# Scilab Textbook Companion for Fundamentals of Electrical Machines by M. A. Salam[1]

Created by
Shivananda F Madiwalar
BE
Electrical Engineering
Sri Jayachamarajendra College of Engineering
College Teacher
None
Cross-Checked by
Chaitanya Potti

June 2, 2016

[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:** Fundamentals of Electrical Machines

**Author:** M. A. Salam

**Publisher:** Narosa Publishing, New Delhi

**Edition:** 2

**Year:** 2009

**ISBN:** 9788184871630

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

**Exa** Example (Solved example)

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

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

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

# Contents

# List of Scilab Codes

# Chapter 1

# REVIEW OF ELECTRIC CIRCUITS

**Scilab code Exa 1.1** calculate current

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
// Example : 1.1

clc;clear; // clears the console and command history

// Given data
Q = 4    // chagre in C
t = 0.54 // time in sec


// caclulations
I = Q/t  // current in A

// display the result
```

```
disp("Example 1.1 solution");
printf(" \n Current is \n I = %.2f A ", I);
```

**Scilab code Exa 1.2** determine magnitude and sign of power

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
// Example : 1.2

clc;clear; // clears the console and command history

// Given data
V = -24 // voltage in V
I = 3  // current in A

// caclulations
 P = V*I // power for the element A in network in W

// display the result
disp("Example 1.2 solution");
printf(" \n Power for the element A in network \n P
   = %.0f W ", P);
```

**Scilab code Exa 1.3** use Thevenin theorem to find current

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION
```

```
// Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
// Example : 1.3

clc;clear; // clears the console and command history
// Given data
R1 = 5  // resistance in ohm
R2 = 4  // resistance in ohm
R3 = 9  // resistance in ohm
R4 = 6  // resistance in ohm
V1 = 10 // voltage in V
V2 = 6  // voltage in V


// caclulations
// Remove R3 and find R_th by short circuiting  V1
   and V2 than R1 and R4  wiil be in parallel
R = (R1*R4)/(R1+R4)    // equivalent resistance in
   ohm
// R is connected in series with R2
R_th = R2+R             // Thevenin's resistance in
   ohm
I = 4/11                // current in the figure
   applying KVL in A
V_th = (6*0.36)+6       // voltage in V
I_9 = V_th/(R_th+R3) // current through R3 in A


// display the result
disp("Example 1.3 solution");
printf(" \n Current through 9   \n I_9   = %.2f A ",
    I_9);
```

**Scilab code Exa 1.4** find the rms value

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
// Example : 1.4

clc;clear; // clears the console and command history

// Given data
V_t1 = 30   // magnitudes of voltages in V  0 < t1 <
    2
V_t2 = -10  // magnitudes of voltages in V  2 < t2 <
    4
T = 4 // time period from figure

// caclulations

V1 = 30;
V2 = -10;
X= sqrt((1/4)*(integrate('V1^2','x',0,2) + integrate
   ('V2^2','x',2,4)));

//display the result
disp("Example 1.4 solution");
printf("\n RMS value of the voltage waveform : \n
   V_rms = %.2f V ", X);
```

**Scilab code Exa 1.5** calculate the magnitude of the line voltage

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION
```

```
// Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
// Example : 1.5

clc;clear; // clears the console and command history


// Given data
V_P = 200                             // magnitude of
   phase voltage
V_an = 200                            // magnitude of
   phase voltage in V
V_bn = 200*exp(%i*(-120)*(%pi/180)) // magnitude of
   phase voltage in V
V_cn = 200*exp(%i*(120)*(%pi/180))  // magnitude of
   phase voltage in V

// caclulations
V_L = sqrt(3)*V_P // magnitude of line voltage in V

// display the result
disp("Example 1.5 solution");
printf(" \n Magnitude of line voltage \n V_L = %.2f
   V ", V_L);
```

**Scilab code Exa 1.6** find line currents and power factor

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
// Example : 1.6

```

```
clc;clear; // clears the console and command history


// Given data
R = 10       // resistance in
L = 15      // inductance reactance in
V_L = 420 // voltage in V
f = 50    // frwuency in Hz

// caclulations
V_an = (V_L/sqrt(3))*exp(%i*(-30)*(%pi/180))    // phase voltage
V_bn = (V_L/sqrt(3))*exp(%i*(-150)*(%pi/180)) // phase voltage
V_cn = (V_L/sqrt(3))*exp(%i*(90)*(%pi/180)) // phase voltage
Z_p = R+%i*L // phase impedance in
I_L1 = V_an/Z_p  // line current
I_L2 = V_bn/Z_p  // line current
I_L3 = V_cn/Z_p  // line current
pf =  R/abs(Z_p) // lagging power factor

// display the result
disp("Example 1.6 solution");
printf(" \n Line currents are \n I_L1 = %.2f<%.2f A \n", abs(I_L1),atand(imag(I_L1),real(I_L1)) );
printf(" I_L2 = %.2f<% 2f A \n", abs(I_L2),atand(imag(I_L2),real(I_L2)) );
printf(" I_L3 = %.2f<% 2f A \n", abs(I_L3),atand(imag(I_L3),real(I_L3)) );
printf(" \n Power factor \n pf = %.1f lag \n", pf );
```

**Scilab code Exa 1.7** find magnitude of phase currents and line current

```
// FUNDAMENTALS OF ELECTICAL MACHINES
```

```
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
// Example : 1.7

clc;clear; // clears the console and command history


// Given data
Z_p = 10+%i*15 // per phase impedance in
V_ab = 420                                // magnitude of phase voltage in V
V_bc = 420*exp(%i*(-120)*(%pi/180))// magnitude of phase voltage in V
V_ca = 420*exp(%i*(120)*(%pi/180)) // magnitude of phase voltage in V



// caclulations
I_ab = V_ab/Z_p  // phase current in A
I_bc = V_bc/Z_p // phase current in A
I_ca = V_ca/Z_p // phase current in A
I_P = V_ab/abs(Z_p)
I_L = sqrt(3)*I_P // line current in A

// display the result
disp("Example 1.7 solution");
printf(" \n Phase currents are \n I_ab = %.2f<%.2f A \n", abs(I_ab),atand(imag(I_ab),real(I_ab)) );
printf(" I_bc = %.2f<% 2f A \n", abs(I_bc),atand(imag(I_bc),real(I_bc)) );
printf(" I_ca = %.2f<% 2f A \n", abs(I_ca),atand(imag(I_ca),real(I_ca)) );
printf(" \n Line current \n I_L = %.2f A \n", I_L);
```

**Scilab code Exa 1.8** find line currents and load phase voltages

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
// Example : 1.8

clc;clear; // clears the console and command history

// Given data
V_L = 280 // generator phase voltage in V rms
Z_1 = 2+%i*3 // line impedance
Z_L = 4+%i*5 // load impedance

// caclulations
Z_t = Z_1+Z_L //
V_An = (V_L*exp(%i*(0)*(%pi/180)))    // phase
    voltage
V_Bn = (V_L*exp(%i*(-120)*(%pi/180))) // phase
    voltage
V_Cn = (V_L*exp(%i*(120)*(%pi/180)))  // phase
    voltage

I_Aa = V_An/Z_t   // line current for phase A in A
I_Bb = V_Bn/Z_t   // line current for phase B in  A
I_Cc = V_Cn/Z_t  // line current for phase C in A

V_an = I_Aa*Z_L  // voltage of load in V
V_bn = I_Bb*Z_L // voltage of load in V
V_cn = I_Cc*Z_L  // voltage of load in V
```

```
// display the result
disp("Example 1.8 solution");
// There is error in the textbook regarding phasor
   angles of
printf(" \n Line currents are \n I_Aa = %.2f<%.2f A
   \n", abs(I_Aa),atand(imag(I_Aa),real(I_Aa)) );
printf(" I_Bb = %.2f<% 2f A \n", abs(I_Bb),atand(
   imag(I_Bb),real(I_Bb)) );
printf(" I_Cc = %.2f<% 2f A \n\n", abs(I_Cc),atand(
   imag(I_Cc),real(I_Cc)) );
printf(" NOTE : PHASOR ANGLES CALCULATED IN TEXTBOOK
    FOR Van,Vbn & Vcn are wrong,\n because 4+j5 =
   6.4<51.34, but in the textbook it is taken as
   6.4<38.6");
// NOTE : PHASOR ANGLES CALCULATED IN TEXTBOOK FOR
   Van,Vbn & Vcn are wrong,because 4+j5 = 6.4<51.34,
    but in the textbook it is taken as 6.4<38.6
printf(" \n\n Phase voltages are ");
printf("\n V_an = %.2f<%.2f V \n", abs(V_an),atand(
   imag(V_an),real(V_an)) );
printf(" V_bn = %.2f<% 2f V \n", abs(V_bn),atand(
   imag(V_bn),real(V_bn)) );
printf(" V_cn = %.2f<% 2f V \n", abs(V_cn),atand(
   imag(V_cn),real(V_cn)) );
```

**Scilab code Exa 1.9** find phase currents and neutral current

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
// Example : 1.9
```

```

clc;clear; // clears the console and command history

// Given data
V_P = 340 //  phase voltage in V
Z = 6+%i*8 // per phase impedance

// caclulations

V_AN = (V_P/exp(%i*(0)*(%pi/180)))   // phase voltage
V_BN = (V_P/exp(%i*(-120)*(%pi/180))) // phase voltage
V_CN = (V_P/exp(%i*(120)*(%pi/180))) // phase voltage

I_Aa = V_AN/Z   // line current for phase A in A
I_Bb = V_BN/Z  // line current for phase B in  A
I_Cc = V_CN/Z  // line current for phase C in A



// display the result
disp("Example 1.9 solution");
printf(" \n Line currents are \n I_Aa = %.2f<%.2f A \n", abs(I_Aa),atand(imag(I_Aa),real(I_Aa)) );
printf(" I_Bb = %.2f<% 2f A \n", abs(I_Bb),atand(imag(I_Bb),real(I_Bb)) );
printf(" I_Cc = %.2f<% 2f A \n", abs(I_Cc),atand(imag(I_Cc),real(I_Cc)) );
printf(" \n The load is balanced , so the value of the neutral current will be zero")
```

**Scilab code Exa 1.10** find current in each load and line currents

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
// Example : 1.10

clc;clear; // clears the console and command history

// Given data
V_AN = 200 // voltage in V
 Z = 3+%i*4 // impedance in

// caclulations

V_AB = sqrt(3)*(V_AN*exp(%i*(30)*(%pi/180)))   // phase voltage in V
V_BC = sqrt(3)*(V_AN*exp(%i*(-90)*(%pi/180))) // phase voltage in V
V_CA = sqrt(3)*(V_AN*exp(%i*(150)*(%pi/180))) // phase voltage in V

I_ab = V_AB/Z  // magnitude of load current in A
I_bc = V_BC/Z  // magnitude of loadcurrent in A
I_ca = V_CA/Z // magnitude of  load current ‘in A

I_Aa = I_ab-I_ca // magnitude of  line current in A
I_Bb = I_bc-I_ab // magnitude of  line current in A
I_Cc = I_ca-I_bc // magnitude of  line current in A

// display the result
disp("Example 1.10 solution");
printf(" \n Current in each load \n I_ab = %.2f<%.2f A \n", abs(I_ab),atand(imag(I_ab),real(I_ab)) );
printf(" I_bc = %.2f<% 2f A \n", abs(I_bc),atand(imag(I_bc),real(I_bc)) );
printf(" I_ca = %.2f<% 2f A \n\n", abs(I_ca),atand(
```

```
    imag(I_ca),real(I_ca)) );

printf(" \n Line currents \n I_Aa = %.2f<%.2f A \n",
     abs(I_Aa),atand(imag(I_Aa),real(I_Aa)) );
printf(" I_Bb = %.2f<% 2f A \n", abs(I_Bb),atand(
    imag(I_Bb),real(I_Bb)) );
printf(" I_Cc = %.2f<% 2f A \n\n", abs(I_Cc),atand(
    imag(I_Cc),real(I_Cc)) );

printf(" NOTE : The  line currents as given in this
    example in text book are wrong since ,the quesion
     says that both the three phase supply and load
    are balanced so the line should be balanced which
     they fail to do in text book.");
```

**Scilab code Exa 1.11** calculate line currents pf power supplied

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
// Example : 1.11

clc;clear; // clears the console and command history

// Given data
Z = 3+%i*4 // impedance in
V_AN = 150 // voltage in V
R = 3 // resistance in    from Z

// caclulations

V_AN = (150*exp(%i*(0)*(%pi/180)))   //source
```

```
voltage in V
V_BN = (150*exp(%i*(-120)*(%pi/180))) // source voltage in V
V_CN = (150*exp(%i*(120)*(%pi/180))) // source voltage in V

I_Aa = V_AN/Z   // line current in A
I_Bb = V_BN/Z   // line current in  A
I_Cc = V_CN/Z  // line current in A

pf = R/abs(Z)     // power factor
I = V_AN/abs(Z) // current in A
P = V_AN*I*pf  // power supplied in W
P_t = 3*P     // total power supplied in W

// display the result
disp("Example 1.11 solution");
printf(" \n Line currents are \n I_Aa = %.2f<%.2f A \n", abs(I_Aa),atand(imag(I_Aa),real(I_Aa)) );
printf(" I_Bb = %.2f<% 2f A \n", abs(I_Bb),atand(imag(I_Bb),real(I_Bb)) );
printf(" I_Cc = %.2f<% 2f A \n\n", abs(I_Cc),atand(imag(I_Cc),real(I_Cc)) );
printf(" Power factor \n pf = %.1f \n", pf);
printf(" \n Power supplied to each phase is \n P = %.2f W \n", P);
printf(" \n Total power supplied \n P_t = %.2f W \n", P_t);
```

**Scilab code Exa 1.12** find second wattmeter reading

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION
```

```
// Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
// Example : 1.12

clc;clear; // clears the console and command history


// Given data
P = 120         // total power in kW
cos_teta = 0.6 // power factor

// caclulations
teta = acosd(0.6)            // power factor angle
P_1 = tand(teta)*P/sqrt(3)
P_2 = (P_1+P)/2              // second wattmeter reading in kW

