle by
       which primary current leads secondary terminal
       voltage referred to primary
20 PF=cos(pfl*(%pi/180));
21 vl=abs(V2)/2;
22 isl=(P*1000)/E2;
23 pl=vl*isl*PF;
24 printf('Load voltage is %f v\n',vl);
25 printf('Load power factor is %f leading\n',PF);
26 printf('Load power is %f W',pl);
```

Scilab code Exa 1.72 Calculating power delivered by each source and power dissipated in given resistor

```
1 clc;
2 // from fig 1.82
3 E1=5; // supply voltage
4 E2=20; // induced secondary voltage
5 k=E2/E1; // turns ratio
6 r1=6; // primary parameters
7 r2=16; // secondary parameters
8 r21=r2/k; // secondary parameters referred to
       primary
9 E21=(E2*2)/k; // secondary voltage referred to
       primary
10 theta=60*(%pi/180); // phase angle assocoated with
       E2
```

86

```
11  // after referring to primary side, with E1 as a
        reference
12  V=E21*(cos(theta)-%i*sin(theta))-E1; //resultant
        voltage
13  I=abs(V)/(r1+r21); // magnitude of current
14  pd1=I^2*r1;
15  pd2=I^2*r21;
16  printf('power dissipated in %f ohm resistor is %f W\
        n',r1,pd1);
17  printf('power dissipated in %f ohm resistor is %f W\
        n',r21,pd2);
18  // Current lags E1 by 90 degree
19  teta1=90*(%pi/180);
20  // Since E2 lags E1 by 60 degree and Current due to
        resultant voltage lags E1 by 90, therefore phase
        difference Current I and E2 is
21  teta2=(90-60)*(%pi/180);
22  Pd1=E1*I*cos(teta1);
23  Pd2=E21*I*cos(teta2);
24  printf('power delivered by %f v source is %f W\n',E1
        ,Pd1);
25  printf('power delivered by %f v source is %f W\n',E2
        ,Pd2);
```

Scilab code Exa 1.73 Calculating efficiency and regulation under given condition

```
1  clc;
2  P=1200; // rated VA of transformer
3  E1=240; // rated primary voltage
4  E2=110; // rated secondary voltage
5  xe=5; // total leakage reactance of transformer
        referred to primary
6  re=1; // total resistance of transformer referred to
        primary
```

```
7  vl=240; // rated voltage of load
8  R=2500; // core resistance
9  pf=1; // power factor
10 ip=P/E1; // full load primary current
11 // assuming E1 as a reference
12 Vin=sqrt((E1+ip*re)^2+(ip*xe)^2); // input voltage
       when load is connected
13 pi=ip^2*re; // ohmic losses
14 pc=Vin^2/R; // core loss
15 n=(P*pf*100)/(P*pf+pi+pc);
16 printf('Efficiency at full load is %f percent\n',n);
17 vr=((Vin-E1)/Vin)*100;
18 printf('Voltage regulation for full load is %f
       percent',vr);
```

Scilab code Exa 1.74 Determining current taken from source primary input impedance and input power

```
1  clc;
2  k1=4; // turns ratio for transformer 1
3  k2=3; // turns ratio for transformer 2
4  E1=120; // supply across which two transformers are
       connected in parallel
5  E21=E1/k1; // secondary voltage for transformer 1
6  E22=E1/k2; // secondary voltage for transformer 2
7  R=10; // load resistance
8  // Secondary windings are connected in series with
       the polarity such that the voltages E21 and E22
          aid each other
9  E2=E21+E22; // resultant output voltage
10 il=E2/R; // Load current
11 // for mmf balance, primary current for T1
12 ip1=il/k1;
13 // for mmf balance, primary current for T2
14 ip2=il/k2;
```

```
15  ip=ip1+ip2; // total current drawn from source
16  printf('Current drawn from source is %f A\n',ip);
17  zi=E1/ip;
18  printf('Primary input impedance is %f ohm\n',zi);
19  pi=E2*il;
20  printf('Primary input power is %f W',pi);
```

Scilab code Exa 1.75 Finding currents drawn by primary and secondary of transformer A and open circuit voltage across secondary of transformer B

```
1  clc;
2  // A and B are two transformer shown in fig 1.85
3  im=0.1; // magnetizing current
4  // Since SA(secondary winding for transformer A) is
        connected directly across 200 v supply, so the
        magnetizing current required to establish the
        flux must flow through SA, therefore current
        taken by winding PA(primary winding for
        transformer A) is zero
5  // Since voltage across SA is 200 v , voltage across
         PA has to be 200 v by transformer action
6  // As a result of it , voltage across PB is zero ,
        therefore induced emf in SB is zero
7  printf('Current in secondary winding of transformer
       A is %f A',im);
8  disp('Current in primary winding of transformer A is
        zero');
9  disp('Voltage across secondary winding of
        transformer B is zero');
```

Scilab code Exa 1.76 Determining the reading of ideal voltmeter

```
1  clc;
```

89

www.mepcafe.com

```
2  // 1 and 2 subscripts are used for transformer 1 and
       transformer 2
3  // P and S are used for primary and secondary
      winding
4  E=230; // Rated primary voltage of both transformer
5  disp('case 1');
6  R=0; // load conneted across secondary of
      transformer 1
7  // R=0 means winding is short circuited , therefore
      voltage across S1 is zero, therefore whole
      voltage is applied across s2 therefore
8  printf('Reading of voltmeter for R=%f is %f v\n',R,E
      );
9  disp('case 2');
10 R=115; // load conneted across secondary of
      transformer 1
11 // A resistance of 115 ohm should cause a current of
       1 A because voltage across P1 is 115 v but this
      means that P1 and P2 carrying 1A current .
      According to this voltage across P2 is
      magnetizing impedance of transformer 2 times
      magnetizing current . But magnetizing impedance is
       very large , due to this voltage across P2 rises
      much above 115 v and voltage across P1 falls , due
       to which magnetizing    current through S1
      decreases .
12 disp('Voltmeter reading is much more than 115 v but
      less than 230 v. Let it be Vb');
13 disp('case 3');
14 R=1000; // load conneted across secondary of
      transformer 1
15 // For R=1000 ohm , current in P1 and S1 are reduced
      . Hence current in P2 is also reduced . Therefore
      voltage across P2 is also reduced but still it is
       more than 115 v
16 disp('Voltmeter reading is more than 115 v but less
      than Vb');
17 disp('case 4');
```

90

```
18  // for R=infinity S1 is open circuited , therefore
       voltage across P1=115 v and P2=115 v
19  disp('Reading of voltmeter is 115 v');
```

Scilab code Exa 1.77 Finding current drawn by transformer under no load test

```
1  clc;
2  E1=200; // rated voltage of l v side
3  E2=400; // rated voltage of h v side
4  f=50; // frequency of supply
5  W=80; // open circuit input wattage=core loss
6  m=1.91; // mutual induction between h v and l v side
7  Ic=W/E2; // core loss current
8  Qmax=E1/(sqrt(2)*%pi*f); // maximum value of flux
       linkage with l v winding
9  Im=Qmax/(sqrt(2)*m);
10 Io=sqrt(Ic^2+Im^2);
11 printf('Current taken by transformer when no load
       test is conducted on h v side is %f A',Io);
```

Scilab code Exa 1.78 Determining the efficiency and voltage regulation of transformer

```
1  clc;
2  P=100000; // rated VA of transformer
3  n=0.98; // maximum possible efficiency
4  l=80000; // rated KVA of load
5  vrm=0.04; // maximum possible voltage regulation is
       equal to ze2 in p u
6  pf=0.8; // power factor at which efficiency anf
       voltage regulation has to be determined
7  pl=((1/n)-1)*l; // total losses in transformer
```

91

www.mepcafe.com

```
8  pc=pl/2;  // core losses; at maximum efficiency ohmic
        losses = core losses
9  po=(1/pf)^2*pc;  // ohmic losses at given power
        factor
10 N=(P*pf*100)/(l+po+pc);
11 printf('Efficiency at %f lagging power factor is %f
        percent\n',pf,N);
12 re2=po/P;  // resistance in p u
13 xe2=sqrt(vrm^2-re2^2);
14 vr=(re2*pf+xe2*sqrt(1-pf^2))*100;
15 printf('Voltage regulation at %f lagging power
        factor is %f percent',pf,vr);
```

Scilab code Exa 1.79 Determining the KVA rating of different section of autotransformer

```
1  clc;
2  // after deriving the expression for both the
        transformer and auto transformer
3  k=0.8;  // ratio of V2/V1(turns ratio)
4  W=100;  // Power to be delivered in KW
5  pf=1;  // unity power factor
6  n=0.96;  // given efficiency
7  disp('for two winding transformer');
8  ks=W;
9  printf('KVA rating for secondary of two winding
        transformer is %f KVA\n',ks);
10 kp=ks/n;
11 printf('KVA rating for primary of two winding
        transformer is %f KVA\n',kp);
12 disp('for auto transformer');
13 kab=(W/n)*(1-k);
14 printf('KVA rating for section AB of autotransformer
        is %f KVA\n',kab);
15 kbc=W*(1-k/n);
```

92

```
16  printf('KVA rating for section BC of autotransformer
        is %f KVA\n',kbc);
```

Chapter 2

Electromechanical Energy Conversion Principle

Scilab code Exa 2.4 Determining magnitude of average magnetic force

```
1  clc;
2  l=0.02; // air gap length
3  i1=20; // intermediate current
4  i2=40; // current during armature movement from open
        to close position
5  // from fig 2.11
6  f1=0.04*i1; // flux linkage during open position at
      A
7  f2=1.2+0.03*(i1-20); // flux linkage during close
      position at D
8  f3=0.04*i2; // flux linkage during open position at
      B
9  f4=1.2+0.03*(i2-20); // flux linkage during close
      position at C
10 // Mechanical work done=area ODCFEO-area OABFEO
11 W=((i1*f2)/2)+(((f2+f4)*i1)/2)-((i2*f3)/2);
12 fe=W/l;
13 printf('Average electromagnetic force is %d N',fe);
```

Scilab code Exa 2.6 Determining force required for axial alignment of electromagnets

```
1  clc;
2  g=0.003; // gap length
3  wp=0.006; // pole width
4  B=0.8; // flux density in air gap
5  uo=4*%pi*10^-7; // free space permeability
6  // after the derivation of expression
7  fe=(B^2*wp*g)/(2*uo);
8  printf('Force tending to bring electromagnets into
       axial alignment is %f N',fe);
```

Scilab code Exa 2.7 Determining 1 air gap flux densities and coil inductance 2 energy stored in magnetic field 3 electromagnetic force 4 mechanical work done 5 electrical energy supplied by load

```
1  clc;
2  N=1500; // number of turns in coil
3  i=3; // current carried by coil
4  uo=4*%pi*10^-7; // free space permeability
5  l=0.04; // side of plunger
6  A=%pi*(l/2)^2; // cross section area of plunger
7  disp('case a');
8  disp('for gap length G=2 cm');
9  g=0.02; // air gap length in cm
10 x=0; // displacement of plunger
11 G1=g-x; // gap length
12 B=(uo*i*N)/G1; // air gap flux density
13 printf('Air gap flux density is %f Wb/m^2\n',B);
14 L1=(N^2*uo*A)/G1;
15 printf('Coil inductance is %f H',L1);
```

95

```scilab
16  disp('for gap length G=1.5 cm');
17  g=0.02; // air gap length in cm
18  x=0.005; // displacement of plunger
19  G2=g-x; // gap length
20  B=(uo*i*N)/G2; // air gap flux density
21  printf('Air gap flux density is %f Wb/m^2\n',B);
22  L2=(N^2*uo*A)/G2;
23  printf('Coil inductance is %f H',L2);
24  disp('for gap length G=1 cm');
25  g=0.02; // air gap length in cm
26  x=0.01; // displacement of plunger
27  G3=g-x; // gap length
28  B=(uo*i*N)/G3; // air gap flux density
29  printf('Air gap flux density is %f Wb/m^2\n',B);
30  L3=(N^2*uo*A)/G3;
31  printf('Coil inductance is %f H',L3);
32  disp('for gap length G=0.5 cm');
33  g=0.02; // air gap length in cm
34  x=0.015; // displacement of plunger
35  G4=g-x; // gap length
36  B=(uo*i*N)/G4; // air gap flux density
37  printf('Air gap flux density is %f Wb/m^2\n',B);
38  L4=(N^2*uo*A)/G4;
39  printf('Coil inductance is %f H',L4);
40  disp('case b');
41  disp('for gap length G=2 cm');
42  W=(i^2*L1)/2;
43  printf('Energy stored is %f watt-sec\n',W);
44  disp('for gap length G=1.5 cm');
45  W=(i^2*L2)/2;
46  printf('Energy stored is %f watt-sec\n',W);
47  disp('for gap length G=1 cm');
48  W=(i^2*L3)/2;
49  printf('Energy stored is %f watt-sec\n',W);
50  disp('for gap length G=0.5 cm');
51  W=(i^2*L4)/2;
52  printf('Energy stored is %f watt-sec\n',W);
53  disp('case c');
```

```
54  disp('for gap length G=2 cm');
55  fe=round((i^2*g*L1)/(2*G1^2));
56  printf('Electromagnetic force is %f N\n',fe);
57  disp('for gap length G=1.5 cm');
58  fe=(i^2*g*L1)/(2*G2^2);
59  printf('Electromagnetic force is %f N\n',fe);
60  disp('for gap length G=1 cm');
61  fe=round((i^2*g*L1)/(2*G3^2));
62  printf('Electromagnetic force is %f N\n',fe);
63  disp('for gap length G=0.5 cm');
64  fe=round((i^2*g*L1)/(2*G4^2));
65  printf('Electromagnetic force is %f N\n',fe);
66  disp('case 4');
67  // for g=2 cm and g=0.5cm, displacement is given by
68  xi=0;
69  xf=0.015;
70  Wm=integrate('(i^2*g*L1)/(2*(g-x)^2)','x',xi,xf);
71  printf('Mechanical work done is %f watt-sec\n',Wm);
72  disp('case e');
73  We=integrate('(i^2*g*L1)/(g-x)^2','x',xi,xf);
74  printf('Electrical energy supplied by source is %f
        watt-sec\n',We);
```

Scilab code Exa 2.8 Determining torque

```
1  clc;
2  r=50*10^-3; // radius of rotor
3  g=2*10^-3; // air gap length
4  l=10*10^-3; // length normal to radius r
5  B=2.2; // maximum air gap flux density
6  uo=4*%pi*10^-7; // free space permeability
7  // after the derivation of the expression
8  T=(B^2*g*r*l)/uo;
9  printf('Magnitude of torque is %f N-m',T);
```

97

www.mepcafe.com

Scilab code Exa 2.9 Determining torque

```
1  clc;
2  r=50*10^-3; // radius of rotor
3  g=2*10^-3; // air gap length
4  l=10*10^-3; // length normal to radius r
5  B=2.2; // maximum air gap flux density
6  uo=4*%pi*10^-7; // free space permeability
7  // after the derivation of the expression
8  T=(B^2*g*(r+(g/2))*l)/uo;
9  printf('Magnitude of torque is %f N-m',T);
```

Scilab code Exa 2.13 Determining magnitude and direction of torque for different cases

```
1  clc;
2  f=50; // frequency
3  w=2*%pi*f; // angular speed
4  y=60; // y=angular position of rotor
5  Ls=0.6+0.2*cosd(2*y) // self inductance of stator
6  dLs=-2*0.2*sind(2*y); // derivative of Ls with y
7  Lr=0.75+0.3*cosd(2*y) // self inductance of rotor
8  dLr=-2*0.3*sind(2*y); // derivative of Lr with y
9  Ms=0.8*cosd(y) // mutual inductance of stator
10 dMs=-0.8*sind(y); // derivative of Ms with y
11 disp('case a');
12 is=20; // stator current
13 ir=10; // rotor current
14 te=(is^2*dLs)/2+(ir^2*dLr)/2+(is*ir)*dMs;
15 printf('Magnitude of torque is %f Nm and since it is
        negative it acts in such a direction so as to
      decrease angular position\n',-te);
```

```
16  is=20; // stator current
17  ir=-10; // rotor current
18  te=((is^2*dLs)/2)+((ir^2*dLr)/2)+((is*ir)*dMs);
19  printf('Magnitude of torque is %f Nm and it acts in
         clockwise direction \n',te);
20  is=20; // stator current
21  ir=0; // rotor current
22  te=((is^2*dLs)/2)+((ir^2*dLr)/2)+((is*ir)*dMs);
23  printf('Magnitude of torque is %f Nm and it acts in
         counter−clockwise direction \n',-te);
24  disp('case b');
25  // rotor winding is short circuited rotor voltage=0
         and is=sqrt(2)*20*sin(wt) here average torque is
         needed so for calculation we need not to worry
         about sin(wt)
26  is=sqrt(2)*20; // stator current
27  ir=(-Ms/Lr)*is; // rotor current
28  te=((is^2*dLs)/2)+((ir^2*dLr)/2)+((is*ir)*dMs);
29  printf('Magnitude of torque is %f Nm and it acts in
         clockwise direction \n',te/2);
30  disp('case c');
31  // vs=sqrt(2)*314*sin(wt) again here average torque
         is needed so for calculation we need not to worry
          about sin(wt)
32  vs=sqrt(2)*314; // stator winding voltage
33  ls1=(Ls-(Ms^2/Lr)); // short circuit inductance of
         stator winding
34  is=vs/(w*ls1); // stator current
35  ir=(-Ms/Lr)*is; // rotor current
36  te=((is^2*dLs)/2)+((ir^2*dLr)/2)+((is*ir)*dMs);
37  printf('Magnitude of torque is %f Nm and it acts in
         clockwise direction \n',te/2);
```

Scilab code Exa 2.15 Determining magnitude of exciting current for closing relay and keeping it closed

```
1  clc;
2  B=[ 0.2 0.4 0.6 0.8 1 1.2];
3  H=[ 50 100 160 225 300 400];
4  plot(H,B);
5  xlabel('magnetic field intensity');
6  ylabel('magnetic flux density');
7  title('B–H curve');
8  p=0.2; // force exerted by spring
9  g1=0.5*10^-2; // air gap length
10 g2=0.1*10^-2; // reduced gap length after coil is
      energised
11 n=2000; // coil turns
12 l=0.2; // mean length of magnetic iron path
13 A=0.2*10^-4; // area of cross-section
14 g=9.81; // acceleration due to gravity
15 l1=6; // gap length
16 l2=3; // gap length between spring and core
17 uo=4*%pi*10^-7; // free space permeability
18 disp('case a');
19 fe=(p*g*l2)/l1; // electromagnetic torque
20 Bg=sqrt((2*fe*uo)/A); // air gap flux density
21 printf('Air gap flux density is %f T\n',Bg);
22 // corresponding to value of Bg from B–H curve H is
23 Hg=87.7; // magnetic flux intensity
24 ATi=l*Hg; // total ampere turn for iron path
25 ATg=(Bg*g1)/uo; // ampere turn for air gap
26 AT=ATi+ATg; // total ampere turns
27 ie=AT/n;
28 printf('Exciting current required to close the
      armature relay is %f A\n',ie);
29 disp('case b');
30 ATg=(Bg*g2)/uo; // ampere turn for air gap
31 AT=ATi+ATg; // total ampere turns
32 ie=AT/n;
33 printf('Exciting current required to keep the
      armature closed is %f A\n',ie);
```

Scilab code Exa 2.16 Determining magnitude of unbalanced magnetic pull on armature

```
1 clc;
2 x=0.1*10^-3; // displacement of armature
3 B=0.8; // air gap flux density
4 A=200*10^-4; // area under pole
5 g=0.6*10^-2; // air gap length
6 uo=4*%pi*10^-7; // free space permeability
7 // after derived expression
8 p=(B^2*A*g*x)/(uo*(g^2-x^2));
9 printf('Unbalanced magnetic pull on armature is %f N
      ',p);
```

Scilab code Exa 2.17 Determining magnitude and direction of magnetic force

```
1 clc;
2 i=sqrt(2)*1000; // maximum current carried by
      conductors
3 z=2; // number of conductors in slot
4 l=1; // embedded length
5 w=0.05; // slot width
6 uo=4*%pi*10^-7; // free space permeability
7 // after derived expression
8 fe=(uo*z^2*l*i^2)/(2*w);
9 printf('Magnitude of magnetic force is %f N and its
      direction is towards the bottom of the slot\n',fe
      /2);
```

Scilab code Exa 2.19 Determining 1 mechanical work done 2 energy supplied by each electrical source 3 change in field energy

```scilab
1  clc;
2  // L1=3+(1/(2*X)) -self inductance of coil 1
3  // L1=2+(1/(2*X)) -self inductance of coil 2
4  // M=1/(2*X) -mutual inductance
5  // X is displacement
6  i1=10; // current of coil 1
7  i2=-5; // current of coil 2
8  // from expression W=(L1*i1^2)/2 + (L2*i2^2)/2 + (i1
       *i2*M) where W is work done and is equal to
       175+25/(4*x);
9  // fe=-25/(4*x^2) where fe is electromagnetic force
       and is equal to rate of change of work done with
       respect to x
10 disp('case a');
11 x1=0.5;
12 x2=1;
13 Wm=integrate('-25/(4*x^2)','x',x1,x2);
14 printf('Mechanical work done for given increment in
       displacement is %f watt-sec\n',Wm);
15 disp('case b')
16 // psi1=(3+(1/(2*x)))*i1+(1/(2*x))*i2 psi1 is flux
       linkage with coil 1
17 // psi2=(2+(1/(2*x)))*i2+(1/(2*x))*i1 psi2 is flux
       linkage with coil 2
18 We1=(((3+(1/(2*x2)))*i1+(1/(2*x2))*i2)-((3+(1/(2*x1)
       ))*i1+(1/(2*x1))*i2))*i1;
19 We2=(((2+(1/(2*x2)))*i2+(1/(2*x2))*i1)-((2+(1/(2*x1)
       ))*i2+(1/(2*x1))*i1))*i2;
20 printf('Electrical energy supplied by source 1 is %f
        watt-sec\n',We1);
21 printf('Electrical energy supplied by source 2 is %f
        watt-sec\n',We2);
22 disp('case c');
23 // W=175+(25/(4*x));
24 dw=175+(25/(4*x2))-(175+(25/(4*x1)));
```

102

```
25 printf('Change in field energy is %f Watt-sec\n',dw)
    ;
```

Scilab code Exa 2.21 Determining force between two magnetic surfaces

```
1 clc;
2 B=1.6; // flux density between magnetic surfaces
3 a=1; // area of magnetic surfaces
4 uo=4*%pi*10^-7; // free space permeability
5 f=(B^2*a)/(2*uo);
6 printf('Force between two magnetic surfaces is %f N'
    ,f);
```

Scilab code Exa 2.22 Determining force between two plates

```
1 clc;
2 A=1; // area of plates
3 E=3*10^6; // electric field intensity between plates
4 Eo=(10^-9/(36*%pi)); // vaccum permittivity
5 // after using both energy and coenergy method
    expression for force is derived
6 f=(E^2*Eo*A)/2;
7 printf('Force between two plates is %f N',f);
```

Scilab code Exa 2.23 Determining magnitude of voltage applied between two plates

```
1 clc;
2 w=5*10^-3; // counter weight
3 a=30*10^-4; // area of cross section of plates
```

```
4  d=1*10^-2; // distance between two plates
5  g=9.81; // acceleration due to gravity
6  Eo=(10^-9/(36*%pi)); // vaccum permittivity
7  l1=8*10^-2; // horizontal distance between pivot and
       plates
8  l2=10*10^-2; // horizontal distance between pivot
     and counter weight
9  // for the bar to remain horizontal electrostatic
     force is given by
10 fe=(w*l2*g)/l1;
11 v=sqrt((fe*2*d^2)/(Eo*a));
12 printf('Voltage applied between the plates is %f KV'
     ,v/1000);
```

Scilab code Exa 2.25 Determining 1 coil inductance and magnetic energy 2 electrical energy supplied by source 3 mechanical work done

```
1  clc;
2  n=1000; // number of turns in exciting coil
3  a=5*5*10^-4; // cross section area of core
4  g=1*10^-2; // length of air gap
5  uo=4*%pi*10^-7; // free space permeability
6  disp('case a');
7  i=5; // coil current
8  l=(n^2*uo*a)/(2*g);
9  printf('Inductance of coil is %f H\n',l);
10 E=(l*i^2)/2;
11 printf('Field energy stored in inductor is %f Watt-
     sec\n',E );
12 fe=(i^2*a*n^2*uo)/(4*g^2);
13 printf('Force on the armature is %f N\n',fe);
14 disp('case b');
15 g1=0.5*10^-2; // reduced length of air gap
16 We=((n^2*uo*a)/(2*g1)-(n^2*uo*a)/(2*g))*i^2;
17 printf('Electrical energy supplied by source is %f
```

104

```
     Watt-sec\n',We);
18  disp('case c');
19  w=integrate('(n^2*uo*a*i^2)/(4*(g-x)^2)','x',0,g1);
20  printf('Mechanical work done is %f Watt-sec',w);
```

Scilab code Exa 2.26 Determining force required for armature alignment with field structure

```
1  clc;
2  de=110*(%pi/180); // pole pitch
3  g=0.4*10^-2; // air gap length
4  B=0.5; // air gap flux density
5  d=0.3; // armature diameter
6  uo=4*%pi*10^-7; // free space permeability
7  fe=(B^2*d*de*g)/(2*uo);
8  printf('Force that tends to pull the armature into
        alignment is %f N',fe)
```

Scilab code Exa 2.28 Determining energy stored in inductor

```
1  clc;
2  r=4; // resistance of inductor
3  v=8; // maximum voltage
4  L=2; // inductance
5  t=2; // time required to reach maximum voltage value
6  disp('case a');
7  // after solving laplace equation we get expression
        for i(t) (transient current) i.e         i(t)=((
        exp^-2*t)+(2*t-1))/2
8  // for t=2 sec i(t) is given by
9  it=((exp(-2*t))+2*t-1)/2;
10 E=(L*it^2)/2;
11 printf('Energy stored during %d sec is %f J\n',t,E);
```

```
12  disp('case b');
13  i=v/r; // current when transients are over
14  E=(L*i^2)/2;
15  printf('Energy stored after transients are over is
       %d J\n',E);
```

Scilab code Exa 2.29 Determining and comparing field energy stored and field energy density in iron and air gap

```
1  clc;
2  l=1.2; // length of iron path
3  a=5*5*10^-4; // area of cross section
4  uo=4*%pi*10^-7; // permeability for free space
5  ur=1500; // relative permeability for iron
6  i=2; // exciting current
7  n=1000; // number of turns of coil
8  g=0.5*10^-2; // air gap length
9  r=(l/(uo*ur*a))+(g/(uo*a)); // net reluctance
10 f=(n*i)/r; // flux in coil
11 fe1=((f^2*l)/(uo*ur*a))/2;
12 printf('Field energy stored in iron is %f J\n',fe1);
13 fe2=((f^2*g)/(uo*a))/2;
14 printf('Field energy stored in air gap is %f J\n',
       fe2);
15 r1=fe2/fe1;
16 printf('Ratio of field energy stored in air gap to
       field energy stored in iron is %f \n',r1);
17 d1=fe1/(l*a);
18 printf('Energy density in iron is %f J/m^3\n',d1);
19 d2=fe2/(g*a);
20 printf('Energy density in air gap is %f J/m^3\n',d2)
       ;
21 r2=d2/d1;
22 printf('Ratio of energy density in air gap to energy
       density in iron is %f \n',r2);
```

106

Scilab code Exa 2.30 Determining 1 coil inductance field energy and force on armature 2 mechanical work done 3 magnetic force 4 mechanical work done for constant flux linkage

```scilab
1
2
3 clc;
4 n=1200; // number of turns in exciting coil
5 a=6*5*10^-4; // area of cross section of core
6 disp('case a');
7 x=0.01; // displacement of coil
8 i=2; // exciting current
9 uo=4*%pi*10^-7; // free space permeability
10 Wf=(n^2*uo*a*i^2)/(4*x);
11 printf('Field energy stored is %f J\n',Wf);
12 F=(-n^2*uo*a*i^2)/(4*x^2);
13 printf('Force on armature is %f N\n',F);
14 disp('case b');
15 x1=0.005; // further displacement of coil
16 v1=(n^2*uo*a*i)/(2*x); // flux linkage corresponding
       to displacement 1 cm
17 v2=(n^2*uo*a*i)/(2*x1); // flux linkage
      corresponding to displacement  0.5 cm
18 M=((v2-v1)*i)/2;
19 printf('Mechanical energy output is %f J\n',M);
20 disp('case c');
21 // after deriving expression
22 Wm=integrate('(-n^2*uo*a)/x^2','x',x,x1);
23 printf('Mechanical work done is %f J\n',Wm);
24 disp('case d');
25 // for x=0.005 flux linkage is constant than current
       will change
26 i2=v1/((2*n^2*uo*a)/(2*x));
27 Wm=((i-i2)*v1)/2;
```

107

```
28  printf('Mechanical work done if flux linkage are
        maintained constant is %f J\n',Wm);
29  disp('case e');
30  // after the expression is derived
31  Wm=integrate('(-v1^2)/(n^2*uo*a)','x',x,x1);
32  printf('Mechanical work done if flux linkage are
        maintained constant is %f J\n',Wm);
```

Chapter 3

Basic Concepts of Rotating Electrical Machines

Scilab code Exa 3.2 Determining rating of generator for given parallel paths

```
1  clc ;
2  n =24;  // Number of armature conductor
3  v =2;  // average voltage per conductor
4  i =5;  // current carrying capacity of each conductor
5  disp ( ' case a ' ) ;
6  a =2;  // number of parallel path
7  sc = n / a ;  // series connected conductor in each path
8  Ea = sc * v ;  // output voltage
9  Ia = i * a ;  // output current
10 p = Ea * Ia ;  // power rating
11 printf ( ' Generator rating is %f W\n ' ,p ) ;
12 disp ( ' case b ' ) ;
13 a =4;  // number of parallel path
14 sc = n / a ;  // series connected conductor in each path
15 Ea = sc * v ;  // output voltage
16 Ia = i * a ;  // output current
17 p = Ea * Ia ;  // power rating
18 printf ( ' Generator rating is %f W\n ' ,p ) ;
```

```
19  disp('case c');
20  a=6; // number of parallel path
21  sc=n/a; // series connected conductor in each path
22  Ea=sc*v; // output voltage
23  Ia=i*a; // output current
24  p=Ea*Ia; // power rating
25  printf('Generator rating is %f W\n',p);
```

Scilab code Exa 3.3 Calculating generated armature voltage for different types of machine

```
 1  clc;
 2  p=4; // number of poles
 3  s=60; // number of slots
 4  c=8; // number of conductors per slot
 5  f=20*10^-3; // flux per pole
 6  nr=1500; // relative speed in rpm between field flux
        and armature winding
 7  disp('case a');
 8  // winding is lap connected
 9  a=p; // for lap connected winding , number of
        parallel path=number of pole
10  z=s*c; // total number of conductors
11  n=nr/60; // speed in rps
12  E=(f*z*n*p)/a;
13  printf('Generated EMF is %f V\n',E);
14  disp('case b');
15  kw=0.96; // winding factor
16  nt=z/2; // Total number of turns
17  nph=nt/3; // number of series turns per phase
18  fg=(p*n)/2; // generated EMF frequency
19  E=sqrt(2)*%pi*fg*nph*kw*f;
20  printf('Generated EMF per phase is %f V\n',E);
21  e=round(sqrt(3)*E);
22  printf('Generated EMF between line terminal is %f V\
```

```
  n ' , e ) ;
```

Scilab code Exa 3.4 Determining 1 frequency of EMF 2 number of poles 3 number of synchronous motor poles

```
1  clc ;
2  p1 =4; // number of poles in slip ring induction
       motor
3  p2 =6; // number of poles in synchronous  motor
4  f =50; // frequency of supply
5  ns =(120* f )/ p2 ; // synchronous motor speed
6  ni =(120* f )/ p1 ; // induction motor speed
7  disp ( ' case a (1) ' ) ;
8  // when synchronous motor is driven in direction
       opposite to the rotating field produced the
       induction motor stator
9  nr = ns + ni ; // relative speed
10 F =( p1 * nr )/120;
11 printf ( ' Frequency of EMF at rotor slip ring
       terminals is %f Hz\n ' ,F ) ;
12 disp ( ' case a (2) ' ) ;
13 // when synchronous motor is driven in direction of
       the rotating field produced the induction motor
       stator
14 nr = ni - ns ; // relative speed
15 F =( p1 * nr )/120;
16 printf ( ' Frequency of EMF at rotor slip ring
       terminals is %f Hz\n ' ,F ) ;
17 disp ( ' case b ' ) ;
18 fn =150; // frequency of rotor terminal voltage
       required
19 // let new number of pole be pn then relative speed
       is  nr = ns +(120*50)/ pn ;
20 pn =(( fn *120) -(120* f ))/ ns ;
21 printf ( ' Number of poles that the induction motor
```

111

```
        must have is %f \n',pn);
22  disp ('case c');
23  pi =8; // number of poles in induction motor
24  ps =(120* f * pi )/(( fn *120) -(120* f ));
25  printf ('Number of synchronous motor poles required
        is %f',ps);
```

Scilab code Exa 3.5 Calculating frequency and magnitude of per phase EMF for different speed of rotor

```
1  clc;
2  p=4; // number of pole
3  f =50; // frequency of supply
4  ns =420; // stator turns
5  nr =240; // rotor turns
6  F=30*10^ -3; // flux per pole
7  kw =0.96; // winding factor for both stator and rotor
8  nsph = ns /3; // stator turn per phase
9  nrph = nr /3; // rotor turn per phase
10  es = sqrt (2) *% pi * f * kw * nsph * F; // stator turn per phase
11  disp ('case a');
12  // rotor is stationary
13  s =1; // at standstill slip =1
14  er = sqrt (2) *% pi * f * kw * nrph * F;
15  printf ('frequency of EMF in stator is %f Hz\n',f);
16  printf ('frequency of EMF in rotor is %f Hz\n',f);
17  printf ('Per phase stator EMF is %f V\n',es);
18  printf ('Per phase rotor EMF is %f V\n',er);
19  disp ('case b');
20  sr =1440; // speed of rotor in rpm in direction of
        rotating flux
21  Ns =(120* f )/p; // speed of rotating flux
22  s =( Ns - sr )/ Ns; // slip
23  fr = s * f; // frequency of EMF in rotor
24  er = sqrt (2) *% pi * fr * kw * nrph * F;
```

112

```
25  printf('frequency  of  EMF  in  stator  is  %f  Hz\n',f);
26  printf('frequency  of  EMF  in  rotor  is  %f  Hz\n',fr);
27  printf('Per  phase  stator  EMF  is  %f  V\n',es);
28  printf('Per  phase  rotor  EMF  is  %f  V\n',er);
29  disp('case  c');
30  sr=1440;  // speed  of  rotor  in  rpm  opposite  to  the
        direction  of  rotating  flux
31  s=(Ns+sr)/Ns;  // slip
32  fr=s*f;  // frequency  of  EMF  in  rotor
33  er=sqrt(2)*%pi*fr*kw*nrph*F;
34  printf('frequency  of  EMF  in  stator  is  %f  Hz\n',f);
35  printf('frequency  of  EMF  in  rotor  is  %f  Hz\n',fr);
36  printf('Per  phase  stator  EMF  is  %f  V\n',es);
37  printf('Per  phase  rotor  EMF  is  %f  V\n',er);
```

Scilab code Exa 3.6 Calculating fundamental third and fifth harmonic belt factors for stator

```
1   clc;
2   disp('case  a');
3   s=54;  // number  of  slots  in  stator ,3  phase
4   p=6;  // number  of  poles
5   spp=s/(3*p);  // slots  per  pole  per  phase
6   v=(p*180)/s;  // slot  angular  pitch
7   k1=sin(((spp*v)/2)*(%pi/180))/(spp*sin((v/2)*(%pi
        /180)));  // fundamental  harmonics
8   k3=sin(((3*spp*v)/2)*(%pi/180))/(spp*sin(((3*v)/2)*(
        %pi/180)));  // third  harmonic
9   k5=sin(((5*spp*v)/2)*(%pi/180))/(spp*sin(((5*v)/2)*(
        %pi/180)));  // fifth  harmonic
10  printf('First  harmonic  component  is  %f\n',k1);
11  printf('Third  harmonic  component  is  %f\n',k3);
12  printf('Fifth  harmonic  component  is  %f\n',k5);
13  disp('case  b');
14  s=48;  // number  of  slots  in  stator ,3  phase
```

```
15  p=6; // number of poles
16  spp=s/(3*p); // slots per pole per phase
17  sk=spp*3;
18  v=(p*180)/s; // slot angular pitch
19  ps=spp*v; // phase spread
20  k1=sin(((ps)/2)*(%pi/180))/(sk*sin(((ps)/(sk*2))*(
       %pi/180))); // fundamental harmonics
21  k3=sin(((3*ps)/2)*(%pi/180))/(sk*sin(((3*ps)/(sk*2))
       *(%pi/180))); // third harmonic
22  k5=sin(((5*ps)/2)*(%pi/180))/(sk*sin(((5*ps)/(sk*2))
       *(%pi/180))); // fifth harmonic
23  printf('First harmonic component is %f\n',k1);
24  printf('Third harmonic component is %f\n',k3);
25  printf('Fifth harmonic component is %f\n',k5);
```

Scilab code Exa 3.8 Determining distribution and winding factor

```
1   clc;
2   disp('case a');
3   cs=160*(%pi/180); // coil span in radian
4   ps=120*(%pi/180); // phase spread
5   kd=sin(ps)/(ps/2); // distribution factor for
       uniformly distributed winding
6   e=180-(cs*(180/%pi)); // chording angle
7   kp=cos((e/2)*(%pi/180)); // Coil span factor
8   wf=kd*kp; // winding factor
9   disp('Distribution factor is');
10  disp(kd);
11  disp('Winding factor is');
12  disp(wf);
13  disp('case b');
14  s=9; // number of slots per pole
15  sa=180/s; // slot angular pitch
16  // for a phase spread of 120 , 6*20=120, 6 adjacent
       slots must belong to the same phase, therefore
```

114

114

www.mepcafe.com


```
17  p=6; // poles belonging to same phase
18  kd=sin(ps/2)/(p*sin(ps/(2*6)));
19  wf=kd*kp;
20  disp('Distribution factor is');
21  disp(kd);
22  disp('Winding factor is');
23  disp(wf);
```

Scilab code Exa 3.10 Finding ratio of outputs and amount of copper required for different combinations

```
1  clc;
2  disp('case a');
3  f1=2/3; // fraction of slot wound
4  p1=f1*180; // phase spread , 2/3 of the slots are
        wound
5  kd1=sin((p1/2)*(%pi/180))/((p1/2)*(%pi/180)); //
        distribution factor
6  p2=180; // phase spread , All the slots are wound
7  f2=1; // fraction of slot wound
8  kd2=sin((p2/2)*(%pi/180))/((p2/2)*(%pi/180)); //
        distribution factor
9  // output is directly proportional to the product of
         fraction of slots wound and distribution factor
10 ro=(f2*kd2)/(f1*kd1); // It is assumed that
        frequency ,flux per pole and the conductor cross
        section is same
11 printf('Ratio of outputs is %f \n',ro);
12 rc=f2/f1;
13 printf('ratio of copper required is %f\n',rc);
14 disp('case b');
15 p3=60; // for 3-phase winding ,phase spread is 60
        degrees
16 kd3=sin((p3/2)*(%pi/180))/((p3/2)*(%pi/180)); //
        distribution factor
```

115

```
17 // since  all  the  slots  are  wound  for  both  1−phase
      and  3−phase ,  fraction  of  the  slots  wound  is  1
18 f3 =1;  // fraction  of  the  slots  wound
19 ro = kd3 / kd2 ;
20 printf ( ' Ratio  of  outputs  is  %f  \n ' ,ro ) ;
21 rc = f2 / f3 ;
22 printf ( ' ratio  of  copper  required  is  %f\n ' ,rc ) ;
23 disp ( ' case  c ' ) ;
24 f4 =1;  // fraction  of  the  slots  wound
25 p4 =90;  // for  2−phase  winding  ,phase  spread  is  90
      degrees
26 kd4 = sin (( p4 /2) *(% pi /180) ) /(( p4 /2) *(% pi /180) ) ;  //
      distribution  factor
27 ro = kd3 / kd4 ;
28 printf ( ' Ratio  of  outputs  is  %f  \n ' ,ro ) ;
29 rc = f3 / f4 ;
30 printf ( ' ratio  of  copper  required  is  %f\n ' ,rc ) ;
```

Scilab code Exa 3.11 1 Estimating resultant line and phase voltage for given configurations 2 Determining circulating current for delta connected machine

```
1 clc ;
2 d =0.28;  // air  gap  diameter
3 l =0.23;  // core  length  of  alternator
4 spp =4;  // slots  per  pole  per  phase
5 b1 =0.87;  // amplitude  of  flux  density  in  fundamental
       harmonic  in  Tesla
6 b3 =0.24;  // amplitude  of  flux  density  in  third
      harmonic  in  Tesla
7 b5 =0.14;  // amplitude  of  flux  density  in  fifth
      harmonic  in  Tesla
8 p =6;  // number  of  poles  in  alternator
9 np =3;  // number  of  phases
10 c =8;  // number  of  conductor  per  slot
```

116

```
11  f=50;  // frequency  of  supply
12  f1=(2*d*l*b1)/p;  // flux  for  fundamental  harmonic
13  f3=(2*d*l*b3)/(p*3);  // flux  for  third  harmonic
14  f5=(2*d*l*b5)/(p*5);  // flux  for  fifth  harmonic
15  ap=180/(spp*np);  // slot  angular  pitch
16  kd1=sin(((spp*ap)/2)*(%pi/180))/(spp*sin((ap/2)*(%pi
       /180)));  // distribution  factor  for  fundamental
       harmonic
17  kd3=sin(((3*spp*ap)/2)*(%pi/180))/(spp*sin(((3*ap)
       /2)*(%pi/180)));  // distribution  factor  for  third
        harmonic
18  kd5=sin(((5*spp*ap)/2)*(%pi/180))/(spp*sin(((5*ap)
       /2)*(%pi/180)));  // distribution  factor  for  fifth
        harmonic
19  // coil  is  short  pitched  by  one  slot ,  therefore
20  e=180/(spp*np);  // chording  angle
21  cs1=cos((e/2)*(%pi/180));  // coil  span  factor  for
       fundamental  harmonic
22  cs3=cos(((3*e)/2)*(%pi/180));  // coil  span  factor
       for  third  harmonic
23  cs5=cos(((5*e)/2)*(%pi/180));  // coil  span  factor
       for  fifth  harmonic
24  kw1=cs1*kd1;  // winding  factor  for  fundamental
       harmonic
25  kw3=cs3*kd3;  // winding  factor  for  third  harmonic
26  kw5=cs5*kd5;  // winding  factor  for  fifth  harmonic
27  ts=spp*np*p;  // total  number  of  slots
28  tt=(ts*c)/2;  // total  number  of  turns
29  nph=tt/np;  // series  turn  per  phase
30  ep1=sqrt(2)*%pi*f*kw1*nph*f1;  // emf  per  phase  for
       fundamental  harmonics
31  ep3=(ep1*kw3*3*f3)/(kw1*f1);  // emf  per  phase  for
       third  harmonics
32  ep5=(ep1*kw5*5*f5)/(kw1*f1);  // emf  per  phase  for
       fifth  harmonics
33  disp('case  a(1):  star  connected  alternator');
34  ep=sqrt(ep1^2+ep3^2+ep5^2);
35  printf('Resultant  EMF  per  phase  is  %f  V\n',ep);
```

117

