# Scilab Textbook Companion for Basic Electrical Engineering by C. L. Wadhwa[1]

Created by
Manoj A
B.Tech Third Year
Electrical Engineering
National Institute of Technology, Karnataka.
College Teacher
Debasisha Jena
Cross-Checked by
Ganesh R

July 4, 2014

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

# Book Description

**Title:** Basic Electrical Engineering

**Author:** C. L. Wadhwa

**Publisher:** New Age International Ltd.

**Edition:** 4

**Year:** 2007

**ISBN:** 978-81-224-2947-3

Scilab numbering policy used in this document and the relation to the above book.

**Exa** Example (Solved example)

**Eqn** Equation (Particular equation of the above book)

**AP** Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

# Contents

# List of Scilab Codes

# List of Figures

# Chapter 1

# AC Circuits

**Scilab code Exa 1.1** To calculate frequency and instantaneous voltage and time of a voltage wave

```
// To calculate frequency, instantaneous voltage and
    time of a voltage wave

clc;
clear;

// The volatage eqaution is v= 0.02 sin (4000t + 30(
   degress)).


Vm=0.02;

deff('a=vol(b)','a=Vm*sind(((4000*b)*(180/%pi))+30)'
   ); // Function for voltage equation

t=320*(10^-6);

w=4000; // angular frequency

// General expression for voltage is given by V=Vm
```

```
    sin ()(2*pi*f*t)+theta)
// Comparing both the eqautions we get 2*pi*f=4000

f=w/(2*%pi);

v=vol(t);

// 360degress is equal to 1/f s.

//Refer the diagram with this code to understand
    better.
// 30degress is

t30=30/(f*360);

disp('Hz',f,'The frequency of the voltage wave =')
disp('V',v,'The instantaneous voltage at t= 320
    micro seconds =')

disp('s',t30,'The time represented by 30 degrees
    phase difference =')
```

**Scilab code Exa 1.2** To find the current and power

```
// To find the current and power

clc;
clear;

Z=10+20*%i;

V=120;

I=V/Z;

```

```
theta=atand(imag(I)/real(I));

P=V*abs(I)*cosd(theta);

disp('A',I,'The current flowing through the given
   impedance =')
disp('watt',P,'The real power delivered to the
   impedance =')
```

**Scilab code Exa 1.3** To find the current and power of a capacitive circuit

```
// To find the current and power of a capacitive
   circuit

clc;
clear;

Z=10-25*%i;

V=15;

I=V/Z;

theta=atand(imag(I)/real(I));

Pr=V*abs(I)*cosd(theta);
Pq=V*abs(I)*sind(theta);

disp('A',I,'The current flowing through the given
   impedance =')
disp('watt',Pr,'The real power loss of the
   capacitive impedance =')
disp('Var',-Pq,'The reactive power of the capacitive
    circuit =')// Negative sign indicates power
   genrated.
```

**Scilab code Exa 1.4** To find the frequency in the RLC Circuit at a phase angle of 45 degrees

```
// To find the frequency in the RLC Circuit at a
   phase angle of 45 degrees

clc;
clear;

R=100;
L=2;
C=20*(10^-6);

f1=poly([0 1],'f1','c');
f2=poly([0 1],'f2','c');
w1=2*%pi*f1;
w2=2*%pi*f2;

//To achieve a phase angle of 45 degrees, the
   difference between Xl and Xc should be equal to R

// We have two different situations

Xl1=L*w1;
Xc1=1/(w1*C);

Xl2=L*w2;
Xc2=1/(w2*C);

cs1=Xl1-Xc1-R;
cs2=Xc2-Xl2-R;

f1=roots(cs1(2));
f2=roots(cs2(2));
```

```

//To find the suitable roots and to differenciate between positive and negative roots.

a=sqrt(f1(1));
b=sqrt(f1(2));
c=sqrt(f2(1));
d=sqrt(f2(2));

if(imag(a))
    f1=f1(2);
end

if(imag(b))
    f1=f1(1);
end

if(imag(c))
    f2=f2(2);
end

if(imag(d))
    f2=f2(1);
end

disp('The frequencies at which the phase angle is 45 degress are')

disp('Hz',f2,'f2 =','Hz',f1,'f1 =')
```

**Scilab code Exa 1.5** To find the inductance of a choke to operate a 120V 500W lamp at 230V

```
// To find the inductance of a choke to operate a 120V 500W lamp at 230V
```

```
clc;
clear;

V=120;
W=500;
Rl=(V^2)/W;
I=W/V; // Circuit Current

// Q of a choke means the ratio of its inductive
   reactance to its resistance

Q=2;
f=60;
w=2*%pi*f;

Vs=230; // Supply Voltage

Xcir=230/I;

L=poly([0 1],'L','c');

Xl=w*L;

Rc=Xl/2; // Q utilised

// total resistance

Rt=Rl+Rc;


ind=(Rt^2)+(Xl^2)-(Xcir^2);// Characteristic
   equation to find L

L=roots(ind);

disp(ind,'The Characteristic equation to find L is')

```

```
if(imag(sqrt(L(1))))
    L=L(2);
else
    L=L(1);
end


disp('H',L,'The inductance of the choke coil = ')
```

**Scilab code Exa 1.6** Determine the value of the circuit components

```
// Determine the value of the circuit components

clc;
clear;

Z=10-(30*%i);

f=1*(10^6);

Y=1/Z;

G=real(Y);
B=imag(Y);

// G= 1/R
// B= wC


w=2*%pi*f;

R=1/G;

C=B/(w);

```

```
cap=abs((10^9)*C)/(10^9);

disp('ohms',R,'The resistance of the circuit =')
disp('nF',cap,'The capacitance of the circuit =')
```

**Scilab code Exa 1.7** To determine circuit impedance and current in a parallel connection of a resistor and capacitor

```
// To determine circuit impedance and current in a
   parallel connection of a resistor and capacitor.

clc;
clear;

R=4700;
V=240;
f=60;
w=2*%pi*f;
C=2*(10^-6);
Xc=-(1/(C*w))*%i;// Reactance in polar form

Ir=V/R;
Ic=V/Xc;

I=Ir+Ic;// Total current

Z=V/I;

theta=atand(imag(Z)/real(Z));

mprintf('Impedance of the circuit = %f /_ %f ohms',
   abs(Z),theta)
```

**Scilab code Exa 1.8** To find the current and impedance and admittance of the circuit

```
//To find the current and impedance and admittance
   of the circuit

clc;
clear;

R=25;
Xl=50*%i;
Z1=R+Xl;

Z2=-40*%i;

V=100;

Zeq=(Z1*Z2)/(Z1+Z2);

Y=1/Zeq;

I=V*Y;

mprintf('The Current of the circuit = %f /_ %f A \n'
   ,abs(I),atand(imag(I)/real(I)))
mprintf('The Impedance of the circuit = %f /_ %f
   ohms \n',abs(Zeq),atand(imag(Zeq)/real(Zeq)))
mprintf('The Admittance of the circuit = %f /_ %f
   siemens \n',abs(Y),atand(imag(Y)/real(Y)))
```

**Scilab code Exa 1.9** Maximum current and frequncy at which it occurs and respective voltages

```
// Maximum current and frequncy at which it occurs
   and respective voltages
```

```

clc;
clear;

R=5;
L=4*(10^-3);
C=0.1*(10^-6);
V=10;

w0=1/(sqrt(L*C));

Ir=V/R;

Vl=w0*L*Ir;

Vc=Ir/(w0*C);

mprintf('The Maximum Current at resonance = %f A \n'
    ,Ir)
mprintf('The frequency for resonance = %f rad/sec \n
    ',w0)
mprintf('The voltage magnitude across the inductor =
     %f V \n',Vl)
mprintf('The voltage magnitude across the capacitor
    = %f V \n',Vc)
```

**Scilab code Exa 1.10** RLC circuit problems on resonace

```
// RLC circuit problems on resonace

clc;
clear;

R=6.28;
L=20*(10^-3);
```

```
f=5*(10^3);

w=2*%pi*f;

C=1/(L*(w^2));

Xc=1/(w*C);
Xl=L*w;

Vc=5;

Z=Xc+R+Xl;

I=Vc/Xc // Total current

V=I*R;

// frequency is inversely proportional to square
   root of capacitance
// So if C is halved; f will increase square root of
    2 times more.

fn=sqrt(2)*f;

Xln=2*%pi*fn*L;

Q=Xln/R;

//Note under resonance conditions Vl and Vc are much
    greater than the supply voltage.

mprintf('i) The value of capacitor = %f micro F \n'
   ,(10^6)*C)
mprintf('ii) The supply voltage = %f V \n',V)
mprintf('iii) The frequency of resonance when C is
   halved = %f Hz \n',fn)
mprintf('     The Q of the new circuit = %f \n',Q)
```

**Scilab code Exa 1.11** RLC circuit problems with quality factor

```
// RLC circuit problems with quality factor

clc;
clear;

C=320*(10^-12);
Q=50;
f0=175*(10^3);
w0=2*%pi*f0;

L=1/(C*(w0^2));

R=w0*L/Q;

Xc=1/(C*w0);
Xl=L*w0;

V=0.85;

I=V/R;

Vc=I*Xc;

bw=f0/Q;

mprintf('The inductance for resonance =%f m H \n'
    ,(10^3)*L);
mprintf('The current flowing in the circuit at
    resonance =%f m A \n',(10^3)*I);
mprintf('The voltage across the capacitor at
    resonance =%f V \n',Vc);
mprintf('The bandwidth of the circuit =%f Hz \n',bw)
```

```
;
```

---

**Scilab code Exa 1.12** RLC circuit problem to find the resonance frequncy and impedance magnitude

```
// RLC circuit problem to find the resonance frequncy and impedance magnitude

clc;
clear;

R=10;
L=100*(10^-3);
C=0.01*(10^-6);

f0=1/((2*%pi)*(sqrt(L*C)));
// New frequencies according to the problem statement
f1=f0-1000;
f2=f0+1000;

w1=2*%pi*f1;
w2=2*%pi*f2;

Xl1=w1*L;
Xc1=1/(w1*C);

Xl2=w2*L;
Xc2=1/(C*w2);

Z1=sqrt((R^2)+((Xl1-Xc1)^2));
Z2=sqrt((R^2)+((Xl2-Xc2)^2));

mprintf('The Impedance magnitude at 1KHz below resonance (Capacitive) =%f ohms \n',Z1)
```

```
mprintf('The Impedance magnitude at 1KHz above
    resonance (Inductive) =%f ohms \n',Z2)
```

**Scilab code Exa 1.13** Series Resonace of a RLC circuit with 2 coils

```
//Series Resonace of a RLC circuit with 2 coils.

clc;
clear;

R1=0.5;
R2=1.5;
R3=0.5;
C1=6*(10^-6);
C2=12*(10^-6);
L1=25*(10^-3);
L2=15*(10^-3);

// Both the coils are connected in series

Req=R1+R2+R3;

Leq=L1+L2;

Ceq=(C1*C2)/(C1+C2);

f=1/((2*%pi)*sqrt(Leq*Ceq));

Q=2*%pi*f*Leq/Req;

Q1=2*%pi*f*L1/(R1+R3);
Q2=2*%pi*f*L2/R2;

mprintf('i) The frequency of resonance = %f Hz or Wo
     = %f rad/sec \n',f,2*%pi*f)
```

```
mprintf('ii) Q of the circuit = %f \n',Q)// The total resistance should be considered, Error in the textbook
mprintf('iii) Q of coil 1 is %f and Q of coil 2 is %f \n',Q1,Q2)
```

**Scilab code Exa 1.14** Series Resonance in RLC circuit to find inductance and power

```
//Series Resonance in RLC circuit to find inductance and power

clc;
clear;

R=5;
C=50*(10^-6);
f0=100;
w=2*%pi*f0;

L=1/(C*(w^2));

V=200;
Xc=1/(C*w);
Xl=L*w;

I=V/R;

P=(I^2)*R;

Vc=I*Xc;

Zc=R+(Xl*%i);

Vz=I*(Zc);
```

```
Q0=Xl/R;

bw=f0/Q0;

printf('a) The inductance of the coil = %f mH \n'
    ,(10^3)*L)
printf('b) The power delivered to the coil = %f kW \
    n',(10^-3)*P)
printf('c) The voltage magnitude across the
    capacitor = %f V \n The voltage magnitude of the
    coil = %f V \n',Vc,abs(Vz)) // Magnitudes with at
     most accuracy
printf('d) The bandwidth of the circuit = %f \n',bw)
```

**Scilab code Exa 1.15** Determine the current in parallel branches and supply current

```
//Determine the current in parallel branches and
    supply current

clc;
clear;

Xl=%i*100; // Inductance
R=10; // Resistance
V=10;
Xco=-%i*100;

Q=abs(Xl)/R;

Z0=Q*abs(Xco);
I0=V/Z0;

Ic=V/Xco;
```

```
Il=V/Xl;

Pi=V*I0; // Power Input

Pc=(I0^2)*10; // Copper Loss

// Frequency reduced to fo/2

Xl1=Xl/2; // New Inductive reactance at half the
    initial frequency
Xco1=Xco*2; // New Capacitative reactance at half
    the initial frequency

Z1=R+Xl1; // Net impedance of the branch containing
    Resistance and inductor

Znet= Z1*Xco1/(Xco1+Z1); // Net Impedance of the
    circuit

I1=V/Znet; // Net Current for reduced frequency

ti1=atand(imag(I1)/real(I1)); // Phase Angle

// Frequncy increased to 2fo

Xl2=2*Xl;// New Inductive reactance at double the
    initial frequency
Xco2=Xco/2;// New Capacitative reactance at double
    the initial frequency
Z2=R+Xl2;// Net impedance of the branch containing
    resistance and inductor

Zt=Z2*Xco2/(Z2+Xco2);// Net Impedance of the circuit

I2=V/Zt; // Net Current

ti2=atand(imag(I2)/real(I2));

```

```
printf('a)  The Current flowing in the inductor =')
disp('mA',Il*1000)
printf('    The current flowing in the capacitor =')
disp('mA',Ic*1000)
printf('    The supply current = %g mA\n \n',I0*1000)
printf('b)  The current for half the intial frequency = %g/_%g mA\n',abs(I1)*1000,ti1)
printf('    The current for double the intial frequency = %g/_%g mA\n',abs(I2)*1000,ti2)
```

**Scilab code Exa 1.16** Determine the original and loaded circuit bandwidth

```
// Determine the original and loaded circuit bandwidth

clc;
clear;

f0=1000*(10^3);
V=5;
Q=50;
Xl=2*(10^3);
Xc0=2*(10^3);
R1=40*(10^3); // Branch near the source
R2=Q*Xl; // Branch with both the inductor and resitance

Z0=Q*Xc0;

bw=f0/Q; // Original Bandwidth

// Considering loading resistance

Reff= R1*R2/(R1+R2);
```

```
Qd=Reff/Xc0;

bw1=f0/Qd; // Bandwidth with loading resistance

printf('The original bandwidth = %g kHz\n',bw/1000)
printf('The loaded circuit bandwidth = %g kHz\n',bw1
   /1000)
```

**Scilab code Exa 1.17** Expression for the sum of energy stored by inductor and capacitor connected in series at resonance

```
//Expression for the sum of energy stored by
   inductor and capacitor connected in series at
   resonance

clc;
clear;

printf(' i= Im*cos(w0)*t\n')
printf(' The energy stored is L*(i^2)/2 = L*(Im^2)*(
   cos(w0*t)^2) \n\n')

printf(' The energy stored in the capacitor (q^2)/2C
    = 1/2C * (Im^2)*[integration of i wrt dt from 0
   to t]^2 \n')
printf('                   = 1/2C * (Im^2) *[integration
   of cos(w0*t) wrt dt from 0 to t]^2\n')
printf('                   = 1/2C * (Im^2) *[(sin(w0*t)/w0
   ) limits 0 to t]^2\n')
printf('                   = (Im^2)/2 * L * (sin(w0*t)^2)\
   n\n')
printf('Therefore total energy = L*(Im^2)/2 * [(cos(
   w0*t)^2)+(sin(w0*t)^2)]\n')
printf('                   = (Im^2)*L/2\n')
printf('                   = L*(I^2)\n')
```

**Scilab code Exa 1.18** Expression for the sum of energy stored by inductor and capacitor connected in parallel at resonance

```
//Expression for the sum of energy stored by
    inductor and capacitor connected in parallel at
    resonance

clc;
clear;

printf('v = Vm * cos(w0*t)\n')
printf('The energy stored by the capacitor = C*(Vm
    ^2)*(cos(w0*t)^2)\n')
printf('The energy stored by the inductor = L*(i^2)
    /2\n\n')
printf('v   = L *(di/dt)\n di = v*dt/L\n (i^2) = [
    integration of (v/L) wrt dt from 0 to t]^2 \n')
printf('    = (Vm^2)/(L^2) * (sin(w0*t)^2)/(w0^2) =
    (Vm/L)^2 * (sin(w0*t)^2) * LC\n')
printf('    = (Vm^2)*C*(sin(w0*t)^2)/L \n\n')
printf(' Energy = L*(i^2)/2 = (Vm^2)*C*(sin(w0*t)^2)
    *L/(L*2)\n')
printf('          = C/2 * (Vm^2)*(sin(w0*t)^2)\n\n')
printf(' Therefore total energy = C*(Vm^2)/2 = C*(V
    ^2)\n')
```

**Scilab code Exa 1.19** To determine bandwidth and half power frequencies

```
// To determine bandwidth and half power frequencies

clc;
```

```
clear;

R=50;
f=750; // Frequency
w0=(2*%pi*f);
L=10*(10^-3);
I=1; // Maximum Current

Q=w0*L/R;

bw=f/Q;

a=(f^2); // let a = f1*f2
b= bw; // let b = f2-f1
c=sqrt((b^2)+(4*a)); // let c = f2+f1

f2=(b+c)/2;
f1=(c-b)/2;

printf('The bandwidth = %g Hz\n',bw)
printf('The half frequencies are f1 = %g Hz and f2 =
    %g Hz\n)',f1,f2)
```

**Scilab code Exa 1.20** Determine the frequency of resonance and Max value of Rc at resonance

```
// Determine the frequency of resonance and Max
   value of Rc at resonance

clc;
clear;

Rl=6;
L=1*(10^-3);
Rc=4;
```

```
C=20*(10^-6);

w0=(1/(sqrt(L*C)))*(sqrt(((Rl^2)-(L/C))/((Rc^2)-(L/C
    ))));

Rcmax=sqrt(L/C);

printf(' The frequency of resonance = %g rad/sec\n',
    w0)
printf(' The Maximum value of Rc = %g ohms \n',Rcmax
    )
```

**Scilab code Exa 1.21** To determine Rl for which resonance can take place

```
// To determine Rl for which resonance can take
    place

clc;
clear;

Rl=poly(0,'Rl');
Rc=5;
Xc=6;
Xl=15;

x=(((Rl^2)+(Xl^2))*Xc)-(((Rc^2)+(Xc^2))*Xl);

Rl=roots(x);

disp(x)

printf('The above eqaution must be eqauted to zero
    to get Rl \n')

disp(Rl)
```

```
printf('The above eqaution leads to imaginary roots
    which is not possible , hence no value of Rl can
    bring resonance in the circuit at the given
    condition \n' )
```

**Scilab code Exa 1.22** To find the resistor for a given Q factor

```
//To find the resistor for a given Q factor

clc;
clear;

f0=600*(10^3);
bw=50*(10^3); // Bandwidth

L=1.3*(10^-3); // Inductance

Q=30;

Xl=2*%pi*f0*L; // Inductive Reactance

Xco=Xl;// At resonance Xl= Xco

Zto=Q*Xco;

Qd=f0/bw; // Required Q for the circuit

Zdto= Qd*Xco; // The equivalent input resistance
    required

Rd=poly(0,'Rd');

x=(Zdto*(Zto+Rd))-(Zto*Rd); // Characteristic
    equation to the shunt resistance
```

```

Rd=roots(x); // Shunt resistance

printf('The resistance that is to be connected across the coil = %g k ohms\n',Rd/1000)
```

**Scilab code Exa 1.23** Find the flux density

```
//Find the flux density

clc;
clear;

l=50*(10^-2); // Mean length
m0=4*%pi*(10^-7); // Constant (Permeablity of air)
ag=1*(10^-3); // Air Gap
mr=300; // Relative permeability
N=200; // No of turns
I=1; // Current
A=poly(0,'A');// Area

Rel=l/(m0*mr*A);//Reluctance of the substance

Relag=ag/(m0*A); // Air gap reluctance

MMF=N*I;

Relt=Rel+Relag; // Total reluctance

phi=MMF/Relt;//Flux

B=phi/A;

// To get the numerical value of the flux density as the polynomial denominator doesn't divide
```

```

x=B(2)-A;
x=roots(x);

y=B(3)-A;
y=roots(y);

B=x/y;

printf('The flux density = %g mWb/(m^2)\n',B*1000)
```

**Scilab code Exa 1.24** Find the number of ampere turns

```
//Find the number of ampere turns

clc;
clear;

l=30*(10^-2); // Length of an iron path
lag=2*(10^-3);//Length of air gap
B=0.8; // Flux density
H=700;
m0=(4*%pi)*(10^-7);
mr=B/(m0*H);

A=poly(0,'A');//Area of the iron path

R1=l/(m0*mr*A);
R2=lag/(m0*A);
R=R1+R2;

phi=B*A;//Flux

NI=phi*R;

```

```
//To find numerical value
y=NI-A;

NI=roots(y(2));

//The answer in the textbook contains Round off
   error
printf('The number of turns necessary to produce a
   flux density of 0.8T in the air gap = %g AT\n',NI
   )
```

**Scilab code Exa 1.25** To find current in the 600 turn exciting coil

```
//To find current in the 600 turn exciting coil

clc;
clear;

N=600;
mr=800;
m0=4*%pi*(10^-7);

phi=100*(10^-6);// Flux in air gap

l1=10*(10^-2);
l2=18*(10^-2);
lg=2*(10^-3); // Air gap length
Ac=(6.25)*(10^-4);// Central limb area
As=3*(10^-4);// Side limb area

Ra=lg/(m0*Ac);
Ri=l1/(mr*m0*Ac);

R=l2/(m0*mr*As);

```

```
Rt=Ra+Ri; // Total reluctacne of the central limb

AT1=Rt*phi; // MMF or Ampere turns for the central
    limb
AT2=R*phi/2; // MMF, Two identical limbs hence flux
    becomes half and only one limb is considered

AT=AT1+AT2;//Total MMF

I=AT/N; // Current in the 600 turns

printf('The current flowing in the 600 turns
    exciting coil = %g A\n',I)
```

**Scilab code Exa 1.26** Find the current required to develop a flux of given mWb

```
// Find the current required to develop a flux of
    1.6 mWb

clc;
clear;

B=1;
H=900;
m0=4*%pi*(10^-7);
mr=B/(m0*H);

//lengths
lg=1*(10^-3);// Air gap
lc=24*(10^-2);// Central Limb
la=60*(10^-2);// Side limbs

//area
A=4*4*(10^-4);
```

```

phi=1.6*(10^-3); // Flux

//Reluctances
Ra=lg/(m0*A); // Air gap
Rc=lc/(m0*mr*A);// Central limb
Rs=la/(m0*mr*A);// One side limbs

//mmf
AT1=(Ra+Rc)*phi; // Central limb
AT2=Rs*phi/2;// One of the side limb
AT=AT1+AT2; // Total

N=680;//Turns

I=AT/N;

printf('The current the current required to produce
    a total flux of 1.6 mWb = %g A\n',I)
```