// display the result
disp("Example 1.12 solution");
printf(" \n Second wattmeter reading \n P_2= %.1f kW \n", P_2);
```

# Chapter 2

# BASICS OF MAGNETIC CIRCUITS

**Scilab code Exa 2.1** calculate total flux

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 2 : BESICS OF MAGNETIC CIRCUITS
// Example : 2.1

clc;clear; // clears the console and command history

// Given data
l = 4        // lenght in m
w = 2       // width in m
B = 0.12 // magnetis flux density in tesla

// caclulations
A = l*w      // area in m^2
flux = B*A   // magnetic flux in Wb

```

```
// display the result
disp("Example 2.1 solution");
printf(" \n Magnetic flux \n flux = %.2f Wb \n",
   flux);
```

---

**Scilab code Exa 2.2** calculate flux density MMF magnetic field intensity

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 2 : BESICS OF MAGNETIC CIRCUITS
// Example : 2.2

clc;clear; // clears the console and command history

// Given data
d_in = 3 // inside diameter of iron toroid in cm
d_out = 6 // outside diameter of iron toroid in cm
N =200 // number of turns
I = 3 // current in A
flux = 0.015 // flux in Wb

// caclulations
d = d_in+((d_out-d_in)/2) // distance in cm
l = %pi*d // mean length in cm
A = %pi*d^2/4 // area in cm^2
B =flux/(A*10^-4) // flux density in mWb/m^2
MMF = N*I // magnetomotive force in At
H = (N*I)/(l*10^-2) // magnetic field intensity in
   At/m

// display the result
```

```
disp("Example 2.2 solution");
printf("\n Flux density \n B= %.6f mWb/m^2 \n", B);
printf(" \n Magnetomotive force \n MMF= %.2f At \n",
    MMF);
printf(" \n Magnetic field intensity \n H= %.2f At/m
    \n", H);
printf(" NOTE:correction in solution they took d=1.5
    insted of 4.5")
```

**Scilab code Exa 9.2** determine pitch factor

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 9 : SYNCHRONOUS GENERATOR
// Example : 9.2

clc;clear; // clears the console and command history

// Given data
P = 8      // number of poles
m = 3      // number of phase
S = 144    // number of slots

// caclulations
T_p = S/P           // pole pitch interms of slots
slots_1 = 180/T_p   // pole pitch per slots
y = 2*slots_1       // short pitch angle in degree
k_p = cosd(y/2)     // pitch factor

// display the result
disp("Example 9.2 solution");
printf(" \n Pitch factor is \n k_p = %.2f \n", k_p )
```

;

---

**Scilab code Exa 2.3** find permeability MFI MFD

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 2 : BESICS OF MAGNETIC CIRCUITS
// Example : 2.3

clc;clear; // clears the console and command history

// Given data
myu_r = 625 // relative permiability of rectangular
    core
N = 25      // number of turns
I = 2       // current in A
a = 5.5     // lenght of rectangular box in cm
b = 1.5     // width of rectangular box in cm

// caclulations
l = 2*(a+b)               // mean length of core in cm
H = N*I/(l*10^-2)         // magnetic field intensity
    in At/m
myu = 4*%pi*10^-7*myu_r // permeabilty
B = myu*H                 // magnetic flux density in
    Wb/m^2;

// display the result
disp("Example 2.3 solution");
printf(" \n Magnetic field intensity \n H= %.0f At/m
    \n", H);
printf(" \n Permeabilty \n myu= %.2e  \n", myu);
```

```
printf(" \n Magnetic flux density \n B= %.2f Wb/m^2
    \n", B);
```

**Scilab code Exa 2.4** find MMF and reluctance

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 2 : BESICS OF MAGNETIC CIRCUITS
// Example : 2.4

clc;clear; // clears the console and command history

// Given data
N = 6          // number of turns
I =3           // current in A
flux = 0.056 // flux in Wb

// caclulations
MMF = N*I        // magnetomotive force in At
R_m = MMF/flux // reluctance in At/Wb

// display the result
disp("Example 2.4 solution");
printf(" \n Magnetomotive force \n MMF= %.0f At \n",
    MMF);
printf( "\n Reluctance \n R_m= %.1f At/Wb \n", R_m);
```

**Scilab code Exa 2.5** find MFD

```
// FUNDAMENTALS OF ELECTICAL MACHINES
```

```
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 2 : BESICS OF MAGNETIC CIRCUITS
// Example : 2.5
clc;clear; // clears the console and command history

// Given data
I = 15                    // current through conductor
   in A
N = 10                    // number of turns
myu_0 = 4*%pi*10^-7     // permiability
myu_r = 1               // relative permiability of air
r = 0.015

// caclulations
B = myu_0*myu_r*N*I/(2*%pi*r) // magnetic flux in T

// display the result
disp("Example 2.5 solution");
printf(" \n Magnetic flux \n B= %.0e T \n", B);
```

**Scilab code Exa 2.6** find MFS flux density and flux

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 2 : BESICS OF MAGNETIC CIRCUITS
// Example : 2.6

clc;clear; // clears the console and command history

```

```
// Given data
N = 200       // number of turns
d_in = 7     // inner diameter of wooden toroidal
    coil in cm
d_out = 10  // outer diameter  of wooden toroidal
    coil in cm
A = 0.005   // cross sectional area m^2
I = 5       // current through coil in A
R = d_out-d_in
myu_0 = 4*%pi*10^-7

// caclulations
l = 2*%pi*R*10^-2   // mean circumference length in m
H = N*I/l           // magnetic field intensity in At
    /m
B = myu_0*H         // flux density in Wb/m^2
flux = B*A         // flux in Wb

// display the result
disp("Example 2.6 solution");
printf(" \n Magnetic field intensity  \n H= %.0f At/
   m \n", H);
printf(" \n Flux density \n B= %.2e Wb/m^2 \n", B);
printf(" \n Flux \n flux= %.1e Wb \n", flux);
```

**Scilab code Exa 2.7** calculate flux flux density and field intensity

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 2 : BESICS OF MAGNETIC CIRCUITS
// Example : 2.7
clc;clear; // clears the console and command history
```

```

// Given data
l = 0.1       // length in m
w = 0.01      // width in m
h =0.1        // height in m
N = 450       // number of turns
I = 0.2       // current in A
myu_r = 850 // relative permiability magnetic material

// caclulations
MMF = N*I                          // magnetomotive force in At
l_c = (h-w)*4                      //  mean length of the path in m
A = w*w                            // cross sectional area in m^2
R_m = l_c/(4*%pi*10^-7*myu_r*A) // relectance in At/Wb
flux = MMF/R_m                     // flux in Wb
B = flux/A                         // magnetic flux density in Wb/m^2
H = B/(4*%pi*10^-7*myu_r)          // field intensity in At/m

// display the result
disp("Example 2.7 solution");
printf(" \n Flux \n flux= %.2e Wb \n", flux);
printf(" \n Magnetic flux density \n B= %.4f Wb/m^2 \n", B);
printf(" \n Field intensity \n H= %.2f At/m \n", H);
```

**Scilab code Exa 2.8** calculate current

```
// FUNDAMENTALS OF ELECTICAL MACHINES
```

```
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 2 : BESICS OF MAGNETIC CIRCUITS
// Example : 2.8

clc;clear; // clears the console and command history

// Given data
N = 450         // number of turns wound on left side of limb
A = 4           // cross sectional area in m^2
I = 5          // current through coil in A
fulx = 3       // flux in right limb Wb
myu_r = 500   // relative permiability magnetic material
l_1 = 0.12
l_2 = 0.24
phi2 = 3

// caclulations
// we have relation
// phi1*rm1 = phi2*rm2
// phi1*l_1/uA = pi2*l_2/uA
phi1 = phi2*l_2/l_1               // flux on left side in Wb
flux = phi1+phi2                  // total flux in Wb
B_1 = flux/A                      // flux density in the left limb
H_1 = B_1/(4*%pi*10^-7*myu_r) // magnetic flux in At/m
MMF_1 = H_1*l_2                   // magnetomotive force in At
B_2 = phi2/A                      // flux density in the right limb
H_2 = B_2/(4*%pi*10^-7*myu_r) // magnetic flux in At/m
```

```
MMF_2 = H_2*l_2                    // magnetomotive force in At
MMF_t = MMF_1+MMF_2                // total magnetomotive force in At
I = MMF_t/N                        // current in A

// display the result
disp("Example 2.8 solution");
printf(" \n Current\n I= %.2f A \n", I);
```

**Scilab code Exa 2.9** find relucatance and relative permeability

```
 // FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 2 : BESICS OF MAGNETIC CIRCUITS
// Example : 2.9
clc;clear; // clears the console and command history

// Given data
l = 0.45                // mean lenght in m
A = 25*10^-4            // cross sectional area in m^2
l_ag = 0.8*10^-3        // air gap in m
N = 500               // number of turns
I = 1.25              // current in A
fulx = 2.25*10^-3     // flux in Wb
phi = 1.5*10^-3     // flux in Wb

// caclulations
B = phi/A                          // magnetic flux density in Wb/m^2
MMF = N*I                          // magnetomotive force in At
```

```
H = B/(4*%pi*10^-7)                  // magnetomotizing
    force in At/m
MMF_ag = H*l_ag                      // magnetomotive
    force in At
MMF_i = MMF-MMF_ag                   // magnetomotive
    force for iron ring in At
H_i = MMF_i/l                        // magnetic field
    intensity for iron part in At/m
myu_r = B/((4*%pi*10^-7)*H_i)        // relative
    permiability for iron

// display the result
disp("Example 2.9 solution");
printf(" \n Relative permiability for iron \n myu_r
    = %.2f  \n", myu_r);
printf(" given current value in question is 2.25A,
    but in solution they took value of current as
    1.25A ");
```

**Scilab code Exa 2.10** calculate current

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 2 : BESICS OF MAGNETIC CIRCUITS
// Example : 2.10
clc;clear; // clears the console and command history

// Given data
A = 2*10^-4          // cross sectional area in m^2
N = 200              // number of turns
flux = 1.5*10^-3     // flux in Wb
myu_r = 4000         // relative permiability of core
```

```
l_1 = 0.01          // length in m
a = 9               // length in cm
w = 3               // width in cm

// caclulations
myu = myu_r*4*%pi*10^-7        // permiability
l_2 = (4*(a-w-w+(1.5+1.5))-1) // mean length in cm
R_mg = l_1/(4*%pi*10^-7*A)     // reluctance of iron
    for air gap At/Wb
R_mi = l_2*10^-2/(myu*A)       // reluctance of iron
    for air gapAt/Wb
R_mt = R_mg+R_mi               // total relectance in
     At/Wb
I = R_mt*flux/N                // current in A

// display the result
disp("Example 2.10 solution");
printf(" \n Current flowing through the coil \n I =
   %.0f A \n", I);

// NOTE : In question they given flux=2.5mWb but in
    solution they took flux=1.5mWb
```

**Scilab code Exa 2.11** determine force

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 2 : BESICS OF MAGNETIC CIRCUITS
// Example : 2.11
clc;clear; // clears the console and command history

// Given data
```

```
I = 150  // current through conductor in A
l = 2    // conductor length in m
B = 0.35 // magnetic flux density in T

// caclulations
F = B*l*I // force in N

// display the result
disp("Example 2.11 solution");
printf("\n Force \n F = %.0f N \n", F);
```

**Scilab code Exa 2.12** determine the inductance of the coil

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 2 : BESICS OF MAGNETIC CIRCUITS
// Example : 2.12
clc;clear; // clears the console and command history

// Given data
l = 25*10^-2    // length of air-core coil in m
A = 4*10^-4     // cross sectional area in m^2
N = 65          // number of turns
myu_0 = 4*%pi*10^-7
myu_r = 1

// caclulations
myu = myu_0*myu_r
L = N^2*myu*A/l  // inductance in H

// display the result
disp("Example 2.12 solution");
```

```
printf(" \n Inductance of the coil \n L = %.1e H \n"
    , L);
```

**Scilab code Exa 2.13** determine hysteresis loss

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 2 : BESICS OF MAGNETIC CIRCUITS
// Example : 2.13
clc;clear; // clears the console and command history

// Given data
k_h = 110        // hysteresis co-efficient in J/m^3
V_cvol = 0.005   // volume of the core in m^3
B_m = 1.12       // flux density in T
f = 60           // frequency in Hz
n = 1.6

// caclulations
P_h = k_h*V_cvol*B_m^n*f // hysteresis loss in W

// display the result
disp("Example 2.13 solution");
printf(" \n Hysteresis loss \n P_h = %.2f W \n", P_h
    );
```

# Chapter 3

# TRANSFORMER AND PER UNIT SYSTEM

**Scilab code Exa 3.1** find no of primary turns primary full load current and secondary full load current

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.1
clc;clear; // clears the console and command history

// Given data
V_1 = 2200  // primary voltage of transformer in V
V_2 = 220   // secondary voltage of transformer in V
N_2 = 56    // number of turns in the secondary coil
     of transformer
kVA = 25    // kVA rating of transformer

// caclulations
a = V_1/V_2         // turn ratio
```

```
N_1 = a*N_2           // number of primary turns
I_1 = kVA*10^3/V_1   // primary full load current in A
I_2 = kVA*10^3/V_2   // secondary full load current in A

// display the result
disp("Example 3.1 solution");
printf(" \n Number of primary turns \n N_1 = %.0f \n", N_1);
printf(" \n Primary full load current \n I_2 = %.2f A \n", I_1);
printf(" \n Secondary full load current \n I_2 = %.1f A \n", I_2);
```

**Scilab code Exa 3.2** determine turns ratio and mutual flux

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.2
clc;clear; // clears the console and command history

// Given data
V_1 = 220 // voltage in V
N_1 = 150 // number of turns in the primary coil of transformer
N_2 = 300 // number of turns in the secondary coil of transformer
f = 50     // frequency in Hz

// caclulations
```

```
a = N_1/N_2                    // turn ratio
phi_m = V_1/(4.44*f*N_1) // mutual flux in Wb

// display the result
disp("Example 3.2 solution");
printf(" \n Mutual flux \n phi_m = %.2e Wb \n",
   phi_m);
```