```
36  // third frequency line emf doesnot appear in line
        voltage
37  el=sqrt(3)*sqrt(ep1^2+ep5^2);
38  printf('Resultant line voltage is %f V\n',el);
39  disp('case a(2): Delta connected alternator');
40  // third frequency line emf doesnot appear in line
        and phase voltage as they are short circuited  by
         closed delta
41  ep=sqrt(ep1^2+ep5^2);
42  printf('Resultant EMF per phase(also line voltage)
        is %f V\n',ep);
43  disp('case b: delta connected alternator ');
44  rpp=10; // reactance per phase
45  // emf to due first and third harmonic cancels each
        other but third harmonic gives rise to
        circulating current
46  I=(3*ep3)/(3*np*rpp);
47  printf('Circulating current is %f A',I);
```

Scilab code Exa 3.12 Calculating percentage increase in per phase rms emf due to harmonic

```
1  clc;
2  spp=3; // slots per pole per phase
3  np=3; // number of phases
4  cs=8; // coil span
5  fp=0.20; // fraction of third harmonic in flux
       density wave in air gap
6  sp=spp*np; // slots per pole
7  v=180/sp; // slot angular pitch
8  kd1=sin(((spp*v)/2)*(%pi/180))/(spp*sin((v/2)*(%pi
       /180))); // distribution factor
9  // for a coil span of 8 slots the coil is short
       pitched by one slot
10 e=v; // chording angle
```

118

```
11  kp1=cos((e/2)*(%pi/180)); // coil span factor
12  kw1=kp1*kd1; // winding factor
13  kd3=sin(((3*spp*v)/2)*(%pi/180))/(spp*sin(((v*3)/2)
      *(%pi/180))); // distribution factor for third
      harmonic
14  kp3=cos(((3*e)/2)*(%pi/180)); // coil span factor
      for third harmonic
15  kw3=kd3*kp3; //  winding factor for third harmonic
16  er=(kw3/kw1)*fp; // ratio of third harmonic emf to
      fundamental emf
17  ep=sqrt(1+er^2); // ratio of net emf to fundamental
      emf
18  pi=((ep-1)/1)*100;
19  printf('Percentage increase in per phase rms emf is
      %f percent ',pi);
```

Scilab code Exa 3.13 Determining phase and line EMF for given windings

```
1  clc;
2  p=6; // number of poles in alternator
3  s=42; // number of slots in alternator
4  f=0.012; // flux per pole
5  t=8; // number of turns in full pitch coil
6  F=50; // frequency of alternator
7  disp('case a');
8  np=2; // number of phases
9  spp=42/(p*np); // slots per pole per phase
10 // spp is not an integer, the 2-phase winding is a
      fractional slot winding, therefore Sk is given by
11 Sk=spp*2;
12 v=90; // phase spread for 2-phase winding
13 kd=sind(v/2)/(Sk*sind(v/(2*Sk))); // distribution
      factor
14 kw=kd; // winding factor as kp=1
15 nph=(s*t)/np; // per phase series turn
```

```
16  eph=sqrt(2)*F*%pi*kw*nph*f;
17  el=sqrt(2)*eph;
18  printf('Phase emf is %f V\n',eph);
19  printf('Line emf is %f V\n ',el);
20  disp('case b');
21  np=3; // number of phases
22  v=60; // phase spraed for 3-phase winding
23  kd=sind(v/2)/(Sk*sind(v/(2*Sk))); // distribution
       factor
24  kw=kd; // winding factor as kp=1
25  nph=(s*t)/np; // per phase series turn
26  eph=sqrt(2)*F*%pi*kw*nph*f;
27  el=sqrt(3)*eph;
28  printf('Phase emf is %f V\n',eph);
29  printf('Line emf is %f V\n ',el);
```

Scilab code Exa 3.14 Calculating third and fifth phase EMF in terms of fundamental phase emfs and the ratio of resultant line emf to resultant phase emf

```
1  clc;
2  s=81; // number of slots
3  p=6; // number of poles
4  np=3; // number of phases
5  cs=13; // coil span in terms of slot pitches
6  v=60; // phase spread for three phase winding
7  f3=0.4; // ratio of third harmonic flux to first
       harmonic flux
8  f5=0.25; // ratio of fifth harmonic flux to first
       harmonic flux
9  spp=s/(p*np); // // spp is not an integer, the 2-
       phase winding is a fractional slot winding,
       therefore Sk is given by
10  Sk=spp*2;
11  ap=(p*180)/s;
```

```
12 Cs=cs*ap; // coil span
13 e=180-Cs; // chording angle
14 kd1=sind(v/2)/(Sk*sind(v/(2*Sk))); // distribution
      factor for fundamental harmonic
15 kp1=cosd(e/2); // coil span factor
16 kd3=sind((3*v)/2)/(Sk*sind((3*v)/(2*Sk))); //
      distribution factor for third harmonic
17 kp3=cosd((3*e)/2); // coil span factor for third
      harmonic
18 kd5=sind((5*v)/2)/(Sk*sind((5*v)/(2*Sk))); //
      distribution factor for fifth harmonic
19 kp5=cosd((5*e)/2); // coil span factor for fifth
      harmonic
20 kw1=kd1*kp1; // winding factor for fundamental
      harmonics
21 kw3=kd3*kp3; // winding factor for third harmonic
22 kw5=kd5*kp5; // winding factor for fifth harmonic
23 ep3=(kw3*f3)/kw1;
24 printf('rms value of third harmonic emf is %f times
      the fundamental harmonic emf\n',ep3);
25 ep5=(kw5*f5)/kw1;
26 printf('rms value of fifth harmonic emf is %f times
      the fundamental harmonic emf\n',ep5);
27 ep=sqrt(1+ep3^2+ep5^2); // resultant phase emf
28 el=sqrt(3)*sqrt(1+ep5^2); // resultant line emf
29 r=el/ep;
30 printf('Ratio of resultant line emf to resultant
      phase emf is %f',r);
```

Scilab code Exa 3.15 Finding rms value of voltage in single turn coil

```
1 clc;
2 B=1; // peak flux density in Tesla
3 l=0.8; // length of armature conductor
4 v=20; // velocity of coil
```

```
5  // for 0< theta <30 coil aa' is moving in zero B--
       wave, emf for this range is zero
6  // for 30< theta < 60 coil side a is cutting through
        B--wave and coil side a' is cutting zero B--wave,
       therefore
7  e1=B*l*v; // emf at given position of coil
8  // for 60< theta < 150 both coil sides are cutting
       through B--wave
9  e2=2*B*l*v; // net emf at given position of coil
10 rms=sqrt((1/%pi)*(((e1^2*%pi*2)/6)+((e2^2*%pi)/2)));
11 printf('RMS value of generated emf in one single
       turn coil is %f V',rms);
```

Scilab code Exa 3.16 Calculating rms value of fundamental emf per phase

```
1  clc;
2  f=50; // frequency of alternator
3  B=1; // peak flux density
4  t=360; // total turns
5  v=60; // phase spread
6  pi=0.6; // pole pitch
7  l=0.8; // stator length
8  cs=180; // coil span in electrical degrees
9  nph=t/3; // series turn per phase
10 Bp=(4*B*cosd(v/2))/%pi; // fundamental value of peak
        flux density
11 F=(2*l*pi*Bp)/%pi; // Fundamental air--gap flux per
       pole
12 kd=sind(v/2)/((v/2)*(%pi/180)); // distribution
       factor
13 kw=kd; // winding factor , as kp=1
14 eph=sqrt(2)*%pi*f*F*kw*nph;
15 printf('RMS value of fundamental emf per phase is %f
        V',eph);
```

Scilab code Exa 3.18 Determining resultant emf for given combination of series coil

```
1  clc;
2  p=6; // number of poles
3  s=54; // number of slots
4  n=1000; // speed of alternator in rpm
5  t=80; // number of turn in coils A and B
6  f=0.015; // flux per pole
7  F=50; // given frequency of alternator
8  // Coil A is over pitched by one slot and coil B is
       short pitched by one slot
9  pp=s/p; // pole pitch
10 sap=(p*180)/s; // slot angular pitch
11 e1=(%pi*F*f*t)/sqrt(2); // EMF generated in one coil
        side of coil A or B
12 // same EMF is generated in col side 11 but with a
       phase of (180+sap) degrees. Resultant of emf in
       coil side 1 and 11 is given by
13 Ea=2*e1*cosd(sap/2); // net emf in coil side 1
14 Eb=Ea; // net emf in coil side 2
15 //Ea and Eb are in phase with each other from phasor
        diagram (fig. 3.26)
16 disp('case a');
17 en=Ea+Eb;
18 printf('Resultant e.m.f when coils A and B are
       connected in series aiding is %f V\n',en);
19 disp('case b');
20 en=Ea-Eb;
21 printf('Resultant e.m.f when coils A and B are
       connected in series opposing is %f V\n',en);
```

123

Scilab code Exa 3.19 Determining maximum and rms value of peak of fundamental mmf wave

```
1  clc;
2  np=3; // number of phases
3  p=2; // number of poles
4  spp=5; // slots per pole per phase
5  n=4; // number of turns in coil
6  i=20; // per phase current
7  v=(spp*180)/(spp*np); // phase spread
8  imax=sqrt(2)*i; // maximum value of current
9  mmf=spp*n*imax; // resultant amplitude of mmf
10 kd=sind(v/2)/((v/2)*(%pi/180)); // distribution
       factor
11 fp=(4*mmf*kd)/%pi; // peak value of fundamental
       component
12 fr=(4*3*spp*n*i)/%pi^2; // rms value of fundamental
       component
13 printf('Maximum value of the peak of fundamental m.m
       .f wave is %f AT/pole\n',fp);
14 printf('RMS value of the peak of fundamental m.m.f
       wave is %f AT/pole\n',fr);
```

Scilab code Exa 3.20 Calculating peak amplitude of mmf wave peak and rms value of fundamental mmf wave

```
1  clc;
2  p=2; // number of pole
3  i=24; // phase current
4  t=300; // full pitched turns
5  v=60; // phase spread
6  np=3; // number of phases
7  nph=t/np; // series turn per phase
8  j=(nph*sqrt(2)*i*180)/(v*%pi); // peak value of
       uniform current density
```

124

```
 9  disp('case a');
10  A=(j*v*%pi)/(2*180); // peak amplitude of
        trapezoidal m.m.f wave
11  printf('Peak amplitude of trapezoidal m.m.f wave is
        %f ATs/pole\n',A);
12  disp('case b');
13  kd=sind(v/2)/((v/2)*(%pi/180)); // distribution
        factor
14  fp=(4*kd*A)/%pi;
15  printf('Peak value of fundamental mmf wave is %f AT/
        pole\n',fp);
16  fr=(4*3*A)/(%pi^2*sqrt(2));
17  printf('RMS value of fundamental mmf wave is %f AT/
        pole\n',fr);
```

Scilab code Exa 3.30 Finding linear velocity of travelling mmf wave

```
1  clc;
2  p=6; // number of poles in induction motor
3  f=50; // frequency of motor
4  d=1.2; // stator bore diameter
5  // in one revolution peripheral distance of Pi*
        diameter is transversed
6  v=(2*f*%pi*d)/p;
7  printf('Linear velocity of travelling mmf wave is %f
        m/sec',v);
```

Scilab code Exa 3.32 Calculating resultant peak gap mmf peak gap flux density total gap energy electromagnetic torque and electromagnetic power

```
1  clc;
2  p=2; // number of poles
3  f=50; // frequency of machine
```

```
4  D=1.6; // diameter of cylindrical rotor
5  l=1.8; // length of cylindrical rotor
6  g=0.012; // air gap length
7  rm=4000; // peak value of rotor mmf
8  rs=6000; // peak value of stator mmf
9  ph=140; // phase difference between stator mmf and
      rotor mmf
10 uo=4*%pi*10^-7; // free space permeability0
11 disp('a');
12 rp=sqrt(rm^2+rs^2+2*rm*rs*cosd(ph));
13 printf('Resultant peak gap mmf is %f AT/pole\n',rp);
14 disp('b');
15 Bp=(uo*rp)/g;
16 printf('Peak gap flux density is %f T\n',Bp);
17 disp('c');
18 ge=(uo*%pi*D*l*rp^2)/(4*g);
19 printf('Total gap energy is %f Joules\n',ge);
20 disp('d');
21 T=(p*uo*%pi*D*l*rs*rm*sind(ph))/(4*g);
22 printf('Electromagnetic torque is %f Nm\n',T);
23 disp('e');
24 wm=(4*%pi*f)/2; // synchronous speed
25 P=(T*wm)/1000
26 printf('Electromagnetic power is %f KW',P);
```

Scilab code Exa 3.33 Determining torque for given pole fields

```
1  clc;
2  d=0.8; // diameter of rotor machine
3  l=0.5; // length of rotor machine
4  g=0.005; // air gap length
5  as=10000; // peak current density for stator
6  ar=6000; // peak current density for rotor
7  t=60; // torque angle
8  disp('case a');
```

126

```
 9  p=2; // number of pole
10  uo=4*%pi*10^-7; // free space permeability
11  Fs=(as*d)/p; // peak stator mmf per pole
12  Fr=(ar*d)/p; // peak rotor mmf per pole
13  Te=(p*uo*%pi*d*l*Fs*Fr*sind(t))/(4*g);
14  printf('Torque for given number of poles is %f Nm\n'
        ,Te);
15  disp('case b');
16  p=6; // number of pole
17  uo=4*%pi*10^-7; // free space permeability
18  Fs=(as*d)/p; // peak stator mmf per pole
19  Fr=(ar*d)/p; // peak rotor mmf per pole
20  Te=(p*uo*%pi*d*l*Fs*Fr*sind(t))/(4*g);
21  printf('Torque for given number of poles is %f Nm\n'
        ,Te);
```

Scilab code Exa 3.35 Finding peak value of fundamental mmf peak value of fundamental air gap flux density wave fundamental value of flux per pole and rms value of phase and line emfs at no load and rated speed

```
 1  clc;
 2  p=4; // number of poles
 3  np=3; // number of phases
 4  f=50; // frequency of alternator
 5  sap=8; // slot angular pitch
 6  c=12; // number of concentric coils in field winding
 7  tf=6; // turns per field coil
 8  ta=28; // series armature turn per phase
 9  ar=0.6; // armature radius
10  la=4; // armature length
11  g=0.06; // gap length
12  w=0.96; // winding factor for armature winding
13  fc=1000; // field current
14  disp('case a');
15  kd=sind((np*sap)/2)/(np*sind(sap/2)); //
```

```
      distribution factor
16 kp=1; // coil span factor
17 kf=kd*kp; // winding factor for field winding
18 nf=tf*c; // number of field turn
19 F=(4*kf*nf*fc)/(%pi*p);
20 printf('Peak value of fundamental mmf produced by
      field winding is %f AT/pole\n',F);
21 disp('case b');
22 uo=4*%pi*10^-7; // free space permeability
23 B=(uo*F)/g;
24 printf('Peak value of fundamental flux density wave
      is %f T\n',B);
25 disp('case c');
26 v=(4*B*la*ar)/p;
27 printf('Fundamental value of air gap flux per pole
      is %f W\n',v);
28 disp('case d');
29 eph=sqrt(2)*%pi*f*v*ta*w;
30 printf('EMF per phase is %f V\n',eph);
31 el=sqrt(3)*round(eph);
32 printf('Line EMF is %f V',el);
```

Scilab code Exa 3.36 Calculating 1 peak value of fundamental air gap flux density 2 peak value of fundamental mmf wave 3 peak value of armature mmf wave and resultant mmf wave per pole 4 rms value of armature current and its power factor

```
1 clc;
2 np=3; // number of phases
3 p=6; // number of poles
4 f=50; // frequency of alternator
5 e=415; // open circuit emf;
6 s=36; // number of slots in armature
7 t=4; // number of turns per coil
8 g=0.18; // air gap diameter
```

128

```
 9  l=0.4; // core length
10  G=0.002; // gap length
11  T=42; // number of turns in field winding
12  kf=0.96; // winding factor
13  uo=4*%pi*10^-7; // free space permeability
14  disp('case a');
15  nph=(s*t)/np; // series turn per phase
16  spp=s/(p*np); // slots per pole per phase
17  v=180/p; // slot angular pitch
18  kd=sind((spp*v)/2)/(spp*sind(v/2)); // distribution
        factor
19  Flu=e/(sqrt(2)*sqrt(3)*%pi*f*nph*kd); // flux per
        pole
20  B=(p*Flu*2)/(4*l*g);
21  printf('Peak value of fundamental flux density wave
         is %f T\n',B);
22  disp('case b');
23  Fl=(G*B)/uo; // peak fundamental field mmf wave
24  printf('Peak value of fundamental mmf wave is %f AT/
        pole\n',Fl);
25  If=(%pi*Fl*p)/(4*kf*T);
26  printf('DC field current is %f A\n',If);
27  disp('case c');
28  Te=114; // given torque
29  Ta=146; // torque angle
30  Fm=floor((Te*4*G)/(p*uo*%pi*g*l*Fl*sind(Ta)));
31  printf('Peak value of fundamental armature mmf is %f
         AT/pole\n',Fm);
32  Fr=sqrt(Fl^2+Fm^2+2*Fl*Fm*cosd(Ta));
33  printf('Resultant mmf per pole is %f AT/pole\n',Fr);
34  disp('case d')
35  ia=(Fm*2*%pi*p)/(12*kd*nph*sqrt(2));
36  printf('RMS value of armature current is %f A\n',ia)
        ;
37  ns=1000; // speed in rpm
38  wm=(2*%pi*ns)/60; // angular speed in rps
39  pf=(Te*wm)/(sqrt(3)*e*ia);
40  printf('Power factor is %f lagging',pf);
```

129

Scilab code Exa 3.37 Determining full load efficiency of given transformer

```
1 clc;
2 n1=0.95; // efficiency of transformer 1
3 lo=((1/n1)-1); // fraction of output lost
4 d=2; // Linear dimension of transformer B is two
     times the Linear dimension of transformer A
5 nb=(1/(1+((1*lo)/d)))*100;
6 printf('Full load efficiency of transformer B is %f
     percent',nb);
```

Scilab code Exa 3.39 Determining KW rating of motor

```
1 clc;
2 t0=0; // accelerating period
3 t1=30; // decelerating period
4 l1=2000; // maximum load during accelerating period
5 lf=600; // maximum load during decelerating period
6 l=1000; // load during full load
7 tf=60; // full load duration
8 td=10; // decelerating duration
9 tde=20; // decting period
10 sa=l1/t1; // slope during accelerating
11 sd=lf/td; // slope during decelerating
12 e1=integrate('(sa*t)^2','t',t0,t1); // term 1 for
     finding motor rating
13 e2=l^2*tf; // term 2 for finding motor rating
14 e3=integrate('(sd*t)^2','t',t0,td); // term 3 for
     finding motor rating
15 T=t1+tf+td+tde; // total duration
16 R=sqrt((1/120)*(e1+e2+e3));
```

```
17 printf('KW rating of motor is %f KW',R);
18 disp('Choose a motor of rating above the calculated
      rating');
```

Scilab code Exa 3.40 Determining continuous KW rating of motor

```
1 clc;
2 T=80; // total duration
3 t1=5-0; // duration of first increasing loading
      period
4 t2=36-5; // duration of second increasing loading
      period
5 t3=39-36; // duration of first decreasing loading
      period
6 t4=55-39; // duration of second decreasing loading
      period
7 t5=80-55; // duration of uniform loading
8 l1=150; // initial load
9 l2=1000; // load at 5th sec
10 l3=1400; // load at 36th sec
11 l4=300; // load at 39th sec
12 l5=150; // load during uniform loading
13 T1=(t1/3)*(l1^2+l2^2+l1*l2); // term 1 for
      evaluating rms power
14 T2=(t2/3)*(l2^2+l3^2+l2*l3); // term 2 for
      evaluating rms power
15 T3=(t3/3)*(l3^2+l4^2+l3*l4); // term 3 for
      evaluating rms power
16 T4=(t4/3)*(l4^2+l5^2+l4*l5); // term 4 for
      evaluating rms power
17 T5=t5*l5^2; // term 5 for evaluating rms power
18 R=sqrt((1/T)*(T1+T2+T3+T4+T5));
19 printf('As per the load time graph rating is %f KW',
      R);
20 disp('Choose a motor of rating above the calculated
```

131

```
    rating ');
```

Scilab code Exa 3.41 Determining final steady temperature rise and new KVA rating of transformer

```
1  clc;
2  p=200;  // rated KVA of transformer
3  n=0.98;  // efficiency
4  t1=20;  // temperature after one hour of operation
5  t2=34;  // temperature after two hour of operation
6  r=1/3;  // ratio of full load core losses to ohmic
       loss
7  disp('case a');
8  t=[(t2/t1)-1];
9  th=-1/log(t);  // heating time constant in hours
10 theta=t1/(1-exp(-1/th));
11 printf('Final steady temperature rise of the
       transformer on rated load is %f degree celsius\n'
       ,theta);
12 disp('case b');
13 f=1.2;  //with increased heat dissipation ,ratio of
       new loss to old loss
14 Pn=sqrt((f*(1+r))-r)*p;
15 printf('New KVA rating of transformer is %f KVA\n',
       Pn);
16 // for a temperature rise of 78 degree
17 t3=78;
18 f=(t3/theta)*f;  // ratio of new loss to old loss
19 Pn=sqrt((f*(1+r))-r)*p;
20 printf('New KVA rating of transformer is %f KVA\n',
       Pn);
```

Scilab code Exa 3.42 Determining one hour rating of induction motor

132

```scilab
1  clc ;
2  p=100;  // KW rating of transformer
3  al =1;  // ratio of core loss to ohmic loss
4  th =3;  // heating time constant in hours
5  h=1;  // duration in hour for which KVA rating has to
       be determined
6  disp ( 'case a ' ) ;
7  // constant losses are equal to variable losses
8  pn=p* sqrt (((1+al)/(1- exp (-h/th)))-al);
9  printf ( 'One hour rating is %f KW\n ' ,pn ) ;
10 disp ( 'case b ' ) ;
11 // consatant losses are neglected
12 al =0;  // ratio of core loss to ohmic loss
13 pn=p* sqrt (((1+al)/(1- exp (-h/th)))-al);
14 printf ( 'One hour rating is %f KW\n ' ,pn ) ;
```

Scilab code Exa 3.43 Calculating ratio of core loss to ohmic loss

```scilab
1  clc ;
2  t =1/2;  // ratio of continuous rating to one hour
     rating
3  p=2;  // ratio of new KVA rating to old KVA rating
4  al =2*( p*t ) ;
5  printf ( 'Ratio of core loss to ohmic loss is %f ' ,al)
     ;
```

Chapter 4

DC machines

Scilab code Exa 4.2 Determining electromagnetic power and internal torque

```
1  clc;
2  l=0.3;//core length
3  r=0.2;//radius
4  n=20;//speed in r.p.s.
5  Ia=20;//armature current
6  Z=500;//total conductors
7  Bav=0.5;//avg. flux density
8  a=4;//lap-wound
9  P=4;//no of poles
10 Wm=2*%pi*n
11 phi=((0.5*2*%pi*0.2*0.3)/4);
12 Ea=((P*n*Z*phi)/a);//generated emf
13 Pm=Ea*Ia;//gross mechanical power developed
14 Te=((Ea*Ia)/Wm);//internal torque
15 printf('Generated E.M.F. is %f V.\n',Ea);
16 printf('Gross mechanical power developed is %f W.\n'
       ,Pm);
17 printf('Internal Torque is %f Nm.',Te);
```

Scilab code Exa 4.3 Determining total current emf power developed in armature and electromagnetic torque

```
1  clc;
2  P=6;//no of poles
3  Z=300;//no of conductors
4  phi=0.015;//flux per pole in webers
5  n=30;//speed in r.p.s.
6  Ic=80;//current per conductor
7  Wm=2*%pi*n;
8  Eav=P*n*phi;//avg. emf per conductor
9  //when conductors are wave connected
10 disp('Wave Connected')
11 a1=2;//no of parallel paths
12 Ia=Ic*a1;//total current
13 Ea=Eav*(Z/a1);//E.M.F.
14 Pa=Ea*Ia;//power developed in armature
15 Te=Ea*Ia/Wm;//Electromagnetic torque
16 printf('Generated E.M.F. is %f V.\n',Ea);
17 printf('Power developed in armature is %f W.\n',Pa);
18 printf('Electromagnetic Torque is %f Nm.\n',Te);
19 //when conductors are lap connected
20 disp('Lap Connected')
21 a2=4;//no of parallel paths
22 Ia2=Ic*a2;//total current
23 Ea2=Eav*(Z/a2);//E.M.F.
24 Pa2=Ea2*Ia2;//power developed in armature
25 Te2=Ea2*Ia2/Wm;//Electromagnetic torque
26 printf('Generated E.M.F. is %f V.\n',Ea2);
27 printf('Power developed in armature is %f W.\n',Pa2)
      ;
28 printf('Electromagnetic Torque is %f Nm.',Te2);
```

Scilab code Exa 4.4 Determining 1 generated emf at no load 2 terminal voltage at full load

```
1  clc;
2  p=6; // number of poles
3  c=240; // number of coils
4  t=2; // number of turns per coil
5  rt=0.03; // resistance of one turn
6  l=0.5; // length of armature
7  d=0.4; // diameter of armature
8  B=0.6; // air gap flux density
9  a=p; // number of parallel paths is same as number
       of poles foe lap winding
10 an=40; // mechanical angle subtended by pole
11 n=1200; //  armature speed
12 th=an*(p/2); // electrical angle subtended by pole
13 f=(2*%pi*(d/2)*l*th*B)/(p*180); // flux per pole
14 Z=2*c*t; // total conductors
15 disp('case a');
16 Ea=(f*Z*n*p)/(60*a);
17 printf('Generated EMF at no load is %f V\n',ceil(Ea)
       );
18 disp('case b');
19 ia=40; // armature current
20 at=(c*t)/a; // number of armature turns per path
21 r=at*rt; // resistance of one path
22 Ra=r/a; // resistance of armature circuit
23 vt=Ea-ia*Ra;
24 printf('Terminal voltage at full load is %f V\n',
       ceil(vt));
```

Scilab code Exa 4.5 Determining terminal voltage of generator

```
1  clc
2  Pout=24000;//rated output power in watts
3  Et=250;//rated terminal voltage
4  Ra=0.1;//armature resistance
5  N=1600;//speed in rpm
```

136

www.mepcafe.com

```
6  //Ea( terminal voltage)= k*(N*phi),where k is
       constant & phi is flux per pole
7  //At no load, 260=k*1600*phi ....(1)
8  Ia=Pout/Et;
9  //if the generated voltage under rated load is Ea1,
       then
10 //Ea=k*1500*phi ....(2)
11 //From equation (1)&(2), (Ea1/260)=((1500*phi)
       /(1600*phi))
12 Ea1=(1500*260)/1600;
13 Vt=Ea1-Ia*Ra//terminal voltage at rated load
14 printf('The terminal voltage of generator under
       given conditions is %f V.',Vt)
```

Scilab code Exa 4.6 Determining ratio of speed as a generator to speed as a motor

```
1  clc;
2  Vt=230;//terminal voltage of dc shunt machine
3  Il=40;//Line current
4  Ra=0.5;//Armature circuit resistance
5  Rf=115;//Field circuit resistance
6  //GENERATOR OPERATION
7  If=Vt/Rf;//field current
8  Ia1=If+Il;//armature current
9  Ea1=Vt+Ia1*Ra;//generated emf
10 //Ea1=k*Ng*phi ....(1), where Ng is the generator
       speed & phi is flux per pole proportional to If
11 //MOTOR OPERATION
12 Ia2=Il-If;//armature current
13 Ea2=Vt-Ia2*Ra;//generated emf
14 //Ea2=k*Nm*phi ....(2), where Nm is the motor speed
       & phi is flux per pole proportional to If
15 //From equation (1)&(2), (Ea1/Ea2)=((Ng*phi)/(Nm*phi
       ))
```

137

```
16  N = Ea1 / Ea2 ; // ratio of speed of generator to motor ,N=
        Ng/Nm
17  printf ( 'The ratio of speed as a generator to the
        speed as a motor is %f. ' , N )
```

Scilab code Exa 4.9 Determining demagnetizing and cross magnetizing ampere turns per pole for different positions of brushes

```
1  clc ;
2  A =2; // No of parallel paths for armature conductors
3  P =6; // No. of poles
4  If =2; // Field current
5  Il =148; // Line current
6  Ia = If + Il ; // Armature current
7  Z =480; // No of conductors
8  // brushes on GNA, theta=0
9  ATd1 =0 // demagnetizing ampere turns
10 ATc1 =(( Ia * Z ) /(2* A * P ) ) // Cross magnetizing ampere
        turns
11 printf ( 'When brushes are on GNA the demagnetizing
        ampere turns & Cross magnetizing ampere turns are
         equal to %f & %f ATs/pole respectively . \ n ' , ATd1 ,
        ATc1 ) ;
12 // brushes are shifted from GNA by 5 degrees
        electrical , theta=5
13 theta =5;
14 ATd2 =((2* theta * Ia * Z ) /(180*2* A * P ) ) // demagnetizing
        ampere turns
15 ATc2 =3000 - ATd2 ; // Cross magnetizing ampere turns
16 printf ( 'When the brushes are shifted from GNA by 5
        degrees electrical the demagnetizing ampere turns
         & Cross magnetizing ampere turns are equal to %f
        & %f ATs/pole respectively . \ n ' , ATd2 , ATc2 ) ;
17 // brushes are shifted from GNA by 5 degrees
        mechanical , theta_m=5
```

138

```
18 theta_m=5;//mechanical angle
19 theta_e=(P/2)*theta_m;//electrical angle
20 ATd3=((2*theta_e*Ia*Z)/(180*2*A*P))//demagnetizing
      ampere turns
21 ATc3=3000-ATd3;//Cross magnetizing ampere turns
22 printf('When the brushes are shifted from GNA by 5
      degrees mechnical the demagnetizing ampere turns
      & Cross magnetizing ampere turns are equal to %f
      & %f ATs/pole respectively',ATd3,ATc3);
```

Scilab code Exa 4.11 Determining number of turns required on each pole

```
1 clc;
2 P=4;//No of poles
3 Pout=100000;//Output power in watts
4 Vt=200;//terminal voltage
5 Z=256;//No of conductors
6 A=4;//no of parallel paths of armature conductors
7 Ia=Pout/Vt;//armature current
8 Bcp=0.25;//interpolar flux density in tesla
9 gcp=0.01;//interpolar air gap length
10 U=4*%pi*0.0000001;//permeability of air
11 Fcp=((Ia*Z)/(2*A*P))+((Bcp/U)*(gcp));//The
      interpolar m.m.f. per pole
12 Ncp=Fcp/Ia;
13 printf('The turns on each interpoles should be equal
       to %f.',round(Ncp));
```

Scilab code Exa 4.12 Determining time of commutation

```
1 clc;
2 D=50;//diameter of commutator
3 N=1000;//speed of rotation of commutator in rpm
```

139

```
4  Wb=1.5;//brush width
5  V=%pi*D*N/60;//peripheral velocity of commutator
6  Tc=(Wb*1000)/V;//time of commutation in ms
7  printf('Time of commutation is %f ms.',Tc);
```

Scilab code Exa 4.13 Determining value of commutating field

```
1  //The answer given in book for this question is
       wrong.
2
3  clc;
4  P=4;//No of poles
5  Ia=120;//armature current
6  A=4;//No of parallel paths for armature conductor
7  L=0.02//inductance in mH
8  //Et=L*(di/dt),Transformer emf in coil
9  //di=2*Ia/A,change of current during commutation
10 //dt=Tc,time of commutation
11 //Et=0.02*0.001*(60/Tc) ....(1)
12 //Er=2*(Bav*l*v),rotational emf in single turn coil
13 //Er=2*(phi_c/Tc) ....(2),phi_c is the avg value of
       flux in the commutating zone
14 //For linear commutation, Er=Et, from equation (1)
       &(2)
15 phi_c=60*0.02*0.001/2;//phi_c is the avg value of
       flux in the commutating zone
16 printf('THE AVG. VALUE OF FLUX IN THE COMMUTATING
       ZONE IS %f Wb.',phi_c)
```

Scilab code Exa 4.14 Determining number of pole face conductors of compensating winding in each pole

```
1  clc;
```

140

```
2  Pout =2000000;//output power in watts
3  Vt=400;//output voltage
4  P=14;//No of poles
5  A=14;//No of parallel paths of conductor
6  Pr=0.7;//pole arc to pole pitch ratio
7  Z=1100;//total armature conductors
8  Ia=Pout/Vt;//armature current
9  A_z=(Ia*Z)/(A*P);//armature ampere conductors per
      pole
10 A_z1=Pr*A_z;//armature ampere conductors per pole to
       be compensated by pole face winding
11 //The compensating winding carries the entire
      armature current of 5000 A.
12 Wc=A_z1/Ia;//compensating winding conductors per
      pole
13 printf('Total compensating winding conductors per
      pole are %f.',round(Wc));
```
_____

Scilab code Exa 4.15 Determining 1 compensating winding conductors per pole 2 number of turns on each interpole

```
1  clc;
2  ATp=15000;//armture ampere turns per pole
3  Pr=0.68;//ratio of pole arc to pole pitch
4  Ia=850;//rated armature current
5  Bcp=0.25;//interpolar flux density in tesla
6  gcp=0.01;//interpolar air gap length
7  U=4*%pi*0.0000001;//permeability of air
8  ATc=Pr*ATp;//compensating winding ampere turns per
      pole
9  C=2*(ATc/Ia);//compensating winding conductors per
      pole
10 MMF_ag=(Bcp/U)*gcp;//M.M.F. required for the air gap
       under the interpole
11 MMF=MMF_ag+ATp;//interpole M.M.F. without
```

```
         compensating  winding
12  MMF_c=MMF-ATc;//ampere  turns  furnished  by  each
        interpole
13  N=MMF_c/Ia;//No.  of  turns  on  each  interpole
14  printf('Number  of  turns  on  each  interpole  is  %f.',
        round(N));
```

Scilab code Exa 4.16 Determining number of series field turns per pole

```
1  clc;
2  Pout=10000;//output  power  of  dc  generator  in  watts
3  Vt=250;//terminal  voltage  in  volts
4  If=2;//field  current  in  ampere  at  no  load
5  If1=2.2;//field  current  in  ampere  at  rated  load
6  Tp=1400;//turns  on  each  pole
7  Ia=Pout/Vt;//armature  current
8  MMF_rl=If1*Tp;//M.M.F.  required  at  rated  load
9  MMF_nl=If*Tp;//M.M.F.  required  at  no  load
10  MMF_s=MMF_rl-MMF_nl;//M.M.F.  supplied  by  series
        winding
11  Is=Ia;//series  current  at  full  load
12  Ts=MMF_s/Is;//series  field  turns
13  printf('Series  field  turns  are  equal  to  %f.',Ts);
```

Scilab code Exa 4.17 Determining demagnetizing effect of armature reaction at rated load and speed

```
1  clc;
2  //  table  is  given  in  question  for  plotting
        magnetising  curve
3  if1=[ 0  0.2  0.4  0.6  1  1.4  1.8  2 ];
4  Ea=[ 6  40  80  120  194  246  269  274];
5  plot(if1,Ea);
```

142

```
6  xlabel('If');
7  ylabel('Ea');
8  title('magnetising curve')
9  v=230; // rated voltage of generator
10 p=10000; // rated power of generator
11 n=1500; // rated speed of generator
12 rf=184; // shunt field resistance
13 ra=0.443; // armature resistance
14 ifl=1.7; // rated field current
15 il=p/v; // full load current
16 printf('Total armature current is %f A\n',il+ifl);
17 printf('Armature resistance drop is %f ohms\n',(il+
       ifl)*ra);
18 disp('In fig 4.17(textbook),AB is made equal to
       armature resistance drop then through B a
       horizontal line is made meeting curve at c');
19 disp('Demagnetising effect is given by BC which is
       equal to 0.25 A');
```

Scilab code Exa 4.18 Determining terminal voltage at given speed

```
1  clc;
2  vt1=50; // terminal voltage
3  rf=100; // resistance of field circuit
4  n1=1000; // speed corresponding to vt1=50
5  vt2=225; // terminal voltage
6  n2=2000; // speed corresponding to vt2=225
7  vt3=405; // terminal voltage
8  n3=3000; // speed corresponding to vt3=405
9  disp('case a');
10 ifl1=vt1/rf; // field current for n=1000 rpm
11 ifl2=vt2/rf; // field current for n=2000 rpm
12 ifl3=vt3/rf; // field current for n=3000 rpm
13 printf('Field current for speed=%f rpm is %f A\n',n1
       ,ifl1);
```

```
14  printf('Field current for speed=%f rpm is %f A\n',n2
        ,ifl2);
15  printf('Field current for speed=%f rpm is %f A\n',n3
        ,ifl3);
16  vt11=vt1*(n2/n1);
17  printf('Terminal voltage=%f V at %f rpm is
        equivalent to %f V at %f rpm\n',vt1,n1,vt11,n2);
18  vt33=vt3*(n2/n3);
19  printf('Terminal voltage=%f V at %f rpm is
        equivalent to %f V at %f rpm\n',vt3,n3,vt33,n2);
20  disp('Using above data, magnetising curve is drawn
        for n=2000 rpm');
21  // from fig 4.37
22  disp('For field resistance=80 ohms terminal voltage
        is given by BC');
23  disp('BC=253, hence terminal voltage corresponding
        to field resistance of 80 ohms is 253 V');
24  disp('For field resistance=70 ohms terminal voltage
        is given by QP');
25  disp('QP=268, hence terminal voltage corresponding
        to field resistance of 70 ohms is 268 V');
```

Scilab code Exa 4.19 Determining 1 open circuit voltage 2 critical value of shunt field resistance 3 critical speed 4 open circuit voltage for given field resistance 5 terminal voltage

```
1  clc;
2  n=1500; // speed of generator
3  // data is given in question for magnetising curve
       at n=1500 rpm
4  If=[ 0 0.4 0.8 1.2 1.6 2 2.4 2.8 3];
5  Ea=[6 60 120 172.5 202.5 221 231 237 240];
6  subplot(221);
7  plot(If,Ea);
8  xlabel('field current');
```

144

```
 9  ylabel('generated EMF');
10  title('Magnetising curve for n=1500');
11  disp('case a')
12  rf=100; // field resistance
13  // rf=100 lets say voltage=240 and field current=2.4
       which is shown by point A, straight line passing
       through A and origin meets magnetising current
       at B which is no load voltage
14  Eo=230;
15  printf('No load voltage is %f V\n',Eo);
16  disp('case b');
17  // a line OF is drawn passing through origin slope
       of this line gives critical resistance
18  vt=180; // terminal voltage
19  ifl=1.2; // field current corresponding to terminal
       voltage
20  rfl=vt/ifl;
21  printf('Critical value of shunt field resistance is
       %f ohms\n',rfl);
22  disp('case c');
23  // Choose S (any point) on linear part of
       magnetising curve.A vertical line from S meets
       field resistance line at t and horizontal line at
        y. Now
24  e1=90; // terminal voltage corresponding to point s
25  e2=60; // terminal voltage corresponding to point t
26  n2=(e2/e1)*n;
27  printf('Critical speed for given shunt field
       resistance is %f rpm\n',n2);
28  disp('case d');
29  n3=1200; // speed at which magnetising curve is
       drawn
30  // data for magnetising curve at n=1200 can be
       obtained by multiplying voltage of magnetising
       curve at n=1500 by factor 1200/1500 and at point
       C field resistance line for 100 ohms meet at
       magnetising curve .This point gives no load EMF
31  EAn=Ea*(n3/n);
```

145

```
32  subplot(222);
33  plot(If,EAn);
34  xlabel('field current');
35  ylabel('generated EMF');
36  title('Magnetising curve for n=1200');
37  Eo=165;
38  printf('No load EMF is %f V\n',Eo);
39  disp('case e');
40  ia=50; // armature current
41  ra=0.3; // armature resistance
42  vd=ia*ra; // armature resistance drop
43  // To obtain terminal voltage cut OD equal to vd and
        draw DG parallel to field resistance line. From
      G draw vertical line  meeting field resistance
      line at H. Point corresponding to H gives
      terminal voltage which is
44  vt=207;
45  printf('Terminal voltage is %f V\n',vt);
```

Scilab code Exa 4.20 Determining 1 no load emf 2 output current and shunt field current 3 maximum output current and terminal voltage 4 steady state short circuit current 5 additional resistance that must be inserted in field circuit

```
1  clc;
2  ra=0.5; // armature resistance
3  rf=180; // shunt field resistance
4  n=1100; // speed at which generator is being driven
5  n1=1000; // speed for which data is given
6  disp('case a');
7  // from the data given in question magnetising curve
        is drawn (fig 4.46)
8  If=[ 0 0.2 0.4 0.6 0.8 1 1.2 1.4 ];
9  Ea=[5 50 100 140 170 190 200 205];
10  Ean=(n/n1)*Ea
```

146

```
11  plot(If,Ean);
12  xlabel('field current');
13  ylabel('generated EMF');
14  title('Magnetising curve for n=1100');
15  // line corresponding to rf=180 ohms to meet
        saturation curve at 221 V which is no load EMF
16  Eo=221;
17  printf('No load EMF is %f V\n',Eo);
18  disp('case b');
19  vt=190; // terminal voltage
20  // from curve armature resistance drop is given by
        line BC
21  vd=22.5; // armature resistance drop
22  ia=vd/ra; // armature current
23  ifl=vt/rf; // field current
24  printf('Shunt field current is %f A\n',ifl);
25  printf('Output current is %f A\n',ia-ifl);
26  disp('case c');
27  // OP represents maximum armature resistance drop i.
        e OP=46.5 V
28  vd=46.5;
29  ia=vd/ra; // armature resistance
30  // tangent point at R gives field current which is
31  ifl=0.635;
32  printf('Maximum output current is %f A',ia-ifl);
33  disp('case d');
34  // under steady state short circuit terminal voltage
        =0 V and residual flux EMF is
35  E=5.5; // residual flux EMF
36  printf('Steady state short circuit current is %f A\n
        ',E/ra);
37  disp('case e');
38  Eo=210; // no load voltage
39  // for Eo OD represents field resistance field
        current is 1.015
40  ifl=1.015; // field current
41  rfn=Eo/ifl; // field resistance
42  printf('Additional resistance required is %f ohms\n'
```

```
      ,rfn-rf);
43  disp('case f');
44  rf=150; // shunt field resistance
45  vt=180; // terminal voltage
46  p=0.04; // reduction in flux due to armature
        reaction
47  ifl=vt/rf; // field current
48  Ea=220*(1-p); // generated voltage
49  ia=(Ea-vt)/ra; // armature current
50  il=ia-ifl; // load current
51  printf('Load power is %f KW',(vt*il)/1000);
```

Scilab code Exa 4.21 Determining range of external rheostat and dissipating power

```
 1  clc;
 2  Vrated=30;//rated output voltage of generater
 3  Irated=200;//rated output current of generator
 4  Ra=0.03;//armature resistance(including brushes)
 5  Rf=2.4;//field winding resistance
 6  //No-load saturation curve at 2200rpm
 7  If=[2 4 6 8 10 12];
 8  Ea=[15 27 35 40 43 45];
 9  plot(If,Ea);//magnetization curve at 2200 rpm
10
11
12  //(1)AT 2200 rpm
13  //at no load
14  Ea1=28;//induced voltage in armature
15  //for this voltage, the field current required, from
        magnetization curve is-
16  If1=4.23;//field current in ampere
17  Rt1=Ea1/If1;//total shunt field resistance
18  Re=Rt1-Rf;//external resistance
19
```