**Scilab code Exa 1.27** Determine the inductance of individual winding

```
//Determine the inductance of individual winding

clc;
clear;

La=15; // Self inductance of first coil
Lb=16;// Self inductance of second coil
M=-8; // Since the flux from both coils oppose each
    other

L1=La+M;
L2=Lb+M;

```

```
L=L1*L2/(L1+L2);

printf('The inductance of the individual windings
    are %g H and %g H respectively.\n',L1,L2)
printf('The equivalent inductance between the
    terminals is %g H\n',L)
```

**Scilab code Exa 1.28** Determine the inductance of a three coil system

```
//Determine the inductance of a three coil system

clc;
clear;
//Self inductances of A B C
L1=25;
L2=30;
L3=35;

//Mutual inductances of AB BC CA
M12=10;//Flux assist each other
M23=-15;//Flux oppose each other
M31=-10;//Flux oppose each other

La=L1+M12+M31;
Lb=L2+M12+M23;
Lc=L3+M31+M23;

Leq=1/(((La*Lb)+(Lb*Lc)+(Lc*La))/(La*Lb*Lc));

printf('The equivalent inductance for a three coil
    system = %g H\n',Leq)
```

**Scilab code Exa 1.29** To determine the parameters of an alternating current of 50Hz frequency

```
// To determine the parameters of an alternating
   current of 50Hz frequency

clc;
clear;

f=50;
Im=20;
w=2*%pi*f;
t=1/100;
It=10;
Irms=Im/(sqrt(2));
Iav=0;//Full Cycle
t10=asin(It/Im)/w;// time taken to rach 10A

Ih=Im*sin(w*t);// Current at 1/100 sec

printf('i) The general ecpression is i(t) = %g sin
   %gt\n',Im,w)
printf('ii) The instantaneous value at t= 1/100 sec
   is %g A\n',floor(Ih*10)/10)
printf('iii) The time taken to reach 10A for the
   first time = %g s\n',t10)
printf('iv) The average and the rms value of current
    is %g A and %g A respectively\n',Iav,Irms)
```

**Scilab code Exa 1.30** Determine the resultant current for two alternating currents

```
//Determine the resultant current for two
   alternating currents

```

```
clc;
clear;

ti1=30;
ti2=45;

i1=20*(expm(%i*%pi/6));
i2=40*(expm(%i*%pi/4));

i=i1+i2;

ti=atand(imag(i)/real(i));

printf('The resultant current = %g sin(314t + %g(
   degrees)) A\n',abs(i),ti)
printf('          Or %g/_%g A\n',abs(i),ti)
```

**Scilab code Exa 1.31** To determine the sum and difference of two alternating voltage sources

```
//To determine the sum and difference of two
   alternating voltage sources

clc;
clear;

//Phase angles
tv1=0;
tv2=-%pi/6;

//Taking v1 as reference voltage
v1=110*(expm(%i*tv1));
v2=80*(expm(%i*tv2));

Vs=v1+v2;//Sum
```

```
Vd=v1-v2;//Difference

ts=atand(imag(Vs)/real(Vs));
td=atand(imag(Vd)/real(Vd));

printf('i) The sum = %g sin(wt + (%g(degrees))) V\n'
    ,abs(Vs),ts)
printf('i) The difference = %g sin(wt + (%g(degrees)
    )) V\n',abs(Vd),td)
```

**Scilab code Exa 1.32** Determine the parameters of the circuit and power and pf

```
//Determine the parameters of the circuit and power
    and pf

clc;
clear;

tv=30;//Phase angle for voltage
ti=-30;// Phase angle for current
t=tv-ti;//Phase difference between v and i

//Keeping i as reference, Voltage leads current by t
     angle

i=10*(expm(%i*0));
v=230*(expm(%i*(%pi*t/180)));

z=v/i;

R=real(z);
X=imag(z);

P=abs(v)*abs(i)*cosd(t)/2; // rms values of voltage
```

```
   and current

printf('The circuit parameters are \n')
printf('Resistance = %g ohms\nReactance = %g ohms\n\
   n',R,X)
printf('The Power consumed = %g W\n',P)
printf('p.f of the circuit = %g\n',cosd(t))
```

**Scilab code Exa 1.33** Determine circuit paramters of an iron coil

```
//Determine circuit paramters of an iron coil

clc;
clear;

f=50; // Frequency
Vdc=40;//DC Voltage source
Idc=5; // Current drawn from DC Voltage source
Vac=200; // AC Voltage Source
Iac=5; // Current drawn form AC Voltage source
P=500; // Power Consumed by ac supply

R=Vdc/Idc; // Resistance of the coil
Z=Vac/Iac; // Impedance of the coil

Pc=(Iac^2)*R; //Power loss in ohmic resistance

Pil=P-Pc; // Iron loss

Reff=P/(Iac^2); // Effective resistance of the coil

Xl=sqrt((Z^2)-(Reff^2)); // Reactance of the coil

L=Xl/(2*%pi*f); // Inductance of the coil

```

```
pf=P/(Vac*Iac);

printf('i) The impedance = %g ohms\n',Z)
printf('ii) The iron loss = %g W\n',Pil)
printf('iii) The inductance of the coil = %g H\n',L)
printf('iv) p.f of the coil = %g\n',pf)
```

**Scilab code Exa 1.34** Determine the phase angle between 220V main and the current

```
//Determine the phase angle between 220V main and
    the current

clc;
clear;

f=50;//Frequency of AC Mains
Vni=100;//Voltage for non inductive load
Ini=10; // Current drawn by a non inductive load
Rni=Vni/Ini;// Resistance of an non inductive load
Vac=220;// Supply from AC Mains

Z=Vac/Ini;

X=sqrt((Z^2)-(Rni^2));

phi=atand(X/Rni);// Phase Angle

L=X/(2*%pi*f);//Inductance

printf('The inductance of a reactor to be connected
    in series = %g H\n',L)
printf('The phase angle between the 220V mains and
    the current = %g degrees\n',phi)
```

**Scilab code Exa 1.35** To determine the coil parameters with resistance of 5 ohms

```
//To determine the coil parameters with resistance
   of 5 ohms

clc;
clear;
//Parameters of the coil
R=5; // Resistance
I=10; // Current flowing
V=200;// Voltage across
f=50;// Frequency of operation
P=750;//Total Power Dissipated

Pc=(I^2)*R; // Copper Loss
Pil=P-Pc;// Iron Loss

Z=V/I;// Impedance

X=sqrt((Z^2)-(R^2));//Reactance

L=X/(2*%pi*f);// Inductance

pf=P/(V*I); // Power Factor

printf('i The iron loss = %g W\n',Pil)
printf('ii) The inductance at the given value of
   current = %g H\n',L)
printf('iii) p.f = %g\n',pf)
```

**Scilab code Exa 1.36** Determine the resonant frequency and the source current

```
//Determine the resonant frequency and the source
    current

clc;
clear;

L=0.24; // Inductance
Rl=150;
Rc=100;
C=3*(10^-6); // Capacitance
f=50;
w=2*%pi*f;
V=200; // Source voltage

fs=1/(2*%pi*sqrt(L*C)); // Frequency at the time of
    series connection

f0=fs*sqrt(((Rl^2)-(L/C))/((Rc^2)-(L/C)))// Resonant
     frequency

//Taking voltage as the reference

Xl=L*w; // Inductive reactance
Xc=1/(C*w); // Capacitive reactance

Ra=Rl+(%i*Xl); // Effective resistance of inductive
    branch
Rb=Rc-(%i*Xc);// Effective resistance of capacitive
    branch

Reff=Ra*Rb/(Ra+Rb); // Effective Resistance
Tr=atand(imag(Reff)/real(Reff)); // Phase Angle

I=V/Reff;// Source current
Ti=atand(imag(I)/real(I))// Phase angle
```

```
printf('The resonant frequency = %g Hz\n',f0)
printf('The source current = %g/_%g A\n',abs(I),Ti)
printf('The input impeadance = %g/_%g ohms\n',abs(Reff),Tr)
```

**Scilab code Exa 1.37** Determine Circuit parameters for a circuit with a current source

```
//Determine Circuit parameters for a circuit with a current source

clc;
clear;

I=2.5*(10^-3); // Current Source
R=100; // Resistance of the coil
L=1*(10^-3); // Inductance of the coil
fr=600*(10^3); // Resonance frequency
R2=60*(10^3); // Resistance in parallel with the coil
wr=2*%pi*fr;// Angular Resonance frequency

Q=wr*L/R; // For the coil

C=1/((wr^2)*L); // Capcitance in the circuit
Xc0=1/(wr*C);

Recc=Q*Xc0; // Equivalent resistance of coil and capacitor

Req=R*Recc/(R+Recc); //Equivalent resistance of the circuit

Qcir=wr*Req*C; // For the circuit
```

```

bw=fr/Qcir; // Bandwidth of the circuit

v=I*Req;

MaxE=C*(v^2); // Maximum energy stored by the capacitor

Pdr=(I^2)*Req; // Power dissipated in the resistor


// Textbook calculation for Equivalent resistance of C and Coil is wrong

printf('a) Q of the coil = %g\n',Q)
printf('b) Capacitance C = %g pF\n',C*(10^12))
printf('c) Q of the circuit = %g\n',Qcir)
printf('d) Bandwidth of the circuit = %g kHz\n',bw/1000)
printf('e) Maximum Energy stored in the capacitor = %g pJ\n',MaxE*(10^12))
printf('f) Power Dissipated in the resistor = %g mW\n',Pdr*1000)
```

**Scilab code Exa 1.38** Determine the instantaneous energy stored in the capacitor and inductor

```
// Determine the instantaneous energy stored in the capacitor and inductor

clc;
clear;

Vc=20*sqrt(2);
C=2; // Capacitor
```

```
L=1; // Inductor
w=poly(0,'w');

// Impedances in order from let to right (as a function of w)
R1=%i*w;
R2=1/(2*%i*w); // Top
R3=1;// Bottom

Rp=R2*R3/(R2+R3); // Effective resistance of the parallel path

Reff=R1+Rp; // Effective resistance

Reff(2)=Reff(2)*conj(Reff(3));
Reff(3)=Reff(3)*conj(Reff(3));

R=imag(Reff(2))/Reff(3); // Imaginary part of the above equation

//From the above equation we get five roots, three are zero and we take the positive value
w=roots(R(2));
w=abs(w(2)); // Numerical Value

// Impedances in order from let to right (Numerical Value)
R1=%i*w;
R2=1/(2*%i*w); // Top
R3=1;// Bottom

Vcrms=Vc/sqrt(2);

// Taking Vc as reference

Ic=Vcrms/R2; // Current through Capacitor
Ir=Vcrms/R3; // Current through Resistor

```

```
Il=Ic+Ir; // Rms value of Current through Inductor
tl=atand(imag(Il)/real(Il)); // Phase angle of
    Inductor Current

Ilmax=abs(Il)*sqrt(2); // Maximum Current

Eins=C*(Vc^2)/2; // Magnitude of Instaneous energy
    stored

Ein=L*(abs(Ilmax)^2)/2; // Energy through the
    inductor
Er=(Ir^2)*R3; // Loss in the resistor

Q0= w*Ein*(1+(1/sqrt(2)))/Er; // Q of the circuit

printf('a) Instaneous Energy Stored in Capacitor =
    %g (sin(%gt)^2) \n',Eins,w)
printf('    Instaneous Energy Stored in Inductor =
    %g (sin( %gt + %g)^2) \n',Ein,w,tl)
printf('b) Q of the circuit = %g \n',Q0)
```

**Scilab code Exa 1.39** To determine the current through all the branches of the given network

```
//To determine the current through all the branches
    of the given network

clc;
clear;

L=1;
R=1*(10^3);
C=400*(10^-6);
i=2; // 2 cos 50t

```

```
w0=1/(sqrt(L*C));

v=i*R; // Voltage across the source

Xl=%i*w0*L; // Inductive reactance
Xc=-%i/(C*w0);// Capacitive reactance

Il=v/Xl; // Inductor current
Ic=v/Xc; // Capacitor Current

//Condition to check if angle is 90
if(real(Il)==0)
    if(imag(Il)>0)
    tl=atand(%inf);
    else
    tl=-1*atand(%inf);
    end

else
    tl=atand(imag(Il)/real(Il));
end

//Condition to check if angle is 90
if(real(Ic)==0)
    if(imag(Ic)>0)
    tc=atand(%inf);
    else
    tc=-1*atand(%inf);
    end

else
    tc=atand(imag(Ic)/real(Ic));
end

printf('The Current through the resistor is %g cos(%g)t A\n',i,w0)
printf('The Current through the inductor is %g cos(%gt + (%g)) A\n',abs(Il),w0,tl)
```

```
printf('The Current through the capacitor is %g cos(
    %gt + (%g)) A\n',abs(Ic),w0,tc)
```

**Scilab code Exa 1.40** To determine parameters to operate the relay

```
//To determine parameters to operate the relay

clc;
clear;

N=500;// No of turns
l=400*(10^-3); // Mean Core length
lg=1*(10^-3); // Air Gap length
B=0.8; // Flux density required to operate the relay
m0=4*%pi*(10^-7); // permeability of free space
H=500; // Magentic Field Intensity

mmfc=H*l; // mmf for the core
mmfg=2*lg*B/(m0); // mmf of the air gap

tmmf=mmfc+mmfg; // Total mmf

Iop=tmmf/N; // Operating current for the relay

m=B/H; // Permeability

mr=m/m0; // Relative permeability

// When gap is zero

mmf=mmfc; // Total mmf required

I=mmf/N; // Current when the gap is zero

printf('i) The current required to operate the relay
```

```
    = %g A\n',Iop)
printf('ii)Wrt the Core, The permeability = %g and
   Relative permeablity = %g \n',m,mr)
printf('iii) The current required when gap is zero =
    %g A \n',I)
```

**Scilab code Exa 1.41** To determine the parameters of a toroid

```
// To determine the parameters of a toroid

clc;
clear;

//Arc length of different materials
lni=0.3; // nickel iron alloy
lss=0.2;//Silicon Steel
lcs=0.1;//Cast steel

A=1*(10^-3); // Area of cross section of Toroid

N=100; // Hundred Turns

phi=6*(10^-4); // Flux
B=0.6; // Flux densities
m0=4*%pi*(10^-7);

//Field intensities
Hni=10; // Ni alloy
Hss=77;// Si Steel
Hcs=270; // CAst steel

mmf=(Hni*lni)+(Hss*lss)+(Hcs*lcs); // Total mmf

B=phi/A;

```

```
I=mmf/N;

//P = Permeability , RP = Relative permeablity

//For Nickel Alloy
mni=B/Hni; // P
mrni=mni/m0; // RP

//For Si Steel
mss=B/Hss; //P
mrss=mss/m0; // RP

//For Cast Steel
mcs=B/Hcs; //P
mrcs=mcs/m0; // RP

deff('x=rel(y,z)','x=(y*z)/phi'); // Fucntion to
    find out Reluctance

//Reluctances
Rni=rel(Hni,lni);
Rss=rel(Hss,lss);
Rcs=rel(Hcs,lcs);

printf('i) The mmf required to establish a magnetic
    flux of 6 *(10^-4) Wb = %g AT\n',mmf)
printf('ii) The Current through the coil = %g A\n',I
    )
printf('iii) Relative permeability and reluctance of
     each ferro magnetic material\n')
printf('    Nickel Iron alloy is %g and %g ohms \n',
    mrni,Rni)
printf('    Silicon Steel is %g and %g ohms \n',mrss
    ,Rss)
printf('    Cast Steel is %g and %g ohms \n',mrcs,
    Rcs)
```

**Scilab code Exa 1.42** Determine the magnetic flux for a toriod

```
//Determine the magnetic flux for a toriod

clc;
clear;

//Arc length of different materials
lni=0.3; // nickel iron alloy
lss=0.2;//Silicon Steel
lcs=0.1;//Cast steel

//Field intensities
Hni=10; // Ni alloy
Hss=77;// Si Steel
Hcs=270; // Cast steel

con=6*(10^-4); // Gives Constant reluctances

deff('x=rel(y,z)','x=(y*z)/con'); // Fucntion to
    find out Reluctance

//Reluctances
Rni=rel(Hni,lni); // Note that the textbook has a
    wrong value calculated for the nickel alloy
    reluctance
Rss=rel(Hss,lss);
Rcs=rel(Hcs,lcs);

mmf=35; // Applied mmf

phi=mmf/(Rni+Rss+Rcs); // Magnetic flux produced

printf('The approximate magnetic flux produced = %g
```

```
Wb\n',phi)
```

---

**Scilab code Exa 1.43** To determine the magnetic parameters of a steel ring

```
//To determine the magnetic parameters of a steel
   ring

clc;
clear;

d=30*(10^-2); // Mean diameter
m0=4*%pi*(10^-7);
lg=1*(10^-3); //Air Gap length
r=(2*(10^-2))/2; // radius of the circular part of
   the air gap
Ag=%pi*r*r; // Area of the air gap
N=600; // No of turns
I=2.5;// Current

Ip=40/100; // Iron path usage
Agp=1-Ip; // Air gap usage

mmf=N*I;
mmfs=Ip*mmf; // mmf required for the steel
mmfg=Agp*mmf; // mmf required for the air gap

Rg=lg/(m0*Ag); // Reluctance of the air gap

phi=mmfg/Rg; // Flux through the gap

Rs=mmfs/phi; // Reluctance of steel
Rt=Rs+Rg; // Total reluctance
B=phi/Ag; // Flux density of air

```

```
printf('i) mmf of air gap and steel ring = %gAT and
    %gAT\n',mmfg,mmfs)
printf('ii) Magnetic flux = %g mWb\n',phi*1000)
printf('iii) Reluctance = %g M ohms \n',Rt/(10^6))
printf('iv) Flux density = %g T\n',B)
```

# Chapter 2

# Network Theory

**Scilab code Exa 2.1** To determine the current using loop analysis

```
// To determine the current using loop analysis

clc;
clear;

// MESH Equations
//6*i1 -2*i2=30
//-2*i1+6*i2=-40

R=[6 -2;-2 6];
E=[30;-40];

//The Loop Currents

I=inv(R)*E; // Matrix Method to solve for two
    unknowns in two eaquations.

i1=I(1);
i2=I(2);
i3=i1-i2;

```

```
disp('A',i2,'i2 =','A',i1,'i1 =','The Calculated Loop Currents are')

disp('The Negative sign indicates that the assumed direction of flow of current should be reveresed.')

if(i1<0);
    i1=abs(i1);
end

if(i2<0);
        i2=abs(i2);
end

disp('A',i1,'The Current through 4 ohm resistor on the 30V side =')
disp('A',i2,'The Current through 4 ohm resistor on the 40V side =')
disp('A',i3,'The Current through 2 ohm resistor =')
```

**Scilab code Exa 2.2** To calculate current in each branch using loop analysis

```
// To calculate current in each branch using loop analysis.

clc;
clear;

// MESH Equations for the given network.
//3*i1-i2+0*i3=11
//-i1+10*i2-2*i3=0
//0*i1+-2*i2+5*i3=13

```

```
//Voltage supplies are 11V and 13V

R=[3 -1 0;-1 10 -2; 0 -2 5];
E=[11;0;13];

// Loop Currents

I=inv(R)*E;

i1=I(1);
i2=I(2);
i3=I(3);

ia=i1-i2; // Assumed direction from Mesh 1
ib=i2-i3; // Assumed direction from Mesh 2

disp('A',ib,'ib (through 2 resistor between 7 ohm and 3 ohm resistor) =','A',ia,'ia(through 1 ohm resistor) =','A',i3,'i3 =','A',i2,'i2 =','A',i1,'i1 =','The Calculated Loop Currents are')

disp('The Negative sign indicates that the assumed direction of flow of current should be reveresed')

// To obtain the magnitude of direction.

if(i1<0)
    i1=abs(i1);
end
if(i2<0)
    i2=abs(i2);
end
if(i3<0)
    i3=abs(i3);
end
if(ia<0)
    ia=abs(ia);
```

```
end
if(ib<0)
    ib=abs(ib);
end

disp('A',i1,'The Current through 2 ohm resistor on the 11V side =')
disp('A',i2,'The Current through 7 ohm resistor =')
disp('A',i3,'The Current through 3 ohm resistor on the 13V side =')
disp('A',ia,'The Current through 1 ohm resistor =')
disp('A',ib,'The Current through 2 ohm resistor between the 7 and 3 ohm resistors =')
```

**Scilab code Exa 2.3** To calculate current in each branch using loop analysis and point voltages in a given network

```
// To calculate current in each branch using loop analysis and point voltages in a given network.

clc;
clear;

// MESH Equations for the given network.
//3.95*i1 -3.75*i2+0*i3=120
//-3.75*i1 +9.5*i2 -5.45*i3=0
//0*i1 -5.45*i2 +5.55*i3=-110

// Positive of 120V DC supply connected to 0.2 ohm resistor
// Positive of 110 DC supply connected to 0.1 ohm resistor

//Voltage supplies are 120V and 110V

```

```
R=[3.95 -3.75 0;-3.75 9.5 -5.45; 0 -5.45 5.55];
E=[120;0;-110];

R1=abs(R(2)); // Resistor carrying ia
R2=abs(R(8)); // Resistor carrying ib

// Loop Currents

I=inv(R)*E;

i1=I(1);
i2=I(2);
i3=I(3);

ia=i1-i2; // Assumed direction from Mesh 1
ib=i2-i3; // Assumed direction from Mesh 2

// Using Nodal Analysis to find V1 and V2.
V1=R1*ia;
V2=R2*ib;

disp('A',ib,'ib (through 2 resistor between 7 ohm
   and 3 ohm resistor) =','A',ia,'ia(through 1 ohm
   resistor) =','A',i3,'i3 =','A',i2,'i2 =','A',i1,'
   i1 =','The Calculated Loop Currents are')

disp('The Negative sign indicates that the assumed
   direction of flow of current should be reveresed'
   )

// To obtain the magnitude of direction.

if(i1<0)
    i1=abs(i1);
end
if(i2<0)
    i2=abs(i2);
end
```

```
if(i3<0)
    i3=abs(i3);
end
if(ia<0)
    ia=abs(ia);
end
if(ib<0)
    ib=abs(ib);
end

disp('A',i1,'The Current through 0.2 ohm resistor on
    the 120V side =')
disp('A',i2,'The Current through 0.3 ohm resistor ='
    )
disp('A',i3,'The Current through 0.1 ohm resistor on
    the 110V side =')
disp('A',ia,'The Current through 3.75 ohm resistor =
    ')
disp('A',ib,'The Current through 5.45 ohm resistor =
    ')

disp('V',V1,'The voltage V1 =')
disp('V',V2,'The voltage V2 =')
```

**Scilab code Exa 2.4** To calculate current from a battery and pd across points A and B

```
// To calculate current from a battery and pd across
    points A and B

clc;
clear;

// Resistances in the given network
R1=4;
```

```
R2=2;
R3=3;
R4=6;
R5=8;

// MESH Equations
//9*i1 -5*i2=10
//-5*i1+19*i2=0

// Supply voltage 10V

R=[(R1+R2+R3) -(R2+R3); -(R2+R3) (R2+R3+R4+R5)];
V=[10;0];

//Loop Currents
I=inv(R)*V;

i1=I(1);
i2=I(2);

i3=i1-i2; // From Mesh 1

// Point Voltages
Va=i3*R3;
Vb=i2*R5;

disp('amperes',abs(i1),'The current through 4 ohm resistor and the battery =')
disp('amperes',abs(i2),'The current through 6 ohm and 8 ohm resistors =')
disp('amperes',abs(i3),'The current through 2 ohm and 3 ohm resistors =')

disp('volts',abs(Va),'The voltage at point A =')
disp('volts',abs(Vb),'The voltage at point B =')
disp('volts',(Va-Vb),'The voltage across Points A and B =')
```