**Scilab code Exa 3.3** find iron loss and magnetizing component of currents

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.3
clc;clear; // clears the console and command history

// Given data
V_1 = 2200 // primary voltage of transformer in V
V_2 = 220  // secondary voltage of transformer in V
I_0 = 0.5  // no-load current in A
P_0 = 350  // absorbed power by transformer

// caclulations
phi = acos(.32)
sin_phi = sin(phi)
cos_phi = P_0/(V_1*I_0) // no-load power factor
I_w = I_0*cos_phi        // iron loss component of
   current A
I_m = I_0*sin_phi        // magnetizing component of
   current A


```

```
// display the result
disp("Example 3.3 solution");
printf(" \n The iron loss component of current A \n I_w = %.2f A \n", I_w);
printf(" \n The magnetizing component of current A \n I_m = %.2f A \n", I_m);
```

**Scilab code Exa 3.4** find turns ratio load impedance primary current

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.4
clc;clear; // clears the console and command history

// Given data
N_1 = 450  // number of turns in the primary coil of transformer
N_2 = 45   // number of turns in the secondary coil of transformer
Z_L = 3    // load impedance in
V_1 = 15   // primary coil voltage of transformer in V

// caclulations
a = N_1/N_2   // turn ratio
Z_1 = a^2*Z_L // load impedance referred to primary ohm
I_1 = V_1/Z_1 // primary current in A


// display the result
```

```
disp("Example 3.4 solution");
printf(" \n Turn ratio \n a = %.0f \n", a);
printf(" \n Load impedance referred to primary \n Z_1 = %.0f     \n", Z_1);
printf(" \n Primary current \n I_1 = %.2f A \n", I_1);
```

**Scilab code Exa 3.5** determine primary current

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.5
clc;clear; // clears the console and command history

// Given data
V_1 = 400       // primary voltage of transformer in V
V_2 = 100       // secondary voltage of transformer in V
I_0 = 0.4       // no-load current in A
I_2 = 100       // load draws current in A
cos_phi0 = 0.3 // power factor lagging from the supply
cos_phi2 = 0.6 // power factor lagging from the secondary

// caclulations
phi0 = acosd(0.3)
phi2 = acosd(0.6)
phi1 = phi0-phi2
a = V_1/V_2  // turn ratio
```

```
I_2! = I_2/a // secondary current equivalent to the
    primary
I_1 = sqrt((I_2!^2)+(I_0^2)+(2*I_2!*I_0*cosd(19.4)))
     // primary current in A

// display the result
disp("Example 3.5 solution");
printf(" \n Primary current \n I_1 = %.2f A \n", I_1
    );
```

**Scilab code Exa 3.6** determine impedance referred to the primary and secondary

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.6
clc;clear; // clears the console and command history

// Given data
V_1 = 2000  // primary voltage of transformer in V
V_2 = 400   // secondary voltage of transformer in V
kVA = 200   // kVA rating of transformer
R_1 = 3     // primary resistance in
X_1 = 12    // primary reactance in
R_2 = 0.3   // secondary resistance in
X_2 = 0.1   // secondary reactance in

// caclulations
a = V_1/V_2 // turn ratio
R_01 = R_1+(a^2*R_2)          // total resistance
    referred to primary side
```

```
X_01 = X_1+(a^2*X_2)             // total reactance referred to primary side
Z_01 = sqrt((R_01^2)+(X_01^2)) // equivalent impedance reffered to primary side in
R_02 = R_2+(R_1/a^2)             // total resistance referred to secondary side
X_02 = X_2+(X_1/a^2)             // total reactance referred to secondary side
Z_02 = sqrt((R_02^2)+(X_02^2)) // equivalent impedance reffered to secondary side in

// display the result
disp("Example 3.6 solution");
printf(" \n Equivalent impedance reffered to primary side \n Z_01 = %.1f    \n", Z_01);
printf(" \n Equivalent impedance reffered to secondary side \n Z_02 = %.2f    \n", Z_02);
```

**Scilab code Exa 3.7** find voltage regulation

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.7
clc;clear; // clears the console and command history

// Given data
V_1 = 200        // primary voltage of transformer in V
V_2 = 400        // secondary voltage of transformer in V
R_1 = 0.3        // primary resistance in
```

```
X_1 = 0.6       // primary reactance in
R_2 = 0.8       // secondary resistance in
X_2 = 1.6       // secondary reactance in
I_2 = 10        // secondary supply current in A
cos_phi2 = 0.8 // power factor lagging

// caclulations
a = V_1/V_2            // turn ratio
phi2 = acosd(0.8)
sin_phi2 = sind(phi2)
R_02 = R_2+(R_1/a^2) // total resistance referred to
    secondary side
X_02 = X_2+(X_1/a^2) // total reactance referred to
    secondary side
E_2 = (V_2*cos_phi2+I_2*R_02)+(%i*(V_2*sin_phi2+I_2*
    X_02)) // no-load voltage
V_r = (abs(E_2)-V_2)/V_2*100
                                        // voltage
    regulation

// display the result
disp("Example 3.7 solution");
printf(" \n Voltage regulation \n V_r = %.0f percent
    \n", V_r);
```

**Scilab code Exa 3.8** calculate efficiency

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.8
clc;clear; // clears the console and command history
```

```

// Given data
P_i = 1   // iron loss of transformer in kW
P_cu = 2  // copper loss of transformer in kW
kVA = 200 // kVA ratingss of transformer
pf = 0.95 // power factor

// caclulations
P_cu1 = (3/4)^2*P_cu   // copper loss at 1/2th of full load in kW
P_cu2 = (1/2)^2*P_cu   // copper loss at 1/2th of full load in kW
P_01 = (3/4)*kVA*P_i   // o/p power at 3/4 full load and unity power factor in kW
P_in1 = P_01+P_i+P_cu1 // i/p power at 3/4 full load and unity power factor in kW
n_1 = (P_01/P_in1)*100 // efficiency at 3/4 full load and unity power factor
P_02 = (1/2)*kVA*pf   // o/p power factor at1/2 full load and 0.95 power factor in kW
P_in2 = P_02+P_i+P_cu2 // i/p power at 1/2 full load and 0.95 power factor in kW
n_2 = (P_02/P_in2)*100 // efficiency at 1/2 full load and 0.95 power factor

// display the result
disp("Example 3.8 solution");
printf(" \n Efficiency at 3/4 full load and unity power factor \n n_1 = %.2f percent \n", n_1);
printf(" \n Efficiency at 1/2 full load and 0.95 power factor \n n_2 = %.2f percent \n", n_2);
```

**Scilab code Exa 3.9** full load efficiency output kVA

```
// FUNDAMENTALS OF ELECTICAL MACHINES
```

```
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.9


clc;clear; // clears the console and command history

// Given data
P_i = 350   // iron loss of transformer in W
P_cu = 650  // copper loss of transformer in W
kVA = 30    // kVA ratingss of transformer
pf = 0.6    // power factor

// caclulations
P_tloss = (P_i+P_cu)*10^-3   // total full load loss in kW
P_out = kVA*pf               // o/p power at full load in kW
P_in = P_out+P_tloss         // i/p power at full load
n_1 = (P_out/P_in)*100       // efficiency at full load
kVA_out = kVA*sqrt(P_i/P_cu) // o/p kVA corresponding to maximum efficiency
P_01 = kVA_out*pf            // o/p power in W
P_tloss1 = 2*P_i             // maximum efficiency iron loss=copper loss in W
P_in1 = P_01+P_tloss1*10^-3  // i/p power in kW
n_2 = (P_01/P_in1)*100       // efficiency

// display the result
disp("Example 3.9 solution");
printf(" \n Efficiency at full load \n n_1 = %.2f percent \n", n_1);
printf(" \n Out put power \n P_01 = %.1f kW \n",
```

```
P_01);
printf(" \n Efficiency \n n_2 = %.2f percent \n", n_2);
```

**Scilab code Exa 3.10** calculate all day efficiency

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.10


clc;clear; // clears the console and command history

// Given data
kVA = 12     // kVA ratingss of transformer
n = 0.97     // maximum efficiency at unity power factor
t_1 = 8      // time in hours
P_1 = 10     // load in kW
pf_1 = 0.8   // lagging power factor
t_2 = 10     // time in hours
P_2 = 15     // load in kW
pf_2 = 0.90  // leading power factor
t_3 = 6      // time in hours at no load
P_3 = 0      // load in kW

// caclulations
P_01 = kVA*1                       // o/p power at full load and unity factor in kW
P_in1 = (P_01/n)                   // i/p power at full load
```

```
P_tloss = P_in1-P_01                    // total loss in kW
P_cu = P_tloss/2                        // copper loss at 12 kVA P_cu=P_i in kW
P_024 = P_1*t_1+P_2*t_2+P_3*t_3         // all day o.p power in kWh
P_i24 = 24*P_cu                         // iron loss for 24 hours in kWh
P_cu24 = P_cu*t_1*((P_1/pf_1)/P_01)^2+P_cu*t_2*((P_2/pf_2)/P_01)^2 // copper loss for 24 hours
P_in24 = P_024+P_i24+P_cu24             // all day i/p power in kWh
n_allday = (P_024/P_in24)*100           // all day efficiency

// display the result
disp("Example 3.10 solution");
printf(" \n All day efficiency \n n_allday = %.0f percent \n", n_allday);
```

---

**Scilab code Exa 3.11** determine noload circuit resisatnce and reactance

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.11


clc;clear; // clears the console and command history

// Given data
V_1 = 200  // voltage in V
```

```
f = 50       // frequency in Hz
I_0 = 0.6    // single phase current in A
P_0 = 80     // power in W

// caclulations
cos_phi0 = P_0/(V_1*I_0) // power factor
sin_phi0 = 0.74          // from above expression
I_w = I_0*cos_phi0       // working component of no
    load current in A
I_m = I_0*sin_phi0       // working component of no
    load current in A
R_0 = V_1/I_w            // no load circuit
    resistance in ohm
X_0 = V_1/I_m            // no load circuit
    reactance in ohm

// display the result
disp("Example 3.11 solution");
printf(" \n No-load circuit resistance \n R_0 = %.2f
    ohm \n", R_0);
printf(" \n No-load circuit reactance \n X_0 = %.1f
    ohm \n", X_0);
```

**Scilab code Exa 3.12** find equivlent R L Z referred to primary and secondary and VR

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.12


```

```
clc;clear; // clears the console and command history

// Given data
kVA = 25     // kVA ratings of transformer
V1 = 2200    // primary side voltage in V
V2 = 220     // secondary side voltage in V
V_1 = 40     // voltage at high voltage side in V
I_1 = 5      // current at high voltage side in A
P = 150      // power at high voltage side in W

// caclulations
Z_01 = V_1/I_1          // reactance to primary sidec in ohm
R_01 = P/I_1^2          // resistance to primary side in ohm
phi = acosd(R_01/Z_01) // power factor angle
X_01 = Z_01*sind(phi)  // impedance to primary side in ohm
a = V1/V2               // turn ratio
Z_02 = Z_01/a^2         // reactance to secondary side in ohm
R_02 = R_01/a^2         // resistance to secondary side in ohm
X_02 = X_01/a^2         // impedance to secondary side in ohm
I_2 = kVA*10^3/V2       // secondary side current in A
E_2 = V2+I_2*Z_02       // secondary induced voltage in V
VR = ((E_2-V2)/V2)*100 // voltage regulation

// display the result
disp("Example 3.12 solution");
printf(" \n Resistance to primary side \n Z_01 = %.2f ohm \n", Z_01);
printf(" \n Resistance to primary side \n R_01 = %.1f ohm \n", R_01);
printf(" \n Impedance to primary side \n X_01 = %.2f
```

```
    ohm \n", X_01);
printf(" \n Reactance to secondary side \n Z_02 = %
    .2f ohm \n", Z_02);
printf(" \n Resistance to secondary side \n R_02 = %
    .2f ohm \n", R_02);
printf(" \n Impedance to secondary side \n X_02 = %
    .3f ohm \n", X_02);
printf(" \n oltage regulation \n VR = %.0f percent \
    n", VR);
```

**Scilab code Exa 3.13** determine kVA ratings

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.13

clc;clear; // clears the console and command history

// Given data
kVA = 120                    // kVA ratings of
    autotransformer
V1 = 2200                    // lower part voltage of
    autotransformer in V
V2 = 220                     // upper part voltage of
    autotransformer in V

// caclulations
I_pq = kVA*10^3/V2          // currents of respective
    windings
I_qr = kVA*10^3/V1          // currents of respective
    windings
```

```
I_1 = I_pq+I_qr          // current in primary side in
    A
V_2 = V1+V2              // voltage across the
   secondary side in V
kVA_1 = I_1*V1/1000      // kVA ratings of
   autotrnsformer
kVA_2 = I_pq*V_2/1000    // kVA ratings of
   autotrnsformer


// display the result
disp("Example 3.13 solution");
printf(" \n kVA ratings of autotrnsformer \n kVA_1 =
    %.0f kVA \n", kVA_1);
printf(" \n kVA ratings of autotrnsformer \n kVA_2 =
    %.0f kVA \n", kVA_2);
```

**Scilab code Exa 3.14** determine circulating current

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.14

clc;clear; // clears the console and command history

// Given data
E_1 = 500   // secondary induced voltages in V
E_2 = 450   // secondary induced voltages in V
kVA_1 = 100 // kVA ratings of transformer
kVA_2 = 200 // kVA ratings of transformer
Z_1 = 0.05  // impedance of transformer
```

```
Z_2 = 0.08  // impedance of transformer

// caclulations
Z1 =Z_1*E_1/(kVA_1*10^3/E_1)  // actual impedance of 1st transformer in ohm
Z2 = Z_2*E_2/(kVA_1*10^3/E_2) // actual impedance of 2nd transformer in ohm
Z = %i*(Z1+Z2)
I_c = (E_1-E_2)/(Z)           // value of the circulating current

