Scilab Textbook Companion for Electric Machinery And Transformers by I. L. Kosow[1]

Created by
Thirumalesh H S
Bachelor of Engineering
Electrical Engineering
Sri Jayachamarajendra College of Engineering
College Teacher
R. S. Ananda Murthy
Cross-Checked by
Lavitha Pereira

August 13, 2013

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

Book Description

Title: Electric Machinery And Transformers

Author: I. L. Kosow

Publisher: Prentice Hall Of India, New Delhi

Edition: 2

Year: 1992

ISBN: 0876927754

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes

Chapter 1

ELECTROMECHANICAL FUNDAMENTALS

Scilab code Exa 1.1 calculate average voltage

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 1: Electromechanical Fundamentals
// Example 1-1

clear; clc; close; // Clear the work space and
   console.

// Given data
t = 50e-3; // t = time in milli second
phi = 8 * 10 ^ 6; // phi = uniform magnetic field in
    maxwells

// Calculations
E_av = (phi / t) * 10 ^ -8; // E_av = average
   voltage generated in the conductor
```

```
// in volt

// Display the result
disp("Example 1-1 Solution : ");
disp("Average voltage generated in the conductor is : ");
printf(" E_av = %.2f V", E_av);
```

Scilab code Exa 1.2 calculate e and E

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 1: Electromechanical Fundamentals
// Example 1-2

clear; clc; close; // Clear the work space and console.

// Given data
l = 18; // l = length of the conductor in inches
B = 50000; // B = uniform magnetic field in lines/sq-inches
d = 720; // d = distance travelled by conductor in inches
t = 1; // t =time taken for the conductor to  move in second

// Calculations
v = d/t; // v = velocity in inches/second with which the conductor moves

// part a
```

```
e = B * l * v * 10 ^ -8; // e = instantaneous
    induced EMF in volt
// part b
A = d * l; // Area swept by the conductor while
    moving
phi = B * A; // phi = uniform magnetic field
E = ( phi / t ) * 10 ^ -8; // E = average induced
    EMF

// Display the result
disp("Example 1-2 Solution : ");

printf(" \n a : e = %.2f V ", e);
printf(" \n b : E = %.2f V ", E);
```

Scilab code Exa 1.3 calculate E

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 1: Electromechanical Fundamentals
// Example 1-3

clear; clc; close; // Clear the work space and
    console.

// Given data
l = 18; // l = length of the conductor in inches
B = 50000; // B = uniform magnetic field in lines/sq
    -inches
d = 720; // d = distance travelled by conductor in
    inches
t = 1; // t =time taken for the conductor to  move
```

```
    in second
theta = 75 // theta = angle between the motion of
    the conductor and field
// in radians

// Calculations
v = d/t; // v = velocity in inches/second with which
     the conductor moves

E = B * l * v * 10 ^ -8 * sind(theta); // E =
    Average induced EMF in volt

// Display the result
disp("Example 1-3 Solution : ");

disp(" Average induced EMF in volt is :")
printf("  E = %.2f V ", E);
```

Scilab code Exa 1.4 calculate E for different theta

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 1: Electromechanical Fundamentals
// Example 1-4

clear; clc; close; // Clear the work space and
    console.

// Given data
v = 1.5; // v = velocity in m/s with which the
    conductor is moving
l = 0.4; // l = length of the conductor
```

```
B = 1; // B = uniform field intensity in tesla
theta_a = 90; // theta_a = angle between the motion
    of the conductor and field
theta_b = 35; // theta_b = angle between the motion
    of the conductor and field
theta_c = 120; // theta_c = angle between the motion
     of the conductor and field

// Calculations
E_a = B * l * v * sind(theta_a); // Voltage induced
    in the conductor for theta_a
E_b = B * l * v * sind(theta_b); // Voltage induced
    in the conductor for theta_b
E_c = B * l * v * sind(theta_c); // Voltage induced
    in the conductor for theta_c

// Display the result
disp("Example 1-1 Solution : ");

printf("\n a: E = %.2f V ", E_a);
printf("\n b: E = %.3f V ", E_b);
printf("\n c: E = %.2f V ", E_c);
```

Scilab code Exa 1.5 calculate Eperpath Eg Ia Ra Vt P

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 1: Electromechanical Fundamentals
// Example 1-5

clear; clc; close; // Clear the work space and
    console.
```

```

// Given data
no_of_conductors = 40;
A = 2; // A = Parallel paths
path = A;
flux_per_pole = 6.48 * 10 ^ 8; // flux lines
S = 30; // S = Speed of the prime mover in rpm
R_per_path = 0.01; // Resistance per path
I = 10; // Current carried by each condutcor
P = 2; // No. of poles

// Calculations
total_flux = P * flux_per_pole; // Total flux linked in one revolution
t = ( 1 / 30 ) * ( 60 ); // time for one revolution

e_av_per_conductor = ( total_flux / t ) * 10^-8; // Average voltage generated
// per conductor
E_path = ( e_av_per_conductor ) * ( no_of_conductors / path ); // Average
//voltage  generated per path

E_g = E_path; // Generated armature voltage

I_a =( I / path ) * ( 2 * path ); // Armature current delivered to an external
// load

R_a = ( R_per_path) / path * 20; // Armature resistance

V_t = E_g - I_a * R_a; // Terminal voltage of generator

P = V_t * I_a; // Genrator power rating

// Display the results
```

```
disp("Example 1−5 Solution");

printf(" \n a : E/path = %.2f V/path ", E_path );
printf(" \n b : Eg = %.2f V ", E_g );
printf(" \n c : Ia = %.2f A ", I_a );
printf(" \n d : Ra = %.2f ohm ", R_a );
printf(" \n e : Vt = %.2f V ", V_t );
printf(" \n f : P = %.2f W ", P );
```

Scilab code Exa 1.6 repeated previous eg with 4poles

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 1: Electromechanical Fundamentals
// Example 1−6

clear; clc; close; // Clear the work space and
    console.

// Given data
no_of_conductors = 40;
I = 10; // Current carried by each condutcor
R_per_path = 0.01; // Resistance per path
flux_per_pole = 6.48 * 10 ^ 8; // flux lines
P = 2; // No. of poles
path = 4; // No. of parallel paths
total_flux = P * flux_per_pole; // Total flux linked
     in one revolution
t = 2; // time for one revolution
e_av_per_conductor = 6.48; // Average voltage
    generated per conductor

```

```
// Calculations
E_path = ( e_av_per_conductor ) * ( no_of_conductors / path ); // Average
//voltage generated per path

E_g = E_path; // Generated armature voltage

I_a =( I / path ) * ( 4 * path ); // Armature current delivered to an external
// load

R_a = ( ( R_per_path) / path ) * 10; // Armature resistance

V_t = E_g - I_a * R_a; // Terminal voltage of generator

P = V_t * I_a; // Genrator power rating

// Display the results
disp("Example 1-6 Solution");

printf(" \n a : E/path = %.2f V/path ", E_path );
printf(" \n b : Eg = %.2f V ", E_g );
printf(" \n c : Ia = %.2f A ", I_a );
printf(" \n d : Ra = %.3f ohm ", R_a );
printf(" \n e : Vt = %.2f V ", V_t );
printf(" \n f : P = %.2f W ", P );
```

Scilab code Exa 1.7 calculate Eav per coil and per coilside

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom
```

```
// Chapter 1: Electromechanical Fundamentals
// Example 1-7

clear; clc; close; // Clear the work space and console.

// Given data
N = 1; // no. of turns
phi = 6.48 * 10 ^ 8; // Magnetic flux in lines
s = 30 / 60; // No. of revolution of the coil per second( refer section 1-14)

// Calculations
E_av_per_coil = 4 * phi * N * s * 10 ^ -8; // average voltage per coil
// for above equation refer section 1-14

E_av_per_coil_side = E_av_per_coil * ( 1 / 2); // average voltage per conductor

// Display the results
disp("Example 1-7 Solution : ")
printf(" \n Eav/coil = % .2f V/coil ", E_av_per_coil);
printf(" \n Eav/coil side = % .2f V/conductor ", E_av_per_coil_side);
```

Scilab code Exa 1.8 verify previous eg with phi in webers

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

```

```
// Chapter 1: Electromechanical Fundamentals
// Example 1-8

clear; clc; close; // Clear the work space and
    console.

// Given data
phi_lines = 6.48 * 10 ^ 8; // magnetic flux in lines
N = 1; // no. of turns

// Calculations
phi = phi_lines * 10 ^ -8; // Magnetic flux in weber

omega = ( 30 ) * ( 2 * %pi ) * ( 1 / 60 ); //
    angular velocity in rad/s

E_av_per_coil = 0.63662 * omega * phi * N; //
    average voltage per coil
// for the above formula refer section 1-14 eqn (1-4
    b)

// Display the result
disp("Example 1-8 Solution : ");
printf("\n Eav/coil  = % 0.2f V/coil ",
    E_av_per_coil);
```

Scilab code Exa 1.9 verify eg1 5b with eq1 5a

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 1: Electromechanical Fundamentals
// Example 1-9
```

```

clear; clc; close; // Clear the work space and
    console.

// Given data
P = 2; // No. of poles
Z = 40; // no of conductors
a = 2; // a = Parallel paths
phi = 6.48 * 10 ^ 8; // magnetic flux
S = 30; // Speed of the prime mover

// Calculations
E_g = ( ( phi * Z * S * P ) / ( 60 * a) ) * 10 ^ -8;
     // average voltage between
// the brushes

// Display the result
disp("Example 1-9 Solution : ");
printf("\n Eg = %.2f V between the brushes ", E_g);
```

Scilab code Exa 1.10 calculate Z and Eg

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 1: Electromechanical Fundamentals
// Example 1-10

clear; clc; close; // Clear the work space and
    console.

// Given data
no_of_coils = 40;
```

```
N = 20; // no of turns in each coil
omega = 200; // angular velocity of armature in rad/s
phi = 5 * 10 ^ -3; // flux per pole
a = 4; // No. of parallel paths
P = 4; // No. of poles

// Calculations
Z = no_of_coils * 2 * N; // No. of conductors

E_g = ( phi * Z *  omega * P ) / ( 2 * %pi * a ); // Voltage generated by the
// armature between brushes

// Display the results
disp("Example 1-10 Solution : ");
printf("\n Z = %d conductors ", Z);
printf("\n Eg = %.2f V between the brushes ", E_g);
```

Scilab code Exa 1.11 calculate F and find its direction

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 1: Electromechanical Fundamentals
// Example 1-11

clear; clc; close; // Clear the work space and console.

// Given data
l = 0.5; // length of the conductor
A = 0.1 * 0.2; // area of the pole face
```

```
phi = 0.5 * 10 ^ -3; // magnetic flux in weber
I = 10; //Current in the conductor

// Calculations
B = ( phi ) / ( A ); // Flux density

F = B * I * l; // Magnitude of force

// Display the result
disp("Example 1-11 Solution : ");

printf("\n a : F = %.3f N", F );

printf("\n b : The force on the conductor is %.3f N in an upward direction as shown in fig 1-13c ", F );
```

Scilab code Exa 1.12 repeat previous eg with angle 75

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 1: Electromechanical Fundamentals
// Example 1-12

clear; clc; close; // Clear the work space and console.

// Given data
l = 0.5; // length of the conductor
A = 0.1 * 0.2; // area of the pole face
phi = 0.5 * 10 ^ -3; // magnetic flux in weber
I = 10; //Current in the conductor
```

```
theta = 75; // angle between the conductor and the
    flux density B

// Calculations
B = ( phi ) / ( A ); // Flux density

F = B * I * l * sind(theta); // Magnitude of force

// Display the result
disp("Example 1-12 Solution : ");

printf("\n F =% f N in a vertically upward direction
    ", F );
```

Scilab code Exa 1.13 calculate counter emf

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 1: Electromechanical Fundamentals
// Example 1-13

clear; clc; close; // Clear the work space and
    console.

// Given data
R_a = 0.25; // Armature resistance
V_a = 125; // dc bus voltage
I_a = 60; // Armature current

// Calculations
E_c = V_a - I_a * R_a; // Counter EMF generated in
    the armature conductors of motor
```

```
// Display the result
disp("Example 1-13 Solution : ");
printf("\n Ec = % d V ", E_c );
```

Scilab code Exa 1.14 calculate Eg phi in linesperpole and mWb

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 1: Electromechanical Fundamentals
// Example 1-13

clear; clc; close; // Clear the work space and
   console.

// Given data
V_a = 110; // voltage across armature
I_a = 60; // Armature current
R_a = 0.25; // Armature resistance
P = 6; // No. of poles
a = 12; // No. of paths
Z = 720; // No. of armature conductors
S = 1800; // Speed in rpm

// Calculations
E_g = V_a + I_a * R_a; // Generated EMF in the
   armature

phi_lines = ( E_g * ( 60 * a ) ) / ( ( Z * S * P ) *
    10 ^ -8 );
// Flux per pole in lines

```

```
phi_Wb = phi_lines * 10 ^ -8; // Flux per pole in
   webers

// Display the results
disp("Example 1-14 Solution : ");

printf("\n a : Eg = %d V ", E_g );

printf("\n b : phi = %f lines/pole ", phi_lines );

printf("\n c : phi = %f Wb ", phi_Wb );
```

Chapter 2

DYNAMO CONSTRUCTION AND WINDINGS

Scilab code Exa 2.1 calculate a for lap and wave windings

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 2: Dynamo Construction and Windings
// Example 2-1

clear; clc; close; // Clear the work space and console.

// Given data
m = 3; // Multipicity of the armature
P = 14; // No. of poles

// Calculations
a_lap = m * P; // No. of parallel paths in the armature for a lap winding
a_wave = 2 * m; // No. of parallel paths in the
```

```
    armature for a wave winding

// Display the result
disp("Example 2-1 Solution : ");

printf("\n a: a = %d paths ", a_lap);
printf("\n b: a = %d paths ", a_wave);
```

Scilab code Exa 2.2 calculate generated emf

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 2: Dynamo Construction and Windings
// Example 2-2

clear; clc; close; // Clear the work space and
    console.

// Given data
P = 14; // No. of poles
phi = 4.2e6; // Flux per pole
S = 60; // Generator speed
coils = 420; // No. of coils
turns_per_coil = 20;
conductors_per_turn = 2;
a_lap = 42; // No. of parallel paths in the armature
     for a lap winding
a_wave = 6; // No. of parallel paths in the armature
     for a wave winding

// Calculations
Z = coils * turns_per_coil * conductors_per_turn; //
```

```
No. of conductors
E_g_lap = (( phi * Z * S * P ) / ( 60 * a_lap )) * 10 ^ -8; // Generated EMF for
// lap winding ( Eq 1-5a)
E_g_wave = ( phi * Z * S * P ) / ( 60 * a_wave ) * 10 ^ -8; // Generated EMF for
// wave winding ( Eq 1-5a)

// Display the result
disp("Example 2-2 Solution : ");

printf("\n a: Eg = %0.1f V ", E_g_lap);
printf("\n b: Eg = %0.1f V ", E_g_wave);
```

Scilab code Exa 2.3 calculate polespan p kp

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 2: Dynamo Construction and Windings
// Example 2-3

clear; clc; close; // Clear the work space and console.

// Given data
slots = 72; // No. of slots
P = 4; // No. of poles
coils_spanned = 14; // 14 slots are spanned while winding the coils

// Calculations
Pole_span = slots / P; // Pole span
```

```
p_not = coils_spanned / Pole_span * 180; // Span of the coil in
// electrical degrees
funcprot(0) ; // Use  to avoid this message "Warning : redefining function: beta "
beta = (180 - p_not);
k_p1 = cosd( beta / 2 ); // Pitch factor using eq (2-7)
k_p2 = sind( p_not / 2 ); // Pitch factor using eq (2-8)

// Display the results
disp("Example 2-3 Solution : ")
printf(" \n a: Full-pitch coil span = %d slots/pole ", Pole_span );
printf(" \n b: p = %d degrees ", p_not );
printf(" \n c: kp = %.2f \t\t eq(2-7)", k_p1 );
printf(" \n d: kp = %.2f \t\t eq(2-8)", k_p2 );
```

Scilab code Exa 2.4 calculate kp

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 2: Dynamo Construction and Windings
// Example 2-4

clear; clc; close; // Clear the work space and console.

// Given data
fractional_pitch = 13 / 16;
slot =96; // No. of slots
```

```
P = 6; // No. of poles

// Calculation
k_p = sind( ( fractional_pitch * 180 ) / 2 ); //
    Pitch factor

// Display the result
disp("Example 2-4 Solution : ")
printf("\n kp = %.4f ", k_p );
```

Scilab code Exa 2.5 find alpha n theta

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 2: Dynamo Construction and Windings
// Example 2-5

clear; clc; close; // Clear the work space and
    console.