**Scilab code Exa 2.5** Determine Current through branch AB of the given network

```
// Determine Current through branch AB of the given
   network

clc;
clear;

// MESH Equations
// 4*i1 -2*i2 +0*i3 =10
// -2*i1 +6*i2 -2*i3 =0
//0*i1 -2*i2 +6*i3 =0

//Supply Voltage is 10V (Note printing mistake)

R=[4 -2 0;-2 6 -2; 0 -2 6];
V=[10;0;0];

// Loop Currents

I=inv(R)*V;

i1=I(1);
i2=I(2);
i3=I(3);

disp('amperes',abs(i2),'The current through branch
   AB of the network =')
```

**Scilab code Exa 2.6** Determine the current in the branches of the network using nodal analysis

```
// Determine the current in the branches of the network using nodal analysis

clc;
clear;

// Supply voltages
V1=30;
V2=40;

// Resistances in the network
R1=4;
R2=2;
R3=4;

Vb=poly([0 1],'Vb','c');

AD=(V1-Vb)/R1;
BD=(V2-Vb)/R3;
CD=Vb/R2;

X=AD+BD-CD;

disp('The Characterictic Equation to find Vb is')

disp(CD,'=',AD,'        +',BD)

Vb=roots(X);// Stores the numerical value of Vb

i1=(V1-Vb)/R1;
i2=(V2-Vb)/R3;
i3=Vb/R2;

disp('amperes',i1,'Current through 4 ohm resistor on the 30V supply side =')
disp('amperes',i2,'Current through 4 ohm resistor on the 40V supply side =')
disp('amperes',i3,'Current through 2 ohm resistor ='
```

```
)
```

---

**Scilab code Exa 2.7** To Calculate current in all branches of the network shown using nodal analysis

```
// To Calculate current in all branches of the
   network shown using nodal analysis

clc;
clear;

// Nodal Equations
//13*Va-4*Vb=300
//-Va+4*Vb=120

X=[13 -4;-1 4];
V=[300;120];

E=inv(X)*V;

Va=E(1);
Vb=E(2);

i1=(100-Va)/20;
i2=(Va-Vb)/15;
i3=(Va/10);
i4=(Vb/10);
i5=(80-Vb)/10;

disp('V',Vb,'Voltage Vb =','V',Va,'Voltage Va =')

disp('The Branch Currents as calculated are')
disp(i5,'i5',i4,'i4',i3,'i3',i2,'i2',i1,'i1')
disp('amperes respectively')

```

```
disp('The Negative sign indicates that the assumed
    direction of flow of current must be reveresed')

disp('amperes',abs(i1),'The Current through 20 ohm
    resistor on the 100V side =')
disp('amperes',abs(i2),'The Current through 15 ohm
    resistor =')
disp('amperes',abs(i3),'The Current through 10 ohm
    resistor (AE) =')
disp('amperes',abs(i4),'The Current through 10 ohm
    resistor (BE) =')
disp('amperes',abs(i5),'The Current through 10 ohm
    resistor on the 80V side =')
```

**Scilab code Exa 2.8** Conversion to current source and nodal analysis

```
// Conversion to current source and nodal analysis

clc;
clear;

// Nodal Equations
// 1.5*Va-0.5*Vb+0*Vc=5
// 0.5*Va-1.5*Vb+0.5*Vc=0
// 0*Va-0.5*Vb+1*Vc=0

Y=[1.5 -0.5 0;0.5 -1.5 0.5; 0 -0.5 1]; // Admittance
    matrix
I=[5;0;0];
V=inv(Y)*I;

Va=V(1);
Vb=V(2);
Vc=V(3);

```

```
Vab=Va-Vb;

disp('V',Va,'Voltage at node A =')
disp('V',Vb,'Voltage at node B =')
disp('V',Vc,'Voltage at node C =')

disp('V',Vab,'The voltage across AB in the circuit =
    ')
disp('A',Vab/2,'The current in branch AB in the
    circuit =')
```

**Scilab code Exa 2.9** Superposition Principle to determine current in branch

```
// Superposition Principle to determine current in
    branch

clc;
clear;

//Order of resistances from left to right in the
    circuit
r1=2;
r2=12;
r3=1;
r4=2;
r5=3;

V1=2;
V2=4;

// We now short circuit by removing one source and
    consider the rest of the cicuit
// Hence we will have two cases

//Case 1 4V supply removed and shorted
```

```

//Resistances between respective nodes.
rab1=(r4*r5)/(r4+r5);
rac1=rab1+r3;
rcd1=(rac1*r2)/(rac1+r2);

Reff1=rcd1+r1;// Effective resistance in case 1

I1=V1/Reff1; // Current from the 2V source

Iac1=I1*(r2/(r2+rac1));
Iab1=Iac1*(r5/(r5+r4)); //Current in AB from 2V source

//Case 2 2V supply removed and shorted

//Resistances between respective nodes.
rcd2=(r1*r2)/(r1+r2);
rac2=rcd2+r3;
rab2=(rac2*r4)/(rac2+r4);

Reff2=rab2+r5;//Effective resistance in case 2

I2=V2/Reff2; // Current from the 4V source

Iab2=I2*(rac2/(rac2+r4)); //Current in AB from 4V source

Iab=Iab1+Iab2;// Combined Current in AB from both the sources

disp('amperes',Iab,'The Current through AB (2 ohm resistor) =')
```

**Scilab code Exa 2.10** Using thevenin theorem determine current through 2 ohm resistor

```
//Using thevenin theorem determine current through 2
    ohm resistor

clc;
clear;

// Characteristic equation to find Vth
// 14i1+12i2 = 2
// 12i1+16i2 = 4

// Resistors in the circuit in order from the 2V
   side
R1=2;
R2=12;
R3=1;
R4=3;

// Voltage Sources
V1=2;
V2=4;

Z=[14 12; 12 16]; // Resistance Matrix
V=[V1; V2]; // Voltage Matrix

I=inv(Z)*V; // Current Matrix

i1=I(1);
i2=I(2);

Vth=V2-(i2*R4);

Reff= R3 +((R1*R2)/(R1+R2));

Zth= Reff*R4/(Reff+R4);

```

```
Zl=2; // Resistor Connected between AB

Current = Vth/(Zth+Zl); // Current Through 2 ohm
    resistor

printf('The Current through 2 ohm resistor connected
     across AB = %g A\n',Current)
```

**Scilab code Exa 2.11** To find the current through the branch AB

```
//To find the current through the branch AB

clc;
clear;

Zl=2; // Resistor Across AB

// Voltage Sources
V1=20;
V2=10;

//Resistances in order as seen from 20 V side
    exculding the resistance between A and B
R1=2;
R2=2;
R3=2;
R4=4;
R5=4;

// Characteristic Equation
//10i1-4i2 =10
//-4i1+8i2 =10

Z=[10 -4; -4 8]; // Resistance Matrix
V=[10;10]; // Effective Voltages Matrix
```

```
I=inv(Z)*V; // Current Matrix
i1=I(1);
i2=I(2);

Vth=V1-(i1*(R1+R2));

Reff=R4*R5/(R4+R5); // Effective resistance of R4
    and R5 (Parallel)
Rt1=Reff+R3; // Effective Resistance on right side
    of AB
Rt2=R1+R2; // Effective Resistance on left side of
    AB
Zth=Rt1*Rt2/(Rt1+Rt2);

Current= Vth/(Zl+Zth); // Current Through branch AB

printf('The Current through branch AB = %g A\n',
    Current)
```

**Scilab code Exa 2.12** Determine current through various values of RL

```
// Determine current through various values of RL

clc;
clear;

Rl=[0 2 5]; // Resistance Vector

//Resistances of the circuit in order from the 40V
    side.
R1=4;
R2=6;
R3=5;

//Voltage Source
```

```
V=40;

i=V/(R1+R2);

Vth=i*R2;

Rth=R3+(R1*R2/(R1+R2));

Cur=(Rth+Rl)/Vth; // Reciprocal of current

printf('The Current through the Resistance RL = %g,
    %g, %g ohm is %g A, %g A, %g A respectively \n',
   Rl(1),Rl(2),Rl(3),1/Cur(1),1/Cur(2),1/Cur(3))
```

**Scilab code Exa 2.13** Current through AB using Nortons theorem

```
//Current through AB using Nortons theorem

clc;
clear;

// Resitances in order from the 2V side
R1=2;
R2=12;
R3=1;
R4=3;

// Voltage Sources
V1=2;
V2=4;

//Using Superposition principle
Iab1=V2/R4;
I1=V1/(R1+(R2*R3/(R2+R3))); // Current drawn from 2V
    supply
```

```
Iab2=I1*R2/(R1+R2);

Iab=Iab1+Iab2; // Current source

Reff= R3 +((R1*R2)/(R1+R2));

Zth= Reff*R4/(Reff+R4);

Zl=2; // Resistor Connected between AB

Current=Iab*(Zth/(Zth+2)); // Current through branch AB

// Errorless Calculation , In the textbook approximations are done
printf('The Current through 2 ohm resistor in branch AB = %g A\n',Current)
```

**Scilab code Exa 2.14** To determine current in RL using nortons theorem

```
//To determine current in RL using nortons theorem

clc;
clear;

//Resistances of the circuit in order from the 40V side.
R1=4;
R2=6;
R3=5;

//Voltage Source
V=40;

I=V/(R1+(R2*R3/(R2+R3)));
```

```
Ieq=I*(R2/(R2+R3));

Rth=R3+(R1*R2/(R1+R2));

Rl=[0 2 5];

Req=Rth+Rl; // Sum of resistances of Rth and each of
    Rl

//Currents for different values of Rl
I0=Ieq*Rth/Req(1);
I2=Ieq*Rth/Req(2);
I5=Ieq*Rth/Req(3);

printf('The Current through the resistance RL = %g
   ohms is %g A\n',Rl(1),I0)
printf('The Current through the resistance RL = %g
   ohms is %g A\n',Rl(2),I2)
printf('The Current through the resistance RL = %g
   ohms is %g A\n',Rl(3),I5)
```

**Scilab code Exa 2.15** To find current across 2ohm resistor using nortons theorem

```
// To find current across 2ohm resistor using
   nortons theorem

clc;
clear;

Zl=2; // Resistor Across AB

// Voltage Sources
V1=20;
```

```
V2=10;

//Resistances in order as seen from 20 V side exculding the resistance between A and B
R1=2;
R2=2;
R3=2;
R4=4;
R5=4;

Reff=R4*R5/(R4+R5); // Effective resistance of R4 and R5 (Parallel)
Rt1=Reff+R3; // Effective Resistance on right side of AB
Rt2=R1+R2; // Effective Resistance on left side of AB
Zth=Rt1*Rt2/(Rt1+Rt2);

// Using superpostion theorem
Iab1=V1/(R1+R2); // Current supplied to AB from 20V source
I1=V2/(R4+(R3*R5/(R3+R5)));// Current supplied from 10V source to the network
Iab2=I1*(R5/(R3+R5)); // Current supplied to AB from 10V Source

Iab=Iab1+Iab2; // Current Source

I=Iab*(Zth/(Zth+Zl));

printf('The current through branch AB flowing in the 2 ohm resistor = %g A\n',I)
```

**Scilab code Exa 2.16** To determine the current in the 2 ohm resistor using superposition theorem

```
// To determine the current in the 2 ohm resistor
   using superposition theorem

clc;
clear;

// Voltage Sources
V1=5;
V2=10;

// Since both Voltage sources are connected in
   parallel and are unequal
R1=%inf; // As seen by 5V Source
R2=%inf; // As seen by 10V Source

I1=V1/R1; // Current Drawn from 5V supply
I2=V2/R2; // Current Drawn from 10V supply

I=I1+I2; // Current through 2 ohms resistor

printf('The Current flowing in the 2 ohm resistor =
   %g A\n',I)
```

**Scilab code Exa 2.17** To determine the value of RL for Max power transfer

```
//To determine the value of RL for Max power
   transfer

clc;
clear;

Vs=10/sqrt(2); // RMS Value of Voltage

//Resistances of the circuit from the Source side
```

```
R1=10;
R2=15;
R3=20;
R4=5;
R5=10;

Ref1=R3+(R1*R2/(R1+R2));
Ref2=R5+(Ref1*R4/(Ref1+R4));

Rab=Ref2;

// Characteristic Loop Equation of the first two
   loops for current flowing in clockwise direction
//25i1 -15i2 = 10/(2^0.5)
//-15i1+40i2 = 0

Z=[10 25;-15 40];
V=[Vs;0];
I=inv(Z)*V;
i1=I(1);
i2=I(2);

Vth=i2*R4;

Powtrns=(Vth^2)/(4*Rab);

printf('The value of resistance RL for maximum power
    transfer = %g ohms\n',Rab)
printf('The value of power transfered = %g mW\n',
   Powtrns*1000)
```

**Scilab code Exa 2.18** Star to delta conversion of a cicuit

```
//Star to delta conversion of a cicuit

```

```
clc;
clear;

Zp=5;
Zq=10;
Zr=%i*10;

Zpq=((Zp*Zq)+(Zq*Zr)+(Zr*Zp))/Zr;
Zqr=((Zp*Zq)+(Zq*Zr)+(Zr*Zp))/Zp;
Zrp=((Zp*Zq)+(Zq*Zr)+(Zr*Zp))/Zq;

printf(' Delta Equivalent : \n')
printf(' Zpq = %g + j(%g) ohm \n',real(Zpq),imag(Zpq))
printf(' Zqr = %g + j(%g) ohm \n',real(Zqr),imag(Zqr))
printf(' Zrp = %g + j(%g) ohm \n',real(Zrp),imag(Zrp))
```

**Scilab code Exa 2.19** Star Equivalent of the delta circuit

```
// Star Equivalent of the delta circuit

clc;
clear;

Z12=%i*5;
Z23=%i*-5;
Z31=%i*5;

Z1=(Z12*Z31)/(Z12+Z23+Z31);
Z2=(Z12*Z23)/(Z12+Z23+Z31);
Z3=(Z23*Z31)/(Z12+Z23+Z31);

printf('Star Equivalent :\n')
```

```
disp('ohms',Z1,'Z1 =')
disp('ohms',Z2,'Z2 =')
disp('ohms',Z3,'Z3 =')
```

**Scilab code Exa 2.20** To determine equivalent resistance using star delta transformation

```
//To determine equivalent resistance using star-
   delta transformation

clc;
clear;

Rax=30;
Rcx=30;
Rac=30;
Ray=30;
Rcy=30;

// Delta to star conversion of the triangle CAX in
   the circuit
Rx=Rax*Rcx/(Rax+Rcx+Rac);
Ra=Rax*Rac/(Rax+Rcx+Rac);
Rc=Rac*Rcx/(Rax+Rcx+Rac);

R1=Ra+Ray;// Resistance from the common to Y of the
   upper limb
R2=Rc+Rcy;// Resistance from the common to Y of the
   lower limb

Reff=R1*R2/(R1+R2); // Effective resistance of both
   the limbs

Rxy=Rx+Reff; // Effective resistance across X and Y

```

```
printf('The Equivalent resistance = %g ohms\n',Rxy)
```

**Scilab code Exa 2.21** Determine current through branch AB using loop and nodal analysis

```
// Determine current through branch AB using loop and nodal analysis

clc;
clear;

Is=4; // Current Source
//Resistances
Rab=2;
R1=4;// After point B towards the right
R2=1;

V=10; // Voltage source

//Using Simple Logic
i1=Is; // Current source connected in series with resistor
i2=10/(R1+R2);

printf('The Current through branch AB: \n')
printf('i) Simple Logic = %g A\n',i1)

//Using Loop Analysis

//Conversion of Current source into voltage source, R tends to infinity

R=poly(0,'R');

Rmat=[R+2 0; 0  5];
```

```
Vmat=[(4*R)-V;V];
Imat=inv(Rmat)*Vmat;
printf('\nii) Loop Analysis\n')
disp(Imat(1,1),'The current through AB is')
printf('\nWhere R tends to infinity\n')

R=%inf;
i1=(4-(V/R))/(1+(Rab/R));
printf('\n  = %g A\n',i1)

// Using Nodal Analysis
// Conversion of voltage source into current source, R then tends to zero

R=poly(0,'R');

// Nodal Eqaution
//0.5V1-0.5V2+0V3 = 4
//-0.5V1+(0.75+(1/R))V2-0.25V3 = 10/R
//0V1-0.25V2+1.25V3=0

Y=[0.5 -0.5 0;-0.5 (0.75+(1/R)) -0.25;0 -0.25 1.25]; // Admittance Matrix
Im=[4; (10/R); 0] ; // Current Matrix
Vm=inv(Y)*Im; // Voltage Matrix
V1=Vm(1,1);
V2=Vm(2,1);
V3=Vm(3,1);

DiffV=V1-V2;

printf('\niii) Nodal Analysis:\n')
disp(V2,'V2:',V1,'V1:')
Vdiff=roots(DiffV(2)-R); // To change data type
disp(DiffV,'V1-V2 :')
In=Vdiff/Rab; // Current due to nodal analysis

printf('\n The Current Through 2 ohm resistor = %g A
```

```
\n',In )
```

**Scilab code Exa 2.22** Current through 2 ohm resistor given a current source

```
//Current through 2 ohm resistor given a current
   source

clc;
clear;

Is=2; // Current Source

//Resistors connected directly to the current source
Rs1=0.5;
Rs2=0.5;

//Resistors in various branches starting from the
   top
R1=1;
R2=1;
R3=2;

//Conversion to voltage sources
V1=Rs1*Is; // Voltage across first half of the
   branch
V2=Rs2*Is; // Voltage across second half of the
   branch

// Voltage sources in the circuit
V3=1;
V4=2;

//Characteristic Equations
//2.5i1-1i2 = 2
//-1i1+3.5i2 = 2
```

```
R=[2.5 -1; -1 3.5]; // Resistor Vector
V=[(V1+V3);(V2+V4-V3)]; // Voltage Vector
I=inv(R)*V; // Current Vector
i1=I(1);
i2=I(2);

printf('The Current through the 2 ohm resistor = %g
    A\n',i2)
```

**Scilab code Exa 2.23** To find voltage v and current through 3 ohm resistor using nodal analysis

```
//To find voltage v and current through 3 ohm
    resistor using nodal analysis

clc;
clear;

V=poly(0,'V');

Va=8-V;
Vb=-6;

//Resistors in order from the 8V side
R1=1;
R2=2;
R3=3;
R4=4;

// Nodal Analysis

X=((8-Va)/R1)+((Vb-Va)/R3)-((Va-(4*V))/R2); //
    Characteristic equation to find V

```

```
V=roots(X);

Va=8-V;

I=(Vb-Va)/3;

printf('i) The Voltage V (across 1 ohm resistor) is %g V\n',V)

if(imag(sqrt(I))) // Condition to check for negative sign
    printf('ii) The Current through 3 ohm resistor is %g A flowing from A to B\n',abs(I))
else
printf('ii) The Current through 3 ohm resistor is %g A flowing from B to A\n',abs(I))
end
```

---

**Scilab code Exa 2.24** Determine the current through 10 ohm resistor using thevenins circuit

```
// Determine the current through 10 ohm resistor using thevenins circuit

clc;
clear;

//Source Voltages
V1=10;
V2=2;

// Resistances of upper limb
R1=15;
R2=25;

```

```
//Resistances of lower limb

R3=30;
R4=20;

//For a thevenin circuit
i1=(V1-V2)/(R1+R2); // Current in upper limb
i2=V1/(R3+R4); // Current in lower limb

Vac=(i1*R2)+2;
Vbc=(i2*R4);

Vab=Vac-Vbc; // Thevenin Voltage

Vth=Vab;
Zl=10; // Load resistance

Reff1=(R1*R2/(R2+R1));
Reff2=(R3*R4/(R3+R4));

Zth=Reff1+Reff2;

I=Vth/(Zl+Zth); // Curent through AB

printf('The current through the 10 ohm resistor = %g
    mA\n',I*1000)
```

# Chapter 3

# Three Phase Supply

**Scilab code Exa 3.1** To determine the parameters of a balanced 3 phase star connected to a resistive load

```
// To determine the parameters of a balanced 3 phase
   star connected to a resistive load

clc;
clear;

V=208;
Vph=V/sqrt(3);
R=35;

// Star Conncected load has its line current = phase
   current

Ia=Vph/R;
Ib=Ia*(expm(%i*(-2*%pi/3)));
Ic=Ia*(expm(%i*(2*%pi/3)));

Pperphase= (abs(Ia)^2)*R;

Pt=3*Pperphase;
```

```
// Resistive Load, p.f is unity

pf=1;

printf('The power factor is %g \n',pf)
printf('The total power dissipated = %g W \n',Pt)
printf('The currents of the system are\n Ia = %g /_0 A \n Ib = %g /_-120 A \n Ic = %g /_120 A \n',abs(Ia),abs(Ib),abs(Ic))
```

---

**Scilab code Exa 3.2** To determine the parameters of a balanced 3 phase star connected to an impedance

```
// To determine the parameters of a balanced 3 phase star connected to an impedance

clc;
clear;

V=208;
Vph=V/sqrt(3);
Z=15+(%i*20);

// Star Conncected load has its line current = phase current

Ia=Vph/Z;
Ib=Ia*(expm(%i*(-2*%pi/3)));
Ic=Ia*(expm(%i*(2*%pi/3)));

Pperphase= (abs(Ia)^2)*real(Z);

Pt=3*Pperphase;

```

```
Atheta=atand(imag(Ia)/real(Ia));
Btheta=atand(imag(Ib)/real(Ib));
Ctheta=atand(imag(Ic)/real(Ic));

pf=cosd(Atheta);

printf('The power factor is %g lagging \n',pf)
printf('The total power dissipated = %g W \n',Pt)
printf('The currents of the system are \n')
printf('Ia= %g /_%g A \n',abs(Ia),Atheta)
printf('Ib= %g /_%g A \n',abs(Ib),Btheta-180)
printf('Ic= %g /_%g A \n',abs(Ic),Ctheta)
```

**Scilab code Exa 3.3** To determine the potential of the star point and line currents

```
//To determine the potential of the star point and
   line currents

clc;
clear;

Zr=10*(expm(%i*%pi/6));
Zy=12*(expm(%i*%pi/4));
Zb=15*(expm(%i*2*%pi/9));

V=440;
Vph=V/(sqrt(3));

//Phase Voltages
Vr=Vph*(expm(%i*0));
Vy=Vph*(expm(%i*-2*%pi/3));
Vb=Vph*(expm(%i*2*%pi/3));

Vs=((Vr/Zr)+(Vy/Zy)+(Vb/Zb))/((1/Zr)+(1/Zy)+(1/Zb));
```

```
tvs=atand(imag(Vs)/real(Vs)); // Phase Angle of the
   star point voltage

Ia=(Vr-Vs)/Zr;
iat=atand(imag(Ia)/real(Ia)); // Angle of current in
    phase R
Ib=(Vy-Vs)/Zy;
ibt=atand(imag(Ib)/real(Ib)); // Angle of current in
    phase Y
Ic=(Vb-Vs)/Zb;
ict=atand(imag(Ic)/real(Ic)); // Angle of current in
    phase B

I=Ia+Ib+Ic;
I=ceil(real(I)*1000)+%i*(ceil(imag(I)*1000));

printf('The potential of the star point = %g /_%g V
   \n',abs(Vs),tvs)
printf('The line currents are : \n')
printf('R phase current = %g /_%g A \n',abs(Ia),iat)
printf('Y phase current = %g /_%g A \n',abs(Ib),ibt
   -180)
printf('B phase current = %g /_%g A \n',abs(Ic),ict)
```