```
20  //at full load
21  Ea1_=28+Irated*Ra;//induced voltage in armature
22  //for this voltage, the field current required, from
        magnetization curve is−
23  If1_=5.67;//field current
24  Rt1_=Ea1/If1_;//total shunt field resistance
25  Re_=Rt1_-Rf;//external resistance
26
27
28  //(2)AT 4500 rpm
29  //at no load
30  Ea2__=28;//induced voltage in armature at 4500 rpm
31  Ea2=28*(2200/4500);//Ea at 2200 rpm
32  //for this voltage, the field current required, from
        magnetization curve is−
33  If2=1.833;//field current in ampere
34  Rt2=Ea2__/If2;//total shunt field resistance
35  Re__=Rt2-Rf;//external resistance
36
37  //at full load
38  Ea2___=28+Irated*Ra;//induced voltage in armature at
        4500 rpm
39  Ea2_=34*(2200/4500);//Ea at 4500 rpm
40  //for this voltage, the field current required, from
        magnetization curve is−
41  If2_=2.17;//field current
42  Rt2_=Ea2__/If2_;//total shunt field resistance
43  Re___=Rt2_-Rf;//external resistance
44  Pmax=If1_*If1_*min(Re,Re_,Re__,Re___);
45  printf('The minimum & maximum value of external
        resistance is %f & %fohm respectively.\nMaximum
        power dissipated through rheostat is %f ohm.',min
        (Re,Re_,Re__,Re___),max(Re,Re_,Re__,Re___),Pmax);
```

Scilab code Exa 4.22 Determining field circuit resistance and flux per pole

```
1  clc;
2  Vt=100;//terminnal voltage
3  P=2;//no of poles
4  Z=1000;//no of conductors
5  A=2;//no of parallel paths for armature conductors
6  Ra_=2*10e-3;//resistance of each armature
7  Ra=500*Ra_*(1/2);//total armature resistance
8  //Let If be field current
9  //Ea=Vt+(Il+If)*0.5
10 //Ea1=100+(10+If)*0.5,because at 1055 rpm Il=10.
11 //Ea2=100+(20+If)*0.5,because at 1105 rpm Il=20.
12 //But, Ea=k1*If*speed
13 //Therefore,((If*1055)/(If*1105))=((100+(10+If)*0.5)
        /(100+(20+If)*0.5)),which gives−
14 If=1;//field current
15 Ea1=100+(10+1)*0.5;//at 1055 rpm
16 N=1055;//speed of rotor
17 phi=(Ea1*60*A)/(Z*N*P);
18 Rf=Vt/If;//field circuit resistance
19 printf('Field circuit resistance is %f ohm.\n',Rf);
20 printf('Flux per pole is %f Wb.',phi);
```

Scilab code Exa 4.23 Determining 1 value of regulator resistance 2 no load terminal voltage for given speeds

```
1  clc;
2  disp('case a');
3  v=90; // voltage build up by regulating resistance
4  rs=(v*v)/(2*v); // shunt winding resistance
5  IF=[500 1000 1500 2000 2500 3000];
6  VA=[154 302 396 458 505 538 ];
7  subplot(313);
8  plot(IF,VA);
9  xlabel('Field ATs/pole');
10 ylabel('Generated e.m.f E,(V)');
```

```
11  title('Magnetizing curve for n=500 r.p.m.');
12  // from magnetizing curve for v=90, field current is
        0.89 A
13  ifl=0.89; // field current
14  re=(v/(2*ifl))-rs;
15  printf('Value of the resistance in the regulator is
        %f ohms\n',re);
16  disp('case b');
17  t1=800; // turns per pole of separately excited
        winding
18  r1=160; // resistance of winding
19  vc=220; // constant supply voltage
20  t2=500; // turns per pole of shunt winding
21  r2=200; // resistance of winding
22  AT1=(vc*t1)/r1; // Ampere turns of separately
        excited winding
23  // AT2=(t2/r2)*E is Ampere turns of shunt winding
        and E is generated EMF
24  AT3=AT1+(t2/r2)*VA
25  n1=500;
26  n2=600; // given speeds
27  VA2=(n2/n1)*VA; // generated EMF for n=600 rpm
28  subplot(323);
29  plot(IF,VA2);
30  xlabel('Field ATs/pole');
31  ylabel('Generated e.m.f E,(V)');
32  title('Magnetizing curve for n=600 r.p.m.');
33  subplot(333);
34  plot(AT3,VA);
35  ylabel('Generated e.m.f E,(V)');
36  xlabel('Total ATs/pole due to both field winding');
37  title('Generated e.m.f E,(V) vs total Ampere turns/
        pole ');
38  // plot of variation of generated e.m.f with total
        Ampere turns per pole intersects magnetizing
        curve for n=500 rpm at P and  magnetizing curve
        for n=600 rpm at Q. (refer fig. 4.48)
39  // Point P gives no-load terminal voltage at 500 rpm
```

151

```
40  disp('No load voltage at 500 rpm is 490 V and at 600
        rpm is 621 V');
```
_____

Scilab code Exa 4.24 Determining terminal voltage and number of series field turns per pole

```
1  clc;
2  //magnetization curve at 1200 rpm
3  If=[0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8];//field
        current in rpm
4  Ea=[6 53 106 160 209 241 258 272 282 288];//induced
        voltage in armature
5  plot(If,Ea);xlabel('If');ylabel('Ea');//
        magnetization curve at 1200 rpm
6  Pout=10000;//genertor output in watts
7  Vt=230;//Terminal Voltage
8  Ra=0.5;//armature resistance with brushes
9  Ns=1000;//turns of shunt winding
10 Nf=4;//turns of field winding
11 Z=1000;//No of conductors
12
13 //PART (A)
14 //At rated output current the speed is 1150 rpm &
        shunt field current is 1 A
15 If_=1;//field current at rated o/p current
16 Il=Pout/Vt;//rated output current
17 Ia=Il+If;//armature current at rated load
18 Is=Ia;//for long shunt compound generator the series
         field current is equal to armature current
19 //Since the compound geenerator is cumulatively
        compounded ,the total pole per m.m.f. is (Nf*If+
        Ns*Is)ampere turns
20 //Thus the equivalent shunt field current is given
        by 1/Nf*(Nf*If+Ns*Is)=1+(4*44.5/1000)=1.18 A. The
```

152

```
          generated emf for this field currenr from the
            magnetization curve is 257 volts.
21  //For speed of 1150 rpm the generated emf is−
22  Ea_=257*(1150/1200);
23  Vt_=Ea_-Ia*Ra;//terminal voltage
24
25  //PART(B)
26  Eg=Vt+Ia*Ra;//generated emf in the armature at 1150
       rpm
27  //By using the magnetization curve, the generated
      emf at 1200 rpm will be 252.25*(1200/1500)=263.3
      volts.
28  //From the open circuit characteristics, the field
      current corresponding to 263.3 volts is 1.26 A.
29  MMFt=1.26*1000;//Total MMF
30  //Total MMF must be produced by the combined action
      of shunt & series windings.
31  //1.26*1000=1.00*1000+Ns*(44.5);
32  Ns_=(0.26*1000)/44.5;//series field turns
33  printf('The number of series field turns should be
      %f.',round(Ns_));
```

___

Scilab code Exa 4.25 Determining series field turns and resistance of diverter for achieving desired performance

```
1  clc;
2  //repeat part (b) of example 4.21
3
4  //PART(a)−
5  //When the demagnetizing effect is accounted for,
      then from equation :−Net mmf = Nf*If+Ns*Is−ATd
      ....(1)
6  //1.26*1000=1.00*1000+10 Is −0.022 Is *1000
7  Ns=round(0.3578*1000/44.5);//no of turns in series
      field winding
```

```
8
9  //PART(b)−
10 // If there are 10 series field turns, then from
       equation (1),
11 // 1.26*1000=1.00*1000+10 Is −0.0022 Is ∗1000
12 Is =0.26/0.0078
13 //Out of the total armature current of 44.5 A, only
       Is (33.3) should flow through the series field.
14 // This can be achieved by putting a resistor in
       parallel with the series field winding.
15 // 33.3=(44.5∗ Rdi)/(0.05+ Rdi )
16 Rdi =0.05/0.3363;
17 printf ('NO OF TURNS IN SERIES FIELD WINDING ARE %f.'
       , Ns );
18 printf ('\nTHE RESISTANCE OF DIVERTER Rdi SHOULD BE
       %f OHMS.', Rdi );
```

Scilab code Exa 4.26 Determining emf generated in armature and percentage change in series field ampere turns due to diverter

```
1  clc ;
2  Vt =250; // rated o/p voltage of generator
3  Pout =10000; //o/p of generator in watts
4  Ra =0.4; // armature resistance
5  Rse =0.2; // series field resistance
6  Rs =125; // shunt field reesistance
7  Vb =2; // total brush cotact drop
8  Il = Pout / Vt ; // load current
9
10 //PART(a)−LONG SHUNT CONNECTION
11 If = Vt / Rs ; // shunt field current
12 Ia = Il + If ; // armature current
13 // series field winding also carries Ia.
14 Eal = Vt + Ia *( Rse + Ra )+ Vb ; // generated emf in armature
15 printf ('The generated EMF in armature when the
```

```
        generated is connected as long shunt machine is
        %f.\n',Eal);
16
17 //PART(b)-SHORT SHUNT CONNECTION
18 V=Vt+Il*Rse;//voltage across shunt field and
        armature terminals
19 If_=V/Rs;//shunt field current
20 Ia_=Il+If_;//armature current
21 Eas=V+Ia*Ra+Vb;//generated emf in armature
22 printf('The generated EMF in armature when the
        generated is connected as short shunt machine is
        %f.',Eas);
23
24 //PART(c)-
25 //Series field ampere turns are proportional to
        series-field current I
26 //Is=0.3/0.5*I, where , Is is series field current
        with diverter.
27 //series field ampere-turns with dvider = K*0.6*Is,
        where K is a constant.
28 //percentage reduction in series field ampere turns
        is - ((I-O.6I)/I)*100.
29 disp('Percentage reduction in series field ampere
        turns is 40%.');
```

Scilab code Exa 4.27 Determining speed and developed torque at full load

```
1 clc;
2 Vt=230;//output voltage
3 Ra=0.3;//armature circuit resistance
4 Rf=160;//field circuit resistance
5 Il=40;//line current at full load & rated voltage
6 Ia1=3.33//armature current at rated voltage & no
        load speed of 1000 rpm
```

155

```
7  //No load counter emf is−
8  Ea1=Vt-Ia1*Ra;
9  If=Vt/Rf;//field current
10 //At full load armature current is−
11 Ia2=Il-If;
12 Ea2=Vt-Ia2*Ra;//Counter emf at full load
13 //At full load, the field flux is−
14 //Phi_2=0.96*phi_1
15 //The conter emf Ea, is given by− Ea=Ka*phi*Wm
16 //Ea1/Ea2=(Ka*phi_1*Wm1)/(Ka*phi_2*Wm2)=(phi_1*n1)/(
      phi_2*n2) or 229/218.43=(1000*phi_1)/n2*(0.96*
      phi_1)...(1)
17 //from equation (1)
18 n2=995;//full load speed
19 //At full load, Ea2=Ka*phi_2*Wm
20 //Ka*phi_2=Ea2/Wm
21 //Electromagnetic or developed, torque at full load
      is, Te=Ka*phi_2*Ia2
22 Te=(Ea2*60)/(2*%pi*n2)*Ia2;//Electromagnetic torque
      developed.
23 printf('Electromagnetic or developed, torque at full
       load is %f.',Te);
```

Scilab code Exa 4.28 Determining motor speed and rated shaft torque

```
1  clc;
2  //Armature reaction is neglected
3  Vt=220;//output voltage
4  Ra=0.2;//armature circuit resistance
5  Rf=110;//field circuit resistance
6  n1=1500;//speed of rotor at no load
7  Il1=5;//curent drawn by motor in ampere at no load &
       1500 rpm
8  Il2=52;//curent drawn by motor in ampere at rated
      load & rated voltage
```

156

```
 9  If=Vt/Rf;//shunt field current
10  Ia1=Il1-If;//armature current at no load
11  Ea1=Vt-Ia1*Ra;//counter emf at no load
12  Pr=Ea1*Ia1;//rotational losses at no load
13  //Rotational losses at full load & no load are same
14  Ia2=Il2-If;//armature current at full load
15  Ea2=Vt-Ia2*Ra;//counter emf at full load
16  Pem=Ia2*Ea2;//electromagnetic power
17  //Here phi_1(no load flux)=phi_2(full load flux),
        because the field current is constant & effect of
         AR is neglected.
18  //Ea1/Ea2=(n1*phi_1)/(n2*phi_2);where n1 & n2 are
        speed of rotor at no load & full load
        respectively.
19  n2=fix((n1*Ea2)/(Ea1));
20  Psh=Pem-Pr;//shaft power
21  Wm=(2*%pi*n2)/60;//angular velocity of shaft at full
        load
22  Tsh=Psh/Wm//shaft torque
23  printf('The motor speed is %f rpm.\n',n2);
24  printf('Rated shaft torque is %f Nm.',Tsh);
```

Scilab code Exa 4.29 Determining external resistance inserted in field circuit

```
1
2  clc;
3  Ra=0.4;//armature resistance in ohm
4  Rf=200;//field circuit resistance in ohm
5  Vt=230;//terminal voltage for dc motor
6  If_1=1.1;//field current for dc generator at open
      circuit voltage of 210 V.
7  If_2=0.9;//field current for dc generator at open
      circuit voltage of 230 V.
8  Ia=24;//armature current for dc shunt motor at 1500
```

157

```
          rpm
 9  Ea=Vt-Ia*Ra;//counter e.m.f. for dc motor at 1500
        rpm and full load
10  //For generated e.m.f., Ea=230 V, field current is
        1.1 A & for Ea=210 V, field current is 0.9 A
11  //The change in generated e.m.f. is 20 V for field
        variation of 0.2 A & this change is linear.
12  //Therefore for a generated e.m.f. of Ea=220.4 V at
        1500 rpm, the field current would be-
13  If=0.9+(0.2/20)*10.4;//0.9 A for 210 V & (0.2/20)
        *10.4 for remaining 10.4 V.
14  Rsh=Vt/If;//Shunt field resistance required for a
        field current(If) with terminal voltage(Vt).
15  Rext=Rsh-Rf;//External resistance that must be
        inserted in shunt field circuit
16  printf('The external resistance that must be
        inserted in shunt field circuit = %f ohm.',Rext);
```

Scilab code Exa 4.30 Determining 1 speed and internal torque developed 2 shaft power shaft torque and efficiency

```
 1  clc;
 2  //Armature reaction is neglected.
 3  Vt=250;//Supply voltage
 4  P=4;//No of poles
 5  A=2;//No of parallel paths for armature conductors
 6  Z=500;//No of armature conductors
 7  Ra=0.25;//armature circuit resistance in ohm
 8  Rf=125;//field resistance in ohm
 9  phi=0.02;//flux per pole in weber
10  Il=14;//current drawn by motor from supply mains
11  Ish=Vt/Rf;//constant shunt field current
12  Pr=300;//rotational losses in watts
13  Pi=Vt*Il;//power input in watts
14
```

```
15  //PART(a)-
16  Ia=Il-Ish;//armature current
17  Ea=Vt-Ia*Ra;//counter/back emf
18  //Ea=(P*phi*Z*N)/(60*A)
19  N=(60*A*Ea)/(P*phi*Z);//speed of rotation of motor
       in rpm
20  Wm=(2*%pi*N)/60;//angular velocity of motor
21  Pe=Ea*Ia;//electromagnetic power
22  Ti=Pe/Wm;//Internal torque developed in Nm.
23  printf('Speed of rotation of motor is %f rpm.',N);
24  printf('\nInternal torque developed = %f Nm.',Ti);
25
26  //PART(b)-
27  Psh=Pe-Pr;//shaft power
28  Tsh=Psh/Wm;//shaft torque
29  %n=(Psh/Pi)*100;//percentage eficiency
30  printf('\nShaft power = %f watts.',Psh);
31  printf('\nShaft torque = %f Nm.',Tsh);
32  printf('\nEfficiency of motor is %f percent.',%n);
```

Scilab code Exa 4.31 Determining shaft power operating speed armature current and motor efficiency

```
1  clc;
2  Vt=230;//Supply voltage
3  P=4;//No of poles
4  A=2;//No of parallel paths for armature conductors
5  Z=600;//No of armature conductors
6  Ra=0.25;//armature circuit resistance in ohm
7  phi=0.01;//flux per pole in weber
8  Pr=500;//rotational losses in watts
9  //generated emf in armature, Ea=(phi*Z*P*n)/(60*A)
       if n is speed in armature
10  //Counter emf is Ea=(0.01*600*4*n)/(60*2)=0.2n volts
11  //Vt=Ea+Ia*Ra
```

159

www.mepcafe.com

```
12  // Ia=(Vt−Ea)/Ra/
13  // Shaft  o/p  in  watts ,  Psh=Ea∗Ia−Pr ,  Psh=(0.2n)
        ∗(920−0.8n)−500  ....(1)
14  n=[700  800  900  1000  1100];// different  speeds  of
        motor  for  which  the  shaft  o/p  power  is  to  be
        measured
15  // Psh=(0.184∗n)−(1.6∗(10e−4)∗n∗n)−0.5,  Shaft  o/p
        power  in  KW.
16  Psh1=49.1;// Shaft  o/p  power  in  KW  at  n=700  rpm
17  Psh2=44.3;// Shaft  o/p  power  in  KW  at  n=800  rpm
18  Psh3=33.5;// Shaft  o/p  power  in  KW  at  n=900  rpm
            ∗Psh1 ,Psh2 ,Psh3 ,Psh4 ,Psh5  are  calclulated
        from  equation  (1)∗
19  Psh4=23.5;// Shaft  o/p  power  in  KW  at  n=1000  rpm
20  Psh5=8.3;// Shaft  o/p  power  in  KW  at  n=1100  rpm
21  Psh=[Psh1  Psh2  Psh3  Psh4  Psh5];
22  Pi=[4.5  8.5  14  21.1  30];// i/p  power  supplied  to  fan
        in  KW.
23  plot(n,Psh,n,Pi);xlabel('RPM');ylabel('KW');// Shaft
        o/p  power  versus  speed  of  motor  and  power  i/p
        versus  speed  of  fan  are  plotted  on  the  same  graph
24  // For  plot  :−∗blue  line−  motor  characteristic ,green
        line−  fan  charactestic∗
25  // The  intesecton  of  these  two  curves  is  called  as
        OPERATING  POINT.
26  // At  operating  point  the  speed  is  1012  rpm  &  pwer  o/
        p  of  motor  or  power  i/p  to  the  fan  is  22  KW  ....(
        from  the  intersection  point  of  two  curves )
27  n_=1012;// speed  at  operating  point
28  P_o=22000;// Power  output  loss
29  Ia=920-(0.8∗n_);// armature  current
30  Parm=Ia∗Ia∗Ra;// armature  loss
31  P_ip=P_o+Pr+Parm;// Power  input
32  %n=(P_o/P_ip)∗100;// motor  efficiency
33  printf('Armature  current  is  %f  A\n.',Ia);
34  printf('Operating  speed  is  %f  rpm\n.',n_)
35  printf('Motor  efficiency  is  %f  percent.',%n);
```

Scilab code Exa 4.32 Determining speed of motor

```
1  clc;
2  Vt=230;//Supply voltage
3  P=4;//No of poles
4  A=2;//No of parallel paths for armature conductors
5  Z=500;//No of armature conductors
6  Ra=0.2;//armature circuit resistance in ohm
7  Rs=0.1;//field resistance in ohm
8  Il=40;//line current
9  N=1000;//rated speed in rpm
10 Ia1=40;//armature current for dc series motor at 40
       A line current
11 Ia2=20;//armature current for dc series motor at 20
       A line current
12 //For 40A line current
13 Ea1=Vt-Ia1*(Ra+Rs);//counter emf
14 //For 20A line current
15 Ea2=Vt-Ia2*(Ra+Rs);//counter emf
16 //Let, phi_1=flux at 40 A, phi_2=flux at 20 A line
       current
17 //phi_2=0.6*(phi_1)
18 //(Ea1/Ea2)=(n1*phi_1)/(n2*phi_2)
19 //(218/224)=(1000*(phi_1))/(n2*(0.6*(phi_1))
20 n2=(1000*224)/(218*0.6);//speed of motor at line
       current of 20 A at 230 V
21 printf('Speed of motor at line current of 20 A at
       230 V is %f rpm.',round(n2));
```

Scilab code Exa 4.33 Determining 1 speed in rpm 2 shaft power output

161

```
1  //ANSWER GIVEN IN THE BOOK FOR THIS QUESTION IS
       INCORRECT.
2
3  clc;
4  //Neglecting armature reaction & magnetic saturation
5  //Assuming rotational losses to remain constant
6  V=230;//Supply voltage
7  P=15000;//power rating of dc series motor in watts
8  Il_1=80;//line current rated
9  Il_2=40;//line current assuming that motor draws
       half the rated current at rated voltage
10 Ia_1=Il_1;//armature current at line current equal
       to 80 A.
11 Ia_2=Il_2;//armature current at line current equal
       to 40 A.
12 n1=1000;//rated speed in rpm
13 //Full load losses expressed as percentage of motor
       input:-
14 //Armature ohmic loss=2.8%(including brush loss)
15 //Field ohmic loss=2.2%
16 //Rotational loss=2.2%
17 P_ip=V*Il_1;//full load input
18 P_ohmic=P_ip*(5.4/100)//As percent of total ohmic
       losses=2.2+2.8=5.4%
19 //But P_ohmic=Il*Il*(Ra+Rs); where (Ra+Rs)=(armature
       + series field) resistance
20 //(Ra+Rs)=P_ohmic/(Il*Il)=0.115 ohms
21 //Let, r=(Ra+Rs)
22 r=0.115;
23
24 //PART(a)-
25 Ea1=V-(Ia_1*r);//counter emf at line current = 80 A
26 Ea2=V-(Ia_2*r);//counter emf at line current = 40 A
27 //Since the magnetic saturation is neglected, phi_1=
       k*80 & phi_2=k*40; where k=constant & phi_1 &
       phi_2 are flux per pole at line currents 80 & 40
       A respectively.
28 //(Ea1/Ea2)=(n1*phi_1)/(n2*phi_2) or (220.8/225.4)
```

162

```
        =(1000*80)/(n2*40); where Ea1=220.8 V Ea2=225.4 V
           .
29  n2=(1000*80*225.4)/(40*220.8); //speed in rpm
30  printf('The speed of rotation of motor when the
         motor draws half the rated current at rated
         voltage is %f rpm.',round(n2));
31
32  //PART(b)-
33  Pr=P_ip*(2.2/100); //rotational losses
34  Psh=Ea2*Ia_2-Pr;
35  printf('\nThe shaft output power is %f W.',Psh);
```

Scilab code Exa 4.34 Determining operating speed and current drawn from source

```
 1
 2  clc;
 3  Pout=40000; //output power
 4  V=250; //supply voltage in volts
 5  r=0.2; //sum of armature circuit resistance & series
          field circuit resistance
 6  n=1500; //speed of dc series motor at rated current
 7
 8  //PART(a)-
 9
10  Ir=Pout/V; //rated current
11  Ea=V-Ir*r; //counter emf at rated load
12  Wm=(2*%pi*n)/60; //angular speed of rotation of motor
13  Te=round((Ea*Ir)/Wm); //rated electromagnetic torque
          in Nm.
14  //Now the relation giving the torque speed
          characteristics of series motor must be developed
          .
15  //Since the magnetic saturation is neglected
16  //Te=Ka*phi*I
```

163

```
17  //Te=K1*Ia^2
18  K1=Te/(Ir^2);//Ia=rated current
19  //Ea=K2*phi*n=K2*Ia*n
20  K2=Ea/(Ir*n);// values of constant of
        proportionality
21  //The values of constants k1 & K2 are obtained from
        rated conditions
22  //Ia=(V-Ea)/r & Ea=K2*Ia*n ; Ia=1250-0.00454*Ia*n
23  //Ia=1250/(1+0.00454n) ....(1); Ia=armature current
        at any speed
24  //Te=K1*Ia^2
25  n=[1400 1450 1500 1550 1600 1650 1700];
26  Te=K1.*(((V/r)^2)./(1+(K2/r).*n).^2); //
        Electromagnetic torque
27  Tl=5.*sqrt(n); // load torque
28  plot(Te,n,Tl,n);
29  xlabel('T(Nm)');
30  ylabel('Speed(rpm)');
31  title('Speed-torque characteristics for Series-motor
         and for load');
32
33  //THE INTERSECTION OF SERIES MOTOR & LOAD
        CHARACTERISTICS GIVES THE OPERATING POINT
34
35  //From the curve the operating point is obtained at
        1591 rpm & torque is 199.5 Nm
36  disp('The operating speed of motor is 1591 rpm.');
37
38  //PART(b)-
39  //Current drawn from the source is -
40  Ia=(V/r)/(1+(K2/r)*1591);//From equation (1)
41  printf('Current drawn from the source is %f A.',Ia);
```

Scilab code Exa 4.35 Determining number of series field turns

```scilab
1  clc;
2  V=230;//supply current
3  Pout=15000;//output voltage
4  Ra=0.2;//armature circuit resistance
5  Rse=0.1;//resistance of series field winding
6  Nf=1000;//shunt field turns per pole
7  //Data for magnetization curve at 1500 rpm
8  If_=[0 0.2 0.4 0.6 0.8 1.02 1.15 1.32 1.56 1.92
       2.4];//field current at 1500 rpm
9  Ea=[6 40 80 120 160 200 220 240 260 280 300];//
       counter emf at 1500 rpm and respective field
       current
10 plot(If_,Ea);//Magnetization curve at 1500 rpm
11 Ia_1=4;//armature current at rated voltage and rated
        load
12 Ia_2=70;//armature current at 1200rpm
13
14 //At no load
15 Ea_=V-Ia_1*Ra;//counter emf
16 //field current required for Ea_(ie.229.2 V),from O.
       C.C. is 1.23 A.
17
18 //At load
19 Ea__=V-Ia_2*(Ra+Rse);//counter emf at 1200 rpm
20 Ea___=209*(1500/1200);//Ea at speed of 1500 rpm
21 //Field current corresponding to Ea___(ie.261.25 V),
       from O.C.C. is 1.575 A.
22 //Total D-axis mmf per pole=Nf*If+Ns*Is
23 If=1.23;//field current at 229.2 V is 1.23 A
24 If1=1.23;//field current at 261.25 V is 1.575 A
25 //1.575*1000=1.23*1000+Ns*(70)
26 Ns=(0.345*1000)/70;//series field turns
27 printf('For long-shunt connection, series field
        turns is equal to %f.',round(Ns));
```

Scilab code Exa 4.36 Determining 1 shunt field current 2 effective armature reaction 3 number of series field turns 4 speed at rated armature current and at rated voltage 5 internal starting torque

```
1  //Answer for part(e) in book is incorrect.
2  clc;
3  //Magnetization curve is same as that of example
       4.33
4  Ra=0.2;//armature resistance (including brushes)
5  Nf=2000;//shunt field turns
6  N=1500;//motor speed in rpm at no load as well as
       rated load.
7  Ia=36;//motor armature current in Amperes at rated
       load
8
9  //PART(a)-At no load,
10 Vt=230;//supply mains in volts
11 Ea_a=Vt;//counter emf, neglecting armature circuit
       resistance
12 If_a=1.23;//field current in amperes
13 printf('(a) Shunt field current is %f A.\n',If_a);
14 //Thus constant shunt field current If from O.C.C.
       is 1.23 A corresponding to 230V.
15
16 //PART(b)-At full load,
17 Ea_b=Vt-Ia*Ra;//counter emf
18 //A point is drawn on magnetization curve with
       coordinates A(If,Ea_b).
19 //The horizontal distance between Pt. A & the
       magnetization curve, gives the effective armature
        reaction in terms of shunt field current, its
       value is 0.06 A.
20 AT_arm=Nf*0.06;//armature reaction in amopere turns
       per pole
21 printf('(b) Effective armature reaction is %f ampere
        turns per pole.\n',AT_arm);
22
23 //PART(c)-At rated load, with series winding in
```

166

```scilab
        circuit & motor is cumulatively compounded ,
24  Rse =0.05;// series field resistance in ohm
25  Ea__c=Vt-Ia*(Ra+Rse);// counter emf at 1350 rpm
26  Ea_c=Ea__c*(1500/1350);//Ea at 1500 rpm
27  //From magnetization curve , Ea=245.5 V requires If_c
        =1.365 A.
28  //From equation − Net MMF =Nf*If+Ns*Is−ATd ....(1)
29  //1.365*2000=1.23*2000+Ns*(36)−120
30  Ns=round(65/6);// Series field current
31  printf('(c) Required no of series field turns are %f
        .\n',Ns);
32
33  //PART(d)−If the series field winding has 20 turns −
34  Ns_=20;// no of turns of series field winding
35  //Net mmf = Nf*If+Ns_*Is−AT_arm ....(formula)
36  mmf_net=If_a*Nf+Ns_*Ia-AT_arm;//Net field mmf in
        terms of ATs
37  If_d=(If_a*Nf+Ns_*Ia-AT_arm)/Nf;//Net field mmf in
        terms of the equivalent shunt field current(A).
38  //From the magnetization curve , the value of Ea
        corresponding to If=1.53A is 258V at 1500rpm.
39  //But the counter emf,Ea corresponding to rated
        current is 230−36(0.2+0.05)=221 V.
40  //Therefore the motor speed n corresponding to Ea
        =221V is−
41  //(221/258)=(n/1500)
42  n=(221/258)*1500;
43  printf('(d) Speed at rated voltage rated armature
        current is %f rpm.\n',round(n));
44
45  //PART(e)−Assuming demagnetizing effect of armature
        reaction to be 200 ampere turns per pole.
46  ATarm=200;// demagnetizing effect of armature
        reaction in ampere turns per pole.
47  Ia_e=50;// armature current in amperes ....( given )
48  mmfnet=If_a*Nf+Ns_*Ia_e-ATarm;// from equation no
        ....(1)
49  If_e=mmfnet/Nf;//Net field mmf in terms of the
```

167

```
         equivalent  shunt  field  current (A).
50  //From  the  magnetizing  curve ,  corresponding  to  field
        current  If_e (1.63  A) ,  Ea  at  1500  rpm  is  264  V.
51  //But ,  Ea=Ka∗phi∗Wm ;  where ,  phi  =  flux  per  pole
52  //Thus ,  Ka∗phi=(264∗60)/(2∗%pi∗1500)
53  Kaphi=264/(50*%pi); //Ka∗phi
54  Test=Kaphi*Ia_e; //starting  torque
55  printf('(e) When  the  armature  current  is  limited  to
     50  A  the  starting  torque  is  %f Nm.',Test);
```

Scilab code Exa 4.37 Determining steady state speed and armature current drawn from source

```
1
2  clc;
3  V=230; //supply  voltage  in  volts
4  Ra=0.5; //armature  resistance  in  ohm
5  N=250; //rated  speed  of  motor
6  I=100; //rated  current  in  ampere
7  //For  the  separately  excited  dc  motor  torque−speed
     characteristics  is  given  by  Tl=500−W,  where  W  is
     rotational  speed  in  rad/sec  &  Tl  is  load  torque
     in  Nm.
8  //At  rated  load ,  motor  counter  emf  is  −
9  Ea=V-I*Ra;
10  //Ea=Km∗Wr;  Km  =  motor  constant ,  Wr  =  rated  motor
      speed  in  rad/sec
11  Wr=(2*%pi*250)/60; //rated  motor  speed  in  rad/sec
12  Km=Ea/Wr; //motor  constant  in  V−s/rad
13  //Armature  current  at  any  speed  W  is  given  by−
14  Ia=(V-Ea)/Ra; //  ie .  Ia=(230−Km∗W)/0.5
15  //Motor  torque ,  Te=Km∗Ia=(Km/0.5)∗(230−Km∗W)
16  //Under  steady  state ,  motor  torque ,Te=load  torque ,
      Tl
17  //Thus ,  (Km/0.5)∗(230−Km∗W)=500−10∗W
```

168

```
18  W=(((V*Km)/Ra)-500)/((Km^2/Ra)-10); //angular  speed
         in  rpm
19  N_=(W*60)/(2*%pi); //Speed  in  rpm
20  Ia_=(230-Km*W)/0.5 //armature  current
21  printf('Steady  state  speed  of  motor  is  %f  rpm\n.',N_
         );
22  printf('Armature  current  drawn  by  motor  at  steady
         state  is  %f  A.',Ia_);
```

Scilab code Exa 4.38 Determining 1 external resistance 2 value of first resistance element 3 external resistance cut out in second step 4 total number of steps

```
1  clc;
2  v=230; // rated  voltage  of  dc  motor
3  p=10000; // rated  power  of  dc  motor
4  rf=115; // field  resistance
5  ra=0.348; // net  armature  resistance
6  ifs=v/rf; // shunt  field  current
7  ia=(p/v)-ifs; // rated  armature  current
8  disp('case  a');
9  rx1=(v/(2*ia))-ra;
10 printf('External  resistance  required  at  the  time  of
        starting  is  %f  ohms\n',rx1);
11 disp('case  b');
12 Ea1=v-ia*(rx1+ra); // counter  emf  at  stud  1
13 r2=(v-Ea1)/(2*ia); // resistance  when  handle  is
        moved  to  2nd  stud
14 rx2=r2-ra; // external  resistance
15 rc=rx1-rx2;
16 printf('Resistance  that  must  be  cut  out  in  first
        step  is  %f  ohms\n',rc);
17 disp('case  c');
18 Ea2=v-ia*rc; // counter  emf  at  stud  2
19 r3=(v-Ea2)/(2*ia); // resistance  when  handle  is
```

169

```
        moved to 3rd stud
20  rc=rc-r3;
21  printf('Resistance that must be cut out in second
        step is %f ohms\n',rc);
22  disp('case d');
23  Ea3=v-ia*rc; // counter emf at stud 3
24  r4=(v-Ea3)/(2*ia); // resistance when handle is
        moved to 4th stud
25  rc=rc-r4;
26  printf('Resistance that must be cut out in third
        step is %f ohms\n',rc);
27  disp('Total number of steps is 3');
```

Scilab code Exa 4.39 Calculating 1 resistance of each step 2 voltage at which contactors should be close

```
 1  clc;
 2  v=240; // rated voltage of dc shunt motor
 3  i=50; // rated current of dc shunt motor
 4  ra=0.2; // armature resistance
 5  n=4; // number of resistance element
 6  N=1500; // rated speed of motor
 7  vb=1; // pu base voltage
 8  ia=1; // pu base current
 9  rb=v/i; // pu base resistance
10  ra=ra/rb; // per unit armature resistance
11  disp('case a');
12  ia1=1.4; // pu maximum allowable armature current
13  R=vb/ia1; // net resistance
14  al=(ra/R)^(1/n); // ratio of total resistances on
        two adjacent studs
15  r1=R*(1-al);
16  printf('Resistance cut out when handle is at stud 2
        is %f pu or %f ohms\n',r1,r1*rb);
17  r2=al*r1;
```

170

```
18  printf('Resistance cut out when handle is at stud 3
         is %f pu or %f ohms\n',r2,r2*rb);
19  r3=al*r2;
20  printf('Resistance cut out when handle is at stud 4
         is %f pu or %f ohms\n',r3,r3*rb);
21  r4=al*r3;
22  printf('Resistance cut out when handle is at stud 5
         is %f pu or %f ohms\n',r4,r4*rb);
23  disp('case b');
24  ia2=al*ia1; // pu minimum armature current
25  // at stud 1 armature current=ia2 after t1 where t
         is time reckoned from the instant motor is
         switched on
26  Ea1=vb-ia2*R; // counter EMF at stud 1
27  Va1=Ea1+ia2*ra; // voltage across at armature
         terminal at instant t1
28  printf('The first contactor should close at %f pu or
          %f V\n',Va1,Va1*v);
29  // at stud 2 armature current=ia2 after t2 where t
         is time reckoned from the instant motor is
         switched on
30  Ea2=vb-ia2*(R-r1); // counter EMF at stud 2
31  Va2=Ea2+ia2*ra; // voltage across at armature
         terminal at instant t2
32  printf('The second contactor should close at %f pu
         or %f V\n',Va2,Va2*v);
33  // at stud 3 armature current=ia2 after t3 where t
         is time reckoned from the instant motor is
         switched on
34  Ea3=vb-ia2*(R-r1-r2); // counter EMF at stud 3
35  Va3=Ea3+ia2*ra; // voltage across at armature
         terminal at instant t3
36  printf('The third contactor should close at %f pu or
          %f V\n',Va3,Va3*v);
37  // at stud 4 armature current=ia2 after t4 where t
         is time reckoned from the instant motor is
         switched on
38  Ea4=vb-ia2*(R-r1-r2-r3); // counter EMF at stud 4
```

171

```
39  Va4=Ea4+ia2*ra; // voltage across at armature
        terminal at instant t4
40  printf('The fourth contactor should close at %f pu
        or %f V\n',Va4,Va4*v);
41  disp('case c');
42  Ea=vb-ia*ra; // pu full load counter EMF
43  n1=Ea1/Ea; // pu speed when handle is at stud 1
44  printf('Speed of dc shunt motor when handle is at
        stud 1 is %f pu or %f rpm\n',n1,n1*N);
45  n2=Ea2/Ea; // pu speed when handle is at stud 2
46  printf('Speed of dc shunt motor when handle is at
        stud 2 is %f pu or %f rpm\n',n2,n2*N);
47  n3=Ea3/Ea; // pu speed when handle is at stud 3
48  printf('Speed of dc shunt motor when handle is at
        stud 3 is %f pu or %f rpm\n',n3,n3*N);
49  n4=Ea4/Ea; // pu speed when handle is at stud 4
50  printf('Speed of dc shunt motor when handle is at
        stud 4 is %f pu or %f rpm\n',n4,n4*N);
51  disp('Using above data sketch of variation of
        armature current and speed can be obtained with
        time');
```

Scilab code Exa 4.40 Determining load torque for different cases

```
1  clc;
2  v=200; // rated voltage of shunt motor
3  i=22; // rated current of dc shunt motor
4  n1=1000; // speed at which motor is running
5  rf=100; // field resistance
6  ra=0.1; // armature resistance
7  n2=800; // reduced speed at which motor is to run
8  iF=v/rf; // field current
9  ia=i-iF; // armature current
10 disp('case a');
11 // load torque is independent of speed
```

```
12  Ea1=v-ia*ra; // counter EMF at 1000 rpm
13  rg=(v-ia*ra-(n2*Ea1)/n1)/ia;
14  printf('Additional resistance inserted in armature
        circuit is %f ohms\n',rg);
15  printf('Loss in additional resistance is %f W\n',ia
        ^2*rg);
16  disp('case b');
17  // load torque is directly proportional to speed
18  ia2=(n2/n1)*ia; // armature current at 800 rpm
19  rg=(v-ia2*ra-(n2*Ea1)/n1)/ia2;
20  printf('Additional resistance inserted in armature
        circuit is %f ohms\n',rg);
21  printf('Loss in additional resistance is %f W\n',ia2
        ^2*rg);
22  disp('case c');
23  // load torque varies as the square of speed
24  ia2=(n2/n1)^2*ia; // armature current at 800 rpm
25  rg=(v-ia2*ra-(n2*Ea1)/n1)/ia2;
26  printf('Additional resistance inserted in armature
        circuit is %f ohms\n',rg);
27  printf('Loss in additional resistance is %f W\n',ia2
        ^2*rg);
28  disp('case d');
29  // load torque varies as the cube of speed
30  ia2=(n2/n1)^3*ia; // armature current at 800 rpm
31  rg=(v-ia2*ra-(n2*Ea1)/n1)/ia2;
32  printf('Additional resistance inserted in armature
        circuit is %f ohms\n',rg);
33  printf('Loss in additional resistance is %f W\n',ia2
        ^2*rg);
```

Scilab code Exa 4.41 Finding motor speed

```
1  clc;
2  V=240; // rated voltage of dc shunt motor
```

```
3  n=800; // rated speed of dc shunt motor
4  i=50; // rated current of dc shunt motor
5  ra=0.2; // armature resistance
6  pr=0.6; // reduction in load torque as a fraction of
       full load torque
7  rg=2; // series resistance in armature circuit
8  fr1=0.04; // weakening of field flux at full load
9  fr2=0.02; // weakening of field flux at 60% of full
       load
10 Ea1=V-(i*ra); // counter EMF at rated load
11 ia2=(i*pr)*((1-fr1)/(1-fr2)); // armature current at
        reduced load torque
12 Ea2=V-ia2*(rg+ra); // counter EMF at reduced load
       torque
13 n2=(n*Ea2*(1-fr1))/(Ea1*(1-fr2));
14 printf('Motor speed at reduced load torque is %f rpm
       ',n2);
```

Scilab code Exa 4.42 Finding motor speed

```
1  clc;
2  N=1000; // speed of dc series motor
3  v=250; // supply from mains
4  i=50; // current drawn from mains
5  r=0.6; // armature + field resistance
6  rg=4.4; // additional resistance
7  // field flux is proportional to armature current
8  Ea1=v-i*r; // counter EMF at 1000 rpm
9  // Ea2=v-(n2/20)*(r+rg) where Ea2 is counter EMf at
      speed n2 . taking ratio of Ea2/Ea1 we obtain a
      quadratic equation in n2 whose terms are given by
10 t1=(Ea1*i)/N;
11 t2=(N*i)*((r+rg)/(N/i));
12 t3=-(N*i*v);
13 p=[ t1 t2 t3];
```

```
14 n=roots(p);
15 printf('New speed of motor is %f rpm',ceil(n(2)));
```

Scilab code Exa 4.43 Finding speed regulation

```
1  clc;
2  v=230; // rated voltage of dc shunt motor
3  n1=900; // speed at which motor is running
4  ia1=2; // armature current at n=900 rpm
5  ra=0.5; // armature resistance
6  ia2=20; // armature current at rated load and rated
       voltage
7  Ea=v-ia1*ra; // counter EMF at no load
8  k=(Ea*60)/(2*%pi*n1); // constant term used for
       calculating back EMF
9  disp('case a');
10 rs=2; // resistance in series with armature
11 rp=3; // resistace in parallel with series
       combination of rs and ra
12 A=rp/(rp+rs);
13 wmo=(1/k)*(A*v-ia1*(A*rs+ra)); // no-load speed
14 wml=(1/k)*(A*v-ia2*(A*rs+ra)); // full-load speed
15 sr=((wmo-wml)/wml)*100; // percent speed regulation
16 printf('Speed regulation for first case is %f
       percent\n',sr);
17 disp('case b');
18 rs=3; // resistance in series with armature
19 wmo=(1/k)*(v-ia1*(rs+ra)); // no-load speed
20 wml=(1/k)*(v-ia2*(rs+ra)); // full-load speed
21 sr=((wmo-wml)/wml)*100; // percent speed regulation
22 printf('Speed regulation for second case is %f
       percent\n',sr);
```

Scilab code Exa 4.44 Determining 1 maximum value of current and corresponding torque 2 ultimate speed and armature current

```
1  clc ;
2  v =200;  // rated voltage of dc shunt motor
3  ra =0.1;  // armature resistance
4  n =1000;  // running speed of motor
5  ia =50;  // armature current at n=1000 rpm
6  re =0.1;  // reduction in field flux
7  disp ('case a ');
8  Ea1 =v - ia * ra;  // initial counter EMF
9  Ea2 = Ea1 *(1 - re );  // counter EMF after reduced field
        flux
10  iam =(v - Ea2 )/ ra ;
11  printf ('Maximum value of armature current is %f A\n '
        ,iam );
12  T =( iam / ia )*(1 - re );
13  printf ('Torque corresponding to maximum armature
        current is %f times initial torque \n ',T );
14  disp ('case b ');
15  ia2 =(1/(1 - re ))* ia ;
16  printf ('Armature current when transients are over is
        %f A\n ',ia2 );
17  Ea2 =v - ia2 * ra ;  // counter EMF when transients are
        over
18  n2 =( Ea2 *n )/( Ea1 *(1 - re ));
19  printf ('Ultimate speed after transients are over is
        %f rpm ',ceil (n2 ));
```

Scilab code Exa 4.45 Determining armature current

```
1  clc ;
2  clc ;
3  v =200;  // rated voltage of dc shunt motor
4  ra =0.1;  // armature resistance
```

176

www.mepcafe.com

