

Scilab Textbook Companion for  
Fundamentals of Electrical Machines  
by M. A. Salam<sup>1</sup>

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

July 31, 2019

<sup>1</sup>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	4
1 REVIEW OF ELECTRIC CIRCUITS	5
2 BASICS OF MAGNETIC CIRCUITS	19
3 TRANSFORMER AND PER UNIT SYSTEM	33
4 DIRECT CURRENT GENERATORS	54
5 DIRECT CURRENT MOTORS	65
6 CONTROL AND STARTING OF A DC MOTOR	75
7 THREE PHASE INDUCTION MOTOR	83
8 STARTING CONTROL AND TESTING OF AN INDUCTION MOTOR	92
9 SYNCHRONOUS GENERATOR	100
10 SYNCHRONOUS MOTOR	114
11 SINGLE PHASE MOTORS	119

# List of Scilab Codes

Exa 1.1	calculate current . . . . .	5
Exa 1.2	determine magnitude and sign of power . . .	6
Exa 1.3	use Thevenin theorem to find current . . .	6
Exa 1.4	find the rms value . . . . .	7
Exa 1.5	calculate the magnitude of the line voltage .	8
Exa 1.6	find line currents and power factor . . . . .	9
Exa 1.7	find magnitude of phase currents and line current . . . . .	10
Exa 1.8	find line currents and load phase voltages . .	12
Exa 1.9	find phase currents and neutral current . . .	13
Exa 1.10	find current in each load and line currents .	14
Exa 1.11	calculate line currents pf power supplied . .	16
Exa 1.12	find second wattmeter reading . . . . .	17
Exa 2.1	calculate total flux . . . . .	19
Exa 2.2	calculate flux density MMF magnetic field intensity . . . . .	20
Exa 9.2	determine pitch factor . . . . .	21
Exa 2.3	find permeability MFI MFD . . . . .	22
Exa 2.4	find MMF and reluctance . . . . .	23
Exa 2.5	find MFD . . . . .	23
Exa 2.6	find MFS flux density and flux . . . . .	24
Exa 2.7	calculate flux flux density and field intensity	25
Exa 2.8	calculate current . . . . .	26
Exa 2.9	find reluctance and relative permeability .	28
Exa 2.10	calculate current . . . . .	29
Exa 2.11	determine force . . . . .	30
Exa 2.12	determine the inductance of the coil . . . . .	31
Exa 2.13	determine hysteresis loss . . . . .	32

Exa 3.1	find no of primary turns primary full load current and secondary full load current . . . . .	33
Exa 3.2	determine turns ratio and mutual flux . . . . .	34
Exa 3.3	find iron loss and magnetizing component of currents . . . . .	35
Exa 3.4	find turns ratio load impedance primary current . . . . .	36
Exa 3.5	determine primary current . . . . .	37
Exa 3.6	determine impedance referred to the primary and secondary . . . . .	38
Exa 3.7	find voltage regulation . . . . .	39
Exa 3.8	calculate efficiency . . . . .	40
Exa 3.9	full load efficiency output kVA . . . . .	41
Exa 3.10	calculate all day efficiency . . . . .	43
Exa 3.11	determine no-load circuit resistance and reactance . . . . .	44
Exa 3.12	find equivalent R L Z referred to primary and secondary and VR . . . . .	45
Exa 3.13	determine kVA ratings . . . . .	47
Exa 3.14	determine circulating current . . . . .	48
Exa 3.15	determine line voltage and current at secondary side . . . . .	49
Exa 3.16	determine the per unit values of R L Z . . . . .	50
Exa 3.17	find grid bus voltage . . . . .	52
Exa 4.2	determine induced voltage . . . . .	54
Exa 4.3	determine induced voltage . . . . .	55
Exa 4.4	calculate generated emf . . . . .	55
Exa 4.5	find electromagnetic torque . . . . .	56
Exa 4.6	calculate the generated voltage . . . . .	57
Exa 4.7	determine generated emf . . . . .	58
Exa 4.8	determine the generated voltage and PD . . . . .	59
Exa 4.9	determine ATc and ATd . . . . .	61
Exa 4.10	determine flux per pole . . . . .	62
Exa 4.11	determine induced voltage and efficiency . . . . .	63
Exa 5.1	determine magnitude of force . . . . .	65
Exa 5.2	determine value of back emf and mechanical power developed . . . . .	66
Exa 5.3	determine change in back emf . . . . .	67

Exa 5.4	find torque developed by armature . . . . .	67
Exa 5.5	find speed and shaft torque . . . . .	68
Exa 5.6	determine the motor speed under load condition . . . . .	69
Exa 5.7	find the speed of the motor . . . . .	70
Exa 5.8	calculate the iron and friction loss and efficiency . . . . .	71
Exa 5.9	determine $P_{cu}$ $T_a$ $T_{sh}$ and efficiency . . . . .	73
Exa 6.1	find value of resistance . . . . .	75
Exa 6.2	find the speed at full load and half load torque . . . . .	76
Exa 6.3	find new speed . . . . .	77
Exa 6.4	determine the speed . . . . .	78
Exa 6.5	calculate value of resistance . . . . .	79
Exa 6.6	find the value of resistance . . . . .	80
Exa 7.1	determine synchronous speed slip and rotor frequency . . . . .	83
Exa 7.2	determine synchronous speed rotor frequency and rotor voltage . . . . .	84
Exa 7.3	find the value of starting torque and current . . . . .	85
Exa 7.4	find speed and ratio . . . . .	86
Exa 7.5	determine slip synchronous speed shaft speed and mechanical power . . . . .	87
Exa 7.6	determine max mechanical power $T_{max}$ and slip . . . . .	88
Exa 7.7	determine slip rotor $c_u$ loss shaft motor efficiency . . . . .	89
Exa 7.8	determine length . . . . .	90
Exa 8.2	determine percentage taping on the autotransformer . . . . .	92
Exa 8.3	determine the full load current . . . . .	93
Exa 8.4	determine torques . . . . .	94
Exa 8.5	determine the equivalent parameters . . . . .	95
Exa 8.6	determine equivalent parameters . . . . .	96
Exa 8.7	determine combined slip combined synchronous speed out put . . . . .	98
Exa 9.1	determine number of poles . . . . .	100
Exa 9.2	determine pitch factor . . . . .	101
Exa 9.3	determine pitch factor . . . . .	101

Exa 9.4	find distribution factor . . . . .	102
Exa 9.5	determine line voltage . . . . .	103
Exa 9.6	determine voltage per pahse . . . . .	105
Exa 9.7	determine rms value of the induced voltage per phase . . . . .	106
Exa 9.8	determine noload induced voltage per phase and VR . . . . .	107
Exa 9.9	determine synchronous impedance reactance full load VR . . . . .	108
Exa 9.10	determine power delivered and three phase max power . . . . .	110
Exa 9.11	determine torque angle induced voltage per phase VR . . . . .	111
Exa 9.12	determine load current terminal voltage and power per pahse . . . . .	112
Exa 10.1	determine excitation voltage per phase . . .	114
Exa 10.2	calculate the excitation voltage per phase and torque angle . . . . .	115
Exa 10.3	calculate maximum power . . . . .	116
Exa 10.4	determine max power and max torque . . .	117
Exa 11.1	determine slip due to forward and backward field and effective rotor resistance . . . . .	119
Exa 11.2	calculate in put current power developed power and torque developed . . . . .	120
Exa 11.3	starting current main winding current and line current . . . . .	122
Exa 11.4	find value of capacitance . . . . .	123
Exa 11.5	calculate the equivalent circuit parameters .	124
Exa 11.6	determine tooth pitch and step angle . . . .	125



# Chapter 1

## REVIEW OF ELECTRIC CIRCUITS

Scilab code Exa 1.1 calculate current

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 1 : REVIEW OF ELECTRIC CIRCUITS
7 // Example : 1.1
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 Q = 4 // charge in C
13 t = 0.54 // time in sec
14
15
16 // calculations
17 I = Q/t // current in A
18
19 // display the result
```

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

---

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

```
1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
7 // Example : 1.2
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V = -24 // voltage in V
13 I = 3 // current in A
14
15 // caclulations
16 P = V*I // power for the element A in network in W
17
18 // display the result
19 disp("Example 1.2 solution");
20 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

```
1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
```

```

5
6 // Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
7 // Example : 1.3
8
9 clc;clear; // clears the console and command history
10 // Given data
11 R1 = 5 // resistance in ohm
12 R2 = 4 // resistance in ohm
13 R3 = 9 // resistance in ohm
14 R4 = 6 // resistance in ohm
15 V1 = 10 // voltage in V
16 V2 = 6 // voltage in V
17
18
19 // caclulations
20 // Remove R3 and find R_th by short circuiting V1
    and V2 than R1 and R4 will be in parallel
21 R = (R1*R4)/(R1+R4) // equivalent resistance in
    ohm
22 // R is connected in series with R2
23 R_th = R2+R // Thevenin's resistance in
    ohm
24 I = 4/11 // current in the figure
    applying KVL in A
25 V_th = (6*0.36)+6 // voltage in V
26 I_9 = V_th/(R_th+R3) // current through R3 in A
27
28
29 // display the result
30 disp("Example 1.3 solution");
31 printf(" \n Current through 9 \n I_9 = %.2f A ",
    I_9);

```

---

Scilab code Exa 1.4 find the rms value

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
7 // Example : 1.4
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V_t1 = 30 // magnitudes of voltages in V 0 < t1 <
           2
13 V_t2 = -10 // magnitudes of voltages in V 2 < t2 <
           4
14 T = 4 // time period from figure
15
16 // caclulations
17
18 V1 = 30;
19 V2 = -10;
20 X= sqrt((1/4)*(integrate('V1^2','x',0,2) + integrate
        ('V2^2','x',2,4)));
21
22 //display the result
23 disp("Example 1.4 solution");
24 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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION

```

```

5
6 // Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
7 // Example : 1.5
8
9 clc;clear; // clears the console and command history
10
11
12 // Given data
13 V_P = 200 // magnitude of
    phase voltage
14 V_an = 200 // magnitude of
    phase voltage in V
15 V_bn = 200*exp(%i*(-120)*(%pi/180)) // magnitude of
    phase voltage in V
16 V_cn = 200*exp(%i*(120)*(%pi/180)) // magnitude of
    phase voltage in V
17
18 // caclulations
19 V_L = sqrt(3)*V_P // magnitude of line voltage in V
20
21 // display the result
22 disp("Example 1.5 solution");
23 printf(" \n Magnitude of line voltage \n V_L = %.2f
    V ", V_L);

```

---

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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
7 // Example : 1.6
8

```

```

9  clc;clear; // clears the console and command history
10
11
12 // Given data
13 R = 10 // resistance in
14 L = 15 // inductance reactance in
15 V_L = 420 // voltage in V
16 f = 50 // frwuency in Hz
17
18 // caclulations
19 V_an = (V_L/sqrt(3))*exp(%i*(-30)*(%pi/180)) //
    phase voltage
20 V_bn = (V_L/sqrt(3))*exp(%i*(-150)*(%pi/180)) //
    phase voltage
21 V_cn = (V_L/sqrt(3))*exp(%i*(90)*(%pi/180)) // phase
    voltage
22 Z_p = R+%i*L // phase impedance in
23 I_L1 = V_an/Z_p // line current
24 I_L2 = V_bn/Z_p // line current
25 I_L3 = V_cn/Z_p // line current
26 pf = R/abs(Z_p) // lagging power factor
27
28 // display the result
29 disp("Example 1.6 solution");
30 printf(" \n Line currents are \n I_L1 = %.2f<% 2f A
    \n", abs(I_L1),atand( imag(I_L1),real(I_L1)) );
31 printf(" I_L2 = %.2f<% 2f A \n", abs(I_L2),atand(
    imag(I_L2),real(I_L2)) );
32 printf(" I_L3 = %.2f<% 2f A \n", abs(I_L3),atand(
    imag(I_L3),real(I_L3)) );
33 printf(" \n Power factor \n pf = %.1f lag \n", pf );

```

---

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

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

```

2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 1 : REVIEW OF ELECRTCIC CIRCUITS
7 // Example : 1.7
8
9 clc;clear; // clears the console and command history
10
11
12 // Given data
13 Z_p = 10+%i*15 // per phase impedance in
14 V_ab = 420 // magnitude of
    phase voltage in V
15 V_bc = 420*exp(%i*(-120)*(%pi/180)) // magnitude of
    phase voltage in V
16 V_ca = 420*exp(%i*(120)*(%pi/180)) // magnitude of
    phase voltage in V
17
18
19
20 // caclulations
21 I_ab = V_ab/Z_p // phase current in A
22 I_bc = V_bc/Z_p // phase current in A
23 I_ca = V_ca/Z_p // phase current in A
24 I_P = V_ab/abs(Z_p)
25 I_L = sqrt(3)*I_P // line current in A
26
27 // display the result
28 disp("Example 1.7 solution");
29 printf(" \n Phase currents are \n I_ab = %.2f<% 2f A
    \n", abs(I_ab),atand(imag(I_ab),real(I_ab)) );
30 printf(" I_bc = %.2f<% 2f A \n", abs(I_bc),atand(
    imag(I_bc),real(I_bc)) );
31 printf(" I_ca = %.2f<% 2f A \n", abs(I_ca),atand(
    imag(I_ca),real(I_ca)) );
32 printf(" \n Line current \n I_L = %.2f A \n", I_L);