**Scilab code Exa 3.4** To determine the line currents if one inductor is short circuited

```
//To determine the line currents if one inductor is
   short circuited

clc;
clear;

V=460; // Line to Line voltage
```

```
pf=0.8; // Power Factor
P=8*(10^3); // Power Consumed by the network

Vph=V/sqrt(3);

Iph=P/(sqrt(3)*V*pf);

theta=acos(pf);// Power factor angle
Z=(Vph/Iph)*(expm(%i*theta));

Va=V*expm(%i*0); // Voltage of Phase A
Vc=V*expm(%i*-2*%pi/3); // Voltage of Phase C

Ia=Va/Z; // Current in phase A
Ic=Vc/Z;// Current in phase C

iat=atand(imag(Ia)/real(Ia)); // Phase angle of Ia
ict=atand(imag(Ic)/real(Ic));// Phase angle of Ic

tac=iat-ict; // Angle between current Ia and Ic

Ib=sqrt((abs(Ia)^2)+(abs(Ic)^2)+(2*abs(Ia)*abs(Ic)*
    cosd(tac)));

printf('The line currents are : \n')
printf('Phase a = %g/_%g A \n',abs(Ia),iat)
printf('Phase b = %g A \n',abs(Ib))
printf('Phase c = %g/_%g A \n',abs(Ic),ict)
```

**Scilab code Exa 3.5** To find line current and pf and powers of a balanced delta load

```
//To find line current and pf and powers of a
    balanced delta load

```

```
clc;
clear;

Z=8+6*%i; // Load
V=230; // Voltage supply

iR=V/Z;
theta= atand(imag(iR)/real(iR));

Il= iR*sqrt(3); // Line current

Pa=sqrt(3)*V*abs(Il)*cosd(theta); // Active Power
Pr=sqrt(3)*V*abs(Il)*sind(theta); // Reactive Power

Pt=sqrt(3)*V*abs(Il); // Total Volt amperes

printf('The line current = %g A \n',abs(Il))
printf('The power factor = %g lagging \n',cosd(theta
    ))
printf('The Active Power = %g kW \n',abs(Pa)/1000)
printf('The Reactive Power = %g kV Ar \n',abs(Pr)
    /1000)
printf('The total volt amperes = %g kVA \n',abs(Pt)
    /1000)
```

**Scilab code Exa 3.6** To find Line currents and star connected resistors for the same power

```
//To find Line currents and star connected resistors
    for the same power

clc;
clear;

// Phase Voltages
```

```
Vr=400*(expm(%i*0));
Vy=400*(expm(%i*-2*%pi/3));
Vb=400*(expm(%i*2*%pi/3));

Zry=100; // Impedance between Phase R and Phase Y
Zyb=%i*100;// Impedance between Phase Y and Phase B
Zbr=-%i*100;// Impedance between Phase B and Phase R

Iry=Vr/Zry;
Iyb=Vy/Zyb;
Ibr=Vb/Zbr;

Ir=Iry-Ibr;
Iy=Iyb-Iry;
Ib=Ibr-Iyb;

//Phase angles of the line currents in RYB sequence
tr=atand(imag(Ir)/real(Ir));

if(real(Iy)==0)
ty=atand((imag(Iy)/abs(imag(Iy)))*%inf);
else
ty=atand(imag(Iy)/real(Iy));
end
if(real(Ib)==0)
tb=atand((imag(Ib)/abs(imag(Ib)))*%inf);
else
tb=atand(imag(Ib)/real(Ib));
end

P=(Iry^2)*Zry; // Power consumed by the circuit (
   Arm RY)

Vph=Vr/sqrt(3); // Phase voltage in a star connected
    system

R=poly([0 1],'R','c');

```

```
x=(3*(Vph^2))-(P*R); // Characteristic Eqaution to
    find R

R=roots(x);

printf('a) The line currents in RYB sequence are : \
    n')
printf(' R line = %g/_%g A \n',abs(Ir),tr);
printf(' Y line = %g/_%g A \n',abs(Iy),ty);
// Error in text book answer
printf(' B line = %g/_%g A \n \n',abs(Ib),tb);
printf('b) The value of resistors to draw same power
    as in problem statement a) = %g ohms \n',R(1))
```

**Scilab code Exa 3.7** Reduction in load when one resistor is removed

```
//Reduction in load when one resistor is removed

clc;
clear;

// Assuming the variables to be eqaul to unit
    quantities

Vph=1;
Vl=sqrt(3)*Vph;
R=1;

// Star connected

Pis=3*(Vph^2)/R; // Initial Power

Pfs=(Vl^2)/(2*R); // Power when one resitor is
    removed

```

```
pers=(Pis-Pfs)*100/Pis; // Percentage decrease in
    Load

// Mesh connected

Pim=3*(Vl^2)/R; // Initial Power

Pfm=2*(Vl^2)/R; // Power when one resitor is removed

perm=(Pim-Pfm)*100/Pim; // Percentage decrease in
    Load

printf(' Vl= square root (3) * Vph \n \n')
printf('a)  Star Connected Power = 3*(Vph^2)/R \n')
printf('    When one resistor is removed Power = (Vl
    ^2)/2R \n')
printf('    The percentage reduction in load = %g \n
     \n',pers)

printf('b)  Mesh Connected Power = 3*(Vl^2)/R \n')
printf('    When one resitor is removed, Power = 2*(
    Vl^2)/R \n' )
printf('    The percentage reduction in load = %g \n
    ',perm)
```

**Scilab code Exa 3.8** To measure power by two wattmeter method

```
//To measure power by two wattmeter method

clc;
clear;

pf=0.85 // Power Factor

Po=37.3*(10^3); // Power Output
```

```

eff=90/100; // Efficiency

V=500; // Rated Voltage

Pi=Po/eff; // Power Input

phi=acosd(pf); // Power Factor angle

printf('W1 + W2 = %g kW \n',Pi/1000)
printf('tan(phi) = square root (3)*(W2-W1)/(W2+W1) =
    %g \n',tand(phi))

x=Pi; // Let x = W1+W2

y= tand(phi)*x/(sqrt(3)); // Let y = W2-W1

printf('W1 + W2 = %g kW \n',x/1000)
printf('W2 - W1 = %g kW \n',y/1000)
printf('W2 = %g kW \n',(x+y)/(2*1000))
printf('W1 = %g kW \n',(x-y)/(2*1000))
```

**Scilab code Exa 3.9** To find power using two wattmeter method of a circuit with non reactive resistances

```
//To find power using two wattmeter method of a
    circuit with non reactive resistances

clc;
clear;

V=400; // Line Voltage
Vph=V/(sqrt(3)); // Magnitude of Phase Voltage

// Phase Voltage in RYB sequnce
```

```
Vr=Vph*(expm(%i*0));
Vy=Vph*(expm(%i*-2*%pi/3));
Vb=Vph*(expm(%i*2*%pi/3));

// Resitances of the RYB limbs
Rr=10;
Ry=15;
Rb=20;

// Taking Vr as reference
// Millain's Theorem

Vs= ((Vr/Rr)+(Vy/Ry)+(Vb/Rb))/((1/Rr)+(1/Ry)+(1/Rb)); // Star point voltage

//Line Currents in RYB sequence
Ir= (Vr-Vs)/Rr;
Iy= (Vy-Vs)/Ry;
Ib= (Vb-Vs)/Rb;

Vry=Vr-Vy;
Vby=Vb-Vy;

W1= real(Vry*conj(Ir));
W2= real(Vby*conj(Ib));

Wt= W1+W2; // Total Power

// Note Iy in the text book there is a error in the sign of the real part of Vy

printf('The line currents in RYB sequence are : \n')
disp(Ir,' R line :')
printf(' Magnitude = %g A \n',abs(Ir))
disp(Iy,' Y line :')
printf(' Magnitude = %g A\n',abs(Iy))
disp(Ib,' B line :')
printf(' Magnitude = %g A\n \n',abs(Ib))
```

```
printf(' Total Power = %g kW \n \n',Wt/1000)
printf(' W1 = %g kW\n',W1/1000)
printf(' W2 = %g kW\n',W2/1000)
```

**Scilab code Exa 3.10** Two wattmeter power dertermination for a delta system

```
// Two wattmeter power dertermination for a delta
   system

clc;
clear;

V=250; // Phase Voltage

// Phase Voltage in RYB sequnce
Vry=V*(expm(%i*0));
Vyb=V*(expm(%i*-2*%pi/3));
Vbr=V*(expm(%i*2*%pi/3));

// Resitances of the RYB limbs
Rry=10+%i*10;
Ryb=20-%i*15;
Rbr=10+%i*20;

// Phase Currents in RYB

Iry= Vry/Rry;
Iyb= Vyb/Ryb;
Ibr= Vbr/Rbr;

// Phase Current Angles wrt to Vr

ary=atand(imag(Iry)/real(Iry));
ayb=atand(imag(Iyb)/real(Iyb));
```

```
abr=atand(imag(Ibr)/real(Ibr));

// Line Currents in RYB
Ir=Iry-Ibr;
Iy=Iyb-Iry;
Ib=Ibr-Iyb;

W1=real(-Vbr*conj(Ir));
W2=real(Vyb*conj(Iy));

Wt= W1+W2; // Total Power

printf('i)\n')
printf(' The Currents in each branch are : \n')
printf(' Branch RY = %g/_%g A \n',abs(Iry),ary)
printf(' Branch YB = %g/_%g A \n',abs(Iyb),ayb)
printf(' Branch BR = %g/_%g A \n',abs(Ibr),abr)

printf('ii) \n')
printf('The line currents in RYB sequence are : \n')
disp(Ir,' R line :')
printf(' Magnitude = %g A \n',abs(Ir))
disp(Iy,' Y line :')
printf(' Magnitude = %g A\n',abs(Iy))
disp(Ib,' B line :')
printf(' Magnitude = %g A\n \n',abs(Ib))

// Precision is more, The Text book includes round
   off error
printf(' W1 = %g W\n',W1)
printf(' W2 = %g W\n',W2)
```

# Chapter 4

# Basic Instruments

**Scilab code Exa 4.1** Torque on the coil at a current of 1mA

```
//Torque on the coil at a current of 1mA

clc;
clear;

N=60;
B=50*(10^-3);
I=1*(10^-3);

l=3*(10^-2);

// w= 2*r; w is the width
w=2*(10^-2);

Td=N*B*I*l*w;

printf('The torque on the coil carrying 1mA = %g
   micro Nm \n',Td*(10^6))
```

**Scilab code Exa 4.2** To find the deflection produced by 200V

```
//To find the deflection produced by 200V

clc;
clear;

R=10*(10^3);
V=200;
B=80*(10^-3);
N=100;
A=9*(10^-4); // The area of the coil is the product
    of the length and width (l.2r)
I=V/R;

Td=N*B*I*A;

K=30*(10^-7);

theta=Td/K;

printf('The deflection produced by 200V = %g degrees
    \n',theta)
```

---

**Scilab code Exa 4.3** Reading on ammeters when their shunts are interchanged

```
// Reading on ammeters when their shunts are
    interchanged

clc;
clear;

I=10;
Ra=1000;
```

```
Rsa=0.02;

Rb=1500;
Rsb=0.01;

deff('x=cur(y,z)','x=I*z/y')

Ia1=cur(Ra,Rsa); // Initial Current in meter A
Ia2=cur(Ra,Rsb); // Changed Current in meter A

Ib1=cur(Rb,Rsb); // Initial Current in meter B
Ib2=cur(Rb,Rsa); // Changed Current in meter B

//Factor by which the current readings change in the
    two ammeters

A=Ia2/Ia1; // Ammeter A
B=Ib2/Ib1; // Ammeter A

printf('The initial current in ammeter A and ammeter
    B are %g A and %g A respectively. \n \n',I,I)

printf('The current in ammeter A and ammeter B when
   the shunt resistances are interchanged are %g A
   and %g A respectively. \n \n',I*A,I*B)
```

**Scilab code Exa 4.4** To create an instrument that measures voltages and currents upto a rated value

```
//To create an instrument that measures voltages and
    currents upto a rated value

clc;
clear;

```

```
Rm=10;
Im=50*(10^-3);
V=750;
I=100;

//To Calculate the required voltage a resistor
   should be added in series with the internal
   resistance

R=poly([0 1],'R','c');

sr=Im*(R+Rm)-V;

R=roots(sr);//Characteristic equation to find R

// To attain the required current , a resistor should
    be added in parallel to the internal resistance

r=poly([0 1],'r','c');// Characteristic equation to
   find r

pr=Im*(r+Rm)-(I*r);

r=roots(pr);

printf('To Read 750 V a series resitance of %g ohms
   should be connected to the instrument \n',R)
printf('To Read 100 A a parallel resitance of %g
   milli ohms should be connected to the instrument
   \n',r*1000)
```

**Scilab code Exa 4.5** To determine the range and current and deflection at various conditions

```
//To determine the range and current and deflection
```

```
   at various conditions

clc;
clear;

I=25;
theta=90;

// Various conditions
ta=360; // Angle in case a
tb=180; // Angle in case b
Ic=20; // Current in case c

// theta directly proportional to the square of the
   current

Ia=sqrt(ta*(I^2)/theta);

Ib=sqrt(tb*(I^2)/theta);

tc=((Ic/I)^2)*theta;

printf('a) Full Scsle deflection (360) current = %g
   A \n',Ia)
printf('b) Half Scsle deflection (180) current = %g
   A \n',Ib)
printf('c) Deflection for a current of 20 A = %g
   degrees \n',tc)
```

**Scilab code Exa 4.6** Error calculation

```
//Error calculation

clc;
clear;
```

```
I=20; //Current
V=230;// Voltage
C=480;// Meter Constant
L=4.6*(10^3);// Load
t=66/3600; // Time in hour

R=40; //No of revolutions

Pc=L*t/1000; // Energy Consumed in kWhr

Pr=R/C; // Energy recorded in kWhr

err=(Pc-Pr)*100/Pc;

printf('The Error in the meter is that the disc
    rotates %g percent slow \n',err)
```

**Scilab code Exa 4.7** Dynamometer wattmeter power calculation of the load

```
//Dynamometer wattmeter power calculation of the
    load

clc;
clear;

P=250; // Power Recorded by the wattmeter

V=200; // Load voltage

R=2000; // Resistance of the highly non-inductive
    pressure coil

I=V/R; // Ohm's Law

```

```
Pcoil=V*I; // Power Absorbed by the pressure coil

Pl=P-Pcoil; // Power taken by the load

printf('The Power taken by the load = %g watts. \n',
   Pl)
```

**Scilab code Exa 4.8** Percentage error calculation in a wattmeter

```
//Percentage error calculation in a wattmeter

clc;
clear;

//Rated Parameters
I=50;
V=230;

R=61;// No. of revolutions
t=37/3600; // Time in hours

C=520; // Normal Disc Speed

Pfl=I*V;// Power at full load

Ps=Pfl*t/1000; // Power Supplied in kWhr

Pr=R/C; //Power recorded in kWhr

err=(Ps-Pr)*100/Ps;

printf('The Percentage Error = %g percent slow \n',
   err)
```

# Chapter 5

# Transformer

**Scilab code Exa 5.1** To find flux density in the core and induced emf in the secondary winding

```
// To find flux density in the core and induced emf
   in the secondary winding

clc;
clear;

E1=500;
A=60*(10^-4);
f=50;
N1=400;
N2=1000;

// E=4.44*f*N*Bm*A Induced EMF equation

Bm=E1/(4.44*f*N1*A);

E2=4.44*f*N2*Bm*A;

printf('a) The peak value of the flux density in the
    core = %f tesla \n',Bm)
```

```
printf('b) The voltage induced in the secondary
    winding = %f V \n',E2)
```

**Scilab code Exa 5.2** To calculate the number of turns per limb on the high and low voltage sides

```
// To calculate the number of turns per limb on the
    high and low voltage sides

clc;
clear;

f=50;
A=400*(10^-4);
Bm=1;
V1=3000;
V2=220;

l=2; // Number of limbs

//Neglecting the series voltage drop

// Induced EMF equation
a=V1/(4.44*f*A*Bm);

b=V2*a/V1;

if(modulo(round(a),2)==0) // No. of turns is a whole
     even number as it has 2 limbs
    N1=round(a);
else
    N1=round(a)+1;
end

if(modulo(round(b),2)==0) // No. of turns is a whole
```

```
    even number as it has 2 limbs
    N2=round(b);
else
    N2=round(b)+1;
end

printf('The number of turns in the high voltage side
    per limb = %d \n',N1/l)
printf('The number of turns in the low voltage side
   per limb = %d \n',N2/l)
```

**Scilab code Exa 5.3** To calculate resistance of primary interms of secondary and vice versa

```
// To calculate resistance of primary interms of
   secondary and vice versa

clc;
clear;

N1=90;
N2=180;

R2=0.233;
R1=0.067;

n=N2/N1; // Transformation ratio

R1w2=(n^2)*R1;
R2w1=R2/(n^2);

Rt=R1+R2w1; // Total resistance in terms of primary

printf('a) Resistance of primary in terms of the
   secondary = %f ohms \n',R1w2)
```

```
printf('b) Resistance of secondary in terms of the
    primary = %f ohms \n',R2w1)
printf('c) Total resistance of the transformer in
    terms of the primary winding =%f ohms \n',Rt)
```

**Scilab code Exa 5.4** Total resitance and total copper loss at full load

```
// Total resitance and total copper loss at full
    load

clc;
clear;

P=40*(10^3);
E1=2000;
E2=250;

n=E2/E1; //Transformation ratio

R1=1.15;
R2=0.0155;

R1w2=R1*(n^2);
R2w1=R2/(n^2);

Rt=R2+R1w2;

// Full load currents
I1=P/E1;
I2=P/E2;

Pc1=(I1^2)*R1; // Primary Loss
Pc2=(I2^2)*R2; // Secondary Loss

Pc= Pc1+Pc2; // Total Copper loss at full load
```

```
printf('a) The total resitance in terms of the
    secondary winding = %f ohms \n',Rt)
printf('b) Total copper loss on full load = %f watts
    ',Pc)
```

**Scilab code Exa 5.5** Voltage regulation at a pf lagging

```
//Voltage regulation at 0.8 pf lagging

clc;
clear;

E1=1100;
E2=110;

P=5*(10^3);

I=P/E1; // Primary full load current

I2=P/E2;// Secondary full load current

V=33;

pf=0.8; // Power Factor lagging , so the angle is
    positive

theta=acosd(pf);// Power factor angle

Pc=85;

R=Pc/(I^2);

Z=V/I;

```

```
V1=E1;

X=sqrt((Z^2)-(R^2));

// Using equation 5.22 to determine V2

V2=poly([0 1],'V2','c');

x=(V2^2)+(2*V2*I*R*pf)+(2*V2*I*X*sind(theta))+((I^2)
    *((R^2)+(X^2)))-(V1^2);

r=roots(x);

a1=sqrt(r(1));
a2=sqrt(r(2));

if(imag(a1))
    V2=r(2);
else
    if(imag(a2))
        V2=r(1);
    else
        disp('Error')
    end
end

reg=(V1-V2)/V2; // Voltage regulation

regper=reg*100;// Voltage regulation percent

disp(x,'The characteristic equation to find V2
    equated to zero is')

disp(regper,'The percentage voltage regulation for a
     load at 0.8 pf lagging is')
```

**Scilab code Exa 5.6** Regulation at lagging leading and unity power factors

```
// Regulation at laggiing leading and unity power
   factors

clc;
clear;

ol=0.01;// Ohmic loss is 1% of the output

// Output = V*I; Ohmic loss =(I^2)*R

//(I*R)/V = 0.01

rd=0.05; // Reactance drop is 5% of the output
   voltage

// Power Factors
pf1=0.8;// lag
pf2=1; // unity
pf3=0.8;// lead

deff('y=angle(x)','y=acosd(x)');// Function to find
   out the angle

// Angles
t1=angle(pf1);// Positive sign as it is lagging
t2=angle(pf2);
t3=-angle(pf3); // Minus sign as it is leading

deff('a=vr(b)','a=((ol*cosd(b))+(rd*sind(b)))*100');
   // Function to find out voltage regulation

printf('The voltage regulation percentages is as
```

```
follows \n')
printf('a) For 0.8 p.f lag = %f percent \n',vr(t1))
printf('b) For unity p.f = %f percent \n',vr(t2))
printf('c) For 0.8 p.f lead = %f percent \n',vr(t3))
```

**Scilab code Exa 5.7** Calculate the circuit parameters of a transformer using OC and SC tests

```
// Calculate the circuit parameters of a transformer
    using OC and SC tests

clc;
clear;

E1=200;
E2=400;

n=E2/E1; // Transformation ratio

// O.C Calculations
V1=200;
Ioc=0.7;
Pi=70;

R0=(V1^2)/Pi;

Iw=V1/R0;

Im=sqrt((Ioc^2)-(Iw^2));

X0=V1/Im;

//S.C Calculations on HT side

Pc=80;
```

```
I=10;
V=15;

Rth= Pc/(I^2);
Z=V/I;

Xth=sqrt((Z^2)-(Rth^2));

// Both these value are referred to HT side , but the answer is required to be referred to LT side

Xtl=Xth/(n^2); // Reactance referred to LT side
Rtl=Rth/(n^2); // Resistance referred to LT side

printf('The Circuit parameters referred to LT side is as follows \n')

printf('Ro = %f ohms \n Xo = %f ohms \n Rt = %f ohms \n Xt = %f ohms \n',R0,X0,Rtl,Xtl)
```

**Scilab code Exa 5.8** To calculate terminal voltage and current and efficiency

```
// To calculate terminal voltage and current and efficiency

clc;
clear;

n=10; // Transformation ratio

E1=200;

R0=400;
X0=251*%i;
```

```
R1=0.16;
X1=0.7*%i;

R2=5.96; // As referred to the primary side
X2=4.44*%i; // As referred to the primary side

I1=E1/(R1+R2+X1+X2);

t1=atand(imag(I1)/real(I1));// Angle for primary current

Iw=E1/R0;
Im=E1/X0;

Ip=Iw+Im+I1;

Zl=R2+X2;

V2p=I1*Zl;// Secondary terminal voltage referred to primary side

V2=n*V2p;

t2=atand(imag(V2)/real(V2)); // Angle for V2

Po= (abs(I1)^2)*R2; // Output power

Pc=(abs(I1)^2)*R1;// Copper Loss

Pil=(abs(Iw)^2)*R0;// Iron Loss

eff= Po*100/(Po+Pc+Pil)// Efficiency

printf('a) The secondary terminal voltage = %f /_%f V \n',abs(V2),t2)
printf('b) The primary current = %f /_%f A \n',abs(I1),t1)
```

```
printf('c) The efficiency is %f percent \n',eff)
```