// display the result
disp("Example 3.14 solution");
printf(" \n Value of the circulating current \n I_c = %.3f<%.f A \n", abs(I_c),atand(imag(I_c),real(I_c)));
```

**Scilab code Exa 3.15** determine line voltage and current at secondary side

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.15

clc;clear; // clears the console and command history

// Given data
V_L1 = 11 // three-phase transformer supply in kV
I_P1 = 6  // three-phase transformer current in A
a = 11    // turns ratio

// caclulations
```

```
// delta-wye
V_dP2 = V_L1*10^3/a          // phase voltage at secondary side in V
V_dL2 = sqrt(3)*V_dP2        // line voltage at secondary side in V
I_dP1 = a/sqrt(3)            // phase current in the primary in A
I_dL2 = a*I_dP1              // line current in secondary in A
//Wye delta connection
V_wP1 = V_L1*10^3/sqrt(3)    // phase voltage at primary in V
V_wP2 = V_wP1/a              // phase voltage at secondary in V, V_L2=V_P2
I_wP2 = a*I_P1               // phase current in secondary in A
I_wL2 = sqrt(3)*I_wP2        // line current in secondary in A

// display the result
disp("Example 3.15 solution");
printf(" \n For delta-wye connection-");
printf(" \n Phase voltage at secondary side \n V_dL2 = %.f V \n", V_dL2);
printf(" \n Line voltage at secondary side \n I_dL2 = %.2f A \n", I_dL2);
printf(" \n For wye-delta connection-")
printf(" \n Phase voltage at secondary side \n V_wL2 = %.2f V \n", V_wP2);
printf(" \n Line current in secondary side \n I_wL2 = %.2f A \n", I_wL2);
```

**Scilab code Exa 3.16** determine the per unit values of R L Z

```
// FUNDAMENTALS OF ELECTICAL MACHINES
```

```
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.16


clc;clear; // clears the console and command history

// Given data
V_b = 220   // voltage in V
f = 50      // frequency in Hz
S_b = 600   // power ratings
R = 3       // resistance in ohm
X_L = 5     // inducatnce in ohm
Z = R+X_L   // impedance

// caclulations
I_b = S_b/V_b           // base value of current in A
Z_b = V_b^2/S_b         // base value of impedance in ohm
R_pu = R/Z_b            // per unit value of resistance in ohm
X_Lpu = X_L/Z_b         // per unit value of impedance in ohm
Z_pu = abs(Z)/Z_b       // per unit of value of impedance in ohm
Z_pu = R_pu+%i*X_Lpu    // per unit of value of impedance in ohm NOTE:alternative method

// display the result
disp("Example 3.16 solution");
printf(" \n Per unit value of resistance \n R_pu = %.3f ohm \n", R_pu);
printf(" \n Per unit value of impedance \n X_Lpu = %.3f ohm \n", X_Lpu);
```

```
printf(" \n Per unit of value of impedance \n Z_pu = %.3f<%.f \n", abs(Z_pu),atand(imag(Z_pu),real(Z_pu)));
```

**Scilab code Exa 3.17** find grid bus voltage

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
// Example : 3.17


clc;clear; // clears the console and command history

// Given data
S_b1 = 100     // base apparent power
V_bT11 = 220   // voltage of 1st transformer in kV
V_bT12 = 132   // voltage of 1st transformer in kV
X_T1 = 0.02     // impedance of 1st transformer in pu
S_b2 = 50     // base apparent power
V_bT21 = 132   // voltage of 2nd transformer in kV
V_bT22 = 66    // voltage of 2nd transformer in kV
X_T2 = 0.05    // impedance of 2nd transformer in pu
X_L = 4        // line impedance in ohm
P = 50         // power absorbed in MW
pf = 0.6       // lagging power factor from transmission line
Z_p = 0.32*%i      //Reactance of transformer in ohm

// caclulations
S_b = S_b1                          //Base power(MW)
```

```
V_b = V_bT11                                   //Base voltage(kV)
a = V_bT11/V_bT12                                  //turn ratio for 1st transformer
Vb_line = (V_bT11/a)                          // base voltage of line in kV
Zb_line = Vb_line^2/S_b1                           //base impedance of line in ohm
Xpu_line = X_L/Zb_line                             //per unit reactance of line
Xpu_T1 = X_T1*(V_bT11/V_b)^2*(S_b/S_b1)   // 1st grid transformer ,the per unit reactance
Vb_load = (V_bT12/(V_bT12/V_bT22))                 //load side base voltage in kV
Xpu_load = X_T2*(V_bT22/Vb_load)^2*(S_b/S_b2) // second load transformer ,the per  unit reactance
I_b = S_b*1000/(sqrt(3)*Vb_load)                        // base current
I_L = S_b2*1000/(sqrt(3)*V_bT22*pf)            //actualcurrent in load in A
I_Lpu = I_L/I_b                                    //per unit value of the load
V_L = V_bT22/V_bT22      //per unit value of the voltage at the load terminal(bus4)
V_gb = I_Lpu*exp(%i*acos(pf))*Z_p + 1 // per unit value of bus voltage
V_gba = abs(V_gb)*V_bT11 // actual value of grid to bus voltage

// display the result
disp("Example 3.17 solution");
printf(" \n Actual value of grid to bus voltage \n V_gba = %.2f kV \n", V_gba);
```

# Chapter 4

# DIRECT CURRENT GENERATORS

**Scilab code Exa 4.2** determine induced voltage

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 4 : DIRECT CURRENT GENERATORS
// Example : 4.2


clc;clear; // clears the console and command history

// Given data
l = 0.65 // length of conductor in m
v = 35   // speed in m/s
B = 0.8  // magnetic flux density in T

// caclulations
e = B*l*v // induced voltage in V

```

```
// display the result
disp("Example 4.2 solution");
printf(" \n Induced voltage \n e = %.1f V \n", e);
```

**Scilab code Exa 4.3** determine induced voltage

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 4 : DIRECT CURRENT GENERATORS
// Example : 4.3


clc;clear; // clears the console and command history

// Given data
l = 1.5     // length of conductor in m
v = 20      // speed in m/s
B = 0.9     // magnetic flux density in Wb/m^2
teta = 35   // angle of rotation in degree

// caclulations
e = B*l*v*sind(teta) // induced voltage in V

// display the result
disp("Example 4.3 solution");
printf(" \n Induced voltage \n e = %.1f V \n", e);
```

**Scilab code Exa 4.4** calculate generated emf

```
// FUNDAMENTALS OF ELECTICAL MACHINES
```

```
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 4 : DIRECT CURRENT GENERATORS
// Example : 4.4


clc;clear; // clears the console and command history

// Given data
P = 4          // number of poles
Z = 40*10      // number of conductors
phi = 0.02     // flux per pole in Wb
N = 1200       // speed in rpm

// caclulations
A = P/2
E_g = (P*phi*Z*N)/(60*A) // generated voltage in V

// display the result
disp("Example 4.4 solution");
printf("\n Generated voltage \n E_g = %.0f V \n",
    E_g);
```

**Scilab code Exa 4.5** find electromagnetic torque

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 4 : DIRECT CURRENT GENERATORS
// Example : 4.5

```

```

clc;clear; // clears the console and command history

// Given data
P = 6 // number of poles
Z = 600 // number of conductors
phi = 0.05 // flux per pole in Wb
N = 1000 // speed in rpm
I_a = 120 // generetor supply current in A

// caclulations
A=6 // for lap-wound A=P
E_g = (P*phi*Z*N)/(60*A) // generated voltage in V
T_em = ((P*Z*phi)/(2*%pi*A))*I_a // electromagnetic
    torque in N-m


// display the result
disp("Example 4.5 solution");
printf(" \n Generated voltage \n E_g = %.0f V \n",
    E_g);
printf(" \n Electromagnetic torque \n T_em = %.2f N-
    m \n", T_em);
```

**Scilab code Exa 4.6** calculate the generated voltage

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 4 : DIRECT CURRENT GENERATORS
// Example : 4.6


```

```
clc;clear; // clears the console and command history

// Given data
V_t = 220     // shunt generator voltage in V
I = 250       // shunt generator current in A
R_sh = 50     // shunt field resistance in ohm
R_a = 0.02    // armature resistance in ohm

// caclulations
I_sh = V_t/R_sh    // shunt field current in A
I_a = I+I_sh       // armature current in A
E_g = V_t+I_a*R_a // generated voltage in V


// display the result
disp("Example 4.6 solution");
printf(" \n Generated voltage \n E_g = %.2f V \n",
   E_g);
```

**Scilab code Exa 4.7** determine generated emf

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 4 : DIRECT CURRENT GENERATORS
// Example : 4.7


clc;clear; // clears the console and command history

// Given data
E = 25 // power of compound generator in kW
V_t = 220 // terminal voltage in V
```

```
R_se = 0.05     // series resistance in ohm
R_sh = 55     // shunt field resistance in ohm
R_a = 0.07    // armature resistance in ohm
brush_drop = 1 // voltage drop per brush in V

// caclulations
I_L = E*10^3/V_t // load current in A
I_sh1 = V_t/R_sh // shunt field current in A
I_a1 = I_sh1+I_L // armature current in A
E_g1 = V_t+I_a1*(R_a+R_se)+2*brush_drop // generator voltage in V
V_ab = V_t+I_L*R_se // voltage across the shunt field in V for short shunt generator
I_sh2 = V_ab/R_sh // current in the shunt field in A for short shunt generator
I_a2 = I_sh2+I_L // armature current in A for short shunt generator
E_g2 = V_ab+I_a2*R_a+2*brush_drop // generator voltage in V for short shunt generator

// display the result
disp("Example 4.7 solution");
printf(" \n Generated emf when generatar is connected in long shunt \n E_g1 = %.f V \n", E_g1);
printf(" \n Generated emf when generatar is connected in short shunt \n E_g2 = %.1f V \n", E_g2);
```

**Scilab code Exa 4.8** determine the generated volatge and PD

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION
```

```
// Chapter 4 : DIRECT CURRENT GENERATORS
// Example : 4.8


clc;clear; // clears the console and command history

// Given data
V_t = 220   // shunt generator voltage in V
I_L = 146   // generator delivering current in A
R_sh = 50   // shunt field resistance in ohm
R_a = 0.012 // armature resistance in ohm
R_s = 0.02  // series field resistance in ohm
R_d = 0.03  // diverter field resistance in ohm

// caclulations
I_sh = V_t/R_sh              // shunt field current in A
I_a = I_L+I_sh               // armature current in A
R_com = R_s*R_d/(R_s+R_d)    // combined resistance in ohm
E_g = V_t+(I_a*(R_a+R_com)) // generated voltage in V
P_lsd = I_a^2*R_com          // power loss in series and diverter in W
P_la = I_a^2*R_com          // power loss in the armature circuit resistance in W
P_lsh =  V_t*I_sh           // power loss in shunt field resistance in W
P_dl = I_L*V_t              // power delivered in W

// display the result
disp("Example 4.8 solution");
printf(" \n Generated voltage \n E_g = %.1f V \n", E_g);
printf(" \n Power distribution \n P_dl = %.0f W \n", P_dl);
```

**Scilab code Exa 4.9** determine ATc and ATd

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 4 : DIRECT CURRENT GENERATORS
// Example : 4.9

clc;clear; // clears the console and command history

// Given data
P = 4      // number of poles
Z = 500    // number of conductors
I_a = 30   // generetor supply current in A
alpa = 6   // brushes displaced angle in degree

// caclulations
A = P/2                 // for wave connected A=P/2
I_c = I_a/A             // current per conductor in A
AT_d = Z*I_c*alpa/360 // demagnetizing ampere turns
    per pole in At
AT_c = Z*I_c*((1/(2*P))-(alpa/360)) // cross
    magnetizing ampere turn per pole in At


// display the result
disp("Example 4.9 solution");
printf(" \n Demagnetizing ampere turns per pole \n
    AT_d = %.1f At \n", AT_d );
printf(" \n Cross magnetizing ampere turn per pole \
    n AT_c = %.1f At \n", AT_c );
```

**Scilab code Exa 4.10** determine flux per pole

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 4 : DIRECT CURRENT GENERATORS
// Example : 4.10

clc;clear; // clears the console and command history

// Given data
Pw = 12           // power in kW
P = 4             // number of poles
Z = 500           // number of conductors
V_t = 250         // generator voltage in V
N = 1000          // speed in rpm
P_cu = 600        // full load copper loss in W
brush_drop = 2    // brush drop in V

// caclulations
A = 4                             // for lab wound A=P
I_a = Pw*10^3/V_t                 // armature current in
   A
R_a = P_cu/I_a^2                  // from copper loss
   equestion R_a in ohm
E_g = V_t+I_a*R_a+brush_drop // generated voltage in
    V
phi = E_g*60*A/(P*Z*N)       // flux per pole in Wb


// display the result
disp("Example 4.10 solution");
```

```
printf(" \n Flux per pole \n phi = %.3f Wb \n", phi
   );
```

**Scilab code Exa 4.11** determine induced voltage and efficiency

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 4 : DIRECT CURRENT GENERATORS
// Example : 4.11

clc;clear; // clears the console and command history

// Given data
P = 4                  // number of poles
I_L = 25               // generator delivering current in
     A
V_t = 230              // generator terminal voltage in V
R_a = 0.2              // armature resistance in ohm
R_sh = 55              // shunt field resistance in ohm
brush_drop = 1         // brush drop in V

// caclulations
I_sh = V_t/R_sh                  // shunt field current
   in A
I_a = I_L+I_sh                   // armature current in
   A
E_g = V_t+I_a*R_a+brush_drop // induced voltage in V
P_arm = E_g*I_a                  // power generated in
   armature in W
P_L = V_t*I_L                    // power absorbed by
   load in W
n = (P_L/P_arm)*100              // efficiency
```

```

// display the result
disp("Example 4.11 solution");
printf(" \n Induced voltage \n E_g = %.1f V \n", E_g
    );
printf(" \n Efficiency \n n = %.1f percent \n", n );
```

# Chapter 5

# DIRECT CURRENT MOTORS

**Scilab code Exa 5.1** determine magnitude of force

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 5 : REVIEW OF ELECRTIC CIRCUITS
// Example : 5.1

clc;clear; // clears the console and command history

// Given data
l = 10 // conductor length in m
B = 0.56 // magnetic flux density in T
I = 2 // current through conductor in A