// Given data
P = 12; // No. of poles
theta = 360; // No. of mechanical degrees of
    rotation
alpha_b = 180; // No. of electrical degrees for
    finding case b in the question

// Calculations
alpha = ( P * theta ) / 2; // No. of electrical
    degrees in one revolution
n = alpha / 360; // No. of ac cycles
theta_b = ( 2 * alpha_b ) / P; // No. of mechanical
```

```
    degrees of rotation
// for finding case b in the question

// Display the results
disp("Example 2-5 Solution : ")
printf("\n a: alpha = %d degrees", alpha);
printf("\n    n = %d cycles ", n);
printf("\n b: theta = %d mechanical degrees ",
    theta_b );
```

Scilab code Exa 2.6 find n alpha kd for different number of slots

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 2: Dynamo Construction and Windings
// Example 2-6

clear; clc; close; // Clear the work space and
    console.

// Given data
P = 4;// No. of poles
phi = 3; // No. of phases
slots_(1) = 12; // No. of slots for case 1
slots_(2) = 24; // No. of slots for case 2
slots_(3) = 48; // No. of slots for case 3
slots_(4) = 84; // No. of slots for case 4

// Calculations
electrical_degrees = 180 * 4;
i=1; // where i is case subscript .eg case1 , case2 ,
    etc
```

```

  while i<=4
    alpha_(i) = electrical_degrees / slots_(i); // electrical degrees
    // per slots for case i
    n_(i) = slots_(i) / ( P * phi ); // No. of ac cycles for case 1
    k_d(i) = sind( n_(i)*( alpha_(i) / 2 ) ) / ( n_(i) * sind( alpha_(i) / 2));
    i=i+1;
  end;

// Display the results
disp("Example 2-6 Solution : ")
printf("\n a:");
 i=1; // where i is case subscript .eg case1, case2, etc

   while i<=4
        printf("\n \t %d: alpha = %.2f degrees/slot", i , alpha_(i) );
        printf("\n\t     n = %d slots/pole-phase ", n_(i) );
        printf("\n\t     kd = %.3f ", k_d(i));
        printf("\n");
        i=i+1;
   end;

printf("\n\n\n b: ");
printf("\n \t \t n \t alpha in degrees  \t\t kd ");
printf("\n \t -----------------------------------------------------------" );
i=1;

 while i<=4
    printf("\n \t \t %d \t %.2f \t\t\t\t%.3f ", n_(i) , alpha_(i) , k_d(i) );
```

```
    i = i +1;
  end;
    printf("\n \t ---------------------------------------------------------- " );
```

Scilab code Exa 2.7 calculate Eg Np kd kp Egp

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 2: Dynamo Construction and Windings
// Example 2-7

clear; clc; close; // Clear the work space and
   console.

// Given data
slots = 72; // No. of slots
P = 6; // No. of poles
phase =3; // three phase stator armature
N_c = 20; // Number of turns per coil
pitch = 5 / 6;
phi = 4.8e+6; // flux per pole in lines
S = 1200; // Rotor speed

// Calculations
f = ( P * S )/ 120; // Frequency of rotor

E_g_percoil = 4.44 * phi * N_c * f *10 ^ -8; //
   Generated effective voltage
// per coil of a full pitch coil

```

```
N_p = ( slots / phase ) * N_c; // Total number of turns per phase

n = slots / ( phase * P ); // No. os slots per pole per phase

alpha = ( P * 180 ) / slots; // No. of electrical degrees between adjacent slots

k_d = sind( n * alpha / 2 ) / ( n * sind( alpha / 2 ) ); // Distribution factor

span = pitch * 180; // Span of the coil in electrical degrees

k_p = sind( span / 2 ); // Pitch factor

E_gp = 4.44 * phi * N_p * f * k_p * k_d * 10 ^ -8; // Total generated voltage
// per phase considering kp and kd

// Display the result
disp("Example 2-7 Solution : ")
printf("\n a: Eg/coil = %.2f V/coil", E_g_percoil );
printf("\n b: Np = %d turns/phase ", N_p );
printf("\n c: kd = %.3f ", k_d );
printf("\n d: kp = %.3f ", k_p );
printf("\n e: Egp = %.2f V/phase ", E_gp );
```

Scilab code Exa 2.8 calculate f S omega

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom
```

```
// Chapter 2: Dynamo Construction and Windings
// Example 2-8

clear; clc; close; // Clear the work space and console.

// Given data
P = 8; // No. of poles
S = 900; // Speed in revolutions / minute
f_1 = 50; // Frequency of generated voltage from generator 1
f_2 = 25; // Frequency of generated voltage from generator 2

// Calculations
f = ( P * S ) / 120; // Frequency of the generated voltage
S_1 = ( 120 * f_1 ) / P; // Speed of generator(rpm) 1 to generate 50 Hz voltage
S_2 = ( 120 * f_2 ) / P; // Speed of generator(rpm) 2 to generate 25 Hz voltage
omega_1 = ( 4 * %pi * f_1 ) / P; // Speed of generator 1 in rad/s
omega_2 = ( 4 * %pi * f_2 ) / P; // Speed of generator 2 in rad/s

// Display the result
disp("Example 2-8 Solution : ")
printf("\n a: f = %d Hz ", f );
printf("\n b: S1 = %d rpm \n    S2 = %d rpm ", S_1, S_2 );
printf("\n c: omega1 = %f rad/s \n    omega2 = %f rad/s", omega_1, omega_2 );
```

Chapter 3

DC DYNAMO VOLTAGE RELATIONS DC GENERATORS

Scilab code Exa 3.1 calculate I1 If Ia Eg

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 3: DC Dynamo Voltage Relations - DC
    Generators
// Example 3-1

clear; clc; close; // Clear the work space and
    console.

// Given data
kW = 150; // Power rating of Shunt generator in kW
V_1 = 250; // Voltage rating of Shunt generator in V
V_a = V_1; // Voltage rating of Shunt generator in V
R_f = 50; // Field resistance in ohm
```

```
R_a = 0.05; // Armature resistance in ohm

// Calculations
I_1 = ( kW * 1000 ) / V_1; // Full-load line current
    flowing to the load in A
I_f = V_1 / R_f; // Field current in A
I_a = I_f + I_1; // Armature current in A
E_g = V_a + I_a * R_a; // Full load generated
    voltage in V

// Display the results
disp("Example 3-1 Solution : ")
printf("\n a: I1 = %d A ", I_1 );
printf("\n b: If = %d A ", I_f );
printf("\n c: Ia = %d A ", I_a );
printf("\n d: Eg = %.2f A ", E_g );
```

Scilab code Exa 3.2 calculate Rd Eg

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 3: DC Dynamo Voltage Relations - DC
    Generators
// Example 3-2

clear; clc; close; // Clear the work space and
    console.

// Given data
kW =100; // Power rating of the generator in kW
V_1 = 500; // Voltage rating of hte generator in V
R_a = 0.03; // Armature resistance in ohm
```

```
R_f = 125; // Shunt field resistance in ohm
R_s = 0.01; // Series field resistance in ohm
I_d = 54; // Diverter current in A

// Calculations
I_1 = ( kW * 1000 ) / V_1; // Full-load line current
    flowing to the load in A
I_f = V_1 / R_f; // Shunt Field current in A
I_a = I_f + I_1; // Armature current in A
I_s = I_a - I_d; // Series Field current in A
R_d = ( I_s * R_s ) / I_d; // Diverter resistance in
    ohm
E_g = V_1 + I_a * R_a + I_s * R_s; // Generated
    voltage at full load in V

// Display the results
disp("Example 3-2 Solution : ")
printf("\n a: Rd = %.4f ohm ", R_d );
printf("\n b: Eg = %.2f V ", E_g );
```

Scilab code Exa 3.3 calculate Vnoload

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 3: DC Dynamo Voltage Relations - DC
    Generators
// Example 3-3

clear; clc; close; // Clear the work space and
    console.

// Given data
```

```
E_orig = 150; // Armature voltage of the generator
    in V
S_orig = 1800; // Speed of the generator in rpm
S_final_a =2000; // Increased Speed of the generator
     in rpm for case a
S_final_b =1600; // Increased Speed of the generator
     in rpm for case b

// Calculations
E_final_a = E_orig * ( S_final_a / S_orig ); // No-
    load voltage of the generator
// generator  in V for case a
E_final_b = E_orig * ( S_final_b / S_orig ); // No-
    load voltage of the generator
// generator  in V for case b

// Display the results
disp("Example 3-3 Solution : ")
printf("\n a: Efinal = %.1f V ", E_final_a );
printf("\n b: Efinal = %.1f V ", E_final_b );
```

Scilab code Exa 3.4 calculate E

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 3: DC Dynamo Voltage Relations - DC
    Generators
// Example 3-4

clear; clc; close; // Clear the work space and
    console.

```

```
// Given data
S_final = 1200; // Speed of th generator in rpm
E_orig_a = 64.3; // Armature voltage of the generator in V for case a
E_orig_b = 82.9; // Armature voltage of the generator in V for case b
E_orig_c = 162.3; // Armature voltage of the generator in V for case c

S_orig_a = 1205; // Varied Speed of the generator in rpm for case a
S_orig_b = 1194; // Varied Speed of the generator in rpm for case b
S_orig_c = 1202; // Varied Speed of the generator in rpm for case c

// Calculations
E_1 = E_orig_a * ( S_final / S_orig_a ); // No- load voltage of the generator
// generator in V for case a
E_2 = E_orig_b * ( S_final / S_orig_b ); // No- load voltage of the generator
// generator in V for case b
E_3 = E_orig_c * ( S_final / S_orig_c ); // No- load voltage of the generator
// generator in V for case c

// Display the results
disp("Example 3-4 Solution : ")
printf("\n a: E1 = %.1f V at %d rpm ", E_1, S_final);
printf("\n b: E2 = %.1f V at %d rpm ", E_2, S_final);
printf("\n c: E3 = %.1f V at %d rpm ", E_3, S_final);
```

Scilab code Exa 3.5 calculate Ia Eg

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 3: DC Dynamo Voltage Relations − DC Generators
// Example 3−5

clear; clc; close; // Clear the work space and console.

// Given data
V = 125; // Rated voltage of the shunt generator in V
R_a = 0.15; // Armature resistance in ohm
V_a = 0; // Shunt generator is loaded progressively until the terminal voltage
// across the load is zero volt
I_1 = 96; // Load current in A
I_f = 4; // Field current in A

// Calculations
I_a = I_f + I_1; // Armature current in A
E_g = V_a + I_a * R_a ; // Voltage generated in the armature in V

// Display the results
disp("Example 3−5 Solution : ")
printf("\n Ia = %d A ", I_a );
printf("\n Eg = %d V ", E_g );
```

Scilab code Exa 3.6 calculate VR

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 3: DC Dynamo Voltage Relations − DC
    Generators
// Example 3−6

clear; clc; close; // Clear the work space and
    console.

// Given data
V_n1 = 135; // No load voltage of the shunt
    generator in V
V_f1 = 125; // Full load voltage of the shunt
    generator in V

// Calculation
VR = ( V_n1 - V_f1 ) / V_f1 * 100; // Percentage
    voltage regulation

// Display the result
disp("Example 3−6 Solution : ")
printf(" \n VR = %d percent ", VR );
```

Scilab code Exa 3.7 calculate Vnoload

```
// Electric Machinery and Transformers
// Irving L kosow
```

```
// Prentice Hall of India
// 2nd editiom

// Chapter 3: DC Dynamo Voltage Relations − DC
    Generators
// Example 3−7

clear; clc; close; // Clear the work space and
    console.

// Given data
VR = 0.105; // voltage regulation
V_fl = 250; // Full load voltage of the shunt
    generator in V

// Calculation
V_nl = V_fl + ( V_fl * VR ); // No−load voltage of
    the generator in V

// Display the result
disp("Example 3−7 Solution : ")
printf("\n Vnl = %.1f V ", V_nl );
```

Scilab code Exa 3.8 calculate IsNs Rd

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 3: DC Dynamo Voltage Relations − DC
    Generators
// Example 3−8

clear; clc; close; // Clear the work space and
```

```
    console.

// Given data
N_f = 1000; // Shunt field winding turns
N_s = 4; // Series field winding turns
I_f = 0.2; // Field current in A
I_a = 80; // Full load armature current in A
R_s =0.05; // Series field resistance in ohm

// Calculations
deba_I_f_N_f = I_f * N_f;
I_s_N_s = deba_I_f_N_f; // Series field ampere-turns
I_s =( I_s_N_s ) / N_s; // Desired current in A in
    the series field required to
// produce voltage rise
I_d = I_a - I_s; // Diverter current in A
R_d = ( I_s * R_s ) / I_d; // Diverter resistance in
    ohm

// Display the result
disp("Example 3-8 Solution : ")
printf("\n a: IsNs = %d At ", I_s_N_s );
printf("\n b: Rd = %.4f ohm ", R_d );
```

Scilab code Exa 3.9 calculate Rd Vnl Vfl

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 3: DC Dynamo Voltage Relations - DC
    Generators
// Example 3-9

```

```
clear; clc; close; // Clear the work space and console.

// Given data
kW = 60; // Power rating of the generator in kW
V = 240; //Voltage rating of the generator in V
I_f = 3; // Increase in the field current in A
OC_V = 275; // Over Compounded Voltage in V
I_l = 250; // Rated load current in A
N_f = 200; // No. of turns per pole in the shunt field winding
N_s = 5; // No. of turns per pole in the series field winding
R_f = 240; //  Shunt field resistance in ohm
R_s = 0.005; //  Series field resistance in ohm

// Calculations
deba_I_f_N_f = I_f * N_f;
I_s_N_s = deba_I_f_N_f; // Series field ampere-turns
I_s =( I_s_N_s ) / N_s; // Desired current in A in the series field required to
// produce voltage rise
I_d = I_l - I_s; // Diverter current in A
R_d = ( I_s * R_s ) / I_d; // Diverter resistance in ohm
NL_MMF = ( V / R_f )* N_f; // No-load MMF
I_f_N_f = NL_MMF;
FL_MMF = I_f_N_f + I_s_N_s; // Full-load MMF

// Display the result
disp("Example 3-9 Solution : ")
printf("\n a: Rd = %.5f ohm ", R_d );
printf("\n b: No-load MMF = %d At/pole ", NL_MMF );
printf("\n    Full-load MMF = %d At/pole ", FL_MMF );
```

Scilab code Exa 3.10 determine approx size of dynamo

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 3: DC Dynamo Voltage Relations − DC
    Generators
// Example 3−10

clear; clc; close; // Clear the work space and
    console.