**Scilab code Exa 5.9** Regulation at full load pf lag

```
// Regulation at full load p.f 0.8 lag

clc;
clear;

Pi=500*(10^3);// Power Input
Meff=97/100;// Max Efficiency
pf1=1;

E1=3300;
E2=500;

Po=Pi*pf1*3/4;

// Iron loss = Copper loss at maximum efficiency

x=poly([0 1],'x','c');

Pin=Po+(2*x);

xx=(Pin*Meff)-Po;

x=roots(xx); // Iron Loss = Copper Loss

I2=Po/E2;

R=x/(I2^2);

I2fl=Pi/E2;

Rfl=E2/I2fl;
```

```
// Per unit values
Rpu=R*100/Rfl;
Zpu=10;
Xpu=sqrt((Zpu^2)-(Rpu^2));

pf2=0.8; // Lagging

ang=acosd(pf2);// Positive Angle as it is lagging

perreg=(Rpu*cosd(ang))+(Xpu*sind(ang));

printf('The regulation at full load, p.f 0.8 lag =
    %f percent\n',perreg)
```

---

**Scilab code Exa 5.10** Calculate efficiency on unity pf at different cases

```
//Calculate efficiency on unity pf at different
    cases

clc;
clear;

Pi=25*(10^3);

E1=2000;
E2=200;

Pil=350;
Pc=400;

// Full load efficiency

nfl=Pi*100/(Pi+Pil+Pc);

```

```
//Half Load efficiency

Pihl=Pi/2;// Half Load
nhl=Pihl*100/(Pihl+Pil+(Pc/4));

// Load at which maximum efficiency occurs

Piml=sqrt(Pil/Pc)*Pi;
Pcm=Pc*((Piml/Pi)^2);

printf('a) Efficiency at full load = %f percent \n',
   nfl)
printf('b) Efficiency at half load = %f percent \n',
   nhl)
printf('c) Maximum Efficiency will occur at %f KVA
   and the losses are each %d watt. \n',(Piml/1000),
   Pcm)
```

**Scilab code Exa 5.11** Calcualte efficiencies at various loads

```
//Calcualte efficiencies at various loads

clc;
clear;

P=100*(10^3);// Power Input
Pc=1000;// Copper Loss
Pil=1000;// Iron Loss
pf=0.8;

deff('y=unity(x)','y=(P*100*x)/((P*x)+Pil+(Pc*(x^2))
   )')// Unit Power Factor
deff('y=pfactor(x)','y=(P*100*x*pf)/((P*pf*x)+Pil+(
   Pc*(x^2)))')// 0.8 p.f

```

```
printf('a) Unity power factor efficiencies at \n \n'
   )
printf('i) Half of full load = %f percent \n',unity
   (1/2))
printf('ii) Full load = %f percent \n',unity(1))
printf('iii) (5/4) of full load = %f percent \n \n',
   unity(5/4))


printf('b) 0.8 power factor efficiencies at \n \n')
printf('i) Half of full load = %f percent \n',
   pfactor(1/2))
printf('ii) Full load = %f percent \n',pfactor(1))
printf('iii) (5/4) of full load = %f percent \n',
   pfactor(5/4))
```

**Scilab code Exa 5.12** To determine all day efficiency

```
// To determine all day efficiency

clc;
clear;

p=15*(10^3);
t1=12;
t2=6;
t3=6;

pf1=0.5;
pf2=0.8;
pf3=0.9;

x=poly([0 1],'x','c');

nm=0.98; // Max Efficiency
```

```

y=(nm*(p+(2*x)))-p;

x=roots(y); // To find the iron loss or copper loss at unity p.f for maximum efficiency

Pil=x; // Iron loss

Pc=x; // Copper Loss at unity p.f for maximum efficiency

deff('a=culoss(b,c)','a=b*Pc*((c/(p/1000))^2)');

Pc1=culoss(12,(2/pf1)); // Total Copper Loss for 12 hrs - 2 kW at p.f 0.5
Pc2=culoss(6,(12/pf2)); // Total Copper Loss for 6 hrs - 12 kW at p.f 0.8
Pc3=culoss(6,(18/pf3)); // Total Copper Loss for 6 hrs - 18 kW at p.f 0.9

Po=((12*2)+(6*12)+(6*18))*(10^3);// Power Output

eff=Po*100/(Po+(Pc1+Pc2+Pc3)+(24*Pil));

// Note the iron loss has to be considered to calculate the Efficiency , Text Error

printf('The all day effciency = %f percent \n',eff)
```

**Scilab code Exa 5.13** Calculating Efficiency using Sumpner test

```
// Calculating Efficiency using Sumpner test

clc;
clear;
```

```
P=200*(10^3);

W1= 4*(10^3); // Total iron loss for both the
    transformers
W2= 6*(10^3); // Total copper loss for both the
    transformers

n=P*100/(P+(W1/2)+(W2/2));// Efficiency

printf('The Efficiency of each transformer at full
    load = %f percent \n',n)
```

**Scilab code Exa 5.14** To determine the ratio of weights of copper

```
// To determine the ratio of weights of copper

clc;
clear;

n=3; //transformation ratio

// Ratio of weights of copper in an ato transformer
    and a two winding transformer

roc=(1-(1/n));

printf('Ratio of weights of copper in an ato
    transformer and a two winding transformer = %f \n
    ',roc)
```

**Scilab code Exa 5.15** To find voltage ratio and output

```
// To find voltage ratio and output

clc;
clear;

E1=11500;
E2=2300;

n1=(E1+E2)/E1; // Voltage ratio of 13.8 kV/11.5 kV
    auto transformer

Pi=100*(10^3);

P1=Pi*n1/(n1-1);

n2=(E1+E2)/E2; // Voltage ratio of 13.8 kV/2.3 kV
    auto transformer

P2=Pi*n2/(n2-1);

printf('The transformation ratio of the auto
    transformer is %g and is rated %g / %g kV, %g KVA
    \n',n1,(E1+E2)/1000,E1/1000,P1/1000)
printf('The transformation ratio of the auto
    transformer is %g and is rated %g / %g kV, %g KVA
    \n',n2,(E1+E2)/1000,E2/1000,P2/1000)
```

**Scilab code Exa 5.16** Determine primary and secondary voltages and current

```
// Determine primary and secondary voltages and
    current

clc;
clear;
```

```
R1=100;
R2=40;

P=2; // Power

r=sqrt(R2/R1); // n2/n1 Turns ratio

if(r<1)
    printf(' The turns ratio is 1 : %g \n',(1/r));
else
    printf('The turns ratio is %g : 1 \n',r);
end

V1=sqrt(P*(R1));
V2=sqrt(P*(R2));

I1=V1/R1;
I2=V2/R2;

printf('\n Voltages are as follows \n')
printf('The primary voltage = %g V \n',V1)
printf('The secondary voltage = %g V \n',V2)
printf('\n Currents are as follows \n')
printf('The primary current = %g A \n',I1)
printf('The secondary current = %g A \n',I2)
```

---

**Scilab code Exa 5.17** Equivalent resistance and leakage reactance wrt primary

```
// Equivalent resistance and leakage reactance wrt
   primary

clc;
clear;
```

```

P=1200;
V=60;
I1=100;
R1eq=P/(I1^2);

Zeq=V/I1;

X1eq=sqrt((Zeq^2)-(R1eq^2));

// Secondary short circuited there the parameters calculated are wrt to primary itself

printf('Equivalent Resistance of the transformer w.r.t primary = %g ohms \n',R1eq)
printf('Leakage Reactance of the transformer w.r.t primary = %g ohms \n',X1eq)// Text Book Error Please note
```

---

**Scilab code Exa 5.18** To determine Input current and voltage during SC test

```
// To determine Input current and voltage during SC test

clc;
clear;

Vh=6600;
Vl=250;
V=400;

a=Vh/Vl; // Turns ratio

Rh=0.21;
```

```
Rl=2.72*(10^-4);

Xh=1;
Xl=1.3*(10^-3);

Rt=Rh+Rl*(a^2); // Equivalent resistance w.r.t the
    primary
Xt=Xh+Xl*(a^2); // Equivalent reactance w.r.t the
    primary

ZHeq= sqrt((Rt^2)+(Xt^2));

Ih=V/ZHeq; // Current on high voltage side

Pi=(Ih^2)*Rt; // Power input

printf('W.R.T High Voltage side the equivalent
    resistance is %g ohms and the equivalent
    reactance is %g ohms \n',Rt,Xt)

printf('The current on the high voltage side is %g A
    \n',Ih)

printf('Power Input on the high voltage side is %g
    kW \n',Pi/1000)
```

**Scilab code Exa 5.19** To determine the load for max efficiency at two power factors

```
// To determine the load for max efficiency at two
    power factors

clc;
clear;

```

```
P=100*(10^3); // Power Input

E1=1000;
E2=10000;

Pil=1200;

I2=P/E2; // Full load current on the HV side

Isc=6; // Current for 500W copper loss in HV winding
Psc=500; // Copper Loss for 6A in HV winding

Pc=((I2/Isc)^2)*Psc; // Copper loss at full load.

Pmax=sqrt(Pil/Pc)*P; // Is a factor of square root
    of the ratio of Iron loss and Copper loss at full
     load.

deff('x=eff(y,z)','x=(P*y*z)*100/((P*y*z)+Pil+(Pc*(z
    ^2)))')// Function to find the eifficiency for a
    given power factor(y) and load(z).

printf('a) The Efficiency at various loads for unity
     power factor are as follows. \n')
printf('i) At 25 percent load = %f percent \n',eff
    (1,0.25))
printf('ii) At 50 percent load = %g percent \n',eff
    (1,0.5))
printf('iii) At 100 percent load = %g percent \n',
    eff(1,1))

printf('\n b) The Efficiency at various loads for
    0.8 power factor are as follows. \n')
printf('i) At 25 percent load = %g percent \n',eff
    (0.8,0.25))
printf('ii) At 50 percent load = %g percent \n',eff
    (0.8,0.5))
printf('iii) At 100 percent load = %g percent \n \n'
```

```
,eff(0.8,1))

printf('The Load at which efficiency is maximum = %g
    kVA \n',(Pmax/1000))
```

**Scilab code Exa 5.20** To determine the max regulation and the pf at which it occurs

```
//To determine the max regulation and the pf at
    which it occurs

clc;
clear;

Vr=2.5;
Vx=5;

printf('The expression for voltage requlation is y=
    %g cos(phi) + %g sin(phi) \n',Vr,Vx )

printf('Differenciating w.r.t phi and equating it to
     zero, we get the power factor angle \n')

printf('We get tan(phi)=> Vr/Vx => 5/2.5 => 2 \n \n'
    )

phi=atand(Vx/Vr); // power factor angle

y= Vr*cosd(phi)+Vx*sind(phi); // Max Volatge
    regulation

printf('The maximum regulation is %g percent \n and
    the power factor at which it occurs is %g degrees
     \n',y,phi)
```

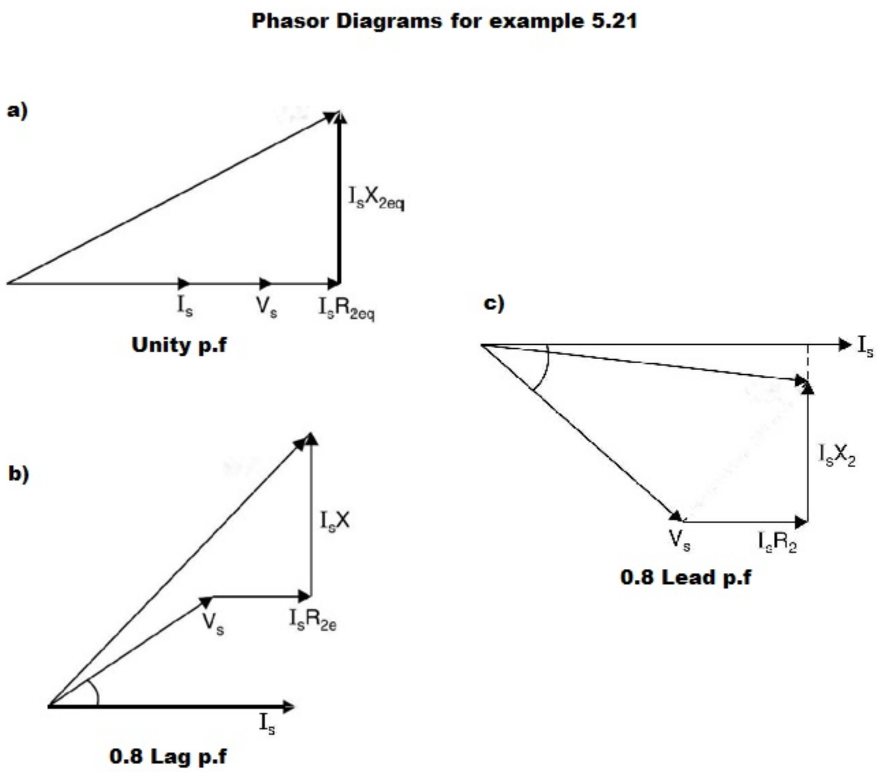


Figure 5.1: To calculate secondary terminal voltage and full load efficiency at unity pf

**Scilab code Exa 5.21** To calculate secondary terminal voltage and full load efficiency at unity pf

```
//To calculate secondary terminal voltage and full
   load efficiency at unity pf

clc;
```

```
clear;

P=4*(10^3);
E1=200;
E2=400;

// O.C Test
V=200;
Pil=70; // Iron Loss
Ioc=0.8;

R0=(V^2)/Pil;

Iw=V/R0;
Im=sqrt((Ioc^2)-(Iw^2));

X0=V/Im;

// S.C Test
Vsc=17.5;
Isc=9;
Psc=50;

R2eq=Psc/(Isc^2);

Z2eq=Vsc/Isc;

X2eq=sqrt((Z2eq^2)-(R2eq^2));

Is=P/E2; // Full load current

Pc=((Is/Isc)^2)*Psc;

fleff=(P*100)/(P+Pil+Pc);// Full load efficiency

printf('i) The Full load efficiency at unity power
   factor = %g percent \n \n',fleff)

```

```
// Secondary Terminal voltages cosidering full load secondary current as reference

Vs=poly([0 1],'Vs','c');

Vz=Is*(R2eq+(X2eq*%i));

// Using the characteristic equation in polar form, 'Is' as reference
// E = V/_theta + Is/_0 *(Z/_phi)

// Function to evalulate the right side of the equation in complex form
deff('a=stv(b)','a=Vs*(complex(cosd(b),sind(b)))+Vz')

case1=stv(acosd(1));
case2=stv(acosd(0.8));
case3=stv(-acosd(0.8));

// Funtion to calculate the characteristic equation of Vs
deff('x=svol(y)','x=(real(y)^2)+(imag(y)^2)-(E2^2)')

cs1=svol(case1);
cs2=svol(case2);
cs3=svol(case3);

// Roots of the characteristic equations

r1=roots(cs1);
r2=roots(cs2);
r3=roots(cs3);


// To find the positive roots
if(imag(sqrt(r1(1))))
    Vs1=r1(2);
```

```
else
    Vs1=r1(1);
end

if(imag(sqrt(r2(1))))
    Vs2=r2(2);
else
    Vs2=r2(1);
end

if(imag(sqrt(r3(1))))
    Vs3=r3(2);
else
    Vs3=r3(1);
end

printf('ii) The Secondary terminal voltages for
   various power factors are as follows \n')
printf('a) At Unity power factor , Vs = %g V \n',Vs1)
printf('b) At 0.8 power factor(Lagging), Vs = %g V \
   n',Vs2)
printf('c) At 0.8 power factor(Leading), Vs = %g V \
   n',Vs3)
```

# Chapter 6

# DC Machines

**Scilab code Exa 6.1** To Determine the useful flux per pole

```
//To Determine the useful flux per pole

clc;
clear;

E=600;
N=1200;
Z=250;
A=2;
P=6;

// EMF Equation

phi=E*A*60/(N*Z*P);// Flux developed

printf('The useful flux per pole = %g Wb \n',phi)
```

**Scilab code Exa 6.2** Determine the EMF generated

```
// Determine the EMF generated

clc;
clear;

phi=40*(10^-3);
A=8; //Lap Winding
P=8;
N=400;
Z=960;

E=P*N*Z*phi/(60*A); // EMF equation of a DC machine

printf('The EMF generated = %g volts \n',E)
```

**Scilab code Exa 6.3** Determine the EMF generated in a wave winding

```
// Determine the EMF generated in a wave winding

clc;
clear;

phi=40*(10^-3);
A=2; //Wave Winding
P=8;
E=400;
Z=960;

N=E*60*A/(phi*Z*P) // EMF equation of a DC machine

printf('The speed generated = %g rpm \n',N)
```

**Scilab code Exa 6.4** Ratio of speeds of a generator and motor

```
//Ratio of speeds of a generator and motor

clc;
clear;

V=250;
Il=80;
Ra=0.12;
Rf=100;// Field Resistance

Ish=V/Rf;// Field Current

Ia1=Il+Ish;// Machine Current genrated
Ia2=Il-Ish;// Motor Current taken by the motor

E=V+(Ia1*Ra);// Generator Induced EMF

Eb=V-(Ia2*Ra);// Motor Operating EMF

//Speeds are directly proportional to the EMFs

Nr=E/Eb; // Ratio of speeds

printf('The ratio of Generator speed to motor speed
    = %g \n',Nr)
```

**Scilab code Exa 6.5** Calculate Load Current in a shunt generator

```
// Calculate Load Current in a shunt generator

clc;
clear;

E=127;
V=120;
```

```
Ra=0.02;
Rf=15;// Field Resistance

Ish=V/Rf;

Ia=(E-V)/Ra;

Il=Ia-Ish;

printf('The load current = %g A \n',Il)
```

**Scilab code Exa 6.6** Useful Flux per pole on no load shunt motor

```
//Useful Flux per pole on no load shunt motor

clc;
clear;
V=250;
Z=2*110;// One turn is two conductors
Ia=13.3;
N=908;
Ra=0.2;
A=2; //Wave Winding
P=6;

Eb=V-(Ia*Ra);// Back EMF

phi=Eb*60*A/(N*Z*P);

printf('The useful flux per pole on no load of a 250
    V, 6 pole shunt motor = %g mWb \n',phi*1000)
```

**Scilab code Exa 6.7** To find efficiency and useful torque

```
//To find efficiency and useful torque

clc;
clear;

V=500;
N=600/60;// Rotation per second
I=18;
Hp=735.5; // The Value of one horse power is 735.5 W
Pd=10*Hp;// Power Output

Pi=V*I; // Power input

eff=Pd*100/Pi;// Efficiency

//Power ouput electrical = Power mechanical = Po= 2*
    pi*N*T
T=Pd/(2*%pi*N);

printf('The efficiency of the shunt motor = %g
    percent \n',eff)
printf('Useful torque = %g Nm \n',T)
```

**Scilab code Exa 6.8** Speed calculation of series motor

```
// Speed calculation of series motor

clc;
clear;

N1=800;// Speed in Case 1
Ra=1;
I=15;
V=200;
Rs=5; // Series resistance
```

```
Eb1=V-(I*Ra);// Back Emf in Case 1
Eb2=V-(I*(Ra+Rs));// Back Emf in case 2

//Speed directly proportional to EMF

N2=Eb2*N1/Eb1;

printf("The speed of the motor when connected in
    series to a resistance of 5 ohms = %g rpm \n",N2)
```

**Scilab code Exa 6.9** Parameters calculated due to armature reaction

```
//Parameters calculated due to armature reaction

clc;
clear;

P=8;
Z=960;
A=2;//Wave Winding

Ia=100;

cmti=Ia*Z/(2*A*P);// Total number of magnetising
    turns

deff('y=dm(x)','y=cmti*2*x/180')// Function to find
    out demagnetising ampere turns

// The demagnetising and cros magnetising ampere
    turns of the three cases respectively

DM1=ceil(dm(0));
CM1=cmti-DM1;
```

```
DM2=ceil(dm(10)); // To avoid decimal error ceil is
    used
CM2=cmti-DM2;


DM3=ceil(dm(10*4)); // Mechanical degree * no of
    pair of poles = Electrical degree
CM3=cmti-DM3;

printf('i) Brushes along GNP. \n Demagnetising and
    cross magnetising ampere turns are %g AT/pole and
     %g AT/pole respectively \n\n',DM1,CM1)

printf('ii) Brushes are shifted by 10 electrical
    degress. \n Demagnetising and cross magnetising
    ampere turns are %g AT/pole and %g AT/pole
    respectively \n\n',DM2,CM2)

printf('iii) Brushes are shifted by 10 mechanical
    degress. \n Demagnetising and cross magnetising
    ampere turns are %g AT/pole and %g AT/pole
    respectively \n\n',DM3,CM3)
```

**Scilab code Exa 6.10** Number of conductors of compensating winding

```
//Number of conductors of compensating winding

clc;
clear;

P=8; // No of Poles
Z=960; // No of conductors
ep=70/100; // Effective pole pitch
```

```
Zp=Z/P; // No os conductors per pole

AZp= ep*Zp; // No of actice armature conductors

Zpc=ceil(AZp/P); // Conductors in compensating
    winding

printf('The number of conductors of compensating
    winding = %g conductor/pole \n',Zpc)
```

**Scilab code Exa 6.11** Swinburne test on a dc shunt motor

```
//Swinburne test on a dc shunt motor

clc;
clear;

V=500;
I=5;
Rf=250;
Ra=0.5;

P=V*I;
If=V/Rf;

Ia=I-If;

Pfc=(If^2)*Rf;// Field Copper Loss

Pac=(Ia^2)*Ra; // Armature Copper Loss

Pil=P-Pfc-Pac;// Iron loss

// Generator
```

```
Vg=500;
Ig=100;

Pog=Vg*Ig; // Power Output

Iag=Ig+If; //Armature current

Pgac=(Iag^2)*Ra; // Armature Copper loss

slg=0.01*Pog;//stray loss

Pgtl=Pgac+Pfc+slg+Pil; // Total losses

effg=Pog*100/(Pog+Pgtl);

// Motor

Vm=500;
Im=100;

Pim=Vm*Im; // Power input to the motor

Iam=Ig-If; // Armature current

Pmac=(Iam^2)*Ra; // Armature Copper Loss

Pom=Pim-Pmac-Pil-Pfc;// Ouput of the motor
slm=0.01*Pom;// Stray loss

Pmtl=Pmac+Pil+Pfc+slm; // Total loss of the motor

effm=(Pom-slm)*100/(Pim);


printf('i) The Efficiency of the machine as a
    generator delivering 100A at 500V = %g percent \n
    ',effg)
```

```
printf('ii) The Efficiency of the machine as a motor
    having a line current 100A at 500V = %g percent
    \n',effm)
```

**Scilab code Exa 6.12** Speed of a belt driven shunt generator

```
//Speed of a belt driven shunt generator

clc;
clear;

Pg=100*(10^3);// Power output of the generator
Pm=10*(10^3); // Power input of the motor
V=220;
Ng=300; // Running speed of the generator
Ra=0.025; // Armature resistance
Rf=60; // Field Resistance
Vb=2;// Voltage drop due to brushes

Im=Pm/V; // Current taken by the motor
Ig=Pg/V; // Current delivered by the generator

Eg=V+(Ig*Ra)+Vb; // Induced EMF of generator

Eb=V-(Im*Ra)-Vb; // Back EMF of the motor

Nm=Eb*Ng/Eg;

printf('The Speed under motoring condition = %g rpm
    \n',Nm)
```

**Scilab code Exa 6.13** Voltage between feeder and bus bar in a series generator

```
//Voltage between feeder and bus bar in a series
   generator

clc;
clear;

V=50;
I=200;

Rf=0.3; // Feeder resistance

//Various Currents
I1=160;
I2=50;

deff('y=vol(x)','y=(x*Rf)-(V*x/I)') // Function to
   calculate the voltages

Va=vol(I1);
Vb=vol(I2);

printf('The voltage between the far end of the
   feeder and the bus bar at a current of \n')
printf('a) 160A = %g V \n',Va)
printf('b) 50A = %g V \n',Vb)
```