// caclulations
F = B*I*l // magnitude of force in N

// display the result
```

```
disp("Example 5.1 solution");
printf(" \n Magnitude of force \n F = %.1f N ", F);
```

**Scilab code Exa 5.2** determine value of back emf and mechanical power developed

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 5 : DIRECT CURRENT MOTORS
// Example : 5.2

clc;clear; // clears the console and command history
// Given data
I = 20      // total current in A
V_t = 250;  // supply voltage in volt
R_sh = 200;  // shunt field resistance in
R_a = 0.3;   // armature resistance in

// caclulations
 I_sh = V_t/R_sh;      // shunt field current in A
 I_a = I-I_sh'         // armature current
 E_b = V_t - R_a*I_a;  // the back emf in V
 P_md = E_b*I_a;       // mechanical power developed in W

 // display the result
disp("Example 5.2 solution");
printf("\n The back emf is \n E_b = %.1f V \n\n",E_b);
printf(" \n Mechanical power developed is \n P_md = %.1f W" ,P_md );
printf(" \n NOTE : error in calculation they has
```

```
taken I_a=18.13, instead of I_a=18.75");
```

**Scilab code Exa 5.3** determine change in back emf

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 5 : DIRECT CURRENT MOTORS
// Example : 5.3

clc;clear; // clears the console and command history
// Given data
R_a = 0.7;      // armature circuit resistance in
V_t = 5;        // applied voltage in V
I_anl = 5;      // no-load armature current in A
I_afl = 35;     // full-load armature current in A


// caclulations
E_bnl =  V_t - R_a*I_anl;   // back emf under no-
    load in V
E_bfl =  V_t - R_a*I_afl;   // back emf under full-
    load in V
E_bc =  E_bnl - E_bfl;      // change in back emf
    from  no-load to full load in V

 // display the result
disp("Example 5.3 solution");
printf("\n The change in back emf is \n E_bc = %d V
    ",E_bc );
```

**Scilab code Exa 5.4** find torque developed by armature

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 5 : DIRECT CURRENT MOTORS
// Example : 5.4

clc;clear; // clears the console and command history
// Given data
I = 40;        // current in A
V_t = 230;     // supply in V
N = 1100;      // speed in rpm
R_a = 0.25;    // armature resistance in
R_sh = 230;    //shunt resistance in


// caclulations
I_sh = V_t/R_sh;        // shunt field current in A
I_a = I - 1;            // armature current in A
E_b = V_t - I_a*R_a;    // back emf
T_a = 9.55*E_b*I_a/N;   //  amrature torque in N-m

 // display the result
disp("Example 5.4 solution");
printf("\n The armature torque is \n T_a = %.2f N-m
   ",T_a );
```

**Scilab code Exa 5.5** find speed and shaft torque

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
```

```
// SECOND EDITION

// Chapter 5 : DIRECT CURRENT MOTORS
// Example : 5.5

clc;clear; // clears the console and command history

// Given data
P = 6             // number of poles
V_t = 230         // supply to shunt motor in V
Z = 450           // number of conductors
R_a = 0.8         // armature resistance in ohm
I = 30             // supply current in A
P_0 = 5560         // out put power in W
I_F = 3            // current through field winding
phi = 25*10^-3     // flux per pole in Wb

// caclulations
A = 6                        // for lap wond A=P
I_a = I-I_F                  // armature current in A
E_b = V_t-I_a*R_a            // back emf in V
N = 60*A*E_b/(P*Z*phi)       // speed in rpm
T_sh = 9.55*P_0/N            // shaft torque in N-m

// display the result
disp("Example 5.5 solution");
printf(" speed \n N = %.1f rpm \n", N );
printf(" shaft torque \n T_sh = %.1f N-m \n", T_sh )
    ;
```

**Scilab code Exa 5.6** determine the motor speed under load condition

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
```

```
// SECOND EDITION

// Chapter 5 : DIRECT CURRENT MOTORS
// Example : 5.6

clc;clear; // clears the console and command history

// Given data
I_L1 = 5        // dc shunt motor current
V_t = 230       // terminal voltage in V
N_1 = 1000      // speed in rpm
R_a = 0.2       // armature resistance in ohm
R_F = 230       // field resistance in ohm
I_L2 = 30       // dc shunt motor current

// caclulations
// at noload condition
I_sh = V_t/R_F          // shunt field current in A
I_a1 = I_L1-I_sh        // armature current in A
E_b1 = V_t-I_a1*R_a     // back emf in V
// under load condition
I_a2 = I_L2-I_sh        // armature current in A
E_b2 = V_t-I_a2*R_a     // back emf in V
N_2 = (E_b2/E_b1)*N_1    // motor speed under load
   condtion in rpm

// display the result
disp("Example 5.6 solution");
printf(" \n Speed under load condition \n N_2 = %.1f
    rpm \n", N_2 );
```

**Scilab code Exa 5.7** find the speed of the motor

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
```

```
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 5 : DIRECT CURRENT MOTORS
// Example : 5.7

clc;clear; // clears the console and command history

// Given data
I_a1 = 65      // supply current to dc series motor in A
V_t = 230      // supply voltage in V
N_1 = 900      // speed in rpm
R_a = 0.2      // armature resistance in ohm
R_sh = 0.25    // field resistance in ohm
I_a2 = 15      // supply current to dc series motor in A
// phi_2 = 0.4*phi_1 value of flux


// caclulations
E_b1 = V_t-I_a1*(R_a+R_sh)      // initial back emf in V
E_b2 = V_t-I_a2*(R_a+R_sh)      // final back emf in V
// phi_2 = 0.4*phi_1 value of flux
N_2 = (E_b2/E_b1)*(2.5)*N_1     // motor speed when line current is 15A in rpm

// display the result
disp("Example 5.7 solution");
printf(" \n motor speed when line current is 15A \n N_2 = %.0f rpm \n", N_2 );

printf(" NOTE: in question they given I_a1=56A , but in solution they took I_a1=65A");
```

**Scilab code Exa 5.8** calculate the iron and friction loss and efficiency

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 5 : DIRECT CURRENT MOTORS
// Example : 5.8

clc;clear; // clears the console and command history

// Given data
I_L1 = 5      // dc shunt motor current in A
V_t = 230     // supply voltage in V
R_a = 0.25    // armature resistance in ohm
R_sh = 115    // field resistance in ohm
I_L2 = 40     // dc shunt motor current in A


// caclulations
// at noload condition
P_in1 = V_t*I_L1                          // input
    power in W
I_sh = V_t/R_sh                           // shunt
    field current in A
I_a1 = I_L1-I_sh                          // armature
    current in A
P_acu1 = I_a1^2*R_a                       // armature
    copper loss in W
P_shcu = I_sh^2*R_sh                      //shunt
    field copper loss in W
P_iron_friction = P_in1-(P_acu1+P_shcu) // iron and
    friction losses in W
```

```
// under load condition
I_a2 = I_L2-I_sh                        // armature
    current in A
P_acu2 = I_a2^2*R_a                     // armature
    copper loss in W
P_loss = P_iron_friction+P_shcu+P_acu2  // total
    losses in W
P_in2 = V_t*I_L2                        // input
    power in W
P_0 = P_in2-P_loss                      // output
    power in W
n = (P_0/P_in2)*100                     //
    efficiency in percent

// display the result
disp("Example 5.8 solution");
printf(" \n iron and friction losses \n
    P_iron_friction = %.2f W \n", P_iron_friction );
printf(" \n efficiency \n n = %.0f percent \n", n)
```

**Scilab code Exa 5.9** determine Pcu Ta Tsh and efficiency

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 5 : DIRECT CURRENT MOTORS
// Example : 5.9

clc;clear; // clears the console and command history

// Given data
I_L= 80          // dc shunt motor current in A
V_t = 220        // supply voltage in V
```

```
N = 800             // speed in rpm
R_a = 0.1           // armature resistance in ohm
R_sh = 50           // shunt field resistance in ohm
P_if = 1600         // iron and friction losses in W

// caclulations
I_sh = V_t/R_sh           // shunt field current in A
I_a = I_L-I_sh            // armature current
E_b = V_t-I_a*R_a         // back emf in V
P_in = V_t*I_L            // i/p power in W
P_md = E_b*I_a            // mechanical power developed
    in the armature in W
P_cu = P_in-P_md          // copper loss in W
T_a = 9.55*(E_b*I_a)/N    // armature torque in N-m
P_0 = P_md-P_if           // o/p power in W
T_sh = 9.55*(P_0/N)       // shaft torque in N-m
n= (P_0/P_in)*100         // efficiency

// display the result
disp("Example 5.9 solution");
printf(" \n Copper loss \n P_cu = %.2f W \n", P_cu);
printf(" \n Armature torque \n T_a = %.2f N-m \n",
   T_a);
printf(" \n Shaft torque \n T_sh = %.2f N-m \n",
   T_sh);
printf(" \n Efficiency \n n = %.0f percent \n", n);
```

# Chapter 6

# CONTROL AND STARTING OF A DC MOTOR

**Scilab code Exa 6.1** find value of resistance

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 6 : CONTROL AND STARTING OF A DC MOTORS
// Example : 6.1

clc;clear; // clears the console and command history

// Given data
V_t = 220     // supply voltage in V
I_a1 = 10     // dc shunt motor armature current in A
N_1 = 900     // speed in rpm
R_a = 1       // armature resistance in ohm
N_2 = 500     // speed in rpm

// caclulations
E_b1 = V_t-I_a1*R_a           // initial back emf in
```

```
V
R = (E_b1/10)*(1-(N_2/N_1))    // additional value of
     resistance in ohm

// display the result
disp("Example 6.1 solution");
printf(" \n additional value of resistance \n R = %
    .1f ohm \n", R);
```

**Scilab code Exa 6.2** find the speed at full load and half load torque

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 6 : CONTROL AND STARTING OF A DC MOTORS
// Example : 6.2

clc;clear; // clears the console and command history

// Given data
V_t = 230    // supply voltage in V
I_a1 = 15    // dc shunt motor armature current in A
N_1 = 650    // speed in rpm
R_a = 0.4    // armature resistance in ohm
R = 1        // variable resistance in series with
    the armature

// caclulations
// at full load
E_b1 = V_t-I_a1*R_a       // initial back emf in V
E_b2 = V_t-I_a1*(R+R_a)   // final back emf in V
N_2 = N_1*(E_b2/E_b1)     // speed at full load in
    rpm
```

```
// at half load
I_a21 = I_a1/2                    // armature current in A
E_b21 = V_t-I_a21*(R+R_a)         // back emf in V
N_21 = N_1*(E_b21/E_b1)           // speed at half load torque in rpm



// display the result
disp("Example 6.2 solution");
printf(" \n speed at full load \n N_2 = %.1f rpm \n", N_2);
printf(" \n speed at half load torque \n N_21 = %.1f rpm \n", N_21);
```

**Scilab code Exa 6.3** find new speed

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 6 : CONTROL AND STARTING OF A DC MOTORS
// Example : 6.3

clc;clear; // clears the console and command history

// Given data
V_t = 230     // supply voltage series motor in V
R_a = 0.2     // armature resistance in ohm
R_f = 0.2     // field resistance in ohm
I_a1 = 10     // dc series motor armature current in A
```

```
N = 1000       // speed in rpm
I_a2 = 8       // dc series motor armature current in
    A when R=2 ohm
R_2 = 2        //  series resistance in ohm
R_1 = 0        // series resistance in ohm
// phi2 = 0.8*phi1
// phi1/phi2 = 1.25

// caclulations
R_a1 = R_a+R_1              // new armature resistance
    in ohm
E_b1 = V_t-I_a1*R_a1        // back emf in V
R_a2 = R_a+R_2              // new armature resistance
    in ohm
E_b2 = V_t-I_a2*R_a2        // back emf in V
N_2 = (E_b2/E_b1)*1.25*N   // new speed in rpm

// display the result
disp("Example 6.3 solution");
printf(" \n New speed \n N_2 = %.1f rpm \n", N_2);
```

**Scilab code Exa 6.4** determine the speed

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 6 : CONTROL AND STARTING OF A DC MOTORS
// Example : 6.4

clc;clear; // clears the console and command history

// Given data
N = 1100          // speed of dc series motor in rpm
```

```
P = 4                // number of poles
I_a1 = 15            // supply current to dc series motor in A
V_t = 220            // supply voltage in V
R_a = 0.9            // series armature resistance in ohm
R_se1 = 0.6          // series field resistance in ohm
I_a2 = 25            // supply current to dc series motor in A
// phi2 = 0.8*phi1
// phi1/phi2 = 1.25

// caclulations
E_b1 = V_t-I_a1*(R_a+R_se1)     // back emf in V
R_se2 = 0.6/4                   // value of resistance per path in ohm
E_b2 = V_t-I_a2*(R_a+R_se2)    // back emf in V
N_2 = (E_b2/E_b1)*1.25*N       // new speed in rpm

// display the result
disp("Example 6.4 solution");
printf(" \n New speed \n N_2 = %.1f rpm \n", N_2);
```

**Scilab code Exa 6.5** calculate value of resistance

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 6 : CONTROL AND STARTING OF A DC MOTORS
// Example : 6.5

clc;clear; // clears the console and command history

```

```
// Given data
V_t = 230           // shunt motor supply voltage in V
R_a = 0.4           // armature resistance in ohm
I_a = 30            // armature current in A
n = 3               // number of steps

// caclulations
R_1 = V_t/I_a          // maximum value of current in ohm
k = (R_1/R_a)^(1/3)   // constant
R_2 = R_1/k           // other value of resistance in ohm
R_3 = R_2/k           // other value of resistance in ohm
R_4 = R_3/k           // other value of resistance in ohm
R_1step = R_1-R_2     // resistance of the 1st step in ohm
R_2step = R_2-R_3     // resistance of the 1st step in ohm
R_3step = R_3-R_4     // resistance of the 1st step in ohm

// display the result
disp("Example 6.5 solution");
printf(" \n resistance of the 1st step in ohm \n R_1step = %.1f ohm \n", R_1step);
printf(" \n resistance of the 2nd step in ohm \n R_2step = %.1f ohm \n", R_2step);
printf(" \n resistance of the 3rd step in ohm \n R_3step = %.2f ohm \n", R_3step);
```