```

---

Scilab code Exa 1.8 find line currents and load phase voltages

```
1
2 // FUNDAMENTALS OF ELECTICAL MACHINES
3 // M.A.SALAM
4 // NAROSA PUBLISHING HOUSE
5 // SECOND EDITION
6
7 // Chapter 1 : REVIEW OF ELECTRIC CIRCUITS
8 // Example : 1.8
9
10 clc;clear; // clears the console and command history
11
12 // Given data
13 V_L = 280 // generator phase voltage in V rms
14 Z_1 = 2+%i*3 // line impedance
15 Z_L = 4+%i*5 // load impedance
16
17 // calculations
18 Z_t = Z_1+Z_L //
19 V_An = (V_L*exp(%i*(0)*(%pi/180))) // phase
    voltage
20 V_Bn = (V_L*exp(%i*(-120)*(%pi/180))) // phase
    voltage
21 V_Cn = (V_L*exp(%i*(120)*(%pi/180))) // phase
    voltage
22
23 I_Aa = V_An/Z_t // line current for phase A in A
24 I_Bb = V_Bn/Z_t // line current for phase B in A
25 I_Cc = V_Cn/Z_t // line current for phase C in A
26
27 V_an = I_Aa*Z_L // voltage of load in V
28 V_bn = I_Bb*Z_L // voltage of load in V
29 V_cn = I_Cc*Z_L // voltage of load in V
```



```

30
31 // display the result
32 disp("Example 1.8 solution");
33 // There is error in the textbook regarding phasor
    angles of
34 printf(" \n Line currents are \n I_Aa = %.2f<%.2f A
    \n", abs(I_Aa), atand( imag(I_Aa), real(I_Aa)) );
35 printf(" I_Bb = %.2f<%.2f A \n", abs(I_Bb), atand(
    imag(I_Bb), real(I_Bb)) );
36 printf(" I_Cc = %.2f<%.2f A \n\n", abs(I_Cc), atand(
    imag(I_Cc), real(I_Cc)) );
37 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");
38 // 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
39 printf(" \n\n Phase voltages are ");
40 printf(" \n V_an = %.2f<%.2f V \n", abs(V_an), atand(
    imag(V_an), real(V_an)) );
41 printf(" V_bn = %.2f<%.2f V \n", abs(V_bn), atand(
    imag(V_bn), real(V_bn)) );
42 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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 1 : REVIEW OF ELECR TIC CIRCUITS
7 // Example : 1.9

```

```

8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V_P = 340 // phase voltage in V
13 Z = 6+%i*8 // per phase impedance
14
15 // caclulations
16
17 V_AN = (V_P/exp(%i*(0)*(%pi/180))) // phase
    voltage
18 V_BN = (V_P/exp(%i*(-120)*(%pi/180))) // phase
    voltage
19 V_CN = (V_P/exp(%i*(120)*(%pi/180))) // phase
    voltage
20
21 I_Aa = V_AN/Z // line current for phase A in A
22 I_Bb = V_BN/Z // line current for phase B in A
23 I_Cc = V_CN/Z // line current for phase C in A
24
25
26
27 // display the result
28 disp("Example 1.9 solution");
29 printf(" \n Line currents are \n I_Aa = %.2f<% 2f A
    \n", abs(I_Aa),atand(imag(I_Aa),real(I_Aa)) );
30 printf(" I_Bb = %.2f<% 2f A \n", abs(I_Bb),atand(
    imag(I_Bb),real(I_Bb)) );
31 printf(" I_Cc = %.2f<% 2f A \n", abs(I_Cc),atand(
    imag(I_Cc),real(I_Cc)) );
32 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

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 1 : REVIEW OF ELECTRIC CIRCUITS
7 // Example : 1.10
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V_AN = 200 // voltage in V
13 Z = 3+%i*4 // impedance in
14
15 // calculations
16
17 V_AB = sqrt(3)*(V_AN*exp(%i*(30)*(%pi/180))) //
    phase voltage in V
18 V_BC = sqrt(3)*(V_AN*exp(%i*(-90)*(%pi/180))) //
    phase voltage in V
19 V_CA = sqrt(3)*(V_AN*exp(%i*(150)*(%pi/180))) //
    phase voltage in V
20
21 I_ab = V_AB/Z // magnitude of load current in A
22 I_bc = V_BC/Z // magnitude of load current in A
23 I_ca = V_CA/Z // magnitude of load current in A
24
25 I_Aa = I_ab-I_ca // magnitude of line current in A
26 I_Bb = I_bc-I_ab // magnitude of line current in A
27 I_Cc = I_ca-I_bc // magnitude of line current in A
28
29 // display the result
30 disp("Example 1.10 solution");
31 printf(" \n Current in each load \n I_ab = %.2f<% 2f
    A \n", abs(I_ab),atand(imag(I_ab),real(I_ab)) );
32 printf(" I_bc = %.2f<% 2f A \n", abs(I_bc),atand(
    imag(I_bc),real(I_bc)) );
33 printf(" I_ca = %.2f<% 2f A \n\n", abs(I_ca),atand(

```

```

    imag(I_ca),real(I_ca)) );
34
35 printf(" \n Line currents \n I_Aa = %.2f<%.2f A \n",
    abs(I_Aa),atand(imag(I_Aa),real(I_Aa)) );
36 printf(" I_Bb = %.2f<%.2f A \n", abs(I_Bb),atand(
    imag(I_Bb),real(I_Bb)) );
37 printf(" I_Cc = %.2f<%.2f A \n\n", abs(I_Cc),atand(
    imag(I_Cc),real(I_Cc)) );
38
39 printf(" NOTE : The line currents as given in this
    example in text book are wrong since ,the question
    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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 1 : REVIEW OF ELECRTCIC CIRCUITS
7 // Example : 1.11
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 Z = 3+%i*4 // impedance in
13 V_AN = 150 // voltage in V
14 R = 3 // resistance in from Z
15
16 // caclulations
17
18 V_AN = (150*exp(%i*(0)*(%pi/180))) //source

```

```

    voltage in V
19 V_BN = (150*exp(%i*(-120)*(%pi/180))) // source
    voltage in V
20 V_CN = (150*exp(%i*(120)*(%pi/180))) // source
    voltage in V
21
22 I_Aa = V_AN/Z // line current in A
23 I_Bb = V_BN/Z // line current in A
24 I_Cc = V_CN/Z // line current in A
25
26 pf = R/abs(Z) // power factor
27 I = V_AN/abs(Z) // current in A
28 P = V_AN*I*pf // power supplied in W
29 P_t = 3*P // total power supplied in W
30
31 // display the result
32 disp("Example 1.11 solution");
33 printf(" \n Line currents are \n I_Aa = %.2f<% 2f A
    \n", abs(I_Aa),atand(imag(I_Aa),real(I_Aa)) );
34 printf(" I_Bb = %.2f<% 2f A \n", abs(I_Bb),atand(
    imag(I_Bb),real(I_Bb)) );
35 printf(" I_Cc = %.2f<% 2f A \n\n", abs(I_Cc),atand(
    imag(I_Cc),real(I_Cc)) );
36 printf(" Power factor \n pf = %.1f \n", pf);
37 printf(" \n Power supplied to each phase is \n P = %
    .2f W \n", P);
38 printf(" \n Total power supplied \n P_t = %.2f W \n"
    , P_t);

```

---

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

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION

```

```

5
6 // Chapter 1 : REVIEW OF ELECRTIC CIRCUITS
7 // Example : 1.12
8
9 clc;clear; // clears the console and command history
10
11
12 // Given data
13 P = 120 // total power in kW
14 cos_teta = 0.6 // power factor
15
16 // caclulations
17 teta = acosd(0.6) // power factor angle
18 P_1 = tand(teta)*P/sqrt(3)
19 P_2 = (P_1+P)/2 // second wattmeter
    reading in kW
20
21 // display the result
22 disp("Example 1.12 solution");
23 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

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 2 : BESICS OF MAGNETIC CIRCUITS
7 // Example : 2.1
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 l = 4 // lenght in m
13 w = 2 // width in m
14 B = 0.12 // magnetis flux density in tesla
15
16 // caclulations
17 A = l*w // area in m^2
18 flux = B*A // magnetic flux in Wb
19
```

```

20
21 // display the result
22 disp("Example 2.1 solution");
23 printf("\n Magnetic flux \n flux = %.2f Wb \n",
        flux);

```

---

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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 2 : BESICS OF MAGNETIC CIRCUITS
7 // Example : 2.2
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 d_in = 3 // inside diameter of iron toroid in cm
13 d_out = 6 // outside diameter of iron toroid in cm
14 N =200 // number of turns
15 I = 3 // current in A
16 flux = 0.015 // flux in Wb
17
18 // caclulations
19 d = d_in+((d_out-d_in)/2) // distance in cm
20 l = %pi*d // mean length in cm
21 A = %pi*d^2/4 // area in cm^2
22 B =flux/(A*10^-4) // flux density in mWb/m^2
23 MMF = N*I // magnetomotive force in At
24 H = (N*I)/(l*10^-2) // magnetic field intensity in
    At/m
25
26 // display the result

```



```

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

```

---

### Scilab code Exa 9.2 determine pitch factor

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 9 : SYNCHRONOUS GENERATOR
7 // Example : 9.2
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 P = 8 // number of poles
13 m = 3 // number of phase
14 S = 144 // number of slots
15
16 // caclulations
17 T_p = S/P // pole pitch interms of slots
18 slots_1 = 180/T_p // pole pitch per slots
19 y = 2*slots_1 // short pitch angle in degree
20 k_p = cosd(y/2) // pitch factor
21
22 // display the result
23 disp("Example 9.2 solution");
24 printf(" \n Pitch factor is \n k_p = %.2f \n", k_p )

```

;

---

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

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 2 : BESICS OF MAGNETIC CIRCUITS
7 // Example : 2.3
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 myu_r = 625 // relative permiability of rectangular
    core
13 N = 25 // number of turns
14 I = 2 // current in A
15 a = 5.5 // lenght of rectangular box in cm
16 b = 1.5 // width of rectangular box in cm
17
18 // caclulations
19 l = 2*(a+b) // mean length of core in cm
20 H = N*I/(1*10^-2) // magnetic field intensity
    in At/m
21 myu = 4*%pi*10^-7*myu_r // permeabilty
22 B = myu*H // magnetic flux density in
    Wb/m^2;
23
24 // display the result
25 disp("Example 2.3 solution");
26 printf(" \n Magnetic field intensity \n H= %.0f At/m
    \n", H);
27 printf(" \n Permeabilty \n myu= %.2e \n", myu);
```

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

---

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

```
1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 2 : BESICS OF MAGNETIC CIRCUITS
7 // Example : 2.4
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 N = 6 // number of turns
13 I =3 // current in A
14 flux = 0.056 // flux in Wb
15
16 // caclulations
17 MMF = N*I // magnetomotive force in At
18 R_m = MMF/flux // reluctance in At/Wb
19
20 // display the result
21 disp("Example 2.4 solution");
22 printf("\n Magnetomotive force \n MMF= %.0 f At \n",
MMF);
23 printf( "\n Reluctance \n R_m= %.1 f At/Wb \n", R_m);
```

---

#### Scilab code Exa 2.5 find MFD

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

```

2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 2 : BESICS OF MAGNETIC CIRCUITS
7 // Example : 2.5
8 clc;clear; // clears the console and command history
9
10 // Given data
11 I = 15 // current through conductor
    in A
12 N = 10 // number of turns
13 myu_0 = 4*pi*10^-7 // permiability
14 myu_r = 1 // relative permiability of air
15 r = 0.015
16
17 // caclulations
18 B = myu_0*myu_r*N*I/(2*pi*r) // magnetic flux in T
19
20 // display the result
21 disp("Example 2.5 solution");
22 printf(" \n Magnetic flux \n B= %.0e T \n", B);

```

---

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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 2 : BESICS OF MAGNETIC CIRCUITS
7 // Example : 2.6
8
9 clc;clear; // clears the console and command history
10

```

```

11 // Given data
12 N = 200 // number of turns
13 d_in = 7 // inner diameter of wooden toroidal
    coil in cm
14 d_out = 10 // outer diameter of wooden toroidal
    coil in cm
15 A = 0.005 // cross sectional area m^2
16 I = 5 // current through coil in A
17 R = d_out-d_in
18 myu_0 = 4*%pi*10^-7
19
20 // caclulations
21 l = 2*%pi*R*10^-2 // mean circumference length in m
22 H = N*I/l // magnetic field intensity in At
    /m
23 B = myu_0*H // flux density in Wb/m^2
24 flux = B*A // flux in Wb
25
26 // display the result
27 disp("Example 2.6 solution");
28 printf(" \n Magnetic field intensity \n H= %.0f At/
    m \n", H);
29 printf(" \n Flux density \n B= %.2e Wb/m^2 \n", B);
30 printf(" \n Flux \n flux= %.1e Wb \n", flux);

```

---

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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 2 : BESICS OF MAGNETIC CIRCUITS
7 // Example : 2.7
8 clc;clear; // clears the console and command history

```

```

9
10 // Given data
11 l = 0.1 // length in m
12 w = 0.01 // width in m
13 h =0.1 // height in m
14 N = 450 // number of turns
15 I = 0.2 // current in A
16 myu_r = 850 // relative permiability magnetic
    material
17
18 // caclulations
19 MMF = N*I // magnetomotive
    force in At
20 l_c = (h-w)*4 // mean length of
    the path in m
21 A = w*w // cross sectional
    area in m^2
22 R_m = l_c/(4*pi*10^-7*myu_r*A) // relectance in At/
    Wb
23 flux = MMF/R_m // flux in Wb
24 B = flux/A // magnetic flux
    density in Wb/m^2
25 H = B/(4*pi*10^-7*myu_r) // field intensity
    in At/m
26
27 // display the result
28 disp("Example 2.7 solution");
29 printf(" \n Flux \n flux= %.2e Wb \n", flux);
30 printf(" \n Magnetic flux density \n B= %.4f Wb/m^2
    \n", B);
31 printf(" \n Field intensity \n H= %.2f At/m \n", H);

```

---

Scilab code Exa 2.8 calculate current

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

```

2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 2 : BESICS OF MAGNETIC CIRCUITS
7 // Example : 2.8
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 N = 450 // number of turns wound on left side
    of limb
13 A = 4 // cross sectional area in m^2
14 I = 5 // current through coil in A
15 flux = 3 // flux in right limb Wb
16 myu_r = 500 // relative permiability magnetic
    material
17 l_1 = 0.12
18 l_2 = 0.24
19 phi2 = 3
20
21 // caclulations
22 // we have relation
23 // phi1*rm1 = phi2*rm2
24 // phi1*l_1/uA = pi2*l_2/uA
25 phi1 = phi2*l_2/l_1 // flux on left side
    in Wb
26 flux = phi1+phi2 // total flux in Wb
27 B_1 = flux/A // flux density in the
    left limb
28 H_1 = B_1/(4*pi*10^-7*myu_r) // magnetic flux in At
    /m
29 MMF_1 = H_1*l_2 // magnetomotive force
    in At
30 B_2 = phi2/A // flux density in the
    right limb
31 H_2 = B_2/(4*pi*10^-7*myu_r) // magnetic flux in At
    /m

```

```

32 MMF_2 = H_2*l_2           // magnetomotive force
    in At
33 MMF_t = MMF_1+MMF_2     // total magnetomotive
    force in At
34 I = MMF_t/N             // current in A
35
36 // display the result
37 disp("Example 2.8 solution");
38 printf(" \n Current\n I= %.2f A \n", I);