// Given data
kW= 50; // Power rating of the dynamo
V = 125; // Rated voltage in V
S = 1800; // Speed of the dynamo in rpm
I_f =20; // Exciting field current
Max_temp_rise = 25; // Maximum Temperature rise in
    degree celsius
I_l = 400; // Load Current in A
// INSULATION CLASS A
// COMPOUND WINDING

// Display the result
disp("Example 3−10 Solution : ")
printf("\n a: Since the speed is reduced in half, we
     must reduce the kW rating in half. Consequently,
     the 25kW, 900 rpm dynamo has the same size. ");
printf("\n\n b: Since we have cut the speed in half
    but maintained the same kW rating, the dynamo has
     twice the size as the original.");
printf("\n\n c: Half the size. ");
```

```
printf("\n\n d: Same size. ");
```

Chapter 4

DC DYNAMO TORQUE RELATIONS DC MOTORS

Scilab code Exa 4.1 calculate force and torque

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-1

clear; clc; close; // Clear the work space and
   console.

// Given data
d = 0.5; // diameter of the coil in m
l = 0.6; // axial length of the coil in m
B = 0.4; // flux density in T
I = 25; // current carried by the coil in A
theta = 60; // angle between the useful force & the
   interpolar ref axis in deg

```

```
// Calculations
F = B * I * l; // force developed on each coil side
    in  N
f = F * sind(theta); // force developed at the
    instant the coil lies at an angle
// of 60 w.r.t the interpolar  ref axis
r = d / 2; // radius of the coil in m
T_c = f * r; // torque developed in N-m
T_c1 = T_c * 0.2248 * 3.281 ; // torque developed in
     lb-ft by first method
T_c2 = T_c * 0.737562 ; // torque developed in lb-ft
     by second method

// Display the results
disp("Example 4-1 Solution : ")
printf("\n a : F = %d N ", F );
printf("\n b : f = %.2f N ", f );
printf("\n c : Tc = %.2f N-m ", T_c );
printf("\n d : 1.3 N-m * 0.2248 lb/N * 3.281 ft/m =
    %.2f lb-ft ", T_c1 );
printf("\n      1.3 N-m * 0.737562 lb.ft/N.m = %.2f
    lb-ft ", T_c2 );
```

Scilab code Exa 4.2 calculate force and torque

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-2

clear; clc; close; // Clear the work space and
    console.
```

```
// Given data
d = 18 ; // diameter of hte coil in inches
l = 24 ; // axial length of the coil in inches
B = 24000 ; // Flux density in lines/sq.inches
I = 26 ; // Current carried by the coil in A
theta = 60 ; // angle between the useful force & the
    interpolar ref axis in deg


// Calculations
F = ( B * I * l * 10 ^ -7 ) / 1.13 ; // force
    developed on each coil side in lb
f = F * sind(theta); // force developed at the
    instant the coil lies at an angle
// of 60 w.r.t the interpolar ref axis
r = d / 2; // radius of the coil in inches
T_c = f * ( r * 1 / 12); // torque developed in lb.
    ft/conductor

// Display the results
disp("Example 4-2 Solution : ")
printf("\n a : F = %.3f lb ", F );
printf("\n b : f = %.2f lb ", f );
printf("\n c : Tc = %.3f lb-ft/conductor ", T_c );
```

Scilab code Exa 4.3 calculate average force and torque

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-3
```

```
clear; clc; close; // Clear the work space and console.

// Given data
Z = 700 ; // no. of conductors
d = 24 ; // diameter of the armature of the dc motor in inches
l = 34 ; // axial length of the coil in inches
B = 50000 ; // Flux density in lines/sq.inches
I = 25 ; // Current carried by the coil in A

// Calculations
F_av = ( B * I * l * 10 ^ -7 ) / 1.13 * ( 700 * 0.7 ) ; // average force
// developed on each coil side in  lb
r = d / 2; // radius of the coil in inches
T_av = F_av * ( r /12 ) ; // armature average torque in lb-ft

// Display the results
disp("Example 4-3 Solution : ")
printf("\n a : Fav = %.2f lb ", F_av );
printf("\n b : Tav = %.2f lb-ft ", T_av );
```

Scilab code Exa 4.4 calculate torque developed

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-4

```

```
clear; clc; close; // Clear the work space and console.

// Given data
slots = 120 ; // No. of armature slots
conductors_per_slot = 6 ;
B = 60000 ; // Flux density in lines/sq.inches
d = 28 ; // diameter of the armature
l = 14 ; // axial length of the coil in inches
A = 4 ; // No. of parallel paths
span = 0.72 ; // Pole arcs span 72% of the armature surface
I = 133.5 ; // Armature current in A

// Calculations
Z_Ta = slots * conductors_per_slot * span ; // No. of armature conductors
F_t = ( B * I * l )/ ( 1.13 * 10 ^ 7  * A ) * Z_Ta ; // Force developed in lb
r = ( d / 2 ) / 12 ; // radius of the armature in feet
T = F_t * r ; // Tital torque developed

// Display the result
disp("Example 4-4 Solution : ")
printf(" \n T = %d lb-ft", T );
```

Scilab code Exa 4.5 calculate armature current

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
```

```
// Example 4-5

clear; clc; close; // Clear the work space and
    console.

// Given data
slots = 120 ; // No. of armature slots
conductors_per_slot = 6 ;
B = 60000 ; // Flux density in lines/sq.inches
d = 28 ; // diameter of the armature
l = 14 ; // axial length of the coil in inches
A = 4 ; // No. of parallel paths
span = 0.72 ; // Pole arcs span 72% of the armature
    surface
T_a = 1500 ; // total armature torque in lb-ft

// Calculation
Z = slots * conductors_per_slot ; // No. of armature
    conductors
r = ( d / 2 ) / 12 ; // radius of the armature in
    feet
I_a = ( T_a * A * 1.13e7 ) / ( B * l * Z * r * span
    ) ; //Armature current in A

// Display the result
disp("Example 4-5 Solution : ")
printf(" \n Ia = %.1f A ", I_a );
```

Scilab code Exa 4.6 calculate torque due to change in field flux

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

```

```
// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-6

clear; clc; close; // Clear the work space and
   console.

// Given data
T_old = 150 ; // Torque developed by a motor in N-m.
disp("Example 4-6")
disp("Given data : ")
printf("\n \t\t\t phi \t I_a \t T ");
printf("\n \t\t\t _________________________");
printf("\n Original condition \t 1 \t 1 \t 150 N-m "
   );
printf("\n New condition \t\t 0.9 \t 1.5 \t ? ");

// Calculation
T_new = T_old * ( 0.9 / 1 ) * ( 1.5 / 1 ) ; // New
   torque produced in N-m

// Display the result
printf("\n\n Solution : ")
printf("\n Using the ratio method, the new torque is
    the product ");
printf("\n of two new ratio changes : ");
printf("\n T = %.1f N-m ", T_new );
```

Scilab code Exa 4.7 calculate Ia and percentage change in Ia and E

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
```

```
// Example 4-7

clear; clc; close; // Clear the work space and
    console.

// Given data
R_a = 0.25 ; // Armature resistance in ohm
BD = 3 ; // Brush contact drop in volt
V = 120 ; // Applied voltage in volt
E_a = 110 ; // EMF in volt at a given load
E_b = 105 ; // EMF in volt due to application of
    extra load

// Calculations
I_a_a = ( V - ( E_a + BD ) ) / R_a ; // Armature
    current for E_a
I_a_b = ( V - ( E_b + BD ) ) / R_a ; // Armature
    current for E_b
del_E = ( ( E_a - E_b ) / E_a ) * 100 ; // % change
    in counter EMF
del_I = ( ( I_a_a - I_a_b ) / I_a_a ) * 100 ; // %
    change in armature current

// Display the result
disp("Example 4-7 Solution : ")
printf("\n a : Ia = %d A " , I_a_a );
printf("\n b : At increased load \n      Ia = %d A "
    , I_a_b );
printf("\n c : del_Ec = %.2f percent \n      del_Ia =
    %.2f percent " , del_E , del_I);
```

Scilab code Exa 4.8 calculate speed at different loads

```
// Electric Machinery and Transformers
// Irving L kosow
```

```
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-8

clear; clc; close; // Clear the work space and console.

// Given data
V_a = 120 ; // Rated terminal voltage of the DC motor in volt
R_a = 0.2 ; // Armature circuit resistance in ohm
R_sh = 60 ; // Shunt field resistance in ohm
I_l = 40 ; // Line current in A @ full load
BD = 3 ; //  Brush voltage drop in volt
S_orig = 1800 ; // Rated full-load speed in rpm

// Calculations
I_f = V_a / R_sh ; // Field current in A
I_a = I_l - I_f ; // Armature current @ full load
E_c_orig = V_a - ( I_a * R_a + BD ) ; // Back EMF @ full load

I_a_a = I_a / 2 ; // Armature current @ half load
E_c_a = V_a - ( I_a_a * R_a + BD ) ; // Back EMF @ half load
S_a = S_orig * ( E_c_a / E_c_orig ) ; // Speed @ full load

I_a_b = I_a * ( 5 / 4 ) ; // Armature current @ 125% overload
E_c_b = V_a - ( I_a_b * R_a + BD ) ; // Back EMF @ 125% overload
S_b = S_orig * ( E_c_b / E_c_orig ) ; // Speed @ 125 % overload

// Display the result
```

```
disp("Example 4-8 Solution : ");

printf(" \n a : At full load ");
printf(" \n      S = %.1f rpm " , S_a );

printf(" \n b : At 125 percenytoverload ");
printf(" \n      S = %.1f rpm " , S_b );
```

Scilab code Exa 4.9 calculate speed with increased line current

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-9

clear; clc; close; // Clear the work space and
   console.

// Given data
I_l_orig = 40; // original Line current in A
I_l_final = 66; // Final Line current in A

phi_orig = 1;
// field flux is increased by 12% so EMF produced
   and terminal
// voltage will also increase by 12%
phi_final = 1.12;

V_a = 120;
R_sh_orig = 60; // Original Field ckt resistance in
   ohm
R_sh_final = 50 ; // Decreased final field ckt
```

```
resistance in ohm

R_a = 0.2; // Armature resistance in ohm
BD = 3; // Brush voltage drop in volt
S_orig = 1800; // Rated full-load speed

// Calculations
I_f_orig = V_a / R_sh_orig ; // Original Field current in A
I_a_orig = I_l_orig - I_f_orig ; // Original Armature current @ full load
E_c_orig = V_a - ( I_a_orig * R_a + BD ) ; // Back EMF @ full load

I_f_final = V_a / R_sh_final ; // Final field current in A
I_a_final = I_l_final - I_f_final ; // Final Armature current in A
E_c_final = V_a - ( I_a_final * R_a + BD ) ; // Final EMF induced
S = S_orig * ( E_c_final / E_c_orig ) * ( phi_orig / phi_final ) ;
// Final speed of the motor

// Display the result
disp("Example 4-9 Solution : ");
printf(" \n S = %.1f rpm ", S );
```

Scilab code Exa 4.10 calculate power developed

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

```

```
// Chapter 4: DC Dynamo Torque Relations–DC Motors
// Example 4–10

clear; clc; close; // Clear the work space and
    console.

// Given data
I_a_1 = 38 ; // Armature current in A @ full–load as
     per example 4–8a
E_c_1 = 109.4 ; // Back EMF in volt @ full–load as
    per example 4–8a
S_1 = 1800 ; // Speed in rpm @ full–load as per
    example 4–8a

I_a_2 = 19 ; // Armature current in A @ half–loadas
    per example 4–8a
E_c_2 = 113.2 ; // Back EMF in volt @ half–load as
    per example 4–8a
S_2 = 1863 ; // Speed in rpm @ half–load as per
    example 4–8a

I_a_3 = 47.5 ; // Armature current in A @ 125%
    overload as per example 4–8b
E_c_3 = 107.5 ; // Back EMF in volt @ 125% overload
    as per example 4–8b
S_3 = 1769 ; // Speed in rpm @ 125% overload as per
    example 4–8b

I_a_4 = 63.6 ; // Armature current in A @ overload
    as per example 4–9
E_c_4 = 104.3 ; // Back EMF in volt @ overload as
    per example 4–9
S_4 = 1532 ; // Speed in rpm @ overload as per
    example 4–9

// Calculations
P_d_1 = E_c_1 * I_a_1 ; // Armature power developed
    @ full–load
```

```

P_d_2 = E_c_2 * I_a_2 ; // Armature power developed @ half-load

P_d_3 = E_c_3 * I_a_3 ; // Armature power developed @ 125% overload

P_d_4 = E_c_4 * I_a_4 ; // Armature power developed @ overload

// Display the results
disp(" Example 4-10 Solution : ");
printf("\n Example \t Ia \t Ec \t Speed \t Pd or (Ec *Ia)");
printf("\n ----------------------------------------------------------------- ");
printf("\n 4-8a \t\t %d \t %.1f \t %d \t %d W at full-load ", I_a_1,E_c_1,S_1,P_d_1);
printf("\n 4-8a \t\t %d \t %.1f \t %d \t %.1f W at half-load ",I_a_2 , E_c_2 , S_2 , P_d_2);
printf("\n 4-8b \t\t %.1f \t %.1f \t %d \t %d W at 125 percent overload ",I_a_3,E_c_3,S_3,P_d_3);
printf("\n 4-9 \t\t %.1f \t %.1f \t %d \t %d W at overload ",I_a_4,E_c_4,S_4,P_d_4);
printf("\n ----------------------------------------------------------------- ");
```

Scilab code Exa 4.11 convert torque readings into Nm and lbft

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom
```

```
// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-11

clear; clc; close; // Clear the work space and console.

// Given data
T_a = 6.5; // Torque in dyne-centimeters
T_b = 10.6; // Torque in gram-centimeters
T_c = 12.2; // Torque in ounce-inches

// Calculations
T_a_Nm = T_a * 1.416e-5 * 7.0612e-3 ; // Torque T_a in N-m
T_b_Nm = T_b * ( 1 / 72.01 ) * 7.0612e-3 ; // Torque T_b in N-m
T_c_Nm = T_c * 7.0612e-3 ; // Torque T_c in N-m

T_a_lbft = T_a * 1.416e-5 * 5.208e-3; // Torque T_a in lb-ft
T_b_lbft = T_b * ( 1 / 72.01 ) * 5.208e-3; // Torque T_b in lb-ft
T_c_lbft = T_c * 5.208e-3; // Torque T_c in lb-ft

// Display the results
disp("Example 4-11 Solution : ");
printf(" \n a : T = %.1e N-m ", T_a_Nm );
printf(" \n     T = %.1e lb-ft \n ", T_a_lbft );

printf(" \n b : T = %.2e N-m ", T_b_Nm );
printf(" \n     T = %.1e lb-ft \n ", T_b_lbft );

printf(" \n c : T = %.3e N-m ", T_c_Nm );
printf(" \n     T = %.2e lb-ft \n ", T_c_lbft );
```

Scilab code Exa 4.12 calculate Ist and percentage of load current

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-12

clear; clc; close; // Clear the work space and
   console.

// Given data
V_a = 120 ; // Rated terminal voltage of dc shunt
   notor in volt
R_a = 0.2 ; // Armature resistance in ohm
BD = 2 ; // Brush drop in volt
I_a = 75 ; // Full load armature current in A

// Calculations
I_st = ( V_a - BD ) / R_a ; // Current @ the instant
    of starting in A
percentage = I_st / I_a * 100 ; // Percentage at
   full load

// Display the results
disp(" Example 4-12 Solution : ");
printf(" \n Ist = %d A ( Back EMF is zero )",I_st );
printf(" \n Percentage at full load = %d percent ",
   percentage );
```

Scilab code Exa 4.13 calculate Rs at various back Emfs and Ec at zero Rs

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-13

clear; clc; close; // Clear the work space and console.

// Given data
V_a = 120 ; // Rated terminal voltage of dc shunt notor in volt
R_a = 0.2 ; // Armature resistance in ohm
BD = 2 ; // Brush drop in volt
I_a = 75 ; // Full load armature current in A
I_a_new = 1.5 * I_a ; // armature current in A at 150% rated load

E_c_a = 0 ; // Back EMF at starting
E_c_b = ( 25 / 100 ) * V_a ; // Back EMF in volt is 25% of Va @ 150% rated load
E_c_c = ( 50 / 100 ) * V_a ; // Back EMF in volt is 50% of Va @ 150% rated load

// Calculations
R_s_a = ( V_a - E_c_a - BD ) / I_a_new - R_a ; // Ra tapping value at starting
// in ohm
R_s_b = ( V_a - E_c_b - BD ) / I_a_new - R_a ; // Ra tapping value @ 25% of Va
//  in ohm
R_s_c = ( V_a - E_c_c - BD ) / I_a_new - R_a ; // Ra tapping value @ 50% of Va
```

```
// in ohm
E_c_d = V_a - ( I_a * R_a + BD ) ; // Back EMF @ full-load without starting resistance

// Display the results
disp(" Example 4-13 Solution : ");
printf(" \n a: At starting , Ec is zero ");
printf(" \n    Rs = %.2f ohm \n ", R_s_a );

printf(" \n b: When back EMF in volt is 25 percent of Va @ 150 percent rated load ");
printf(" \n    Rs = %.3f ohm \n ", R_s_b );

printf(" \n c: When back EMF in volt is 50 percent of Va @ 150 percent rated load ");
printf(" \n    Rs = %.3f ohm \n ", R_s_c );

printf(" \n d: Back EMF at full-load without starting resistance ");
printf(" \n    Ec = %d V ", E_c_d );
```

Scilab code Exa 4.14 calculate field flux in percent and final torque developed

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd edition

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-14

clear; clc; close; // Clear the work space and console.

```

```
// Given data
// Cumulative DC compound motor acting as shunt motor
T_orig = 160 ; // Original torque developed in lb.ft
I_a_orig = 140 ; // Original armature current in A
phi_f_orig = 1.6e+6 ; // Original field flux in lines

// Reconnected as a cumulative DC compound motor
T_final_a = 190 ; // Final torque developed in lb.ft (case a)

// Calculations
phi_f = phi_f_orig * ( T_final_a / T_orig ) ; // Field flux in lines
percentage = ( phi_f / phi_f_orig ) * 100 - 100 ; // percentage increase in flux

phi_f_final = 1.1 * phi_f ; // 10% increase in load causes 10% increase in flux
I_a_b = 154 ; // Final armature current in A (case b)
T_f = T_final_a * ( I_a_b / I_a_orig ) * ( phi_f_final / phi_f ) ;
// Final torque developed

// Display the results
disp(" Example 4-14 Solution : ");
printf(" \n a: phi_f = %.1e lines \n ", phi_f );
printf(" \n    Percentage of flux increase = %.1f percent \n ", percentage );

printf(" \n b: The final field flux is 1.1 * 1.9 * 10 ^ 6 lines " );
printf(" \n    (due to the 10 percent increase in load).The final torque is\n");
printf(" \n    T_f = %.1f lb-ft ", T_f );
```

Scilab code Exa 4.15 calculate torque developed for varying flux and Ia

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-15

clear; clc; close; // Clear the work space and console.