**Scilab code Exa 6.6** find the value of resistance

```
// FUNDAMENTALS OF ELECTICAL MACHINES
```

```
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 6 : CONTROL AND STARTING OF A DC MOTORS
// Example : 6.6

clc;clear; // clears the console and command history

// Given data
V_t = 220       // shunt motor supply voltage in V
P_0 = 3550      // o/p power in W
n = 0.85        // efficiency
// condition given : starting current of the motor should not exceed twice the load current

// caclulations
P_in = P_0/n             // i/p power in W
P_tloss = P_in-P_0       // total loss in W
I_a = P_in/V_t           // neglet the field current then armature current in A
P_cu = P_tloss/2         // copper loss in W
R_a = P_cu/I_a^2         // armature resistance in ohm
I_1 = 2*I_a              // maximum starting current in A
R_1 = V_t/I_1            // maximum resistance in ohm
k = (R_1/R_a)^(1/4)      // constant
R_2 = R_1/k              // other value of resistance in ohm
R_3 = R_2/k              // other value of resistance in ohm
R_4 = R_3/k              // other value of resistance in ohm
R_5 = R_4/k              // other value of resistance in ohm
R_1step = R_1-R_2        // resistance of the 1st step in ohm
R_2step = R_2-R_3        // resistance of the 1st step
```

```
    in ohm
R_3step = R_3-R_4      // resistance of the 1st step
    in ohm
R_4step = R_4-R_5      // resistance of the 1st step
    in ohm

// display the result
disp("Example 6.6 solution");
printf(" \n resistance of the 1st step in ohm \n
    R_1step = %.1f ohm \n", R_1step);
printf(" \n resistance of the 2nd step in ohm \n
    R_2step = %.2f ohm \n", R_2step);
printf(" \n resistance of the 3rd step in ohm \n
    R_3step = %.2f ohm \n", R_3step);
printf(" \n resistance of the 4th step in ohm \n
    R_3step = %.2f ohm \n", R_4step);
```

# Chapter 7

# THREE PHASE INDUCTION MOTOR

**Scilab code Exa 7.1** determine synchronous speed slip and rotor frequency

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 7 : THREE-PHASE INDUCTION MOTOR
// Example : 7.1

clc;clear; // clears the console and command history

// Given data
V = 230      // supply voltage in V
P = 4       // number of poles
f = 50      // frequecny Hz
N_l = 1445  // speed in rpm

// caclulations
N_s = 120*f/P      // synchronous speed in rpm
```

```
s = (N_s-N_l)/N_s   // slip
f_r = s*f           // rotor frequency in Hz

// display the result
disp("Example 7.1 solution");
printf(" \n Synchronous speed \n N_s = %.1f rpm \n",
    N_s);
printf(" \n Slip \n s = %.4f \n", s);
printf(" \n Rotor frequency \n f_r = %.1f Hz \n",
   f_r);
```

---

**Scilab code Exa 7.2** determine synchronous speed rotor frequency and rotor voltage

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 7 : THREE-PHASE INDUCTION MOTOR
// Example : 7.2

clc;clear; // clears the console and command history

// Given data
E_BR = 120     // voltage found under blocked
   condition in V
P = 4          // number of poles
f = 50         // frequecny in Hz
N_l = 1450     // speed in rpm

// caclulations
N_s = 120*f/P      // synchronous speed in rpm
s = (N_s-N_l)/N_s  // slip
f_r = s*f          // rotor frequency in Hz
```

```
E_r = s*E_BR       // Rotor voltage in V

// display the result
disp("Example 7.2 solution");
printf(" \n Synchronous speed \n N_s = %.1f rpm \n",
    N_s);
printf(" \n Slip \n s = %.3f \n", s);
printf(" \n Rotor frequency \n f_r = %.2f Hz \n",
   f_r);
printf(" \n Rotor voltage \n E_r = %.2f V \n", E_r);
```

**Scilab code Exa 7.3** find the value of starting torque and current

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 7 : THREE-PHASE INDUCTION MOTOR
// Example : 7.3

clc;clear; // clears the console and command history

// Given data
V_0 = 230 // supply voltage in V
P = 4     // number of poles
T_0 = 230 // original torque in N-m
V_s = 150 // stator voltage in V
I_0 = 560 // starting cuurent in A

// caclulations
T_st = (V_s/V_0)^2*T_0 // starting torque in N-m
I_st = I_0*(V_s/V_0)   // starting current in A

// display the result
```

```
disp("Example 7.3 solution");
printf(" \n Starting torque \n T_st = %.1f N-m \n", T_st);
printf(" \n Starting current \n I_st = %.1f A \n", I_st);
```

**Scilab code Exa 7.4** find speed and ratio

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 7 : THREE-PHASE INDUCTION MOTOR
// Example : 7.4

clc;clear; // clears the console and command history

// Given data
f = 50          // frequency in Hz
P = 8           // number of poles
a = 0.03        // full load slip
R_2 = 0.01      // rotor resistance in ohm
X_2 = 0.1       // standstill resistance in ohm

// caclulations
N_s = 120*f/P                // synchronous speed in rpm
s = R_2/X_2                  // slip at maximum torque
N_l = (1-s)*N_s              // rotor maximum speed in rpm
T = (2*a*s)/(s^2+a^2)        // ratio of full load torque to maximum torque
T_ratio = 1/T                // ratio of maximum torque to full load torque
```

```

// display the result
disp("Example 7.4 solution");
printf(" \n Rotor speed at maximum torque \n N_l = %
    .0f rpm \n", N_l);
printf(" \n Ratio of maximum torque to full load
    torque \n T_ratio = %.2f \n", T_ratio);
```

**Scilab code Exa 7.5** determine slip synchronous speed shaft speed and mechanical power

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 7 : THREE-PHASE INDUCTION MOTOR
// Example : 7.5

clc;clear; // clears the console and command history

// Given data
V = 440      // supply voltage in V
P = 4        // number of poles
P_ag = 1500  // rotor i/p in kW
P_rcu = 250  // copper loss in W
f = 50       // frequency in Hz

// caclulations
s = P_rcu/P_ag          // slip
N_s = 120*f/P           // synchronous speed in rpm
N_l = (1-s)*N_s         // rotor maximum speed in rpm
P_mech = (1-s)*P_ag     // mechanical power developed
    in W

```

```
// display the result
disp("Example 7.5 solution");
printf(" \n Slip \n s = %.3f \n", s);
printf(" \n Synchronous speed \n N_s = %.f rpm \n",
    N_s);
printf(" \n Rotor speed \n N_l = %.0f rpm \n", N_l);
printf(" \n Mecahnical power developed \n P_mech = %
    .0f W \n", P_mech);

// NOTE : small change in answer instead 1245 got
    1250W
```

**Scilab code Exa 7.6** determine max mechanical power Tmax and slip

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 7 : THREE-PHASE INDUCTION MOTOR
// Example : 7.6

clc;clear; // clears the console and command history

// Given data
V_1 = 150               // supply voltage in V
P = 4                   // number of poles
f = 50                  // frequency in Hz
Z_1 = 0.12+%i*0.16      // per phase standstill
    stator impedance in ohm
Z_2 = 0.22+%i*0.28      // per phase standstill rotor
     impedance in ohm
R_2 = real(Z_2)              // from Z_2

// caclulations
```

```
Z_eq = Z_1+Z_2                                    // equivalent impedance in ohm
P_mech =  3*V_1^2/(2*(R_2+abs(Z_eq)))   // maximum mechanical power developed in W
s_mp = R_2/(abs(Z_eq)+R_2)                        // slip
W_s = 2*%pi*2*f/P                                 // since N_s = f/(P/2) and W_s = 2*%pi*N_s
W = (1-s_mp)*W_s                                  // speed of rotor in rad/s
T_mxm = P_mech/W                                  // miximum torque in N-m

// display the result
disp("Example 7.6 solution");
printf(" \n maximum mechanical power developed \n P_mech = %.f W \n", P_mech);
printf(" \n Maximum torque \n T_mxm = %.3f N-m \n", T_mxm);
printf(" \n Maximum slip \n s_mp = %.2f \n", s_mp);
printf(" \n NOTE : Error in calculation of P_mech and T_mxm ");
```

**Scilab code Exa 7.7** determine slip rotor cu loss shaft motor efficiency

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 7 : THREE-PHASE INDUCTION MOTOR
// Example : 7.7

clc;clear; // clears the console and command history

// Given data
```

```
V = 440            // supply voltage in V
P = 6              // number of poles
f = 50             // frequency in Hz
P_a = 45000        // i/p power
N_l = 900          // speed in rpm
P_tloss = 2000     // total power loss in W

// caclulations
N_s = 120*f/P              // synchronous speed in rpm
s = (N_s-N_l)/N_s          // slip
P_ag = (P_a-P_tloss)       // air gap power in W
P_rcu = s*P_ag             // rotor copper loss in W
P_mech = P_ag-P_rcu        // mechanical power in W
P_0 = P_mech-3000          // o/p power in W
n = (P_0/P_ag)*100         // efficiency since n = P_o
    /P_in

// display the result
disp("Example 7.7 solution");
printf(" \n Slip \n s = %.1f  \n", s );
printf(" \n Rotor copper loss  \n P_rcu = %.f W \n",
    P_rcu );
printf(" \n Out put power \n P_0 = %.f W \n", P_0 );
printf(" \n Efficiency  \n n = %.f percent \n", n );
```

**Scilab code Exa 7.8** determine length

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 7 : THREE-PHASE INDUCTION MOTOR
// Example : 7.8

```

```
clc;clear; // clears the console and command history

// Given data
v_s = 120       // train speed in km/h
f = 50          // frequency in Hz

// caclulations
v_s = 120*1000/(60*60)     // train speed in m/s
w = v_s/(2*f)              // length of the pole-
    pitch in m

// display the result
disp("Example 7.8 solution");
printf(" \n Length of the pole-pitch lenear
    induction motor \n w = %.2f m \n", w );
```

# Chapter 8

# STARTING CONTROL AND TESTING OF AN INDUCTION MOTOR

**Scilab code Exa 8.2** determine percentage taping on the autotransformer

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 8 : STARTING, CONTROL AND TESTING OF AN
   INDUCTION MOTOR
// Example : 8.2

clc;clear; // clears the console and command history

// Given data
T_ratio = 1/2    // T_st/T_f ratio of starting
   torque to full load torque
s_f = 0.03       // full load slip
I_ratio = 5      // I_sc/I_f ratio pf short circuit
   current to full load current
```

```

// caclulations
x = (1/I_ratio)*sqrt(T_ratio*(1/s_f)) // percentage
    of tapping

// display the result
disp("Example 8.2 solution");
printf(" \n Percentage of tapping \n x = %.3f \n", x
    );
```

**Scilab code Exa 8.3** determine the full load current

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 8 : STARTING, CONTROL AND TESTING OF AN
    INDUCTION MOTOR
// Example : 8.3

clc;clear; // clears the console and command history

// Given data
T_ratio = 0.25     // T_st/T_f ratio of starting
    torque to full load torque
I_ratio = 3.6      //  I_sc/I_f ratio pf short
    circuit current to full load current
                   // since given I_sc = 3*1.2*I_f

// caclulations
s_f = T_ratio*3/(I_ratio)^2 // full load slip

// display the result
disp("Example 8.3 solution");
```

```
printf(" \n Full load slip \n s_f = %.2f \n", s_f );
```

**Scilab code Exa 8.4** determine torques

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 8 : STARTING, CONTROL AND TESTING OF AN
   INDUCTION MOTOR
// Example : 8.4

clc;clear; // clears the console and command history

// Given data
Z_icr = 0.04+%i*0.5              // inner cage impedance
    per phase at standstill
Z_ocr = 0.4+%i*0.2               // outer cage impedance
    per phase at standstill
V = 120                          // per phase rotor
   induced voltage at standstill in V

// caclulations
Z_com = (Z_icr*Z_ocr)/(Z_icr+Z_ocr)       // combined
    impedance in ohm
I_2 = V/abs(Z_com)                         // rotor
   current per phase in A
R_2 = real(Z_com)                          // combined
    rotor resistance in ohm from Z_com
T = I_2^2*R_2                              // torque
   at stand still condition in syn.watt
s = 0.06                                  // slip of 6
   %
Z_icr6 = 0.04/s+%i*0.5           // inner cage
```

```
impedance per phase at standstill
Z_ocr6 = 0.4/s+%i*0.2              // outer cage impedance per phase at standstill
Z_com6 = ((Z_icr6)*Z_ocr6)/((Z_icr6)+Z_ocr6) // combined impedance in ohm at 6% slip
I2_6 = V/abs(Z_com6)                          // rotor current per phase in A at 6% slip
R2_6 = real(Z_com6)                              // combined rotor resistance in ohm from Z_com6
T6 = I2_6^2*R2_6                              // torque at stand still condition in syn.watt

// display the result
disp("Example 8.4 solution");
printf(" \n Torque at stand still condition \n T = %.2f syn.watt \n", T );
printf(" \n Torque at stand 6 percent slip \n T6 = %.2f syn.watt \n", T6 );
```

**Scilab code Exa 8.5** determine the equivalent parameters

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 8 : STARTING, CONTROL AND TESTING OF AN INDUCTION MOTOR
// Example : 8.5

clc;clear; // clears the console and command history

// Given data
V = 210        // supply voltage in V
f = 50         // supply frequency in Hz
```

```
P = 50          // i/p power in W
I_br = 2.5      // line current in A
V_br = 25       // line voltgae
R_1 = 2.4       // resistance between two resistance in
     ohm

// caclulations
V_br = V_br/sqrt(3)          // phase voltage
P_br = P/3                   // power per phase in W
R_eq = P_br/I_br^2           // equivalent resistance
    in ohm
R_2 = R_eq-(R_1/2)           // per phase rotor
    resistance in ohm
Z_eq = V_br/I_br             // equivalent impedance
    in ohm
X_eq = sqrt(Z_eq^2-R_2^2)   // equivalent reactance
    in ohm
X_1 = 0.5*X_eq

// display the result
disp("Example 8.5 solution");
printf(" \n Equivalent resistance \n R_eq = %.1f ohm
     \n", R_eq );
printf(" \n Equivalent reactance \n X_eq = %.1f ohm
    \n", X_eq );
printf(" \n Equivalent reactance \n X_1 = %.1f ohm \
    n", X_1 );
```