**Scilab code Exa 6.14** Induced EMF and Armature current in a long shunt compound generator

```
// Induced EMF and Armature current in a long shunt
   compound generator

```

```
clc;
clear;

Il=50; // Load Current
Vl=500; // Load Voltage
Ra=0.05; // Armature Resistance
Rfs=0.03; // Series Field Resistance
Rfp=250; // Shunt Field Resistance
Vb=2; // Contact drop

Ish=Vl/Rfp;

Ia=Il+Ish;

E=Vl+(Ia*(Ra+Rfs))+Vb; // Induced EMF

printf('The Induced EMF and Armature Current is %g V
    and %g A respectively \n',E,Ia)
```

**Scilab code Exa 6.15** Speed at 50A considering armature reaction of a shunt motor

```
//Speed at 50A considering armature reaction of a
   shunt motor

clc;
clear;

N=1000; // Speed at No load
I=5; // Current at no load
V=250;
Ra=0.2; // Armature Resistance
Rf=250;// Field Resistance

Ish=V/Rf; // Field Current
```

```

Ia=I-Ish; //Armature Current at no load
Eb=V-(Ia*Ra); // Back EMF at no load
Il=50; // Curent taken when loaded

Ebl=V-(Il-Ish)*Ra;// Back EMF when loaded

Nl=Ebl*N/(0.97*Eb)

printf('The Speed at 50A considering weakening of the field due to armature reaction = %g rpm \n', ceil(Nl) )
```

**Scilab code Exa 6.16** Speed of shunt motor taking 50kW input

```
// Speed of shunt motor taking 50kW input

clc;
clear;

Pog=50*(10^3);// Power ouput of the generator
Ng=400; // Speed of the generator
Vg=250;
Ra=0.02;
Rf=50;

Pim=50*(10^3); // Power Input of motor
Vm=250;

Vb=2;// Contact drop

// Generator

Ish=Vg/Rf; // Field Current
Ilg=Pog/Vg; // Load Current
```

```
Iag=Ish+Ilg; // Armature Current

Eg=Vg+(Iag*Ra)+Vb;

// Motor

Ilm=Pim/Vm; // Input Current
Ish=Vm/Rf; // Field Current
Iam=Ilm-Ish; // Armature Current

Eb=Vm-(Iam*Ra)-Vb; // Back EMF

Nm=Eb*Ng/Eg;

printf('The speed of shunt generator as a motor = %g
    rpm',ceil(Nm))
```

**Scilab code Exa 6.17** Useful torque and efficiency of a shunt motor

```
//Useful torque and efficiency of a shunt motor

clc;
clear;

Po=10.14*735; // 1 HP is 735 W, Power Developed
N=600/60;// Speed in rotations per sec
I=18;
V=500;
Pi=V*I; // Power input

eff=Po*100/Pi;

T=Po/(2*%pi*N);

printf('The Efficiency and the useful torque of the
```

```
    shunt motor are %g percent and %g Nm respectively
     \n',eff,T)
```

**Scilab code Exa 6.18** Total torque developed in a 4 pole shunt motor

```
//Total torque developed in a 4 pole shunt motor

clc;
clear;

P=4;
A=4;// Lap Winding
Z=60*20; // Slots * no of comductors in each slot
phi=23*(10^-3);
Ia=50; // Armature current

printf('The eqaution to find out torque is given by
    \n \n 2*pi*N*T = phi*Z*P*N/(60*A) * Ia \n\n')
T=((phi*Z*P*Ia)/(60*A))/(2*%pi/60);

printf('Total torque developed = %g Nm \n',T)
```

**Scilab code Exa 6.19** EMF and copper losses of a Shunt Motor

```
//EMF and copper losses of a Shunt Motor

clc;
clear;

V=250;
I=200;
Ra=0.02; // Armature Resistance
Rf=50; // Field Resistance
```

```
Pil=950; // Iron and frictional losses

Ish=V/Rf; // Field Current
Ia=Ish+I; // Armature Current

Pac=(Ia^2)*Ra; // Armature copper loss
Pfc=(Ish^2)*Rf;// Field copper loss

Pc=Pac+Pfc;

E=V+(Ia*Ra);

Prime=(V*I)+Pil+Pc;// Ouput of prime mover is the
    input to the generator

eff=(V*I*100)/Prime;

printf('i) The EMF generated = %g V \n',E)
printf('ii) Total Copper Loss = %g watts \n',Pc)
printf('iii) Output of the prime mover is %g watts
    and the efficiency is %g percent \n',Prime,eff)
```

**Scilab code Exa 6.20** Current taken by a motor at 90 percent efficiency

```
//Current taken by a motor at 90 percent efficiency

clc;
clear;

V=500;
N=400/60;// Speed in rotations per sec
eff=90/100;
T=195

Pd=2*%pi*N*T; // Power developed by the motor
```

```
Pi=Pd/eff; // Power input to the motor

I=Pi/V;

printf('The Current taken by the motor = %g A \n',I)
```

**Scilab code Exa 6.21** Rated torque calculation by resistance addition

```
//Rated torque calculation by resistance addition

clc;
clear;

V=240;
I=40;
Ra=0.3;
N1=1500/60;// speed in rotations per sec
N2=1000/60;

Pi=V*I;// Power input

Pc=(I^2)*Ra; // Copper loss

Po=Pi-Pc;

T=Po/(2*%pi*N1);// Rated torque

R1=V/I; // Back EMF is zero

Rex1=R1-Ra;

//Eb directly to N(speed)

Eb1500=V-(I*Ra);
```

```
Eb1000=N2*Eb1500/N1;

R=poly([0 1],'R','c');

x=(V-Eb1000)-(I*(R+Ra)); // Characteristic equation
    to find external resistance

Rex2=roots(x);

printf('The value of the resistance to be added to
    obtain rated torque \n')
printf('a) At starting = %g ohms\n',Rex1)
printf('b) At 1000 rpm = %g ohms\n',Rex2)
```

**Scilab code Exa 6.22** Efficiency at full load

```
//Efficiency at full load

clc;
clear;

V=400;
Inl=5; // No load current
Ra=0.5; // Armature Resistance
Rf=200; // Field Resistance
Ifl=40; // Full load current

Ish=V/Rf; //  Field Current

Psc=(Ish^2)*Rf; // Field copper loss

Prs=(V*Inl)-Psc; // Stray losses assuming no
    armature losses

Ia=Ifl-Ish; // Armature Current
```

```
Pc=(Ia^2)*Ra;// Armature copper loss

Pi=Ifl*V; // Power input

Po=Pi-Psc-Prs-Pc;

eff=Po*100/Pi;

printf('The efficiency at full load = %g percent \n'
   ,eff)
```

# Chapter 7

# Three Phase Synchronous Machines

**Scilab code Exa 7.1** Power delivered to 3 phase synchronous motor

```
//Power delivered to 3 phase synchronous motor

clc;
clear;

Vl=2300;
Il=8.8;
pf=0.8// Lead Power Factor
theta=acosd(pf)

P=sqrt(3)*Vl*Il; // Power delivered by the pump

I=P/(sqrt(3)*Vl*pf); // Increase in Current

Pr=sqrt(3)*Vl*I*sind(theta); // kVAr supplied

printf('The Power delivered by the pump = %g kW \n',
    P/1000)
printf('The Rheostat should be decreased such that
```

```
    the ammeter reads %g A \n',I)
printf('The kVAr supplied by the motor = %g kVAr',Pr
    /1000)
```

**Scilab code Exa 7.2** New plant pf and percent decrease in line current

```
//New plant pf and percent decrease in line current

clc;
clear;

Pmp=5000*(10^3); // Electrical load
pfmp=0.8; // Lag

Pim=500*735;// One horse power is 735W
Effim=96/100; // Efficiency of the motor
pfim=0.9; // Lag
pfsm=0.8; // Lead

Pime=Pim/Effim;// Effective power delivered by the
    induction motor

deff('x=com(y,z)','x=y+(%i*y*tand(acosd(z)))')//
    Function to find the complex powers

//Complex Powers
Pcmp=com(Pmp,pfmp); // Manufacturing Plant Load
Pcim=com(Pime,pfim);// Induction Motor
Pcsm=com(Pime,-pfsm);// Synchronous Machine, Minus
    Sign indicates Lead

Pr=Pcmp-Pcim+Pcsm; // Plant Requirement after
    replacement

pfar=real(Pr)/abs(Pr); // New Power Factor of the
```

```
plant

Pnp=abs(Pr);

Vl=poly([0 1],'Vl','c');

Io=Pmp/(pfmp*sqrt(3)*Vl);
In=Pnp/(sqrt(3)*Vl); // Improved Factor Value =1;

red=(Io-In)*100/Io; // Reduction percent in fractions

redeq=Vl-red;// Reduction percent in decimal characteristic equation

redper=roots(redeq(2));

printf('The New Power Factor of the plant = %g lag \n',pfar )
printf('The Percentage decrease in line current that will result in improved p.f = %g percent \n',redper)
```

**Scilab code Exa 7.3** kVAr rating of a synchronous condenser

```
//kVAr rating of a synchronous condenser

clc;
clear;

P=5000*(10^3); // Power delivered to the load
pfo=0.8;// Original Power Factor
pfn=0.9;// New Power Factor
Pcomo=P+%i*(P*tand(acosd(pfo)));//Original Complex Power
```

```
Pcomn=P+%i*(P*tand(acosd(pfn)));//New Complex Power

Psc=abs(imag(Pcomo-Pcomn)); // Difference in kVAr;

printf('The kVAr rating of the synchronous condenser to correct the original p.f to 0.9 = %g kVAr \n',Psc/1000)
```

**Scilab code Exa 7.4** Calculate E per phase and Current and pf

```
//Calculate E per phase and Current and pf

clc;
clear;

V=2300;
delta=20;
Pd=255*735.5; // Power delivered converted to W from HP
Xs=10;
eff=90/100; //Efficiency

P=Pd/eff;

E=poly([0 1],'E','c');

x=(sqrt(3)*E*V*sind(delta))-(P*Xs); // Characteristic Equation to find E

E=roots(x);

Vph=V/(sqrt(3));// Phase Voltage

I=((Vph*expm(%i*0))-(E*expm(%i*(-%pi/9))))/(%i*Xs); // Current Eqaution
```

```
[Im,phi]=polar(I); // Angle in radians and magnitude

phid=(abs(phi)/%pi)*180;// Power Factor Angle in
    Degrees

pf=cosd(phid);

// High Precision Answers
printf('a) E per phase = %g V \n',E)
disp('amperes',I,'b) I =')
printf('\n c) p.f = %g lead \n',pf)
```

**Scilab code Exa 7.5** Voltage Regulation of a 3 Phase alternator

```
//Voltage Regulation of a 3 Phase alternator

clc;
clear;

Ra=0.093;
Xs=8.5;
Z=(Ra+(%i*Xs)); // Total Impedance

P=1500*(10^3); // Power delivered at full load
V=6.6*(10^3); // Voltage per line
Vph=V/(sqrt(3)); // Voltage per phase

Il=P/(sqrt(3)*V); // Full Load Current


// Taking voltage as reference
//Power Angles
theta1=-acos(0.8); // Negative Sign as It is lagging
theta2=acos(0.8);
```

```
deff('a=pot(b)','a=Vph+((Il*expm(%i*b))*Z)')//
    Function to find out the output phase voltage

E1=pot(theta1);
E2=pot(theta2);

deff('y=vg(x)','y=(abs(x)-Vph)*100/Vph') // Function
     to find out the voltage regulation using the
    formuala

Vreg1=vg(E1);
Vreg2=vg(E2);

printf('The Voltage regulation of a 3-Phase 1500 kVA
    , 6.6 kV alternator at \n')
printf('i) 0.8 lag = %g percent \n',Vreg1)
printf('ii) 0.8 lead = %g percent \n',Vreg2)
```

**Scilab code Exa 7.6** Internal Voltage drop in an alternator

```
// Internal Voltage drop in an alternator

clc;
clear;

If=10;
Voc=900; // Open Circuit Voltage

Isc=150; // Short Circuit Current

Zs=Voc/Isc;

I=60; // Load current

```

```
Vd=I*Zs; // Internal Voltage Drop

printf('The internal voltage drop with a load
    current of 60 A = %g V \n',Vd)
```

# Chapter 8

# Three Phase Induction Motors

**Scilab code Exa 8.1** Find the percentage slip and poles of the motor

```
//Find the percentage slip and poles of the motor

clc;
clear;

p=12;
n=500;
nlim=1440;
f=p*n/120;
c=1;

while(c>0) // Used to find out the poles of the
    motor nearest to the full load slip
    P=2*c;
    N=120*f/P;
    g=(5/100)*N;
    if((N-nlim)>(0.05*N))
        c=c+1;
    else
        c=0;
    end
```

```
end

slip=(N-nlim)*100/N;

printf('The Number of poles of the induction motor is %g \n',P)
printf('The percentage slip is %g percent \n',slip)
```

**Scilab code Exa 8.2** To calculate motor speed and its slip

```
//To calculate motor speed and its slip

clc;
clear;

f=50;
sf=3/2;
s=sf/f;

p=8;
N=120*f/8;

Nr=poly([0 1],'Nr','c'); // Actual Speed Variable

x=(750*s)-(750-Nr); // Equation To find the Actual Speed

Nr=roots(x); // Actual Speed Constant

printf('The motor runs at a speed of %g rpm and has a slip of %g \n',ceil(Nr),s)
```

**Scilab code Exa 8.3** To Calculate Parameters of a 3 phase 4 pole induction machine

```
// To Calculate Parameters of a 3 phase 4 pole induction machine.

clc;
clear;

V1=200;
R2=0.1;
X2=0.9;
f=50;
p=4;
s=4/100;
a=0.67; // The Ratio of rotor to stator turns

P=((a*V1)^2)*R2*(1-s)*s/(((R2)^2)+((s*X2)^2)); // Power Delivered referred to the rotor side (Mechanical Power)

N=120*f/p;// Rated Speed

N1=N*(1-s);// Speed at 4% slip

T4=P*60/(2*%pi*N1); // Total Torque at 4% slip

sm=floor((R2/X2)*1000)/1000; // Condition for Maximum Torque

Pmax=((a*V1)^2)*R2*(1-sm)*sm/((R2^2)+((sm*X2)^2));// Power at maximum torque

Nmax=ceil(N*(1-sm)); // Speed at Maximum Torque

Tmax=Pmax*60/(2*%pi*Nmax); // Maixmum Torque

// Please Note that the answers are accurate and no
```

```
   quantities are neglected as in the text book.
printf('a) Total torque at 4 percent slip = %g Nm \n
   ',T4)
printf('b) Total Mechanical Power at 4 percent slip
   = %g watts or %g H.P \n',P,(P/735))
printf('c) Maximum Torque = %g Nm \n',Tmax)
printf('d) Speed at maximum torque = %g rpm \n',Nmax
   )
printf('e) Maximum Mechanical Power = %g watt or %g
   H.P \n',Pmax,(Pmax/735))
```

**Scilab code Exa 8.4** Calculation of slip from losses

```
//Calculation of slip from losses

clc;
clear;

eff=0.9; // Efficiency
P=50*735; // Load in watts
x=poly([0 1],'x','c'); // Rotor Copper Loss Variable

tx=(x+x+x+(x/3)); // Total loss

loss=((P+tx)*eff)-P; // Equation to calculate x

x=roots(loss); // Rotor Copper Loss Constant

s=poly([0 1],'s','c'); // Variable for slip
slip=(P*s)-(x*(1-s));// Gives the variable equation
   of slip

s=roots(slip); // Numerical Value of slip

printf('The slip of an induction motor of 0.9
```

```
efficiency at 50 HP load = %g \n',s)
```

**Scilab code Exa 8.5** apping of an auto transformer to limit current in squirrel cage motor

```
//Tapping of an auto transformer to limit current in
    squirrel cage motor

clc;
clear;

V=400; // Line to line voltage
Vph=V/sqrt(3); // Phase voltage
Z=1.54; // Standstill impedance
Ifl=30;// Full Load Current
Imax=75; // Max current which can be taken by the
   line
s=4/100; // Full load slip

t=poly([0 1],'t','c'); // Variable for tapping
   percent of normal voltage

Is=t*(Vph/(100*Z)); // Starting current in the motor

Ias=(t/100)*Is; // Current on supply side of the
   auto transformer

Tap=Ias-Imax; // Equation to find t

t=roots(Tap);// Numerical Value for t

if(imag(sqrt(t(1))))
    t=t(2);
else
    t=t(1);
```

```
end


Ism=Imax*100/t; // Starting current in the motor (
    Numerical Value)

st=((Ism/Ifl)^2)*s; // Starting torque to full load
    torque ratio

printf('The tapping provided to the auto transformer
     = %g percent of Normal Voltage \n',t)

printf('The starting torque available is %g times
    the full load torque \n',st)
```

---

**Scilab code Exa 8.6** To find the total mechannical power and rotor copper loss

```
//To find the total mechannical power and rotor
    copper loss

clc;
clear;

P=60*(10^3); // Power input
Pstat=1*(10^3);// Stator Losses
Pirot=P-Pstat;// Rotor Input
s=3/100;// Running slip

Prc=poly([0 1],'Prc','c'); // Variable for rotor
    copper loss
// Prc = I2^2 * R2

rloss=(Pirot*s)-(3*Prc);

```

```
Prc=roots(rloss); // Numerical Value of rotor loss
   per phase;

Pm=Pirot-(3*Prc); // Mechanical power developed

printf('The rotor loss per phase = %g W \n',Prc)
printf('The Mechanical Power developed = %g kW \n',
   Pm/1000)
```

**Scilab code Exa 8.7** To determine the starting torque and current using different starters

```
// To determine the starting torque and current
   using different starters

clc;
clear;

// Rated Parameters of the motor
V=400; // Delta connected
P=50*735.5; // Power developed
N=750; // Speed

Ifl=50; // Full Load current
Z=2.5; // Impedance per phase
sf=4.5/100; // Slip
f=50;

Tfl=P*60/(2*%pi*N);

deff('y=curr(x)','y=(sqrt(3))*x/Z');

deff('a=stor(b)','a=((b/Ifl)^2)*sf*Tfl');

//Case 1
```

```
I1=curr(V);
T1=stor(I1);

//Case 2
I3=curr(70*V/100);
T3=stor(I3);

T2=Tfl*((1/sqrt(3))^2); // Case 2 torque
I2=I1;

printf('The starting torque and the starting current using different starters are : \n')

printf('i) D.O.L starter = %g Nm and %g A \n',T1,I1)
printf('ii) Star-delta starter = %g Nm and %g A \n',T2,I2)
printf('iii) An auto transformer starter with 70 percent tapping = %g Nm and %g A \n',T3,I3)
```

**Scilab code Exa 8.8** To actual rotor speed and the rotor frequency at 3 percent slip

```
// To actual rotor speed and the rotor frequency at 3 percent slip

clc;
clear;

P=2;
f=50;
V=400;
Vph=V/sqrt(3);
s=3/100;

Ns=120*f/P;
```

```

Nr=Ns*(1-s);

rf=s*f; // Rotor Frequency

printf('The Actual rotor speed = %g rpm \n',Nr)
printf('The rotor frequency = %g Hz \n',rf)
```

**Scilab code Exa 8.9** To determine the various parameters of a 3 phase 400V 6 poles Induction Motor

```
// To determine the various parameters of a 3 phase
   400V 6 poles Induction Motor

clc;
clear;

f=50;
p=6;
s=3/100;
V=400;

N=120*f/p; // Synchronous speed
Ns=0; // Speed of stator
rf=s*f; // Rotor Frequency
Nr=N*(1-s); // Rotor speed
Nrs=N-Ns; // Speed of Rotor field wrt stator
Nrr=120*rf/6; // Speed of rotor field wrt rotor
Nrmsm=0; // Speed of rotor field wrt stator field

printf('i) The speed of the rotor = %g rpm \n',Nr)
printf('ii) The frequency of rotor current = %g Hz \
   n',rf)
printf('iii) The Speed of the rotor magnetic field w
   .r.t the stator = %g rpm \n',Nrs)
```

```
printf('iv) The speed of the rotor magnetic field w.
    r.t the rotor = %g rpm \n',Nrr)
printf('v) The speed of the rotor magnetic field w.r
    .t the stator magnetic field = %g rpm \n',Nrmsm)
```

**Scilab code Exa 8.10** To determine parameters of an 3 phase delta connected 4 pole induction motor

```
//To determine parameters of an 3 phase delta
    connected 4 pole induction motor

clc;
clear;

V=440;
f=50;
p=4;
E2=150;
Nr=1450;

Ns=120*f/p;

s=(Ns-Nr)/Ns;

rf=s*f; // Rotor frequency

Eir=s*E2; // Rotor induced EMF per phase
gc=1;

// To find the GCD of both the voltages
for i=E2:-1:1
    a=modulo(V,i);
    b=modulo(E2,i);
    if(a==0&b==0)
        gc=gc*i;
```

```
        break;
    end
end

printf('The parameters for a motor running at 1450
    rpm are \n')
printf('i) The slip = %g percent \n',s)
printf('ii) The frequency of the rotor induced EMF =
    %g Hz \n',rf)
printf('iii) The rotor induced EMF per phase = %g V
    \n',Eir)
printf('iv) Stator to rotor turn ratio = %g : %g \n'
    ,V/gc,E2/gc)
```

**Scilab code Exa 8.11** Determine the shaft power of 6 pole Induction Motor

```
// Determine the shaft power of 6 pole Induction
    Motor

clc;
clear;

f=50;
p=6;
rf=120/60; // Rotor Frequency
T=150; // Full Load torque

s=rf/f;

Ns=120*f/p;

Nr=Ns*(1-s);

Ps=2*%pi*Nr*T/60; // Shaft power
```

```
printf('The shaft power of the motor = %g kW \n',Ps
   /1000)
```

**Scilab code Exa 8.12** Motor parameters at a load power factor

```
// Motor parameters at a load power factor

clc;
clear;

p=4;
f=50;
V=400;
pf=0.8;
Nr=1440;
Pm=20*(10^3); // Mechanical Power Developed
Ns=120*f/p;
s=(Ns-Nr)/Ns;

rf=s*f; // Rotor frequency

Pstat=1000; //Stator Loss

// Power input to the rotor = Mechanical Power
   Developed / (1-s)

Pirot=Pm/(1-s); // Rotor Power Input

Pi=Pirot+Pstat; // Power input to stator

Il=Pi/(sqrt(3)*V*pf); // Line Current

printf('For a 4 pole motor running at 1440 rpm and
   0.8 p.f \n')
```

```
printf('i) Rotor Current frequency = %g Hz \n',rf)
printf('ii) Total input if stator loss is 1000W = %g
    kW \n',Pi/1000)
printf('iii) The line current = %g A \n',Il)
```

**Scilab code Exa 8.13** To determine the auto tranformer ratio and starting torque

```
// To determine the auto tranformer ratio and
   starting torque

clc;
clear;

V=400;
f=50;
p=4;
sfl=4/100;

Ria=2.5; // Ratio of starting current to full load
   current (Auto transformer)
Rir=4; // Ratio of starting current to full load
   current ( For the Rated Voltage)

x=sqrt(Ria/Rir);

Rt=((x*Rir)^2)*sfl; // Ratio of starting torque to
   full load torque;

printf('The auto-transformer ratio = %g \n',x)
printf('The starting torque at the above transformer
    ratio = %g percent of full load torque \n',100*
   Rt)
```