```

---

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

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 2 : BESICS OF MAGNETIC CIRCUITS
7 // Example : 2.9
8 clc;clear; // clears the console and command history
9
10 // Given data
11 l = 0.45           // mean lenght in m
12 A = 25*10^-4      // cross sectional area in m^2
13 l_ag = 0.8*10^-3 // air gap in m
14 N = 500           // number of turns
15 I = 1.25          // current in A
16 fulx = 2.25*10^-3 // flux in Wb
17 phi = 1.5*10^-3  // flux in Wb
18
19 // caclulations
20 B = phi/A         // magnetic flux
    density in Wb/m^2
21 MMF = N*I        // magnetomotive
    force in At

```



```

22 H = B/(4*%pi*10^-7)           // magnetomotizing
    force in At/m
23 MMF_ag = H*l_ag               // magnetomotive
    force in At
24 MMF_i = MMF-MMF_ag           // magnetomotive
    force for iron ring in At
25 H_i = MMF_i/l                // magnetic field
    intensity for iron part in At/m
26 myu_r = B/((4*%pi*10^-7)*H_i) // relative
    permiability for iron
27
28 // display the result
29 disp("Example 2.9 solution");
30 printf("\n Relative permiability for iron \n myu_r
    = %.2f \n", myu_r);
31 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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 2 : BESICS OF MAGNETIC CIRCUITS
7 // Example : 2.10
8 clc;clear; // clears the console and command history
9
10 // Given data
11 A = 2*10^-4           // cross sectional area in m^2
12 N = 200              // number of turns
13 flux = 1.5*10^-3     // flux in Wb
14 myu_r = 4000         // relative permiability of core

```

```

15 l_1 = 0.01           // length in m
16 a = 9               // length in cm
17 w = 3               // width in cm
18
19 // caclulations
20 myu = myu_r*4*%pi*10^-7 // permiability
21 l_2 = (4*(a-w-w+(1.5+1.5))-1) // mean length in cm
22 R_mg = l_1/(4*%pi*10^-7*A) // reluctance of iron
    for air gap At/Wb
23 R_mi = l_2*10^-2/(myu*A) // reluctance of iron
    for air gapAt/Wb
24 R_mt = R_mg+R_mi // total relectance in
    At/Wb
25 I = R_mt*flux/N // current in A
26
27 // display the result
28 disp("Example 2.10 solution");
29 printf("\n Current flowing through the coil \n I =
    %.0f A \n", I);
30
31 // NOTE : In question they given flux=2.5mWb but in
    solution they took flux=1.5mWb

```

---

### Scilab code Exa 2.11 determine force

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 2 : BESICS OF MAGNETIC CIRCUITS
7 // Example : 2.11
8 clc;clear; // clears the console and command history
9
10 // Given data

```

```

11 I = 150 // current through conductor in A
12 l = 2 // conductor length in m
13 B = 0.35 // magnetic flux density in T
14
15 // caclulations
16 F = B*l*I // force in N
17
18 // display the result
19 disp("Example 2.11 solution");
20 printf("\n Force \n F = %.0f N \n", F);

```

---

Scilab code Exa 2.12 determine the inductance of the coil

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 2 : BESICS OF MAGNETIC CIRCUITS
7 // Example : 2.12
8 clc;clear; // clears the console and command history
9
10 // Given data
11 l = 25*10^-2 // length of air-core coil in m
12 A = 4*10^-4 // cross sectional area in m^2
13 N = 65 // number of turns
14 myu_0 = 4*pi*10^-7
15 myu_r = 1
16
17 // caclulations
18 myu = myu_0*myu_r
19 L = N^2*myu*A/l // inductance in H
20
21 // display the result
22 disp("Example 2.12 solution");

```

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

---

**Scilab code Exa 2.13** determine hysteresis loss

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 2 : BESICS OF MAGNETIC CIRCUITS
7 // Example : 2.13
8 clc;clear; // clears the console and command history
9
10 // Given data
11 k_h = 110 // hysteresis co-efficient in J/m^3
12 V_cvool = 0.005 // volume of the core in m^3
13 B_m = 1.12 // flux density in T
14 f = 60 // frequency in Hz
15 n = 1.6
16
17 // caclulations
18 P_h = k_h*V_cvool*B_m^n*f // hysteresis loss in W
19
20 // display the result
21 disp("Example 2.13 solution");
22 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 second

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.1
8 clc;clear; // clears the console and command history
9
10 // Given data
11 V_1 = 2200 // primary voltage of transformer in V
12 V_2 = 220 // secondary voltage of transformer in V
13 N_2 = 56 // number of turns in the secondary coil
    of transformer
14 kVA = 25 // kVA rating of transformer
15
16 // caclulations
17 a = V_1/V_2 // turn ratio
18 N_1 = a*N_2 // number of primary turns
```

```

19 I_1 = kVA*10^3/V_1 // primary full load current in
    A
20 I_2 = kVA*10^3/V_2 // secondary full load current
    in A
21
22 // display the result
23 disp("Example 3.1 solution");
24 printf(" \n Number of primary turns \n N_1 = %.0f \n
    ", N_1);
25 printf(" \n Primary full load current \n I_1 = %.2f
    A \n", I_1);
26 printf(" \n Secondary full load current \n I_2 = %.1
    f A \n", I_2);

```

---

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

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.2
8 clc;clear; // clears the console and command history
9
10 // Given data
11 V_1 = 220 // voltage in V
12 N_1 = 150 // number of turns in the primary coil of
    transformer
13 N_2 = 300 // number of turns in the secondary coil
    of transformer
14 f = 50 // frequency in Hz
15
16 // caclulations
17 a = N_1/N_2 // turn ratio

```

```

18 phi_m = V_1/(4.44*f*N_1) // mutual flux in Wb
19
20 // display the result
21 disp("Example 3.2 solution");
22 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

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.3
8 clc;clear; // clears the console and command history
9
10 // Given data
11 V_1 = 2200 // primary voltage of transformer in V
12 V_2 = 220 // secondary voltage of transformer in V
13 I_0 = 0.5 // no-load current in A
14 P_0 = 350 // absorbed power by transformer
15
16 // caclulations
17 phi = acos(.32)
18 sin_phi = sin(phi)
19 cos_phi = P_0/(V_1*I_0) // no-load power factor
20 I_w = I_0*cos_phi // iron loss component of
    current A
21 I_m = I_0*sin_phi // magnetizing component of
    current A
22
23
24 // display the result

```

```

25 disp("Example 3.3 solution");
26 printf(" \n The iron loss component of current A \n
        I_w = %.2f A \n", I_w);
27 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

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.4
8 clc;clear; // clears the console and command history
9
10 // Given data
11 N_1 = 450 // number of turns in the primary coil of
        transformer
12 N_2 = 45 // number of turns in the secondary coil
        of transformer
13 Z_L = 3 // load impedance in
14 V_1 = 15 // primary coil voltage of transformer in
        V
15
16 // caclulations
17 a = N_1/N_2 // turn ratio
18 Z_1 = a^2*Z_L // load impedance referred to primary
        ohm
19 I_1 = V_1/Z_1 // primary current in A
20
21
22 // display the result
23 disp("Example 3.4 solution");

```



```

24 printf(" \n Turn ratio \n a = %.0f \n", a);
25 printf(" \n Load impedance referred to primary \n
      Z_1 = %.0f \n", Z_1);
26 printf(" \n Primary current \n I_1 = %.2f A \n", I_1
      );

```

---

**Scilab code Exa 3.5** determine primary current

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.5
8 clc;clear; // clears the console and command history
9
10 // Given data
11 V_1 = 400 // primary voltage of transformer in
      V
12 V_2 = 100 // secondary voltage of transformer
      in V
13 I_0 = 0.4 // no-load current in A
14 I_2 = 100 // load draws current in A
15 cos_phi0 = 0.3 // power factor lagging from the
      supply
16 cos_phi2 = 0.6 // power factor lagging from the
      secondary
17
18 // caclulations
19 phi0 = acosd(0.3)
20 phi2 = acosd(0.6)
21 phi1 = phi0-phi2
22 a = V_1/V_2 // turn ratio
23 I_2! = I_2/a // secondary current equivalent to the

```

```

    primary
24 I_1 = sqrt((I_2!2)+(I_02)+(2*I_2!*I_0*cosd(19.4)))
    // primary current in A
25
26 // display the result
27 disp("Example 3.5 solution");
28 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

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.6
8 clc;clear; // clears the console and command history
9
10 // Given data
11 V_1 = 2000 // primary voltage of transformer in V
12 V_2 = 400 // secondary voltage of transformer in V
13 kVA = 200 // kVA rating of transformer
14 R_1 = 3 // primary resistance in
15 X_1 = 12 // primary reactance in
16 R_2 = 0.3 // secondary resistance in
17 X_2 = 0.1 // secondary reactance in
18
19 // caclulations
20 a = V_1/V_2 // turn ratio
21 R_01 = R_1+(a2*R_2) // total resistance
    referred to primary side
22 X_01 = X_1+(a2*X_2) // total reactance
    referred to primary side

```

```

23 Z_01 = sqrt((R_01^2)+(X_01^2)) // equivalent
    impedance referred to primary side in
24 R_02 = R_2+(R_1/a^2)           // total resistance
    referred to secondary side
25 X_02 = X_2+(X_1/a^2)           // total reactance
    referred to secondary side
26 Z_02 = sqrt((R_02^2)+(X_02^2)) // equivalent
    impedance referred to secondary side in
27
28 // display the result
29 disp("Example 3.6 solution");
30 printf(" \n Equivalent impedance referred to primary
    side \n Z_01 = %.1f \n", Z_01);
31 printf(" \n Equivalent impedance referred to
    secondary side \n Z_02 = %.2f \n", Z_02);

```

---

### Scilab code Exa 3.7 find voltage regulation

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.7
8 clc;clear; // clears the console and command history
9
10 // Given data
11 V_1 = 200 // primary voltage of transformer in
    V
12 V_2 = 400 // secondary voltage of transformer
    in V
13 R_1 = 0.3 // primary resistance in
14 X_1 = 0.6 // primary reactance in
15 R_2 = 0.8 // secondary resistance in

```

```

16 X_2 = 1.6          // secondary reactance in
17 I_2 = 10          // secondary supply current in A
18 cos_phi2 = 0.8    // power factor lagging
19
20 // caclulations
21 a = V_1/V_2        // turn ratio
22 phi2 = acosd(0.8)
23 sin_phi2 = sind(phi2)
24 R_02 = R_2+(R_1/a^2) // total resistance referred to
    secondary side
25 X_02 = X_2+(X_1/a^2) // total reactance referred to
    secondary side
26 E_2 = (V_2*cos_phi2+I_2*R_02)+(%i*(V_2*sin_phi2+I_2*
    X_02)) // no-load voltage
27 V_r = (abs(E_2)-V_2)/V_2*100
                                     // voltage
    regulation
28
29 // display the result
30 disp("Example 3.7 solution");
31 printf("\n Voltage regulation \n V_r = %.0f percent
    \n", V_r);

```

---

### Scilab code Exa 3.8 calculate efficiency

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.8
8 clc;clear; // clears the console and command history
9
10 // Given data

```

```

11 P_i = 1 // iron loss of transformer in kW
12 P_cu = 2 // copper loss of transformer in kW
13 kVA = 200 // kVA ratings of transformer
14 pf = 0.95 // power factor
15
16 // caclulations
17 P_cu1 = (3/4)^2*P_cu // copper loss at 1/2th of
    full load in kW
18 P_cu2 = (1/2)^2*P_cu // copper loss at 1/2th of
    full load in kW
19 P_01 = (3/4)*kVA*P_i // o/p power at 3/4 full load
    and unity power factor in kW
20 P_in1 = P_01+P_i+P_cu1 // i/p power at 3/4 full load
    and unity power factor in kW
21 n_1 = (P_01/P_in1)*100 // efficiency at 3/4 full
    load and unity power factor
22 P_02 = (1/2)*kVA*pf // o/p power factor at 1/2 full
    load and 0.95 power factor in kW
23 P_in2 = P_02+P_i+P_cu2 // i/p power at 1/2 full load
    and 0.95 power factor in kW
24 n_2 = (P_02/P_in2)*100 // efficiency at 1/2 full
    load and 0.95 power factor
25
26 // display the result
27 disp("Example 3.8 solution");
28 printf(" \n Efficiency at 3/4 full load and unity
    power factor \n n_1 = %.2f percent \n", n_1);
29 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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE

```

```

4 // SECOND EDITION
5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.9
8
9
10 clc;clear; // clears the console and command history
11
12 // Given data
13 P_i = 350 // iron loss of transformer in W
14 P_cu = 650 // copper loss of transformer in W
15 kVA = 30 // kVA ratings of transformer
16 pf = 0.6 // power factor
17
18 // caclulations
19 P_tloss = (P_i+P_cu)*10^-3 // total full load loss
    in kW
20 P_out = kVA*pf // o/p power at full
    load in kW
21 P_in = P_out+P_tloss // i/p power at full
    load
22 n_1 = (P_out/P_in)*100 // efficiency at full
    load
23 kVA_out = kVA*sqrt(P_i/P_cu) // o/p kVA
    corresponding to maximum efficiency
24 P_01 = kVA_out*pf // o/p power in W
25 P_tloss1 = 2*P_i // maximum efficiency
    iron loss=copper loss in W
26 P_in1 = P_01+P_tloss1*10^-3 // i/p power in kW
27 n_2 = (P_01/P_in1)*100 // efficiency
28
29 // display the result
30 disp("Example 3.9 solution");
31 printf(" \n Efficiency at full load \n n_1 = %.2f
    percent \n", n_1);
32 printf(" \n Out put power \n P_01 = %.1f kW \n",
    P_01);
33 printf(" \n Efficiency \n n_2 = %.2f percent \n",

```

```
n_2);
```

---

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

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.10
8
9
10 clc;clear; // clears the console and command history
11
12 // Given data
13 kVA = 12 // kVA ratings of transformer
14 n = 0.97 // maximum efficiency at unity power
    factor
15 t_1 = 8 // time in hours
16 P_1 = 10 // load in kW
17 pf_1 = 0.8 // lagging power factor
18 t_2 = 10 // time in hours
19 P_2 = 15 // load in kW
20 pf_2 = 0.90 // leading power factor
21 t_3 = 6 // time in hours at no load
22 P_3 = 0 // load in kW
23
24 // calculations
25 P_01 = kVA*1 // o/p power at
    full load and unity factor in kW
26 P_in1 = (P_01/n) // i/p power at
    full load
27 P_tloss = P_in1-P_01 // total loss in
    kW
```