```scilab
5  n =1000;  // running speed of motor
6  ia =50;  // armature current at n=1000 rpm
7  re =0.1;  // reduction in field flux
8  ia2 =(1/(1 - re ))* ia ;  // armature current when
       transients are over
9  Ea1 =v - ia2 * ra ;  // counter EMF when transients are
       over
10 // with sudden increase from 0.9* f to f ( f=flux ) ,
       counter EMF rises to
11 Ea2 = Ea1 *(1/(1 - re ));
12 i =(v - Ea2 )/ ra ;
13 printf ('Armature current is %f A',i);
14 disp ('Since armature current is negative , machine
       acts as a generator . Speed reduces till counter
       EMF becomes less than supply voltage , so that
       motor action takes place and torque balance is
       obtained ')
```

Scilab code Exa 4.46 Determining new speed and armature current

```scilab
1  clc ;
2  v =220;  // supply voltage
3  n1 =2000;  // speed of fan motor
4  ia1 =60;  // current corresponding to n=2000 rpm
5  // flux is directly proportional to exciting current
        and load torque increase as square of speed
6  // four field coils are connected in two parallel
       groups also n^2 is directly proportional to
       armature current therefore
7  r= sqrt ((2* ia1 ^2)/ n1 ^2);  // ratio of armature current
        corresponding to n2 and n2 where n2=new speed
8  //  counter EMF are directly proportional to product
        n* ia and ra ( armature resistance ) and rs ( series )
       resistance are not given , therefore takig ratio
       of n1 * ia1 and n2 * ia2 we can determine value of n2
```

177

```
 9  n2=sqrt((ia1*n1*2)/r);
10  printf('New speed is %f rpm\n',n2);
11  ia2=n2*r;
12  printf('New armature current is %f A\n',ia2);
```

Scilab code Exa 4.48 Determining resistance inserted in shunt field

```
 1  clc;
 2  v=230; // rated voltage of dc shunt motor
 3  ra=0.4; // armature circuit resistance
 4  rf=115; // field resistance
 5  n1=800; // initial speed
 6  n2=1000; // final speed
 7  ia1=20; // armature current at n=800 rpm
 8  // torque at both speed is same therefore f1*ia1=f2*
        ia2 where f=field flux therefore
 9  Ea1=v-ia1*ra; // counter EMF at 800rpm
10  // ia2=ia1*k where k=f1/f2 now writing Ea2(counter
        EMF at 1000rpm) in terms of k and finding value
        of k by solving quadratic equation in k whose
        terms are
11  t1=ia1*ra*n1;
12  t2=-v*n1;
13  t3=Ea1*n2;
14  p=[ t1 t2 t3];
15  k=roots(p);
16  if1=v/rf; // initial field current
17  if2=if1/k(2); // final field current correponding to
         n=1000rpm
18  rs=v/if2; // new shunt field circuit resistance
19  re=rs-rf;
20  printf('Resistance that must be inserted in shunt
        field circuit is %f ohms\n',floor(re));
```

178

Scilab code Exa 4.49 Determining new speed of motor

```
1  clc;
2  v=250; // rated voltage of dc shunt motor
3  ra=0.5; // armature resistance
4  rf=250; // field resistance
5  n1=600; // speed of motor
6  i=21; // current drawn by motor when n=600 rpm
7  re=250; // additional resistance in field circuit
8  if1=v/rf; // field current
9  ia=i-if1; // armature current
10 Ea1=v-ia*ra; // counter EMF at n=600 rpm
11 if2=v/(rf+re); // field current after addition of
       external resistance
12 ia2=ia*(if1/if2); // armature current after addition
        of external resistance
13 Ea2=v-ia2*ra; // counter EMF at new speed
14 n2=(n1*Ea2*if1)/(Ea1*if2);
15 printf('New speed of motor is %f rpm',n2);
```

Scilab code Exa 4.50 Determining percentage change in field flux

```
1  clc;
2  v=250; // supply voltage
3  i=50; // current drawn from supply
4  ic=0.4; // percentage increase in speed
5  T=1.2; // ratio of final and initial torque
6  n=1.4;// ratio of final and initial speed
7  ra=0.5; // armature resistance
8  Ea1=v-i*ra; // counter EMF at initial speed
9  // ia2=(T2/T1)*ia1*k where k=f1/f2 and T1 is initial
       torque and T2 is final torque now writing Ea2(
```

179

```
      counter EMF at 1000rpm) in terms of k and finding
       value of k by solving quadratic equation in k
      whose terms are
10  t1=T*ra*i;
11  t2=-v;
12  t3=n*Ea1;
13  p=[ t1 t2 t3 ];
14  k=roots(p);
15  fr=(1-(1/k(2)))*100;
16  printf('Percentage reduction in field flux is %f
       percent',fr);
```

Scilab code Exa 4.51 Determining number of series turns per pole to reduce speed

```
 1  clc;
 2  v=250; // rated voltage of dc shunt motor
 3  n1=1200; // no load speed
 4  N=1000; // turns per pole in shunt field winding
 5  n2=900; // reduced speed
 6  ia=100; // full load armature current
 7  rf=0.2; // series field resistance
 8  ra=0.1; // armature resistance
 9  ar=0.04; // armature reaction as afraction of main
       field m.m.f
10  // At no load counter EMF=v therefore from
       magnetization curve given in fig 4.76 field
       current is
11  IF=[ 0.38 0.58 0.8 1.1 1.36 1.76];
12  EA=[125 180 215 250 275 300];
13  plot(IF,EA);
14  xlabel('field current');
15  ylabel('counter EMF');
16  title('Magnetising curve');
17  ifs1=1.1; // field current
```

180

```
18  Ats1=ifs1*N; // ampere turns for field current
19  Ea2=v-ia*(ra+rf); // counter EMF at full load
20  // magnetization curve is for 1200 rpm, therefore
        full load counter EMF corresponding to it is
21  Ea2=Ea2*(n1/n2);
22  // corresponding to above counter EMF field current
        from magnetization curve is
23  ifs2=1.62;
24  Ats2=(ifs2*N)/(1-ar); // ampere turns for field
        current at full load
25  at=Ats2-Ats1; // series field
26  t=at/ia;
27  printf('Number of series turns per pole to reduce
        speed to %f rpm is %f ',n2,ceil(t));
```

Scilab code Exa 4.52 Determining motor speed

```
1  clc;
2  v=240; // supply voltage
3  n=1000; // speed of motor
4  i=40; // current drawn from supply
5  rf=0.2; // field resistance
6  ra=0.25; // armture resistance
7  rd=0.3; // diverter resistance
8  // torque is constant for different speeds
9  // when diverter is put in parallel with series
        resistance then some fraction of armature current
         flows through series circuit this current for
        constant torque is given by
10  ia2=sqrt(i^2/(rd/(rf+rd)));
11  Ea1=v-i*(ra+rf); // counter EMf at n=1000 rpm
12  Ea2=v-ia2*(ra+((rf*rd)/(rf+rd))); // counter EMF at
        new speed
13  n2=(Ea2*n*i)/(Ea1*(rd/(rf+rd))*ia2);
14  printf('Motor speed after diverter is put in
```

181

www.mepcafe.com

Scilab code Exa 4.53 Determining motor speed and current

```
1  clc ;
2  v =230;  // rated voltage of motor
3  p =6;  // number of poles
4  f =4*10^-3;  // flux per pole in Wb/A
5  T =20;  // load torque
6  n =800;  // speed at T=20 N-m
7  a =2;  // for wave connected conductors number of
          parallel path
8  z =432;  // number of conductors
9  r =1;  // total resistance of motor
10 // it is given that T=kn^2, therefore
11 k =T/n^2;  // proportionality of constant
12 r1 =(f*z*p)/(60*a);  // ratio of back EMF to product
          of speed and armature current
13 t =(r1*60)/(2*%pi);  // ratio of full load torque to
          square of armature current
14 // also Ea(back EMf)=v-ia*ra and r1=Ea/(n*ia)
          compairing both we get ia=v/(1+r1*n);
15 // Also k*n2^2=t*ia^2 , in this expression putting
          value of ia and solving quadratic equation in n2
16 t1 =sqrt(k)*r1 ;
17 t2 =sqrt(k);
18 t3 =-sqrt(t)*v ;
19 p =[ t1 t2 t3 ];
20 n2 =roots(p);
21 printf('Speed of motor is %f rpm\n',n2(2));
22 ia =v/(1+r1*n2(2));
23 printf('Armature current when motor is connected to
          rated supply is %f A',ia);
```

182

Scilab code Exa 4.54 Determining 1 armature current and speed of motor
2 value of external resistance inserted in field winding

```scilab
1  clc;
2  v=230; // rated voltage of dc shunt motor
3  n=1000; // rated speed of motor
4  rf=115; // field resistance
5  ra=0.5; // armature resistance
6  ia=4; // no load armature current
7  k=(v-ia*ra)/(2*%pi*n/60); // constant term in
       formula of back EMF
8  disp('case a');
9  t=80; // load torque
10 ia2=t/k; // armature load current
11 Ea2=v-ia2*ra; // counter EMF corresponding to load
       armature current
12 printf('Armature current for given load is %f A\n',
       ia2);
13 n2=(Ea2*60)/(k*2*%pi);
14 printf('Speed of motor at given load is %f rpm\n',n2
       );
15 disp('case b');
16 pd=8000; // power developed by motor
17 n3=1250; // speed at power is developed
18 // determining value of armature current
       corresponding to power by solving quadratic
       equation whose terms are
19 t1=ra;
20 t2=-v;
21 t3=pd;
22 p=[ t1 t2 t3];
23 ia3=roots(p);
24 Ea3=v-ia3(2)*ra; // counter EMF for load armature
       current
```

183

```
25  k1=k/(v/rf); // constant term in formula of back EMF
        for field current = 1 A
26  ifn=(Ea3*60)/(2*%pi*n3*k1);
27  rfn=v/ifn;
28  printf('External resistance that must be inserted in
        series with field winding is %f ohms',rfn-rf);
```

Scilab code Exa 4.55 Determining current and percentage in flux

```
1  clc;
2  v=250; // rated voltage of dc series motor
3  ra=0.25; // armature resistance
4  rf=0.15; // series field resistance
5  disp('case a');
6  t=80; // developed torque
7  n1=1200; // speed at developed torque
8  // solving quadratic equation in ia
9  t1=ra+rf;
10  t2=-v;
11  t3=(t*2*%pi*n1)/60;
12  p=[ t1 t2 t3];
13  ia=roots(p);
14  printf('Current for developing given torque is %f A\
      n',ia(2));
15  disp('case b');
16  n2=1800;
17  ia2=ia(2)/2;
18  Ea1=v-ia(2)*(ra+rf); // counter EMF corresponding to
        armature current of case 1
19  Ea2=v-ia2*(ra+rf); // counter EMF corresponding to
      armature current ia2
20  fr=(n1*Ea2)/(n2*Ea1); // ratio of fluxes for two
      armatures current
21  pr=(1-fr)*100;
22  printf('Percentage reduction in flux is %f pecent',
```

184

www.mepcafe.com

```
        pr ) ;
```

Scilab code Exa 4.56 Determining additional resistance to obtain rated torque

```
1  clc ;
2  v =230; // rated voltage of dc series motor
3  n =1500; // speed at rated output
4  i =20; // current drawn at rated output
5  ra =0.3; // armature resistance
6  rf =0.2; // field resistance
7  disp ( ' case a ' ) ;
8  // At starting Ea=0, therefore
9  re =( v / i ) -( ra + rf ) ;
10 printf ( ' External resistance to be added in motor
      armature circuit to develop rated torque is %f
      ohms \ n ' , re ) ;
11 disp ( ' case b ' ) ;
12 n2 =1000; // speed at rated torque has to be
      developed
13 Ea2 =( n2 / n ) *( v - i *( ra + rf ) ) ; // counter EMF at n=1000
      rpm
14 re =( v - Ea2 - i *( ra + rf ) ) / i ;
15 printf ( ' External resistance to be added in motor
      armature circuit to develop rated torque is %f
      ohms \ n ' , re ) ;
```

Scilab code Exa 4.57 Determining voltage and current in magnetic circuit

```
1  clc ;
2  // in book voltages are calculated for r=0.5 not for
      r =0.4( as asked in question ) that is why answer
      is differing
```

185

www.mepcafe.com

```
 3  v=230; // supply voltage
 4  n1=800; // speed at supply voltage
 5  i=20; // current drawn from supply
 6  r=0.4; // dc series motor resistance
 7  n2=1000; // raised speed
 8  Ea1=v-i*r; // counter EMF at 800 rpm
 9  disp('case a');
10  // when magnetic circuit is saturated flux is
        constant.Under steady state condition full load
        torque=torque at any load therefore
11  i2=i*(n2/n1)^2; // new current drawn from supply
12  Ea2=Ea1*(n2/n1); // counter EMF at 1000 rpm
13  vt=Ea2+i2*r;
14  printf('Current for saturated magnetic circuit is %f
        A\n',i2);
15  printf('Voltage for saturated magnetic circuit is %f
        V\n',vt);
16  disp('case b');
17  // when magnetic circuit is not saturated flux is
        directly proportional to armature current and
        torque is directly proportional to square of
        armature current
18  i3=(n2/n1)*i;
19  Ea3=(n2*Ea1*i3)/(n1*i); // counter EMF at 1000 rpm
20  vt=Ea3+i3*r;
21  printf('Current for unsaturated magnetic circuit is
        %f A\n',i3);
22  printf('Voltage for unsaturated magnetic circuit is
        %f V\n',vt);
```

Scilab code Exa 4.58 Calculating armature current for different control methods

```
 1  clc;
 2  s=4; // speed range
```

186

```
3  ia=60; // armature current at speed n
4  disp('Field flux control');
5  // For constant power load  Ea*Ia is constant
       therefore ia is conatant at 4*n
6  printf('The armature current at required speed is %f
       A\n',ia);
7  // For constant torque load , speed is 4 times of
       initial speed therefore flux changes by 1/4 times
        and hence to maintain torque constant armature
       current should be four times
8  printf('The armature current at required speed is %f
       A\n',4*ia);
9  disp('Armature voltage control');
10 // For constant power load  Ea*Ia is constant
       therefore at 4 times speed armature voltage is 4
       times and the armature current gets reduced by
       1/4 times
11 printf('The armature current at required speed is %f
       A\n',ia/4);
12 // For constant power load  Ea*Ia is constant
       therefore at 4 times speed ,flux is constant
       therefore
13 // armature current is constant
14 printf('The armature current at required speed is %f
       A\n',ia);
```

Scilab code Exa 4.59 Determining speed range for full load and no load
2 minimum motor field current

```
1  clc;
2  v=220; // rated voltage of motor
3  i=15; // rated current of motor
4  ra=0.4; // net armature resistance
5  n=1500; // speed for which magnetization curve is
       given
```

187

```
6  IF=[ 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.2 1.45];
7  EA=[120 160 197 210 220 228 232 236 243 248];
8  plot(IF,EA);
9  xlabel('field current');
10 ylabel('voltage');
11 title('magnetising curve');
12 disp('case a');
13 ifg1=0.15; // initial generator field current
14 ifg2=1.4; //  final generator field current
15 ifm=0.6; // motor field current
16 // corresponding ifg1 counter EMF of generator from
       magnetization curve is
17 Ea1=60;
18 vd=i*(ra+ra); // voltage drop in two armature
       resistance
19 Ea2=Ea1-vd; // counter EMF of motor
20 // but motor counter EMF for 0.6 A field current at
       1500 rpm is
21 Ea3=210;
22 nmin1=(Ea2/Ea3)*n; // minimum motor speed
23 // corresponding ifg2 counter EMF of generator from
       magnetization curve is
24 Ea4=247;
25 Ea5=Ea4-vd; // counter EMF of motor
26 nmax1=(Ea5/Ea3)*n; // maximum motor speed
27 sr=nmax1/nmin1; // speed range
28 printf('Speed range for full load armature current
       is %f:1\n',sr);
29 // for no load generator counter EMF= motor counter
       EMf
30 nmin2=(Ea1/Ea3)*n; // minimum motor speed
31 pr=((nmin2-nmin1)/nmin2)*100;
32 printf('Percent speed drop from no load to full load
        for condition of minimum speed is %f percent\n',
     pr);
33 // for maximum generator field current generator
       counter EMF= motor counter EMf at no load
34 nmax2=(Ea4/Ea3)*n; // maximum motor speed
```

188

```
35  sr=nmax2/nmin2; // speed range
36  printf('Speed range for full load armature current
        is %f:1\n',sr);
37  pr=((nmax2-nmax1)/nmax2)*100;
38  printf('Percent speed drop from no load to full load
         for condition of maximum speed is %f percent\n',
        pr);
39  disp('case b')
40  // for generator field current=1 A counter EMF from
        magnetization curve is
41  Ea6=236;
42  Ea7=Ea6-vd; // motor counter EMF at full load
43  Ea8=(Ea7/(2*nmax1))*n;
44  printf('Motor counter EMF for %f rpm is %f V\n',n,
        Ea8);
45  // Corresponding to Ea8, field current is
46  ifmi=0.25;
47  printf('Minimum motor field current is %f A\n',ifmi)
        ;
```

Scilab code Exa 4.60 Determining motor torque and motor speed

```
1  clc;
2  p=4000; // rated power of separately excited dc
        series motor
3  v=230; // rated voltage of motor
4  n=1000; // rated speed of motor
5  e=260; // ac voltage supplied to motor through
        converter
6  i=2; // current drawn at no load
7  no=1100; // speed at no load
8  ra=0.5; // armature resistance
9  vd=2; // voltage drop in thyristor
10  de=30; // firing angle delay
11  ia=20; // rated armature current
```

189

```
12 k=((((2*sqrt(2)*e)/%pi)-(i*ra)-vd)*60)/(2*%pi*no);
       // conatant term in formula of back EMf
13 disp('case a');
14 printf('Motor torque is %f Nm\n',k*ia);
15 disp('case b');
16 wm=((((2*sqrt(2)*e*cosd(de))/%pi)-vd)/k)-((ra*ia)/k)
     ;
17 printf('Motor speed is %f rpm',ceil((wm*60)/(2*%pi))
     );
```

Scilab code Exa 4.61 Determining 1 speed and current for torque 2 speed and torque for given armature current 3 speed and torque for given armature current

```
1 clc;
2 n=800; // speed at which magnetization curve is
      given
3 // case a
4 v=600; // dc voltage source
5 ra=0.3; // armature resistance
6 rf=0.25; // field resistance
7 T=300; // given torque
8 VT=[ 200 375 443 485 510 518]; // terminal voltage
9 IF=[ 15 30 45 60 75 90 ]; // field current
10 EA1=VT+IF*ra; // generated EMF
11 EA2=v-IF*(ra+rf); // generated EMF for v=600 V
12 N2=n*(EA2./EA1); // speed for v= 600 V
13 TE=(EA2.*IF.*EA1*60)./(2*%pi*n*EA2); //torque
14 subplot(221);
15 plot(TE,N2);
16 xlabel('Torque(Nm)');
17 ylabel('speed(rpm)');
18 title('speed-torque');
19 subplot(222);
20 plot(TE,IF);
```

190

```
21  xlabel('Torque(Nm)');
22  ylabel('current(A)');
23  title('current-torque');
24  disp('from curves, for a torque of 300 Nm, speed is
       940 rpm and current is 52.5 A');
25  disp('case b');
26  rd=0.25; // diverter resistance put in parallel with
        series combination of armature and field
       resistance
27  ia1=30;
28  ia2=60; // armature currents
29  if1=ia1*(rd/(rd+rf)); // field current corresponding
        to ia1
30  Ea1=204.5; // given counter EMF for field current
31  Ea2=v-ia1*(ra+((rd*rf)/(rd+rf))); // counter EMF for
        voltage supply of 600 V
32  n2=n*(Ea2/Ea1);
33  printf('Speed at %f A armature current is %f rpm\n',
       ia1,n2);
34  T=(Ea1*60*ia1)/(2*%pi*n);
35  printf('Torque at %f A armature current is %f Nm\n',
       ia1,T);
36  if1=ia2*(rd/(rd+rf)); // field current corresponding
        to ia1
37  Ea1=384; // given counter EMF for field current
38  Ea2=v-ia2*(ra+((rd*rf)/(rd+rf))); // counter EMF for
        voltage supply of 600 V
39  n2=n*(Ea2/Ea1);
40  printf('Speed at %f A armature current is %f rpm\n',
       ia2,n2);
41  T=(Ea1*60*ia2)/(2*%pi*n);
42  printf('Torque at %f A armature current is %f Nm\n',
       ia2,T);
43  disp('case c');
44  ia3=75; // armature current
45  t=0.8; // tapping percentage of field winding as a
       fration of full series turn
46  ifl=t*ia3; // corresponding field current
```

191

```
47  Ea=503; // given counter EMF for field current
48  Ea2=v-ia3*(ra+t*rf); // counter EMF for voltage
       supply of 600 V
49  n2=n*(Ea2/Ea);
50  printf('Speed at %f A armature current is %f rpm\n',
       ia3,n2);
51  T=(Ea*60*ia3)/(2*%pi*n);
52  printf('Torque at %f A armature current is %f Nm\n',
       ia3,T);
```

Scilab code Exa 4.62 Determining size of DC motor and hoist speed

```
 1  clc;
 2  ra=0.3; // armature resistance
 3  n=0.7; // efficiency of dc shunt motor
 4  l=800; // weight of load
 5  v=3; // speed of raising a load
 6  vi=230; // initial value of supply voltage
 7  vf=190; // final value of supply voltage
 8  g=9.81; // acceleration due to gravity
 9  f=l*g; // resisting force due to gravitational pull
10  p=f*v; // power required for lifting the load
11  P=p/n; // power rating of dc machine
12  printf('Required rating of power is %f KW\n',P/1000)
       ;
13  // for supply voltage of 230 V  Ea=230-ia*ra finding
        quadratic equation in ia whose terms are
14  t1=ra;
15  t2=-vi;
16  t3=P;
17  p=[ t1 t2 t3 ] ;
18  ia=roots(p);
19  Ea=vf-ia(2)*ra; // counter EMF for supply voltage of
        190 V
20  v=(Ea*ia(2)*n)/(l*g);
```

192

```
21 printf('New hoist speed is %f m/s',v);
```

Scilab code Exa 4.63 Determining additional resistance inserted in motor circuit

```
1 clc;
2 v1=6; // hoist speed
3 i=60; // series current
4 v=600; // supply voltage
5 r=0.5; // net resistance
6 g=9.81; // acceleration due to gravity
7 v2=4; // reduced hoist speed
8 Ea1=v-i*r; // counter EMf corresponding to v1
9 rx=((v-((v2/v1)*Ea1))/i)-r;
10 printf('External resistance to be added is %f ohms',
      rx);
```

Scilab code Exa 4.64 Determining speed of motor for given condition

```
1 clc;
2 v=450; // supply voltage
3 i=25; // current drawn from supply
4 n=600; // full load speed
5 z=500; // number of conductors
6 f=1.7*10^-2*sqrt(i); // flux per pole
7 p=4; // number of poles
8 a=p; // number of parallel paths for wave wound
      winding is same as number of poles
9 Ea1=(f*n*z*p)/(60*a); // counter EMF
10 ra=(v-Ea1)/i; // armature resistance
11 // T=k*f*ia    where f is flux and ia is armature
      current As per question new torque is half of
      initial torque
```

193

```
12  i2=((i^1.5)/2)^(1/1.5);  // new armature current
13  Ea2=(v/2)-i2*ra;  // counter EMF for new armature
       current
14  n2=(Ea2*f*sqrt(i)*n)/(Ea1*f*sqrt(i2));
15  printf('New speed at which motor will run is %f rpm'
       ,floor(n2));
```

Scilab code Exa 4.66 Determining motor speed and armature current

```
1  clc;
2  // answer is calculated for torque=30 but it is
       asked for torque=40 i.e why answer varies
3  p=4;  // number of dc series motor
4  f=4*10^-3;  // ratio of flux per pole to armature
       current
5  T=40;  // torque of fan
6  n=1000;  // speed of motor
7  a=2;  // number of parallel path for waave winding
8  z=480;  // number of conductors
9  ra=1;  // armature resistance
10 v=230;  // supply voltage
11 re=sqrt((T*2*%pi*a)/(p*z*f*n^2));  // ratio of
       armature current and new speed
12 // Ea=vt-ia*ra writing ia in terms of n solving for
       n (n is new speed)
13 n2=v/(re+((p*f*z)/(60*a)));
14 printf('Motor speed is %f rpm\n',n2);
15 ia=re*n2;
16 printf('Armature current is %f A',ia);
```

Scilab code Exa 4.67 Determining shaft power input efficiency at rated load maximum efficiency and power output

194

```
1  clc;
2  p=10000; // rated power of generator
3  v=250; // rated voltage of generator
4  l1=400; // rotationl losses
5  ra=0.5; // armature resistance
6  rf=250; // shunt field resistance
7  ifl=v/rf; // constant field current
8  lc=ifl*rf+l1; // constant losses
9  io=p/v; // output current of generator
10 ia=io+ifl; // armature current
11 la=ia^2*ra; // armature circuit loss
12 ps=p+lc+la; // generator shaft power input
13 printf('Generator shaft power input is %f W\n',ps);
14 ng=(1-((lc+la)/ps))*100;
15 printf('Efficiency at rated load is %f percent\n',ng
      );
16 // at maximum efficiency variable losses= constant
      losses
17 ia=sqrt(lc/ra); // armature current at maximum
      efficiency
18 io=floor(ia)-ifl; // output current of generator
19 po=v*io; // output power
20 printf('Generator output at maximum efficiency is %f
       W\n',po);
21 pi=po+2*lc;
22 nm=(1-((lc+lc)/pi))*100;
23 printf('Maximum efficiency is %f percent\n',nm);
```

Scilab code Exa 4.68 Determining efficiency and speed of motor

```
1  clc;
2  v=250; // rated voltage of shunt motor
3  p=15000; // rated power of motor
4  nm=0.88; // maximumu efficiency of motor
5  n=700; // speed of motor
```

```
 6  rf=100; // resistance of shunt field
 7  i=78; // current drawn by mains
 8  f=0.8; // fraction of rated output being delivered
 9  l=((1/nm)-1)*f*p; // total losses
10  // at maximum losses constant losses= variable
       losses
11  lc=l/2; // constant losses
12  pi=f*p+l; // input to motor at maximum efficiency
13  il=pi/v; // input line current
14  ia=il-(v/rf); // armature current
15  ra=lc/ia^2; // armature resistance
16  ia2=i-(v/rf); // armature current at given load
17  pi=i*v; // total power input
18  tl=ia2^2*ra+lc; // total losses
19  n1=(1-(tl/pi))*100; // efficiency at line current of
       75 A
20  Ea1=v-ia*ra; // counter EMF
21  Ea2=v-ia2*ra; // counter EMF corresponding to line
       current of 75 A
22  // field current is constant so flux is constant
23  n2=(Ea2/Ea1)*n;
24  printf('Efficiency at line current of %d A is %f
       percent\n',i,ceil(n1));
25  printf('Speed at line current of %d A is %f rpm',i,
       floor(n2));
```

Scilab code Exa 4.69 Determining 1 no load current 2 speed 3 armature current

```
1  clc;
2  p=10000; // rated power of transformer
3  n=900; // speed of motor
4  v=400; // rated voltage of motor
5  ra=1; // armatyre resistance
6  rf=400; // field resistance
```

196

www.mepcafe.com

```
7  ne=0.85; // efficiency at rated load
8  l=((1/ne)-1)*p; // total losses
9  disp('case a');
10 pi=p+l; // power input
11 il=pi/v; // line current
12 ia=il-(v/rf); // armature current
13 la=ia^2*ra; // armature circuit losses
14 lf=v*(v/rf); // shunt field losses
15 wo=l-la-lf; // no load losses
16 iao=wo/v; // no load current neglecting armature
       losses at no load
17 printf('No load armature current is %f A\n',iao);
18 disp('case b');
19 Ea1=v-ia*ra; // counter EMF at rated load
20 il=20; // current drawn by motor
21 ia=il-(v/rf); // armature current
22 Ea2=v-ia*ra; // counter EMF at line current of 20 A
23 n2=(Ea2/Ea1)*n;
24 printf('Speed of motor while drawing current of %d A
        from mains is %f rpm\n',il,ceil(n2));
25 disp('case c');
26 k=(Ea1*60)/(2*%pi*n); // constant term in counter
     EMF formula
27 T=98.5; // electromagnetic torque
28 ia=T/k;
29 printf('Armature current at given torque is %f A',
     ceil(ia));
```

Scilab code Exa 4.70 Determining efficiency of motor

```
1 clc;
2 v=240; // rated voltage of motor and supply voltage
3 i=5.2; // line current
4 p=10000; // rated power of motor
5 no=1200; // no load speed
```

```
6  ra=0.25; // armature resistance
7  rf=160; // field resistance
8  ifl=v/rf; // constant field current
9  iao=i-ifl; // no load armature current
10 wo=v*iao-iao^2*ra; // no load rotational losses
11 //  by using equation of electromagnetic power
       solving quadratic equation in armature current
       whose terms are
12 t1=ra;
13 t2=-v;
14 t3=p+wo;
15 P=[ t1 t2 t3 ];
16 ia=roots(P);
17 pi=(v-ia(2)*ra)*ia(2)+ia(2)^2*ra+ifl*v; // motor
       input
18 nm=(p/pi)*100;
19 printf('Motor efficiency at rated load is %f percent
       ',nm);
```

Scilab code Exa 4.71 Determining armature voltage drop

```
1  clc;
2  v=440; // rated voltage of mootor
3  no=2000; // no load speed
4  n1=1000; // speed at full load torque
5  Tl=0.5; // load torque as a fraction of rated torque
6  n2=1050; // increased speed due to redued torque
7  // field current is constant so flux is constant
8  // since torqu gets reduced by half new armature
       current also gets reduced half i.e ia2=ia1/2;
9  vd=(v*(n2-n1))/(n2-(n1/2));
10 printf('Armature voltage drop at full load is %d V',
       vd);
```

Scilab code Exa 4.72 Determining shaft torque and efficiency of motor

```
1  clc;
2  v=230; // source voltage
3  ra=0.1; // resistance of armature
4  ia=100; // armature current
5  n=1600; // speed of dc shunt motor
6  wl=300; // friction and windage losses
7  lo=1200; // no load core loss
8  lc=2500; // copper losses
9  Ls=0.01; // stray losses as a fraction of output
10 Ea=v-ia*ra; // counter EMF
11 pe=Ea*ia; // electromagnetic power
12 wo=wl+lo; // no load rotational losses
13 po=pe-wo; // shaft power + stray load losses
14 psh=po/(1+Ls);
15 Tsh=(psh*60)/(2*%pi*n);
16 printf('Shaft torque is %f Nm\n',Tsh);
17 pi=pe+lc; // power input to motor
18 nm=(psh/pi)*100;
19 printf('Motor efficiency is %f percent',nm);
```

Scilab code Exa 4.73 Determining shaft torque shaft power and motor efficiency

```
1  clc;
2  // shaft power is given little bit more than actual
        value in question
3  w1=25;
4  w2=9; // spring balance readings in kg
5  d=19.5*10^-2; // outside pulley diameter
6  t=0.5*10^-2; // belt thickness
```

199

```
 7  g=9.81; // acceleration due to gravity
 8  n=1500; // motor speed
 9  v=230; // applied voltage
10  il=12.5; // line current
11  Ts=(w1-w2)*((d/2)+(t/2))*g;
12  printf('Shaft torque is %f Nm\n',Ts);
13  psh=(2*%pi*n*Ts)/60;
14  printf('Shaft power is %f W\n',psh);
15  pi=v*il; // motor input
16  nm=(psh/pi)*100;
17  printf('Motor efficiency at rated load is %f percent
        ',nm);
```

Scilab code Exa 4.74 Determining KW output efficiency and percentage change in speed from no load to full load

```
 1  clc;
 2  v=400; // rated voltage of dc shunt motor
 3  io=5; // current at no load
 4  ra=0.5; // armature resistance
 5  rf=200; // field resistance
 6  i=50; // current at full load
 7  ifl=v/rf; // constant shunt field current
 8  iao=io-ifl; // no load armature current
 9  wo=v*iao-iao^2*ra; // no load rotational losses
10  ia=i-ifl; // full load armature current
11  la=ia^2*ra; // full load armature circuit losses
12  lf=v*ifl; // constant shunt feld losses
13  tl=la+lf+wo; // total field losses
14  pi=i*v; // motor input at full load
15  nm=(1-(tl/pi))*100;
16  printf('Output power is %f KW\n',(pi-tl)/1000);
17  printf('Efficiency on full load is %f percent\n',nm)
        ;
18  Ea1=v-iao*ra; // no load counter EMF
```

```
19  Ea2=v-ia*ra; // full load counter EMF
20  pr=((Ea1-Ea2)/Ea1)*100; // Ea is directly
        proportioal to speed so percentage change in Ea
        is same as percentage in speed;
21
22  printf('Percentage change in speed from no load to
        full load is %f percent',pr);
```

Scilab code Exa 4.75 Determining full load efficiency of motor

```
1   clc;
2   v=400; // rated voltage of dc shunt motor
3   p=20000; // rated power of motor
4   i=2.5; // no load current
5   ra=0.5; // armature resistance
6   rf=800; // field current
7   vb=2; // voltage drop in brush
8   ifl=v/rf; // constant shunt field current
9   iao=i-ifl; // no load armature current
10  wo=v*iao-iao^2*ra; // no load rotational losses
11  tl=wo+v*ifl; // total losses
12  //  by using equation of power input= output power +
        losses, solving quadratic equation in armature
        current whose terms are
13  t1=ra;
14  t2=vb-v;
15  t3=p+tl-v*(v/rf);
16  P=[ t1 t2 t3];
17  ia=roots(P);
18  lo=ia(2)^2*ra; // armature ohmic losses
19  lb=ia(2)*vb; // brush drop loss
20  tl=tl+lo+lb; // total losses at rated load
21  pi=p+tl; // input power
22  nm=(p/pi)*100;
23  printf('Full load efficiency is %f percent',nm);
```

201

Scilab code Exa 4.77 Determining efficiency of both machines

```
1  clc;
2  // Hopkinson's method gave following result for two
       identical dc shunt machines
3  v=230; // line voltage
4  il=30; // line current excluding both field currents
5  ia=230; // motor armature current
6  ifl1=4; ifl2=5;  // field currents
7  ra=0.025; // armature current
8  // from fig 4.85
9  ig=ia-il; // generator armature current
10 la1=ig^2*ra; // armature circuit losses in generator
11 la2=ia^2*ra; // armature circuit losses in motor
12 pd=v*il; // power drawn from supply (excluding field
       loss)
13 wo=pd-la1-la2; // no load rotational losses for both
       machines
14 pg=v*ig; // generator outputk
15 tl=(wo/2)+v*ifl2+la1; // total losses for generator
16 ng=(1-(tl/(tl+pg)))*100;
17 pi=v*(ia+ifl1); // input power for motor
18 tl=(wo/2)+v*ifl1+la2; // total losses for motor
19 nm=(1-(tl/pi))*100;
20 printf('Motor efficiency is %f percent\n',nm);
21 printf('Generator efficiency is %f percent\n',ng);
22 // If both machine are assumed to have same
       efficiency then
23 n=sqrt(ig/ia)*100;
24 printf('Efficiency of machine is %d percent',n);
```

Scilab code Exa 4.78 Determining efficiency of both machines

```
1  clc;
2  // fields test on two similar machine gave following
       test
3  iam=60; // motor armature current
4  vam=500; // voltage across armature
5  vfm=40; // voltage across field
6  vt=450; // terminal voltage for generator
7  io=46; // output current for generator
8  vfg=40; // voltage across field
9  ra=0.25; // armture resistance
10 pi=(vam+vfm+vfg)*iam; // power input to whole set
11 pog=vt*io; // generator output
12 tl=pi-pog; // total loss in whole set
13 poh=iam^2*ra+iam*(vfm+vfg)+io^2*ra; // total ohmic
       losses
14 wo=(tl-poh)/2; // no load roational losses for each
       machines
15 pim=(vam+vfm)*iam; // motor power input
16 plm=iam^2*ra+iam*vfm+wo; // total motor loss
17 nm=(1-(plm/pim))*100;
18 printf('Motor efficiency is %f percent\n',nm);
19 plg=io^2*ra+iam*vfm+wo; // total motor loss
20 pgm=pog+plg; // generator input
21 ng=(1-(plg/pgm))*100;
22 printf('Generator efficiency is %f percent',ng);
```

Scilab code Exa 4.81 Determining field current and power gain at rated output and when compensation is zero

```
1  clc;
2  p=3000; //  power of amplidyne
3  v=300; // voltage of amplidyne
4  w=200; // angular velocity of amplidyne
5  rf=50; // field resistance
6  ra=5; // armature resistance
```

203

```
7  rc =1;  // compensating winding resistance
8  kqf =250;
9  kdq =100;
10 kqd =80;  // voltage constants
11 A=(kdq*kqf)/(ra*rf);  // voltage amplification factor
12 id=p/v;  // rated current
13 vf=(v+id*(ra+rc))/A;  // field voltage
14 ifl=vf/rf;  // field current
15 pk=(v*id)/(vf*ifl);  // power gain
16 printf('Field current is %f A\n',ifl);
17 printf('Power gain at rated output is %f \n',pk);
18 // when compensation is zero
19 vf=(v+id*(((kdq*kqd)/ra)+ra))/A;  // field voltage
20 ifl=vf/rf;
21 printf('Field current at zero compensation is %f A\n
       ',ifl);
22 pk=(v*id)/(vf*ifl);  // power gain
23 printf('Power gain at rated output at zero
       compensation is %f \n',pk);
```

Scilab code Exa 4.82 To plot the external characteristics for given field current

```
1  clc;
2  // plot for open circuit characteristics is given in
       fig 4.10
3  IF=[ 0 11.5 23 36.5 59.5 79 110 160];
4  EA=[0 40 80 120 160 180 200 220 ];
5  subplot (221);
6  plot(IF,EA);
7  xlabel('field ATs');
8  ylabel('voltage');
9  title('magnetising curve');
10 nf=800;  // field winding turns
11 rd=0.5;  // total armature resistance along d-axis
```

```
12  ifl=0.2; // field winding current
13  d=10; // product of ( difference between mmf of
        compensating winding and armature mmf along d-
        circuit )and load current
14  nf1=nf*ifl; // field winding turns for field current
        of 200mA
15  il=nf1/d; // maximum load current
16  printf('Maximum field current is %d A\n',il);
17  IL=[0 2 4 6 8 10 12 14 16]; // load currents
18  ATD=nf1-d*IL; // net d-axis ATs
19  disp('Net d-axis ATs is ');
20  disp(ATD);
21  // corresponding to each ATD open circuit EMF is
        obtained from magnetising curve
22  EO=[220 213 204.7 194 180.5 161.4 128 70 0 ]; //
        open circuit EMF
23  VRD=rd*IL; // d-axis resistance drop
24  VO=EO-VRD;
25  disp('Output voltage(V) is ');
26  disp(VO);
27  subplot(222);
28  plot(IL,VO);
29  xlabel('load current(A)');
30  ylabel('Output voltage(v)');
31  title('Output voltage vs Load current');
```

Scilab code Exa 4.83 Determining field winding current

```
1  clc;
2  // from fig 4.79
3  vo=206; // output voltage
4  il=8; // load current
5  ifl=0.2; // field current
6  Eo=280; // open circuit voltage for which field
        winding current is to be determined
```

```
7  r=0.5; // net resistance
8  n=800; // d-axis ampere turns
9  // with saturation ignored output voltage vd is
       given by vd=(n*if-10*il)K-il*r
10 K=(vo+il*r)/(800*ifl-10*il); // slope of straight
       line in curve
11 ifl=Eo/(K*n);
12 printf('For given open circuit voltage field current
        is %f mA',ifl*1000);
```

Scilab code Exa 4.84 Determining output voltage of generator and reference voltage of potentiometer

```
1  clc;
2  A=100; // amplidyne voltage amplification
3  vo=200; // DC generator output voltage
4  rf=125; // field winding resistance
5  vfb=0.1; // ratio of feedback voltage to output
       voltage of generator
6  vr=50; // reference voltage
7  // amplidyne output voltage ,Va =(vr-vfb*vt)*A
8  // ig=va/rf ig is generator field current
9  // vog=ig*vo vog is generator output voltage-1
10 // simplifying 1 we get
11 vt=(vr*A)/((vfb*A)+(rf/vo));
12 printf('Output voltage of generator is %f V\n',vt);
13 // now feedback voltage is reduced to zero
14 vr=(vt*rf)/(A*vo);
15 printf('Reference voltage to obtain required output
       generator voltage is %f V ',vr);
```

Scilab code Exa 4.85 Determining reference voltage of potentiometer

```
1  clc ;
2  g1 =1.5; // gain factor of amplifier
3  g2 =80; // gain factor of generator
4  vo =250; // output voltage at no load
5  s=0.2; // feedback potentiometer setting
6  // for generated voltage= 80V field current is 1 A
7  ifl=vo/g2; // field current for generated voltage=
      250V
8  vi=ifl/g1; // amplifier input voltage for field
      current corresponding to generated voltage= 250V
9  vfb=s*vo; // feedback voltage
10 vr=vfb+vi ;
11 printf('Reference voltage for given potentiometer
      setting is %f V\n',vr);
12 printf('When feedback setting is zero, reference
      voltage is %f V',vi);
```

Scilab code Exa 4.86 Determining amplifier gain

```
1  clc ;
2  p=4000; // rated power of generator
3  v=250; // rated voltage of generator
4  ra=0.25; // armature resistance
5  rf=100; // fiel resistance
6  vr=20; // improving factor for voltage regulation
7  g1=120; // generator gain
8  // after deriving required expression
9  il=p/v; // load current
10 vgr=((il*ra)/v)*(1/vr); // pu full load generator
      regulation
11 dvt=-vgr*v; // decrease in terminal voltage of
      generator from no load to full load
12 disp('case a');
13 s=0.1; // feedback potentiometer setting
14 A=(-dvt*rf-il*ra*rf)/(dvt*s*g1);
```

207

```
15  printf('Amplifier gain is %f\n',A);
16  disp('case b');
17  s=1; // feedback potentiometer setting
18  A=(-dvt*rf-il*ra*rf)/(dvt*s*g1);
19  printf('Amplifier gain is %f\n',A);
```

Scilab code Exa 4.87 Determining 1 no load rotational losses 2 motor output 3 stall torque

```
1  clc;
2  v=48; // supply voltage
3  n=2400; // speed of permanent magnet DC motor
4  i=0.8; // current drawn by motor
5  ra=1; // armature resistance of motor
6  disp('case a');
7  Ea=v-i*ra; // generated EMF
8  l=Ea*i;
9  printf('No load rotational losses is %f W\n',l);
10 disp('case b');
11 km=(Ea*60)/(2*%pi*n); // speed voltage constant
12 v=40; // supply voltage
13 n1=1600; // speed at supply voltage
14 Ea=(km*2*%pi*n1)/60; // generated EMF
15 ia=(v-Ea)/ra; // new armature current
16 pe=Ea*ia; // Electromagnetic power developed
17 po=pe-l;
18 printf('Output power is %f W\n',po);
19 disp('case c');
20 v=20; // supply voltage
21 // when motor stalls Ea=0
22 ia=v/ra; // stall current
23 T=km*ia;
24 printf('Stall torque is %f Nm',T);
```

208

Chapter 5

Polyphase synchronous machine

Scilab code Exa 5.1 Determining percentage voltage regulation of alternator by different methods

```
1  clc ;
2  v =220; // rated voltage of alternator
3  f =50; // frequency of supply
4  r =0.06; // resistance per phase
5  p =6; // number of poles
6  i =40; // full load current
7  pf =0.8; // lagging power factor
8  vt = v / sqrt (3) ; // rated per phase voltage
9  IF =[ 0.2 0.4 0.6 0.8 1 1.2 1.4 1.8 2.2 2.6 3 3.4];
10 EA =[ 29 58 87 116 146 172 194 232 261.5 284 300
       310];
11 subplot (313) ;
12 plot ( IF , EA / sqrt (3) );
13 xlabel ( 'Field current ') ;
14 ylabel ( 'open circuit voltage ') ;
15 title ( 'open circuit characteristics ') ;
16 IF1 =[ 0.2 0.4 0.6 0.8 1 1.2 1.4 1.8 ];
17 ISC =[ 6.6 13.2 20 26.5 32.4 40 46.3 59 ];
```

209