**Scilab code Exa 8.6** determine equivalent parameters

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

```

```
// Chapter 8 : STARTING, CONTROL AND TESTING OF AN INDUCTION MOTOR
// Example : 8.5

clc;clear; // clears the console and command history

// Given data
V = 210        // supply voltage in V
f = 50         // supply frequency in Hz
P = 4          // number of poles
P_0 = 400      // i/p power in W
I_0 = 1.2      // line current in A
V_0 = 210      // line voltage
P_fw = 150     // total friction and windage losses in W
R = 2.2        // stator resistance between two terminals in ohm

// caclulations
R_1 = R/2                              // per phase stator resistance in ohm
P_scu = 3*I_0^2*R_1                    // copper loss in W
P_core = P_0-P_fw-P_scu                // stator core loss in W
R_0 = (V_0/sqrt(3))^2/(P_core/3)       // no-load resistance in ohm
// alternate approach
phi_0 = acosd(P_core/(sqrt(3)*V_0*I_0)) // power factor angle
X_0 = (V_0/sqrt(3))/(I_0*sind(phi_0))   // magnetizing reactance per phase in ohm

// display the result
disp("Example 8.6 solution");
printf(" \n No-load resistance \n R_0 = %.1f ohm \n", R_0 );
printf(" \n Magnetizing reactance per phase \n X_0 =
```

```
%.0f ohm \n", X_0 );
```

**Scilab code Exa 8.7** determine combined slip combined synchronous speed out put

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 8 : STARTING, CONTROL AND TESTING OF AN INDUCTION MOTOR
// Example : 8.7

clc;clear; // clears the console and command history

// Given data
P_1 = 6       // number of pole
P_2 = 4       // number of pole
f =50         // supply frequency in Hz
P = 60        // power in kW

// caclulations
s = P_2/(P_1+P_2)          // combined slip
N_cs = 120*f/(P_1+P_2)     // combined synchronous speed in rpm
P_0 = P*(P_2/(P_1+P_2))    // o/p of 4-pole motor in kW

// display the result
disp("Example 8.7 solution");
printf(" \n Combined slip \n s = %.1f  \n", s );
printf(" \n Combined synchronous speed \n N_cs = %.0f rpm \n", N_cs );
printf(" \n Out-power of 4-pole motor \n P_0 = %.f
```

```
kW \n", P_0 );
```

# Chapter 9

# SYNCHRONOUS GENERETOR

**Scilab code Exa 9.1** determine number of poles

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 9 : SYNCHRONOUS GENERATOR
// Example : 9.1

clc;clear; // clears the console and command history

// Given data
N = 300      // speed of water turbine in rpm
f = 50       // frequency in Hz

// caclulations
P = 120*f/N // number of poles

// display the result
disp("Example 9.1 solution");
```

```
printf(" \n Number of poles of the generator \n P =
    %.0f poles \n", P );
```

**Scilab code Exa 9.2** determine pitch factor

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 9 : SYNCHRONOUS GENERATOR
// Example : 9.2

clc;clear; // clears the console and command history

// Given data
P = 8      // number of poles
m = 3      // number of phase
S = 144    // number of slots

// caclulations
T_p = S/P            // pole pitch interms of slots
slots_1 = 180/T_p    // pole pitch per slots
y = 2*slots_1        // short pitch angle in degree
k_p = cosd(y/2)      // pitch factor

// display the result
disp("Example 9.2 solution");
printf(" \n Pitch factor is \n k_p = %.2f \n", k_p )
    ;
```

**Scilab code Exa 9.3** determine pitch factor

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 9 : SYNCHRONOUS GENERATOR
// Example : 9.3

clc;clear; // clears the console and command history

// Given data
P = 4  // number of poles
m = 3  // number of phase
S = 40 // number of slots

// caclulations
T_p = S/P                // pole pitch interms of slots
T_c = 9-1                // coil span 1 to 9 i.e. coil
    pitch in terms of slots
slots_1 = 180/T_p        // pole pitch per slots
y = T_p-T_c              // short pitch angle
y_angle = y*slots_1      // in terms of angle
k_p = cosd(y_angle/2) // pitch factor

// display the result
disp("Example 9.3 solution");
printf(" \n Pitch factor is \n k_p = %.2f \n", k_p )
    ;
```

---

**Scilab code Exa 9.4** find distribution factor

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION
```

```

// Chapter 9 : SYNCHRONOUS GENERATOR
// Example : 9.4

clc;clear; // clears the console and command history

// Given data
P = 4   // number of poles
S = 48  // number of slots

// caclulations
T_p = S/P                                   // pole pitch interms of slots
slots_1 = 180/T_p                           // pole pitch per slots
n = S/(P*3)                                 // number of slots or coils per pole per phase
k_d = sind((n*slots_1)/2)/(n*sind(slots_1/2)) // distribution factor

// display the result
disp("Example 9.4 solution");
printf(" \n Distribution factor is \n k_d = %.2f \n", k_d );
```

**Scilab code Exa 9.5** determine line voltage

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 9 : SYNCHRONOUS GENERATOR
// Example : 9.5

```

```
clc;clear; // clears the console and command history

// Given data
P = 12        // number of poles
S = 180       // number of slots
phi_m = 0.05 // flux per pole in Wb
N = 600       // speed of machine in rpm

// caclulations
T_p = S/P                                          //
    pole pitch interms of slots
slots_1 = 180/T_p                                  //
    pole pitch per slots
n = S/(P*3)                                    // number
    of slots or coils per pole per phase
k_d = sind((n*slots_1)/2)/(n*sind(slots_1/2)) //
    distribution factor
k_p = 1                                            //
    pitch factor
Z = (180/3)*slots_1                                //
    number of conductor per phase
T = Z/2                                            //
    number of turns per phase
f = P*N/120                                        //
    frequency in Hz
E = 4.44*k_p*k_d*f*phi_m*T                         //
    induced voltage in V
E_L = sqrt(3)*E                                    //
    line voltage in V

// display the result
disp("Example 9.5 solution");
printf(" \n Line voltage is \n E_L = %.0f V \n", E_L
    );

// NOTE : correction in answer
```

**Scilab code Exa 9.6** determine voltage per pahse

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 9 : SYNCHRONOUS GENERATOR
// Example : 9.6

clc;clear; // clears the console and command history

// Given data
P = 4         // number of poles
m = 3         // number of phase
f = 50        // frequency in Hz
phi_m = 0.05 // flux per pole in Wb

// caclulations
T_p = 6*3 // pole pitch interms of slots 6 slots per pole per phase hence for 3  phase
slots_1 = 180/T_p                                    // pole pitch per slots
T_c = (5/6)*T_p                                      // coil per pitch
y = T_p-T_c                                          // short pitch angle
y_angle = y*10                                       // short pitch in terms of angle
k_p = cosd(y_angle/2)                                // pitch factor
n = 6                                                // number of slots
k_d = sind((n*slots_1)/2)/(n*sind(slots_1/2)) //
```

```
    distribution factor
T = (1/2)*n*P*2*5                              // 2=
    no. of layers , 5=condctor per layer
E = 4.44*k_p*k_d*f*phi_m*T                     //
    induced voltage in V

// display the result
disp("Example 9.6 solution");
printf(" \n Voltage per phase is \n E = %.0f V \n",
    E );
```

**Scilab code Exa 9.7** determine rms value of the induced voltage per phase

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 9 : SYNCHRONOUS GENERATOR
// Example : 9.7

clc;clear; // clears the console and command history

// Given data
P = 10          // number of poles
m = 3           // number of phase
f = 50          // frequency in Hz
phi_m1 = 0.05   // flux per pole in Wb
phi_m3 = 0.006  // flux per pole in Wb
T_c = 150       // coil

// caclulations
T_p = 3*3 // pole pitch interms of slots 3 slots per
    pole per phase hence for 3  phase
slots_1 = 180/T_p                                  //
```

```
pole pitch per slots
y = 180-T_c                                      // short pitch angle
n = 3                                            // number of slots
k_p1 = cosd(y/2)                                 // pitch factor
k_d1 = sind((n*slots_1)/2)/(n*sind(slots_1/2))  // distribution factor
E_1 = 4.44*k_p1*k_d1*f*phi_m1*T_c                // induced voltage in V
k_p3 = cosd(y/2)                                 // pitch factor
k_d3 = sind((n*slots_1)/2)/(n*sind(slots_1/2))  // distribution factor
E_3 = 4.44*k_p3*k_d3*f*phi_m3*T_c                // induced voltage in V
E = sqrt(E_1^2+E_3^2)                            // induced voltage per phase in V

// display the result
disp("Example 9.7 solution");
printf(" \n Induced voltage per phase is \n E = %.0f V \n", E );
```

---

**Scilab code Exa 9.8** determine noload induced voltage per phase and VR

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 9 : SYNCHRONOUS GENERATOR
// Example : 9.8

```

```
clc;clear; // clears the console and command history

// Given data
kVA = 50     // kVA ratings
V_t = 220    // terminal voltage in V
R_a = 0.011  // effective resistance in ohm
X_s = 0.09   // synchronous reactance in ohm

// caclulations
phi = acosd(0.85)    // since power factor in 0.85
I_a = kVA*10^3/V_t    // armature current in A
E_f = sqrt((V_t*cosd(phi)+I_a*R_a)^2+(V_t*sind(phi)+I_a*X_s)^2) // induced voltage per phase in V
VR = ((E_f-V_t)/V_t)*100    // voltage regulation

// display the result
disp("Example 9.8 solution");
printf(" \n No-load induced voltage per phase \n E_f = %.1f V \n", E_f );
printf(" \n Voltage regulation is \n VR = %.1f percent \n", VR );
```

---

**Scilab code Exa 9.9** determine synchronous impedance reactance full load VR

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 9 : SYNCHRONOUS GENERATOR
// Example : 9.9
```

```

clc;clear; // clears the console and command history

// Given data
kVA = 200       // kVA ratings
V_t = 33*10^3   // terminal voltage in V
R_a = 0.54      // armature resistance in ohm
V_L = 415       // voltage between lines for SC test in V
I_sh = 25       // short circuit current in A

// caclulations
phi = acosd(0.9)                // since power factor in 0.9
V_P = V_L/sqrt(3)               // phase voltage during short circuit test in V
Z_s = V_P/I_sh                  // synchronous impedance in ohm
X_s = sqrt(Z_s^2 - R_a^2)        // synchronous reactance in ohm
I_a = kVA/(sqrt(3)*V_t*10^-3) // full loa current in A
V_ta = V_t/sqrt(3)              // voltage per phase alternator
E_f = sqrt((V_ta*cosd(phi)+I_a*R_a)^2+(V_ta*sind(phi)+I_a*X_s)^2) // no-load voltage per phase in V
VR = ((E_f-V_ta)/V_ta)*100      // voltage regulation

// display the result
disp("Example 9.9 solution");
printf(" \n Synchronous impedance \n Z_s = %.1f ohm \n", Z_s );
printf(" \n Synchronous reactance is \n X_s = %.2f ohm \n", X_s );
printf(" \n Voltage regulation is \n VR = %.2f percent \n", VR );
printf(" \n NOTE : error in calculation , R_a is taken instead of X_s in E_f calculation \n");
```

**Scilab code Exa 9.10** determine power delivered and three phase max power

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 9 : SYNCHRONOUS GENERATOR
// Example : 9.10

clc;clear; // clears the console and command history

// Given data
MVA = 30    // MVA ratings
V = 20      // supply voltage in kV
N = 1800    // speed in rpm
V_t = 15    // voltage per phase in kV
E_f = 10    // per phase terminal voltage in kV
teta = 40   // power angle in degree
X_s = 6     // 3 phase synchronous reactance in ohm

// caclulations
P = 3*V_t*E_f*sind(teta)/X_s    // power delivered to
    the load in MW
P_max = 3*V_t*E_f/X_s           // three phase
    maximum power in MW

// display the result
disp("Example 9.10 solution");
printf(" \n Three phase power delivered to the load
    \n P = %.2f MW \n", P );
printf(" \n Three phase maximum power \n P_max = %.0
    f MW \n", P_max );
```

**Scilab code Exa 9.11** determine torque angle induced voltage per phase VR

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 9 : SYNCHRONOUS GENERATOR
// Example : 9.11

clc;clear; // clears the console and command history

// Given data
kVA = 25          // kVA ratings
V = 440           // suppy voltage in V
f = 50            // supply frequency in Hz
pf = 0.8          // lagging power factor
R_a = 0.3         // resistance of machine per phase in ohm
X_d = 5           // reactance of machine per phase in ohm
X_q = 3           // reactance of machine per phase in ohm

// caclulations
phi = acosd(pf)
V_t = V/sqrt(3)                                  // terminal voltage per phase in V
I_a = 25*10^3/(sqrt(3)*V)                        // armature current
delta = atand(I_a*X_q*pf/(V_t+I_a*X_q*sind(phi))) // torque angle
I_d = I_a*sind(delta+phi)                   // direct
```

```
axis component of the current in A
E_f = V_t*cosd(delta)+I_d*X_d    // induced voltage per phase in V
VR = ((E_f-V_t)/V_t)*100    // voltage regulation in V