```

28 P_cu = P_tloss/2           // copper loss at
    12 kVA P_cu=P_i in kW
29 P_024 = P_1*t_1+P_2*t_2+P_3*t_3 // all day o.p
    power in kWh
30 P_i24 = 24*P_cu           // iron loss for
    24 hours in kWh
31 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
32 P_in24 = P_024+P_i24+P_cu24 // all day i/p
    power in kWh
33 n_allday = (P_024/P_in24)*100 // all day
    efficiency
34
35 // display the result
36 disp("Example 3.10 solution");
37 printf("\n All day efficiency \n n_allday = %.0f
    percent \n", n_allday);

```

---

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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.11
8
9
10 clc;clear; // clears the console and command history
11
12 // Given data
13 V_1 = 200 // voltage in V
14 f = 50 // frequency in Hz
15 I_0 = 0.6 // single phase current in A

```



```

16 P_0 = 80    // power in W
17
18 // caclulations
19 cos_phi0 = P_0/(V_1*I_0) // power factor
20 sin_phi0 = 0.74         // from above expression
21 I_w = I_0*cos_phi0     // working component of no
    load current in A
22 I_m = I_0*sin_phi0     // working component of no
    load current in A
23 R_0 = V_1/I_w          // no load circuit
    resistance in ohm
24 X_0 = V_1/I_m          // no load circuit
    reactance in ohm
25
26 // display the result
27 disp("Example 3.11 solution");
28 printf(" \n No-load circuit resistance \n R_0 = %.2 f
    ohm \n", R_0);
29 printf(" \n No-load circuit reactance \n X_0 = %.1 f
    ohm \n", X_0);

```

---

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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.12
8
9
10 clc;clear; // clears the console and command history
11
12 // Given data

```

```

13 kVA = 25      // kVA ratings of transformer
14 V1 = 2200    // primary side voltage in V
15 V2 = 220     // secondary side voltage in V
16 V_1 = 40     // voltage at high voltage side in V
17 I_1 = 5      // current at high voltage side in A
18 P = 150      // power at high voltage side in W
19
20 // caclulations
21 Z_01 = V_1/I_1 // reactance to primary sidec
    in ohm
22 R_01 = P/I_1^2 // resistance to primary side
    in ohm
23 phi = acosd(R_01/Z_01) // power factor angle
24 X_01 = Z_01*sind(phi) // impedance to primary side
    in ohm
25 a = V1/V2     // turn ratio
26 Z_02 = Z_01/a^2 // reactance to secondary
    side in ohm
27 R_02 = R_01/a^2 // resistance to secondary
    side in ohm
28 X_02 = X_01/a^2 // impedance to secondary
    side in ohm
29 I_2 = kVA*10^3/V2 // secondary side current in
    A
30 E_2 = V2+I_2*Z_02 // secondary induced voltage
    in V
31 VR = ((E_2-V2)/V2)*100 // voltage regulation
32
33 // display the result
34 disp("Example 3.12 solution");
35 printf("\n Resistance to primary side \n Z_01 = %.2
    f ohm \n", Z_01);
36 printf("\n Resistance to primary side \n R_01 = %.1
    f ohm \n", R_01);
37 printf("\n Impedance to primary side \n X_01 = %.2 f
    ohm \n", X_01);
38 printf("\n Reactance to secondary side \n Z_02 = %
    .2 f ohm \n", Z_02);

```

```

39 printf(" \n Resistance to secondary side \n R_02 = %
    .2f ohm \n", R_02);
40 printf(" \n Impedance to secondary side \n X_02 = %
    .3f ohm \n", X_02);
41 printf(" \n oltage regulation \n VR = %.0f percent \
    n", VR);

```

---

### Scilab code Exa 3.13 determine kVA ratings

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.13
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 kVA = 120 // kVA ratings of
    autotransformer
13 V1 = 2200 // lower part voltage of
    autotransformer in V
14 V2 = 220 // upper part voltage of
    autotransformer in V
15
16 // caclulations
17 I_pq = kVA*10^3/V2 // currents of respective
    windings
18 I_qr = kVA*10^3/V1 // currents of respective
    windings
19 I_1 = I_pq+I_qr // current in primary side in
    A
20 V_2 = V1+V2 // voltage across the

```

```

    secondary side in V
21 kVA_1 = I_1*V1/1000    // kVA ratings of
    autotrnsformer
22 kVA_2 = I_pq*V_2/1000 // kVA ratings of
    autotrnsformer
23
24
25 // display the result
26 disp("Example 3.13 solution");
27 printf(" \n kVA ratings of autotrnsformer \n kVA_1 =
    %.0f kVA \n", kVA_1);
28 printf(" \n kVA ratings of autotrnsformer \n kVA_2 =
    %.0f kVA \n", kVA_2);

```

---

#### Scilab code Exa 3.14 determine circulating current

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.14
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 E_1 = 500 // secondary induced voltages in V
13 E_2 = 450 // secondary induced voltages in V
14 kVA_1 = 100 // kVA ratings of transformer
15 kVA_2 = 200 // kVA ratings of transformer
16 Z_1 = 0.05 // impedance of transformer
17 Z_2 = 0.08 // impedance of transformer
18
19 // caclulations

```

```

20 Z1 =Z_1*E_1/(kVA_1*10^3/E_1) // actual impedance of
    1st transformer in ohm
21 Z2 = Z_2*E_2/(kVA_1*10^3/E_2) // actual impedance of
    2nd transformer in ohm
22 Z = %i*(Z1+Z2)
23 I_c = (E_1-E_2)/(Z) // value of the
    circulating current
24
25 // display the result
26 disp("Example 3.14 solution");
27 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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.15
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V_L1 = 11 // three-phase transformer supply in kV
13 I_P1 = 6 // three-phase transformer current in A
14 a = 11 // turns ratio
15
16 // caclulations
17 // delta-wye
18 V_dP2 = V_L1*10^3/a // phase voltage at
    secondary side in V

```

```

19 V_dL2 = sqrt(3)*V_dP2           // line voltage at
    secondary side in V
20 I_dP1 = a/sqrt(3)              // phase current in the
    primary in A
21 I_dL2 = a*I_dP1                // line current in
    secondary in A
22 //Wye delta connection
23 V_wP1 = V_L1*10^3/sqrt(3)     // phase voltage at
    primary in V
24 V_wP2 = V_wP1/a               // phase voltage at
    secondary in V, V_L2=V_P2
25 I_wP2 = a*I_P1                // phase current in
    secondary in A
26 I_wL2 = sqrt(3)*I_wP2         // line current in
    secondary in A
27
28 // display the result
29 disp("Example 3.15 solution");
30 printf(" \n For delta-ye connection-");
31 printf(" \n Phase voltage at secondary side \n V_dL2
    = %.f V \n", V_dL2);
32 printf(" \n Line voltage at secondary side \n I_dL2
    = %.2f A \n", I_dL2);
33 printf(" \n For wye-delta connection-")
34 printf(" \n Phase voltage at secondary side \n V_wL2
    = %.2f V \n", V_wP2);
35 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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION

```

```

5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.16
8
9
10 clc;clear; // clears the console and command history
11
12 // Given data
13 V_b = 220 // voltage in V
14 f = 50 // frequency in Hz
15 S_b = 600 // power ratings
16 R = 3 // resistance in ohm
17 X_L = 5 // inductance in ohm
18 Z = R+X_L // impedance
19
20 // calculations
21 I_b = S_b/V_b // base value of current in A
22 Z_b = V_b^2/S_b // base value of impedance in
    ohm
23 R_pu = R/Z_b // per unit value of
    resistance in ohm
24 X_Lpu = X_L/Z_b // per unit value of impedance
    in ohm
25 Z_pu = abs(Z)/Z_b // per unit of value of
    impedance in ohm
26 Z_pu = R_pu+%i*X_Lpu // per unit of value of
    impedance in ohm NOTE: alternative
    method
27
28 // display the result
29 disp("Example 3.16 solution");
30 printf(" \n Per unit value of resistance \n R_pu = %
    .3f ohm \n", R_pu);
31 printf(" \n Per unit value of impedance \n X_Lpu = %
    .3f ohm \n", X_Lpu);
32 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

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 3 : TRANSFORMER AND PER UNIT SYSTEM
7 // Example : 3.17
8
9
10 clc;clear; // clears the console and command history
11
12 // Given data
13 S_b1 = 100 // base apparent power
14 V_bT11 = 220 // voltage of 1st transformer in kV
15 V_bT12 = 132 // voltage of 1st transformer in kV
16 X_T1 = 0.02 // impedance of 1st transformer in pu
17 S_b2 = 50 // base apparent power
18 V_bT21 = 132 // voltage of 2nd transformer in kV
19 V_bT22 = 66 // voltage of 2nd transformer in kV
20 X_T2 = 0.05 // impedance of 2nd transformer in pu
21 X_L = 4 // line impedance in ohm
22 P = 50 // power absorded in MW
23 pf = 0.6 // lagging power factor from
    transmission line
24 Z_p = 0.32*%i //Reactance of transformer in ohm
25
26 // caclulations
27 S_b = S_b1 //Base power(MW
    )
28 V_b = V_bT11 //Base
    voltage(kV)
29 a = V_bT11/V_bT12 //
```



```

    turn ratio for 1st transformer
30 Vb_line = (V_bT11/a)           // base voltage
    of line in kV
31 Zb_line = Vb_line^2/S_b1       //
    base impedance of line in ohm
32 Xpu_line = X_L/Zb_line        //
    per unit reactance of line
33 Xpu_T1 = X_T1*(V_bT11/V_b)^2*(S_b/S_b1) // 1st grid
    transformer ,the per unit reactance
34 Vb_load = (V_bT12/(V_bT12/V_bT22)) //
    load side base voltage in kV
35 Xpu_load = X_T2*(V_bT22/Vb_load)^2*(S_b/S_b2)
    // second load transformer ,the per unit
    reactance
36 I_b = S_b*1000/(sqrt(3)*Vb_load)
    // base current
37 I_L = S_b2*1000/(sqrt(3)*V_bT22*pf) //
    actualcurrent in load in A
38 I_Lpu = I_L/I_b              //
    per unit value of the load
39 V_L = V_bT22/V_bT22         //per unit value of the
    voltage at the load terminal(bus4)
40 V_gb = I_Lpu*exp(%i*acos(pf))*Z_p + 1 // per unit
    value of bus voltage
41 V_gba = abs(V_gb)*V_bT11 // actual value of grid to
    bus voltage
42
43 // display the result
44 disp(" Example 3.17 solution");
45 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

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 4 : DIRECT CURRENT GENERATORS
7 // Example : 4.2
8
9
10 clc;clear; // clears the console and command history
11
12 // Given data
13 l = 0.65 // length of conductor in m
14 v = 35 // speed in m/s
15 B = 0.8 // magnetic flux density in T
16
17 // caclulations
18 e = B*l*v // induced voltage in V
19
```

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

---

#### Scilab code Exa 4.3 determine induced voltage

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 4 : DIRECT CURRENT GENERATORS
7 // Example : 4.3
8
9
10 clc;clear; // clears the console and command history
11
12 // Given data
13 l = 1.5 // length of conductor in m
14 v = 20 // speed in m/s
15 B = 0.9 // magnetic flux density in Wb/m^2
16 teta = 35 // angle of rotation in degree
17
18 // caclulations
19 e = B*l*v*sind(teta) // induced voltage in V
20
21 // display the result
22 disp("Example 4.3 solution");
23 printf(" \n Induced voltage \n e = %.1f V \n", e);
```

---

#### Scilab code Exa 4.4 calculate generated emf

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
```

```

2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 4 : DIRECT CURRENT GENERATORS
7 // Example : 4.4
8
9
10 clc;clear; // clears the console and command history
11
12 // Given data
13 P = 4 // number of poles
14 Z = 40*10 // number of conductors
15 phi = 0.02 // flux per pole in Wb
16 N = 1200 // speed in rpm
17
18 // caclulations
19 A = P/2
20 E_g = (P*phi*Z*N)/(60*A) // generated voltage in V
21
22 // display the result
23 disp("Example 4.4 solution");
24 printf("\\n Generated voltage \\n E_g = %.0f V \\n",
        E_g);

```

---

**Scilab code Exa 4.5** find electromagnetic torque

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 4 : DIRECT CURRENT GENERATORS
7 // Example : 4.5
8

```

```

9
10 clc;clear; // clears the console and command history
11
12 // Given data
13 P = 6 // number of poles
14 Z = 600 // number of conductors
15 phi = 0.05 // flux per pole in Wb
16 N = 1000 // speed in rpm
17 I_a = 120 // generator supply current in A
18
19 // calculations
20 A=6 // for lap-wound A=P
21 E_g = (P*phi*Z*N)/(60*A) // generated voltage in V
22 T_em = ((P*Z*phi)/(2*pi*A))*I_a // electromagnetic
    torque in N-m
23
24
25 // display the result
26 disp("Example 4.5 solution");
27 printf(" \n Generated voltage \n E_g = %.0f V \n",
    E_g);
28 printf(" \n Electromagnetic torque \n T_em = %.2f N-
    m \n", T_em);

```

---

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

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 4 : DIRECT CURRENT GENERATORS
7 // Example : 4.6
8
9

```

```

10 clc;clear; // clears the console and command history
11
12 // Given data
13 V_t = 220 // shunt generator voltage in V
14 I = 250 // shunt generator current in A
15 R_sh = 50 // shunt field resistance in ohm
16 R_a = 0.02 // armature resistance in ohm
17
18 // caclulations
19 I_sh = V_t/R_sh // shunt field current in A
20 I_a = I+I_sh // armature current in A
21 E_g = V_t+I_a*R_a // generated voltage in V
22
23
24 // display the result
25 disp("Example 4.6 solution");
26 printf(" \n Generated voltage \n E_g = %.2 f V \n",
    E_g);

```

---

**Scilab code Exa 4.7** determine generated emf

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 4 : DIRECT CURRENT GENERATORS
7 // Example : 4.7
8
9
10 clc;clear; // clears the console and command history
11
12 // Given data
13 E = 25 // power of compound generator in kW
14 V_t = 220 // terminal voltage in V

```

```

15 R_se = 0.05 // series resistance in ohm
16 R_sh = 55 // shunt field resistance in ohm
17 R_a = 0.07 // armature resistance in ohm
18 brush_drop = 1 // voltage drop per brush in V
19
20 // caclulations
21 I_L = E*10^3/V_t // load current in A
22 I_sh1 = V_t/R_sh // shunt field current in A
23 I_a1 = I_sh1+I_L // armature current in A
24 E_g1 = V_t+I_a1*(R_a+R_se)+2*brush_drop // generator
    voltage in V
25 V_ab = V_t+I_L*R_se // voltage across the shunt
    field in V for short shunt generator
26 I_sh2 = V_ab/R_sh // current in the shunt field in A
    for short shunt generator
27 I_a2 = I_sh2+I_L // armature current in A for short
    shunt generator
28 E_g2 = V_ab+I_a2*R_a+2*brush_drop // generator
    voltage in V for short shunt generator
29
30 // display the result
31 disp("Example 4.7 solution");
32 printf(" \n Generated emf when generatar is
    connected in long shunt \n E_g1 = %.f V \n", E_g1
    );
33 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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION

```

```

5
6 // Chapter 4 : DIRECT CURRENT GENERATORS
7 // Example : 4.8
8
9
10 clc;clear; // clears the console and command history
11
12 // Given data
13 V_t = 220 // shunt generator voltage in V
14 I_L = 146 // generator delivering current in A
15 R_sh = 50 // shunt field resistance in ohm
16 R_a = 0.012 // armature resistance in ohm
17 R_s = 0.02 // series field resistance in ohm
18 R_d = 0.03 // diverter field resistance in ohm
19
20 // caclulations
21 I_sh = V_t/R_sh // shunt field current
    in A
22 I_a = I_L+I_sh // armature current in A
23 R_com = R_s*R_d/(R_s+R_d) // combined resistance
    in ohm
24 E_g = V_t+(I_a*(R_a+R_com)) // generated voltage in
    V
25 P_lsd = I_a^2*R_com // power loss in series
    and diverter in W
26 P_la = I_a^2*R_a // power loss in the
    armature circuit resistance in W
27 P_lsh = V_t*I_sh // power loss in shunt
    field resistance in W
28 P_dl = I_L*V_t // power delivered in W
29
30 // display the result
31 disp("Example 4.8 solution");
32 printf(" \n Generated voltage \n E_g = %.1f V \n",
    E_g);
33 printf(" \n Power distribution \n P_dl = %.0f W \n",
    P_dl);

```

---



Scilab code Exa 4.9 determine ATc and ATd

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 4 : DIRECT CURRENT GENERATORS
7 // Example : 4.9
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 P = 4 // number of poles
13 Z = 500 // number of conductors
14 I_a = 30 // generator supply current in A
15 alpha = 6 // brushes displaced angle in degree
16
17 // calculations
18 A = P/2 // for wave connected A=P/2
19 I_c = I_a/A // current per conductor in A
20 AT_d = Z*I_c*alpha/360 // demagnetizing ampere turns
    per pole in At
21 AT_c = Z*I_c*((1/(2*P))-(alpha/360)) // cross
    magnetizing ampere turn per pole in At
22
23
24 // display the result
25 disp("Example 4.9 solution");
26 printf(" \n Demagnetizing ampere turns per pole \n
    AT_d = %.1f At \n", AT_d );
27 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

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 4 : DIRECT CURRENT GENERATORS
7 // Example : 4.10
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 Pw = 12 // power in kW
13 P = 4 // number of poles
14 Z = 500 // number of conductors
15 V_t = 250 // generator voltage in V
16 N = 1000 // speed in rpm
17 P_cu = 600 // full load copper loss in W
18 brush_drop = 2 // brush drop in V
19
20 // caclulations
21 A = 4 // for lab wound A=P
22 I_a = Pw*10^3/V_t // armature current in
    A
23 R_a = P_cu/I_a^2 // from copper loss
    equestion R_a in ohm
24 E_g = V_t+I_a*R_a+brush_drop // generated voltage in
    V
25 phi = E_g*60*A/(P*Z*N) // flux per pole in Wb
26
27
28 // display the result
29 disp("Example 4.10 solution");
```

```

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

```

---

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

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 4 : DIRECT CURRENT GENERATORS
7 // Example : 4.11
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 P = 4 // number of poles
13 I_L = 25 // generator delivering current in
    A
14 V_t = 230 // generator terminal voltage in V
15 R_a = 0.2 // armature resistance in ohm
16 R_sh = 55 // shunt field resistance in ohm
17 brush_drop = 1 // brush drop in V
18
19 // caclulations
20 I_sh = V_t/R_sh // shunt field current
    in A
21 I_a = I_L+I_sh // armature current in
    A
22 E_g = V_t+I_a*R_a+brush_drop // induced voltage in V
23 P_arm = E_g*I_a // power generated in
    armature in W
24 P_L = V_t*I_L // power absorbed by
    load in W
25 n = (P_L/P_arm)*100 // efficiency

```

```
26
27 // display the result
28 disp("Example 4.11 solution");
29 printf(" \n Induced voltage \n E_g = %.1f V \n", E_g
        );
30 printf(" \n Efficiency \n n = %.1f percent \n", n );
```

---

# Chapter 5

## DIRECT CURRENT MOTORS

Scilab code Exa 5.1 determine magnitude of force

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 5 : REVIEW OF ELECTRIC CIRCUITS
7 // Example : 5.1
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 l = 10 // conductor length in m
13 B = 0.56 // magnetic flux density in T
14 I = 2 // current through conductor in A
15
16 // calculations
17 F = B*I*l // magnitude of force in N
18
19 // display the result
```

```

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

```

---

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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 5 : DIRECT CURRENT MOTORS
7 // Example : 5.2
8
9 clc;clear; // clears the console and command history
10 // Given data
11 I = 20 // total current in A
12 V_t = 250; // supply voltage in volt
13 R_sh = 200; // shunt field resistance in
14 R_a = 0.3; // armature resistance in
15
16 // caclulations
17 I_sh = V_t/R_sh; // shunt field current in A
18 I_a = I-I_sh' // armature current
19 E_b = V_t - R_a*I_a; // the back emf in V
20 P_md = E_b*I_a; // mechanical power developed
    in W
21
22 // display the result
23 disp("Example 5.2 solution");
24 printf("\n The back emf is \n E_b = %.1f V \n\n",E_b
    );
25 printf("\n Mechanical power developed is \n P_md =
    %.1f W" ,P_md );
26 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

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 5 : DIRECT CURRENT MOTORS
7 // Example : 5.3
8
9 clc;clear; // clears the console and command history
10 // Given data
11 R_a = 0.7; // armature circuit resistance in
12 V_t = 5; // applied voltage in V
13 I_anl = 5; // no-load armature current in A
14 I_afl = 35; // full-load armature current in A
15
16
17 // caclulations
18 E_bnl = V_t - R_a*I_anl; // back emf under no-
    load in V
19 E_bfl = V_t - R_a*I_afl; // back emf under full-
    load in V
20 E_bc = E_bnl - E_bfl; // change in back emf
    from no-load to full load in V
21
22 // display the result
23 disp("Example 5.3 solution");
24 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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 5 : DIRECT CURRENT MOTORS
7 // Example : 5.4
8
9 clc;clear; // clears the console and command history
10 // Given data
11 I = 40;      // current in A
12 V_t = 230;  // supply in V
13 N = 1100;   // speed in rpm
14 R_a = 0.25; // armature resistance in
15 R_sh = 230; //shunt resistance in
16
17
18 // caclulations
19 I_sh = V_t/R_sh; // shunt field current in A
20 I_a = I - I_sh; // armature current in A
21 E_b = V_t - I_a*R_a; // back emf
22 T_a = 9.55*E_b*I_a/N; // amrature torque in N-m
23
24 // display the result
25 disp("Example 5.4 solution");
26 printf("\\n The armature torque is \\n T_a = %.2f N-m
      ",T_a );

```

---

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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5

```



```

6 // Chapter 5 : DIRECT CURRENT MOTORS
7 // Example : 5.5
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 P = 6 // number of poles
13 V_t = 230 // supply to shunt motor in V
14 Z = 450 // number of conductors
15 R_a = 0.8 // armature resistance in ohm
16 I = 30 // supply current in A
17 P_0 = 5560 // out put power in W
18 I_F = 3 // current through field winding
19 phi = 25*10^-3 // flux per pole in Wb
20
21 // caclulations
22 A = 6 // for lap wond A=P
23 I_a = I-I_F // armature current in A
24 E_b = V_t-I_a*R_a // back emf in V
25 N = 60*A*E_b/(P*Z*phi) // speed in rpm
26 T_sh = 9.55*P_0/N // shaft torque in N-m
27
28 // display the result
29 disp("Example 5.5 solution");
30 printf(" speed \n N = %.1f rpm \n", N );
31 printf(" shaft torque \n T_sh = %.1f N-m \n", T_sh )
    ;

```

---

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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5

```

```

6 // Chapter 5 : DIRECT CURRENT MOTORS
7 // Example : 5.6
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 I_L1 = 5 // dc shunt motor current
13 V_t = 230 // terminal voltage in V
14 N_1 = 1000 // speed in rpm
15 R_a = 0.2 // armature resistance in ohm
16 R_F = 230 // field resistance in ohm
17 I_L2 = 30 // dc shunt motor current
18
19 // caclulations
20 // at noload condition
21 I_sh = V_t/R_F // shunt field current in A
22 I_a1 = I_L1-I_sh // armature current in A
23 E_b1 = V_t-I_a1*R_a // back emf in V
24 // under load condition
25 I_a2 = I_L2-I_sh // armature current in A
26 E_b2 = V_t-I_a2*R_a // back emf in V
27 N_2 = (E_b2/E_b1)*N_1 // motor speed under load
    condition in rpm
28
29 // display the result
30 disp("Example 5.6 solution");
31 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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION

```

```

5
6 // Chapter 5 : DIRECT CURRENT MOTORS
7 // Example : 5.7
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 I_a1 = 65 // supply current to dc series motor
    in A
13 V_t = 230 // supply voltage in V
14 N_1 = 900 // speed in rpm
15 R_a = 0.2 // armature resistance in ohm
16 R_sh = 0.25 // field resistance in ohm
17 I_a2 = 15 // supply current to dc series motor
    in A
18 // phi_2 = 0.4*phi_1 value of flux
19
20
21 // caclulations
22 E_b1 = V_t-I_a1*(R_a+R_sh) // initial back emf
    in V
23 E_b2 = V_t-I_a2*(R_a+R_sh) // final back emf in
    V
24 // phi_2 = 0.4*phi_1 value of flux
25 N_2 = (E_b2/E_b1)*(2.5)*N_1 // motor speed when
    line current is 15A in rpm
26
27 // display the result
28 disp("Example 5.7 solution");
29 printf(" \n motor speed when line current is 15A \n
    N_2 = %.0f rpm \n", N_2 );
30
31 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

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 5 : DIRECT CURRENT MOTORS
7 // Example : 5.8
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 I_L1 = 5 // dc shunt motor current in A
13 V_t = 230 // supply voltage in V
14 R_a = 0.25 // armature resistance in ohm
15 R_sh = 115 // field resistance in ohm
16 I_L2 = 40 // dc shunt motor current in A
17
18
19 // caclulations
20 // at noload condition
21 P_in1 = V_t*I_L1 // input
    power in W
22 I_sh = V_t/R_sh // shunt
    field current in A
23 I_a1 = I_L1-I_sh // armature
    current in A
24 P_acu1 = I_a1^2*R_a // armature
    copper loss in W
25 P_shcu = I_sh^2*R_sh //shunt
    field copper loss in W
26 P_iron_friction = P_in1-(P_acu1+P_shcu) // iron and
    friction losses in W
27 // under load condition
28 I_a2 = I_L2-I_sh // armature
    current in A
29 P_acu2 = I_a2^2*R_a // armature
```

```

    copper loss in W
30 P_loss = P_iron_friction+P_shcu+P_acu2 // total
    losses in W
31 P_in2 = V_t*I_L2 // input
    power in W
32 P_0 = P_in2-P_loss // output
    power in W
33 n = (P_0/P_in2)*100 //
    efficiency in percent
34
35 // display the result
36 disp("Example 5.8 solution");
37 printf(" \n iron and friction losses \n
    P_iron_friction = %.2f W \n", P_iron_friction );
38 printf(" \n efficiency \n n = %.0f percent \n", n)

```

---

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

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 5 : DIRECT CURRENT MOTORS
7 // Example : 5.9
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 I_L= 80 // dc shunt motor current in A
13 V_t = 220 // supply voltage in V
14 N = 800 // speed in rpm
15 R_a = 0.1 // armature resistance in ohm
16 R_sh = 50 // shunt field resistance in ohm
17 P_if = 1600 // iron and friction losses in W

```

```

18
19 // caclulations
20 I_sh = V_t/R_sh // shunt field current in A
21 I_a = I_L-I_sh // armature current
22 E_b = V_t-I_a*R_a // back emf in V
23 P_in = V_t*I_L // i/p power in W
24 P_md = E_b*I_a // mechanical power developed
    in the armature in W
25 P_cu = P_in-P_md // copper loss in W
26 T_a = 9.55*(E_b*I_a)/N // armature torque in N-m
27 P_0 = P_md-P_if // o/p power in W
28 T_sh = 9.55*(P_0/N) // shaft torque in N-m
29 n= (P_0/P_in)*100 // efficiency
30
31 // display the result
32 disp("Example 5.9 solution");
33 printf(" \n Copper loss \n P_cu = %.2f W \n", P_cu);
34 printf(" \n Armature torque \n T_a = %.2f N-m \n",
    T_a);
35 printf(" \n Shaft torque \n T_sh = %.2f N-m \n",
    T_sh);
36 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

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 6 : CONTROL AND STARTING OF A DC MOTORS
7 // Example : 6.1
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V_t = 220 // supply voltage in V
13 I_a1 = 10 // dc shunt motor armature current in A
14 N_1 = 900 // speed in rpm
15 R_a = 1 // armature resistance in ohm
16 N_2 = 500 // speed in rpm
17
18 // caclulations
19 E_b1 = V_t - I_a1 * R_a // initial back emf in
```

```

20 R = (E_b1/10)*(1-(N_2/N_1)) // additional value of
    resistance in ohm
21
22 // display the result
23 disp("Example 6.1 solution");
24 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

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 6 : CONTROL AND STARTING OF A DC MOTORS
7 // Example : 6.2
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V_t = 230 // supply voltage in V
13 I_a1 = 15 // dc shunt motor armature current in A
14 N_1 = 650 // speed in rpm
15 R_a = 0.4 // armature resistance in ohm
16 R = 1 // variable resistance in series with
    the armature
17
18 // caclulations
19 // at full load
20 E_b1 = V_t-I_a1*R_a // initial back emf in V
21 E_b2 = V_t-I_a1*(R+R_a) // final back emf in V
22 N_2 = N_1*(E_b2/E_b1) // speed at full load in
    rpm

```