```scilab
18  subplot(323);
19  plot(IF1,ISC);
20  xlabel('Field current');
21  ylabel('short circuit current');
22  title('short circuit characteristics');
23  ZPF=[ 0 0 0 0 0 0 29 88 140 177 208 230];
24  subplot(333);
25  plot(IF,ZPF);
26  xlabel('Field current');
27  ylabel('terminal voltage');
28  title('full load zero power factor characteristics')
       ;
29  disp('EMF method');
30  // value of synchronous reactance is taken from
       given table
31  EA1=[ 29 58 87 116 146 172 194 232]
32  ZS=EA1./(ISC*sqrt(3));
33  disp('synchronous impedance (ohms) is');
34  disp(ZS);
35  XS=ZS; // RS^2 is negligible
36  disp('synchronous reactance (ohms) is');
37  disp(XS);
38  xs=2.27;
39  ia=i*(pf-%i*sqrt(1-pf^2)); // full load current in
       complex form
40  E=vt+ia*(r+%i*xs); // Excitation voltage
41  vr=floor(((abs(E)-vt)/vt)*100);
42  printf('Voltage regulation is %f percent\n',vr);
43  disp('Mmf method');
44  // with ia as reference
45  E=vt*(pf+%i*sqrt(1-pf^2))+i*r; // Excitation voltage
46  // from fig 5.30 ,E=127 V
47  oc=1.69; // current for given excitation voltage
       obtained from open circuit characteristics
48  sc=1.2; // field current required to circulate full
       load short circuit current
49  al=atand(imag(E),real(E)); // angle between ia and E
50  Ff=(oc*(-sind(al)+%i*cosd(al)))-sc; // field mmf
```

```
51  printf('field mmf is %f A\n',abs(Ff));
52  // corresponding to Ff,E=163.5 v from O.C.C
53  Ef=163.5;
54  vr=((Ef-vt)/vt)*100;
55  printf('Voltage regulation is %f percent\n',vr);
56  disp('Zero power factor method');
57  // As per the description given in method
58  vd=30; // voltage drop armature leakage reactance
59  xa=vd/i; // armature leakage reactance
60  // with ia as reference
61  Er=vt*(pf+%i*sqrt(1-pf^2))+i*(r+%i*xa); //
        Excitation voltage
62  // from fig 5.30 ,E=148.6 V
63  oc=2.134; // current for given excitation voltage
        obtained from open circuit characteristics
64  Fa=0.84; // armature mmf from potier triangle
65  be=atand(imag(Er),real(Er)); // angle between ia and
         E
66  Ff=(oc*(-sind(be)+%i*cosd(be)))-Fa; // field mmf
67  printf('field mmf is %f A\n',abs(Ff));
68  // corresponding to Ff=2.797 A,E=169 v from O.C.C
69  Ef=169;
70  vr=((Ef-vt)/vt)*100;
71  printf('Voltage regulation is %f percent\n',vr);
72  disp('New A.S.A method');
73  // parameters needed in this method are calculated
        in part c
74  id=0.366; // difference in field current between OCC
         and air gap line from fig 5.30
75  th=acosd(pf);
76  ig=1.507; // field current corresponding to rated
        rated per phase voltage
77  Ff=ig+sc*(%i*pf+sqrt(1-pf^2)); // field mmf without
        saturation
78  Ff=abs(Ff)+id; // ield mmf with saturation
79  printf('field mmf is %f A\n',Ff);
80  // corresponding to Ff=2.791 A,E=169 v from O.C.C
81  Ef=169;
```

211

```
82 vr=((Ef-vt)/vt)*100;
83 printf('Voltage regulation is %f percent\n',vr);
84 disp('Saturated synchronous reactance method');
85 // for E=148.5 v (from part c),
86 Era=179.5; // air line gap voltage
87 k=Era/abs(Er); // saturation factor
88 vdg=100.5; // voltage drop in unsaturated
      synchronous reactance
89 xag=vdg/i; // unsaturated synchronous reactance
90 xas=xa+((xag-xa)/k); // saturated synchronous
      reactance
91 // with vt as reference
92 Ef=vt+ia*(r+%i*xas);
93 ok=2.15; // resultant mmf from fig 5.30
94 Ff=(abs(Ef)/abs(Er))*ok;
95 printf('field mmf is %f A\n',Ff);
96 // corresponding to Ff=2.78 A,E=169 v from O.C.C
97 Ef=169;
98 vr=((Ef-vt)/vt)*100;
99 printf('Voltage regulation is %f percent\n',vr);
```

Scilab code Exa 5.3 Determining shaft power line current pf and efficiency for maximum power output and maximum power input

```
1 clc;
2 e=400; // rated voltage of motor
3 E=510; // excitation emf
4 zs=0.5+%i*4; // synchronous impedance per phase
5 l=900; // net loss
6 al=90-atand(imag(zs),real(zs));
7 disp('case a');
8 Pmax=((e*E)/abs(zs))-((E^2*real(zs))/abs(zs)^2); //
      maximum output power
9 sp=Pmax*3-l;
10 printf('Shaft power is %f W\n',sp);
```

```
11  ia=(sqrt(e^2+E^2-2*e*E*cosd(atand(imag(zs),real(zs))
       )))/abs(zs);
12  il=sqrt(3)*ia; // line current
13  printf('Line current is %f A\n',il);
14  di=acosd((e*abs(zs)-E*real(zs))/(ia*abs(zs)^2));
15  pf=cosd(atand(imag(zs),real(zs))-di);
16  printf('Power factor is %f lagging\n',pf);
17  disp('case b');
18  Pmax=((e*E)/abs(zs))+((e^2*real(zs))/abs(zs)^2); //
       maximum input power
19  ia=(sqrt(e^2+E^2-2*e*E*cosd(90+al)))/abs(zs);
20  sp=floor(Pmax*3-ia^2*real(zs)*3-900);
21  printf('Shaft power is %f W\n',sp);
22  il=sqrt(3)*ia; // line current
23  printf('Line current is %f A\n',il);
24  di=acosd((e+E*cosd(atand(imag(zs),real(zs))))/(ia*
       abs(zs)));
25  pf=cosd(atand(imag(zs),real(zs))-di);
26  printf('Power factor is %f lagging\n',pf);
```

Scilab code Exa 5.4 Determining line current and pf for synchronous motor

```
1  clc;
2  v=3300; // rated voltage of motor
3  zs=0.4+%i*5; // synchronous impedance per phase
4  E=4000; // excitation EMF
5  pi=1000; // input power
6  vp=v/sqrt(3); // per phase rated voltage
7  ep=E/sqrt(3); // per phase excitation EMF
8  al=atand(real(zs),imag(zs));
9  t1=(pi*1000)/3;
10 t2=(vp^2/abs(zs)^2)*real(zs);
11 t3=abs(zs)/(vp*ep); // terms needed to evaluate load
       angle
```

```
12  di=asind((t1-t2)*t3)+al; // load angle
13  ia=(sqrt(vp^2+ep^2-2*ep*vp*cosd(di)))/abs(zs);
14  pf=(pi*1000)/(3*ia*vp);
15  // here ep*cos(di)+ia*ra*pf> vp; hence leading power
        factor
16  printf('Line current is %f A\n',ia);
17  printf('Power factor is %f leading ',pf);
```

Scilab code Exa 5.5 Determining torque developed and new pf

```
1  clc;
2  v=230; // rated voltage of motor
3  f=50; // frequency
4  p=4; // number of poles
5  zs=0.6+3*%i; // synchronous impedance
6  ia1=10; // current drawn by motor at upf
7  ia2=40; // final current after load is inceased to
       certain value
8  vt=v/sqrt(3); // per phase voltage
9  al=atand(real(zs),imag(zs));
10 Ef=sqrt((vt-ia1*real(zs))^2+(ia1*imag(zs))^2); //
       excitation EMF
11 t1=(ia2*abs(zs))^2;
12 t2=Ef^2+vt^2;
13 t3=-2*Ef*vt; // terms needed to evaluate load angle
14 de=acosd((t1-t2)/t3); // load angle
15 pi=(Ef*vt*sind(de-al))/abs(zs)+(vt^2*real(zs))/abs(
       zs)^2; // input power
16 pf=pi/(vt*ia2);
17 printf('Power factor is %f lagging\n',pf);
18 pd=3*(pi-ia2^2*real(zs)); // developed power
19 ns=(120*f)/p; // synchronous speed
20 T=(pd*60)/(2*%pi*ns);
21 printf('Torque developed is %f N-m',T);
```

214

Scilab code Exa 5.6 Determining pf for increased input power

```
1  clc ;
2  v =6600; // rated voltage of motor
3  zs =1.5+12*%i ; // per phase synchronous impedance
4  pi1 =1000; // input power
5  pf =0.8; // power factor
6  pi2 =1500; // power at which power factor is to be
        found out
7  vt=v/sqrt(3) ; // per phase voltage
8  al=atand(real(zs),imag(zs));
9  ia =( pi1 *1000)/( sqrt (3)* v * pf );
10 Ef=sqrt((vt*pf-ia*real(zs))^2+(vt*sqrt(1-pf^2)+ia*
        imag(zs))^2); // excitation EMF
11 t1 =( pi2 *1000)/3;
12 t2 =( vt^2/ abs ( zs )^2)* real ( zs );
13 t3=abs(zs)/(vt*Ef); // terms needed to evaluate load
          angle
14 di=asind((t1-t2)*t3)+al; // load angle
15 ia =( sqrt ( vt^2+ Ef^2-2* Ef * vt * cosd ( di )))/ abs ( zs ); //
        new armature current
16 pfn =(( pi2 - pi1 )*1000)/( ia * vt );
17 // as Ef*cos(di)+ia*ra> vt hence leading power
        factor
18 printf ('New power factor is %f leading ',pfn);
```

Scilab code Exa 5.7 Determining new value of armature current and power factor

```
1  clc ;
2  v =2300; // rated voltage of motor
3  xs =12 ; // per phase synchronous reactance
```

```
4  p =200000;  // VA rating of motor
5  l1 =120000;  // initial load
6  l2 =60000;  // final load
7  vt=v/sqrt (3) ;  // rated per phase voltage
8  ia=l1 /(3* vt);  // minimum armature current
9  ia1 =1.5* ia;  // armature current at reduced load (50%
       increment )
10 pf =1/1.5;  // power factor
11 Ef=sqrt ((vt*pf)^2+(vt*sqrt (1-pf^2)+ia1*xs)^2);  //
       excitation EMF
12 de=asind ((l2*xs)/(3*vt*Ef));  // new load angle
13 ia2 =( sqrt (vt^2+Ef^2 -2*Ef*vt*cosd (de )))/xs;  // new
       armature current
14 printf ('New value of armature current is %f A\n',ia2
       );
15 pfn=l2 /(3*vt*ia2 );
16 printf ('Power factor at new armature current is %f
       leading ',pfn );
```

Scilab code Exa 5.8 Determining power factor at which machine is oper-
ating

```
1  clc ;
2  ef =1.2;  // ratio of excitation voltage to rated per
       phase voltage
3  i =0.7;  // ratio of armature current to rated current
4  r =0.01;  // percentage resistance of motor
5  x =0.5;  //  percentage reactance of motor
6  // as per the expression given in book
7  t1=ef^2 -(r*i)^2 -(x*i)^2 -1;
8  t2=sqrt ((2*i*r)^2+(2*i*x)^2);
9  t3=atand ((2*i*r)/(2*i*x));  // terms needed to find
       out power factor
10 pf=cosd (asind (t1/t2)-t3 );
11 printf ('Power factor is %f lagging ',pf );
```

216

Scilab code Exa 5.9 Calculating 1 pf efficiency excitation emfs input current and 2 load angle and maximum output

```scilab
1  clc;
2  v=400; // rated voltage of motor
3  zs=0.13+%i*1.3 ; // per phase synchronous impedance
4  p=100000; // VA rating of motor
5  l=4000; // stray losses
6  pl=75000; // power delivered to load
7  disp('case a');
8  il=p/(sqrt(3)*v); // line current
9  vt=v/sqrt(3); // per phase rated voltage
10 pd=pl+l ; // power developed
11 poh=3*il^2*real(zs);
12 lt=poh+l; // total losses
13 pi=pl+lt; // input power
14 pf=pi/p; // power factor
15 n=(1-(lt/pi))*100; // efficiency
16 printf('Power factor is %f\n',pf);
17 printf('Efficiency is %f percent\n',n);
18 Ef1=round(sqrt((vt*pf-il*real(zs))^2+(-vt*sqrt(1-pf
      ^2)+il*imag(zs))^2)); // excitation EMF
19 de=atand((-vt*sqrt(1-pf^2)+il*imag(zs))/(vt*pf-il*
      real(zs)))+acosd(pf); // load angle
20 printf('Excitation EMf at under excitation is %f v\n
      ',Ef1);
21 printf('Load angle at under excitation is %f degrees
      \n',de);
22 Ef2=round(sqrt((vt*pf-il*real(zs))^2+(vt*sqrt(1-pf
      ^2)+il*imag(zs))^2)); // excitation EMF
23 de=atand((vt*sqrt(1-pf^2)+il*imag(zs))/(vt*pf-il*
      real(zs)))-acosd(pf); // load angle
24 printf('Excitation EMf at over excitation is %f v\n'
      ,Ef2);
```

```
25  printf('Load angle at over excitation is %f degrees\
        n',de);
26  i=pi/(sqrt(3)*v);
27  printf('Input current is %f A\n',i);
28  disp('caes b');
29  de=acosd(real(zs)/abs(zs)); // load angle
30  pmax=((vt*Ef1)/abs(zs))-((Ef1^2*real(zs))/abs(zs)^2)
        ;
31  pt=pmax*3;
32  printf('Load angle for maximum power output is %f
        degrees\n',de);
33  printf('Maximum output per phase is %f W\n',pmax);
34  printf('Total maximum output is %f W\n',pt);
```

Scilab code Exa 5.10 Determining 1 excitation emfs 2 mechanical power developed and pf 3 minimum excitation voltage

```
1   clc;
2   v=6600; // rated voltage of motor
3   xs=20 ; // per phase synchronous reactance
4   p=500000; // VA rating of motor
5   il=p/(sqrt(3)*v); // rated armature current
6   vt=v/sqrt(3); // per phase rated voltage
7   disp('case a');
8   de=10; // load angle
9   c1=1;
10  c2=-2*vt*cosd(de);
11  c3=vt^2-(il*xs)^2; // coefficients of quadratic
        equation in Ef
12  p= [ c1 c2 c3 ];
13  Ef=roots(p);
14  printf('Per phase excitation EMF at lagging pf is %f
        v\n',Ef(2));
15  printf('Excitation line EMF at lagging pf is %f v\n'
        ,sqrt(3)*Ef(2));
```

```
16  printf ('Per phase excitation EMF at leading pf is %f
         v\n',Ef(1));
17  printf ('Excitation line EMF at leading pf is %f v\n'
         ,sqrt(3)*Ef(1));
18  disp ('case b');
19  disp ('For lagging pf');
20  pd=(3*vt*Ef(2)*sind(de))/xs;
21  pf=pd/(sqrt(3)*v*il);
22  printf ('Mechanical power developed is %f W\n',pd);
23  printf ('Power factor is %f lagging\n',pf);
24  disp ('For leading pf');
25  pd=(3*vt*Ef(1)*sind(de))/xs;
26  pf=pd/(sqrt(3)*v*il);
27  printf ('Mechanical power developed is %f W\n',pd);
28  printf ('Power factor is %f leading\n',pf);
29  disp ('case c');
30  p=200000; // delivered power
31  de=90; // load angle for falling out of step
32  // motor falls out of step at de= 90 degrees
33  Ef=(p*xs)/(3*sind(de)*vt);
34  printf ('Minimum excitation voltage per phase is %f v
         ',Ef);
```

Scilab code Exa 5.11 Calculating current drawn from supply and pf

```
1   clc;
2   v=415; // rated voltage of motor
3   f=50; // frequency of motor
4   ef=520; // line to line excitation emf
5   p=6; // number of poles
6   xs=2; // per phase synchronous reactance
7   t=220; // torque developed
8   vt=v/sqrt(3); // rated per phase voltage
9   eft=ef/sqrt(3); // per phase excitation emf
10  ws=(4*%pi*f)/p; // synchronous speed
```

219

```
11 de=asind((t*ws*xs)/(3*vt*eft)); //  load angle
12 ia=(sqrt(vt^2+eft^2-2*eft*vt*cosd(de)))/xs; //  from
        phasor diagram(fig 5.48),armature current
13 pf=(ef*sind(de))/(xs*ia*sqrt(3));
14 printf('Current drawn from supply is %f A\n',ia);
15 printf('Power factor is %f leading',pf);
```

Scilab code Exa 5.12 Determining synchronous reactance

```
1 clc;
2 v=415; // rated volatge of motor
3 pf=0.9; // leading power factor
4 ps=15000; // shaft power
5 E=400; // excitation emf
6 r=0.5; // per phase resistance
7 vt=v/sqrt(3); // rated per phase voltage
8 e=E/sqrt(3); // per phase excitation emf
9 c1=1.5;
10 c2=-sqrt(3)*v*pf;
11 c3=ps; // coefficients of quadratic equation in
      armature current
12 p= [ c1 c2 c3 ];
13 ia=roots(p);
14 // higher value of armature current is neglected
15 xs=((vt*sqrt(1-pf^2))-(sqrt(e^2-(vt*pf+ia(2)*r)^2)))
      /ia(2);
16 printf('Synchronous reactance is %f ohm',xs);
```

Scilab code Exa 5.13 Finding mechanical power delivered by motor and
input KVA

```
1 clc;
2 e=1.2; // pu excitation emf
```

```
3  xs=0.8; // pu synchronous reactance
4  vt=1; // pu rated voltage
5  ia=1; // pu armature current for KVA=100 %, vt*ia=1
      therefore ia=1;
6  pf=cosd(asind((e^2-xs^2-1)/(-2*xs))); // leading
      power factor
7  pd=vt*ia*pf;
8  printf('Mechanical power developed by motor is %f pu
      \n',pd);
9  e=1; // pu excitation emf reduced to generate 100%
      emf
10 de=asind((pf*xs)/(vt*e)); // load angle
11 ia=(sqrt(vt^2+e^2-2*e*vt*cosd(de)))/xs; // new
      armature current
12 p=(vt*ia)*100;
13 printf('New KVA rating is %f percent ',p);
```

Scilab code Exa 5.14 Determining 1 load angle pu armature current and pf 2 excitation voltage load angle and pf

```
1  clc;
2  xs=0.8; // pu reactance
3  Ef=1.3; // pu excitation EMF
4  p=0.5; // pu output power
5  vt=1; // rated per phase voltage
6  disp('case a');
7  de=asind((p*xs)/(vt*Ef));
8  printf('Load angle is %f degrees\n',de);
9  ia=(sqrt(vt^2+Ef^2-2*vt*Ef*cosd(de)))/xs; // from
      phasor diagram (fig5.49)
10 printf('Armature current is %f p.u.\n',ia);
11 pf=p/(vt*ia);
12 printf('Power factor is %f lagging\n',pf);
13 disp('case b');
14 // Under given condition magnitude of power factor
```

221

```
             remains same but it becomes leading
15  Ef=sqrt((vt*pf)^2+(vt*sqrt(1-pf^2)-ia*xs)^2); //
        excitation EMF
16  printf('Excitation EMF is %f p.u.\n',Ef);
17  de=asind((p*xs)/(vt*Ef));
18  printf('Load angle is %f degrees\n',de);
```

Scilab code Exa 5.15 Determining 1 new value of armature current load angle and pf 2 armature current load angle and power delivered

```
1   clc;
2   v=11000; // rated voltage of motor
3   zs=1+10*%i; // per phase synchronous impedance
4   ia=100; // armature current at unity power factor
5   vt=v/sqrt(3); // per phase voltage
6   Ef=sqrt((vt+ia*real(zs))^2+(ia*imag(zs))^2); //
        excitation EMF from phasor diagram
7   al=atand(real(zs),imag(zs));
8   de=atand((ia*imag(zs))/(vt+ia)); // load angle
9   p=(Ef*vt*sind(de+al)/abs(zs))-((vt^2*real(zs))/abs(
        zs)^2); // per phase power delivered
10  disp('case a');
11  Ef1=1.15*Ef; // Excitation EMF after an increment of
         15%
12  t1=p;
13  t2=(vt^2/abs(zs)^2)*real(zs);
14  t3=abs(zs)/(vt*Ef1); // terms needed to evaluate
        load angle
15  di=asind((t1+t2)*t3)-al; // load angle
16  ia1=(sqrt(vt^2+Ef1^2-2*Ef1*vt*cosd(di)))/abs(zs); //
         armature current
17  pf=p/(vt*ia1);
18  printf('New value of armature current is %f A\n',ia1
        );
19  printf('New value of load angle is %f degrees\n',di)
```

```
      ;
20 printf ('New power factor is %f lagging\n',pf);
21 disp ('case b');
22 // at unity pf
23 pf =1;
24 c1 =1+ imag ( zs )^2;
25 c2 =2* vt ;
26 c3 = vt ^2 - Ef1 ^2; // coefficients of quadratic equation
       in armature current
27 p= [ c1 c2 c3 ];
28 ia = roots (p);
29 printf ('Armature current under given condition is %f
      A\n', ia (2) );
30 P=( vt * ia (2) * pf )/1000;
31 Pt =P *3;
32 printf ('Per phase power delivered is %f KW\n',P);
33 printf ('Net power delivered is %f KW\n',Pt);
```

Scilab code Exa 5.16 Determining operating pf and load angle

```
1 clc ;
2 vt =1; // rated per phase voltage
3 zs =0.02+0.8*% i; // per phase p.u synchronous
     impedance
4 // At the time of synchronization excitation EMF=
     rated per phase voltage and load angle =0
5 ef =1; // pu excitation EMF
6 ia =1; // pu armature current
7 // from phasor diagram 5.51
8 t1 = ef ^2 - real ( zs )^2 - imag ( zs )^2 -1;
9 t2 = sqrt ((2* real ( zs ))^2+(2* imag ( zs ))^2);
10 t3 = atand ( - real ( zs )/ imag ( zs )); // terms needed to
     find out power factor
11 pf = cosd ( asind ( t1 / t2 )+ t3 );
12 printf ('Operating power factor is %f leading\n',pf);
```

223

```
13  al=atand(real(zs),imag(zs));
14  t1=vt*ia*pf;
15  t2=(vt^2/abs(zs)^2)*real(zs);
16  t3=abs(zs)/(vt*ef); // terms needed to evaluate load
        angle
17  de=floor(asind((t1+t2)*t3))-al; // load angle
18  printf('Load angle of generator is %f degrees',de);
```

Scilab code Exa 5.17 Determining armature current power factor and excitation emf

```
1   clc;
2   p=40*10^6; // rated power of turbogenerator
3   v=11000; // rated voltage of generator
4   xs=0.8; // p.u. synchronous reactance
5   ra=0.5; // series reactance of infinite bus
6   vt=v/sqrt(3); // rated per phase voltage
7   disp('case b');
8   ia=p/(v*sqrt(3)); // armature current
9   printf('Armature current is %f A\n',ia);
10  xs=xs*(vt/ia); // xs in ohms
11  vd=ia*ra; // voltage drop in series resistance
12  pf=cosd(asind((vd/2)/vt));
13  printf('Alternate power factor is %f lagging\n',pf);
14  disp('case c');
15  Ef=sqrt((vt*pf)^2+((vd/2)+(ia*xs))^2);
16  printf('Excitation EMF line to neutral is %f V\n',Ef
        );
17  printf('Excitation EMF line to line is %f V\n',sqrt
        (3)*Ef);
```

Scilab code Exa 5.18 Determining load angle power factor and armature current

```
1  clc;
2  v=400; // rated voltage of motor
3  pi=5472; // input power
4  np=3; // number of phases
5  xs=10; // synchronous reactance
6  ef=v; // excitation voltage
7  vt=v/sqrt(3); // rated per phase voltage
8  de=round(asind((pi*xs*np)/(np*v^2)));
9  printf('Load angle is %f degrees\n',de);
10 // from fig. 5.53, vt=ef(excitation voltage per
       phase) armature resistance=0
11 pf=cosd(de/2);
12 printf('Power factor is %f lagging\n',pf);
13 // from fig. 5.53
14 ia=floor((2*vt*sind(de/2))/xs);
15 printf('Armature current is %f A',ia);
```

Scilab code Exa 5.19 Determining 1 excitation emf 2 power transfer and 3 maximum power transfer armature current terminal voltage and power factor

```
1  clc;
2  v=2000; // rated voltage of motor
3  xsm=2; // synchronous reactance of motor
4  xsg=3; // synchronous reactance of generator
5  xt=1.5; // transmission line reactance
6  ia=100; // current drawn by motor
7  pf=1; // power factor
8  disp('case a');
9  vt=v/sqrt(3); // rated per phase voltage
10 Efm=floor(sqrt(vt^2+(ia*xsm)^2)); // excitation EMF
11 printf('Excitation EMF for motor is %f V\n',Efm);
12 Efg=sqrt(vt^2+(ia*(xsg+xt))^2); // excitation EMF
13 printf('Excitation EMF for alternator is %f V\n',Efg
       );
```

225

```
14  disp('case b');
15  de1=acosd(vt/Efm); // load angle for motor
16  de2=acosd(vt/Efg); // load angle for alternator
17  de=de1+de2; // power angle between Efm and Efg
18  pt=(Efg*Efm*sind(de))/(xsm+xsg+xt);
19  P=pt*3;
20  printf('Per phase power transfer between alternator
        and motor is %f KW\n',pt/1000);
21  printf('Net power transfer between alternator and
        motor is %f KW\n',P/1000);
22  disp('case c');
23  // from phasor diagram fig 5.54
24  ia=sqrt(Efm^2+Efg^2)/(xsm+xsg+xt);
25  // for maximum transfer of power , power angle=90
        degrees
26  de=90
27  pmax=(Efg*Efm*sind(de))/(xsm+xsg+xt);
28  P=pmax*3;
29  printf('Per phase maximum power transfer between
        alternator and motor is %f KW\n',pmax/1000);
30  printf('Net maximum power transfer between
        alternator and motor is %f KW\n',P/1000);
31  // from phasor diagrams determining various
        parameters needed to find power factor
32  be=acosd(Efm/(ia*(xsm+xsg+xt)));
33  Vp=sqrt((Efm-ia*xsm*cosd(be))^2+(ia*xsm*sind(be))^2)
        ; // phase voltage
34  Vl=sqrt(3)*Vp; // line voltage
35  printf('Armature current for given condition is %f A
        \n',ia);
36  printf('Terminal voltage of synchronous motor is %f
        V\n',Vp);
37  // from phasor diagram
38  aoc=asind((ia*xsm*sind(be))/Vp);
39  pf=cosd(90-be-aoc);
40  printf('Power factor angle of motor is %f leading',
        pf);
```

Scilab code Exa 5.20 Determining 1 maximum power output 2 armature current and power factor

```
1  clc;
2  p=20*10^6; // VA rating of alternator
3  z=5; // impedance of alternator
4  r=0.5; // resistance of alternator
5  v=11000; // voltage rating of bus bars
6  e=12000; // excitation voltage
7  vt=v/sqrt(3); // alternator per phase voltage
8  Ef=e/sqrt(3); // alternator per phase excitation
       voltage
9  pmax=round(((((Ef*vt)/z)-((vt^2*r)/z^2))/10^6);
10 P=round(pmax*3);
11 printf('Per phase maximum output power from
       alternator is %f MW\n',pmax);
12 printf('Total maximum output power from alternator
       is %f MW\n',P);
13 disp('case b');
14 pf=r/z; // power factor
15 ia=round((sqrt(vt^2+Ef^2-2*Ef*vt*pf))/z); //
       armature current
16 printf('Armature current is %f A\n',ia);
17 pf=(Ef*z-vt*r)/(ia*z^2);
18 printf('Power factor under maximum power condition
       is %f leading ',pf);
```

Scilab code Exa 5.21 Calculating data needed to plot v curves and pf variation with field current

```
1  clc;
2  v=2200; // rated voltage of motor
```

```
3  r=0.32; // per phase armature resistance
4  p=1500; // KW rating of motor
5  ie=15; // exciting current
6  is=750; // short circuit current
7  cl=60; // core loss in KW
8  fl=40; // frictional and windage loss in KW
9  IF=[5 10 15 20 25 30];
10 EFO=[ 760 1500 2140 2650 3040 3340]; // excitation
       EMF per phase
11 EFP=EFO/sqrt(3);
12 disp('Excitation EMF per phase(V) is ');
13 disp(EFP);
14 // from table given for ie=15,
15 Ef=2140; // Excitation EMF
16 np=3; // number of phases
17 ef=Ef/sqrt(3); // per phase open circuit voltage
18 zs=ef/is; // synchronous impedance
19 vt=floor(v/sqrt(3)); // per phase terminal voltage
20 i=floor(vt/zs); // current phasor lagging vt
21 ia=vt/(2*r); // armature current
22 pd=((p/2)+cl+fl)/np; // mechanical power developed
       per phase at half-full load output
23 R=ceil(sqrt((ia^2-((pd*1000)/r))));
24 printf('Radius of power circle is %f A\n',R);
25 printf('Current phasor is %f A\n',i);
26 printf('Synchronous impedance is %f ohm\n',zs);
27 disp('using above data and table given in solution,
       V-curves and variation of p.f. with field
       currents can be plotted');
```

Scilab code Exa 5.22 Determining data to plot v curves and pf variation
with field current neglecting armature resistance

```
1  clc;
2  v=2200; // rated voltage of motor
```

228

```
3  p=1500;  // KW rating of motor
4  ie=15;  // exciting current
5  is=750;  // short circuit current
6  cl=60;  // core loss in KW
7  fl=40;  // frictional and windage loss in KW
8  // from table given in question for ie=15,
9  Ef=2140;  // Excitation EMF
10 np=3;  // number of phases
11 ef=Ef/sqrt(3);  // per phase open circuit voltage
12 xs=ef/is;  // synchronous reactance
13 vt=floor(v/sqrt(3));  // per phase terminal voltage
14 i=floor(vt/xs);  // current phasor lagging vt
15 pd=((p/2)+cl+fl)/np;  // mechanical power developed
      per phase at half−full load output
16 ia=pd*1000/vt;  // working component of armature
      current
17 // as resistance =0 , armature current=0 therefore
      radius of power circle=0 that is it becomes line
      with centre at zero
18 printf('Working component of armature current is %f
      A\n',ia);
19 printf('Terminal voltage is %f A\n',vt);
20 printf('Synchronous reactance is %f ohm\n',xs);
21 disp('using above data and table given in solution,
      V−curves and variation of p.f. with field
      currents can be plotted');
```

Scilab code Exa 5.23 Determining armature current power factor and load angle

```
1  clc;
2  v=1100;  // rated voltage of motor
3  ef=1650;  // emf
4  p=350;  // input power in KW
5  zs=0.7+3.2*%i;  // synchronous impedance per phase
```

```
6  vt=v/sqrt(3); // rated per phase voltage
7  eft=ef/sqrt(3); // per phase emf
8  i1=vt/abs(zs);
9  printf('Current phasor lagging terminal voltage is
      %f A\n', i1);
10 i2=eft/abs(zs);
11 printf('Current phasor lagging excitation EMF is %f
      A\n', i2);
12 ia=(p*1000)/(3*vt);
13 printf('Working component of armature current is %f
      A',ia);
14 disp('using this data vector diagram is drawn and
      value of ia and power factor is obtained');
15 ia=194.5;
16 pf=19.5;
17 printf('Power factor is %f leading\n',pf);
18 printf('Armature current is %f A\n',ia);
19 de=acosd((ia^2-i1^2-i2^2)/(-2*i1*i2));
20 printf('Load angle is %f degrees\n',de);
```

Scilab code Exa 5.24 Determining pu excitation voltage

```
1  clc;
2  xd=1.2; // d axis synchronous reactance
3  xq=0.8; // q axis synchronous reactance
4  ra=0.025; // armature resistance
5  vt=1; // pu rated per phase voltage
6  disp('case a');
7  disp('For lagging power factor')
8  pf=0.8; // power factor
9  ia=1*(pf-sqrt(1-pf^2)*%i); // armature current
10 Ef1=vt+%i*(ia*xq)+ia*ra; // excitation EMF
11 id=1*sind(atand(imag(Ef1),real(Ef1))+acosd(pf)); //
      d component of armature current
12 iq=1*cosd(atand(imag(Ef1),real(Ef1))+acosd(pf)); //
```

```
        q  component  of  armature  current
13  Ef=abs(Ef1)+id*(xd-xq); // excitation EMF
14  printf('Excitation EMF is %f p.u. at a load angle of
        %f degrees\n',abs(Ef),atand(imag(Ef1),real(Ef1))
        );
15  disp('case b');
16  disp('For leading power factor')
17  pf=0.8; // power factor
18  ia=1*(pf+sqrt(1-pf^2)*%i); // armature current
19  Ef1=vt+%i*(ia*xq)+ia*ra; // excitation EMF
20  id=1*sind(atand(imag(Ef1),real(Ef1))-acosd(pf)); //
        d  component  of  armature  current
21  iq=1*cosd(atand(imag(Ef1),real(Ef1))-acosd(pf)); //
        q  component  of  armature  current
22  Ef=abs(Ef1)+id*(xd-xq); // excitation EMF
23  printf('Excitation EMF is %f p.u. at a load angle of
        %f degrees\n',abs(Ef),atand(imag(Ef1),real(Ef1)))
        ;
```

Scilab code Exa 5.27 Calculating 1 power angle armature current and pf 2 maximum power output and power angle

```
1  clc;
2  v=400; // rated voltage of synchronous machine
3  vt=v/sqrt(3); // per phase rated voltage
4  ps=9500; // shaft load
5  xd=5; // per phase d-axis synchronous reactance
6  xq=3.2; // per phase q-axis synchronous reactance
7  l=500; // friction windage and core losses
8  np=3; // number of phases
9  // at rated voltage excitation EMF
10 Ef=v/sqrt(3); // excitation EMF
11 disp('case a');
12 pt=ps+l; // power developed
13 // by using formula pd=np*(((Ef*Ef*sind(de))/xd)+((
```

231

www.mepcafe.com

```
      Ef^2*sind(2*de)*0.5*(xd-xq))/(xd*xq))), and hit
      and trial method we obtain value of load angle
14 de=11.623; // load angle
15 id=(Ef-Ef*cosd(de))/xd; // d-axis component of
      armature current
16 iq=(Ef*sind(de))/xq; // q-axis component of armature
         current
17 ia=sqrt(id^2+iq^2);
18 printf('Armature current is %f A\n',ia);
19 pf=cosd(acosd(iq/ia)-de);
20 printf('Power factor is %f lagging\n',pf);
21 disp('case b');
22 de=acosd((-Ef*xq)/(4*Ef*(xd-xq))+(sqrt(0.5+((Ef*xq)
      /(4*vt*(xd-xq)))^2)));
23 pd=np*(((Ef*Ef*sind(de))/xd)+((Ef^2*sind(2*de)*0.5*(
      xd-xq))/(xd*xq))); // maximum power developed
24 po=pd-l;
25 printf('Maximum power output is %f W',po);
```

Scilab code Exa 5.29 Determining change in synchronous power and reactive power

```
 1 clc;
 2 Ef=1.4; // p.u excitation EMF
 3 xs=1.2; // p.u synchronous reactance
 4 p=0.5; // p.u synchronous power being delivered
 5 i=1; // percentage increment in prime mover torque
 6 vt=1; // rated per phase voltage
 7 de=asind((p*xs)/(Ef*vt)); // load angle
 8 dp=(i*p)/100; // increase in p.u real power
 9 ip=(dp/p)*100;
10 printf('Percentage increase in real power is %d
      percent of its previous value\n',ip);
11 iq=-tand(de)*ip;
12 printf('Percentage decrease in reactive power is %f
```

232

```
        percent of its previous value\n',-iq);
```

Scilab code Exa 5.31 Determining maximum load armature current and pf

```
1  clc;
2  v=400; // rated voltage of motor
3  xd=6; // d−axis synchronous reactance
4  xq=4; // q−axis synchronous reactance
5  vt=400/sqrt(3); // rated per phase voltage
6  // As per the theory given in question, phasor
       diagram is drawn and formula is derived
7  // After the derived expression, for maximum power
       load angle=45
8  de=45;
9  P=(vt^2/2)*((1/xq)-(1/xd))*sind(2*de);
10 printf('Maximum load that can be put on synchronous
        motor is %f W per phase\n',P);
11 printf('Maximum load that can be put on synchronous
        motor is %f W for 3−phase\n',3*P);
12 iq=(vt*sind(de))/xq; // q−axis component of armature
         current
13 id=(vt*sind(de))/xd; // d−axis component of armature
         current
14 ia=sqrt(iq^2+id^2);
15 printf('Armature current is %f A\n',ia);
16 pf=(3*P)/(sqrt(3)*v*ia);
17 printf('Power factor at maximum power is %f lagging '
       ,pf);
```

Scilab code Exa 5.32 Determining minimum excitation voltage and maximum stable load angle

233

```
1  clc ;
2  // Answer for minimum excitation voltage is given
       wrong in book
3  v=400; // rated voltage of motor
4  xd=6; // d-axis synchronous reactance
5  xq=4; // q-axis synchronous reactance
6  vt=400/sqrt(3); // rated per phase voltage
7  p=21; // load carried by motor
8  pph=(p/3)*1000; // per phase load carried by motor
9  // As per the theory given in question , expression
       is derived
10 // After the derived expression ,
11 // cos(de)=(vt^2*(xd-xq)*sin(de)^3)/(pph*xd*xq) ,
       value of de ( load angle is obtained by trial and
       error method  and value of load angle is )
12 de=63.2;
13 Ef=(pph-(vt^2/2)*((xd-xq)/(xd*xq))*sind(2*de))/((vt/
       xd)*sind(de));
14 printf('Maximum stable load angle is %f degrees\n',
       de);
15 printf('Minimum excitation voltage is %f v\n',Ef);
```

Scilab code Exa 5.34 Determining load angle and excitation voltage

```
1  clc ;
2  vt=1; // pu rated voltage
3  xd=1; // pu d-axis synchronous reactance
4  xq=0.6; // pu q-axis synchronous reactance
5  p=0.9; // pu power being delivered
6  pf=0.8; // power factor
7  ia=p/(vt*pf); // pu armature current
8  de=atand((ia*xq+vt*sind(acosd(pf)))/(vt*pf))-acosd(
       pf);
9  printf('Load angle is %f degrees\n',de);
10 Ef=vt*cosd(de)+ia*sind(de+acosd(pf))*xd;
```

```
11  printf('Excitation  voltage  is  %f  p.u.\n',Ef);
12  // when  loss  of  excitation  takes  place  Ef=0,  for
        maximum  power  load  angle=45
13  de=45; // load angle
14  pmax=(vt^2*(xd-xq)*sind(2*de))/(2*xd*xq);
15  printf('Maximum  power  is  %f  p.u.\n',pmax);
16  disp('As  maximum  power  is  less  than  the  power  being
        delivered  generator  will  lose  synchronism');
```

Scilab code Exa 5.35 Determining excitation voltage power synchronizing power per electrical and mechanical degrees and corresponding torque maximum value of load angle and corresponding power

```
1  clc;
2  v=11000; // rated voltage of motor
3  P=20*10^6; // rated power of motor
4  p=12; // number of poles
5  f=50; // frequency
6  xd=5; // d-axis synchronous reactance
7  xq=3; // q-axis synchronous reactance
8  vt=v/sqrt(3); // per phase rated voltage
9  ia=P/(sqrt(3)*v); // per phase armature current
10  disp('case a');
11  // from phasor diagram
12  Ef=vt-%i*(ia*xq);
13  de=atand((ia*xq),(vt)); // load angle
14  id=ia*sind(de); // d-axis component of armature
        current
15  Ef=ceil(abs(Ef)+id*(xd-xq));
16  printf('Excitation  voltage  is  %f  V  to  neutral\n',Ef)
        ;
17  disp('case b');
18  po=(((Ef*vt*sind(de))/xd)+((vt^2*(xd-xq)*sind(2*de))
        /(2*xd*xq)))/1000;
19  printf('Power  is  %f  KW  per  phase\n',po);
```

235

```
20  po=3*po*1000;
21  printf('Calculated power %f W is almost equal to
       given rated power i.e %f W\n',po,P);
22  disp('case c');
23  ps=((Ef*vt*cosd(de))/xd)+((vt^2*(xd-xq)*cosd(2*de))
       /(xd*xq));
24  printf('Synchronozing power per electrical degree is
        %f KW\n',(3*ps*%pi)/(180*1000));
25  ws=(4*%pi*f)/p; // synchronous speed
26  t=(3*ps*%pi)/(180*ws);
27  printf('Torque corresponding to synchronous power is
        %f N-m\n',t);
28  disp('case d');
29  printf('Synchronozing power per mechanical degree is
        %f KW\n',(3*ps*%pi*p)/(2*180*1000));
30  t=(3*ps*%pi*p)/(2*180);
31  printf('Torque corresponding to synchronous power is
        %f N-m\n',t);
32  disp('case e');
33  de=acosd((-Ef*xq)/(4*Ef*(xd-xq))+(sqrt(0.5+((Ef*xq)
       /(4*vt*(xd-xq)))^2)));
34  printf('Maximum value of power angle is %f degrees\n
       ',de);
35  pmax=(((Ef*vt*sind(de))/xd)+((vt^2*(xd-xq)*sind(2*de
       ))/(2*xd*xq)))/1000;
36  printf('Maximum power is %f KW per phase\n',pmax);
37  printf('Maximum power is %f KW for 3 phase\n',pmax
       *3);
```

Scilab code Exa 5.36 Calculate synchronizing power and torque per mechanical degrees 1 at no load and 2 at full load

```
1  clc;
2  v=6600; // rated voltage of motor
3  p=8; // number of poles
```

```
4  f=50; // frequency of motor
5  xs=20; // percentage synchronous reactance
6  P=3000; // rated power of motor
7  m=3; // number of phases
8  vt=v/sqrt(3); // per phase rated voltage
9  ia=(P*1000)/(sqrt(3)*v); // per phase armature
      current
10 xs=(xs*vt)/(100*ia); // synchronous reactance in ohm
11 disp('case a');
12 de=0; // at no load load angle=0 and excitation
      voltage=per phase rated voltage
13 ps=ceil(((m*vt^2*cosd(de)*%pi*p)/(xs*360))/1000);
14 printf('Synchronozing power per mechanical degree is
       %f KW\n',ps);
15 T=((ps*1000*p*60)/(2*%pi*120*f))/1000;
16 printf('Corresponding synchronizing torque is %f KN–
      m\n',T);
17 disp('case b');
18 pf=0.8; //  lagging power factor
19 Ef=vt+%i*ia*(pf-sqrt(1-pf^2)*%i)*xs; // Excitation
      EMF
20 de=atand(imag(Ef),real(Ef));
21 ps=((m*vt*abs(Ef)*cosd(de)*%pi*p)/(xs*360))/1000;
22 printf('Synchronozing power per mechanical degree is
       %f KW\n',ceil(ps));
23 T=((ps*1000*p*60)/(2*%pi*120*f))/1000;
24 printf('Corresponding synchronizing torque is %f KN–
      m\n',T);
```

Scilab code Exa 5.37 Determining 1 synchronizing current power and torque 2 armature current and angle by which rotor slips back

```
1  clc;
2  vt=3300; // terminal voltage
3  xs=11; // synchronous reactance per phase
```

237