// Given data
I_a_orig = 25 ; // Original armature current in A
I_a_final = 30 ; // Final armature current in A
T_orig = 90 ; // Original torque developed in lb.ft
phi_orig = 1.0 ; // Original flux
phi_final = 1.1 ; // Final flux

// Calculations
T_a = T_orig * ( I_a_final / I_a_orig ) ^ 2 ; // Final torque developed if field
// is unsaturated
T_b = T_orig * ( I_a_final / I_a_orig ) * ( phi_final / phi_orig ) ;
// Final torque developed when Ia rises to 30 A and flux by 10%

// Display the results
disp(" Example 4-15 Solution : ");
printf(" \n a: T = %.1f lb-ft \n ", T_a );
printf(" \n b: T = %.1f lb-ft ", T_b );
```

Scilab code Exa 4.16 calculate speed at rated load and P and hp

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-16

clear; clc; close; // Clear the work space and
    console.

// Given data
V_a = 230 ; // Rated armature voltage in volt
P = 10 ; // Rated power in hp
S = 1250 ; // Rated speed in rpm
R_A = 0.25 ; // Armature resistance in ohm
R_p = 0.25 ; // Interpolar resistance
BD = 5 ; // Brush voltage drop in volt
R_s = 0.15 ; // Series field resistance in ohm
R_sh = 230 ; // Shunt field resistance in ohm

// shunt connection
I_l = 54 ; // Line current in A at rated load
I_ol = 4 ; // No-load line current in A
S_o = 1810 ; // No-load speed in rpm

// Calculations
R_a = R_A + R_p ; // Effective armature resistance
    in ohm
I_f = V_a / R_sh ; // Field current in A ( Shunt
    connection )
I_a = I_ol - I_f ; // Armature current in A
```

```
E_c_o = V_a - ( I_a * R_a + BD ); // No-load BACK EMF in volt
E_c_full_load = V_a - ( I_l * R_a + BD ); // No-load BACK EMF in volt at full-load

S_r = S_o * ( E_c_full_load / E_c_o ); // Speed at rated load

P_d = E_c_full_load * I_l ; // Internal power in watts
hp = P_d / 746 ; // Internal horse power

// Display the results
disp("Example 4-16 Solution : ");
printf(" \n a: S_r = %d rpm\n ", S_r );
printf(" \n b: P_d = %d W ", P_d );
printf(" \n    hp = %.1f hp ", hp );
```

Scilab code Exa 4.17 calculate speed torque and horsepower

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-17

clear; clc; close; // Clear the work space and console.

// Given data
V_a = 230 ; // Rated armature voltage in volt
P = 10 ; // Rated power in hp
```

```
S = 1250 ; // Rated speed in rpm
R_A = 0.25 ; // Armature resistance in ohm
R_p = 0.25 ; // Interpolar resistance
BD = 5 ; // Brush voltage drop in volt
R_s = 0.15 ; // Series field resistance in ohm
R_sh = 230 ; // Shunt field resistance in ohm
phi_1 = 1 ;// Original flux per pole

// Long-shunt cumulative connection
I_l = 55 ; // Line current in A at rated load
phi_2 = 1.25 ; // Flux increased by 25% due to long-shunt cumulative connection
I_ol = 4 ; // No-load line current in A
S_o = 1810 ; // No-load speed in rpm

// Calculations
R_a = R_A + R_p ; // Effective armature resistance in ohm
I_f = V_a / R_sh ; // Field current in A in shunt winding
I_a = I_ol - I_f ; // Armature current in A for shunt connection
E_c_o = V_a - ( I_a * R_a + BD ); // No-load BACK EMF in volt for shunt connection
E_c_o1 = V_a - ( I_a * R_a + I_a * R_s + BD ); // No-load BACK EMF in volt for
// long shunt cumulative connection
S_n1 = S_o * ( E_c_o1 / E_c_o ); // Speed at no load

I_f = V_a / R_sh ; // Field current in A in shunt winding
I_a_lsh = I_l - I_f ; // Armature current in A
E_c_full_load = V_a - ( I_a_lsh * R_a + BD ); // No-load BACK EMF in volt at
// full-load for long-shunt cumulative connection

E_c_full_load_lsh = V_a - ( I_a_lsh * R_a + I_a_lsh * R_s + BD ); // BACK EMF in volt
```

```
// at full-load for long-shunt cumulative motor

S_r = S_o * ( E_c_full_load / E_c_o ); // Speed at rated load for shunt connection
S_r_lsh = S_n1 * ( E_c_full_load_lsh / E_c_o1 ) * ( phi_1 / phi_2 );
// Speed at rated load for shunt connection

P_d = E_c_full_load * I_a_lsh ; // Internal power in watts
hp = P_d / 746 ; // Internal horse power

T_shunt = ( hp * 5252 ) / S_r ; // Internal torque @ full-load for shunt motor

I_a1 = I_a_lsh; // Armature current for shunt motor in A
I_a2 = I_a_lsh; // Armature current for long-shunt cumulative motor in A
T_comp = T_shunt * ( phi_2 / phi_1 ) * ( I_a2 / I_a1 ); // Internal torque
// at full-load for long-shunt cumulative motor in A

Horsepower = ( E_c_full_load_lsh * I_a_lsh ) / 746 ; // Internal horsepower of
// compound motor based on flux increase

// Display the results
disp(" Example 4-17 Solution : ");
printf(" \n a: S_n1 = %d rpm \n", S_n1 );
printf(" \n b: S_r = %d rpm \n", S_r_lsh );
printf(" \n c: Internal torque of shunt motor at full-load : ");
printf(" \n    T_shunt = %.2f lb-ft ", T_shunt );
printf(" \n    T_comp = %.2f lb-ft \n", T_comp );
printf(" \n d: Horsepower = %.1f hp \n", Horsepower );
printf(" \n e: The internal horsepower exceeds the
```

```
    rated horsepower because ");
printf(" \n    the power developed in the motor must
     also overcome the internal");
printf(" \n    mechanical rotational losses. ");
```

Scilab code Exa 4.18 calculate speed with and without diverter

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-18

clear; clc; close; // Clear the work space and
    console.

// Given data
P = 25 ; // Power rating of a series motor in hp
V_a = 250 ; // Rated voltage in volt
R_a = 0.1 ; // Armature ckt resistance in ohm
BD = 3 ; // Brush voltage drop in volt
R_s = 0.05 ; // Series field resistance in ohm
I_a1 = 85 ; // Armature current in A (case a)
I_a2 = 100 ; // Armature current in A (case b)
I_a3 = 40 ; // Armature current in A (case c)
S_1 = 600 ; // Speed in rpm at current I_a1
R_d = 0.05 ; // Divertor resistance in ohm

// Calculations
E_c2 = V_a - I_a2 * ( R_a + R_s ) - BD ; // BACK EMF
     in volt for I_a2
E_c1 = V_a - I_a1 * ( R_a + R_s ) - BD ; // BACK EMF
     in volt for I_a1
```

```

S_2 = S_1 * ( E_c2 / E_c1 ) * ( I_a1 / I_a2 ); // Speed in rpm at current I_a2

E_c3 = V_a - I_a3 * ( R_a + R_s ) - BD ; // BACK EMF in volt for I_a3

S_3 = S_1 * ( E_c3 / E_c1 ) * ( I_a1 / I_a3 ); // Speed in rpm at current I_a3

// When divertor is connected in parallel to R_s
R_sd = ( R_s * R_d ) / ( R_s + R_d ); // Effective series field resistance in ohm

E_c2_new = V_a - I_a2 * ( R_a + R_sd ) - BD ; // BACK EMF in volt for I_a2
S_2_new = S_1 * ( E_c2_new / E_c1 ) * ( I_a1 / ( I_a2 / 2 ) ); // Speed in rpm
// at current I_a2

E_c3_new = V_a - I_a3 * ( R_a + R_sd ) - BD ; // BACK EMF in volt for I_a3
S_3_new = S_1 * ( E_c3_new / E_c1 ) * ( I_a1 / ( I_a3 / 2 ) ); // Speed in rpm
// at current I_a3

// Display the results
disp(" Example 4-18 Solution : ");
printf(" \n a: S_2 = %d rpm \n", S_2 );
printf(" \n b: S_3 = %d rpm \n", S_3 );
printf(" \n c: The effect of the divertor is to reduce the series field current");
printf(" \n    (and flux) to half their previous values. ");
printf(" \n     S_2 = %d rpm ", S_2_new );
printf(" \n     S_3 = %d rpm \n", S_3_new );

printf(" \n    The results may be tabulated as
```

```
   follows : \n ");
printf(" \n     Case \t I_a in A \t S_o in rpm \t
   S_d in rpm ");
printf(" \n
   -----------------------------------------------------------
   ");
printf(" \n      1     \t   %d        \t   %d
   \t   ___ ", I_a1 , S_1  );
printf(" \n      2.    \t   %d        \t   %d
   \t   %d ", I_a2 , S_2 , S_2_new );
printf(" \n      3.    \t   %d        \t   %d        \t
      %d ", I_a3 , S_3 , S_3_new );
```

Scilab code Exa 4.19 calculate percentage speed regulation

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-19

clear; clc; close; // Clear the work space and
   console.

// Given data
// From the calculations of Ex.4-16 , Ex.4-17 , Ex
   .4-18 we get no-load and
// full-load speeds as follows
S_n1 = 1810 ; // No-load speed in rpm (Ex.4-16)
S_f1 = 1603 ; // Full-load speed in rpm (Ex.4-16)

S_n2 = 1806 ; // No-load speed in rpm (Ex.4-17)
S_f2 = 1231 ; // Full-load speed in rpm (Ex.4-17)
```

```
S_n3 = 1311 ; // No-load speed in rpm (Ex.4-18)
S_f3 = 505 ; // Full-load speed in rpm (Ex.4-18)

// Calculations
SR_1 = ( S_n1 - S_f1 ) / S_f1 * 100 ; // Speed
    regulation for shunt motor

SR_2 = ( S_n2 - S_f2 ) / S_f2 * 100 ; // Speed
    regulation for compound motor

SR_3 = ( S_n3 - S_f3 ) / S_f3 * 100 ; // Speed
    regulation for series motor

// Display the results
disp("Example 4-19 Solution : ");
printf(" \n a: SR(shunt) = %.1f percent \n ", SR_1 )
    ;
printf(" \n b: SR(compound) = %.1f percent \n ",
    SR_2 );
printf(" \n c: SR(series) = %.1f percent \n ", SR_3
    );
```

Scilab code Exa 4.20 calculate no load speed

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-20

clear; clc; close; // Clear the work space and
    console.
```

```
// Given data
SR = 0.1 ; // Given percent speed regulation 10% of
    a shunt motor
omega_f1 = 60 * %pi ; // Full-load speed in rad/s

// Calculations
omega_n1 = omega_f1 * ( 1 + SR ); // No-load speed
    in rad/s

S = omega_n1 * ( 1 / ( 2 * %pi )) * ( 60 / 1 ); //
    No-load speed in rpm

// Display the results
disp("Example 4-20 Solution : ");
printf(" \n a: omega_n1 = %.2f \n ", omega_n1);
printf(" \n b: S = %d rpm ", S );
```

Scilab code Exa 4.21 calculate internal and external torque

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-21

clear; clc; close; // Clear the work space and
    console.

// Given data
S_int = 1603 ; // Internal rated speed in rpm (Ex
    .4-16)
S_ext = 1250 ; // External rated speed in rpm (Ex
```

```
.4-16)
hp_int = 14.3 ; // Internal horsepower
hp_ext = 10 ; // External horsepower

// Calculations
T_int = ( hp_int * 5252 ) / S_int ; // Internal torque in lb-ft

T_ext = ( hp_ext * 5252 ) / S_ext ; // External torque in lb-ft

// Display the results
disp("Example 4-21 Solution : ");
printf(" \n a: T_int = %.2f lb-ft \n ", T_int );
printf(" \n b: T_ext = %.2f lb-ft \n ", T_ext );
printf(" \n c: Internal horsepower is developed as a result of electromagnetic");
printf(" \n    torque produced by energy conversion. Some of the mechanical energy");
printf(" \n    is used internally to overcome mechanical losses of the motor,");
printf(" \n    reducing the torque available at its shaft to perform work.");
```

Scilab code Exa 4.22 calculate output torque in ounceinches

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-22

clear; clc; close; // Clear the work space and
```

```
console.

// Given data
P = 50 ; // Power rating of the servo-motor in W
S = 3000 ; // Full-load speed of the servo-motor in rpm

// Calculation
T_lbft = ( 7.04 * P ) / S ; // Output torque in lb-ft
T_ounceinch = T_lbft * 192 ; // Output torque in ounce-inches

// Display the result
disp(" Example 4-22 Solution : ");
printf(" \n  T = %.1f oz.in ", T_ounceinch );
```

Scilab code Exa 4.23 calculate speed and torque

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 4: DC Dynamo Torque Relations-DC Motors
// Example 4-23

clear; clc; close; // Clear the work space and console.

// Given data
P = 50 ; // Power rating of the servo-motor in W
S_rpm = 3000 ; // Full-load speed of the servo-motor in rpm

```

```
// Calculations
S_rad_per_sec = S_rpm * 2 * %pi / 60 ; // Full-load speed of the servo-motor
// in rad/s
omega = 314.2 ; // Angular frequency in rad/s
T_Nm = P / omega ; // Output torque in Nm
T_ounceinch = T_Nm * ( 1 / 7.0612e-3 ) ; // Output torque in oz.in

// Display the results
disp("Example 4-23 Solution : ");
printf(" \n a: Speed in rad/s = %.1f rad/s \n ", S_rad_per_sec );
printf(" \n b: T = %.4f N-m \n ", T_Nm );
printf(" \n c: T = %.1f oz.in \n ", T_ounceinch );
printf(" \n d: Both answers are the same.");
```

Chapter 5

ARMATURE REACTION AND COMMUTATION IN DYNAMOS

Scilab code Exa 5.1 calculate Zp

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 5: ARMATURE REACTION AND COMMUTATION IN
   DYNAMOS
// Example 5-1

clear; clc; close; // Clear the work space and
   console.

// Given data
conductors = 800 ; // No. of conductors
I_a = 1000 ; // Rated armature current in A
P = 10 ; // No. of poles
pitch = 0.7 ; // Pole-face covers 70% of the pitch
```

```
a = P ; // No. of parallel paths ( Simplex lap-wound )

// Calculations
// Using Eq.(5-1)
Z = conductors / P ; // No. of armature conductors/path under each pole
Z_a = Z * pitch ; // Active armature conductors/pole

// Solving for Z_p using Z_p = Z_a / a
Z_p = Z_a / a ; // No. of pole face conductors/pole

// Display the results
disp("Example 5-1 Solution : ");
printf(" \n No. of pole face conductors/pole to give full armature reaction ");
printf(" \n compensation, if the pole covers 70 percent of the pitch is : \n ");
printf(" \n Z_p = %.1f conductors/pole ", Z_p );
```

Scilab code Exa 5.2 calculate cross and de magnetising ampereconductorsperpole and ampereturnsperpole

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 5: ARMATURE REACTION AND COMMUTATION IN DYNAMOS
// Example 5-2

clear; clc; close; // Clear the work space and console.

```

```
// Given data
conductors = 800 ; // No. of conductors
I_a = 1000 ; // Rated armature current in A
I_l = I_a ; // load or total current entering the armature in A
P = 10 ; // No. of poles
pitch = 0.7 ; // Pole-face covers 70% of the pitch
a = P ; // No. of parallel paths ( Simplex lap-wound )
alpha = 5 ; // No. of electrical degress that the brushes are shifted

// Calculations
Z = conductors / P ; // No. of armature conductors/path under each pole
A_Z_per_pole = ( Z * I_l ) / ( P * a ); // Cross magnetizing
// ampere-conductors/pole

At_per_pole = ( 1 / 2 ) * ( 8000 / 1 ); // Ampere-turns/pole

frac_demag_At_per_pole  = (2*alpha) / 180 * ( At_per_pole);
// Fraction of demagnetizing ampere-turns/pole

funcprot(0); // to avoid redefining function: beta warning message

beta = 180 - 2*alpha ; // cross-magnetizing electrical degrees

cross_mag_At_per_pole = (beta/180)*(At_per_pole);
// cross-magnetizing ampere-turns/pole