```

23
24 // at half load
25 I_a21 = I_a1/2           // armature current in
    A
26 E_b21 = V_t-I_a21*(R+R_a) // back emf in V
27 N_21 = N_1*(E_b21/E_b1)  // speed at half load
    torque in rpm
28
29
30
31 // display the result
32 disp("Example 6.2 solution");
33 printf("\n speed at full load \n N_2 = %.1f rpm \n"
    , N_2);
34 printf("\n speed at half load torque \n N_21 = %.1f
    rpm \n", N_21);

```

---

### Scilab code Exa 6.3 find new speed

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 6 : CONTROL AND STARTING OF A DC MOTORS
7 // Example : 6.3
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V_t = 230 // supply voltage series motor in V
13 R_a = 0.2 // armature resistance in ohm
14 R_f = 0.2 // field resistance in ohm
15 I_a1 = 10 // dc series motor armature current in
    A

```

```

16 N = 1000      // speed in rpm
17 I_a2 = 8      // dc series motor armature current in
    A when R=2 ohm
18 R_2 = 2      // series resistance in ohm
19 R_1 = 0      // series resistance in ohm
20 // phi2 = 0.8*phi1
21 // phi1/phi2 = 1.25
22
23 // caclulations
24 R_a1 = R_a+R_1      // new armature resistance
    in ohm
25 E_b1 = V_t-I_a1*R_a1      // back emf in V
26 R_a2 = R_a+R_2      // new armature resistance
    in ohm
27 E_b2 = V_t-I_a2*R_a2      // back emf in V
28 N_2 = (E_b2/E_b1)*1.25*N  // new speed in rpm
29
30 // display the result
31 disp("Example 6.3 solution");
32 printf(" \n New speed \n N_2 = %.1f rpm \n", N_2);

```

---

**Scilab code Exa 6.4** determine the speed

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 6 : CONTROL AND STARTING OF A DC MOTORS
7 // Example : 6.4
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 N = 1100      // speed of dc series motor in rpm

```

```

13 P = 4           // number of poles
14 I_a1 = 15      // supply current to dc series
    motor in A
15 V_t = 220     // supply voltage in V
16 R_a = 0.9     // series armature resistance in
    ohm
17 R_se1 = 0.6   // series field resistance in ohm
18 I_a2 = 25     // supply current to dc series
    motor in A
19 // phi2 = 0.8*phi1
20 // phi1/phi2 = 1.25
21
22 // caclulations
23 E_b1 = V_t-I_a1*(R_a+R_se1) // back emf in V
24 R_se2 = 0.6/4           // value of resistance
    per path in ohm
25 E_b2 = V_t-I_a2*(R_a+R_se2) // back emf in V
26 N_2 = (E_b2/E_b1)*1.25*N   // new speed in rpm
27
28 // display the result
29 disp("Example 6.4 solution");
30 printf(" \n New speed \n N_2 = %.1f rpm \n", N_2);

```

---

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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 6 : CONTROL AND STARTING OF A DC MOTORS
7 // Example : 6.5
8
9 clc;clear; // clears the console and command history
10

```

```

11 // Given data
12 V_t = 230 // shunt motor supply voltage in V
13 R_a = 0.4 // armature resistance in ohm
14 I_a = 30 // armature current in A
15 n = 3 // number of steps
16
17 // caclulations
18 R_1 = V_t/I_a // maximum value of current in
    ohm
19 k = (R_1/R_a)^(1/3) // constant
20 R_2 = R_1/k // other value of resistance in
    ohm
21 R_3 = R_2/k // other value of resistance in
    ohm
22 R_4 = R_3/k // other value of resistance in
    ohm
23 R_1step = R_1-R_2 // resistance of the 1st step
    in ohm
24 R_2step = R_2-R_3 // resistance of the 1st step
    in ohm
25 R_3step = R_3-R_4 // resistance of the 1st step
    in ohm
26
27 // display the result
28 disp("Example 6.5 solution");
29 printf("\n resistance of the 1st step in ohm \n
    R_1step = %.1f ohm \n", R_1step);
30 printf("\n resistance of the 2nd step in ohm \n
    R_2step = %.1f ohm \n", R_2step);
31 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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES

```

```

2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 6 : CONTROL AND STARTING OF A DC MOTORS
7 // Example : 6.6
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V_t = 220 // shunt motor supply voltage in V
13 P_0 = 3550 // o/p power in W
14 n = 0.85 // efficiency
15 // condition given : starting current of the motor
    should not exceed twice the load current
16
17 // caclulations
18 P_in = P_0/n // i/p power in W
19 P_tloss = P_in-P_0 // total loss in W
20 I_a = P_in/V_t // neglet the field current
    then armature current in A
21 P_cu = P_tloss/2 // copper loss in W
22 R_a = P_cu/I_a^2 // armature resistance in ohm
23 I_1 = 2*I_a // maximum starting current in
    A
24 R_1 = V_t/I_1 // maximum resistance in ohm
25 k = (R_1/R_a)^(1/4) // constant
26 R_2 = R_1/k // other value of resistance
    in ohm
27 R_3 = R_2/k // other value of resistance
    in ohm
28 R_4 = R_3/k // other value of resistance
    in ohm
29 R_5 = R_4/k // other value of resistance
    in ohm
30 R_1step = R_1-R_2 // resistance of the 1st step
    in ohm
31 R_2step = R_2-R_3 // resistance of the 1st step

```

```

    in ohm
32 R_3step = R_3-R_4      // resistance of the 1st step
    in ohm
33 R_4step = R_4-R_5      // resistance of the 1st step
    in ohm
34
35 // display the result
36 disp("Example 6.6 solution");
37 printf(" \n resistance of the 1st step in ohm \n
    R_1step = %.1f ohm \n", R_1step);
38 printf(" \n resistance of the 2nd step in ohm \n
    R_2step = %.2f ohm \n", R_2step);
39 printf(" \n resistance of the 3rd step in ohm \n
    R_3step = %.2f ohm \n", R_3step);
40 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

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 7 : THREE-PHASE INDUCTION MOTOR
7 // Example : 7.1
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V = 230 // supply voltage in V
13 P = 4 // number of poles
14 f = 50 // frequency Hz
15 N_1 = 1445 // speed in rpm
16
17 // calculations
18 N_s = 120*f/P // synchronous speed in rpm
19 s = (N_s-N_1)/N_s // slip
```

```

20 f_r = s*f           // rotor frequency in Hz
21
22 // display the result
23 disp("Example 7.1 solution");
24 printf(" \n Synchronous speed \n N_s = %.1f rpm \n",
        N_s);
25 printf(" \n Slip \n s = %.4f \n", s);
26 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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 7 : THREE-PHASE INDUCTION MOTOR
7 // Example : 7.2
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 E_BR = 120 // voltage found under blocked
           condition in V
13 P = 4 // number of poles
14 f = 50 // frequencny in Hz
15 N_1 = 1450 // speed in rpm
16
17 // caclulations
18 N_s = 120*f/P // synchronous speed in rpm
19 s = (N_s-N_1)/N_s // slip
20 f_r = s*f // rotor frequency in Hz
21 E_r = s*E_BR // Rotor voltage in V
22

```



```

23 // display the result
24 disp("Example 7.2 solution");
25 printf(" \n Synchronous speed \n N_s = %.1f rpm \n",
        N_s);
26 printf(" \n Slip \n s = %.3f \n", s);
27 printf(" \n Rotor frequency \n f_r = %.2f Hz \n",
        f_r);
28 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

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 7 : THREE-PHASE INDUCTION MOTOR
7 // Example : 7.3
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V_0 = 230 // supply voltage in V
13 P = 4 // number of poles
14 T_0 = 230 // original torque in N-m
15 V_s = 150 // stator voltage in V
16 I_0 = 560 // starting current in A
17
18 // calculations
19 T_st = (V_s/V_0)^2*T_0 // starting torque in N-m
20 I_st = I_0*(V_s/V_0) // starting current in A
21
22 // display the result
23 disp("Example 7.3 solution");
24 printf(" \n Starting torque \n T_st = %.1f N-m \n",

```

```

    T_st);
25 printf("\n Starting current \n I_st = %.1f A \n",
    I_st);

```

---

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

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 7 : THREE-PHASE INDUCTION MOTOR
7 // Example : 7.4
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 f = 50 // frequency in Hz
13 P = 8 // number of poles
14 a = 0.03 // full load slip
15 R_2 = 0.01 // rotor resistance in ohm
16 X_2 = 0.1 // standstill resistance in ohm
17
18 // caclulations
19 N_s = 120*f/P // synchronous speed in
    rpm
20 s = R_2/X_2 // slip at maximum torque
21 N_1 = (1-s)*N_s // rotor maximum speed in
    rpm
22 T = (2*a*s)/(s^2+a^2) // ratio of full load
    torque to maximum torque
23 T_ratio = 1/T // ratio of maximum torque
    to full load torque
24
25 // display the result

```

```

26 disp("Example 7.4 solution");
27 printf(" \n Rotor speed at maximum torque \n N_1 = %
    .0f rpm \n", N_1);
28 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 po

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 7 : THREE-PHASE INDUCTION MOTOR
7 // Example : 7.5
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V = 440 // supply voltage in V
13 P = 4 // number of poles
14 P_ag = 1500 // rotor i/p in kW
15 P_rcu = 250 // copper loss in W
16 f = 50 // frequency in Hz
17
18 // caclulations
19 s = P_rcu/P_ag // slip
20 N_s = 120*f/P // synchronous speed in rpm
21 N_1 = (1-s)*N_s // rotor maximum speed in rpm
22 P_mech = (1-s)*P_ag // mechanical power developed
    in W
23
24 // display the result
25 disp("Example 7.5 solution");
26 printf(" \n Slip \n s = %.3f \n", s);

```

```

27 printf(" \n Synchronous speed \n N_s = %.f rpm \n",
    N_s);
28 printf(" \n Rotor speed \n N_l = %.0f rpm \n", N_l);
29 printf(" \n Mecahnical power developed \n P_mech = %
    .0f W \n", P_mech);
30
31 // NOTE : small change in answer instead 1245 got
    1250W

```

---

Scilab code Exa 7.6 determine max mechanical power Tmax and slip

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 7 : THREE-PHASE INDUCTION MOTOR
7 // Example : 7.6
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V_1 = 150 // supply voltage in V
13 P = 4 // number of poles
14 f = 50 // frequency in Hz
15 Z_1 = 0.12+%i*0.16 // per phase standstill
    stator impedance in ohm
16 Z_2 = 0.22+%i*0.28 // per phase standstill rotor
    impedance in ohm
17 R_2 = real(Z_2) // from Z_2
18
19 // caclulations
20 Z_eq = Z_1+Z_2 //
    equivalent impedance in ohm
21 P_mech = 3*V_1^2/(2*(R_2+abs(Z_eq))) // maximum

```

```

    mechanical power developed in W
22 s_mp = R_2/(abs(Z_eq)+R_2)           // slip
23 W_s = 2*pi*2*f/P                   // since N_s
    = f/(P/2) and W_s = 2*pi*N_s
24 W = (1-s_mp)*W_s                   // speed of
    rotor in rad/s
25 T_mxm = P_mech/W                   // maximum
    torque in N-m
26
27 // display the result
28 disp("Example 7.6 solution");
29 printf("\n maximum mechanical power developed \n
    P_mech = %.f W \n", P_mech);
30 printf("\n Maximum torque \n T_mxm = %.3f N-m \n",
    T_mxm);
31 printf("\n Maximum slip \n s_mp = %.2f \n", s_mp);
32 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

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 7 : THREE-PHASE INDUCTION MOTOR
7 // Example : 7.7
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V = 440           // supply voltage in V
13 P = 6            // number of poles
14 f = 50          // frequency in Hz

```

```

15 P_a = 45000      // i/p power
16 N_1 = 900       // speed in rpm
17 P_tloss = 2000  // total power loss in W
18
19 // caclulations
20 N_s = 120*f/P    // synchronous speed in rpm
21 s = (N_s-N_1)/N_s // slip
22 P_ag = (P_a-P_tloss) // air gap power in W
23 P_rcu = s*P_ag   // rotor copper loss in W
24 P_mech = P_ag-P_rcu // mechanical power in W
25 P_0 = P_mech-3000 // o/p power in W
26 n = (P_0/P_ag)*100 // efficiency since n = P_o
    /P_in
27
28 // display the result
29 disp("Example 7.7 solution");
30 printf(" \n Slip \n s = %.1f \n", s );
31 printf(" \n Rotor copper loss \n P_rcu = %.f W \n",
    P_rcu );
32 printf(" \n Out put power \n P_0 = %.f W \n", P_0 );
33 printf(" \n Efficiency \n n = %.f percent \n", n );

```

---

#### Scilab code Exa 7.8 determine length

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 7 : THREE-PHASE INDUCTION MOTOR
7 // Example : 7.8
8
9 clc;clear; // clears the console and command history
10
11 // Given data

```

```

12 v_s = 120          // train speed in km/h
13 f = 50            // frequency in Hz
14
15 // caclulations
16 v_s = 120*1000/(60*60) // train speed in m/s
17 w = v_s/(2*f)      // length of the pole-
    pitch in m
18
19 // display the result
20 disp("Example 7.8 solution");
21 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

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 8 : STARTING, CONTROL AND TESTING OF AN
  INDUCTION MOTOR
7 // Example : 8.2
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 T_ratio = 1/2 // T_st/T_f ratio of starting
  torque to full load torque
13 s_f = 0.03 // full load slip
14 I_ratio = 5 // I_sc/I_f ratio of short circuit
  current to full load current
```



```

15
16 // caclulations
17 x = (1/I_ratio)*sqrt(T_ratio*(1/s_f)) // percentage
    of tapping
18
19 // display the result
20 disp("Example 8.2 solution");
21 printf(" \n Percentage of tapping \n x = %.3f \n", x
    );

```

---

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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 8 : STARTING, CONTROL AND TESTING OF AN
    INDUCTION MOTOR
7 // Example : 8.3
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 T_ratio = 0.25 // T_st/T_f ratio of starting
    torque to full load torque
13 I_ratio = 3.6 // I_sc/I_f ratio pf short
    circuit current to full load current
14 // since given I_sc = 3*1.2*I_f
15
16 // caclulations
17 s_f = T_ratio*3/(I_ratio)^2 // full load slip
18
19 // display the result
20 disp("Example 8.3 solution");