```
4  p=8; // number of poles
5  f=50; // frequency of motor
6  m=3; // number of phases
7  // from fig 5.82
8  // at no load load angle=0 and excitation voltage=
       terminal voltage
9  de=0;
10 s=p/2; // electrical degree equivalent of one
       mechanical degree in space
11 es=2*vt*sind(s/2); // synchronizing voltage
12 is=es/xs;
13 printf('Synchronizing current in the armature is %f
       A\n',is);
14 ps=m*vt*is*cosd(de+s/2);
15 printf('synchonizing power is %f KW\n',floor(ps
       /1000));
16 ws=(2*%pi*120*f)/(60*p);
17 T=ps/ws;
18 printf('Synchronizing torque tending to restore
       rotor to its previous position is %f Nm\n',T);
19 disp('case b');
20 ia=30; // armature current
21 dde=2*(asind((ia*xs)/(2*vt))); // change in load
       angle in electrical degrees
22 s=dde*(2/p);
23 printf('Rotor slips back by %f mechanical degrees',s
       );
```

Scilab code Exa 5.38 Determining 1 load angle armature current and pf
2 maximum load load angle and armature current 3 minimum excitation
voltage

```
1  clc;
2  v=400; // rated voltage of motor
3  f=50; // frequency of motor
```

```
4  r=1; // per phase resistance
5  x=5; // per phase reactance
6  m=3; // number of phases
7  p=15000; // rated power of motor
8  disp('case a');
9  EF=480; // Excitation voltage
10 ph=p/m; // per phase power
11 vt=v/sqrt(3); // terminal voltage
12 R0=vt/(2*r);
13 printf('Radius of zero power circle is %f A\n',R0);
14 R1=sqrt(R0^2-(ph/r));
15 printf('Radius of per phase power circle is %f A\n',
       R1);
16 Ef=EF/sqrt(3); // per phase excitation voltage
17 i1=vt/sqrt(r^2+x^2);
18 i2=Ef/sqrt(r^2+x^2); // current phasors lagging
       terminal and excitation voltage
19 printf('Current phasors lagging terminal voltage is
        %f A\n',i1);
20 printf('Current phasors lagging exciation voltage is
         %f A\n',i2);
21 disp('using the above data power circle diagram is
       drawn and value of armature current and power
       factor is obtained');
22 ia=26;
23 pf=0.955;
24 printf('Armature current is %f A\n',ia);
25 printf('Power factor is %f leading\n',pf);
26 disp('case b');
27 // from power circle diagram, radius for maximum
       power is 61 A
28 R2=61; //  radius for maximum power
29 pmax=(R0^2-R2^2)*r;
30 printf('Maximum power per phase is %f KW\n',pmax
       /1000);
31 printf('Maximum power for 3-phase is %f KW\n',(3*
       pmax)/1000);
32 disp('case c');
```

239

```
33  l=12; // load on motor in KW
34  lp=(l/3)*1000; // per phase load
35  ef=(sqrt(r^2+x^2))*(R0-sqrt(R0^2-lp/r));
36  printf('Minimum excitation voltage is %f V',ef);
```

Scilab code Exa 5.41 Determining 1 excitation voltage 2 reluctance power developed

```
1  clc;
2  n=1490; // speed of machine in rpm
3  p=4; // number of poles
4  f=50; // frequency
5  v=11000; // rated voltage of machine
6  p=20*10^6; // rated power of machine
7  v1=30;
8  v2=25; // per phase voltage for phase A of machine
9  i1=10;
10 i2=6.5; // per phase current for phase A of machine
11 ns=(120*f)/p; // synchronous speed of machine
12 xd=v1/i2; // d-axis synchronous reactance
13 xq=v2/i1; // q-axis synchronous reactance
14 disp('case a');
15 ia=p/(sqrt(3)*v); // armature current
16 vt=v/sqrt(3); // per phase armature voltage
17 Ef=vt+ia*xq*%i;
18 de=atand(imag(Ef),real(Ef)); // load angle
19 id=ia*sind(de); // d-axis current
20 Ef1=abs(Ef)+id*(xd-xq);
21 printf('Per phase excitation value is %f V\n',ceil(
       Ef1));
22 printf('Line value of excitation EMf is %f V\n ',
       ceil(sqrt(3)*Ef1));
23 disp('case 2');
24 pr=(vt^2*(xd-xq)*sind(2*de))/(2*xd*xq);
25 printf('Reluctance power developed by machine is %f
```

240

```
    KW per phase\n',pr/1000);
26 printf('Total reluctance power developed by machine
     is %f KW',(3*pr)/1000);
```

Scilab code Exa 5.43 Determining efficiency at 1 half load 2 full load

```
 1 clc;
 2 p=100*10^3; // rated power of alternator
 3 v=440; // rated voltage of alternator
 4 m=3; // number of phases
 5 l1=340; // friction and windage losses
 6 l2=480; // open circuit core losses
 7 rf=180; // field winding resistance at 75 degree cel
      .
 8 ra=0.02; // armature resistance per phase
 9 vf=220; // voltage applied to field winding
10 pf=0.8; // power factor
11 disp('At half load');
12 ia=p/(sqrt(3)*v); // full load armature current
13 l3=(m*ia^2*ra)/4; // short circuit load loss at half
       load
14 l4=vf^2/rf; // field circuit loss
15 lt=l1+l2+l3+l4; // net loss
16 n=(1-(lt/((p/2)*pf+lt)))*100;
17 printf('Efficiency is %f percent\n',n);
18 disp('At full load');
19 l3=m*ia^2*ra; // short circuit load loss at full
     load
20 lt=l1+l2+l3+l4; // net loss
21 n=(1-(lt/(p*pf+lt)))*100;
22 printf('Efficiency is %f percent\n',n);
```

Scilab code Exa 5.44 Determining effective resistance in pu and ohms and ratio of ac to dc resistance

```
1  clc;
2  p=40000; // rated power of machine
3  v=400; // rated voltage of machine
4  l=1500; // short circuit load loss
5  m=3; // number of phases
6  ia1=1; // armature current in p.u.
7  ra=0.118; // dc armature resistance at 30 degree cel
      .
8  ia2=p/(sqrt(3)*v); // rated armature current
9  l1=l/p; // short circuit loss in p.u.
10 ra1=l1/ia1^2;
11 printf('Effective armature resistance is %f p.u.\n',
      ra1);
12 l2=l/m; // short circuit load loss per phase
13 ra2=l2/ia2^2;
14 printf('Effective ac armature resistance is %f ohm
      per phase\n',ra2);
15 r=ra2/ra;
16 printf('Ratio of ac to dc armature resistance is
      given as %f ',r);
```

Scilab code Exa 5.45 Determining 1 efficiency and 2 maximum efficiency

```
1  clc;
2  p=500*10^3; // rated power of alternator
3  v=11000; // rated voltage of alternator
4  m=3; // number of phases
5  l1=1500; // friction and windage losses
6  l2=2500; // open circuit core losses
7  ra=4; // armature resistance per phase
8  l3=1000; // field copper loss
9  pf=0.8; // power factor
```

242

```
10  disp('case a: Half load');
11  ia=p/(sqrt(3)*v); // armature current
12  l4=(m*ia^2*ra)/4; // short circuit load loss at half
        load
13  lt=l1+l2+l3+l4; // net loss
14  n=(1-(lt/((p/2)*pf+lt)))*100;
15  printf('Efficiency is %f percent\n',n);
16  disp('case b');
17  // for maximum efficiency variable losses=constant
        losses
18  iam=sqrt((l1+l2+l3)/(m*ra)); // armature current at
        maximum efficiency
19  pout=m*(v/sqrt(3))*iam*pf; // output power ta
        maximum efficiency
20  lt=2*(l1+l2+l3); // net losses
21  nm=(1-(lt/(pout+lt)))*100;
22  printf('Maximum efficiency is %f percent\n',nm);
```

Scilab code Exa 5.46 Determining KVA rating of synchronous condenser and KVA of factory

```
1  clc;
2  l=1800; // total load on factory
3  pf=0.6; // power factor
4  pfn=0.95; // desired power factor
5  // from phasor diagram 5.107
6  l1=l/pf; // load in VA
7  a1=acosd(pf); // phase angle between terminal
        voltage and armature current
8  a2=acosd(pfn); // phase angle corresponding to
        desired power factor
9  k1=l1*sind(a1); // KVAr of load
10  k2=l*tand(a2); // combined KVAr
11  disp('case a');
12  s=k1-k2;
```

243

```
13 printf('Synchronous condenser rating is %f KVA\n',
      floor(s));
14 disp('case b');
15 r=sqrt(l^2+k2^2);
16 printf('Total KVA of factory is %f KVA',r);
```

Scilab code Exa 5.47 Determining 1 total load KVA 2 KVA capacity of motor 3 power factor of motor

```
1 clc;
2 l0=300; // total load on factory
3 pf=0.6; // lagging power factor
4 n=88; // percentage efficiency of motor
5 pfn=0.9; // desired power factor
6 l1=60; // mechanical load to be supplied
7 // from phasor diagram 5.108
8 pi=l1/(n/100); // synchronous motor input
9 lt=pi+l0; // combined load
10 disp('case a');
11 k=lt/pfn;
12 printf('Total load is %f KVA\n',k);
13 disp('case b');
14 a1=acosd(pf); // phase angle between terminal
      voltage and armature current
15 a2=acosd(pfn); // phase angle corresponding to
      desired power factor
16 k1=l0*tand(a1); // KVAr of load
17 k2=lt*tand(a2); // combined KVAr
18 s=k1-k2; // leading KVAr supplied by synchronous
      motor
19 r=sqrt(s^2+pi^2);
20 printf('Capacity of synchronous motor is %f KVA\n',r
      );
21 disp('case c');
22 pfs=pi/r;
```

```
23  printf('Synchronous motor operating power factor is
        %f leading ',pfs);
```

Scilab code Exa 5.48 Calculating permissible additional load and rating of synchronous condenser

```
1  clc;
2  p0=1000; // full load power rating of substation
3  pf=0.71; // lagging power factor
4  pfn=0.87; // desired power factor
5  // from phasor dagram 5.109
6  p1=p0*pf; // load KW
7  p2=sqrt(p0^2-p1^2); // load KVAr
8  pn=p0*pfn; // new power delivered to load
9  p0n=pn/pf; // new load KVA
10  pl=p0n-p0;
11  printf('Permissible additional load at %f lagging
        power factor is %f KVA\n',pf,pl);
12  p2n=sqrt(p0n^2-pn^2); // new load KVAr
13  p2ns=sqrt(p0^2-pn^2); // new load KVAr with the use
        of synchronous condensor
14  R=p2n-p2ns;
15  printf('Rating of synchronous condensor is %f KVA',R
        );
```

Scilab code Exa 5.49 Determining new pf and percentage reduction in line current

```
1  clc;
2  p=4000; // load taken by industrial plant in KW
3  pf=0.8; // lagging power factor
4  l=400; // rating of induction motor to be replaced
        by synchronous motor
```

```
 5  n=0.9; // efficiency of induction motor and
        synchronous motor
 6  pf1=0.9; // lagging power factor at which induction
        motor operates
 7  pf2=0.8; // leading power factor at which
        synchronous motor operates
 8  A=p-%i*p*tand(acosd(pf)); // complex power of plant
 9  pi=l/pf1; // power input to induction motor
10  B=pi-%i*pi*tand(acosd(pf1)); // complex power
        requirement of induction motor
11  C=pi+%i*pi*tand(acosd(pf2)); // complex power
        requirement of synchronous motor
12  pfn=cosd(atand(imag(A-B+C),real(A-B+C)));
13  printf('New power factor of the plant is %f lagging\
        n',pfn);
14  r=(abs(A-B+C)/sqrt(3))/(p/(sqrt(3)*pf)); // ratio of
         new line current to original line current
15  pr=(1-r)*100;
16  printf('Percentage reduction in line current is %f
        percent',pr);
```

Scilab code Exa 5.53 Determining voltage regulation at full load

```
 1  clc;
 2  m=3; // number of phases
 3  p=2; // number of poles
 4  P=4*10^6; // rated power of generator
 5  v=11000; // rated voltage of generator
 6  as=72; // armature slots
 7  cs=4; // conductor per armature slot
 8  rs=24; // rotor slots
 9  rp=10; // rotor slot angular pitch
10  cr=20; // conductors per rotor slot
11  z=0.1+2*%i; // armature leakage impedance per phase
12  pf=0.8; // lagging power factor
```

```scilab
13  vt=v/sqrt(3); // terminal voltage
14  ia=P/(sqrt(3)*v); // full load armature current
15  // Open circuit characteristics have been plotted
        using table given in question.Per phase value of
         excitation voltage is obtained fron table
16  IF=[ 40 80 120 160 200 240 280 320 360];
17  EA=[ 2490 4980 7470 9390 10620 11460 12030 12450
        12660 ];
18  plot(IF,EA/sqrt(3));
19  xlabel('Field current');
20  ylabel('open circuit voltage');
21  title('open circuit characteristics');
22  Er=vt+ia*(pf-%i*sqrt(1-pf^2))*z; // air gap voltage
23  printf('Air gap voltage is %f V\n',abs(Er));
24  disp('Corresponding to magnitude of air gap voltage
        value of field MMF in terms of field current is
        obtained from O.C.C (for textbook refer fig.
        5.114)');
25  Fr=242; // field MMF in terms of field current
26  q=rs/p; // rotor slots per pole
27  kd=sind(q*rp/2)/(q*sind(rp/2)); // distribution
        factor
28  kp=1 ; // coil span factor for full pitch field coil
29  kw=kd*kp; // winding factor
30  Nf=(rs*cr)/p; // total field turns
31  F1f=(4*kw*Nf)/(%pi*p); // ratio of fundamental field
        mmf per pole to field current
32  Nph=(as*cs)/(m*p); // series turn per phase
33  q1=as/(m*p); // armature slots per pole per phase
34  v1=(p*180)/as; // armature slot angular pitch
35  kd=(sind(q1*v1/2))/(q1*sind(v1/2)); // distribution
        factor
36  kw=kd*kp; // winding factor
37  Fa=(0.9*m*Nph*ia*kw)/(p*F1f); // armature mmf in
        terms of field current
38  B=acosd(pf)+atand(imag(Er),real(Er)); // power
        factor angle + angle by which air gap voltage
        leads terminal voltage
```

247

```
39  Ff=sqrt(Fr^2+Fa^2-2*Fr*Fa*cosd(90+B)); // equivalent
        field current
40  printf('Equivalent field current is %f A\n',Ff);
41  // corresponding to equivalent field current,
        excitation voltage is obtained from O.C.C
42  Ef=7168; // excitation EMF per phase
43  vr=((Ef-vt)/vt)*100;
44  printf('Voltage regulation at full load %f lagging
        power factor is %f percent',pf,vr);
```

Scilab code Exa 5.54 Determining terminal voltage and current

```
1  clc;
2  p=2*10^6; // rated power of alternator
3  v=11000; // rated voltage of alternator
4  zs=0.3+5*%i; // synchronous impedance per phase
5  pf=0.8; // lagging power factor
6  vt=v/sqrt(3); // terminal voltage
7  ia=p/(sqrt(3)*v); // full load armature current
8  // with vt as reference phasor
9  Ef=vt+ia*(pf-sqrt(1-pf^2)*%i)*zs;
10 // now excitation level is same but load power
        fcator is leading
11 printf('Load current is %f A\n',ia);
12 de=cosd(atand(imag(Ef),real(Ef))); // angle between
        excitation and terminal voltage
13 vt=abs(Ef)*(de+sqrt(1-de^2)*%i)-ia*(pf+sqrt(1-pf^2)*
        %i)*zs;
14 printf('Terminal voltage at %f leading power factor
        is %f V per phase\n',pf,abs(vt));
15 printf('Line terminal voltage is %f KV',(sqrt(3)*abs
        (vt))/1000);
```

248

Scilab code Exa 5.55 Determining load angle and power factor

```
1  clc;
2  p=30*10^6; // rated power of alternator
3  v=11000; // rated voltage of alternator
4  zs=0.005+0.70*%i; // p.u synchronous reactance
5  Ef=1.5; // p.u. excitation EMF
6  ia=1; // p.u. armature current
7  vt=1; // p.u. terminal voltage
8  t1=Ef^2-(real(zs)*ia)^2-(imag(zs)*ia)^2-1;
9  t2=sqrt((2*ia*real(zs))^2+(2*ia*imag(zs))^2);
10 t3=atand((2*ia*real(zs))/(2*ia*imag(zs))); // terms
       needed to find out power factor
11 pf=cosd(asind(t1/t2)-t3);
12 printf('Power factor is %f lagging\n',pf);
13 de=acosd((ia*abs(zs)^2-Ef^2-vt^2)/(-2*Ef*vt));
14 printf('Load angle is %f degrees',de);
```

Scilab code Exa 5.56 Determining 1 terminal voltage and armature current 2 load angle and excitation voltage

```
1  clc;
2  xd=1.2; // pu d-axis synchronous reactance
3  xq=0.8; // pu q-axis synchronous reactance
4  xe=0.1; // pu external reactance
5  vb=1; // voltage of infinite bus
6  pf=0.9; // lagging power factor
7  disp('case a');
8  // vb=vt-j*ia*xe -(1)where ia is armature current
9  // ia=ia*(pf-%i*sqrt(1-pf^2)); // complex form of
       armature current
10 // vt*ia=1 therefore ia=1/vt solving equation 1 we
       get a quadratic equation in vt whose terms are
11 t1=1;
12 t3=-2*xe*sqrt(1-pf^2)-vb;
```

249

```
13  t5=(xe*sqrt(1-pf^2))^2-(pf*xe)^2; // terms of
        quadratic equation in terminal voltage
14  p= [t1 0 t3 0 t5];
15  vt=roots(p);
16  ia=1/vt(2); // pu armature current
17  printf('Generator terminal voltage is %f pu\n',vt(2)
        );
18  printf('Armature current is %f pu\n',ia);
19  disp('case b');
20  Ef=vt(2)+ia*(pf-%i*sqrt(1-pf^2))*%i*xq;
21  de=atand(imag(Ef),real(Ef));
22  pa=acosd(pf); // power factor angle
23  id=ia*sind(de+pa); // d-axis component of armature
        current
24  Ef=abs(Ef)+id*(xd-xq);
25  printf('Load angle is %f degrees\n',de);
26  printf('Excitation voltage is %f pu',Ef);
```

Scilab code Exa 5.57 Calculating maximum power output and minimum pu excitation

```
1   clc;
2   xd=0.85; // reactance along d-axis
3   xq=0.55; // reactance along q-axis
4   vt=1; // pu bus voltage
5   Ef=1.2; // pu excitation EMF
6   // P=(Ef*vt*sin(de))/xd + (vt^2/2)*((1/xq)-(1/xd))*
        sin(2*de) where p is power and de is load angle
7   // for maximum power dp/dde(derivative with respect
        to load angle) is zero. Solving we get a
        quadratic equation whose terms are
8   p=[ (vt^2/2)*((1/xq)-(1/xd))*4 (Ef*vt)/xd -(vt^2/2)
        *((1/xq)-(1/xd))*2 ];
9   l=roots(p);
10  an=l(2);
```

```
11  de=acos(an)*(180/%pi); // load angle
12
13  pmax=(Ef*vt*sin(de*(%pi/180)))/xd + (vt^2/2)*((1/xq)
        -(1/xd))*sin(2*de*(%pi/180));
14  printf('Maximum power output that motor can supply
        without loss of synchronization is %f pu\n',pmax)
        ;
15  //  cos(de)=(vt^2/p)*((xd-xq)/(xd+xq))*sin(de)^3
        where de is load angle for minimum excitation EMF
16  // by trial and error value of de is
17  de=63;
18  P=1; // pu power
19  Ef=(P-((vt^2/2)*((xd-xq)/(xd*xq))*sind(2*de)))/((vt/
        xd)*sind(de));
20  printf('Minimum excitation EMF for machine to stay
        in synchronism is %f pu\n',Ef);
```

Scilab code Exa 5.58 Determining voltage regulation

```
1  clc;
2  p=3*10^6; // rated power of alternator
3  v=11000; // rated voltage of alternator
4  r=0.4; // per phase effective resistance
5  vl=12370; // line to line voltage at zero leading
        power factor
6  i=100; // load current at zero power factor
7  pf=0.8; // lagging power factor at which voltage
        regulation has to be determined
8  vt=vl/sqrt(3); // per phase terminal voltage
9  Ef=v/sqrt(3); // per phase excitation EMF
10  ia=p/(sqrt(3)*v); // full load phase current
11  // for zero power factor load angle=0
12  zs=(vt-Ef)/i; // synchronous impedance
13  xs=sqrt(zs^2-r^2); // synchronous reactance
14  // From phasor diagram
```

251

```
15 Ef1=sqrt((Ef*pf+ia*r)^2+(Ef*sqrt(1-pf^2)+ia*xs)^2);
      // excitation EMF at 0.8 power factor
16 vr=((Ef1-Ef)/Ef)*100;
17 printf('Voltage regulation at %f lagging power
      factor is %f percent',pf,vr);
```

Scilab code Exa 5.59 Determining synchronous reactance

```
1 clc;
2 v=11000; // voltage of infinite bus
3 po=15*10^6; // output power of alternator
4 pf=0.8; // operating power factor of synchronous
      machine
5 p=130; // percentage increase in excitation EMF
6 m=3; // number of phases
7 ia=po/(sqrt(3)*pf*v); // per phase armature current
8 vb=v/sqrt(3); // per phase bus voltage
9 //from phasor diagrams in fig 5.117(a) and 5.117(b)
10 xs=(sqrt(((p/100)*vb)^2-(vb*pf)^2)-(vb*sqrt(1-pf^2))
      )/ia; // synchronous reactance
11 printf('Synchronous reactance of machine is %f ohms\
      n',xs);
12 de=asind((po*xs)/(m*vb^2));
13 printf('Load angle of machine before excitation EMF
      is increased is %f degrees\n',de);
14 pf=cosd(de/2);
15 printf('Power factor of the machine before
      excitation EMF is increased is %f leading\n',pf);
16 ia=(2*vb*sind(de/2))/xs;
17 printf('Armature current of the machine before
      excitation EMF is increased is %f A',ia);
```

Chapter 6

Polyphase Induction Motors

Scilab code Exa 6.1 Determining the mechanical angle through which rotor moves

```scilab
 1 clc ;
 2 // after changing dc supply terminals from phase a
      to phase b
 3 disp('case a');
 4 P=2; // number of poles
 5 te=(2/P)*120;
 6 printf('Number of mechanical degrees through which
      rotor moves is %d degrees\n',te);
 7 disp('case b');
 8 P=4; // number of poles
 9 te=(2/P)*120;
10 printf('Number of mechanical degrees through which
      rotor moves is %d degrees\n',te);
11 disp('case c');
12 P=6; // number of poles
13 te=(2/P)*120;
14 printf('Number of mechanical degrees through which
      rotor moves is %d degrees\n',te);
```

Scilab code Exa 6.2 Determining 1 full load slip and rotor frequency 2 relative speed of stator with stator structure and rotor structure 3 relative speed of rotor with stator structure rotor structure and stator field

```
1  clc;
2  Nf=1440;  //full load speed
3  f=50;  //frequency
4
5  disp('case a');
6
7  P=fix((120*f)/Nf);  //formula for finding poles
8  mprintf('The number of Poles is %d\n',P);
9
10  disp('case b');
11
12
13  Ns=(120*f)/P;  //finding synchronous speed
14  s=(Ns-Nf)/Ns;  //finding slip at full load
15  f2=s*f;  //rotor frequency
16  mprintf('The full load slip is %f and the rotor
       frequency is %f Hz\n',s,f2);
17
18  disp('case c');
19
20
21  //speed of stator field w.r.t stator structure is Ns
22  Nss=Ns;
23  // answer for speed of stator field with respect to
       stator structure is given wrong in book
24  Wss=(2*%pi*Ns)/60;
25  Nsr=Ns-Nf;  //speed of stator field w.r.t rotor
       structure
26  Wsr=(2*%pi*Nsr)/60;
27  printf('The speed of stator field w.r.t stator is %f
```

```
        rad/sec ,%f rpm\n and w.r.t rotor is %f rad/sec
        ,%f rpm\n',Wss,Nss,Wsr,Nsr);
28
29 disp('case d');
30
31
32 //speed of rotor field w.r.t stator structure is Nf+
       Ns
33 Nrr=(120*f2)/P; //speed of rotor field w.r.t rotor
       structure
34 Nrs=Nf+Nrr;
35 // answer for speed of rotor field with respect to
       rotor structure is given wrong in book
36 Wrs=(2*%pi*Nrs)/60;
37
38 Wrr=(2*%pi*Nrr)/60;
39 //The stator and rotor fields are stationary w.r.t
       to each other
40 printf('The speed of rotor field w.r.t stator
       structure is %f rad/sec, %f rpm\n and w.r.t rotor
        structure is %f rad/sec, %f rpm and speed of
       rotor field w.r.t stator field is 0',Wrs,Nrs,Wrr,
       Nrr);
```

Scilab code Exa 6.3 Determining rotor speed

```
1
2 clc;
3 f=50; //frequency of stator
4 P=6;
5 NofO=90; //number of oscillation
6 f2=NofO/60; //rotor frequency
7 s=f2/f; //slip
8 Ns=(120*f)/P; //synchronous speed
9 Nr=Ns*(1-s); //rotor speed
```

```
10
11 mprintf('The rotor speed is %f rpm',Nr);
```

Scilab code Exa 6.4 Determining 1 speed of rotor 2 ratio of voltages at slip rings

```
1 clc;
2 f1=50; //frequency of supply
3 f2=20; //frequency required by the load
4 P=4;
5 //for part a
6
7 Nrf_ss=(120*f1)/P; //Speed of rotor field w.r.t
       stator structure
8 Nrf_rs=(120*f2)/P; //Speed of stator field w.r.t
       rotor structure
9 //Nr (+or-) speed of rotor field w.r.t rotor = speed
        of stator field w.r.t stator
10 //for +ve sign rotor must be driven in the direction
        of stator field at a speed
11 Nr1=Nrf_ss-Nrf_rs;
12 Nr2=Nrf_ss+Nrf_rs;
13 mprintf('The two speeds are %d and %d \n',Nr1,Nr2);
14
15
16 //for part b
17
18 //for rotor speed Nr1
19 s1=(Nrf_ss-Nr1)/Nrf_ss;
20 //for rotor speed Nr2
21 s2=(Nrf_ss-Nr2)/Nrf_ss;
22 //On evaluation the ratio of voltages is found as
23  R=s1/s2; //R=E1/E2
24 mprintf('The ratio of two voltages available at the
       slip rings at the two speeds is %d',R);
```

256

```
25
26 // for part c
```

Scilab code Exa 6.5 Determining 1 rotor current rotor power factor and torque 2 rotor current rotor power factor and torque after insertion of external resistance

```
 1  clc ;
 2  P =4;
 3  N =1440;
 4  f =50;
 5  r2 =0.2;
 6  x2 =1;
 7  E2 =120;
 8
 9  // mistake in Te_fl
10
11
12  // for part a
13  disp ( 'For part a ') ;
14  Ns =(120* f ) /P ;
15  I2_st =120/( sqrt (( r2 * r2 ) +( x2 * x2 ) ) ) ;
16  Rpf =( r2 ) /( sqrt (( r2 * r2 ) +( x2 * x2 ) ) ) ;
17  Ws =(2*3.14* Ns ) /60;
18  Te_st =(3/ Ws ) *( I2_st ) *( I2_st ) *( r2 /1) ;
19  s_fl =( Ns -N ) / Ns ;
20  I2_fl =( s_fl * E2 ) /( sqrt ( r2 * r2 +( s_fl * x2 * s_fl * x2 ) ) ) ;
21  Rpf_fl =( r2 ) /( sqrt ( r2 * r2 +( s_fl * x2 * s_fl * x2 ) ) ) ;
22  Te_fl =((3) *( I2_fl ) *( I2_fl ) *( r2 ) ) /( Ws * s_fl ) ;
23  RATIOst_fl = I2_st / I2_fl ;
24  RATIOtst_tfl = Te_st / Te_fl ;
25  mprintf ( 'At starting \n the rotor current is %f amp
        \n Rotor power factor is %f \n Torque is %f rad/
        sec \n ' , I2_st , Rpf , Te_st ) ;
26  mprintf ( 'At full load \n the rotor current is %f amp
```

```
           \n Rotor power factor is %f \n Torque is %f rad/
           sec\n',I2_fl,Rpf_fl,Te_fl);
27
28
29  //for part b
30  disp('For part b');
31  r2_n=r2+1;
32  I2_stn=E2/(sqrt((r2_n*r2_n)+(x2*x2)));
33  Rpf_stn=(r2_n)/(sqrt(((r2_n)*(r2_n))+((x2)*(x2))));
34  Te_stn=(3/Ws)*(I2_stn)*(I2_stn)*(r2_n/1);
35  mprintf('At starting \n the rotor current is %f amp
           \n Rotor power factor is %f \n Torque is %f rad/
           sec\n',I2_stn,Rpf_stn,Te_stn);
```

Scilab code Exa 6.6 Comparing rotor ohmic losses

```
1  clc;
2  P=6;
3  f=50;
4  N_f=960;
5  Ns=(120*f)/P;
6  n1=800;
7  n2=400;
8
9  s_fl=(Ns-N_f)/Ns;
10 s_1=(Ns-n1)/Ns;
11 s_2=(Ns-n2)/Ns;
12 Ratio_1=s_1/s_fl;
13 Ratio_2=s_2/s_fl;
14 mprintf('The Ratio at %d rpm is %f \n',n1,Ratio_1);
15 mprintf('The Ratio at %d rpm is %f \n',n2,Ratio_2);
```

Scilab code Exa 6.7 Determining 1 stator core loss 2 total rotor losses at full load 3 total rotor ohmic loss at full load 4 full load speed 5 internal torque shaft torque and efficiency

```
1  clc;
2  Psh =10000;
3  P=4;
4  f=50;
5  Pi =660;
6  Pw =420;
7  I_l =8;
8  rs =1.2;
9  Pi_fl =11200;
10 I_fl =18;
11 Ns =(120*f)/P;
12 Ws =(2*3.14*Ns)/60;
13
14
15 //for part a
16 disp('for part a ');
17
18 Pstl=Pi-Pw-((3*I_l*I_l*rs)/(3));
19 mprintf('The stator core loss is \n %f W \n',Pstl);
20
21 //for part b
22 disp('for part b ');
23
24 Pg=Pi_fl-Pstl-(3*(I_fl/sqrt(3))*(I_fl/sqrt(3))*rs);
25 Prl=Pg-Psh;
26 mprintf('The rotor loss is %f W \n',Prl);
27
28 //for part c
29 disp('for part c ');
30
31 Prol=Prl-Pw;
32 mprintf('The rotor ohmic loss is %f W \n',Prol);
33
34 //for part d
```

```
35 disp('for part d ');
36
37 s_fl=Prol/Pg;
38 Nr=Ns*(1-s_fl);
39 mprintf('Full Load speed of rotor is %f rpm \n',Nr);
40
41 //for part e
42 disp('for part e ');
43
44 Te=Pg/Ws;
45 Tsh=Psh/((Ws)*(1-s_fl));
46 E=(Psh/Pi_fl)*100;
47 mprintf('The internal torque is %f Nm \n The shaft
       torque is %f Nm \n The motor Efficiency is %f
       percent ',Te,Tsh,E);
```

Scilab code Exa 6.8 Determining slip of motor

```
1 clc;
2 E=0.9;
3 L=45000;
4 Tl=((1/0.9)-1)*L;
5
6 Rl=(Tl*2)/7; //According to the given conditoins
7 Pg=L+Rl+(Rl/2);
8
9 s=Rl/Pg;
10
11 mprintf('Slip is %f',s);
```

Scilab code Exa 6.9 Determining stator current rotor speed output torque
and efficiency

260

```
 1
 2  clc;
 3  P=4;
 4  r1=0.15;
 5  x1=0.45;
 6  r2=0.12;
 7  x2=0.45;
 8  Xm=28.5;
 9  s=0.04;
10  V=400;
11  f=50;
12  Pfixed=400;
13
14  t1=complex((r2/s),x2);
15  t2=complex(0,Xm);
16  t3=complex((r2/s),(x2+Xm));
17  T=(t1*t2)/t3;
18  Zab=complex(r1,x1)+T;
19  Rf=real(T);
20  I1=V/(sqrt(3)*abs(Zab));
21  ian=atand(imag(Zab),real(Zab));
22  Pf=cosd(ian);
23  I1_mag=sqrt(real(I1)*real(I1)+imag(I1)*imag(I1));
24  Psti=sqrt(3)*I1_mag*V*Pf;
25  Pg=3*I1*I1*Rf;
26  ns=(2*f)/P;
27  nr=(1-s)*ns;
28  Ws=2*3.14*ns;
29  Pm=(1-s)*Pg;
30  Psh=Pm-Pfixed;
31  To=(Psh)/((1-s)*Ws);
32  Psto=3*I1_mag*I1_mag*r1;
33  Prto=s*Pg;
34  Tls=Psto+Prto+Pfixed;
35  Pi=Psh+Tls;
36  E=(1-(Tls/Pi))*100;
37
38  mprintf('staror current = %f amp at lagging phase
```

```
          angle  of  %f w.r.t  input  voltage   \n  rotor  speed  =
            %f  rps  or  %f  rpm  output  torque  =  %f Nm \n
          Efficiency  =  %f  percent ',I1,ian,nr,nr*60,To,E);
```

Scilab code Exa 6.10 Determining 1 slip for maximum torque and maximum torque 2 rotor current and starting torque 3 external resistance inserted in rotor circuit 4 internal power developed 5 maximum internal power developed and corresponding slip

```
 1
 2
 3  clc ;
 4
 5  // from  6.9  problem
 6  P=4;
 7  r1 =0.15;
 8  x1 =0.45;
 9  r2 =0.12;
10  x2 =0.45;
11  Xm =28.5;
12  s =0.04;
13  V =400;
14  f =50;
15  Pfixed =400;
16  t =1.2; // rotor  effective  turns  ratio
17
18  // for  part  a
19  // According  to  the  conditions  and  diagram
20  t1=complex (r1 ,x1);
21  t2=complex (0 ,Xm);
22  t3=complex (r1 ,x2+Xm);
23  Ze =(t1*t2)/(t3);
24  Re=real (Ze);
25  Xe=imag (Ze);
26  t4=complex (Re ,(x2+Xe));
```

262

```
27  SmT=(r2)/(sqrt((Re*Re)+((x2+Xe)*(x2+Xe))));
28  Ve=(V/sqrt(3))*(Xm/(x2+Xm));
29  Ws=(4*%pi*f)/P;
30  Tem=(3/Ws)*Ve^2*(1/2)*(1/(Re+sqrt(Re^2+(x2+Xe)^2)));
31  Pm=Tem*(1-SmT)*Ws;
32  Psh=Pm-Pfixed;
33  Tsh=Psh/(Ws*(1-SmT));
34  mprintf('for  part  a  \n  slip  =  %f  \n  maximun  torque  =
          %f  Nm  \n  power  output  =  %f  KW  \n',SmT,Tem,    Psh
          /1000);
35
36
37  //for  part  b
38  s=1;
39  I2st=(Ve)/(sqrt((r2+Re)*(r2+Re)+(x2+Xe)*(x2+Xe)));
40  Test=(3/Ws)*I2st*I2st*(r2);
41  mprintf('  for  part  b  rotor  current  =  %f  A  \n  torque
          =  %f  Nm  \n',I2st,Test);
42
43
44  //for  part  c
45  R=sqrt(Re^2+(x2+Xe)^2)-r2;
46  Ra=R/(t^2);
47  mprintf('for  part  c  \n  external  resisitance  value  is
          =  %f  Ohm  \n',Ra);
48
49  //for  part  d
50  s1=0.04;
51  Pm=((3*(Ve)*(Ve))*r2*((1-s1)/s1))/(((Re+r2+((r2*(1-
          s1)/s1))))*((Re+r2+((r2*(1-s1)/s1))))+((x2+Xe)*(
          x2+Xe)));
52  mprintf('for  part  d  \n  power  developed  is  %f  KW  \n',
          Pm/1000);
53
54  //for  part  e
55  SmP=(r2)/(sqrt(((Re+r2)*(Re+r2))+((x2+Xe)*(x2+Xe)))+
          r2);
56  Pmn=((3/2)*Ve*Ve)/(Re+r2+sqrt((r2+Re)*(r2+Re)+(x2+Xe
```

263

```
     ) * ( x2 + Xe ) ) ) ;
57  mprintf ( ' f o r   p a r t   e   \n   s l i p   =   %f   \n   power   developed
        =   %f  KW ' , SmP , Pmn / 1000 ) ;
```

Scilab code Exa 6.11 Determining maximum internal torque for different cases

```
 1  clc ;
 2  P =4;
 3  r1 =0.15;
 4  x1 =0.45;
 5  r2 =0.12;
 6  x2 =0.45;
 7  Xm =28.5;
 8  s =0.04;
 9  V =400;
10  f =50;
11  Pfixed =400;
12
13  // from  problem  6.10
14  Re =0.1476;
15  Xe =0.443;
16  r2 =0.12;
17  x2 =0.45;
18
19  a=Xm /( x2 + Xm ) ;
20  // Ve=a*V1 ;
21  Wr =(4* %pi*f )/P ;
22  b =(3/ Wr ) *(1/2) *(1/(( Re ) +( sqrt (( Re*Re ) +(( x2 + Xe ) *( x2 +
        Xe ) ) ) ) ) ) ;
23  // Tem=b*Ve*Ve
24
25  // for  part  a
26  V1 =230;
27  Ve1 =a*V1 ;
```

```
28  Tem1=b*Ve1*Ve1;
29  mprintf('for part a \n maximum internal torque
        developed is %f Nm \n',Tem1);
30
31  //for part b
32  V2=115;
33  Ve2=a*V2;
34  Tem2=b*Ve2*Ve2;
35  mprintf('for part b \n maximum internal torque
        developed is %f Nm \n',Tem2);
36
37  //for f=25 Hz
38  Xe1=(1/2)*Xe;
39  x21=(1/2)*x2;
40  Ws1=(1/2)*Wr;
41
42
43  //for part c
44  V3=115;
45  Ve3=a*V3;
46  Tem3=(3/Ws1)*Ve3*Ve3*(1/2)*(1/((Re)+(sqrt((Re*Re)+((
        x21+Xe1)*(x21+Xe1))))));
47  mprintf('for part c \n maximum internal torque
        developed is %f Nm \n',Tem3);
48
49  //for f=5 Hz
50  Xe2=(1/10)*Xe;
51  x22=(1/10)*x2;
52  Ws2=(1/10)*Wr;
53
54
55  //for part d
56  f3=5;  //f3=(1/10)*f
57  V4=23;
58  Ve4=a*V4;
59  Tem4=(3/Ws2)*Ve4*Ve4*(1/2)*(1/((Re)+(sqrt((Re*Re)
        +(((x22+Xe2)*(x22+Xe2)))))));
60  mprintf('for part d \n maximum internal torque
```

265

```
     developed  is  %f Nm \n ',Tem4);
```

Scilab code Exa 6.13 determining 1 slip and rotor speed 2 rotor ohmic loss 3 starting torque 4 starting current 5 stator current 6 full load efficiency 7 slip at maximum torque for new resistance 8 full load slip 9 starting torque 10 starting current 11 rotor losses 12 power

```
1  // answer  match + roots
2
3  clc;
4  Pm=10000;
5  V=400;
6  f=50;
7  smT=0.1;
8  P=4;
9  Ns=(120*f)/P;
10
11  // for (i)
12  disp('(i)');
13  //As per given conditions the slip is given by
        equation  Sfl2 −0.4 Sfl +0.01=0
14  V=[1 -0.4 0.01];
15  R=roots(V);
16  Sfl=R(2);
17  Nr=Ns*(1-Sfl);
18  mprintf('The slip is %f \n The rotor speed is %f r.p
        .m',Sfl,ceil(Nr));
19
20  // for (ii)
21  disp('(ii)');
22  Pg=Pm/(1-Sfl);
23  Prot=Sfl*Pg;
24  mprintf('The rotor ohmic loss is %f W \n',Prot);
25
26  // for (iii)
```

266

```
27  disp('(iii)');
28  Tefl=Pg/(2*3.14*(Ns/60));
29  Test=(4*Tefl)/((smT)+(1/smT));
30  mprintf('starting torque is %f Nm \n',Test);
31
32  //for (iv)
33  disp('(iv)');
34  a=sqrt(((Sfl*Sfl)+(smT*smT))/((Sfl)*(Sfl)*(1+(smT)*(
       smT))));
35  mprintf('starting current = %f full load current\n',
       a);
36
37  //for (v)
38  disp('(v)');
39  // answer is slightly different in book
40  b=sqrt((1/2)*(1+(smT/Sfl)^2));
41  mprintf('stator current at maximun torque = %f full
       load current \n',b);
42
43  //for (vi)
44  disp('(vi)');
45  E=(Pm/Pg)*100;
46  mprintf('full load efficiency is = %f percent\n',E);
47
48  //for (vii)
49  disp('(vii)');
50  //As per given conditions
51  smT1=3*smT;
52  mprintf('New slip value is %f \n',smT1);
53
54  //for (viii)
55  disp('(viii)');
56  //According to the given conditions s1(2)-1.2s+0.09
57  VV=[1 -1.2 0.09];
58  RR=roots(VV);
59  s1=RR(2);
60  Nr1=Ns*(1-s1);
61  mprintf('full load slip is %f rotor speed is %f r.p.
```

267

```
      m',s1,Nr1);
62
63  //for (ix)
64  disp('(ix)');
65  Test1=((2)/((1/0.3)+(0.3)))*(2*Tefl);
66  mprintf('starting torque is %f Nm \n',Test1);
67
68  //for (x)
69  disp('(x)');
70  c=sqrt((s1^2+smT1^2)/(s1^2*(1+smT1^2)));
71  mprintf('starting current = %f full load current \n'
      ,c);
72
73  //for (xi)
74  disp('(xi)');
75  Protfl=s1*Pg;
76  mprintf('Rotor ohmic loss at full load torque is %f
      W \n',Protfl);
77
78  //for (xii)
79  disp('(xii)');
80  Pm1=(1-s1)*Pg;
81  E=Pm1/Pg;
82  mprintf('Efficiency is %f percent',E*100);
```

Scilab code Exa 6.14 Determining 1 maximum torque 2 full load rotor ohmic loss 3 slip at maximum torque 4 full load slip 5 full load torque

```
1
2  clc;
3  Pm=60000;
4  P=6;
5  s=0.04;
6  V=400;
7  smT=0.2;
```

268

```
 8  f =50;
 9  Ns =(120* f )/P ;
10
11  Ws =(2* % pi * Ns )/60;
12  Wr = Ws *(1 - s );
13  Tefl = Pm / Wr ;
14
15  // for part a
16  Tem =((( smT / s )+( s / smT ))/2)* Tefl ;
17  mprintf ( ' for part a \n the maximun torque is %f Nm\n
        ', Tem );
18
19  // for part b
20  Prot =( s /(1 - s ))*( Pm );
21  mprintf ( ' for part b \n the rotor ohmic loss is %f W
        \n ', Prot );
22
23  // for part c
24  smT1 =2* smT ;
25  mprintf ( ' for part c \n THe new slip is %f \n ', smT1 );
26
27  // for part d
28  // On analysis the slip is given by
29  s2 =0.084;
30  mprintf ( ' for part d \n full load slip is %f \n ', s2 );
31
32  // for part e
33  T2 = Pm /(( Ws )*(1 - s2 ));
34  mprintf ( ' for part e \n the full load torque is %f Nm
        \n ', T2 );
```

Scilab code Exa 6.15 Determining percentage reduction in rotor circuit resistance

1

269