// Display the results
disp("Example 5-2 Solution : ");
printf(" \n a: With the brushes on the GNA, the
```

```
entire armature reaction effect");
printf(" \n    is completely cross-magnetizing. The cross-magnetizing  ");
printf(" \n    ampere-conductors/pole are ");
printf(" \n    = %d ampere-conductots/pole \n", A_Z_per_pole);

printf(" \n    and since there are 2 conductors/turn, the cross-magnetizing ");
printf(" \n    ampere-turns/pole are \n    = %d At/pole \n\n", At_per_pole );


printf(" \n b: Let alpha = the no. of electrical degrees that the brushes are ");
printf(" \n    shifted. Then the total no. of demagnetizing electrical degrees ");
printf(" \n    are 2*alpha, while the (remaining) cross-magnetizing electrical");
printf(" \n    degrees,beta, are 180 - 2*alpha. The ratio of demagnetizing to ");
printf(" \n    cross-magnetizing ampere-turns is always 2*alpha/beta. The ");
printf(" \n    fraction of demagnetizing ampere-turns/pole is ");
printf(" \n    = %.1f At/pole \n\n", frac_demag_At_per_pole );
printf(" \n    Note: Slight calculation mistake in the textbook for case b\n")


printf(" \n c: Since beta = 180-2*alpha = 170, the cross-magnetizing ampere-turns/pole ");
printf(" \n    are \n    = %.1f At/pole ", cross_mag_At_per_pole );
```

Chapter 6

AC DYNAMO VOLTAGE RELATIONS ALTERNATORS

Scilab code Exa 6.1 calculate Eg at unity PF and point75 lagging PF

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 6: AC DYNAMO VOLTAGE RELATIONS−
   ALTERNATORS
// Example 6−1

clear; clc; close; // Clear the work space and
   console.

// Given data
kVA = 1000 ; // kVA rating of the 3−phase alternator
V_L = 4600 ; // Rated line voltage in volt
// 3−phase, Y−connected alternator
R_a = 2 ; // Armature resistance in ohm per phase
X_s = 20 ; // Synchronous armature reactance in ohm
   per phase
```

```
cos_theta_a = 1 ; // Unity power factor (case a)
cos_theta_b = 0.75 ; // 0.75 power factor lagging (case b)
sin_theta_b = sqrt( 1 - (cos_theta_b)^2 );

// Calculations
V_P = V_L / sqrt(3) ; // Phase voltage in volt
I_P = ( kVA * 1000 ) / ( 3*V_P ) ; // Phase current in A
I_a  = I_P ; // Armature current in A

// a: At unity  PF
E_g_a = ( V_P + I_a * R_a ) + %i*(I_a*X_s);
// Full-load generated voltage per-phase (case a)
E_g_a_m=abs(E_g_a);//E_g_a_m=magnitude of E_g_a in volt
E_g_a_a=atan(imag(E_g_a) /real(E_g_a))*180/%pi;//E_g_a_a=phase angle of E_g_a in degrees

// b: At 0.75 PF lagging
E_g_b = ( V_P*cos_theta_b + I_a * R_a ) + %i*( V_P*sin_theta_b + I_a*X_s );
// Full-load generated voltage per-phase (case b )
E_g_b_m=abs(E_g_b);//E_g_b_m=magnitude of E_g_b in volt
E_g_b_a=atan(imag(E_g_b) /real(E_g_b))*180/%pi;//E_g_b_a=phase angle of E_g_b in degrees


// Display the results
disp("Example 6-1 Solution : ");
printf("\n root 3 value is taken as %f , so slight variations in the answer.", sqrt(3));
printf("\n\n a: At unity PF, \n ");
printf("\n     Rectangular form :\n   E_g = "); disp(E_g_a);
printf("\n     Polar form :");
printf(" \n      E_g = %d <%.2f V/phase ", E_g_a_m ,
```

```
E_g_a_a );
printf(" \n      where %d is magnitude and %.2f is phase angle\n",E_g_a_m,E_g_a_a);

printf(" \n b: At 0.75 PF lagging , \n ");
printf("\n     Rectangular form :\n    E_g = "); disp(E_g_b);
printf("\n     Polar form :");
printf(" \n      E_g = %d <%.2f V/phase ", E_g_b_m , E_g_b_a );
printf(" \n      where %d is magnitude and %.2f is phase angle\n",E_g_b_m,E_g_b_a);
```

Scilab code Exa 6.2 calculate Eg at point75 PF and point4 lead

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 6: AC DYNAMO VOLTAGE RELATIONS–ALTERNATORS
// Example 6–2

clear; clc; close; // Clear the work space and console.

// Given data
kVA = 1000 ; // kVA rating of the 3-phase alternator
V_L = 4600 ; // Rated line voltage in volt
// 3-phase, Y-connected alternator
R_a = 2 ; // Armature resistance in ohm per phase
X_s = 20 ; // Synchronous armature reactance in ohm per phase
cos_theta_a = 0.75 ; // 0.75 PF leading (case a)
```

```
cos_theta_b = 0.40 ; // 0.40 PF leading (case b)
sin_theta_a = sqrt( 1 - (cos_theta_a)^2 ); // (case a)
sin_theta_b = sqrt( 1 - (cos_theta_b)^2 ); // (case b)

// Calculations
V_P = V_L / sqrt(3) ; // Phase voltage in volt
I_P = ( kVA * 1000 ) / ( 3*V_P ) ; // Phase current in A
I_a  = I_P ; // Armature current in A

// a: At 0.75 PF leading
E_g_a = ( V_P*cos_theta_a + I_a * R_a ) + %i*( V_P*sin_theta_a - I_a*X_s);
// Full-load generated voltage per-phase (case a)
E_g_a_m=abs(E_g_a);//E_g_a_m=magnitude of E_g_a in volt
E_g_a_a=atan(imag(E_g_a) /real(E_g_a))*180/%pi;//E_g_a_a=phase angle of E_g_a in degrees

// b: At 0.40 PF leading
E_g_b = ( V_P*cos_theta_b + I_a * R_a ) + %i*( V_P*sin_theta_b - I_a*X_s );
// Full-load generated voltage per-phase (case b )
E_g_b_m=abs(E_g_b);//E_g_b_m=magnitude of E_g_b in volt
E_g_b_a=atan(imag(E_g_b) /real(E_g_b))*180/%pi;//E_g_b_a=phase angle of E_g_b in degrees


// Display the results
disp("Example 6-2 Solution : ");
printf("\n root 3 value is taken as %f , so slight variations in the answer.", sqrt(3));
printf("\n\n a: 0.75 PF leading, \n ");
printf("\n     Rectangular form :\n   E_g = "); disp(E_g_a);
```

```
printf("\n      Polar form :");
printf(" \n      E_g = %d <%.2f V/phase ", E_g_a_m ,
   E_g_a_a );
printf(" \n      where %d is magnitude and %.2f is
   phase angle\n",E_g_a_m,E_g_a_a);

printf(" \n b: At 0.40 PF leading , \n ");
printf("\n      Rectangular form :\n   E_g = "); disp
   (E_g_b);
printf("\n      Polar form :");
printf(" \n      E_g = %d <%.2f V/phase ", E_g_b_m ,
   E_g_b_a );
printf(" \n      where %d is magnitude and %.2f is
   phase angle\n",E_g_b_m,E_g_b_a);
```

Scilab code Exa 6.3 calculate percent voltage regulation

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 6: AC DYNAMO VOLTAGE RELATIONS−
   ALTERNATORS
// Example 6−3

clear; clc; close; // Clear the work space and
   console.

// Given data
// From Ex.6−1 and Ex.6−2 we have V_P and E_g values
    as follows
// Note : approximated values are considered when
   root 3 value is taken as 1.73
// as in textbook
```

```
V_P = 2660 ; // Phase voltage
E_g_a1 = 3836 ; // E_g at unity PF (Ex.6-1 case a)
E_g_b1 = 4814 ; // E_g at 0.75 PF lagging (Ex.6-1 case b)

E_g_a2 = 2364 ; // E_g at 0.75 PF leading (Ex.6-2 case a)
E_g_b2 = 1315 ; // E_g at 0.40 PF leading (Ex.6-2 case b)

// Calculations
VR_a = ( E_g_a1 - V_P )/V_P * 100 ; // voltage regulation at unity PF  (Ex.6-1 case a)
VR_b = ( E_g_b1 - V_P )/V_P * 100 ; // voltage regulation at 0.75 PF lagging (Ex.6-1 case b)

VR_c = ( E_g_a2 - V_P )/V_P * 100 ; // voltage regulation at 0.75 PF leading (Ex.6-2 case a)
VR_d = ( E_g_b2 - V_P )/V_P * 100 ; // voltage regulation at 0.40 PF leading (Ex.6-2 case b)

// Display the results
disp("Example 6-3 Solution : ");
printf(" \n a: At unity PF : ");
printf(" \n    VR = %.1f percent \n ", VR_a );

printf(" \n b: At 0.75 PF lagging : ");
printf(" \n    VR = %.2f percent \n ", VR_b );

printf(" \n c: At 0.75 PF leading : ");
printf(" \n    VR = %.2f percent \n ", VR_c );

printf(" \n d: At 0.40 PF leading : ");
printf(" \n    VR = %.1f percent \n ", VR_d );
```

Scilab code Exa 6.4 calculate Rdc Rac Zp Xs VR at point8 PF lag and lead

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 6: AC DYNAMO VOLTAGE RELATIONS–ALTERNATORS
// Example 6–4

clear; clc; close; // Clear the work space and console.

// Given data
kVA = 100 ; // kVA rating of the 3–phase alternator
V_L = 1100 ; // Line voltage of the 3–phase alternator in volt

// dc–resistance test data
E_gp1 = 6 ; // generated phase voltage in volt
V_l = E_gp1 ; // generated line voltage in volt
I_a1 = 10 ; // full–load current per phase in A
cos_theta_b1 = 0.8 ; // 0.8 PF lagging (case b)
cos_theta_b2 = 0.8 ; // 0.8 PF leading (case b)
sin_theta_b1 = sqrt( 1 - (cos_theta_b1)^2 ); // (case b)
sin_theta_b2 = sqrt( 1 - (cos_theta_b2)^2 ); // (case b)

// open–circuit test data
E_gp2 = 420 ; // generated phase voltage in volt
I_f2 = 12.5 ; // Field current in A

// short–circuit test data
I_f3 = 12.5 ; // Field current in A
// Line current I_l = rated value   in A
```

```
// Calculations
// Assuming that the alternator is Y-connected
// case a :
I_a_rated = (kVA*1000)/(V_L*sqrt(3)); // Rated current per phase in A
I_a = sqrt(3)*I_a_rated ; // Rated Line current in A

R_dc = V_l/(2*I_a1); // effective dc armature resistance in ohm/winding
R_ac = R_dc * 1.5 ; // effective ac armature resistance in ohm.phase
R_a = R_ac ; // effective ac armature resistance in ohm.phase from dc resistance test

Z_p = E_gp2 / I_a ; // Synchronous impedance per phase
X_s = sqrt( Z_p^2 - R_a^2 ); // Synchronous reactance per phase

// case b :
V_p = V_L / sqrt(3); // Phase voltage in volt (Y-connection)

// At 0.8 PF lagging
E_gp1 = ( V_p*cos_theta_b1 + I_a_rated * R_a ) + %i *( V_p*sin_theta_b1 + I_a_rated * X_s);
E_gp1_m=abs(E_gp1);//E_gp1_m=magnitude of E_gp1 in volt
E_gp1_a=atan(imag(E_gp1) /real(E_gp1))*180/%pi;//E_gp1_a=phase angle of E_gp1 in degrees
V_n1 = E_gp1_m ; // No-load voltage in volt
V_f1 = V_p ; // Full-load voltage in volt
VR1 = ( V_n1 - V_f1 )/ V_f1 * 100; // percent voltage regulation at 0.8 PF lagging


// At 0.8 PF leading
```

```
E_gp2 = ( V_p*cos_theta_b2 + I_a_rated * R_a ) + %i *( V_p*sin_theta_b2 - I_a_rated*X_s);
E_gp2_m=abs(E_gp2);//E_gp2_m=magnitude of E_gp2 in volt
E_gp2_a=atan(imag(E_gp2) /real(E_gp2))*180/%pi;//E_gp2_a=phase angle of E_gp2 in degrees
V_n2 = E_gp2_m ; // No-load voltage in volt
V_f2 = V_p ; // Full-load voltage in volt
VR2 = ( V_n2 - V_f2 )/V_f2 * 100 ; // percent voltage regulation at 0.8 PF leading

// Display the results
disp("Example 6-4 Solution : ");
printf(" \n Assuming that the alternator is Y-connected ");
printf(" \n a: R_dc = %.1f ohm/winding ", R_dc );
printf(" \n    R_ac = %.2f ohm/phase ", R_ac );
printf(" \n    Z_p = %.2f ohm/phase ", Z_p );
printf(" \n    X_s = %.2f ohm/phase \n", X_s );

printf(" \n b: At 0.8 PF lagging ");
printf(" \n    Percent voltage regulation = %.1f percent \n", VR1 );

printf(" \n    At 0.8 PF leading ");
printf(" \n    Percent voltage regulation = %.1f percent ", VR2 );
```

Scilab code Exa 6.5 calculate prev eg values for delta connection

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

```

```
// Chapter 6: AC DYNAMO VOLTAGE RELATIONS—
   ALTERNATORS
// Example 6−5

clear; clc; close; // Clear the work space and
   console.

// Given data
kVA = 100 ; // kVA rating of the 3−phase alternator
V_L = 1100 ; // Line voltage of the 3−phase
   alternator in volt

// dc−resistance test data
E_gp1 = 6 ; // generated phase voltage in volt
V_l = E_gp1 ; // generated line voltage in volt
I_a1 = 10 ; // full−load current per phase in A
cos_theta_b1 = 0.8 ; // 0.8 PF lagging (case b)
cos_theta_b2 = 0.8 ; // 0.8 PF leading (case b)
sin_theta_b1 = sqrt( 1 - (cos_theta_b1)^2 ); // (
   case b)
sin_theta_b2 = sqrt( 1 - (cos_theta_b2)^2 ); // (
   case b)

// open−circuit test data
E_gp2 = 420 ; // generated phase voltage in volt
I_f2 = 12.5 ; // Field current in A

// short−circuit test data
I_f3 = 12.5 ; // Field current in A
// Line current I_l = rated value  in A

// Calculations
// Assuming that the alternator is delta−connected
// case a :
I_a_rated = (kVA*1000)/(V_L*sqrt(3)); // Rated
   current per phase in A
I_L = I_a_rated ; // Line current in A

```

```
V_p = E_gp2 ; // Phase voltage in volt
V_l = V_p ; // Line voltage in volt (from short circuit data)

I_p = I_L / sqrt(3) ; // Phase current in A (delta connection)
I_a = I_p ; // Rated current in A

Z_s = V_l / I_p ; // Synchronous impedance per phase
R_dc = E_gp1/(2*I_a1); // effective dc armature resistance in ohm/winding
R_ac = R_dc * 1.5 ; // effective ac armature resistance in ohm.phase

// R_eff in delta = 3 * R_eff in Y
R_eff = 3 * R_ac ; // Effective armature resistance in ohm
R_a = R_eff ; // effective ac armature resistance in ohm.phase from dc resistance test

X_s = sqrt( Z_s^2 - R_a^2 ); // Synchronous reactance per phase

V_p = V_L ; // Phase voltage in volt (delta-connection)

// At 0.8 PF lagging
E_gp1 = ( V_p*cos_theta_b1 + I_a * R_a ) + %i*( V_p*sin_theta_b1 + I_a*X_s);
E_gp1_m=abs(E_gp1);//E_gp1_m=magnitude of E_gp1 in volt
E_gp1_a=atan(imag(E_gp1) /real(E_gp1))*180/%pi;//E_gp1_a=phase angle of E_gp1 in degrees
V_n1 = E_gp1_m ; // No-load voltage in volt
V_f1 = V_p ; // Full-load voltage in volt
VR1 = ( V_n1 - V_f1 )/ V_f1 * 100; // percent voltage regulation at 0.8 PF lagging

```

```
// At 0.8 PF leading
E_gp2 = ( V_p*cos_theta_b2 + I_a * R_a ) + %i*( V_p*sin_theta_b2 - I_a*X_s);
E_gp2_m=abs(E_gp2);//E_gp2_m=magnitude of E_gp2 in volt
E_gp2_a=atan(imag(E_gp2) /real(E_gp2))*180/%pi;//E_gp2_a=phase angle of E_gp2 in degrees
V_n2 = E_gp2_m ; // No-load voltage in volt
V_f2 = V_p ; // Full-load voltage in volt
VR2 = ( V_n2 - V_f2 )/V_f2 * 100 ; // percent voltage regulation at 0.8 PF leading

// Display the results
disp("Example 6-5 Solution : ");
printf(" \n Assuming that the alternator is delta-connected : \n ");
printf(" \n a: I_p = %.3f A ", I_p );
printf(" \n    Z_s = %.2f ohm/phase ", Z_s );
printf(" \n    R_eff in delta = %.2f ohm/phase ", R_eff );
printf(" \n    X_s = %.1f ohm/phase \n", X_s );
printf(" \n    R_eff, reactance and impedance per phase in delta is 3 times")
printf(" \n    the value when connected in Y. \n")

printf(" \n b: At 0.8 PF lagging ");
printf(" \n    Percent voltage regulation = %.1f percent \n", VR1 );

printf(" \n    At 0.8 PF leading ");
printf(" \n    Percent voltage regulation = %.1f percent \n", VR2 );
printf(" \n    Percentage voltage regulation remains the same both in Y and delta connection.");
```

Scilab code Exa 6.6 calculate Imax overload and Isteady

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 6: AC DYNAMO VOLTAGE RELATIONS−
   ALTERNATORS
// Example 6−6

clear; clc; close; // Clear the work space and
   console.