```

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

---

#### Scilab code Exa 8.4 determine torques

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 8 : STARTING, CONTROL AND TESTING OF AN
  INDUCTION MOTOR
7 // Example : 8.4
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 Z_ocr = 0.04+%i*0.5 // inner cage impedance
    per phase at standstill
13 Z_ocr = 0.4+%i*0.2 // outer cage impedance
    per phase at standstill
14 V = 120 // per phase rotor
    induced voltage at standstill in V
15
16 // caclulations
17 Z_com = (Z_ocr*Z_ocr)/(Z_ocr+Z_ocr) // combined
    impedance in ohm
18 I_2 = V/abs(Z_com) // rotor
    current per phase in A
19 R_2 = real(Z_com) // combined
    rotor resistance in ohm from Z_com
20 T = I_2^2*R_2 // torque
    at stand still condition in syn.watt
21 s = 0.06 // slip of 6
    %
22 Z_ocr6 = 0.04/s+%i*0.5 // inner cage
```

```

    impedance per phase at standstill
23 Z_ocr6 = 0.4/s+%i*0.2          // outer cage
    impedance per phase at standstill
24 Z_com6 = ((Z_ocr6)*Z_ocr6)/((Z_ocr6)+Z_ocr6) //
    combined impedance in ohm at 6% slip
25 I2_6 = V/abs(Z_com6)          // rotor
    current per phase in A at 6% slip
26 R2_6 = real(Z_com6)           //
    combined rotor resistance in ohm from Z_com6
27 T6 = I2_6^2*R2_6             // torque at
    stand still condition in syn.watt
28
29 // display the result
30 disp("Example 8.4 solution");
31 printf(" \n Torque at stand still condition \n T = %
    .2f syn.watt \n", T );
32 printf(" \n Torque at stand 6 percent slip \n T6 = %
    .2f syn.watt \n", T6 );

```

---

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

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 8 : STARTING, CONTROL AND TESTING OF AN
  INDUCTION MOTOR
7 // Example : 8.5
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V = 210      // supply voltage in V
13 f = 50      // supply frequency in Hz

```

```

14 P = 50          // i/p power in W
15 I_br = 2.5     // line current in A
16 V_br = 25     // line voltage
17 R_1 = 2.4     // resistance between two resistance in
    ohm
18
19 // calculations
20 V_br = V_br/sqrt(3) // phase voltage
21 P_br = P/3        // power per phase in W
22 R_eq = P_br/I_br^2 // equivalent resistance
    in ohm
23 R_2 = R_eq-(R_1/2) // per phase rotor
    resistance in ohm
24 Z_eq = V_br/I_br  // equivalent impedance
    in ohm
25 X_eq = sqrt(Z_eq^2-R_2^2) // equivalent reactance
    in ohm
26 X_1 = 0.5*X_eq
27
28 // display the result
29 disp("Example 8.5 solution");
30 printf(" \n Equivalent resistance \n R_eq = %.1f ohm
    \n", R_eq );
31 printf(" \n Equivalent reactance \n X_eq = %.1f ohm
    \n", X_eq );
32 printf(" \n Equivalent reactance \n X_1 = %.1f ohm \
    n", X_1 );

```

---

**Scilab code Exa 8.6** determine equivalent parameters

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5

```

```

6 // Chapter 8 : STARTING, CONTROL AND TESTING OF AN
  INDUCTION MOTOR
7 // Example : 8.5
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V = 210 // supply voltage in V
13 f = 50 // supply frequency in Hz
14 P = 4 // number of poles
15 P_0 = 400 // i/p power in W
16 I_0 = 1.2 // line current in A
17 V_0 = 210 // line voltage
18 P_fw = 150 // total friction and windage losses in
  W
19 R = 2.2 // stator resistance between two
  terminals in ohm
20
21 // caclulations
22 R_1 = R/2 // per phase
  stator resistance in ohm
23 P_scu = 3*I_0^2*R_1 // copper
  loss in W
24 P_core = P_0-P_fw-P_scu // stator
  core loss in W
25 R_0 = (V_0/sqrt(3))^2/(P_core/3) // no-load
  resistance in ohm
26 // alternate approach
27 phi_0 = acosd(P_core/(sqrt(3)*V_0*I_0)) // power
  factor angle
28 X_0 = (V_0/sqrt(3))/(I_0*sind(phi_0)) //
  magnetizing reactance per phase in ohm
29
30 // display the result
31 disp("Example 8.6 solution");
32 printf(" \n No-load resistance \n R_0 = %.1f ohm \n"
  , R_0 );
33 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

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 8 : STARTING, CONTROL AND TESTING OF AN
  INDUCTION MOTOR
7 // Example : 8.7
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 P_1 = 6 // number of pole
13 P_2 = 4 // number of pole
14 f =50 // supply frequency in Hz
15 P = 60 // power in kW
16
17 // caclulations
18 s = P_2/(P_1+P_2) // combined slip
19 N_cs = 120*f/(P_1+P_2) // combined synchronous
  speed in rpm
20 P_0 = P*(P_2/(P_1+P_2)) // o/p of 4-pole motor in
  kW
21
22 // display the result
23 disp("Example 8.7 solution");
24 printf(" \n Combined slip \n s = %.1f \n", s );
25 printf(" \n Combined synchronous speed \n N_cs = %.0
  f rpm \n", N_cs );
26 printf(" \n Out-power of 4-pole motor \n P_0 = %.f
  kW \n", P_0 );
```



# Chapter 9

## SYNCHRONOUS GENERATOR

Scilab code Exa 9.1 determine number of poles

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 9 : SYNCHRONOUS GENERATOR
7 // Example : 9.1
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 N = 300 // speed of water turbine in rpm
13 f = 50 // frequency in Hz
14
15 // calculations
16 P = 120*f/N // number of poles
17
18 // display the result
19 disp("Example 9.1 solution");
```



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

---

**Scilab code Exa 9.2** determine pitch factor

```
1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 9 : SYNCHRONOUS GENERATOR
7 // Example : 9.2
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 P = 8 // number of poles
13 m = 3 // number of phase
14 S = 144 // number of slots
15
16 // caclulations
17 T_p = S/P // pole pitch interms of slots
18 slots_1 = 180/T_p // pole pitch per slots
19 y = 2*slots_1 // short pitch angle in degree
20 k_p = cosd(y/2) // pitch factor
21
22 // display the result
23 disp("Example 9.2 solution");
24 printf(" \n Pitch factor is \n k_p = %.2f \n", k_p )
    ;
```

---

**Scilab code Exa 9.3** determine pitch factor

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 9 : SYNCHRONOUS GENERATOR
7 // Example : 9.3
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 P = 4 // number of poles
13 m = 3 // number of phase
14 S = 40 // number of slots
15
16 // caclulations
17 T_p = S/P // pole pitch interms of slots
18 T_c = 9-1 // coil span 1 to 9 i.e. coil
    pitch in terms of slots
19 slots_1 = 180/T_p // pole pitch per slots
20 y = T_p-T_c // short pitch angle
21 y_angle = y*slots_1 // in terms of angle
22 k_p = cosd(y_angle/2) // pitch factor
23
24 // display the result
25 disp("Example 9.3 solution");
26 printf(" \n Pitch factor is \n k_p = %.2f \n", k_p )
    ;

```

---

**Scilab code Exa 9.4** find distribution factor

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION

```

```

5
6 // Chapter 9 : SYNCHRONOUS GENERATOR
7 // Example : 9.4
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 P = 4 // number of poles
13 S = 48 // number of slots
14
15 // caclulations
16 T_p = S/P //
    pole pitch interms of slots
17 slots_1 = 180/T_p //
    pole pitch per slots
18 n = S/(P*3) //
    number of slots or coils per pole per phase
19 k_d = sind((n*slots_1)/2)/(n*sind(slots_1/2)) //
    distribution factor
20
21 // display the result
22 disp("Example 9.4 solution");
23 printf(" \n Distribution factor is \n k_d = %.2f \n"
    , k_d );

```

---

**Scilab code Exa 9.5** determine line voltage

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 9 : SYNCHRONOUS GENERATOR
7 // Example : 9.5
8

```

```

9  clc;clear; // clears the console and command history
10
11 // Given data
12 P = 12 // number of poles
13 S = 180 // number of slots
14 phi_m = 0.05 // flux per pole in Wb
15 N = 600 // speed of machine in rpm
16
17 // caclulations
18 T_p = S/P //
    pole pitch interms of slots
19 slots_1 = 180/T_p //
    pole pitch per slots
20 n = S/(P*3) // number
    of slots or coils per pole per phase
21 k_d = sind((n*slots_1)/2)/(n*sind(slots_1/2)) //
    distribution factor
22 k_p = 1 //
    pitch factor
23 Z = (180/3)*slots_1 //
    number of conductor per phase
24 T = Z/2 //
    number of turns per phase
25 f = P*N/120 //
    frequency in Hz
26 E = 4.44*k_p*k_d*f*phi_m*T //
    induced voltage in V
27 E_L = sqrt(3)*E //
    line voltage in V
28
29 // display the result
30 disp("Example 9.5 solution");
31 printf(" \n Line voltage is \n E_L = %.0f V \n", E_L
    );
32
33 // NOTE : correction in answer

```

---

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

```
1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 9 : SYNCHRONOUS GENERATOR
7 // Example : 9.6
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 P = 4 // number of poles
13 m = 3 // number of phase
14 f = 50 // frequency in Hz
15 phi_m = 0.05 // flux per pole in Wb
16
17 // caclulations
18 T_p = 6*3 // pole pitch interms of slots 6 slots per
    pole pitch per phase hence for 3 phase
19 slots_1 = 180/T_p //
    pole pitch per slots
20 T_c = (5/6)*T_p //
    coil per pitch
21 y = T_p-T_c //
    short pitch angle
22 y_angle = y*10 //
    short pitch in terms of angle
23 k_p = cosd(y_angle/2) //
    pitch factor
24 n = 6 //
    number of slots
25 k_d = sind((n*slots_1)/2)/(n*sind(slots_1/2)) //
```

```

    distribution factor
26 T = (1/2)*n*P*2*5 // 2=
    no. of layers , 5=conductor per layer
27 E = 4.44*k_p*k_d*f*phi_m*T //
    induced voltage in V
28
29 // display the result
30 disp("Example 9.6 solution");
31 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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 9 : SYNCHRONOUS GENERATOR
7 // Example : 9.7
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 P = 10 // number of poles
13 m = 3 // number of phase
14 f = 50 // frequency in Hz
15 phi_m1 = 0.05 // flux per pole in Wb
16 phi_m3 = 0.006 // flux per pole in Wb
17 T_c = 150 // coil
18
19 // caclulations
20 T_p = 3*3 // pole pitch interms of slots 3 slots per
    pole per phase hence for 3 phase
21 slots_1 = 180/T_p //

```

```

    pole pitch per slots
22 y = 180-T_c //
    short pitch angle
23 n = 3 //
    number of slots
24 k_p1 = cosd(y/2) //
    pitch factor
25 k_d1 = sind((n*slots_1)/2)/(n*sind(slots_1/2)) //
    distribution factor
26 E_1 = 4.44*k_p1*k_d1*f*phi_m1*T_c //
    induced voltage in V
27 k_p3 = cosd(y/2) //
    pitch factor
28 k_d3 = sind((n*slots_1)/2)/(n*sind(slots_1/2)) //
    distribution factor
29 E_3 = 4.44*k_p3*k_d3*f*phi_m3*T_c //
    induced voltage in V
30 E = sqrt(E_1^2+E_3^2) //
    induced voltage per phase in V
31
32 // display the result
33 disp("Example 9.7 solution");
34 printf("\n Induced voltage per phase is \n E = %.0 f
    V \n", E );

```

---

**Scilab code Exa 9.8** determine no-load induced voltage per phase and VR

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 9 : SYNCHRONOUS GENERATOR
7 // Example : 9.8
8

```

```

9  clc;clear; // clears the console and command history
10
11 // Given data
12 kVA = 50 // kVA ratings
13 V_t = 220 // terminal voltage in V
14 R_a = 0.011 // effective resistance in ohm
15 X_s = 0.09 // synchronous reactance in ohm
16
17 // caclulations
18 phi = acosd(0.85)
    // since power factor in 0.85
19 I_a = kVA*10^3/V_t //
    armature current in A
20 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
21 VR = ((E_f-V_t)/V_t)*100 //
    voltage regulation
22
23 // display the result
24 disp("Example 9.8 solution");
25 printf(" \n No-load induced voltage per phase \n E_f
    = %.1f V \n", E_f );
26 printf(" \n Voltage regulation is \n VR = %.1f
    percent \n", VR );

```

---

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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 9 : SYNCHRONOUS GENERATOR
7 // Example : 9.9
8

```



```

 9 clc;clear; // clears the console and command history
10
11 // Given data
12 kVA = 200 // kVA ratings
13 V_t = 33*10^3 // terminal voltage in V
14 R_a = 0.54 // armature resistance in ohm
15 V_L = 415 // voltage between lines for SC test
    in V
16 I_sh = 25 // short circuit current in A
17
18 // caclulations
19 phi = acosd(0.9) // since power factor
    in 0.9
20 V_P = V_L/sqrt(3) // phase voltage
    during short circuit test in V
21 Z_s = V_P/I_sh // synchronous
    impedance in ohm
22 X_s = sqrt(Z_s^2 - R_a^2) // synchronous
    reactance in ohm
23 I_a = kVA/(sqrt(3)*V_t*10^-3) // full loa current in
    A
24 V_ta = V_t/sqrt(3) // voltage per phase
    alternator
25 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
26 VR = ((E_f-V_ta)/V_ta)*100 // voltage regulation
27
28 // display the result
29 disp("Example 9.9 solution");
30 printf(" \n Synchronous impedance \n Z_s = %.1f ohm
    \n", Z_s );
31 printf(" \n Synchronous reactance is \n X_s = %.2f
    ohm \n", X_s );
32 printf(" \n Voltage regulation is \n VR = %.2f
    percent \n", VR );
33 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

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 9 : SYNCHRONOUS GENERATOR
7 // Example : 9.10
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 MVA = 30 // MVA ratings
13 V = 20 // supply voltage in kV
14 N = 1800 // speed in rpm
15 V_t = 15 // voltage per phase in kV
16 E_f = 10 // per phase terminal voltage in kV
17 teta = 40 // power angle in degree
18 X_s = 6 // 3 phase synchronous reactance in ohm
19
20 // caclulations
21 P = 3*V_t*E_f*sind(teta)/X_s // power delivered to
    the load in MW
22 P_max = 3*V_t*E_f/X_s // three phase
    maximum power in MW
23
24 // display the result
25 disp("Example 9.10 solution");
26 printf(" \n Three phase power delivered to the load
    \n P = %.2f MW \n", P );
27 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