```
2  clc;
3  sfl=0.05; //Full load slip
4  //Test/Tem=a
5  //Tfl/Tem=b
6  a=1/2;
7  b=1/1.6;
8  //As per the given equation we get smT1^2-2.5smT1
       +1=0
9  Q=[1 -2.5 1];
10 R=roots(Q);
11 smT1=R(2);
12
13 //For full load slip of 0.05 we get the equation
       smT2^2-0.20smT2+0.0025
14 Q1=[1 -0.20 0.0025];
15 R1=roots(Q1);
16 smT2=R1(1);
17
18 P=((smT1-smT2)/smT1)*100;
19 mprintf('Percentage reduction in rotor circuit
       resistance is %f percent',P);
```

Scilab code Exa 6.16 Determining external resistance inserted in rotor circuit and percentage change in current and power factor

```
1
2  clc;
3  r2=0.04;
4  x2=0.2;
5
6  //As per given conditions we get a quadratic
       equation in smT which is smT^2-4*smT+1
7  t1=1; t2=-4; t3=1;
8  p=[ t1 t2 t3];
9  smT=roots(p);
```

```
10
11  r22=x2* smT (2) ;
12  R=r22 - r2 ;
13  mprintf ( 'The external resistane needed to be
        inserted is %f Ohm \n' ,R) ;
14
15
16  // say V=400(Input voltage )
17  V =400;
18  // without external resistance
19  Ist=V/( sqrt ((r2) *(r2) +(x2) *(x2) ));
20  pf=r2/( sqrt ((r2) *(r2) +(x2) *(x2) ));
21
22  // with external resistance
23  Ist1=V/( sqrt ((r22) *(r22) +(x2) *(x2) ));
24  pf1=r22/( sqrt ((r22) *(r22) +(x2) *(x2) ));
25
26  a=(( Ist - Ist1 )/ Ist ) *100;
27  b=(( pf1 - pf )/ pf ) *100;
28  mprintf ( 'Percentage in starting current is %f \n' ,a)
        ;
29  mprintf ( 'Percentage in power factor is %f \n' ,b) ;
```

Scilab code Exa 6.17 Determining starting torque

```
1  clc ;
2  // r2 / x2=a
3  a = .5;
4  Test =25;
5
6  // for part a
7  disp ( 'For part a ') ;
8  //b=3(V1) 2/ r2Ws
9  // As per given conditions
10 b=Test *5;
```

```
11  //When rotor resistace is doubled
12  Test1=b*(1/4);
13  mprintf('The starting torque is %f Nm\n',Test1);
14  //for part b
15  disp('For part b');
16  //resisance is half
17  Test2=b*(2/17);
18
19
20  mprintf('The starting torque is %f Nm',Test2);
```

Scilab code Exa 6.18 Determining 1 slip at maximum torque 2 full load slip 3 rotor current in terms of full load current

```
1   //equation
2   clc;
3   //Test/Tefl=1.5;
4   d=1.5;
5   //Tem/Tefl=2.5;
6   e=2.5;
7
8   //for part a
9
10  //d=Test/Tefl;
11  //equation of torque gives following equation
12  Q=[1 -3.33 1];
13  R=roots(Q);
14  smT=R(2);
15  mprintf('The slip at maximun torque is %f \n',smT)
16
17  //for part b
18  //equation of torque gives
19  Q=[1 -1.665 0.111];
20  R=roots(Q);
21  sfl=R(2);
```

272

```
22  mprintf('The slip at full load is %f \n',sfl)
23
24  //for part c
25  //I2st=c*Isfl As per torque equation
26  c=sqrt((d)*(1/sfl));
27  mprintf('The rotor current = %f times full load
        current \n',c)
```

Scilab code Exa 6.19 Determining starting and maximum torque

```
1  clc;
2  Te=200;
3  s=0.04;
4  c=4; //given multiplying factor of leakage reactance
5
6  //3V*V=a*WS
7  a=Te*s*(((1+(1/s))*(1+(1/s)))+((c+c)*(c+c)));
8  Test=a*(1/((1+1)*(1+1)+(c+c)*(c+c)));
9  Tem=a*(1/2)*(1/(1+sqrt((1)*(1)+(c+c)*(c+c))));
10 mprintf('The starting torque is %f Nm \n The maximun
        Torque is %f Nm',Test,Tem);
```

Scilab code Exa 6.20 Determining 1 slip 2 rotor current rotor ohmic loss and rotor power factor 3 power output

```
1  clc;
2  sA=0.05; //slip
3
4  //for part a
5  disp('for part a ');
6  //Torque is proportional to s/r2
7  //As per given conditions sB=a*sA
8  a=4;
```

273

```
9  sB=a*sA;
10 mprintf('The slip is %d times previous slip and \n',
      a);
11
12 //for part b
13 disp('for part b ');
14 //I2 is directly proportional to s/r2
15 //As per given conditions I2B=b*I2A
16 b=sB/(a*sA);
17 //Rotor ohmic losses is directly proportional to I*I
      *r2
18 //As per given conditions P2=c*P1
19 c=a*b;
20 //As per given conditions Pf2=d*Pf1
21 d=b;
22 mprintf('rotor current for new rotor resistance is
      equal to initial rotor current \n Rotor ohmic
      losses for new rotor resistance=%f times initial
      ohmic losses \n power factor for new rotor
      resistance is equal to initial power factor',c);
23
24 //for part c
25 disp('for part c ');
26 //As per given conditions Wa=e*Ws
27 e=1-sA;
28 //Wb=f*Ws
29 b=1-sB;
30 //PB=g*PA
31 g=b/e;
32 mprintf('The power output is reduced to %f times
      previous value',g);
```

Scilab code Exa 6.21 Determining 1 rotor and stator input 2 starting torque

```
1 clc;
```

274

```
2  f =50;
3  P=6;
4  Pmsh =10000;  //Shaft Output
5  N=930;
6  Pw=600;
7  Pf =0.01* Pmsh;  //Friction and Windage losses
8  Ns =(120* f)/P;
9  NmT =800;  //Speed at maximum torque
10
11
12 //for part a
13 disp('for part a');
14 sfl =(Ns -N)/Ns;
15 Pm=Pmsh+Pf;
16 Pg=Pm/(1-sfl);
17 Pst=Pg+Pw;
18 mprintf('Total Rotor input is %f W \n Total Stator
      input is %f W \n',Pg,Pst);
19
20 //for part b
21 disp('for part b');
22 smT =(Ns -NmT)/Ns;
23 Ws =(2*%pi*Ns)/60;
24 Tefl=Pg/Ws;
25 Test =(((smT/sfl)+(sfl/smT))/2)*(2/((smT)+(1/smT)))*
     Tefl;
26 mprintf('Maximun Torque is %f Nm',Test);
```

Scilab code Exa 6.22 Determining maximum torque corresponding slip and starting torque

```
1 clc;
2 Pm=7500;
3 V=420;
4 f =50;
```

```
5  P=4;
6  s=0.04;
7  r1=1.2;
8  x1=1.4;
9  x2=1.4;
10 Xm=38.6;
11
12 //As per Thevenin's Equivalent circuit
13 Re=(r1*Xm)/(Xm+x2);
14 Xe=(x1*Xm)/(x2+Xm);
15 Ve=(V/sqrt(3))*(Xm/(x2+Xm));
16 r2=(3)*(1-s)*s*Ve*Ve*(1/Pm);
17 smT=r2/(sqrt((Re*Re)+((Xe+x2)*(Xe+x2))));
18 Tem=((3*Ve*Ve)/(((((120*f)/P)/60)*2*%pi))*(1/2)*(1/(
      Re+(sqrt((Re*Re)+((Xe+x2)*(Xe+x2)))))));
19 Test=((3*Ve*Ve)/(((((120*f)/P)/60)*2*%pi))*(r2/(((Re+
      r2)*(Re+r2))+((Xe+x2)*(Xe+x2)))));
20 mprintf('maximum torque is %f Nm \n slip is %f \n
      starting torque is %f Nm',Tem,smT,Test);
```

Scilab code Exa 6.23 Determining 1 maximum torque 2 starting torque 3 full load rotor ohmic loss 4 slip at full load 5 full load torque 6 slip at maximum torque

```
1  clc;
2  Pm=100000;
3  V=420;
4  P=6;
5  f=50;
6  sfl=0.04;
7  smT=0.2;
8
9  //for part a
10 disp('for part a');
11 Pg=Pm/(1-sfl);
```

```
12  Ws=(4*%pi*f)/P;
13  Tefl=Pg/Ws;
14  //a=Tefl/Tem
15  a=(1/(2/((sfl/smT)+(smT/sfl))));
16  Tem=a*Tefl;
17  mprintf('Maximum Torque is %f Nm \n',Tem);
18
19  //for part b
20  disp('for part b');
21  //b=Test/Tem
22  b=2/((1/smT)+(smT));
23  Test=b*Tem;
24  mprintf('The starting Torque is %f Nm \n',Test)
25
26  //for part c
27  disp('for part c');
28  Prot=sfl*Pg;
29  mprintf('Rotor Ohmic losses are %f W \n',Prot)
30
31  //for part d
32  disp('for part d');
33  //Output is proportional to (s(1-s))/r2
34  //Given conditions gives the equation as s1*s1-s1
       +0.0768
35  Q=[1 -1 0.0768];
36  R=roots(Q);
37  s1=R(2);
38  mprintf('Slip is %f \n',s1)
39
40  //for part e
41  disp('for part e');
42  Tefl=(Pm/(1-s1))/Ws;
43  mprintf('full-load torque is %f Nm \n',Tefl)
44
45  //for part f
46  disp('for part f');
47  smT1=2*smT;
48  mprintf('slip at maximum torque is %f',smT1);
```

277

Scilab code Exa 6.25 Determining 1 speed range of DC motor 2 KVA rating of stator 3 DC motor rating and maximum torque 4 number of poles 5 new speed range

```
1  clc;
2  P=10;
3  f=50;
4  Pm=48000;
5  pf=0.8;
6  f21=120; //min frequency range
7  f22=300; //max frequency range
8  Ns=(120*f)/P;
9
10 //for f2=300
11 Nr1=((120*f21)/P)-Ns;
12 //for f2=600
13 Nr2=((120*f22)/P)-Ns;
14 mprintf('Thus the dc motor changes speed from %f to
       %f rpm \n',Nr1,Nr2)
15
16 //for part b and c
17 s1=(Nr1+Ns)/Ns;
18 s2=(Nr2+Ns)/Ns;
19 Pr=Pm/pf;
20 Pr1=Pr/s1;
21 Pr2=Pr/s2;
22 R1=(s1-1)*Pr1*pf;
23 R2=(s2-1)*Pr2*pf;
24 T1=(R1*60)/(2*%pi*Nr1);
25 T2=(R2*60)/(2*%pi*Nr2);
26 // stator should be able to handle higher KVA
27 mprintf('KVA rating of induction motor stator is %f
       KVA\n',Pr1/1000)
28 mprintf('DC motor rating is %f KW \n Maximum torque
```

278

```
        output  from  DC  motor  is  %f Nm \n',R2/1000,T1);
29
30  //for  part  d
31  //When  speed  is  limited  to  2700  rpm
32  P1=((120*f22)-(120*f))/2700;
33  P1=ceil(P1);
34  mprintf('Number  of  Poles  is  %d \n',P1);
35
36  //for  part  e
37  Nr11=((f22*120)/P1)-((120*f)/P1);
38  Nr22=((f21*120)/P1)-((120*f)/P1);
39  mprintf('Thus  the  new  speed  range  of  dc  motor  is
        from  %f  to  %f  rpm \n',Nr22,Nr11);
```

Scilab code Exa 6.26 Determining starting torque

```
 1  clc;
 2  f=50;
 3  P=4;
 4  Pm=10000;  //Rated  output
 5  N=1425;
 6  Nm=1200;  //Speed  at  which  maximun  torque  is
       developed
 7
 8  Ns=(120*f)/P;
 9  s=(Ns-N)/Ns;
10  Ws=(2*%pi*Ns)/60;
11  Tefl=(Pm/Ws)*(1/(1-s));
12  smT=(Ns-Nm)/Ns;
13  Tem=Tefl*((s/smT)+(smT/s))*(1/2);
14  Test=Tem*(2)*(1/((1/smT)+(smT/1)));
15  mprintf('The  starting  torque  is  %f Nm',Test);
```

Scilab code Exa 6.27 Determining slip frequency and slip at maximum torque

```
1  clc ;
2  fs =2;  // slip  frequency
3  V =400;
4  f =50;
5  V2 =340;  // New  voltage
6  f2 =40;    // New  frequency
7  smT =0.1;  // slip  at  which  it  develops  maximum  torque
8
9  // maximun  torque 's  slip  is  directly  proportional  to
       (1/ f )
10  smT1 =( f/ f2 ) * smT ;
11
12  // Maximun  Torque  is  directly  proportional  to  (( V*V)
       /( f* f ))
13  s =fs / f ;
14  // Ted( Developed  Torque )  is  proportional  to  (Tem/smT)
       *( s /smT)
15  // Ted1(400V,50 Hz) proportional  a
16  a =(( V* V )/( f* f )) *( s / smT ) ;
17  // equating  the  developed  torque  equation
18  s1 =a *((( f2 ) *( f2 ))/(( V2 ) *( V2 ))) *( smT1 ) ;
19  fs1 =s1 * f2 ;
20  mprintf ( 'The  new  slip  frequency  is  %f  Hz ' ,fs1 ) ;
```

Scilab code Exa 6.28 Determining frequency of motor

```
1  clc ;
2  f =50;
3  V =440;
4  P =4;
5  N =1490;  // Rated  speed
6  N1 =1600;  // New  Speed
```

```
 7
 8  Ns=(120*f)/P;
 9  s=(Ns-N)/Ns;
10  //With neglecting resistances and leakage reactances
11  //Torque is directly proportional to s/(fr2)
12  //Appllying the condition for same torque we get
13  //a=s/f
14  a=(s/f);
15  //Ns/s=b
16  b=120/P;
17  //s=(Ns-N1)/Ns
18  //Using above equation we get equation (f*f)-7500f
        -400000
19  Q=[1 -7500 400000]
20  R=roots(Q);
21  f1=R(2);
22  mprintf('Value of new Frequency is %f Hz',f1);
```

Scilab code Exa 6.29 Determining power factor input current equivalent rotor current and torque developed maximum torque and corresponding speed

```
 1  //debug
 2  clc;
 3  V1=420; //supply voltage
 4  r1=2.95;
 5  x1=6.82;
 6  r2=2.08;
 7  x2=4.11;
 8  Iml=6.7; //magnetizing line current
 9  Pw=269; //core loss
10  s=0.03; //slip
11  P=12;
12  f=50;
13  N=(120*f)/P;
```

281

```
14  Ns=(120*f)/P;
15
16  Im=Iml/sqrt(3);
17  //V1=E1+Im(r1+jx1)
18  //Above equation on solving gives the solution as E1
        *E1+52.8E1-175572.65
19  Q=[1 52.8 -175572.62];
20  R=roots(Q);
21  E1=R(2);
22  Xm=E1/Im;
23  //As per the circuit diagram
24  a=r2/s;
25  Zf=(((r2/s)+x2*%i)*Xm*%i)/((r2/s)+((x2+Xm)*%i));
26  Rf=real(Zf);
27  Zab=complex((real(Zf)+r1),(imag(Zf)+x1));
28  I1=420/Zab;
29  I1M=sqrt((real(I1)*real(I1))+(imag(I1)*imag(I1)));
30  an1=atand(imag(I1),real(I1));
31  pf=cosd(atand(imag(I1)/real(I1)));
32  I2=I1*(Xm*%i)*(1/((r2/s)+((x2+Xm)*%i)));
33  an2=atand(imag(I2),real(I2));
34  I2M=sqrt((real(I2)*real(I2))+(imag(I2)*imag(I2)));
35  T=3*(60/(2*%pi*N))*I1M*I1M*Rf;
36
37  mprintf('The power factor is %f Lag\n The input
        current is %f A lagging by an angle of %f degrees
        \n The output rotor current is %f A lagging by
        an angle of %f degrees \n The Torque developed is
        %f Nm \n',pf,I1M,-an1,I2M,-an2,T);
38
39
40  //For maximun Torque
41  X1=x1+Xm;
42  Re=(r1*Xm)/X1;
43  Xe=(x1*Xm)/X1;
44  smT=r2/(sqrt((Re)*(Re)+(x2+Xe)*(x2+Xe)));
45  Nm=Ns*(1-smT);
46  Tem=3*(E1)*(E1)*(1/(Re+(sqrt((Re)*(Re)+(x2+Xe)*(x2+
```

282

```
      Xe)))))*(1/2)*(1/(2*%pi*(N/60)));
47 mprintf('maximum torque developed is %f Nm \n
       corresponding speed is %f rpm',Tem,Nm);
```

Scilab code Exa 6.30 Determining motor speed and power output

```
 1
 2 //In solution they have taken different value of
       speed at rated torque from what is given in
       question that is why answer is varying
 3 clc;
 4 P=4;
 5 Pm=10000;  //OUTPUT POWER
 6 f=50;  //FREQUENCY
 7 N=1440;   //SPEED AT WHICH RATED TORQUE IS OBTAINED
 8 Ns=(120*f)/P;   //SYNCHRONOUS SPEED
 9
10 s=(Ns-N)/Ns;
11 //Torque is directly proportional to the slip
12 //As per given conditions
13 s1=(1/2)*s;
14 Nr=Ns*(1-s1);
15 Pm1=(1/2)*(((Pm*60)/(2*%pi*N)))*((2*%pi*Nr)/(60));
16 mprintf('The motor speed is %f rpm \n The power
       output is %f W',Nr,Pm1);
```

Scilab code Exa 6.31 Determining percentage change in motor speed and losses

```
 1
 2 clc;
 3 N=1455;
 4 Ns=1500;  //General case considered in the problem
```

283

```
5  s1 =( Ns -N )/ Ns ;
6
7  // for  V1=0.9V
8  // V1/V=a
9  a =0.9;
10 // T=(3VVs)/(Wsr2)
11 // As  torque  is  constant
12 s2 =( s1 )/( a*a );
13 Nr = Ns *(1 - s2 );
14 // I=s1V/r2
15 // I22/I21=b
16 b =( s2*a )/ s1 ;
17 // Losses  Ratio=c
18 R = b*b ;
19
20 d =(( N - Nr )/ N )*100;
21 e =(( R -1)/1)*100;
22 mprintf ( 'Percentage  reduction  in  speed  is  %f  percent
      \n',d );
23 mprintf ( 'Percentage  reduction  in  ohmic  losses  is  %f
      percent \n',e );
```

Scilab code Exa 6.32 Determining no load speed of motor

```
1  clc ;
2  P =7500; // rated  power  of  induction  motor
3  v =400; // rated  voltage  of  motor
4  To =6; // no  load  torque
5  fs =0.04; // full  load  slip
6  p =6; // number  of  poles
7  f =50; // frequency
8  ns =(120* f )/ p ; // synchronous  speed
9  Tl =( P *60)/(2*% pi * ns *(1 - fs )); // full  load  torque
10 s =( To * fs * v ^2)/( Tl *( v /2)^2); // slip  at  no  load
11 no = ns *(1 - s );
```

284

```
12 printf('No load speed of motor is %f rpm\n',no);
```

Scilab code Exa 6.33 Determining minimum voltage impressed on motor and additional resistance inserted in rotor circuit

```
1  clc;
2  tr=2.5; // ratio of maximum torque to full load
        torque
3  sm=0.18; // maximum slip
4  r=1; // per phase rotor resistance
5  x2=r/sm; // rotor reactance
6  // using expression for tr we obtain a quadratic
        equation is s(full load slip) whose terms are
7  t1=1;
8  t2=-tr*2*sm;
9  t3=sm^2;
10 t=[ t1 t2 t3 ];
11 s=roots(t);
12 x=sqrt((2*x2)/(((r/s(2))^2+x2^2)*s(2)));
13 printf('Minimum voltage that could be impressed so
        that motor can supply rated torque is %f times
        rated voltage or %f percent of rated voltage\n',x
        ,x*100);
14 // from expression for maximum torque and full load
        torque we get a quadratic equation in R(externall
        resistance) whose terms are
15 t1=1;
16 t2=2-2*x2;
17 t3=1+x2^2-2*x2;
18 t=[ t1 t2 t3 ];
19 R=roots(t);
20 printf('External resistance inserted in rotor
        circuit is %f ohms\n',R(2));
```

Scilab code Exa 6.34 Comparing starting current starting torque and maximum torque at given frequencies and comparing voltage for given frequencies

```scilab
1  clc;
2  f1=50; // rated frequency of 3- phase induction
       motor
3  f2=40; // applied frequency
4  vr=0.9; // ratio of applied voltage to rated voltage
5  m=3; // number o phases
6  fr=f2/f1; // ratio of frequencies
7  ir=fr/vr;
8  printf('Ratio of starting current at %d Hz to
       starting current at %d Hz is %f \n',f1,f2,ir);
9  tr=(m/f1)*(f2/m)*(fr/vr)^2;
10 printf('Ratio of starting torque at %d Hz to
       starting torque at %d Hz is %f \n',f1,f2,tr);
11 tmr=(m/f1)*(f2/m)*(fr/(vr)^2);
12 printf('Ratio of maximum torque at %d Hz to maximum
       torque at %d Hz is %f \n',f1,f2,tmr);
13 vr1=sqrt((m/f1)*(f2/m)*fr^2);
14 printf('For the same starting torque ratio of
       voltage at %d Hz to ratio of voltage at %d Hz is
       %f\n',f2,f1,vr1);
15 vr2=sqrt((m/f1)*(f2/m)*fr);
16 printf('For the same maximum torque ratio of voltage
        at %d Hz to ratio of voltage at %d Hz is %f\n',
       f2,f1,vr2);
17 // answer for ratio of v2/v1 for same starting
       torque is slightly different from what is given
       in book
```

Scilab code Exa 6.35 Determining 1 slip at rated load 2 starting torque

```
1  clc;
2  P=60000; // rated power of 3−phase induction motor
3  p=4; // number of poles
4  f=50; // frequency
5  po=3000; // no load losses
6  i=0.3; // ratio of rated current to rated voltage
      when motor is prevented from rotating
7  pi=4000; // power input when motor is prevented from
        rotating
8  pr=0.3; //ratio of mechanical losses to no load
      losses
9  pm=pr*po; // mechanical losses
10 lsc1=po-pm; // stator core loss
11 lsc2=pi/2; // stator copper loss=rotor copper loss
12 disp('case a');
13 pg=P+pm+lsc2; // air gap power
14 s=lsc2/pg;
15 printf('Slip at rated load is %f\n',s);
16 disp('case b');
17 pim=pi/i^2; // power input to motor during blocked
      rotor test
18 pg=pim-lsc1-lsc2; // air gap power
19 ws=(4*%pi*f)/p; // synchronous speed
20 T=pg/ws;
21 printf('Starting torque at rated applied voltage is
      %f Nm\n',T);
```

Scilab code Exa 6.36 Finding ratio of starting current starting torque and maximum torque at given frequencies

```
1  clc;
2  sm=0.2; // slip
3  f1=50; // rated frequency of 3− phase induction
      motor
4  f2=45; // applied frequency
```

287

```
5  fr=f2/f1; // ratio of frequenciesir=fr/vr;
6  ir=sqrt((sm^2+1)/(sm^2+fr^2));
7  printf('Ratio of starting current at %d Hz to
       starting current at %d Hz is %f \n',f2,f1,ir);
8  tr=(sm^2+1)/(sm^2+fr^2);
9  printf('Ratio of starting torque at %d Hz to
       starting torque at %d Hz is %f \n',f2,f1,tr);
10 tmr=1/fr;
11 printf('Ratio of maximum torque at %d Hz to maximum
       torque at %d Hz is %f \n',f2,f1,tmr);
```

Scilab code Exa 6.37 Determining 1 maximum internal torque and internal starting torque 2 slip at maximum torque

```
1  clc;
2  P=20000; // rated power of induction motor
3  v=400; // rated voltage of motor
4  f=50;  // frequency
5  m=3; // number of phases
6  p=4; // number of poles
7  r1=0.2; // stator resistance
8  x=0.45; // stator/rotor leakage reactance
9  xm=18; // magnetising reactance
10 s=0.04; // slip
11 pg=P/(1-s); // air gap power
12 pr=s*pg; // rotor copper loss
13 vp=v/sqrt(3); // per phase voltage
14 ve=(vp*xm)/(x+xm); // Thevenin voltage
15 re=(r1*xm)/(x+xm); // Thevenin resistance
16 xe=(x*xm)/(x+xm); // Thevenin reactance
17 // using Thevenin's theorrm and rotor copper loss
       expression we get a quadratic equation in r2 (
       rotor resistance) whose terms are
18 t1=pr/s^2;
19 t2=((2*pr*re)/s)-(m*ve^2);
```

```
20  t3=pr*((xe+x)^2+re^2);
21  t=[ t1 t2 t3];
22  r2=roots(t);
23  disp('case a');
24  ws=(4*%pi*f)/p; // synchronous speed
25  Tm=(m*ve^2)/(ws*2*(re+sqrt(re^2+(x+xe)^2)));
26  printf('Maximum internal torque is %f Nm\n',Tm);
27  Ti=(m*ve^2*r2(1))/(ws*((re+r2(1))^2+(x+xe)^2));
28  printf('Initial starting torque is %f Nm\n',Ti);
29  disp('case b');
30  sm=r2(1)/(sqrt(re^2+(xe+x)^2));
31  printf('Slip at maximum torque is %f ',sm);
```

Scilab code Exa 6.41 Determining rotational losses and equivalent circuit parameters

```
 1  clc;
 2  P=10000; // rated power of squirrel cage induction
        motor
 3  V=400; // rated voltage of motor
 4  m=3; // number of phases
 5  // no load test results
 6  Vo=400; // applied voltage
 7  io=8; // no load current
 8  Po=250; // no load power
 9  // blocked rotor test
10  vb=90; // applied voltage
11  ib=35; // current
12  pb=1350; // input power
13  // ac resistance is 1.2 times dc resistance
14  rs=0.6; // per phase dc resistance of stator winding
15  pr=Po-m*(io/sqrt(3))^2*(1.2*rs); // no load
        rotational losses
16  znl=Vo/(io/sqrt(3)); // no load impedance
17  rnl=Po/(m*(io/sqrt(3))^2); // no load resistance
```

289

```
18  xnl=sqrt(znl^2-rnl^2); // no load reactance
19  zbr=vb/(ib/sqrt(3)); // block rotor test impedance
20  Rbr=pb/(m*(ib/sqrt(3))^2); // block rotor resistance
21  xbr=sqrt(zbr^2-Rbr^2); // block rotor reactance
22  x1=xbr/2;
23  xm=xnl-x1;
24  X2=xm+x1;
25  r2=(Rbr-1.2*rs)*(X2/xm)^2;
26  printf('Rotational losses are %f watts\n',pr);
27  printf('Stator resistance is %f ohms\n',1.2*rs);
28  printf('Rotor resistance is %f ohms\n',r2);
29  printf('Magnetising reactance is %f ohms\n',xm);
30  printf('Stator reactance is %f ohms\n',x1);
31  printf('Rotor reactance is %f ohms',x1);
```

Scilab code Exa 6.43 Determining starting torque

```
1  clc;
2  p=10000; // rated power of SCIM
3  v=420; // rated voltage of SCIM
4  p=4; // number of poles
5  f=50; // frequency of SCIM
6  // results of blocked rotor test
7  vb=210; // applied voltage
8  ib=20; // applied current
9  pb=5000; // power dissipated
10 l=300; // stator core loss
11 rs=0.6; // dc stator resistance
12 m=3; // number of phases
13 R=(rs*3)/2; // per phase stator resistance
14 Rs=1.2*R; // Effective stator resistance per phase
15 pi=pb*(v/vb)^2; // power input at rated voltage
       during block rotor test
16 is=ib*(v/vb); // stator current at rated voltage
       during block rotor test
```

```
17 pg=pi-m*(is/sqrt(3))^2*Rs-l; // air gap power
18 ws=(4*%pi*f)/p;
19 printf('synchronous speed is %f rad/sec\n',ws);
20 T=pg/ws;
21 printf('Starting torque is %f Nm',T);
```

Scilab code Exa 6.44 Determining starting torque

```
1 clc;
2 p=6; // number of poles
3 m=3; // number of phases
4 f=50; // frequency of motor
5 P=40000; // rated power of induction motor
6 v=400; // rated voltage of induction motor
7 // results of blocked rotor test
8 vb=200; // applied voltage
9 ib=110; // applied current
10 pf=0.4; // power factor
11 f1=45; // frequency at starting torque is to be
       determined
12 e=380; // voltage at starting torque is to be
       determined
13 vbp=vb/sqrt(3); // per phase voltage during blocked
        rotor test
14 zb=vbp/ib; // total impedance referred to stator
15 R=zb*pf; // net resistance referred to stator
16 X=zb*(sqrt(1-pf^2)); // net reactance referred to
       stator
17 X=X*(f1/f); // net reactance at frequency=45
18 Z=R+X*%i; // impedance at frequency=45
19 v1=e/sqrt(3); // per phase stator
20 is=v1/(Z); // starting current
21 ws=(4*%pi*f)/p; // synchronous speed
22 T=(3/ws)*abs(is)^2*(R/2);
23 printf('Starting torque is %f Nm',T);
```

291

Scilab code Exa 6.45 Determining 1 mechanical power output 2 net torque 3 efficiency of motor

```
1  clc;
2  v=400; // rated voltage of motor
3  m=3; // number of phases
4  r=2; // ratio of leakage reactance of stator to
         leakage reactance of rotor
5  ns=1000; // synchronous speed
6  n=960; // speed of motor
7  f=50; // frequency
8  // no load test results
9  Vo=400; // applied voltage
10 io=7.5; // no load current
11 pfo=0.135; // power factor
12 // blocked rotor test
13 vb=150; // applied voltage
14 ib=35; // current
15 pfb=0.44; // power factor
16 znl=Vo/(io*sqrt(3)); // no load impedance
17 rnl=znl*pfo; // no load resistance
18 xnl=sqrt(znl^2-rnl^2); // no load reactance
19 zbr=vb/(ib*sqrt(3)); // block rotor test impedance
20 Rbr=zbr*pfb; // block rotor resistance
21 xbr=sqrt(zbr^2-Rbr^2); // block rotor reactance
22 x2=xbr/3; // leakage reactance of rotor
23 x1=x2*2; // leakage reactance of stator
24 xm=xnl-x1; // magnetising reactance
25 r1=Rbr/2; // stator resistance/rotor resistance
26 V1=v/sqrt(3); // per phase stator voltage
27 Ve=(V1*xm)/(x1+xm); // thevenin voltage
28 Re=(r1*xm)/(x1+xm); // thevenin resistance
29 Xe=(x1*xm)/(x1+xm); // thevenin resistance
30 lr=sqrt(3)*v*io*pfo-m*io^2*r1; // rotational losses
```

292

```scilab
31  s=(ns-n)/ns; // slip
32  ir=Ve/(Re+(r1/s)+%i*(Xe+x2)); // rotor current at
        slip
33  Pm=m*abs(ir)^2*r1*((1-s)/s);
34  disp('case a');
35  Psh=Pm-lr;
36  printf('Mechanical power output is %f KW\n',Psh
        /1000);
37  disp('case b');
38  wr=((2*%pi*f)*(1-s))/m; // speed at which motor is
        running
39  T=Psh/wr;
40  printf('Net torque is %f Nm\n',T);
41  disp('case c');
42  lor=(Pm*s)/(1-s); // rotor/stator ohmic losses
43  Tl=lor*2+lr; // total losses
44  pi=Tl+Psh; // input power
45  ne=Psh/pi;
46  printf('Efficiency of motor is %f percent',ne*100);
```

Scilab code Exa 6.46 Determining new operating speed

```scilab
1   clc;
2   f=60; // frequency
3   p=6; // number of poles
4   n=1175; // speed of induction motor
5   re=0.06; // reduction in frequency
6   dv=0.1; // reduction in voltage
7   ws1=(120*f)/p; // synchronous speed
8   s1=(ws1-n)/ws1; // slip
9   s2=((1-re)/((1-dv)^2))*s1; // new slip
10  ws2=ws1*(1-s2)*(1-re);
11  printf('New operating speed is %f rpm',ws2);
```

293

Scilab code Exa 6.47 Determining 1 line current pf slip torque and efficiency 2 maximum possible pf and corresponding line current 3 maximum power output and maximum power input 4 slip at maximum torque and maximum torque 5 starting torque

```scilab
1  clc;
2  P=15000; // rated power of induction motor
3  V=400; // rated voltage of motor
4  f=50; // frequency
5  m=3; // number of phases
6  po=4; // number of poles
7  // no load test results
8  Vo=400; // applied line voltage
9  io=9; // no load line current
10 Po=1310; //  power input
11 // blocked rotor test
12 vb=200; // line voltage
13 ib=50; // line current
14 pb=7100; // input power
15 pfo=po/(sqrt(3)*io*Vo); // no load power factor
16 pfb=pb/(sqrt(3)*ib*vb); // short circuit power
        factor
17 isc=(V/vb)*ib; // short circuit current
18 printf('Short circuit current is %d A\n',isc);
19 // circle diagram is drawn in fig 6.37 with scale 6
        A= 1 cm
20 disp('case a');
21 x=6; // scale
22 pps=(V/sqrt(3))*x; // per phase power scale
23 fp=P/3; // full load power per phase
24 // as per the construction we obtain OP=6.05 which
        corresponds to full load current
25 ifl=x*6.05;
26 printf('Full load line current is %f A\n',ifl);
```

```scilab
27  // from fig angle POV1=29.5;
28  fpf=cosd(29.5);
29  printf('Full load power factor is %f lagging\n',fpf)
       ;
30  // full load slip is given by ratio ba/bP where ba
       =2.5, bP=38.5
31  fs=2.5/38.5;
32  printf('Full load slip is %f \n',fs);
33  ws=(2*%pi*f*120)/(po*60); // synchronous speed
34  Ft=(3.85*pps*m)/ws;
35  printf('Full load torque is %f Nm\n',Ft);
36  // efficiency is given by ratio aP/dP where aP=3.6,
       dP=4.45
37  ne=3.6/4.45;
38  printf('Full load efficiency is %f percent\n',ne
       *100);
39  disp('case b');
40  // OP turns out to be tangent to circular locus,
       therefore
41  disp('Maximum power factor is 0.87 lagging');
42  disp('Maximum line current is 36.3 A');
43  disp('case c');
44  // according to constructions given in solution we
       obtain AA'=5.3 from which maximum power output
       can be calculated
45  mpo=5.3*m*pps;
46  printf('Maximum output power is %f KW\n',mpo/1000);
47  // according to constructions given in solution we
       obtain CC'=8.45=radius of circle from which
       maximum power input can be calculated
48  mpi=8.45*m*pps+po;
49  printf('Maximum input power is %f KW\n',mpi/1000);
50  disp('case d');
51  // according to constructions given in solution we
       obtain BB'=6.65 from which maximum torque can be
       calculated
52  Mt=(6.65*m*pps)/ws;
53  printf('Maximum torque is %f Nm\n',Mt);
```

295

```
54  // maximum slip is given by ratio fb'/BB' where fb
       '=1.58, BB'=6.65
55  s=1.58/6.65;
56  printf('Maximum slip is %f \n',s);
57  disp('case e');
58  // according to constructions given in solution we
       obtain DG=3.3 from which starting torque can be
       calculated
59  St=(3.3*m*pps)/ws;
60  printf('Starting torque is %f Nm\n',St);
```

Scilab code Exa 6.48 Determining 1 external resistance inserted in rotor circuit 2 stator currents power factor 3 power output operating power factor and efficiency

```
 1  clc;
 2  P=4500; // rated power of induction motor
 3  V=400; // rated voltage of motor
 4  f=50; // frequency
 5  m=3; // number of phases
 6  // no load test results
 7  Vo=400; // applied line voltage
 8  io=4.2; // no load line current
 9  Po=480; //  power input
10  // blocked rotor test
11  vb=215; // line voltage
12  ib=15; // line current
13  pb=1080; // input power
14  rs=1.2; // rotor resistance referred to stator per
       phase
15  nt=2; // stator to rotor turns ratio
16  pfo=Po/(sqrt(3)*io*Vo); // no load power factor
17  pfb=pb/(sqrt(3)*ib*vb); // short circuit power
       factor
18  isc=(V/vb)*(ib*sqrt(3)); // per phase short circuit
```

```
            current
19  iop=io/sqrt(3); // per phase no load current
20  x=1; // scale 1 A= 1 cm
21  // circle diagram is drawn in fig 6.38
22  disp('case a');
23  // value of maximum torque at starting is not given
24  // now we note Bf=4.6 and B'f=1.25 using these
        values external resistance to be inserted is
        calculated
25  re=(4.6/1.25)*1.2; // external resistance
26  printf('External resistance referred to rotor is %f
        ohms\n',re/nt^2);
27  // as per the construction we obtain OB=11.24 which
        is needed to calculate starting line current
28  is=11.24*sqrt(3);
29  printf('Starting current is %f A\n',is);
30  // angle OBB'=45.5 which is needed to calculate
        power factor
31  pf=cosd(45.5);
32  printf('power factor is %f lagging\n',pf);
33  pps=x*V; // per phase power scale
34  fp=P/m; // full load power per phase
35  disp('case b');
36  // now torque is 1.25 times full load torque
37  // now we note NK=2.9 and N'K=2.1 using these values
         external resistance to be inserted is calculated
38  re=(2.9/2.1)*1.2; // external resistance
39  printf('External resistance referred to rotor is %f
        ohms\n',re/nt^2);
40  // as per the construction we obtain ON=14.35 which
        is needed to calculate starting line current
41  is=14.35*sqrt(3);
42  printf('Starting current is %f A\n',is);
43  // angle ONN'=58.3 which is needed to calculate
        power factor
44  pf=cosd(58.3);
45  printf('power factor is %f lagging\n',pf);
46  disp('case c');
```

```
47  // we obtain OH=5.35 which is per phase output
        current
48  // thetag=41.3
49  opf=cosd(41.3);
50  printf('Operating power factor is %f leading\n',opf)
        ;
51  po=m*5.35*V*opf;
52  printf('Output power is %f KW\n',po/1000);
53  // we note HL=3.95 and Ha=4.90 which is needed for
        efficiency
54  ne=3.95/4.9;
55  printf('Induction generator efficiency is %f percent
        ',ne*100);
```

Scilab code Exa 6.49 Determining 1 rotational and core loss 2 electric power output power factor and efficiency

```
1  clc;
2  p=150000; // rated power of induction motor
3  v=400; // rated voltage of induction motor
4  m=3; // number of phases
5  r1=0.02; // stator resistance
6  r2=0.04; // rotor resistance
7  xm=9.8; // magnetising reactance
8  x1=0.2; // leakage reactance of stator or rotor
9  s=0.04; // slip
10 n=0.93; // efficiency
11 disp('case a');
12 Zf=(((r2/s)+%i*x1)*%i*xm)/((r2/s)+%i*(xm+x1)); //
       per phase impedance offered to stator by rorating
        air gap field
13 z=r1+%i*x1; // impedance of stator
14 Z=Zf+z; // total impedance
15 is=v/(sqrt(3)*abs(Z)); // stator current
16 pg=m*is^2*real(Zf); // air gap power
```

298

```
17  l1=m*is^2*r1; // stator copper loss
18  l2=s*pg; // rotor copper loss
19  Tl=((1/n)-1)*p; // total losses
20  lr=Tl-(l1+l2); // rotational and core losses
21  printf('Rotational and core losses are %f W\n',lr);
22  disp('case b');
23  s=-0.04; // slip
24  Zf=(((r2/s)+%i*x1)*%i*xm)/((r2/s)+%i*(xm+x1)); //
       per phase impedance offered to stator by rorating
        air gap field
25  Z=Zf+z; // total impedance
26  is=v/(sqrt(3)*abs(Z)); // stator current
27  pf=cosd(180-atand(imag(Z),real(Z))); // power factor
28  printf('Power factor at the generator terminal is %f
        leading\n',pf);
29  po=sqrt(3)*is*v*pf; // electrical output
30  printf('Electrical output is %f KW\n',po/1000);
31  pg=-m*is^2*real(Zf); // air gap power
32  l1=m*is^2*r1; // stator copper loss
33  l2=-s*pg; // rotor copper loss
34  Tl=l1+l2+lr; // total losses
35  pi=Tl+po; // mechanical power input
36  ne=po/pi;
37  printf('Efficiency is %f percent',ne*100);
```

Scilab code Exa 6.50 Determining per phase value of capacitance and total KVA rating of capacitor bank

```
1  clc;
2  v=400; // balanced supply voltage
3  i=10; // line current
4  f=50; // frequency of supply
5  m=3; // number of phases
6  pf=0.8; // lagging power factor
7  pfn=0.9; // improved power factor
```

299

```scilab
 8  disp('staor in star');
 9  i=i*(pf-%i*sqrt(1-pf^2)); // complex form of line
       current
10  il=real(i)/pfn; // line current at improved power
       factor
11  il=il*(pfn-%i*sqrt(1-pfn^2)); // complex form of new
        line current
12  //from fig. 6.39
13  ic=-(imag(i)-imag(il)); // reactive component of
       current to be neutralised
14  // capacitor bank is star connected
15  xcs=v/(ic*sqrt(3)); // capacitance reactance
16  Cs=1/(2*%pi*f*xcs); // capacitance
17  K=m*ic*v/sqrt(3);
18  printf('Per phase value of capacitance for star
       connected capacitor bank is %f microfarad\n',Cs
       *10^6);
19  printf('Total KVA rating for star connected
       capacitor bank is %f KVA\n',K/1000);
20  // delta connected capacitor bank
21  // capacitor bank is delta connected, converting
       into equivalent star Xstar=Xdelta/3
22  xcd=v/(ic*sqrt(3)); // capacitance reactance
23  Cd=1/(2*%pi*f*xcd*m); // capacitance
24  printf('Per phase value of capacitance for delta
       connected capacitor bank is %f microfarad\n',Cd
       *10^6);
25  printf('Total KVA rating for delta connected
       capacitor bank is %f KVA\n',K/1000);
26  disp('Stator in delta');
27  i=(abs(i)/sqrt(3))*(pf-%i*sqrt(1-pf^2)); // complex
       form of line current
28  il=real(i)/pfn; // line current at improved power
       factor
29  il=il*(pfn-%i*sqrt(1-pfn^2)); // complex form of new
        line current
30  //from fig. 6.39
31  ic=-(imag(i)-imag(il)); // reactive component of
```

300

```
          current  to  be  neutralised
32  //  capacitor  bank  is  star  connected
33  //  capacitor  bank  is  star  connected ,  converting
        into  equivalent  delta  Xdelta=3*Xstar
34  xcs=v/ic;  //  capacitance  reactance
35  Cs=m/(2*%pi*f*xcs);  //  capacitance
36  K=m*ic*v;
37  printf('Per  phase  value  of  capacitance  for  star
        connected  capacitor  bank  is  %f  microfarad\n',Cs
        *10^6);
38  printf('Total  KVA  rating  for  star  connected
        capacitor  bank  is  %f  KVA\n',K/1000);
39  //   delta  connected  capacitor  bank
40  xcd=v/ic;  //  capacitance  reactance
41  Cd=1/(2*%pi*f*xcd);  //  capacitance
42  printf('Per  phase  value  of  capacitance  for  delta
        connected  capacitor  bank  is  %f  microfarad\n',Cd
        *10^6);
43  printf('Total  KVA  rating  for  delta  connected
        capacitor  bank  is  %f  KVA\n',K/1000);
```

Scilab code Exa 6.51 Determining capacitance of bank and each unit and percentage saving in energy lost

```
1  clc;
2  v=3300;  //  balanced  supply  voltage
3  p=500000;  //  rated  power  of  induction  motor
4  f=50;  //  frequency  of  supply
5  m=3;  //  number  of  phases
6  pf=0.7;  //  lagging  power  factor
7  pfn=0.9;  //  improved  power  factor
8  vc=420;  //  rated  voltage  of  capacitor
9  n=0.86;  //  motor  efficiency
10  i=p/(sqrt(3)*v*pf*n);  //  line  current
11  i=i*(pf-%i*sqrt(1-pf^2));  //  complex  form  of  line
```

301