**Scilab code Exa 8.14** To determine the starting torque in terms of full load torque

```
//To determine the starting torque in terms of full
   load torque

clc;
clear;

sfl=4/100;

Rir=5; // Ratio of starting current to the full load
    current at rated voltage

x=70.7/100; // Auto transformer tapping

Rsd=((Rir)^2)*sfl/3; // Ratio of the starting load
   to full load torque for a star -delta starter

Ra=((x*Rir)^2)*sfl; // Ratio of the starting load to
    full load torque for an 70.7% tapped auto
   transformer

printf('The starting torque in terms of full load
   torque by \n')
printf('i) Star-Delta starter = %g Tfl \n',Rsd)
printf('ii) An auto-tranformer starter with 70.7
   percent tapping = %g Tfl \n',Ra)
```

**Scilab code Exa 8.15** Stator input of 3 phase 4 pole induction motor

```
// Stator input of 3 phase 4 pole induction motor
```

```

clc;
clear;

p=4;
f=50;
Pd=4000; // Power Developed
Nr=1440;
Ps=320;// Stator loss

Ns=120*f/p;

s=(Ns-Nr)/Ns;

Pir=Pd/(1-s); // Power to the rotor

Pi=Pir+Ps; // The input to the stator

printf('The stator input of a 440V 3 phase 4 pole
    induction motor = %g W \n',ceil(Pi))
```

**Scilab code Exa 8.16** Motor parameters of a 6 pole motor with 40 hp mechanical power

```
// Motor parameters of a 6 pole motor with 40 hp
    mechanical power

clc;
clear;

f=50;
p=6;
Pd=40*735.5; // Mechanical Power developed
V=500;
Nr=960;
```

```
pf= 0.8; // Lag
Pm=1500; // Mechanical Loss

Ns=120*f/p;

s=(Ns-Nr)/Ns;
Ps=1800; // Stator Loss

Po=Pd-Pm; // Power Output

Pir=Pd/(1-s); // Power input to rotor

Prc=s*Pir; // Copper Loss of the Rotor

Pi=Pir+Ps; // Power input to the stator

eff=Po*100/Pi;

Il=Pi/(sqrt(3)*V*pf);// Line Current

printf('For a 6 pole 3 phase motor at 500V with a
   power factor of 0.8 lag \n')
printf('i) Rotor Copper Loss = %g W \n',Prc)
printf('ii) Total input to stator if the stator loss
    is 1500W = %g W \n',Pi)
printf('iii) The line Current = %g A \n',Il)
printf('iv) Efficiency = %g percent \n',eff)
```

**Scilab code Exa 8.17** To determine parameters of 4 pole induction motor considering circuit parameters

```
//To determine parameters of 4 pole induction motor
   considering circuit parameters

clc;
```

```
clear;

R1=0.5;
R2=0.35;
X1=1.2;
X2=X1;
Xm=25;
f=50;
p=4;

Pd=25*735.5; // Power Developed
Prl=800;// Rotational Losses
V=400;
Vph=V/sqrt(3);

Ns=120*f/p;
s=2.5/100;
Nr=(1-s)*Ns;
rf=s*f; // Rotor Frequency

Z1=R1+(%i*X1);
Z2=(R2/s)+(%i*X2);
Zm=%i*Xm;

Z2m=(Zm*Z2)/(Zm+Z2);

Zeff=Z1+Z2m; // Effective Impedance

Is= Vph/Zeff; // Stator Current

Psc= 3*(abs(Is)^2)*R1; // Copper Loss in the Stator
Ztheta= atand(imag(Zeff)/real(Zeff)); // Phase angle
    of impedance
Ctheta= atand(imag(Is)/real(Is)); // Phase angle of
   current

pf= cosd(Ctheta);// Lagging Power Factor
Ir=Is*(Zm/(Zm+Z2));// Rotor Current
```

```
Prc= 3*(abs(Ir)^2)*R2; // Rotor Copper Loss

Pim= sqrt(3)*V*abs(Is)*cosd(Ctheta); // Power input to the motor
Pom= Pim-Prc-Psc-Prl; // Power Output to the motor

eff=Pom*100/Pim; // Efficiency

printf('For a rotor slip of 2.5 percent at rated voltage and frequency \n');
printf('i) The motor speed = %g rpm \n',Nr)
printf('ii) The stator Current = %g /_%g A \n',abs(Is),Ctheta)
printf('iii) The p.f = %g lagging \n',pf)
printf('iv) The efficiency = %g percent \n',eff)
```

**Scilab code Exa 8.18** Stator Current and pf and efficiency of a motor operating at rated slip

```
//Stator Current and pf and efficiency of a motor operating at 0.03 slip

clc;
clear;

V=400;
Vph=V/sqrt(3);
R1=0.2;
R2=0.15;
X1=%i*0.5;
X2=%i*0.3;

s=3/100;

```

```
Ptl=2000; // Total Losses

Z1=R1+X1;
Z2=(R2/s)+X2;
Zt=Z1+Z2; // Total Impedance of the circuit

Is= Vph/Zt; // Stator Current

Ctheta=atand(imag(Is)/real(Is)); // Phase angle of
    stator current

pf= cosd(Ctheta); // Power factor lagging

Pi=sqrt(3)*V*abs(Is)*cosd(Ctheta);

Po=Pi-Ptl; // Power Output

eff=Po*100/Pi;

printf('For a 3 phase, 4 pole, 400V Induction Motor
    operating at 3 percent slip \n')
printf('i) The Stator current = %g /_%g A \n',abs(Is
    ),Ctheta)
printf('ii) The p.f = %g lagging \n',pf)
printf('iii) The efficiency = %g percent \n',eff)
```

# Chapter 10

# Power System

**Scilab code Exa 10.1** Determine the additional load which can be supplied

```
//Determine the additional load which can be
   supplied

clc;
clear;

printf('a) D.C two wire: \nPower transmitted by DC
   two wire = P and Voltage between the wires = V\n'
   )
printf('Copper Loss = 2 *(P/V)^2)*R ; where R is the
    resistance of each wire\n')
printf('Per Unit Loss = 2*P*R/(V^2)\n\n')
printf('b) 3 phase 3 wire: \nPower transmitted = P''
   \n ')
printf('I'' = P''/(sqrt(3)*V) for unity p.f\n')
printf('Copper Loss = 3*((P''/(sqrt(3)*V))^2)*R = ((
   P/V)^2)*R\n')
printf('Per Unit Loss = P''*R/(V^2)\n\n')
printf('Equating the per unit loss we have\n')
printf('2*P*R/(V^2) = P''*R/(V^2) or P'' = 2P\n
```

```
Which proves that 100 %% aditional pwer can be
supplied by 3 phase 3 wire system\n')
```

**Scilab code Exa 10.2** Pf at which the slow machine will work

```
//Pf at which the slow machine will work

clc;
clear;

pf1=0.8;//Power Factor Lag (Combined)
Pa=2500*(10^3); // Combined Power
Pr=Pa*(tand(acosd(pf1))); // Combined VAr
Pat=1000*(10^3); // Active power of the turbo
    alternator unity pf so no VAr
Prt=0;
Pas=Pa-Pat; // Active power of the slow speed
    alternator
Prs=Pr-Prt; // Reactive power of the slow speed
    alternator

theta=atand(Prs/Pas); // Power Factor Angle

powfac=cosd(theta); // Power factor

printf('The Power Factor of the slow speed
    alternator is %g\n',powfac)
```

**Scilab code Exa 10.3** Detemine the load and pf of the other machine

```
//Detemine the load and pf of the other machine

clc;
```

```
clear;

Pa=3000*(10^3);// Lighting load
Pma=5000*(10^3); // Aggregate Motor load
pfm=0.71; // power factor of motor load

P1a=5000*(10^3); // One Machine load
pf1=0.8;// Power factor machine 1 (lagging)

Pta=Pa+Pma; // Total load active power requirement

// Reactive power

Pr=0; // Lighting
Pmr=Pma*tand(acosd(pfm)); // Motor
P1r=P1a*tand(acosd(pf1)); // Machine 1

P2a=Pta-P1a; // Active power by other machine
P2r=Pr+Pmr-P1r; // Reactive power by other machine

pf2=cosd(atand(P2r/P2a)); // Power factor of other
    machine

printf('The other machine supplies:\n')
printf('     A load of %g kw at a p.f of %g\n',P2a
    /1000,pf2)
```

**Scilab code Exa 10.4** Determine the value of a shunt capacitor

```
//Determine the value of a shunt capacitor

clc;
clear;

V=440; // Line to line voltage
```

```
f=50; // Frequency of operation
w=2*%pi*f; // Angular frequency
Vph= V/sqrt(3); // Phase voltage
I=40; // Magnitude of current
pfi=0.7; // Lagging power factor of the current
ti=acosd(0.7);

//Iv=I*(expm(%i*-1*%pi*ti/180));

// For pf = 0.7
Pa=Vph*I*pfi; // Active power
Pr=Vph*I*sind(ti); // Reactive power

// To gain a pf of 0.9
pfn=0.9;
Pnr=Pa*tand(acosd(pfn)); // Reactive power at pf of
    0.9

PRC=Pr-Pnr; // VArs supplied from the capacitor

C=PRC/((Vph^2)*w); // Capacitance required to meet
    the condition

printf('The value of the shunt capacitor should
    raise the pf to 0.9 = %g mF\n',C*1000)
```

**Scilab code Exa 10.5** Calculate the inductance of a choke to enable the lamp

```
// Calculate the inductance of a choke to enable the
    lamp

clc;
clear;

```

```
P=500; // Power Rating of a discharge lamp
I=4; // Current drawn by the lamp
w=2*%pi*50; // Angular frequency

Vdl=P/I; // Supply voltage for the proper working of the lamp
V=250; // AC supply voltage

//According the Voltage triangle

Vil=sqrt((V^2)-(Vdl^2)); // Voltage drop across inductor

Xl=Vil/I; //Reactance
L=Xl/w; // Inductor

Prl=(I^2)*Xl; // Reactive power requirement of the coke

C=Prl/((V^2)*w); // Capacitor supplying the necessary reactive power

printf('The inductance that should be connected in series with the lamp to make it work = %g mH\n',L*1000)
printf('The capacitor that should be connected in paralle to make the power factor unity = %g mF \n',C*1000)
```

# Chapter 11

# Domestic Wiring

**Scilab code Exa 11.1** Determine the size of the conductor for power and lighting circuit

```
// Determine the size of the conductor for power and
    lighting circuit

clc;
clear;

P5=3*800; // Load wattage in 5A circuit
P15=2*1500;// Load wattage in 15 A circuit

Pt=P5+P15; // Total Load

// Assume a average of 0.8 pf, common for
   distribution systems

pf=0.8;

V=230; // Supply voltage

```

| Size of conductor | | 2 cables d.c. or single phase a.c. | | 3 or 4 cables or balanced three phase | | Four cables d.c. or single phase a.c. | |
|---|---|---|---|---|---|---|---|
| *Nomial area in Sq. mm* | *No. and diameter of wire in mm.* | *Current rating in Amperes* | *Approx. length of run for one volt. (M.)* | *Current rating in Amps.* | *Approx. run for one volt drop in (Mt.)* | *Current rating (amps.)* | *Approx. run for one volt drop in Mts.* |
| 1.5 | 1/1.40 | 10 | 2.3 | 9 | 2.9 | 9 | 2.5 |
| 2.5 | 1/1.80 | 15 | 2.5 | 12 | 3.6 | 11 | 3.4 |
| 4.0 | 1/2.24 | 20 | 2.9 | 17 | 3.9 | 15 | 4.4 |
| 6.0 | 1/2.80 | 27 | 3.4 | 24 | 4.3 | 21 | 4.3 |
| 10.0 | 1/3.55 | 34 | 4.3 | 31 | 5.4 | 27 | 5.4 |
| 16.0 | 7/1.70 | 43 | 5.4 | 38 | 7.0 | 35 | 6.8 |
| 25.0 | 7/2.24 | 59 | 6.8 | 54 | 8.5 | 48 | 8.5 |
| 35.0 | 7/2.50 | 69 | 7.2 | 62 | 9.3 | 55 | 9.0 |
| 50.0 | 7/3.50<br>19/1.80 | 91 | 7.9 | 82 | 10.1 | 69 | 10.0 |
| 70.0 | 19/2.24 | 134 | 8.0 | 131 | 9.5 | — | — |
| 95.5 | 19/2.50 | 153 | 8.8 | 152 | 10.1 | — | — |

Figure 11.1: Determine the size of the conductor for power and lighting circuit

| *Size of conductor* | | *2 cables d.c. or single phase a.c.* | | *3 or 4 cables or balanced three phase* | | *Four cables d.c. or single phase a.c.* | |
|---|---|---|---|---|---|---|---|
| *Nomial area in Sq. mm* | *No. and diameter of wire in mm.* | *Current rating in Amperes* | *Approx. length of run for one volt. (M.)* | *Current rating in Amps.* | *Approx. run for one volt drop in (Mt.)* | *Current rating (amps.)* | *Approx. run for one volt drop in Mts.* |
| 1.5 | 1/1.40 | 10 | 2.3 | 9 | 2.9 | 9 | 2.5 |
| 2.5 | 1/1.80 | 15 | 2.5 | 12 | 3.6 | 11 | 3.4 |
| 4.0 | 1/2.24 | 20 | 2.9 | 17 | 3.9 | 15 | 4.4 |
| 6.0 | 1/2.80 | 27 | 3.4 | 24 | 4.3 | 21 | 4.3 |
| 10.0 | 1/3.55 | 34 | 4.3 | 31 | 5.4 | 27 | 5.4 |
| 16.0 | 7/1.70 | 43 | 5.4 | 38 | 7.0 | 35 | 6.8 |
| 25.0 | 7/2.24 | 59 | 6.8 | 54 | 8.5 | 48 | 8.5 |
| 35.0 | 7/2.50 | 69 | 7.2 | 62 | 9.3 | 55 | 9.0 |
| 50.0 | 7/3.50<br>19/1.80 | 91 | 7.9 | 82 | 10.1 | 69 | 10.0 |
| 70.0 | 19/2.24 | 134 | 8.0 | 131 | 9.5 | — | — |
| 95.5 | 19/2.50 | 153 | 8.8 | 152 | 10.1 | — | — |

Figure 11.2: Determine the size of the conductor at 25 m distance

```
I=Pt/(V*pf); // Current at 230 supply

Isc=1.5*I; // Short Circuit Current

printf('The Current is %g A and the short circuit
    current is %g A\n\n',I,Isc)
printf('From the result sheet provided along with
    this code,\n for aluminium wire the size of the
    conductor comes out to be 25 mm^2.\nIn fact for
    43 A it is 16 mm^2 but we should always go for
    one higher size of the conductor\n and hence we
    select conductor of size 25 mm^2 or 7/2.24 mm.')
printf('\n \n Refer the table in the result sheet \n
    ')
```

**Scilab code Exa 11.2** Determine the size of the conductor at 25 m distance

```
//Determine the size of the conductor at 25 m distance

clc;
clear;

V=230; // Supply Voltage
d=25; // Distance between mains and residence
I=5; // Supply current

pvd=1+((2/100)*V); // Permissible Voltage drop

// From the table given in the result sheet along with this code, Minimum size of wire for 10A

A=1.5*(10^-6);

dm=2.3;
Vd=d/dm; // Voltage drop at 10A

Vd5=Vd/2; // Voltage drop at 5A

//According to the table (Refer below) Permissible drop is 5.6 V

printf('The pemissible voltage drop = %g V\n',pvd)
printf('The voltage drop at 5 A = %g V\n',Vd5)
printf('As the permissible drop is 5.6 volts \nand the conductor with 1.5 mm^2 section gives \nvoltage drop of 5.4 volts hence the suitable size is 1/1.40 mm.')
printf('\n \n Refer the table in the result sheet \n
```

| Size of conductor | | 2 cables d.c. or single phase a.c. | | 3 or 4 cables or balanced three phase | | Four cables d.c. or single phase a.c. | |
|---|---|---|---|---|---|---|---|
| Nomial area in Sq. mm | No. and diameter of wire in mm. | Current rating in Amperes | Approx. length of run for one volt. (M.) | Current rating in Amps. | Approx. run for one volt drop in (Mt.) | Current rating (amps.) | Approx. run for one volt drop in Mts. |
| 1.5 | 1/1.40 | 10 | 2.3 | 9 | 2.9 | 9 | 2.5 |
| 2.5 | 1/1.80 | 15 | 2.5 | 12 | 3.6 | 11 | 3.4 |
| 4.0 | 1/2.24 | 20 | 2.9 | 17 | 3.9 | 15 | 4.4 |
| 6.0 | 1/2.80 | 27 | 3.4 | 24 | 4.3 | 21 | 4.3 |
| 10.0 | 1/3.55 | 34 | 4.3 | 31 | 5.4 | 27 | 5.4 |
| 16.0 | 7/1.70 | 43 | 5.4 | 38 | 7.0 | 35 | 6.8 |
| 25.0 | 7/2.24 | 59 | 6.8 | 54 | 8.5 | 48 | 8.5 |
| 35.0 | 7/2.50 | 69 | 7.2 | 62 | 9.3 | 55 | 9.0 |
| 50.0 | 7/3.50<br>19/1.80 | 91 | 7.9 | 82 | 10.1 | 69 | 10.0 |
| 70.0 | 19/2.24 | 134 | 8.0 | 131 | 9.5 | — | — |
| 95.5 | 19/2.50 | 153 | 8.8 | 152 | 10.1 | — | — |

Figure 11.3: Size of conductor to be used for wiring a 10 kW 400V 3 Phase induction motor

```
')
```

**Scilab code Exa 11.3** Size of conductor to be used for wiring a 10 kW 400V 3 Phase induction motor

```
// Size of conductor to be used for wiring a 10 kW
   400V 3 Phase induction motor

clc;
clear;

P=10*(10^3); // Power Rating
```

```
V=400; // Voltage Rating
f=50;// Frequency of operation

// Assumptions Made
eff=85/100; // Efficiency
pf=0.8// Power Factor

I=P/(sqrt(3)*V*eff*pf); // Current flowing in the
    conductor

Is=2*I; //At the time of starting the induction
    motor may take 2 times the rated current

printf('The Rated Current = %g A\n\n',I)
printf(' At the time of starting the induction motor
     may take\n 2 times the rated current and hence
    starting current = %g A \n\n',Is)

printf(' From the table corresponding to 42.4 A that
     is 43 A\n the wire used is 7 /1. 7 mm or 16 mm
    ^2.\n')
```

# Chapter 12

# DC Circuits

**Scilab code Exa 12.1** To Compute the number of electrons

```
// To Compute the number of electrons.

clc;
clear;

I=(25)*(10^-3);
t=(30)*(10^-3);
C=I*t;
// 1C = 6.242*(10^18)
n= 6.242*(10^18);
e_s=C*n;
disp(e_s,'The Number Of Electrons passing through
    the person is')
```

---

**Scilab code Exa 12.2** Computing the Average lighting current

```
// Compute Average lighting current
clc;
```

```
clear;

q=20;
t=(10)*(10^-3);

// Coulomb's Law

I=q/t;

disp('amperes',I,'The Average Lightning current =')
```

**Scilab code Exa 12.3** To Calculate the average voltage

```
// To Calculate the average voltage.
clc;
clear;

W=500;
I=40;
t=15*(10^-3);

V=W/(I*t);

disp('volts',V,'The Average volatage across the
    terminals of the device =')
```

**Scilab code Exa 12.4** Calculating resistance

```
// Calculating resistance.
clc;
clear;

```

```
L=2.5*(10^-2); // Length of rectangular cross-section.
B=0.05*(10^-2);// Breadth of rectangular cross-section.
A=L*B;

l=1*(10^3);

p=1.724*(10^-8);

R=p*l/A;

disp('ohms',R,'The Resistance of the copper strip =')
```

**Scilab code Exa 12.5** Current Calculation using ohms law

```
//Current Calculation using ohm's law.
clc;
clear;

V=220;
R=80;

// Using Ohm's Law V=I*R

I=V/R;

disp('amperes',I,'The Load Current =')
```

**Scilab code Exa 12.6** Determination of conductance in a short circuit

```
// Determination of conductance of a short circuit
```

```
clc;
clear;

V=120;
I=500;

G=I/V;

disp('siemens',G,'The Conductance =')
```

**Scilab code Exa 12.7** Power Rating Calculation

```
// Power Rating Calculation

clc;
clear;

V=250;
I=15;

// Power Equation or Watt's Law P=V*I.

P=V*I;

disp('watts',P,'The power rating of the device =')
```

**Scilab code Exa 12.8** To calculate current ratings and maximum voltage of a rated resistor

```
//To calculate current ratings and maximum voltage
   of a rated resistor.