// Given data
// 3−phase Y−connected alternator
E_L = 11000 ; // Line voltage generated in volt
kVA = 165000 ; // kVA rating of the alternator
R_p = 0.1 ; // Armature resistance in ohm/per phase
Z_p = 1.0 ; // Synchronous reactance/phase
Z_r = 0.8 ; // Reactor reactance/phase

// Calculations
E_p = E_L / sqrt(3); // Rated phase voltage in volt
I_p = (kVA * 1000)/(3*E_p); // Rated current per
   phase in A

// case a
I_max_a = E_p / R_p ; // Maximum short−circuit
   current in A (case a)
overload_a = I_max_a / I_p ; // Overload (case a)

// case b
I_steady = E_p / Z_p ; // Sustained short−circuit
```

```
current in A
overload_b = I_steady / I_p ; // Overload (case b)

// case c
Z_t = R_p + %i*Z_r ; // Total reactance per phase
I_max_c = E_p / Z_t ; // Maximum short-circuit current in A (case b)
I_max_c_m=abs(I_max_c);//I_max_c_m=magnitude of I_max_c in A
I_max_c_a=atan(imag(I_max_c) /real(I_max_c))*180/%pi;//I_max_c_a=phase angle of I_max_c in degrees
overload_c = I_max_c_m / I_p ; // Overload (case a)

// Display the results
disp("Example 6-6 Solution : ");
printf("\n root 3 value is taken as %f , so slight variations in the answer.\n", sqrt(3));
printf(" \n a: I_max = %d A ", I_max_a );
printf(" \n    overload = %.1f * rated current \n", overload_a );

printf(" \n b: I_steady = %d A ", I_steady );
printf(" \n    overload = %.2f * rated current \n", overload_b );

printf(" \n c: Rectangular form :\n   I_max = "); disp(I_max_c);
printf(" \n    Polar form :");
printf(" \n    I_max = %d <%.2f A ", I_max_c_m , I_max_c_a );
printf(" \n    where %d is magnitude and %.2f is phase angle\n",I_max_c_m,I_max_c_a);
printf(" \n    overload = %.3f * rated current \n", overload_c );
```

Scilab code Exa 6.7 calculate P and Pperphase and Egp magnitude phase angle and torque angle

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 6: AC DYNAMO VOLTAGE RELATIONS−
    ALTERNATORS
// Example 6−7

clear; clc; close; // Clear the work space and
    console.

// Given data
kVA = 100 ; // kVA rating of the 3−phase alternator
V_L = 1100 ; // Line voltage of the 3−phase
    alternator in volt

// dc−resistance test data
E_gp1 = 6 ; // generated phase voltage in volt
V_l = E_gp1 ; // generated line voltage in volt
I_a1 = 10 ; // full−load current per phase in A
cos_theta = 0.8 ; // 0.8 PF lagging
sin_theta = sqrt( 1 - (cos_theta)^2 ); //

// open−circuit test data
E_gp2 = 420 ; // generated phase voltage in volt
I_f2 = 12.5 ; // Field current in A

// short−circuit test data
I_f3 = 12.5 ; // Field current in A
// Line current I_l = rated value in A

// Calculated data from Ex.6−4
I_L = 52.5 ; // Rated line current in A
I_a = I_L ; // Rated current per phase in A
```

```
E_gp = 532 + %i*623 ; // Generated voltage at 0.8 PF lagging
X_s = 4.6 ; // Synchronous reactance per phase
V_p = 635 ; // Phase voltage in volt

// Calculations
// case a
P_T = sqrt(3) * V_L * I_L * cos_theta ; // Total output 3-phase power

// case b
P_p_b = P_T / 3 ; // Total output 3-phase power per phase

// case c
E_gp_m=abs(E_gp);//E_gp_m=magnitude of E_gp in volt
E_gp_a=atan(imag(E_gp) /real(E_gp))*180/%pi;//E_gp_a=phase angle of E_gp in degrees

// case d
theta = acos(0.8)*180/%pi; // phase angle for PF in degrees
theta_plus_deba = E_gp_a ; // phase angle of E_gp in degrees
deba = theta_plus_deba - theta ; // Torque angle in degrees

// case e
P_p_e = (E_gp_m/X_s)*V_p*sind(deba); // Approximate output power/phase (Eq.(6-10))

// case f
P_p_f = E_gp_m * I_a * cosd(theta_plus_deba); // Approximate output power/phase (Eq.(6-9))

// Display the results
disp("Example 6-7 Solution : ");
printf("\n root 3 value is taken as %f , so slight
```

```
    variations in the answer.\n", sqrt(3));
printf(" \n a: P_T = %d W \n", P_T );
printf(" \n b: P_p = %.2f W \n", P_p_b );
printf(" \n c: E_gp = %d <%.2f V \n", E_gp_m , E_gp_a
    );
printf(" \n    where %d is magnitude in V and %.2f
    is phase angle in degrees.\n",E_gp_m,E_gp_a);
printf(" \n d: Torque angle , deba = %.2f degrees \n"
    , deba );
printf(" \n e: P_p = %d W \n", P_p_e );
printf(" \n f: P_p = %d W ", P_p_f );
```

Scilab code Exa 6.8 calculate torqueperphase and total torque

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 6: AC DYNAMO VOLTAGE RELATIONS-
    ALTERNATORS
// Example 6-8

clear; clc; close; // Clear the work space and
    console.

// Given data

kVA = 100 ; // kVA rating of the 3-phase alternator
V_L = 1100 ; // Line voltage of the 3-phase
    alternator in volt
S = 1200 ; // Synchronous speed in rpm

// dc-resistance test data
E_gp1 = 6 ; // generated phase voltage in volt
```

```
V_l = E_gp1 ; // generated line voltage in volt
I_a1 = 10 ; // full-load current per phase in A
cos_theta = 0.8 ; // 0.8 PF lagging
sin_theta = sqrt( 1 - (cos_theta)^2 ); //

// open-circuit test data
E_gp2 = 420 ; // generated phase voltage in volt
I_f2 = 12.5 ; // Field current in A

// short-circuit test data
I_f3 = 12.5 ; // Field current in A
// Line current I_l = rated value  in A

// Calculated data from Ex.6-4 & Ex.6-7
I_L = 52.5 ; // Rated line current in A
I_a = I_L ; // Rated current per phase in A
E_gp = 532 + %i*623 ; // Generated voltage at 0.8 PF lagging
E_g = 819 ; // E_g = magnitude of E_gp in volt
X_s = 4.6 ; // Synchronous reactance per phase
V_p = 635 ; // Phase voltage in volt
deba = 12.63 ; // Torque angle in degrees

// Calculations
// case a
T_p_a  = ( 7.04 * E_g * V_p * sind(deba) ) / (S*X_s) ; // Output torque per phase in lb.ft
T_3phase_a = 3 * T_p_a ; // Output torque for 3-phase in lb.ft

// case b
omega = S * 2*%pi *(1/60); // Angular frequency in rad/s
T_p_b = ( E_g * V_p * sind(deba))/(omega*X_s); // Output torque per phase in lb.ft
T_3phase_b = 3 * T_p_b ; // Output torque for 3-phase in lb.ft

```

```
// case c
T_p_c = T_p_a * 1.356 ; // Output torque per phase in N.m
T_3phase_c = 3 * T_p_c ; // Output torque for 3-phase in N.m

// Display the results
disp("Example 6-8 Solution : ");
pi = %pi;
printf(" \n    Slight variations in the answers are due to value of pi = %f ",pi);
printf(" \n    and omega = %f, which are slightly different as in the textbook.\n",omega);
printf(" \n a: T_p = %d lb-ft ",T_p_a);
printf(" \n    T_3phase = %d lb-ft \n", T_3phase_a);

printf(" \n b: T_p = %.1f N-m ",T_p_b);
printf(" \n    T_3phase = %.1f N-m \n", T_3phase_b);

printf(" \n c: T_p = %.1f N-m ",T_p_c);
printf(" \n    T_3phase = %.1f N-m \n", T_3phase_c);
printf(" \n    Answers from cases b and c almost tally each other ");
```

Chapter 7

PARALLEL OPERATION

Scilab code Exa 7.1 calculate I Ia and P

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 7: PARALLEL OPERATION
// Example 7-1

clear; clc; close; // Clear the work space and
   console.

// Given data
R_sh = 120 ; // Shunt field resistance in ohm
R_a = 0.1 ; // Armature resistance in ohm
V_L = 120 ; // Line voltage in volt
E_g1 = 125 ; // Generated voltage by dynamo A
E_g2 = 120 ; // Generated voltage by dynamo B
E_g3 = 114 ; // Generated voltage by dynamo C

// Calculations
// case a
```

```
// 1:
I_gA = ( E_g1 - V_L ) / R_a ; // Current in the generating source A ( in A)
I_f = V_L / R_sh ; // Shunt field current in A
I_a1 = I_gA + I_f ; // Armature current in A for generator A
I_L1 = I_gA ; // Current delivered by dynamo A to the bus in A

// 2:
I_gB = ( E_g2 - V_L ) / R_a ; // Current in the generating source B ( in A)
I_a2 = I_gB + I_f ; // Armature current in A for generator B
I_L2 = I_gB ; // Current delivered by dynamo B to the bus in A

// 3:
I_gC = ( V_L - E_g3 ) / R_a ; // Current in the generating source C ( in A)
I_a3 = I_gC ; // Armature current in A for generator C
I_L3 = I_gC + I_f ; // Current delivered by dynamo C to the bus in A

// case b
// 1:
P_LA = V_L * I_L1 ; // Power delivered to the bus by dynamo A in W
P_gA = E_g1 * I_a1 ; // Power generated by dynamo A

// 2:
P_LB = V_L * I_L2 ; // Power delivered to the bus by dynamo B in W
P_gB = E_g2 * I_a2 ; // Power generated by dynamo B

// 3:
P_LC = V_L * I_L3 ; // Power delivered to the bus by
```

```
dynamo C in W
P_gC = E_g3 * I_a3 ; // Power generated by dynamo C

// Display the results
disp("Example 7-1 Solution : ");
printf(" \n a: 1. I_gA = %d A \t I_f = %d A ", I_gA, I_f );
printf(" \n        Thus,dynamo A delivers %d A to the bus and has an armature", I_gA);
printf(" \n        current of %d A + %d A = %d \n", I_gA,I_f,I_a1 );

printf(" \n    2. I_gB = %d A ", I_gB);
printf(" \n        Thus, dynamo B is floating and has as armature & field current of %d A \n",I_f);

printf(" \n    3. I_gC = %d A ",I_gC);
printf(" \n        Dynamo C receives %d A from the bus & has an armature current of %d A\n",I_L3, I_a3);

printf(" \n b: 1. Power delivered to the bus by dynamo A is : ");
printf(" \n        P_LA = %d W ",P_LA);
printf(" \n        Power generated by dynamo A is \n        P_gA = %d W \n",P_gA);

printf(" \n    2. Since dynamo B neither delivers power to nor receives power from the bus, ");
printf(" \n        P_B = %d W ",P_LB);
printf(" \n        Power generated by dynamo B,to excite its field , is");
printf(" \n        P_gB = %d W \n ", P_gB);

printf(" \n    3. Power delivered by the bus to dynamo C is ");
printf(" \n        P_LC = %d W ", P_LC);
printf(" \n        while the internal power delivered
```

```
in the direction of rotation");
printf(" \n         of its prime mover to aid rotation
    is \n         P_gC = %d W", P_gC );
```

Scilab code Exa 7.2 calculate all currents and power of the generator

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 7: PARALLEL OPERATION
// Example 7-2

clear; clc; close; // Clear the work space and
    console.

// Given data
R_a = 0.1 ; // Armature resistance in ohm
R_f = 100 ; // Field ckt resistance in ohm
V_L_b = 120 ; // Bus voltage in volt
V_L_a = 140 ; // Voltage of the generator in volt
V_f = V_L_a ; // Voltage across the field in volt

// Calculations
// case a
I_f_a = V_f / R_f ; // Field current in A
I_a_a = I_f_a ; // Armature current in A
E_g_a = V_L_a + I_a_a * R_a ; // Generated EMF in
    volt
P_g_a = E_g_a * I_a_a ; // Generated power in W

// case b
I_a_b = ( E_g_a - V_L_b ) / R_a ; // Armature
    current in A
```

```
I_f_b = V_L_b / R_f ; // Field current in A
I_Lg = I_a_b - I_f_b ; // Generated line current in A
P_L = V_L_b * I_Lg ; // Power generated across the lines in W
E_g_b = V_L_a ;
P_g_b = E_g_b * I_a_b ; // Generated power in W

// Display the results
disp("Example 7-2 Solution : ");
printf(" \n a: Before it is connected to the bus ");
printf(" \n     I_a = I_f = %.1f A \n     E_g = %.2f V \n     P_g = %.1f W \n", I_a_a,E_g_a,P_g_a);

printf(" \n b: After it is connected to the bus ");
printf(" \n     I_a = %.1f A \n     I_f = %.1f A \n     I_Lg = %.1f A \n", I_a_b, I_f_b, I_Lg );
printf(" \n     P_L = %.f W \n     P_g = %.f W ", P_L , P_g_b );
```

Scilab code Exa 7.3 calculate VL IL Pg and PL

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 7: PARALLEL OPERATION
// Example 7-3

clear; clc; close; // Clear the work space and console.

// Given data
R_a = 0.1 ; // Armature resistance in ohm of 3 shunt
```

```
generators
R_a1 =R_a ;
R_a2 =R_a ;
R_a3 =R_a ;
R_L = 2 ; // Load resistance in ohm
E_g1 = 127 ; // Voltage generated by generator 1 in volt
E_g2 = 120 ; // Voltage generated by generator 2 in volt
E_g3 = 119 ; // Voltage generated by generator 3 in volt
// Neglect field currents

// Calculations
// case a
// Terminal bus voltage in volt
V_L = ( (127/0.1) + (120/0.1) + (119/0.1) ) / ( (1/0.1) + (1/0.1) + (1/0.1) + 0.5);

// case b
I_L1 = (E_g1 - V_L)/R_a1 ; // Current delivered by generator 1 in A
I_L2 = (E_g2 - V_L)/R_a2 ; // Current delivered by generator 2 in A
I_L3 = (E_g3 - V_L)/R_a3 ; // Current delivered by generator 3 in A
I_L_2ohm = V_L / R_L ; // Current delivered by 2 ohm load in A

// case c
I_a1 = I_L1 ; // Armature current in A for generator 1
I_a2 = I_L2 ; // Armature current in A for generator 2
I_a3 = I_L3 ; // Armature current in A for generator 3

P_g1 = E_g1 * I_a1 ; // Power generated by generator
```

```
1 in W
P_g2 = E_g2 * I_a2 ; // Power generated by generator 2 in W
P_g3 = E_g3 * I_a3 ; // Power generated by generator 3 in W

// case d
P_L1 = V_L * I_L1 ; // Power delivered to or received from generator 1 in W
P_L2 = V_L * I_L2 ; // Power delivered to or received from generator 2 in W
P_L3 = V_L * I_L3 ; // Power delivered to or received from generator 3 in W
P_L = V_L * -I_L_2ohm ; // Power delivered to or received 2 ohm load in W

// Display the results
disp("Example 7-3 Solution : ");
printf(" \n a: Converting each voltage source to a current source and applying");
printf(" \n    Millman's theorem yields ")
printf(" \n    V_L = %d V \n ", V_L );

printf(" \n b: I_L1 = %d A (to bus)", I_L1 );
printf(" \n    I_L2 = %d A ", I_L2 );
printf(" \n    I_L3 = %d A (from bus)", I_L3 );
printf(" \n    I_L_2ohm = -%d A (from bus) \n", I_L_2ohm );

printf(" \n c: P_g1 = %d W ",P_g1 );
printf(" \n    P_g2 = %d W (floating)",P_g2 );
printf(" \n    P_g3 = %d W \n",P_g3 );

printf(" \n d: P_L1 = %d W ", P_L1);
printf(" \n    P_L2 = %d W ", P_L2 );
printf(" \n    P_L3 = %d W ", P_L3 );
printf(" \n    P_L = %d W ", P_L );
```

Scilab code Exa 7.4 calculate total load and kW output of each G

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 7: PARALLEL OPERATION
// Example 7-4

clear; clc; close; // Clear the work space and console.