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 9 : SYNCHRONOUS GENERATOR
7 // Example : 9.11
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 kVA = 25 // kVA ratings
13 V = 440 // supply voltage in V
14 f = 50 // supply frequency in Hz
15 pf = 0.8 // lagging power factor
16 R_a = 0.3 // resistance of machine per phase
    in ohm
17 X_d = 5 // reactance of machine per phase in
    ohm
18 X_q = 3 // reactance of machine per phase in
    ohm
19
20 // caclulations
21 phi = acosd(pf)
22 V_t = V/sqrt(3) //
    terminal voltage per phase in V
23 I_a = 25*10^3/(sqrt(3)*V) //
    armature current
24 delta = atand(I_a*X_q*pf/(V_t+I_a*X_q*sind(phi))) //
    torque angle
25 I_d = I_a*sind(delta+phi) // direct
    axis component of the current in A
```

```

26 E_f = V_t*cosd(delta)+I_d*X_d //
    induced voltage per phase in V
27 VR = ((E_f-V_t)/V_t)*100 //
    voltage regulation in V
28
29 // display the result
30 disp("Example 9.11 solution");
31 printf(" \n Torque angle \n delta = %.2f degree \n",
    delta );
32 printf(" \n Induced voltage per phase \n E_f = %.2f
    V \n", E_f );
33 printf(" \n Voltage regulation \n VR = %.2f percent
    \n", VR );

```

---

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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 9 : SYNCHRONOUS GENERATOR
7 // Example : 9.12
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 E_1 = 220 // induced voltage per phase by an
    alternator1
13 E_2 = 220*exp(%i*5*pi/180) // induced voltage per
    phase by an alternator2
14 Z_1 = %i*3 // impedance of an alternator1
15 Z_2 = %i*4 // impedance of an alternator2
16 R = 5 // resistance in ohm
17 Z = 5

```

```

18
19 // caclulations
20 I = (E_1*Z_2+E_2*Z_1)/(Z_1*Z_2+Z*(Z_1+Z_2))
    // load current in A
21 V_t = I*R
    // terminal voltage in V
22 I_a1 = ((E_1-E_2)*Z+E_1*Z_2)/(Z_1*Z_2+Z*(Z_1+Z_2))
    // armature current in A
23 D = atand(imag(I),real(I))// from V_t
24 A = atand(imag(V_t),real(V_t))
                                     //
    from I_a1
25 P_1 = abs(V_t*I_a1)*cosd(D-A)      // power per
    phase delivered by the 1st alternator in W
26
27 // display the result
28 disp("Example 9.12 solution");
29 printf(" \n load current \n I = %.1f<%.2f degree \n"
    , abs(I),atand(imag(I),real(I)) );
30 printf(" \n terminal voltage \n V_t = %.f<%.2f V \n"
    , abs(V_t),atand(imag(V_t),real(V_t)) );
31 printf(" \n Power per phase delivered by the 1st
    alternator \n P_1 = %.2f W \n", P_1 );
32 printf(" \n NOTE : ERROR : Calculation mistakes in
    textbook \n")

```

---

# Chapter 10

## SYNCHRONOUS MOTOR

Scilab code Exa 10.1 determine excitation voltage per phase

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 10 : SYNCHRONOUS MOTOR
7 // Example : 10.1
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V = 2.5*10^3 // supply voltage in V
13 R_r = 0.12 // per phase resistance in ohm
14 X_r = 3.2 // synchronous reactance in ohm
15 I_a = 185 // line current in A
16 pf = 0.8 // leading power factor
17
18 // caclulations
19 phi = acosd(pf)
20 V_t = V/sqrt(3) // terminal voltage per phase in
    V
```

```

21 Z_s = R_r+%i*X_r // impedance per phase ohm
22 beta = atand(X_r/R_r)
23 E_r = I_a*Z_s // resultant voltage due to
    impedance in V
24 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
25
26
27 // display the result
28 disp("Example 10.1 solution");
29 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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 10 : SYNCHRONOUS MOTOR
7 // Example : 10.2
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 kVA = 1200 // kVA ratings
13 V = 14*10^3 // supply voltage in V
14 R_r = 4.8 // per phase resistance in ohm
15 X_r = 35 // synchronous reactance in ohm
16 pf = 0.95 // leading power factor
17
18 // caclulations
19 phi = acosd(pf)
20 Z_s = R_r+%i*X_r // impedance per phase

```

```

    ohm
21 I_a = kVA*10^3/(sqrt(3)*V) // armature current in A
22 E_r = I_a*Z_s // resultant voltage due
    to impedance in V
23 V_t = V/sqrt(3) // terminal voltage per
    phase in V
24 b = atand(X_r/R_r) // beta value
25 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
26 teta = sind(64)
27 D = (E_r*teta/E_f) // torque angle
28 delta = asind(abs(D))
29
30 // display the result
31 disp("Example 10.2 solution");
32 printf(" \n Excitation voltage per phase \n E_f = %
    .2f V \n", E_f );
33 printf(" \n Torque angle at 0.95 power factor
    lagging \n delta = %.2f degree \n", delta );

```

---

**Scilab code Exa 10.3** calculate maximum power

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 10 : SYNCHRONOUS MOTOR
7 // Example : 10.3
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V = 440 // supply voltage in V
13 R_a = 1.5 // per phase resistance in ohm

```



```

14 X_a = 8           // synchronous reactance in ohm
15 P = 4            // number of poles
16 f = 50          // supply frequency in Hz
17 pf = 0.9        // leading power factor
18 I_a = 50        // armature current in A
19
20 // caclulations
21 V_t = V/sqrt(3)  // terminal voltage per phase
    in V
22 phi = acosd(pf) // angle in degree
23 Z_s = R_a+%i*X_a // impedance per phase ohm
24 E_r = I_a*abs(Z_s) // resultant voltage due
    to impedance in V
25 bet = atand(X_a/R_a)
26 E_f = sqrt(V_t^2+E_r^2-2*V_t*E_r*cosd(bet+phi)) //
    excitation voltage per phase in V
27 P_dm = (((E_f*V_t)/Z_s)-((E_f^2*R_a)/Z_s^2))
    // maximum power per phase in W
28
29 // display the result
30 disp("Example 10.3 solution");
31 printf(" \n Maximum power per phase \n P_dm = %.2f W
    \n", P_dm );
32 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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 10 : SYNCHRONOUS MOTOR
7 // Example : 10.4

```

```

8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 P = 4 // number of poles
13 f = 50 // supply frequency in Hz
14 V_t = 1500 // terminal voltage per phase in V
15 E_f = 1000 // excitation voltage per phase in V
16 Z_s = 12 // synchronous impedance in ohm
17 R_a = 1.5 //armature resistance in ohm
18
19 // caclulations
20 P_dm = (((E_f*V_t)/Z_s)-((E_f^2*R_a)/Z_s^2)) //
    maximum power per phase in W
21 T_dm = 9.55*P_dm/1500 // maximum torque in N-m
22
23 // display the result
24 disp("Example 10.4 solution");
25 printf(" \n Maximum power developed \n P_dm = %.0f W
    \n", P_dm );
26 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

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 11 : SINGLE-PHASE MOTORS
7 // Example : 11.1
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V = 220 // supply voltage in V
13 P = 4 // number of poles
14 f = 50 // frequency in Hz
15 N_1 = 1450 // speed in rpm
16 P_tloss = 2000 // total power loss in W
17 R_2 = 10 // rotor resistance at standstill
    in ohm
18
19 // caclulations
20 N_s = 120*f/P // synchronous speed in rpm
```

```

21 s_f = (N_s-N_1)/N_s // slip due to forward field
22 s_b = 2-s_f // slip due to backward field
23 R_f = R_2/s_f // effective rotor resistance
    due to forward slip in ohm
24 R_b = R_2/(2-s_f) // effective rotor resistance
    due to backward slip in ohm
25
26 // display the result
27 disp("Example 11.1 solution");
28 printf(" \n Slip due to forward field \n s_f = %.2f
    \n", s_f );
29 printf(" \n Slip due to backward field \n s_b = %.2f
    \n", s_b );
30 printf(" \n Effective rotor resistance due to
    forward slip \n R_f = %.2f ohm \n", R_f );
31 printf(" \n Effective rotor resistance due to
    backward slip \n R_b = %.2f ohm \n", R_b );
32
33 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 dev

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 11 : SINGLE-PHASE MOTORS
7 // Example : 11.2
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V_t = 220 // supply voltage in V

```

```

13 R_1 = 6 // equivalent parameters of single phase
    induction motor in ohm
14 R_2 = 6 // equivalent parameters of single phase
    induction motor in ohm
15 X_1 = 10 // equivalent parameters of single phase
    induction motor in ohm
16 X_2 = 10 // equivalent parameters of single phase
    induction motor in ohm
17 N = 1500 // speed in rpm
18 s = 0.03 // slip
19 X_m = 150 // equivalent parameters of single phase
    induction motor in ohm
20
21 // caclulations
22 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
23 R_f = real(Z_f) // from Z_f
24 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
25 R_b = real(Z_b) // from Z_b
26 Z_t = R_1+i*X_1+Z_f+Z_b // total impedance in ohm
27 I_1 = V_t/Z_t // input current in A
28 P_d = (abs(I_1))^2*(R_f-R_b)*(1-s) // power
    developed in W
29 T_d = 9.55*P_d/N // torque in N-m
30
31 // display the result
32 disp("Example 11.2 solution");
33 printf(" \n input current \n I_1 = %.2f<%.2f A \n",
    abs(I_1),atand(imag(I_1),real(I_1)));
34 printf(" \n power developed \n P_d = %.2f \n", P_d )
    ;
35 printf(" \n torque \n T_d = %.2f \n", T_d );
36 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

```
1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 11 : SINGLE-PHASE MOTORS
7 // Example : 11.3
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V_t = 220 // supply voltage in V
13 f = 50 // frequency in Hz
14 Z_m = 3+%i*5 // main winding impedance of motor in
    ohm
15 Z_s = 5+%i*3 // starting impedance of motor in ohm
16
17 // caclulations
18 alpha_s = atand(imag(Z_s),real(Z_s))
    // angle in degree
    from Z_s
19 I_s = V_t/Z_s // starting
    current in A
20 alpha_m = atand(imag(Z_m),real(Z_m))
    // angle in degree
    from Z_m
21 I_m = V_t/(Z_m) // main
    winding current in A
22 alpha = alpha_m-alpha_s // angle
    of line current
23 I = sqrt((abs(I_s))^2+(abs(I_m))^2+2*abs(I_s)*abs(
    I_m)*cosd(alpha)) // line current in A
```

```

24
25 // display the result
26 disp("Example 11.3 solution");
27 printf(" \n Input current \n I_s = %.2f<%.2f A \n",
        abs(I_s), atand(imag(I_s), real(I_s)) );
28 printf(" \n Main winding current \n I_m = %.2f<%.f A
        \n", abs(I_m), atand(imag(I_m), real(I_m)));
29 printf(" \n Line current \n I = %.2f A \n", I );

```

---

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

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 11 : SINGLE-PHASE MOTORS
7 // Example : 11.4
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 V_t = 220 // supply voltage in V
13 f = 50 // frequency in Hz
14 Z_m = 4+%i*3.5 // main winding impedance of motor in
    ohm
15 Z_s = 5+%i*3 // starting impedance of motor in ohm
16 R_s = 5 // from Z_s
17 X_s = 3 // from Z_s
18
19 // caclulations
20 alpha_m = 41.2 // angle in degree from Z_m
21 // Let X_c be connected in series with the starting
    winding. Then the total impedance of starting
    winding is  $Z_s = Z_m - \%i * X_c$ 

```

```

22 // 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
23 alpha_s = alpha_m-90
24 X_c = X_s-R_s*tand(alpha_s)
25 C = 1/(2*pi*f*X_c) // starting capacitance to get
    maximum torque in F
26
27 // display the result
28 disp("Example 11.4 solution");
29 printf("\n Starting capacitance for getting maximum
    torque \n C = %.2e F \n", C );

```

---

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

```

1 // FUNDAMENTALS OF ELECTRICAL MACHINES
2 // M.A.SALAM
3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 11 : SINGLE-PHASE MOTORS
7 // Example : 11.5
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 f = 50 // supply frequency in Hz
13 V_n1 = 100 // no-load voltage in v
14 I_n1 = 2.5 // no-load current in A
15 P_n1 = 60 // no-load power in W
16
17 V_br = 60 // Block rotor test voltage in v
18 I_br = 3 // Block rotor test current in A
19 P_br = 130 // Block rotor test power in W

```



```

20 R_1 = 2          // main windning resistance in ohm
21
22 // caclulations
23 Z_br = V_br/I_br          // impedance due
    to blocked rotor test
24 R_br = P_br/I_br^2        // resistance
    due to blocked rotor test in ohm
25 X_br = sqrt(Z_br^2-R_br^2) // reactance
    under blocked condition in ohm
26 X_1 = X_br/2             // reactance in
    ohm X_1=X_2
27 R_2 = R_br-R_1           // resistance in
    ohm
28 Z_n1 = V_n1/I_n1         // impedance due
    to no-load in ohm
29 R_n1 = P_n1/I_n1^2        // resistance
    due to no-load in ohm
30 X_n1 = sqrt(Z_n1^2-R_n1^2) // reactance due
    to no-load in ohm
31 X_m =2*(X_n1-X_1-0.5*X_1) // magnetizing
    reactance in ohm
32 P_rot = P_n1-I_n1^2*(R_1+((R_2)/4)) // rotational
    loss in W
33
34 // display the result
35 disp("Example 11.5 solution");
36 printf(" \n Magnetizing reactance \n X_m = %.1f ohm
    \n", X_m );
37 printf(" \n Rotational loss \n P_rot = %.0f W \n",
    P_rot );

```

---

Scilab code Exa 11.6 determine tooth pitch and step angle

```

1 // FUNDAMENTALS OF ELECTICAL MACHINES
2 // M.A.SALAM

```

```

3 // NAROSA PUBLISHING HOUSE
4 // SECOND EDITION
5
6 // Chapter 11 : SINGLE-PHASE MOTORS
7 // Example : 11.6
8
9 clc;clear; // clears the console and command history
10
11 // Given data
12 r_t = 36 // rotor teeth of stepper motor
13 N = 4 // stator phases
14
15 // caclulations
16 T_p = 360/r_t // tooth pitch
17 teta = 360/(N*r_t) // step angle
18
19 // display the result
20 disp("Example 11.6 solution");
21 printf(" \n Tooth pitch \n T_p = %.0i degree \n",
        T_p );
22 printf(" \n Strp angle \n teta = %.1f degree \n",
        teta );

```

---