```

```
clc;
clear;

P=1;
R=10*(10^3);

// Using Power Equation and Ohm's Law.

V=sqrt(P*R);

I=sqrt(P/R);

disp('volts',V,'The Maximum voltage of the resistor =')

disp('amperes',I,'The Current rating of the resistor =')
```

**Scilab code Exa 12.9** Determine the output power of the motor

```
//Determine the output of the motor.

clc;
clear;

eff=80/100;

V=220;
I=8;

// Power Equation P=V*I

P=V*I;// Input Power

Pout=eff*P;// Output Power
```

```
disp('watts',Pout,'The output power of the motor =')
```

**Scilab code Exa 12.10** Calculation of Current and power dissipated in resistors connected in series

```
//Calculation of Current and power dissipated in
   resistors connected in series.

clc;
clear;

R1=100;
R2=200;
R3=300;

Rt=R1+R2+R3;

V=250;

//Ohm's Law V=I*R

I=V/Rt;

// Power Loss Equation P=(I^2)*R

P1=(I^2)*R1;
P2=(I^2)*R2;
P3=(I^2)*R3;

Pt=P1+P2+P3;

P=V*I;

disp('ohms',Rt,'The total resistance in the circuit
```

```
    =')
disp('amperes',I,'The Current in the circuit =')
disp('watts',P1,'The power loss in the 100 ohms
    resistor =')
disp('watts',P2,'The power loss in the 200 ohms
    resistor =')
disp('watts',P3,'The power loss in the 300 ohms
    resistor =')
disp('watts',Pt,'The total power loss in the circuit
    =')
disp('watts',P,'The power loss in the circuit (using
    P=V*I ) =')
```

**Scilab code Exa 12.11** To find the value of the unknown resitance in the series of resistances in a circuit

```
// To find the value of the unknown resitance in the
    series of resistances in a circuit.

clc;
clear;

R1=20;

V=220;

P=50;

R=poly([0 1],'R','c');
Rt=R1+R;

I=V/Rt;

A=(I^2)*R;// To get the characteristic eqaution to
    find R.
```

```
B=A-50;
C=B(2);

rts=roots(C);// To find the two resistances

R=round(10000.*rts)./10000;// Rounding off to four
    decimal points.

Rt=R1+R;// Total resistance

I=V./Rt;// Currents

pow=(I.^2)*(R)';

power=diag(pow);

disp(B(2),'The Characteristic polynomial to find
    resistance R equated to zero is')

disp('ohms',R,'The solution of the above equation
    yields two resistances')

disp('Now to check which resistance is suitable by
    finding out the power dissipated by each of them'
    )

disp('watts',power,'The Power dissipated by both the
     resistors are')

disp('ohms',R(1),'From comparison with the given
    value (50 watts), We find that the suitable
    resistance is')

// The higher resistance is preferred because it
    limits the amount of current, ( Please see the
    current ratings of the resistors (Heating effect)
    )
```

**Scilab code Exa 12.12** To Compute the resistance when operating voltage is altered

```
// To Compute the resistor , when operating voltage
   is altered.

clc;
clear;

V=120;
P=100;

Rd=(V^2)/P;

Vr=80; // Reduced voltage

Ir= Vr/Rd;// Reduced current

Rt=V/Ir; // The Total Resistance required to
   circulate the reduced current.

Re= Rt-Rd; // External resistance required.

disp('ohms',Re,'The external resistance required to
   be connected in series to operate at 80V')
```

---

**Scilab code Exa 12.13** To Determine the voltage and branch currents in a cicuit with resistors connected in parallel

```
// To Determine the voltage and branch currents in a
    cicuit with resistors connected in parallel

```

```
clc;
clear;

R1=750;
R2=600;
R3=200;

C1=1/R1;
C2=1/R2;
C3=1/R3;

C= C1+C2+C3;// Total Conductance

I=1;

// 1/C is total resistance R, We use Ohm's Law to
   find the voltage applied.

V=I/C;// V=I*R

// Branch Currents
I1=V/R1;
I2=V/R2;
I3=V/R3;

disp('volts',V,'The applied voltage = ')
disp('amperes',I1,'The Current through 750 ohm
   Resistor =')
disp('amperes',I1,'The Current through 600 ohm
   Resistor =')
disp('amperes',I1,'The Current through 200 ohm
   Resistor =')
disp('amperes, Hence Verified.',(I1+I2+I3),'The
   Total Current through the circuit =')
```

**Scilab code Exa 12.14** To determine resistances in parallel

```
// To determine resistances in parallel.

clc;
clear;

I=25;
V=200;
P1=1500;

// Voltage remains the same in both the coils.
// Power Equation and Ohm's Law is being
   incorporated.

I1=P1/V;

R1=V/I1;

I2=I-I1;

R2= V/I2;

disp('ohms',R1,'The resistance of coil 1 =')
disp('ohms',R2,'The resistance of coil 2 =')
```

**Scilab code Exa 12.15** To determine the currents in parallel branches of a network

```
// To determine the currents in parallel branches of
    a network.

clc;
clear;

```

```
I=40;

R1=20;
R2=60;

//Current Divider equation I1= I*(R2/(R1+R2))

I1=I*(R2/(R1+R2));
I2=I*(R1/(R1+R2));

disp('A',I1,'The Current in the 20 ohm branch =')
disp('A',I2,'The Current in the 60 ohm branch =')
```

**Scilab code Exa 12.16** To determine current through each resistor in series and parallel combinational circuit

```
// To determine current through each resistor in
   series and parallel combinational circuit

clc;
clear;

R=10;
R1=20;
R2=30;

// R is the resistance in series with the parallel
   combination of R1 and R2.

V=100;

Reff=(R1*R2)/(R1+R2);

Rt=R+Reff;

```

```
I=V/Rt;

V1=I*R; // Voltage drop across 10 ohm resistor.

I1=I*(R2/(R1+R2));
I2=I*(R1/(R1+R2));

disp('ohms',Rt,'The total resistance of the network
    =')
disp('A',I,'The current through 10 ohm resistor =')
disp('A',I1,'The current through 20 ohm resistor =')
disp('A',I2,'The current through 30 ohm resistor =')
```

**Scilab code Exa 12.17** To calculate current in each branch of the given network

```
// To calculate current in each branch of the given
    network.

clc;
clear;


// Refer diagram (a) in the book

R1=6;// one of the resistance between a and b
R2=3;// one of the resistance between a and b
R3=8;// resistance between c and a
R4=15;// resistance in the middle branch
R5=4;// resistance between d and e

V=40;

Rab=(R1*R2)/(R1+R2);// Effective resistance between
    a and b
```

```
Rcb= Rab+R3;// Effective resistance of the top
    branch between c and b

Reff=(Rcb*R4)/(Rcb+R4);

Rt=Reff+R5;

I=V/Rt;

I1=I*(Rcb/(Rcb+R4));

I2=I*(R4/(Rcb+R4));

I3=I2*(R2/(R1+R2));

I4=I2*(R1/(R1+R2));

disp('amperes',I,'The current through 4 ohm resistor
    =')
disp('amperes',I1,'The current through 15 ohm
    resistor =')
disp('amperes',I2,'The current through 8 ohm
    resistor =')
disp('amperes',I4,'The current through 3 ohm
    resistor =')
disp('amperes',I3,'The current through 6 ohm
    resistor =')
```

**Scilab code Exa 12.18** To determine the current using loop analysis

```
// To determine the current using loop analysis

clc;
clear;
```

```
// MESH Equations
//6*i1 -2*i2=30
//-2*i1+6*i2=-40

R=[6 -2;-2 6];
E=[30;-40];

//The Loop Currents

I=inv(R)*E; // Matrix Method to solve for two
    unknowns in two eaquations.

i1=I(1);
i2=I(2);
i3=i1-i2;

disp('A',i2,'i2 =','A',i1,'i1 =','The Calculated
    Loop Currents are')

disp('The Negative sign indicates that the assumed
    direction of flow of current should be reveresed.
    ')

if(i1<0);
    i1=abs(i1);
end

if(i2<0);
        i2=abs(i2);
end

disp('A',i1,'The Current through 4 ohm resistor on
    the 30V side =')
disp('A',i2,'The Current through 4 ohm resistor on
    the 40V side =')
disp('A',i3,'The Current through 2 ohm resistor =')
```

**Scilab code Exa 12.19** To calculate current in each branch using loop analysis

```
// To calculate current in each branch using loop
   analysis.

clc;
clear;

// MESH Equations for the given network.
//3*i1-i2+0*i3=11
//-i1+10*i2-2*i3=0
//0*i1+-2*i2+5*i3=13

//Voltage supplies are 11V and 13V

R=[3 -1 0;-1 10 -2; 0 -2 5];
E=[11;0;13];

// Loop Currents

I=inv(R)*E;

i1=I(1);
i2=I(2);
i3=I(3);

ia=i1-i2; // Assumed direction from Mesh 1
ib=i2-i3; // Assumed direction from Mesh 2

disp('A',ib,'ib (through 2 resistor between 7 ohm
   and 3 ohm resistor) =','A',ia,'ia(through 1 ohm
   resistor) =','A',i3,'i3 =','A',i2,'i2 =','A',i1,'
   i1 =','The Calculated Loop Currents are')
```

```
disp('The Negative sign indicates that the assumed direction of flow of current should be reveresed')

// To obtain the magnitude of direction.

if(i1<0)
    i1=abs(i1);
end
if(i2<0)
    i2=abs(i2);
end
if(i3<0)
    i3=abs(i3);
end
if(ia<0)
    ia=abs(ia);
end
if(ib<0)
    ib=abs(ib);
end

disp('A',i1,'The Current through 2 ohm resistor on the 11V side =')
disp('A',i2,'The Current through 7 ohm resistor =')
disp('A',i3,'The Current through 3 ohm resistor on the 13V side =')
disp('A',ia,'The Current through 1 ohm resistor =')
disp('A',ib,'The Current through 2 ohm resistor between the 7 and 3 ohm resistors =')
```

**Scilab code Exa 12.20** To calculate current in each branch using loop analysis and point voltages in a given network

```
// To calculate current in each branch using loop
   analysis and point voltages in a given network.

clc;
clear;

// MESH Equations for the given network.
//3.95*i1 -3.75*i2+0*i3=120
//-3.75*i1 +9.5*i2 -5.45*i3=0
//0*i1 -5.45*i2 +5.55*i3=-110

// Positive of 120V DC supply connected to 0.2 ohm
   resistor
// Positive of 110 DC supply connected to 0.1 ohm
   resistor

//Voltage supplies are 120V and 110V

R=[3.95 -3.75 0;-3.75 9.5 -5.45; 0 -5.45 5.55];
E=[120;0;-110];

R1=abs(R(2)); // Resistor carrying ia
R2=abs(R(8)); // Resistor carrying ib

// Loop Currents

I=inv(R)*E;

i1=I(1);
i2=I(2);
i3=I(3);

ia=i1-i2; // Assumed direction from Mesh 1
ib=i2-i3; // Assumed direction from Mesh 2

// Using Nodal Analysis to find V1 and V2.
V1=R1*ia;
V2=R2*ib;
```

```

disp('A',ib,'ib (through 2 resistor between 7 ohm and 3 ohm resistor) =','A',ia,'ia(through 1 ohm resistor) =','A',i3,'i3 =','A',i2,'i2 =','A',i1,'i1 =','The Calculated Loop Currents are')

disp('The Negative sign indicates that the assumed direction of flow of current should be reveresed')

// To obtain the magnitude of direction.

if(i1<0)
    i1=abs(i1);
end
if(i2<0)
    i2=abs(i2);
end
if(i3<0)
    i3=abs(i3);
end
if(ia<0)
    ia=abs(ia);
end
if(ib<0)
    ib=abs(ib);
end

disp('A',i1,'The Current through 0.2 ohm resistor on the 120V side =')
disp('A',i2,'The Current through 0.3 ohm resistor =')
disp('A',i3,'The Current through 0.1 ohm resistor on the 110V side =')
disp('A',ia,'The Current through 3.75 ohm resistor =')
disp('A',ib,'The Current through 5.45 ohm resistor =')
```

```
disp('V',V1,'The voltage V1 =')
disp('V',V2,'The voltage V2 =')
```

**Scilab code Exa 12.21** To calculate current from a battery and pd across points A and B

```
// To calculate current from a battery and pd across
    points A and B

clc;
clear;

// Resistances in the given network
R1=4;
R2=2;
R3=3;
R4=6;
R5=8;

// MESH Equations
//9*i1 -5*i2=10
//-5*i1 +19*i2=0

// Supply voltage 10V

R=[(R1+R2+R3) -(R2+R3); -(R2+R3) (R2+R3+R4+R5)];
V=[10;0];

//Loop Currents
I=inv(R)*V;

i1=I(1);
i2=I(2);

```

```
i3=i1-i2; // From Mesh 1

// Point Voltages
Va=i3*R3;
Vb=i2*R5;

disp('amperes',abs(i1),'The current through 4 ohm
    resistor and the battery =')
disp('amperes',abs(i2),'The current through 6 ohm
    and 8 ohm resistors =')
disp('amperes',abs(i3),'The current through 2 ohm
    and 3 ohm resistors =')

disp('volts',abs(Va),'The voltage at point A =')
disp('volts',abs(Vb),'The voltage at point B =')
disp('volts',(Va-Vb),'The voltage across Points A
    and B =')
```

**Scilab code Exa 12.22** Determine Current through branch AB of the given network

```
// Determine Current through branch AB of the given
    network

clc;
clear;

// MESH Equations
// 4*i1 -2*i2 +0*i3 =10
// -2*i1 +6*i2 -2*i3 =0
//0*i1 -2*i2 +6*i3 =0

//Supply Voltage is 10V (Note printing mistake)

R=[4 -2 0;-2 6 -2; 0 -2 6];
```

```
V=[10;0;0];

// Loop Currents

I=inv(R)*V;

i1=I(1);
i2=I(2);
i3=I(3);

disp('amperes',abs(i2),'The current through branch
   AB of the network =')
```

**Scilab code Exa 12.23** Determine the current in the branches of the network using nodal analysis

```
// Determine the current in the branches of the
   network using nodal analysis

clc;
clear;

// Supply voltages
V1=30;
V2=40;

// Resistances in the network
R1=4;
R2=2;
R3=4;

Vb=poly([0 1],'Vb','c');

AD=(V1-Vb)/R1;
BD=(V2-Vb)/R3;
```

```
CD=Vb/R2;

X=AD+BD-CD;

disp('The Characterictic Equation to find Vb is')

disp(CD,'=',AD,'        +',BD)

Vb=roots(X);// Stores the numerical value of Vb

i1=(V1-Vb)/R1;
i2=(V2-Vb)/R3;
i3=Vb/R2;

disp('amperes',i1,'Current through 4 ohm resistor on
    the 30V supply side =')
disp('amperes',i2,'Current through 4 ohm resistor on
    the 40V supply side =')
disp('amperes',i3,'Current through 2 ohm resistor ='
   )
```

**Scilab code Exa 12.24** To Calculate current in all branches of the network shown using nodal analysis

```
// To Calculate current in all branches of the
   network shown using nodal analysis

clc;
clear;

// Nodal Equations
//13*Va-4*Vb=300
//-Va+4*Vb=120

X=[13 -4;-1 4];
```

```
V=[300;120];

E=inv(X)*V;

Va=E(1);
Vb=E(2);

i1=(100-Va)/20;
i2=(Va-Vb)/15;
i3=(Va/10);
i4=(Vb/10);
i5=(80-Vb)/10;

disp('V',Vb,'Voltage Vb =','V',Va,'Voltage Va =')

disp('The Branch Currents as calculated are')
disp(i5,'i5',i4,'i4',i3,'i3',i2,'i2',i1,'i1')
disp('amperes respectively')

disp('The Negative sign indicates that the assumed
   direction of flow of current must be reveresed')

disp('amperes',abs(i1),'The Current through 20 ohm
   resistor on the 100V side =')
disp('amperes',abs(i2),'The Current through 15 ohm
   resistor =')
disp('amperes',abs(i3),'The Current through 10 ohm
   resistor (AE) =')
disp('amperes',abs(i4),'The Current through 10 ohm
   resistor (BE) =')
disp('amperes',abs(i5),'The Current through 10 ohm
   resistor on the 80V side =')
```

# Chapter 13

# Electromagnetic Induction

**Scilab code Exa 13.1** Computing Induced EMF

```
// Computing Induced EMF
clc;
clear;

l=0.5;
v=50;
b=1;

// Angles
x=90;
y=30;
z=0;

// EMFs

e1=b*l*v*(sind(x));
e2=b*l*v*(sind(y));
e3=b*l*v*(sind(z));

disp('volts',e1,'i) The Induced EMF perpendicular
    to the field')
```

```
disp('volts',e2,'ii) The Induced EMF at an angle 30
    degrees to the field')
disp('volts',e3,'iii)The Induced EMF parallel to the
    field')
```

**Scilab code Exa 13.2** Computing Instantaneous Induced EMF

```
// Computing InstantaneousInduced EMF
clc;
clear;

l=0.2;
n=1000;
b=0.5;
r=l/2;
t=200;

//Number of conductors
c=2*t;

//Velocity Equation.
v=2*(%pi)*r*1000/60;

// Angles
x=90;
y=30;
z=0;

// EMFs

e1=c*b*l*v*(sind(90-x));
e2=c*b*l*v*(sind(90-y));
e3=c*b*l*v*(sind(90-z));

disp('volts',e1,'i)  The Induced EMF perpendicular
```

```
    to the field')
disp('volts',e2,'ii) The Induced EMF at an angle 30
    degrees to the field')
disp('volts',e3,'iii)The Induced EMF parallel to the
    field')
```

**Scilab code Exa 13.3** EMF Induced between wing tips

```
// EMF Induced between wing tips.
clc;
clear;

l=6.1;
vel=800;
v=vel*1000/3600;
b=(50)*(10^-6);
e=b*l*v;

disp('volts',e,'The EMF induced between the wing
    tips=')
```

**Scilab code Exa 13.4** EMF generated due to a bar magnet

```
// EMF generated due to a bar magnet.
clc;
clear;

b=(300)*(10^-6);
N=500;
t=(1/10);

// Faraday's LAW.

```

```
e=b*N/t;

disp('volts',e,'The EMF generated between the coil
    ends =')
```

**Scilab code Exa 13.5** EMF induced between two coils in a circular iron core

```
//EMF induced between two coils in a circular iron
    core
clc;
clear;

area= 5*(10^-4);
l=0.5;

na=200;
nb=500;

dI=15;
dt=(1/10);

mr=250;
mo= 4*%pi*(10^-7);

mfa=na*dI;

H=mfa/l;

B=mo*mr*H;

// Flux Linked

flux=B*area;

```

```
eb=nb*flux/dt;

disp('volts',eb,'The Induced EMF by coil B =')
```

**Scilab code Exa 13.6** Force on the conductor due to a uniform magentic field

```
// Force on the conductor due to a uniform magentic
   field
clc;
clear;

l=0.5;
B=0.12;
I=15;

F=B*I*l;

disp('N',F,'The Force on the conductor =')
```

**Scilab code Exa 13.7** To determine force between single phase bus bars

```
// To determine force between single phase bus bars

clc;
clear;

eload=10^7;
voltage=15*(10^3);

mo=4*%pi*(10^-7);

l=1;
```

```
r=0.3;

//Normally

I=eload/voltage;
F=mo*I*I*l/(2*%pi*r);

//Short Circuit

Isc=10*I;
Fsc=mo*Isc*Isc*l/(2*%pi*r);

disp('N',F,'i) Force per metre under normal
    condition =')
disp('N',Fsc,'ii) Force per metre under short
    circuit condition =')
```

**Scilab code Exa 13.8** To Compute the Maximum Induced EMF

```
// To Compute the Maximum Induced EMF

clc;
clear;

T=1/50;
fluxmax=0.02;
N=100;

flux= fluxmax*(poly([1 -4/T],'t','c'));// Flux
    Equation under consideration.

e=N*(-1)*derivat(flux);

disp('volts',e,'The Maximum induced EMF= ')
```

**Scilab code Exa 13.9** Compute Inductance and EMF induced

```
// Compute Inductance and EMF induced
clc;
clear;

d=0.25;
l=%pi*d;
n=1000;
area=6.25*(10^-4);
I=200;
mo=4*%pi*(10^-7);

L=mo*area*n*n/l;

V=L*I;

disp('H',L,'The Inductance of toriod')
disp('volts',V,'The Induced EMF in the toriod')
```

**Scilab code Exa 13.10** Change in Inductance

```
// Change in Inductance
clc;
clear;

L=120*(10^-3);
N=1000;
mr=75;
Nr=200;
Nc=N-Nr;

```

```
// Inductance directly proportional to the product
   of the square of turns and the relative
   permeability

Lc= L*((Nc/N)^2)*75;

disp('H',Lc,'The New value of inductance =')
```

**Scilab code Exa 13.11** Computation of Current

```
// Compute Current
clc;
clear;

i=3;
L=10;
t=20*(10^-3);
V=20*(10^3);

E=L*i*i/2;

P=E/t;

I=P/V;

disp('amperes',I,'The Current in the spark plug =')
```

**Scilab code Exa 13.12** To determine Mutual and Self Inductances

```
// To determine Mutual and Self Inductances

clc;
clear;
```

```
K=0.8;
I1=3;
flux1=0.4*(10^-3);
E2=85;
t=3*(10^-3);
N1=300;

L1=N1*flux1/I1;

M= E2*t/I1;

L2=(((M/K)^2)/L1);

flux2=K*flux1;

N2=(M*I1/flux2);

disp('H',L1,'The inductance of coil 1');
disp('H',L2,'The inductance of coil 2');
disp('H',M,'The inductance between the coils');
disp('Turns',N2,'The number of turns of coil 2');
```

**Scilab code Exa 13.13** Computation of Mutual and self inductance in coils

```
//Computation of Mutual and self inducatances

clc;
clear;

n1=600;
i1=2.5;
flux1=0.4*(10^-3);
flux2=0.8*(10^-3);
n2=2000;
```

```
tflux=flux1+flux2;

L1=n1*tflux/i1;

K=flux2/tflux;

M=n2*flux2/i1;

L2=((M/K)^2)/L1;

disp('H',L1,'The self inductance of coil 1= ')
disp('H',L2,'The self inductance of coil 2= ')

disp('(Note the accuracy of the answer.)')
disp('H',M,'The Mutual inductance of both coils= ')
disp(K,'The coupling co-efficient')
```

**Scilab code Exa 13.14** EMF induced in coils parallel to each other

```
// EMF induced in coils parallel to each other

clc;
clear;

Nx=1000;
Ix=5;
flux1=0.05*(10^-3);
di=12;
dt=10^-2;
K=60/100;

Lx= Nx*flux1/Ix;

// Since two coils are identical, Both will have equal self inductances.
```

```
Ly=Lx;

M=K*sqrt(Lx*Ly);

Ey=M*di/dt;

disp('volts',Ey,'The EMF induced by the coil Y =')
disp('H',Lx,'The Self Inductance of Coil X= ')
disp('H',Ly,'The Self Inductance of Coil X= ')
disp('H',M,'The Mutual Inductance of Coils')
```

**Scilab code Exa 13.15** To Compute the maximum flux set by an coil

```
//To Compute the maximum flux set by an coil.

clc;
clear;

L1=0.5;
L2=1;
K=0.8;

N2=1500;

M=K*sqrt(L1*L2);
i1=20;

theta=poly(0,'t');

fact=derivat(theta);// Derivative of the time factor
    (Linear function of 't').

omega=314;// The Angular Frequency Factor

```

```
v2=M*fact*omega*i1;// Maximum Coil 2 Voltage

disp('cos 314t V',v2,'The Voltage Across Coil 2= ')

dflx=v2/N2;

Mxflux=(dflx/omega)*(1/fact);


disp('sin 314t Wb',Mxflux,'The flux setup by Coil 1=
    ')

disp('Wb',Mxflux,'The Maximum Flux Setup by Coil 1=
    ')
```

**Scilab code Exa 13.16** Compute Loss of energy

```
//Compute Loss of energy

clc;
clear;

w=10;
f=50;// 50 cycles in a second.

ls_vol=250;

density=7.5;// Density in gm/cm^3

d= density*(10^6)/(10^3);

vol=w/d;

ls_cycle= ls_vol*vol;

```

```
ls_sec= ls_cycle*50;

ls_hr= ls_sec*3600;

disp('joules',ls_hr,'The Loss of energy per hour of
   an iron loop')
```

**Scilab code Exa 13.17** Determining Hysteresis loss

```
// Determining Hysteresis loss
clc;
clear;

fluxmax=1.5;

x=15/(10^-2);
y=1;

f=50;

a=x*y;// loss for one centimetre square.

area=0.6;// in centimetre square.

hy_ls= area*a;

vol=1500*(10^-6);

hyls_cycle= vol*hy_ls;

hyls_sec= hyls_cycle*f;

disp('watts',hyls_sec,'The Hysteresis loss =')
```

**Scilab code Exa 13.18** Compute the Loss per Kg at a particular frequency

```
//Compute the Loss per Kg at a particular frequency.
clc;
clear;

hy_ls=4.9;
f1=50;
maxflux=1;

density=7.5;

d=density*(10^6)/(10^3);

hy_ls_cycle= hy_ls*d/f1;

n=hy_ls_cycle/((maxflux)^1.7);

disp(n,'i) The value of the Co-Efficient= ')

mflux2=1.8;

f2=25;

hy_ls2=hy_ls*(f2/f1)*((1.8)^1.7);

disp('watt/kg',hy_ls2,'ii) The Loss per kg at 25Hz
   and 1.8 Wb per square metre= ')
```

**Scilab code Exa 13.19** Theory Based Proof relation between self mutual inductances

```
// Theory Based Proof relation between self mutual
    inductances

clc;
clear;

disp('L1=((N1)^2)*Mo*A/l')
disp('L2=((N2)^2)*Mo*A/l')
disp('R=l/(Mo*A)')

disp('L1=((N1)^2)/R)')
disp('L2=((N2)^2)/R)')

disp('M=K*((L1*L2)^(1/2))')

disp('(Substituting L1 and L2 in the above equation)
    ')

disp('M=K*N1*N2/R')
disp('Hence Proved')
```