// Given data
P1 = 300 ; // Power rating of generator 1 in kW
P2 = 600 ; // Power rating of generator 2 in kW
V = 220 ; // Voltage rating of generator 1 and 2 in volt
V_o = 250 ; // No-load voltage applied to both the generators in volt
// Assume linear characteristics
V_1 = 230 ; // Terminal voltage in volt (case a)
V_2 = 240 ; // Terminal voltage in volt (case b)

// Calculations
// case a
kW1_a = (V_o - V_1)/(V_o - V) * P1 ; // kW carried by generator 1
kW2_a = (V_o - V_1)/(V_o - V) * P2 ; // kW carried by generator 2

// case b
kW1_b = (V_o - V_2)/(V_o - V) * P1 ; // kW carried by generator 1
```

```
kW2_b = (V_o - V_2)/(V_o - V) * P2 ; // kW carried by generator 2

// case c
frac_a = (V_o - V_1)/(V_o - V); // Fraction of rated kW carried by each generator
frac_b = (V_o - V_2)/(V_o - V); // Fraction of rated kW carried by each generator

// Display the results
disp("Example 7-4 Solution : ");
printf(" \n a: At 230 V, using Eq.(7-3) below : ");
printf(" \n    Generator 1 carries = %d kW ", kW1_a );
printf(" \n    Generator 2 carries = %d kW \n", kW2_a );

printf(" \n b: At 240 V, using Eq.(7-3) below : ");
printf(" \n    Generator 1 carries = %d kW ", kW1_b );
printf(" \n    Generator 2 carries = %d kW \n", kW2_b );

printf(" \n c: Both generators carry no-load at 250 V; ");
printf(" \n    %f rated load at %d V; ", frac_b , V_2 );
printf(" \n    %f rated load at %d V; ", frac_a , V_1 );
printf(" \n    and rated load at %d V. ", V );
```

Scilab code Exa 7.5 calculate max and min E and frequency and Epeak and n

```
// Electric Machinery and Transformers
```

```
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 7: PARALLEL OPERATION
// Example 7-5

clear; clc; close; // Clear the work space and console.

// Given data
E_1 = 220 ; // Terminal voltage of alternator 1 in volt
E_2 = 222 ; // Terminal voltage of alternator 2 in volt
f_1 = 60 ; // Frequency of alternator 1 in Hz
f_2 = 59.5 ; // Frequency of alternator 2 in Hz
// Switch is open

// Calculations
// case a
E_max = (E_1 + E_2)/2 ; // Maximum effective voltage across each lamp in volt
E_min = (E_2 - E_1)/2 ; // Minimum effective voltage across each lamp in volt

// case b
f = f_1 - f_2 ; // Frequency in Hz of the voltage across the lamps

// case c
E_peak = E_max / 0.7071 ; // Peak value of the voltage in volt across each lamp

// case d
n = (1/2)*f_1 ; // Number of maximum light pulsations per minute

```

```
// Display the results
disp("Example 7-5 Solution : ");
printf(" \n a: E_max/lamp = %d V (rms)\n ", E_max );
printf(" \n    E_min/lamp = %d V \n ", E_min );
printf(" \n b: f = %.1f Hz \n ", f );
printf(" \n c: E_peak = %.f V \n ", E_peak );
printf(" \n d: n = %d pulsations/min ", n );
```

Scilab code Exa 7.6 calculate max and min E and f and phase relations

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 7: PARALLEL OPERATION
// Example 7-6

clear; clc; close; // Clear the work space and console.

// Given data
E = 220 ; // Voltage generated in volt
E_1 = E ; // Voltage generated by alternator 1 in volt
E_2 = E ; // Voltage generated by alternator 2 in volt
f_1 = 60 ; // Frequency in Hz of alternator 1
f_2 = 58 ; // Frequency in Hz of alternator 2
// Switch is open

// Calculations
// case a
E_max = (E_1 + E_2)/2 ; // Maximum effective voltage across each lamp in volt
```

```
f = f_1 - f_2 ; // Frequency in Hz of the voltage
    across the lamps

// case c
E_min = (E_2 - E_1)/2 ; // Minimum effective voltage
     across each lamp in volt

// Display the results
disp("Example 7-6 Solution : ");
printf(" \n a: E_max/lamp = %d V \n    f = %d Hz \n
    ", E_max, f );
printf(" \n b: The voltages are equal and opposite
    in the local circuit. \n ");
printf(" \n c: E_min/lamp = %d V at zero frequency \
    n ", E_min );
printf(" \n d: The voltages are in phase in the
    local circuit. ");
```

Scilab code Exa 7.7 calculate Is in both alternators

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 7: PARALLEL OPERATION
// Example 7-7

clear; clc; close; // Clear the work space and
    console.

// Given data as per Ex.(7-5)
E1 = 220 ; // Terminal voltage of alternator 1 in
    volt
E2 = 222 ; // Terminal voltage of alternator 2 in
```

```
volt
f1 = 60 ; // Frequency of alternator 1 in Hz
f2 = 59.5 ; // Frequency of alternator 2 in Hz
// Switch is open

// Given data as per Ex.(7-6)
E = 220 ; // Voltage generated in volt
E_1 = E ; // Voltage generated by alternator 1 in volt
E_2 = E ; // Voltage generated by alternator 2 in volt
f_1 = 60 ; // Frequency in Hz of alternator 1
f_2 = 58 ; // Frequency in Hz of alternator 2
// Switch is open

// Given data as per Ex.(7-7)
R_a1 = 0.1 ; // armature resistance of alternator 1 in ohm
R_a2 = 0.1 ; // armature resistance of alternator 2 in ohm
X_a1 = 0.9 ; // armature reactance of alternator 1 in ohm
X_a2 = 0.9 ; // armature reactance of alternator 2 in ohm

Z_1  = R_a1 + %i*X_a1 ; // Effective impedance of alternator 1 in ohm
Z_2  = R_a1 + %i*X_a2 ; // Effective impedance of alternator 2 in ohm
// Switches are closed at the proper instant for paralleling.

// Calculations
// In Ex.7-5,
E_r = E2 - E1 ; // Effective voltage generated in volt
I_s = E_r / (Z_1 + Z_2); // Synchronizing current in the armature in A
```

```
I_s_m = abs(I_s);//I_s_m=magnitude of I_s in A
I_s_a = atan(imag(I_s) /real(I_s))*180/%pi;//I_s_a=phase angle of I_s in degrees

// In Ex.7-6,
Er = E_2 -E_1 ; // Effective voltage generated in volt
Is = Er / ( Z_1 + Z_2); // Synchronizing current in the armature in A

// Display the results
disp("Example 7-7 Solution : ");
printf(" \n In Ex.7-5, ");
printf(" \n E_r = %d V ", E_r);
printf(" \n I_s = ");disp(I_s);
printf(" \n I_s = %.3f <%.2f A ",I_s_m, I_s_a);
printf(" \n where %.3f is magnitude in A and %.2f is phase angle in degrees \n",I_s_m,I_s_a);

printf(" \n In Ex.7-6, ");
printf(" \n E_r = %d V ", Er );
printf(" \n I_s = %d A",Is);
```

Scilab code Exa 7.8 calculate generator and motor action and P loss and terminal V and phasor diagram

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 7: PARALLEL OPERATION
// Example 7-8

clear; clc; close; // Clear the work space and
```

```
console.

// Given data
// EMF's are opposed exactly by 180 degrees
E_gp1 = 200 ; // Terminal voltage of alternator 1 in volt
E_gp2 = 220 ; // Terminal voltage of alternator 2 in volt
R_a1 = 0.2 ; // armature resistance of alternator 1 in ohm
R_a2 = 0.2 ; // armature resistance of alternator 2 in ohm
X_a1 = 2 ; // armature reactance of alternator 1 in ohm
X_a2 = 2 ; // armature reactance of alternator 2 in ohm

Z_p1  = R_a1 + %i*X_a1 ; // Effective impedance of alternator 1 in ohm
Z_p2  = R_a1 + %i*X_a2 ; // Effective impedance of alternator 2 in ohm
// Switches are closed at the proper instant for paralleling.

// Calculations
// case a
E_r = (E_gp2 - E_gp1) ; // Effective voltage generated in volt
I_s = E_r / (Z_p1 + Z_p2); // Synchronizing current in the armature in A
I_s_m = abs(I_s);//I_s_m=magnitude of I_s in A
I_s_a = atan(imag(I_s) /real(I_s))*180/%pi;//I_s_a=phase angle of I_s in degrees

P_2 = E_gp2 * I_s_m * cosd(I_s_a); // Generator action developed by alternator 2 in W

// case b
```

```
theta = I_s_a;
// P_1 = E_gp1 * I_s_m * cosd(180 - theta)
// P_1 = -E_gp1 * I_s_m * cosd(theta),
P_1 = -E_gp1 * I_s_m * cosd(theta); // Synchronizing power received by alternator 1 in W

// case c
// but consider +ve vlaue for P_1 for finding losses , so
P1 = abs(P_1);
losses = P_2 - P1 ; // Power losses in both armatures in W
check = E_r * I_s_m * cosd(I_s_a); // Verifying losses by Eq.7-7
double_check = (I_s_m)^2 * (R_a1 + R_a2); // Verifying losses by Eq.7-7

// case d
V_p2  = E_gp2 - I_s*Z_p1 ; // Generator action
V_p1  = E_gp1 + I_s*Z_p1 ; // Motor action

// Display the results
disp("Example 7-8 Solution : ");
printf(" \n a: E_r = %d V ",E_r);
printf(" \n    I_s = %.2f <%.2f A ", I_s_m, I_s_a );
printf(" \n    P_2 = %.1f W (total power delivered by alternator 2 ) \n", P_2);

printf(" \n b: P_1 = %f W (synchronizing power received by alternator 1)",P_1);
printf(" \n    Note:Scilab considers phase angle of I_s as %f instead ",I_s_a);
printf(" \n          of -84.3 degrees,so slight variation in the answer P_1.\n");

printf(" \n c: Consider +ve value of P_1 for calculating losses");
printf(" \n    Losses: P_2 - P_1 = %.1f W ",losses )
```

```
;
printf(" \n     Check: E_a*I_s*cos(theta) = %.1f W ",
   check );
printf(" \n     Double check : (I_s)^2*(R_a1+R_a2) =
   %.1f W as given in Eq.(7-1)",double_check );

printf("\n\n d: From Fig.7-14, V_p2, the terminal
   phase voltage of ");
printf(" \n     alternator 2, is, from Eq.(7-1)");
printf(" \n     V_p2 = %d V (generator action)\n\n
      From section 7-2.1 ",V_p2);
printf(" \n     V_p1 = %d V ( motor action)\n",V_p1);

printf(" \n e: The phasor diagram is shown in Fig
   .7-14.");
```

Scilab code Exa 7.9 calculate synchronizing I and P and P losses

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 7: PARALLEL OPERATION
// Example 7-9

clear; clc; close; // Clear the work space and
   console.

// Given data
E_2_mag = 230 ; // Magnitude of voltage generated by
    alternator 2 in volt
E_1_mag = 230 ; // Magnitude of voltage generated by
    alternator 1 in volt

```

```
theta_2 = 180 ; // Phase angle of generated voltage by alternator 2 in degrees
theta_1 = 20 ; // Phase angle of generated voltage by alternator 1 in degrees

R_a1 = 0.2 ; // armature resistance of alternator 1 in ohm
R_a2 = 0.2 ; // armature resistance of alternator 2 in ohm

// writing given voltage in exponential form as follows
// %pi/180 for degrees to radians conversion
E_2 = E_2_mag * expm(%i * theta_2*(%pi/180) ); // voltage generated by alternator 2 in volt
E_1 = E_1_mag * expm(%i * theta_1*(%pi/180) ); // voltage generated by alternator 1 in volt

// writing given impedance(in ohm)in exponential form as follows
Z_1 = 2.01 * expm(%i * 84.3*(%pi/180) ); // %pi/180 for degrees to radians conversion
Z_2 = Z_1 ;
Z_1_a = atan(imag(Z_1) /real(Z_1))*180/%pi;//Z_1_a= phase angle of Z_1 in degrees

// Calculations
E_r = E_2 + E_1 ; // Total voltage generated by Alternator 1 and 2 in volt
E_r_m = abs(E_r);//E_r_m=magnitude of E_r in volt
E_r_a = atan(imag(E_r) /real(E_r))*180/%pi;//E_r_a= phase angle of E_r in degrees

// case a
I_s = E_r / (Z_1 + Z_2); // Synchronozing current in A
I_s_m = abs(I_s);//I_s_m=magnitude of I_s in A
I_s_a = atan(imag(I_s) /real(I_s))*180/%pi;//I_s_a=
```

```
phase angle of I_s in degrees

// case b
E_gp1 = E_1_mag;
P_1 = E_gp1 * I_s_m * cosd(I_s_a - theta_1); // Synchronozing power developed by alternator 1 in W

// case c
E_gp2 = E_2_mag;
P_2 = E_gp2 * I_s_m * cosd(I_s_a - theta_2); // Synchronozing power developed by alternator 2 in W

// case d
// but consider +ve vlaue for P_2 for finding losses , so
P2 = abs(P_2);
losses = P_1 - P2 ; // Losses in the armature in W

// E_r_a yields -80 degrees which is equivalent to 100 degrees , so
theta = 100 - I_s_a ; // Phase difference between E_r and I_a in degrees

check = E_r_m * I_s_m * cosd(theta); // Verifying losses by Eq.7-7
R_aT = R_a1 + R_a2 ; // total armature resistance of alternator 1 and 2 in ohm
double_check = (I_s_m)^2 * (R_aT); // Verifying losses by Eq.7-7

// Display the results
disp("Example 7-9 Solution : ");
printf(" \n a: I_s = ");disp(I_s);
printf(" \n    I_s = %.2f <%.2f A \n ",I_s_m, I_s_a);

```

```
printf(" \n b: P_1 = %.f W (power delivered to bus)"
    ,P_1);
printf(" \n    Slight variation in P_1 is due slight
      variations in ")
printf(" \n    magnitude of I_s,& angle btw (E_gp1,
    I_s)\n")
printf(" \n    P_2 = %.f W (power received from bus)
    \n",P_2);

printf(" \n c: Losses: P_1 - P_2 = %d",losses);
printf(" \n    Check: E_a*I_s*cos(theta) = %d W ",
    check );
printf(" \n    Double check : (I_s)^2*(R_a1+R_a2) =
    %d W ",double_check );
```

Scilab code Exa 7.10 calculate synchronizing I and P and P losses

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 7: PARALLEL OPERATION
// Example 7-10

clear; clc; close; // Clear the work space and
    console.

// Given data
E_2_mag = 230 ; // Magnitude of voltage generated by
      alternator 2 in volt
E_1_mag = 230 ; // Magnitude of voltage generated by
      alternator 1 in volt

theta_2 = 180 ; // Phase angle of generated voltage
```

```
    by alternator 2 in degrees
theta_1 = 20 ; // Phase angle of generated voltage by alternator 1 in degrees

// writing given voltage in exponential form as follows
// %pi/180 for degrees to radians conversion
E_2 = E_2_mag * expm(%i * theta_2*(%pi/180) ); // voltage generated by alternator 2 in volt
E_1 = E_1_mag * expm(%i * theta_1*(%pi/180) ); // voltage generated by alternator 1 in volt

// writing given impedance(in ohm)in exponential form as follows
Z_1 = 6 * expm(%i * 50*(%pi/180) ); // %pi/180 for degrees to radians conversion
Z_2 = Z_1 ;
Z_1_a = atan(imag(Z_1) /real(Z_1))*180/%pi;//Z_1_a= phase angle of Z_1 in degrees

// Calculations
E_r = E_2 + E_1 ; // Total voltage generated by Alternator 1 and 2 in volt
E_r_m = abs(E_r);//E_r_m=magnitude of E_r in volt
E_r_a = atan(imag(E_r) /real(E_r))*180/%pi;//E_r_a= phase angle of E_r in degrees

// case a
I_s = E_r / (Z_1 + Z_2); // Synchronozing current in A
I_s_m = abs(I_s);//I_s_m=magnitude of I_s in A
I_s_a = atan(imag(I_s) /real(I_s))*180/%pi;//I_s_a= phase angle of I_s in degrees

// case b
E_gp1 = E_1_mag;
P_1 = E_gp1 * I_s_m * cosd(I_s_a - theta_1); // Synchronozing power developed by alternator 1 in
```

```
W

// case c
E_gp2 = E_2_mag;
P_2 = E_gp2 * I_s_m * cosd(I_s_a - theta_2); // Synchronozing power developed by alternator 2 in W

// case d
// but consider +ve vlaue for P_2 for finding losses , so
P2 = abs(P_2);
losses = P_1 - P2 ; // Losses in the armature in W

// E_r_a yields -80 degrees which is equivalent to 100 degrees , so
theta = 100 - I_s_a ; // Phase difference between E_r and I_s in degrees

check = E_r_m * I_s_m * cosd(theta); // Verifying losses by Eq.7-7
R_aT = 12*cosd(50) ; // total armature resistance of alternator 1 and 2 in ohm
double_check = (I_s_m)^2 * (R_aT); // Verifying losses by Eq.7-7

// Display the results
disp("Example 7-10 Solution : ");
printf(" \n a: I_s = ");disp(I_s);
printf(" \n    I_s = %.2f <%.2f A \n ",I_s_m, I_s_a);

printf(" \n b: P_1 = %.f W (power delivered to bus)",P_1);
printf(" \n    Note:Slight variation in P_1 is due slight variations in ")
printf(" \n          phase angle of I_s,& angle btw (E_gp1,I_s)\n")
```

```
printf(" \n     P_2 = %.f W (power received from bus)
   \n",P_2);

printf(" \n c: Losses: P_1 - P_2 = %.f W",losses);
printf(" \n     Check: E_a*I_s*cos(theta) = %.f W ",
   check );
printf(" \n     Double check : (I_s)^2*(R_a1+R_a2) =
   %.f W ",double_check );
```

Scilab code Exa 7.11 calculate mesh currents line currents phase voltages phasor diagram

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 7: PARALLEL OPERATION
// Example 7-11

clear; clc; close; // Clear the work space and
   console.