```
         current
12  il=real(i)/pfn; // line current at improved power
        factor
13  il=il*(pfn-%i*sqrt(1-pfn^2)); // complex form of new
         line current
14  //from fig. 6.39
15  ic=-(imag(i)-imag(il)); // reactive component of
        current to be neutralised
16  // capacitor bank is delta connected
17  // capacitor bank is delta connected, converting
        into equivalent star Xstar=Xdelta/3
18  xcd=v/(ic*sqrt(3)); // capacitance reactance
19  Cd=1/(2*%pi*f*xcd*m); // capacitance
20  // now each capacitor is rated at 420 V, number of
        capacitor connected in series is
21  n=ceil(v/vc);
22  C=Cd*n;
23  printf('Per phase value of each capacitance for
        delta connected capacitor bank is %f microfarad\n
        ',C*10^6);
24  // let R be resistance of distribution circuit
25  // power lost without capacitor bank is m*abs(i)^2*R
26  // power lost with capacitor bank is m*abs(il)^2*R
        therefore
27  ps=(abs(i)^2-abs(il)^2)/abs(i)^2
28  printf('Percentage saving in losses is %f percent',
        ps*100);
```

Scilab code Exa 6.53 Determining tapping on autotransformer and line current at starting

```
1  clc;
2  fs=0.05; // full load slip
3  ir=6; // ratio of starting current and full load
        current
```

```
4 t=1; // ratio of starting torque to full load torque
5 x=sqrt(t/((ir^2)*fs));
6 printf('Tapping point is at %f percent\n',x*100);
7 is=x^2*ir;
8 printf('Starting current is %f times full load
     current\n',is);
```

Scilab code Exa 6.54 Determining starting torque in terms of full load torque

```
1 clc;
2 vr=0.4; // voltage applied during blocked rotor test
      as a fraction of rated voltage
3 ir=2.5; // line current during blocked rotor test as
      a fraction of full load current
4 tr=0.3; // starting torque as a fraction of rated
      torque
5 is=1.5; // starting current as a fraction of full
     load current
6 isc=ir/vr; // short circuit current at rated load
7 x=sqrt(is/isc); // starting current as a fraction of
      short circuit current at rated load
8 T=(x/vr)^2*tr;
9 printf('Starting torque is %f percent of full load
     torque',T*100);
```

Scilab code Exa 6.55 Determining maximum permissible KW rating of motor for different cases

```
1 clc;
2 v=440; // rated voltage of distribution circuit
3 im=1200; // maximum current that can be supplied
4 n=0.85; // efficiency of induction motor
```

303

```
 5  pf=0.8; // power factor of motor
 6  ir=5; // ratio of starting current to full load
        current
 7  disp('case a');
 8  il=im/ir; //rated line current
 9  p=sqrt(3)*v*il*n*pf;
10  printf('Maximum KW rating is %f KW\n',p/1000);
11  disp('case b');
12  x=0.8; // rated of applied voltage and stepped down
        voltage
13  il=im/(x^2*ir); //rated line current
14  p=sqrt(3)*v*il*n*pf;
15  printf('Maximum KW rating is %f KW\n',p/1000);
16  disp('case c');
17  // star-delta converter is same as autotransformer
        starter with 57.8 % tapping therefore
18  il=im/(0.578^2*ir); //rated line current
19  p=sqrt(3)*v*il*n*pf;
20  printf('Maximum KW rating is %f KW\n',p/1000);
```

Scilab code Exa 6.56 Determining 1 voltage applied to motor terminals 2 current drawn by motor 3 line current drawn from supply mains

```
 1  clc;
 2  p=10000; // rated power of motor
 3  v=400; // rated voltage of motor
 4  n=0.87; // full load efficiency
 5  pf=0.85; // power factor
 6  ir=5; // ratio of starting current to full load
        current
 7  tr=1.5; // ratio of starting torque to full load
        torque
 8  disp('case a');
 9  vt=v/sqrt(tr);
10  printf('Voltage applied to motor terminal is %f V\n'
```

```
       , vt ) ;
11  disp ( ' case  b ' ) ;
12  ifl =p /( sqrt (3) * v * pf * n ) ;  // full load current
13  il =( ir * vt * ifl ) / v ;
14  printf ( ' Current  drawn  by  motor  is  %f A\n ' , il ) ;
15  disp ( ' case  c ' ) ;
16  i =( vt / v ) * il ;
17  printf ( ' Line  current  drawn  from  supply  mains  is  %f A
        ' , i ) ;
```

Scilab code Exa 6.57 Determining ratio of starting torque to full load torque for different starters

```
1  clc ;
2  tm =2;  // ratio of maximum torque to full load torque
3  r =0.2;  // per phase rotor resistance referred to
        stator
4  x =2;  // per phase reactance referred to stator
5  s = r / x ;  // slip at maximum torque
6  disp ( ' case  a ' ) ;
7  ts1 =(2* s * tm ) /( s ^2+1) ;
8  printf ( ' Ratio  of  starting  torque  to  full  load  torque
        is  %f\n ' , ts1 ) ;
9  disp ( ' case  b ' ) ;
10 ts2 = ts1 /3;
11 printf ( ' Ratio  of  starting  torque  to  full  load  torque
        with  star−delta  starter  is  %f\n ' , ts2 ) ;
12 disp ( ' case  c ' ) ;
13 t =0.7;  // tapping point
14 ts3 = ts1 * t ^2;
15 printf ( ' Ratio  of  starting  torque  to  full  load  torque
        with  autotransformer  starter  is  %f\n ' , ts3 ) ;
```

305

Scilab code Exa 6.58 Determining resistance of feeder and percentage increase in starting torque

```
1  clc;
2  v=400; // supply voltage
3  f=50;  // frequency of supply
4  // results of short circuit test
5  V=200; // applied voltage
6  i=100; // short circuit current
7  pf=0.4; // power factor
8  zsc=(V*sqrt(3))/i; // short circuit impedance
9  rsc=zsc*pf; // short circuit resistance
10 xsc=sqrt(zsc^2-rsc^2); // short circuit reactance
11 R=sqrt(((xsc^2+rsc^2)-3*((rsc/3)^2+(xsc/3)^2))/2);
       // resistance of feeder
12 disp('with star connection');
13 ts1=(3*(v/sqrt(3))^2*rsc)/((R+rsc)^2+xsc^2); //
       product of starting torque and synchronous speed
14 // now two feeders are connected in parallel
       therefore net resistace of feeder is
15 rp=R^2/(R+R);
16 ts2=(3*(v/sqrt(3))^2*rsc)/((rp+rsc)^2+xsc^2); //
       product of new starting torque and synchronous
       speed
17 pr=(ts2-ts1)/ts1;
18 printf('Percentage increase in starting torque with
       star connection is %f percent\n',pr*100);
19 disp('With delta connection');
20 ts1=(3*(v/sqrt(3))^2*(rsc/3))/((R+(rsc/3))^2+(xsc/3)
       ^2); // product of starting torque and
       synchronous speed
21 // now two feeders are connected in parallel
       therefore net resistace of feeder is
22 rp=R^2/(R+R);
23 ts2=(3*(v/sqrt(3))^2*(rsc/3))/((rp+rsc/3))^2+(xsc
       /3)^2); // product of new starting torque and
       synchronous speed
24 pr=(ts2-ts1)/ts1;
```

```
25  printf('Percentage increase in starting torque with
        delta connection is %f percent\n',pr*100);
```

Scilab code Exa 6.59 Determining minimum allowable cross section area of each conductor of feeder

```
1  clc;
2  z=1.2+3*%i; // per phase standstill impedance
3  v=400; // supply voltage
4  l=500; // length of feeder line
5  tr=30; // maximum percentage reduction possible in
        starting torque
6  ro=0.02; // resistivity of feeder material
7  // equating expression of starting torque with and
        without feeder we get a quadratic equation in R (
        feeder resistance) whose terms are
8  t1=(1-(tr/100));
9  t2=2*real(z)*t1;
10 t3=t1*abs(z)^2-abs(z)^2;
11 p=[ t1 t2 t3 ];
12 R=roots(p);
13 A=(ro*l)/R(2);
14 printf('Minimum allowable cross section is %f mm^2',
        A);
```

Scilab code Exa 6.60 Determining starting torque for different connections

```
1  clc;
2  f=50; // frequency
3  p=10; // number of poles
4  pb=120000; // power dissipated in block rotor test
5  // stator ohmic losses = rotor ohmic losses
```

307

```
 6  pr=pb/2; // total rotor loss
 7  disp('case a');
 8  ws=(4*%pi*f)/p; // synchronous speed
 9  Ts=pr/ws;
10  printf('Starting torque is %f Nm\n',Ts);
11  disp('case b');
12  pr=pr/3; // total rotor ohmic loss
13  Ts=pr/ws;
14  printf('Starting torque is %f Nm\n',Ts);
```

Scilab code Exa 6.62 Determining value of resistance elements for a 4 step starter

```
 1  clc;
 2  s=0.03; // full load slip
 3  R=0.015; // rotor resistance per phase
 4  n=4; // number of step in starter
 5  al=s^(1/n);
 6  R1=R/s; // resistance of whole section
 7  r1=R1*(1-al);
 8  printf('Resistance of first element is %f ohms\n',r1
       );
 9  r2=r1*al;
10  printf('Resistance of second element is %f ohms\n',
       r2);
11  r3=r1*al^2;
12  printf('Resistance of third element is %f ohms\n',r3
       );
13  r4=r1*al^3;
14  printf('Resistance of fourth element is %f ohms\n',
       r4);
```

Scilab code Exa 6.63 Designing 5 sections of a 6 stud starter

308

```
1  clc;
2  fs=0.02; // full load slip
3  ir=2; // ratio of starting current to full load
       current
4  n=5; // number of section
5  R=0.03; // rotor resistance
6  //ir*ifl=(E2/R)*sm where ifl is full load current
       and E2 is induced voltage in rotor therefore
7  sm=fs*ir; // maximum slip
8  al=sm^(1/n);
9  R1=R/sm; // resistance of whole section
10 r1=R1*(1-al);
11 printf('Resistance of first element is %f ohms\n',r1
       );
12 r2=r1*al;
13 printf('Resistance of second element is %f ohms\n',
       r2);
14 r3=r1*al^2;
15 printf('Resistance of third element is %f ohms\n',r3
       );
16 r4=r1*al^3;
17 printf('Resistance of fourth element is %f ohms\n',
       r4);
18 r5=r1*al^4;
19 printf('Resistance of fifth element is %f ohms\n',r5
       );
```

Scilab code Exa 6.64 Determining 1 starting current and starting torque 2 external resistance to limit starting current and starting torque under this condition

```
1  clc;
2  v=3300; // rated voltage of induction motor
3  p=6; // number of poles
4  f=50; // frequency
```

```
 5  t =3.2;  // stator to rotor turns
 6  r =0.1;  // rotor resistance
 7  x =1;  // rotor leakage reactance
 8  R=t^2*r;  // rotor resistance referred to stator
 9  X=t^2*x;  // rotor reactance referred to stator
10  ws =(4*%pi*f)/p;  // synchronous speed
11  disp('case a');
12  is =(v/sqrt(3))/(sqrt(R^2+X^2));
13  printf('Starting current at rated voltage is %f A\n'
       ,is);
14  Ts =(3* is^2*R)/ws;
15  printf('Starting torque at rated voltage is %f Nm\n'
       ,Ts);
16  disp('case b');
17  is =50;  // starting current
18  // is=Vp/(sqrt((R+rex)^2+X^2) where rex is external
       resistance and Vp is phase voltage
19  // solving above equation we get a quadratic
       equation in rex whose terms are
20  t1 =1;
21  t2 =2*R;
22  t3 =(R^2+X^2) -((v/sqrt(3))/is)^2;
23  p =[ t1 t2 t3 ];
24  rex =roots(p);
25  printf('External resistance referred to rotor is %f
       ohms\n',rex(2)/t^2);
26  Ts =(3* is^2*(R+rex(2)))/ws;
27  printf('Starting torque under new condition is %f Nm
       \n',Ts);
```

Scilab code Exa 6.65 Determining 1 line current 2 power returned to three phase supply 3 efficiency of motor

```
 1  clc;
 2  p =6;  // number of poles
```

310

```
 3  v=400; // rated voltage of induction motor
 4  m=3; // number of phases
 5  f=50; // frequency
 6  r1=0.2; // stator resistance
 7  r2=0.5; // rotor resistance
 8  xm=48; // magnetising reactance
 9  x1=2; // leakage reactance of stator or rotor
10  n=1050; // speed of motor
11  ns=(120*f)/p; // synchronous speed
12  s=(ns-n)/ns; // operating slip
13  disp('case a');
14  Zf=(((r2/s)+%i*x1)*%i*xm)/((r2/s)+%i*(xm+x1)); //
       per phase impedance offered to stator by rorating
        air gap field
15  z=r1+%i*x1; // impedance of stator
16  Z=Zf+z; // total impedance
17  is=v/(sqrt(3)*abs(Z)); // stator current
18  pf=cosd(atand(imag(Z),real(Z)));
19  printf('Stator line current is %f A\n',is);
20  disp('case b');
21  Po=m*(v/sqrt(3))*is*pf;
22  // negative power indicates induction machine is
       acting as generator
23  printf('Power fed back to 3 phase supply system is
       %f W\n',-Po);
24  disp('case c');
25  lr=600; // rotational and core losses
26  pg=m*is^2*real(Zf); // air gap power
27  l1=m*is^2*r1; // stator copper loss
28  l2=s*pg; // rotor copper loss
29  Tl=lr+l1+l2; // total losses
30  pi=-Po+Tl; // mechanical power input
31  ne=-Po/pi;
32  printf('Efficiency of induction motor is %f percent\
       n',ne*100);
```

Chapter 7

Armature Windings

Scilab code Exa 7.1 Determining the number of commutator segments back pitch and front pitch and commutator pitch

```
1  clc;
2  p=6;  // number of poles
3  c=40;  // number of coils
4  w=2;  // winding pitch for simplex lap winding
5  printf('Number of commutator segments is equal to
       number of coils=%f\n ',c);
6  k=1/3;  // integer added(or subtracted) to calculate
       back pitch to make it an odd integer
7  yb=((2*c)/p)-k;
8  printf('Back pitch is %f \n',yb);
9  yf=yb-w;
10 printf('Front pitch for progressive winding is %f\n'
       ,yf);
11 yf=yb+w;
12 printf('Front pitch for retrogressive winding is %f\
       n',yf)
13 yc=1;
14 printf('For simplex lap winding, commutator pitch is
        equal to %f ',yc);
```

Scilab code Exa 7.2 Designing the progressive simplex lap winding with two coil sides per slot

```
1  clc;
2  p=4; // number of poles
3  c=12; // number of coils
4  // Number of commutator segments is equal to number
        of coils=12
5  // Each coil has two coil side therefore total coil
        sides are 24
6  s=(2*c)/2 ; // total number of slots required
7  k=1; // integer added(or subtracted) to calculate
        back pitch to make it an odd integer
8  w=2; // winding pitch
9  yb1=((2*c)/p)-k; // back pitch
10 // or
11 yb2=((2*c)/p)+k; // back pitch
12 disp('Back pitch is ');
13 disp(yb1,'or',yb2);
14 yf1=yb1-2; // front pitch for yb=5
15 yf2=yb2-2; // front pitch for yb=7
16 disp('front pitch for progressive winding is ');
17 disp(yf1,'or',yf2);
18 disp('It is desirable that (yb+yf)/2 should be equal
         to pole pitch that is 6(in terms of coil sides
        per pole). So choose yb=7 and yf=5');
19 disp('Commutator pitch for progressive lap winding
        is ');
20 disp(1);
```

Scilab code Exa 7.3 Determining winding table and position of brushes on commutator

313

```scilab
1  clc;
2  p=4; // number of poles
3  s=14; // number of slots
4  cp=2; // coil sides per slots
5  w=2; // winding pitch
6  C=(s*cp)/2; // number of coils
7  yb=(2*C)/p;
8  disp('Back pitch is');
9  disp(yb);
10 yf=yb-w;
11 disp('Front pitch is');
12 disp(yf);
13 disp('winding table for progressive lap winding is')
      ;
14 disp('(1-8)-(3-10)-(5-12)-(7-14)-(9-16)-(11-18)
      -(13-20)-(15-22)-(17-24)-(19-26)');
15 disp('-(21-28)-(23-2)-(25-4)-(27-6)');
16 disp('from winding diagram')
17 disp('Brush A is touching segments 1 and 2 partly');
18 disp('Brush B is at segment 5');
19 disp('Brush C is at segment 8');
```

Scilab code Exa 7.5 Designing a simplex lap winding with the given details

```scilab
1  clc;
2  disp('case a');
3  s=30; // number of slots
4  c=60; // number of coils
5  p=4; // number of poles
6  k=1; // integer added(or subtracted) to calculate
      back pitch to make it an odd integer
7  tc=c*2; //  total coil sides
8  u=tc/s; // coil sides per slots
9  yb1=(tc/p)+k;
```

314

```
10  yb2=(tc/p)-k;
11  disp('Back pitch is');
12  disp(yb1);
13  disp('or');
14  disp(yb2);
15  disp('for back pitch=29, top coil sides 1 and 3 in
        slot 1 are connected to bottom coil 30 and 32 in
        slot 8. Due to this arrangement split coils can
        be avoided. But for back pitch= 31, coil sides 34
         which is in slot 9 has to be used, so split
        coils are needed ')
16  disp('case b');
17  s=20; // number of slots
18  c=60; // number of coils
19  p=4; // number of poles
20  k=1; // integer added(or subtracted) to calculate
        back pitch to make it an odd integer
21  tc=c*2; //  total coil sides
22  u=tc/s; // coil sides per slots
23  yb1=(tc/p)+k;
24  yb2=(tc/p)-k;
25  disp('Back pitch is');
26  disp(yb1);
27  disp('or');
28  disp(yb2);
29  disp('for back pitch=29, top coil sides 1,3 and 5
        are connected to bottom coil 30, 32 and 34. Due
        to this arrangement split coils cannot be avoided
        . But for back pitch= 31, coil sides 1,3 and 5
        are connected to bottom coil sides 32, 34 and 36
        which are in slot 6,so split coils are not needed
         ');
```

Scilab code Exa 7.6 Designing a simplex wave winding with given details

315

```
1  clc;
2  p=4; // number of poles
3  s=11; // number of slots
4  ts=2; // coil sides per slot
5  C=(s*ts)/2; // total coils
6  w=((2*C)+2)/(p/2); // winding pitch
7  // since both back and front pitch should be odd
      choose
8  Yb=7;
9  Yf=5;
10 disp('Back pitch is ')
11 disp(Yb);
12 disp('Front pitch is ')
13 disp(Yf);
14 yc=(C+1)/(p/2);
15 disp('commutator pitch ');
16 disp(yc);
17 disp('Using this data winding diagram can be drawn')
      ;
18 disp('Winding table is ');
19 disp('(1-8)-(13-20)-(3-10)-(15-22)-(5-12)-(17-2)
      -(7-14)-(19-4)-(9-16)-(21-6)-(11-18)-1');
```

Scilab code Exa 7.7 Designing the simplex wave winding

```
1  clc;
2  p=6; // number of poles
3  s=72; // number of slots
4  ts=4; // number of coil sides per slot
5  C=(s*ts)/2; // total number of coils
6  // To make commutator pitch an integer one coil is
      made dummy coil therefore
7  C=C-1;
8  yc=(C+1)/(p/2);
9  disp('commutator pitch ');
```

```
10  disp(yc);
11  yw=((2*C)+2)/(p/2);
12  disp('Winding pitch is');
13  disp(yw);
14  // since back and front pitch should be odd choose
15  yb=49;
16  disp('Back pitch is');
17  disp(yb);
18  yf=47;
19  disp('Front pitch is');
20  disp(yf);
```

Scilab code Exa 7.8 Designing the winding and determining the speed

```
1  clc;
2  p=4; // number of poles
3  z=2540; // number of conductors
4  s=32; // number of slots
5  c=127; // number of commutator sectors=total number
        of coils
6  v=500; // induced voltage required
7  f=5*10^-3; // field flux per pole
8  a=2; // number of parallel paths
9  zs=ceil(z/s); // conductors per slot
10  // for zs=80
11  Z=zs*s; // total conductors
12  t=floor(Z/(2*c)); // turn per coil
13  C=Z/(2*t); // actual number of coils
14  // It is necessary that actual coils should be same
        as commutator segments so one coil is made dummy
15  disp('commutor pitch is')
16  disp((c+1)/(p/2));
17  disp('or');
18  disp((c-1)/(p/2));
19  disp('Winding pitch is')
```

317

```
20  disp(((2*c)+2)/(p/2));
21  disp('or');
22  disp(((2*c)-2)/(p/2));
23  disp('For progressive winding, back pitch=65 and
        front pitch=63');
24  disp('For retrogressive winding, back pitch=63 and
        front pitch=63');
25  // since dumy coil is not in circuit, number of
        active conductor is
26  Z=c*t*2;
27  n=(v*a*60)/(f*Z*p);
28  printf('Speed for required induced voltage is %f rpm
        ',n);
```

Scilab code Exa 7.9 Determining the suitable arrangement of equilizer ring for given details

```
1  clc;
2  p=8; // number of poles
3  c=240; // number of coils
4  r=10; // number of equilizer ring
5  Yeq=(2*c)/p;
6  printf('Equipotential pitch is %f coils\n',Yeq);
7  Ytp=(2*c)/(r*p);
8  printf('Tapping point pitch is %f coils',Ytp);
9  disp('Arrangement is shown in tabular form in
        example 7.9');
```

Scilab code Exa 7.10 Determining the resistance measured between two adjacent commutator segments

```
1  clc;
2  disp('b(1)');
```

```
3 c =12;   // number of coils
4 r =0.1;  // resistance of each coil
5 // any one coil connected to commutator segment is
      in parallel with other 11 series connected coils
      therefore
6 R=11*r;  // resistance of 11 coil
7 req=(r*R)/(r+R);
8 printf('Resistance measured between two adjacent
      commutator segments is %f ohm\n',req);
```

Scilab code Exa 7.11 Determining the details for winding diagram and distribution factor

```
1 clc;
2 disp('a');
3 s=24;  // total number of slots
4 p=4;  // number of poles
5 np=3;  // number of phases
6 ph=60;  // phase spread
7 // given armature has double layer winding and full
      pitch coil span
8 v=(p*180)/s;
9 printf('Slot angular pitch is %d degrees\n',v);
10 disp('Number of adjacent slots in one phase belt is '
      );
11 disp(ph/v);
12 cs=s/p;
13 printf('Coil span is %d slots\n',cs);
14 disp('Using this data winding table for the three
      phases is shown in Ex7.11')
15 disp('d');
16 sp=s/(p*np);  // slots per pole per phase
17 disp('Distribution factor is ');
18 disp(sind(ph/2)/(sp*sind(v/2)));
```

319

Scilab code Exa 7.12 Determining the details for winding diagram and distribution factor

```
1  clc;
2  disp('a');
3  s=24; // total number of slots
4  p=4; // number of poles
5  np=3; // number of phases
6  ph=120; // phase spread
7  // given armature has double layer winding and full
       pitch coil span
8  v=(p*180)/s;
9  printf('Slot angular pitch is %d degrees\n',v);
10 disp('Number of adjacent slots in one phase belt is '
       );
11 disp(ph/v);
12 cs=s/p;
13 printf('Coil span is %d slots\n',cs);
14 disp('Using this data winding table for the three
       phases is shown in Ex7.12')
15 disp('d');
16 sp=s/(p*np); // slots per pole per phase
17 disp('Distribution factor is');
18 disp(sind(ph/2)/(sp*sind(ph/(2*sp))));
```

Scilab code Exa 7.13 Determining the details for winding table and effective turns per phase

```
1  clc;
2  np=3; // number of phase
3  sp=9; // slots per pole
4  zs=4; // conductors per slot
```

```
5  f=0.8; // coil span as a fraction of pole pitch
6  ph=60; // phase spread
7  v=180/sp; // slot angular pitch
8  disp('Number of adjacent slots belonging to any
      phase is ');
9  disp(ph/v);
10 printf('Pole pitch is %f slots\n',sp);
11 c=floor(0.8*sp);
12 printf('Coil span is of %f slots\n',c);
13 disp('Using this data, winding table is shown in Ex7
      .13');
14 t=(sp*zs*4)/2; // total turns in machine
15 spp=sp/np; // slots per pole per phase
16 kd=sind(ph/2)/(spp*sind(v/2)); // distribution
      factor
17 cp=c*v; // coil span in degrees
18 e=180-cp; // chording angle
19 kp=cosd(e/2); // coil span factor
20 kw=kd*kp; // winding factor
21 tp=(t*kw)/np;
22 printf('Number of effective turns per phase is %f',
      tp);
```

Scilab code Exa 7.15 Determining and designing the details for winding of 3 phase machine

```
1  clc;
2  s=24; // number of slots
3  p=4; // number of poles
4  ph=60; // phase spread
5  ap=(p*180)/s; // slot angular pitch
6  pp=s/p; // pole pitch
7  printf('Pole pitch is %d slots\n',pp);
8  printf('slot angular pitch is %d degrees',ap);
9  disp('using these data, half coil and whole coil
```

single layer concentric windings diagram are
drawn');

Chapter 8

Appendix A

Scilab code Exa 8.1 Determining the reluctance of ring and current required to establish the required flux

```
1  clc;
2  // answer is given wrong in the book
3  d=0.2; // mean diameter of mild steel ring
4  ac=50*10^-4; // cross sectional area of core
5  uo=4*%pi*10^-7; // free space permeability
6  ur=800; // relative permeability
7  f=1*10^-3; // required flux
8  N=200; // Number of turns
9  l=%pi*d // length of core
10 R=l/(uo*ur*ac); // reluctance of ring
11 printf('reluctance offered by ring is %f AT/Wb\n',R)
       ;
12 mmf=f*R; // mmf produced in ring
13 i=mmf/N;
14 printf('current required to produce the desired flux
        is %f A',i);
```

Scilab code Exa 8.2 Determining the exciting current in coil

```
 1  clc;
 2  ur=10000;  // relative permeability of iron
 3  lc=0.5;  // core length
 4  lg=4*10^-3;  // air gap length
 5  N=600;  // number of turns
 6  B=1.2;  // desired flux density
 7  uo=4*%pi*10^-7;  // free space permeability
 8  Ac=25*10^-4;  // core area
 9  mfc=(B*lc)/(uo*ur);  // mmf for core
10  mfg=(B*lg)/uo;  // mmf for air gap
11  mft=mfc+mfg;  // net mmf
12  i=mft/N;
13  printf('exciting current required to establish the
         desired flux is %f A',i);
```

Scilab code Exa 8.3 Determining the exciting current with and without magnetic leakage and fringing

```
 1  clc;
 2  lc=0.5;  // core length in metre
 3  dc=2.85*10^-2;  // diameter of cross section of core
 4  lg=2*10^-3;  // length of air gap
 5  N=500;  // Number oof turns of coil
 6  f=0.8*10^-3;  // air gap flux
 7  uo=4*%pi*10^-7;  // permeability of free space
 8  HATM=[1500 2210 2720 3500 4100];
 9  BT=[0.9 1.1 1.2 1.275 1.3];
10  plot(HATM,BT);
11  xlabel('magnetic field intensity');
12  ylabel('flux density');
13  disp('case a');
14  ur=500;  // relative permeability
15  Ac=(%pi/4)*dc^2;  // Area of core
16  Rlg=lg/(uo*Ac);  // reluctance of air gap
17  Rlc=lc/(uo*ur*Ac);  // reluctance of iron core
```

324

```
18  Rt = Rlg + Rlc ;  // Total reluctance
19  I =( f * Rt )/ N ;  // Exciting current
20  printf ( 'Exciting current in coil is %f A\n' , I );
21  disp ( 'case b ');
22  Ag =(% pi /4)*( dc +2* lg )^2;  // air gap area
23  Rlg = lg /( uo * Ag );  // reluctance of air gap
24  I =( f *( Rlc + Rlg ))/ N ;  // Exciting current
25  printf ( 'Exciting current after accounting for flux
        fringing is %f A\n' , I );
26  disp ( 'case c ');
27  Bg = f / Ac ;  // Air gap flux density
28  Atg =( Bg * lg )/ uo ;  // air gap mmf
29  // from the plot we can get the values of core flux
        density and magnetic field intensity
30  Bc =1.245;  // core flux density in Tesla
31  H =3200;  // magnetic field intensity in Ats/m
32  Atc = H * lc ;  // core mmf
33  mt = Atg + Atc ;  // total mmf
34  I = mt / N ;  // Exciting current
35  printf ( 'Exciting current for third case is %f A' , I );
```

Scilab code Exa 8.4 Determining the coil current for different values of relative permeability

```
1  clc ;
2  N =1000;  // Number of turns
3  f =1*10^ -3;  // flux in central limb
4  Ac =8*10^ -4;  // Area of central limb
5  Ao =4*10^ -4;  // Area of outer limb
6  lg =2*10^ -3;  // length of air gap
7  lc =0.15;  // length of central limb in metre
8  lo =0.25;  // length of outer limb in metre
9  uo =4*% pi *10^ -7;  // permeability of free space
10  disp ( 'case a ');
11  // for ur = infinity , reluctance offered by cast steel
```

```
          is  zero
12  Rl1=lg/(uo*Ao); // reluctance offered by outer limb
13  Rl2=lg/(uo*Ac); // reluctance offered by central
       limb
14  // Assuming magnetic circuit as a close circuit,
       applying KVl in one of loop gives
15  I=(f*(Rl2+(Rl1/2)))/N;
16  printf('Coil current for first case is %f A\n',I);
17  disp('case b');
18  ur=6000; // relative permability
19  Rlc1=(lc+lo)/(uo*ur*Ao); // reluctance of outer
       steel core (including the top)
20  Rlc2=(lc)/(uo*ur*Ac); // reluctance offered by
       central steel core
21  r=(Rlc1+Rl1)/2; // resultant of outer reluctance
22  // By kVL we get
23  I=(f*(Rlc2+Rl2+r))/N;
24  printf('Coil current for second case is %f A\n',I);
```

Scilab code Exa 8.5 Calculating the exciting current to set up required flux

```
1  clc;
2  N=500; // number of turns in central limb
3  ac=600*10^-6; // cross sectional area of central
       limb
4  ao=375*10^-6; // cross sectional area of outer limb
5  f=0.9*10^-3; // required flux in Weber
6  lg=0.8*10^-3; // length of air gap
7  lc=180*10^-3; // length of central limb
8  lo=400*10^-3; // length of outer limb
9  uo=4*%pi*10^-7; // free space permeability
10 Bg=f/ac; // air gap flux density
11 Hg=Bg/uo; // magnetic field intensity in air gap
12 mg=Hg*lg; // mmf required for air gap
```

326

```
13  // from fig A.7, for B=1.5T, H for cast steel is 3000
        Ats/m
14  H=3000; // magnetic field intensity for cast steel
15  mc=H*lc; // mmf in central limb
16  Bo=f/(2*ao); // flux density in each outer limb
17  // for B=1.2, H=1400
18  H=1400; // magnetic field intensity for cast steel
        for given flux density
19  mo=H*lo; // mmf for outer limb
20  // By KVL
21  I=(mg+mo+mc)/N;
22  printf('The exciting current required to establish
        the desired flux is %f A',I);
```

Scilab code Exa 8.6 Calculating the coil current to establish required flux

```
1   clc;
2   N=400; // number of turns in coil
3   ac=20*10^-4; // area of cemntral limb
4   ao=15*10^-4; // area of outer iimb
5   lg=1*10^-3; // length of air gap
6   lc=40*10^-2; // length of central limb
7   lo=60*10^-2; // length of each outer limb
8   f=0.9*10^-3; // required flux
9   uo=4*%pi*10^-7; // free space permeability
10  Bg=f/ao; // air gap flux density
11  mg=(Bg*lg)/uo; // mmf or air gap
12  // for B=0.6,H=575 AT/m from fig A.7
13  H=575; // magnetic flux intensity for given flux
        density
14  mo=H*lo; // mmf of outer limb which contain air gap
15  mt=mo+mg; // combined mmf of air gap and outer limb
16  // this mmf acts across the other outer limb
17  haeb=mt/lo; // magnetic field intensity in outer
        limb which does not contain air gap
```

327

```
18  // for H=1370.77, B=1.19 T from fig A.7
19  Bo=1.19; // flux density for given magnetic field
       intensity
20  faeb=Bo*ao; // flux in outer limb
21  fnet=f+faeb; // net flux through central limb
22  Bc=fnet/ac; // flux density in central limb
23  // from fig A.7
24  H=1900; // magnetic field intensity for given flux
       density
25  mc=H*lc; // mmf in central limb
26  // by KVL in one of the loop
27  I=(mc+mt)/N;
28  printf('Exciting current required to establish the
       given flux is %f A',I)
```

Scilab code Exa 8.7 Finding the exciting current in given coil

```
1  clc;
2  a=30*10^-4; // cross sectional area of ferromagnetic
        core
3  uo=4*%pi*10^-7; // free space permeability
4  ur=4000; // relative permeability for core
5  f=10*10^-3; // flux in central limb
6  n1=200; // number of turns in coil 1
7  m1=5000; // mmf for coil 1
8  n2=100; // number of turns in coil 2
9  lc=0.3; // length of central limb
10  lo=0.6; // length of outer limb
11  lg=1*10^-3; // length of air gap
12  rc=lc/(uo*ur*a); // reluctance for central limb
13  ro=lo/(uo*ur*a); // reluctance for outer limb
14  rg=lg/(uo*a); // reluctance for air gap
15  mc=f*(rc+rg); // mmf in central limb
16  // by KML, flux in outer limb containing coil 1 is
17  f1=(m1-mc)/ro;
```

```
18  // By flux law at node a in fig A.17, flux in outer
        limb contaning coil 2 is
19  f2=f1-f;
20  // by mmf law , mmf in coil 2 is
21  m2=mc-f2*ro;
22  I2=m2/n2; // current in coil 2, upper polarity is
        assumed positive
23  printf('Current in coil 2 is %f A',I2);
24  disp('As the mmf of coil 2 is positive , assumed
        polarity is correct. Therefore terminal A is
        positive because current enters through it and
        terminal B is negative ');
```

Scilab code Exa 8.8 Determining the emf induced in conductor for different angles of conductor with field flux

```
1   clc;
2   l=0.8; // length of conductor
3   B=1.2; // flux density of uniform magnetic field
4   v=30; // speed of conductor
5   disp('case a');
6   // conductor motion is normal to field flux
7   theta=90; // angle between direction of motion and
        field flux
8   e=B*l*v*sin(theta*(%pi/180));
9   printf('EMF induced is %f V\n',e);
10  disp('case b');
11  // conductor motion is at an angle of 30 degrees
        from direction of field
12  theta=30; // angle between direction of motion and
        field flux
13  e=B*l*v*sin(theta*(%pi/180));
14  printf('EMF induced is %f V\n',e);
15  disp('case c');
16  // conductor motion is parllel to field flux
```

```
17  theta=0; // angle between direction of motion and
       field flux
18  e=B*l*v*sin(theta*(%pi/180));
19  printf('EMF induced is %f V\n',e);
```

Scilab code Exa 8.9 Determining the emf induced in square coil for different angles of coil with field flux

```
1  clc;
2  // After deriving the expression
3  a=0.1; // side of square coil
4  N=100; // number of turns
5  n=1000; // speed of rotation on rpm
6  B=1; // flux density of a uniform magnetic field
7  disp('case a');
8  theta=90; // angle of coil with the field
9  w=(2*%pi*n)/60; // angular speed of coil in rad/s
10  e=N*B*a^2*w*cos(theta*(%pi/180));
11  printf('Emf induced in coil is %f V\n',e);
12  disp('case b');
13  theta=30; // angle of coil with the field
14  w=(2*%pi*n)/60; // angular speed of coil in rad/s
15  e=N*B*a^2*w*cos(theta*(%pi/180));
16  printf('Emf induced in coil is %f V\n',e);
17  disp('case c');
18  theta=0; // angle of coil with the field
19  w=(2*%pi*n)/60; // angular speed of coil in rad/s
20  e=N*B*a^2*w*cos(theta*(%pi/180));
21  printf('Emf induced in coil is %f V\n',e);
```

Scilab code Exa 8.10 Determining the emf induced in conductor

```
1  clc;
```

```
2  l =0.5;  // length of conductor lying along Y-axis
3  B=1.2;  // Flux density along the X-axis
4  v=2;  // velocity of conductor
5  //e=Blv; for maximum induced emf all the three
       quantities should be perpendicular to each other
6  e=B*l*v;
7  printf('Maximum induced EMF in conductor is %f V',e)
       ;
```

Scilab code Exa 8.12 Determining the inductance for given circuit by using the given expression

```
1  clc ;
2  disp ('case a');
3  // as per the data taken from Ex 1_3
4  rlg=24.948*10^5;  // air gap reluctance for example 1
       _3(a)
5  rlc=12.474*10^5;  // iron core reluctance for example
       1_3(a)
6  rl=rlg+rlc;  // net reluctance
7  N=500;  // Number of turns
8  L=(N^2/rl)*1000;
9  printf ('Inductance for case a is %f mH\n',L);
10  disp ('case b');
11  // as per the data taken from Ex 1_3 part(c)
12  B=1.254;  // calculated flux density
13  H=3200;  // magnetic field intensity obtained from
       magnetisation curve corresponding to the flux
       density calculated
14  uo=4*%pi*10^-7;  // free space permeability
15  ur=B/(H*uo);  // relative permeability of iron core
16  d=2.85*10^-2;  // diameter of cross section
17  A=(%pi*d^2)/4;  // area of core
18  l=0.5;  // core length
19  rlc=l/(ur*uo*A);  // reluctance of iron core for part
```

331

```
          C
20  rt=rlg+rlc;  // net reluctance
21  L=(N^2/rt)*1000;
22  printf('Inductance for case b is %f mH\n',L);
```

Scilab code Exa 8.13 Determining the inductance of coils for given circuit

```
1  clc;
2  // data taken from Ex A.7, fig A.16
3  N1=200;  // number of turns in coil 1
4  f1=53.97*10^-3;  // flux in outer limb containing
       coil 1
5  m1=5000;  // mmf for coil 1
6  I1=m1/N1;  // current in coil 1
7  N2=100;  // number of turns in coil 2
8  f2=43.97*10^-3;  // flux in outer limb containing
       coil 2
9  m2=1102;  // mmf for coil 2
10  I2=m2/N2;  // current in coil 2
11  L1=(N1*f1)/I1;
12  printf('Inductance for coil 1 is %f H\n',L1);
13  L2=(N2*f2)/I2;
14  printf('Inductance for coil 2 is %f H\n',L2);
```

Chapter 9

Appendix B

Scilab code Exa 9.3 Determining phase and line current power factor total active and reactive power for star and delta connected load

```
1  clc;
2  vl=400; // line voltage
3  z=10+7.5*%i; // load impedance per phase
4  disp('For star connected load');
5  vp=vl/sqrt(3); // phase voltage
6  ip=vp/abs(z);// phase and line current are same in
       the case of star connected load
7  an=atand(-imag(z),real(z));
8  pf=cosd(an);
9  P=sqrt(3)*vl*ip;
10 pa=sqrt(3)*vl*ip*pf;
11 pr=-sqrt(3)*vl*ip*sind(an);
12 printf('Phase and line currents are %f A\n',ip);
13 printf('Power factor is %f lagging \n',pf);
14 printf('Total volt ampere is %f VA\n',P);
15 printf('Total active power is %f W\n',pa);
16 printf('Total reactive power is %f VAr\n',pr);
17 disp('For delta connected load');
18 vp=vl // phase voltage and line voltage are same in
       the case of star connected load
```

```
19  ip=vp/abs(z);
20  il=ip*sqrt(3);
21  an=atand(-imag(z),real(z));
22  pf=cosd(an);
23  P=sqrt(3)*vl*il;
24  pa=sqrt(3)*vl*il*pf;
25  pr=-sqrt(3)*vl*il*sind(an);
26  printf('Phase current is %f A\n',ip);
27  printf('Line current is %f A\n',il);
28  printf('Power factor is %f lagging\n',pf);
29  printf('Total volt ampere is %f VA\n',P);
30  printf('Total active power is %f W\n',pa);
31  printf('Total reactive power is %f VAr\n',pr);
```

Scilab code Exa 9.4 Determining per phase circuit parameters

```
1   clc;
2   il=48; // load current(leading)
3   p=30; // load power in KW
4   vl=500; // line voltage
5   f=50; // supply frequency
6   pf=(p*1000)/(sqrt(3)*vl*il);
7   vp=vl/sqrt(3); // phase voltage
8   zp=vp/il; // magnitude of phase impedance
9   rp=zp*pf;
10  // since current is leading other parameter must be
        a capacitor
11  xc=zp*sqrt(1-pf^2); // reactance
12  c=(10^6)/(2*%pi*f*xc);
13  disp('circuit parameters are');
14  printf('Load resistance is %f ohm\n',rp);
15  printf('Load capacitance is %f micro farad',c);
```

334

Scilab code Exa 9.5 Finding total line current power factor total and reactive power

```
1  clc;
2  zs=10+15*%i; // star connected load per phase
3  zd=12-15*%i; // delta connected load per phase
4  vl=400; // supply line voltage
5  disp('case a');
6  // converting delta connected load to star connected
        load
7  zd=zd/3;
8  vp=vl/sqrt(3);
9  i1=vp/zs; // line current in star connected load
10 i2=vp/zd; // line current in delta connected load
11 i=abs(i1+i2);
12 printf('Total line current is %f A\n',i);
13 an=atand(imag(i1+i2),real(i1+i2));
14 pf=cosd(an);
15 P=(sqrt(3)*vl*i*pf);
16 pr=sqrt(3)*vl*i*sqrt(1-pf^2);
17 printf('Power factor is %f leading\n',pf);
18 printf('Total power is %f W\n',P);
19 printf('Total reactve power is %f VAr',pr);
```

Scilab code Exa 9.6 Determining input power line current power factor and shaft power

```
1  clc;
2  w1=85; // reading of wattmeter 1;
3  w2=35; // reading of wattmeter 2;
4  P=w1+w2; // total input power
5  n=0.85; // efficiency of motor
6  vl=1100; // supply voltage
7  pf=cosd(atand((sqrt(3)*(w1-w2))/(w1+w2)));
8  il=(P*1000)/(sqrt(3)*vl*pf); // line current
```

```
9  ps=n*P;
10 printf('Input power is %f KW\n',P);
11 printf('Line current is %f A\n',il);
12 printf('power factor is %f lagging\n',pf);
13 printf('shaft power is %f KW',ps);
```

Scilab code Exa 9.7 Determining no load losses and no load power factor

```
1  clc;
2  w1=2000; // reading of wattmeter 1 under no load
3  w2=-400; // reading of wattmeter 2 under no load,
      since the connections are reversed that is why
      negative sign
4  theta=atand((sqrt(3)*(w1-w2))/(w1+w2));
5  pl=w1+w2;
6  pf=cosd(theta);
7  printf('No load losses are %f W\n',pl);
8  printf('No load power factor is %f lagging',pf);
```

Scilab code Exa 9.8 Calculating reading of wattmeters and input power to load

```
1  clc;
2  vl=230; // line voltage
3  f=50; // frequency of supply
4  c=100*10^-6; // value of capacitance in each phase
5  vp=230/sqrt(3); // phase voltage
6  zp=1/(2*%pi*f*c); // phase impedance
7  il=vp/zp; // line current
8  // value of cos(theta) is taken from figB.15
9  w1=vl*il*cosd(120);
10 w2=vl*il*cosd(60);
11 printf('Reading of wattmeter 1 is %f W\n',w1);
```

336

```
12  printf('Reading of wattmeter 2 is %f W\n',w2);
13  p=w1+w2;
14  printf('Total input power is %f W',p);
```