// display the result
disp("Example 9.11 solution");
printf(" \n Torque angle \n delta = %.2f degree \n", delta );
printf(" \n Induced voltage per phase \n E_f = %.2f V \n", E_f );
printf(" \n Voltage regulation \n VR = %.2f percent \n", VR );
```

**Scilab code Exa 9.12** determine load current terminal voltage and power per pahse

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 9 : SYNCHRONOUS GENERATOR
// Example : 9.12

clc;clear; // clears the console and command history

// Given data
E_1 = 220   // induced voltage per phase by an alternator1
E_2 = 220*exp(%i*5*%pi/180)   // induced voltage per phase by an alternator2
Z_1 = %i*3  // impedance of an alternator1
Z_2 = %i*4  // impedance of an alternator2
```

```
R = 5          // resistance in ohm
Z = 5

// caclulations
I = (E_1*Z_2+E_2*Z_1)/(Z_1*Z_2+Z*(Z_1+Z_2))
    // load current in A
V_t = I*R
    // terminal voltage in V
I_a1 = ((E_1-E_2)*Z+E_1*Z_2)/(Z_1*Z_2+Z*(Z_1+Z_2))
    // armature current in A
D = atand(imag(I),real(I))// from V_t
A = atand(imag(V_t),real(V_t))
                                                          //
    from I_a1
P_1 = abs(V_t*I_a1)*cosd(D-A)          // power per
    phase delivered by the 1st alternator in W

// display the result
disp("Example 9.12 solution");
printf(" \n load current \n I = %.1f<%.2f degree \n"
    , abs(I),atand(imag(I),real(I)) );
printf(" \n terminal voltage \n V_t = %.f<%.2f V \n"
    , abs(V_t),atand(imag(V_t),real(V_t)) );
printf(" \n Power per phase delivered by the 1st
    alternator \n P_1 = %.2f W \n", P_1 );
printf(" \n NOTE : ERROR : Calculation mistakes in
    textbook \n")
```

# Chapter 10

# SYNCHRONOUS MOTOR

**Scilab code Exa 10.1** determine excitation voltage per phase

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 10 : SYNCHRONOUS MOTOR
// Example : 10.1

clc;clear; // clears the console and command history

// Given data
V = 2.5*10^3 // supply voltage in V
R_r = 0.12   // per phase resistance in ohm
X_r = 3.2    // syncronous reactance in ohm
I_a = 185    // line current in A
pf = 0.8     // leading power factor

// caclulations
phi = acosd(pf)
V_t = V/sqrt(3)     // terminal voltage per phase in V
```

```
Z_s = R_r+%i*X_r    // impedance per phase ohm
beta = atand(X_r/R_r)
E_r = I_a*Z_s       // resultant voltage due to
   impedance in V
E_f = sqrt(V_t^2+abs(E_r)^2-2*V_t*abs(E_r)*cosd(beta
   +phi)) // excitation voltage per phase in V


// display the result
disp("Example 10.1 solution");
printf(" \n Excitation voltage per phase \n E_f = %
   .2f V \n", E_f );
```

**Scilab code Exa 10.2** calculate the excitation voltage per phase and torque angle

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 10 : SYNCHRONOUS MOTOR
// Example : 10.2

clc;clear; // clears the console and command history

// Given data
kVA = 1200    // kVA ratings
V = 14*10^3   // supply voltage in V
R_r = 4.8     // per phase resistance in ohm
X_r = 35      // syncronous reactance in ohm
pf = 0.95     // leading power factor

// caclulations
phi = acosd(pf)
```

```
Z_s = R_r+%i*X_r              // impedance per phase
   ohm
I_a = kVA*10^3/(sqrt(3)*V)  // armature current in A
E_r = I_a*Z_s                 // resultant voltage due
   to impedance in V
V_t = V/sqrt(3)               // terminal voltage per
   phase in V
b = atand(X_r/R_r)            // beta value
E_f = sqrt(V_t^2+abs(E_r)^2-2*V_t*abs(E_r)*cosd(b-
   phi)) // excitation voltage per phase in V
teta = sind(64)
D = (E_r*teta/E_f)            // torque angle
delta = asind(abs(D))

// display the result
disp("Example 10.2 solution");
printf(" \n Excitation voltage per phase \n E_f = %
   .2f V \n", E_f );
printf(" \n Torque angle at 0.95 power factor
   lagging \n delta = %.2f degree \n", delta );
```

**Scilab code Exa 10.3** calculate maximum power

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 10 : SYNCHRONOUS MOTOR
// Example : 10.3

clc;clear; // clears the console and command history

// Given data
V = 440            // supply voltage in V
```

```
R_a = 1.5          // per phase resistance in ohm
X_a = 8            // syncronous reactance in ohm
P = 4              // number of poles
f = 50             // supply frequency in Hz
pf = 0.9           // leading power factor
I_a = 50           // armature current in A

// caclulations
V_t = V/sqrt(3)        // terminal voltage per phase
    in V
phi = acosd(pf)        // angle in degree
Z_s = R_a+%i*X_a       // impedance per phase ohm
E_r = I_a*abs(Z_s)          // resultant voltage due
    to impedance in V
bet = atand(X_a/R_a)
E_f = sqrt(V_t^2+E_r^2-2*V_t*E_r*cosd(bet+phi)) //
    excitation voltage per phase in V
P_dm = (((E_f*V_t)/Z_s)-((E_f^2*R_a)/Z_s^2))
             // maximum power per phase in W

// display the result
disp("Example 10.3 solution");
printf(" \n Maximum power per phase \n P_dm = %.2f W
     \n", P_dm );
printf(" \n In textbook solution they took E_f =
    513.5V instead of 533.33V");
```

**Scilab code Exa 10.4** determine max power and max torque

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 10 : SYNCHRONOUS MOTOR
```

```
// Example : 10.4

clc;clear; // clears the console and command history

// Given data
P = 4 // number of poles
f = 50 // supply frequency in Hz
V_t = 1500 // terminal voltage per phase in V
E_f = 1000 // excitation voltage per phase in V
Z_s = 12 // synchronnous impedance in ohm
R_a = 1.5 //armature resistance in ohm

// caclulations
P_dm = (((E_f*V_t)/Z_s)-((E_f^2*R_a)/Z_s^2)) //
    maximum power per phase in W
T_dm = 9.55*P_dm/1500 // maximum torque in N-m

// display the result
disp("Example 10.4 solution");
printf(" \n Maximum power developed \n P_dm = %.0f W
    \n", P_dm );
printf(" \n Maximum toruqe \n T_dm = %.1f N-m \n",
    T_dm );
```

# Chapter 11

# SINGLE PHASE MOTORS

**Scilab code Exa 11.1** determine slip due to forward and backward field and effective rotor resistance

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 11 : SINGLE-PHASE MOTORS
// Example : 11.1

clc;clear; // clears the console and command history

// Given data
V = 220            // supply voltage in V
P = 4              // number of poles
f = 50             // frequency in Hz
N_l = 1450         // speed in rpm
P_tloss = 2000     // total power loss in W
R_2 = 10           // rotor resistance at standstill
    in ohm

// caclulations
```

```
N_s = 120*f/P          // synchronous speed in rpm
s_f = (N_s-N_l)/N_s  // slip due to forward field
s_b = 2-s_f          // slip due to backward field
R_f = R_2/s_f        // effective rotor resistance
    due to forward slip in ohm
R_b = R_2/(2-s_f)    // effective rotor resistance
    due to backward slip in ohm

// display the result
disp("Example 11.1 solution");
printf(" \n Slip due to forward field \n s_f = %.2f
    \n", s_f );
printf(" \n Slip due to backward field \n s_b = %.2f
     \n", s_b );
printf(" \n Effective rotor resistance due to
    forward slip \n R_f = %.2f ohm \n", R_f );
printf(" \n Effective rotor resistance due to
    backward slip \n R_b = %.2f ohm \n", R_b );

printf(" \n NOTE : for caclulating R_f, s_f is taken
     as 0.033333 so we got R_f=300");
```

**Scilab code Exa 11.2** calculate in put current power developed power and torque developed

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 11 : SINGLE-PHASE MOTORS
// Example : 11.2

clc;clear; // clears the console and command history

```

```
// Given data
V_t = 220   // supply voltage in V
R_1 = 6     // equivalent parameters of single phase induction motor in ohm
R_2 = 6     // equivalent parameters of single phase induction motor in ohm
X_1 = 10    // equivalent parameters of single phase induction motor in ohm
X_2 = 10   // equivalent parameters of single phase induction motor in ohm
N = 1500    // speed in rpm
s = 0.03    // slip
X_m = 150   // equivalent parameters of single phase induction motor in ohm

// caclulations
Z_f = 0.5*%i*X_m*((R_2/s)+%i*X_2)/((R_2/s)+%i*(X_2+X_m)) // impedance due to forward field in ohm
R_f = real(Z_f) // from Z_f
Z_b = 0.5*%i*X_m*((R_2/(2-s))+%i*X_2)/((R_2/(2-s))+%i*(X_2+X_m)) // impedance due to backward field in ohm
R_b = real(Z_b) // from Z_b
Z_t = R_1+%i*X_1+Z_f+Z_b // total impedence in ohm
I_1 = V_t/Z_t // input current in A
P_d = (abs(I_1))^2*(R_f-R_b)*(1-s) // power developed in W
T_d = 9.55*P_d/N // torque in N-m

// display the result
disp("Example 11.2 solution");
printf(" \n input current \n I_1 = %.2f<%.2f A \n", abs(I_1),atand(imag(I_1),real(I_1)));
printf(" \n power developed \n P_d = %.2f \n", P_d );
printf(" \n torque \n T_d = %.2f \n", T_d );
printf("\n NOTE : ERROR : There is calculation mistake in Z_b in textbook. So there is change in
```

```
answers from textbook")
```

**Scilab code Exa 11.3** starting current main winding current and line current

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 11 : SINGLE-PHASE MOTORS
// Example : 11.3

clc;clear; // clears the console and command history

// Given data
V_t = 220    // supply voltage in V
f = 50       // frequency in Hz
Z_m = 3+%i*5 // main winding impedance of motor in
    ohm
Z_s = 5+%i*3 // starting impedance of motor in ohm

// caclulations
alpha_s = atand(imag(Z_s),real(Z_s))
                                        // angle in degree
    from Z_s
I_s = V_t/Z_s                                // starting
    current in A
alpha_m = atand(imag(Z_m),real(Z_m))
                                          // angle in degree
     from Z_m
I_m = V_t/(Z_m)                               // main
    winding current in A
alpha = alpha_m-alpha_s                         // angle
     of line current
```

```
I = sqrt((abs(I_s))^2+(abs(I_m))^2+2*abs(I_s)*abs(I_m)*cosd(alpha)) // line current in A

// display the result
disp("Example 11.3 solution");
printf(" \n Input current \n I_s = %.2f<%.2f A \n", abs(I_s),atand(imag(I_s),real(I_s)) );
printf(" \n Main winding current \n I_m = %.2f<%.f A \n", abs(I_m),atand(imag(I_m),real(I_m)));
printf(" \n Line current \n I = %.2f A \n", I );
```

**Scilab code Exa 11.4** find value of capacitance

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 11 : SINGLE-PHASE MOTORS
// Example : 11.4

clc;clear; // clears the console and command history

// Given data
V_t = 220       // supply voltage in V
f = 50          // frequency in Hz
Z_m = 4+%i*3.5 // main winding impedance of motor in ohm
Z_s = 5+%i*3   // starting impedance of motor in ohm
R_s = 5         // from Z_s
X_s = 3         // from Z_s

// caclulations
alpha_m = 41.2 // angle in degree from Z_m
// Let X_c be connected in series with the starting
```

```
    winding. Then the total impedance of starting
    winding is Z_s = Z_m-%i*X_c
// The torque will be the maximum when the angle
    between the starting winding and main winding
    currents in 90 electrical degree.The value of the
     angle of the starting winding current is
alpha_s = alpha_m-90
X_c = X_s-R_s*tand(alpha_s)
C = 1/(2*%pi*f*X_c) // starting capacitance to get
    maximum torque in F

// display the result
disp("Example 11.4 solution");
printf(" \n Starting capacitance for getting maximum
     torque \n C = %.2e F \n", C );
```

**Scilab code Exa 11.5** calculate the equivalent circuit parameters

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 11 : SINGLE-PHASE MOTORS
// Example : 11.5

clc;clear; // clears the console and command history

// Given data
f = 50       // supply frequency in Hz
V_nl = 100   // no-load voltage in v
I_nl = 2.5   // no-load current in A
P_nl = 60    // no-load power in W

V_br = 60     // Block rotor test voltage in v
```

```
I_br = 3      // Block rotor test current in A
P_br = 130    // Block rotor test  power in W
R_1 = 2       // main windning resistance in ohm

// caclulations
Z_br = V_br/I_br                        // impedance due to blocked rotor test
R_br = P_br/I_br^2                      // resistance due to blocked rotor test in ohm
X_br = sqrt(Z_br^2-R_br^2)              // reactance under blocked condition in ohm
X_1 = X_br/2                            // reactance in ohm X_1=X_2
R_2 = R_br-R_1                          // resistance in ohm
Z_nl = V_nl/I_nl                        // impedance due to no-load in ohm
R_nl = P_nl/I_nl^2                      // resistance due to no-load in ohm
X_nl = sqrt(Z_nl^2-R_nl^2)              // reactance due to no-load in ohm
X_m =2*(X_nl-X_1-0.5*X_1)               // magnetizing reactance in ohm
P_rot = P_nl-I_nl^2*(R_1+((R_2)/4)) // rotational loss in W

// display the result
disp("Example 11.5 solution");
printf(" \n Magnetizing reactance \n X_m = %.1f ohm \n", X_m );
printf(" \n Rotational loss \n P_rot = %.0f W \n", P_rot );
```

**Scilab code Exa 11.6** determine tooth pitch and step angle

```
// FUNDAMENTALS OF ELECTICAL MACHINES
// M.A.SALAM
// NAROSA PUBLISHING HOUSE
// SECOND EDITION

// Chapter 11 : SINGLE-PHASE MOTORS
// Example : 11.6

clc;clear; // clears the console and command history

// Given data
r_t = 36 // rotor teeth of stepper motor
N = 4    // stator phases

// caclulations
T_p = 360/r_t       // tooth pitch
teta = 360/(N*r_t) // step angle

// display the result
disp("Example 11.6 solution");
printf(" \n Tooth pitch \n T_p = %.0i degree \n",
   T_p );
printf(" \n Strp angle \n teta = %.1f degree \n",
   teta );
```