// Given data
// writing supply voltage in exponential form as
   follows
// %pi/180 for degrees to radians conversion
V_AB = 100 * expm(%i * 0*(%pi/180) ); // voltage
   supplied across A & B in volt
V_BC = 100 * expm(%i * -120*(%pi/180) ); // voltage
   supplied across B & C in volt
V_CA = 100 * expm(%i * 120*(%pi/180) ); // voltage
   supplied across C & A in volt

disp("Example 7-11 : ");
```

```
printf("\n Writing two mesh equations for I_1 and I_2 in fig.7-23a yields following\n array :");
printf(" \n  I_1 \t\t I_2 \t\t V ");
printf(" \n --------------------------------------------");
printf(" \n  6 + j0 \t -3 + j0 \t 100 + j0 ");
printf(" \n -3 + j0 \t 3 - j4 \t -50 - j86.6 ");

// Calculations
A = [ (6+%i*0) (-3+%i*0) ; (-3+%i*0) (3-%i*4) ]; // Matrix containing above mesh eqns array
delta = det(A); // Determinant of A

// case a
I_1 = det( [ (100+%i*0) (-3+%i*0) ; (-50-%i*86.60) (3-%i*4) ] ) / delta ;
// Mesh current I_1 in A
I_1_m = abs(I_1);//I_1_m=magnitude of I_1 in A
I_1_a = atan(imag(I_1) /real(I_1))*180/%pi;//I_1_a=phase angle of I_1 in degrees

I_2 = det( [ (6+%i*0) (100+%i*0) ; (-3+%i*0) (-50-%i*86.6) ] ) / delta ;
// Mesh current I_2 in A
I_2_m = abs(I_2);//I_2_m=magnitude of I_2 in A
I_2_a = atan(imag(I_2) /real(I_2))*180/%pi;//I_2_a=phase angle of I_2 in degrees

// case b
I_A = I_1 ; //  Line current I_A in A
I_A_m = abs(I_A);//I_A_m=magnitude of I_A in A
I_A_a = atan(imag(I_A) /real(I_A))*180/%pi;//I_A_a=phase angle of I_A in degrees

I_B = I_2 - I_1 ; //  Line current I_B in A
I_B_m = abs(I_B);//I_B_m=magnitude of I_B in A
I_B_a = atan(imag(I_B) /real(I_B))*180/%pi - 180;//I_B_a=phase angle of I_B in degrees
```

```
I_C = -I_2 ; //  Line current I_C in A
I_C_m = abs(I_C);//I_C_m=magnitude of I_C in A
I_C_a = 180 + atan(imag(I_C) /real(I_C))*180/%pi;//I_C_a=phase angle of I_C in degrees

// case c
Z_A = 3 * expm(%i * 0*(%pi/180) ); // Impedance in line A in ohm
Z_B = 3 * expm(%i * 0*(%pi/180) ); // Impedance in line B in ohm
Z_C = 4 * expm(%i * -90*(%pi/180) ); // Impedance in line C in ohm

V_AO = I_A * Z_A ; // Phase voltage V_AO in volt
V_AO_m = abs(V_AO);//V_AO_m=magnitude of V_AO in volt
V_AO_a = atan(imag(V_AO) /real(V_AO))*180/%pi;//V_AO_a=phase angle of V_AO in degrees

V_BO = I_B * Z_B ; // Phase voltage V_BO in volt
V_BO_m = abs(V_BO);//V_BO_m=magnitude of V_BO in volt
V_BO_a = atan(imag(V_BO) /real(V_BO))*180/%pi - 180;//V_BO_a=phase angle of V_BO in degrees

V_CO = I_C * Z_C ; // Phase voltage V_CO in volt
V_CO_m = abs(V_CO);//V_CO_m=magnitude of V_CO in volt
V_CO_a = atan(imag(V_CO) /real(V_CO))*180/%pi;//V_CO_a=phase angle of V_CO in degrees

// Display the results
disp("Solution : ");
printf(" \n a: I_1 in A = ");disp(I_1);
printf(" \n    I_1 = %.2f <%.2f A \n ",I_1_m, I_1_a);
printf(" \n    I_2 in A = ");disp(I_2);
```

```
printf(" \n     I_2 = %.2f <%.2f A\n ",I_2_m, I_2_a );

printf(" \n b: I_A in A = ");disp(I_1);
printf(" \n     I_A = %.2f <%.2f A\n",I_A_m, I_A_a );

printf(" \n     I_B in A = ");disp(I_B);
printf(" \n     I_B = %.2f <%.2f A\n",I_B_m, I_B_a );

printf(" \n     I_C in A = ");disp(I_C);
printf(" \n     I_C = %.2f <%.2f A \n",I_C_m,  I_C_a );

printf(" \n c: V_AO = %.2f <%.2f V",V_AO_m, V_AO_a );
printf(" \n     V_BO = %.2f <%.2f V",V_BO_m, V_BO_a );
printf(" \n     V_CO = %.2f <%.2f V\n",V_CO_m, V_CO_a );

printf(" \n d: The phasor diagram is shown in Fig.7-23b, with the phase voltages");
printf(" \n     inscribed inside the (equilateral) triangle of given line voltages");
```

Chapter 8

AC DYNAMO TORQUE RELATIONS SYNCHRONOUS MOTORS

Scilab code Exa 8.1 calculate alpha Er Ia Pp Pt Power loss Pd

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 8: AC DYNAMO TORQUE RELATIONS -
   SYNCHRONOUS MOTORS
// Example 8-1

clear; clc; close; // Clear the work space and
   console.

// Given data
// 3- phase Y-connected synchronous motor
P = 20 ; // No. of poles
hp = 40 ; // power rating of the synchronous motor
   in hp
```

```
V_L = 660 ; // Line voltage in volt
beta = 0.5 ; // At no-load , the rotor is retarded 0.5 mechanical degree from
// its synchronous position.
X_s = 10 ; // Synchronous reactance in ohm
R_a = 1.0 ; // Effective armature resistance in ohm

// Calculations
// case a
funcprot(0); // To avoid this message "Warning : redefining function: beta"
alpha = P * (beta/2); // The rotor shift from the synchronous position in
// electrical degrees.

// case b
V_p = V_L / sqrt(3); // Phase voltage in volt
E_gp = V_p ; // Generated voltage/phase at no-load in volt (given)
E_r = (V_p - E_gp*cosd(alpha)) + %i*(E_gp*sind(alpha));
 // Resultant emf across the armature per phase in V/phase
E_r_m = abs(E_r);//E_r_m=magnitude of E_r in volt
E_r_a = atan(imag(E_r) /real(E_r))*180/%pi;//E_r_a=phase angle of E_r in degrees

// case c
Z_s = R_a + %i*X_s ; // Synchronous impedance in ohm
Z_s_m = abs(Z_s);//Z_s_m=magnitude of Z_s in ohm
Z_s_a = atan(imag(Z_s) /real(Z_s))*180/%pi;//Z_s_a=phase angle of Z_s in degrees

I_a  = E_r / Z_s ; // Armature current/phase in A/phase
I_a_m = abs(I_a);//I_a_m=magnitude of I_a in A
I_a_a = atan(imag(I_a) /real(I_a))*180/%pi;//I_a_a=phase angle of I_a in degrees
```

```
// case d
theta = I_a_a ; // Phase angle between V_p and I_a in degrees
P_p = V_p * I_a_m * cosd(theta); // Power per phase drawn by the motor from the bus
P_t = 3*P_p ; // Total power drawn by the motor from the bus

// csae e
P_a = 3 * (I_a_m)^2 * R_a ; // Armature power loss at no-load in W
P_d = (P_t - P_a)/746 ; // Internal developed horsepower at no-load

// Display the results
disp("Example 8-1 Solution : ");
printf(" \n a: alpha = %d degrees (electrical degrees)\n",alpha );

printf(" \n b: E_gp = %d V also, as given ",E_gp);
printf(" \n    E_r in V/phase = ");disp(E_r);
printf(" \n    E_r = %.1f <%.1f V/phase \n",E_r_m, E_r_a );

printf(" \n c: Z_s in ohm/phase = ");disp(Z_s);
printf(" \n    Z_s = %.2f <%.1f ohm/phase \n",Z_s_m, Z_s_a );
printf(" \n    I_a in A/phase = ");disp(I_a);
printf(" \n    I_a = %.2f <%.2f A/phase \n ",I_a_m, I_a_a);

printf(" \n d: P_p = %.2f W/phase ",P_p );
printf(" \n    P_t = %.2f W ",P_t);
printf(" \n    Note: Slight variations in power values is due to slight variations");
printf(" \n          in V_p , I_a and theta values from those of the textbook\n");
```

```
printf(" \n e: P_a = %.f W ",P_a );
printf(" \n    P_d = %d hp ", P_d );
```

Scilab code Exa 8.2 calculate alpha Er Ia Pp Pt Power loss Pd

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 8: AC DYNAMO TORQUE RELATIONS − SYNCHRONOUS MOTORS
// Example 8−2

clear; clc; close; // Clear the work space and console.

// Given data
// 3− phase Y−connected synchronous motor
P = 20 ; // No. of poles
hp = 40 ; // power rating of the synchronous motor in hp
V_L = 660 ; // Line voltage in volt
beta = 5 ; // At no−load, the rotor is retarded 0.5 mechanical degree from
// its synchronous position.
X_s = 10 ; // Synchronous reactance in ohm
R_a = 1.0 ; // Effective armature resistance in ohm

// Calculations
// case a
funcprot(0); // To avoid this message "Warning : redefining function: beta"
alpha = P * (beta/2); // The rotor shift from the
```

```
synchronous position in
// electrical degrees.

// case b
V_p = V_L / sqrt(3); // Phase voltage in volt
E_gp = V_p ; // Generated voltage/phase at no-load in volt (given)
E_r = (V_p - E_gp*cosd(alpha)) + %i*(E_gp*sind(alpha));
E_r_m = abs(E_r);//E_r_m=magnitude of E_r in volt
E_r_a = atan(imag(E_r) /real(E_r))*180/%pi;//E_r_a=phase angle of E_r in degrees

// case c
Z_s = R_a + %i*X_s ; // Synchronous impedance in ohm
Z_s_m = abs(Z_s);//Z_s_m=magnitude of Z_s in ohm
Z_s_a = atan(imag(Z_s) /real(Z_s))*180/%pi;//Z_s_a=phase angle of Z_s in degrees

I_a  = E_r / Z_s ; // Armature current/phase in A/phase
I_a_m = abs(I_a);//I_a_m=magnitude of I_a in A
I_a_a = atan(imag(I_a) /real(I_a))*180/%pi;//I_a_a=phase angle of I_a in degrees

// case d
theta = I_a_a ; // Phase angle between V_p and I_a in degrees
P_p = V_p * I_a_m * cosd(theta); // Power per phase drawn by the motor from the bus
P_t = 3*P_p ; // Total power drawn by the motor from the bus

// csae e
P_a = 3 * (I_a_m)^2 * R_a ; // Armature power loss at no-load in W
P_d = (P_t - P_a)/746 ; // Internal developed horsepower at no-load
```

```
// Display the results
disp("Example 8-2 Solution : ");
printf(" \n a: alpha = %d degrees (electrical
    degrees)\n",alpha );

printf(" \n b: E_gp = %d V also , as given ",E_gp);
printf(" \n    E_r in V/phase = ");disp(E_r);
printf(" \n    E_r = %d <%.1f V/phase \n",E_r_m,
    E_r_a );

printf(" \n c: Z_s in ohm/phase = ");disp(Z_s);
printf(" \n    Z_s = %.2f <%.1f ohm/phase \n",Z_s_m,
     Z_s_a );
printf(" \n    I_a in A/phase = ");disp(I_a);
printf(" \n    I_a = %.2f <%.2f A/phase \n ",I_a_m,
    I_a_a);

printf(" \n d: P_p = %.2f W/phase ",P_p );
printf(" \n    P_t = %.2f W ",P_t);
printf(" \n    Note: Slight variations in power
    values is due to slight variations");
printf(" \n          in V_p , I_a and theta values
    from those of the textbook\n");


printf(" \n e: P_a = %.f W ",P_a );
printf(" \n    P_d = %.1f hp ", P_d );
```

Scilab code Exa 8.3 calculate Ia PF hp

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom
```

```
// Chapter 8: AC DYNAMO TORQUE RELATIONS − SYNCHRONOUS MOTORS
// Example 8−3

clear; clc; close; // Clear the work space and console.

// Given data
// 3− phase Y−connected synchronous motor
P = 6 ; // No. of poles
hp = 50 ; // power rating of the synchronous motor in hp
V_L = 440 ; // Line voltage in volt
X_s = 2.4 ; // Synchronous reactance in ohm
R_a = 0.1 ; // Effective armature resistance in ohm
alpha = 20 ; // The rotor shift from the synchronous position in
// electrical degrees.
E_gp_a = 240 ; // Generated voltage/phase in volt when the motor is under−excited(case a)
E_gp_b = 265 ; // Generated voltage/phase in volt when the motor is under−excited(case b)
E_gp_c = 290 ; // Generated voltage/phase in volt when the motor is under−excited(case c)

// Calculations
V_p = V_L / sqrt(3); // Phase voltage in volt
// case a
E_ra = (V_p - E_gp_a * cosd(alpha)) + %i*(E_gp_a * sind(alpha));
E_ra_m = abs(E_ra);//E_ra_m=magnitude of E_ra in volt
E_ra_a = atan(imag(E_ra) /real(E_ra))*180/%pi;//E_ra_a=phase angle of E_ra in degrees

Z_s = R_a + %i*X_s ; // Synchronous impedance in ohm

```

```
I_ap1  = E_ra / Z_s ; // Armature current/phase in A/phase
I_ap1_m = abs(I_ap1);//I_ap1_m=magnitude of I_ap1 in A
I_ap1_a = atan(imag(I_ap1) /real(I_ap1))*180/%pi;//I_ap1_a=phase angle of I_ap1 in degrees

cos_theta_a = cosd(I_ap1_a); // Power factor
Ia_m1 = abs(I_ap1_m); // Absoulte value of magnitude of I_ap1

P_d1 = 3 * (E_gp_a*Ia_m1) * cosd(160 - I_ap1_a); // // Internal developed power in W
// 160 + I_ap1_a is the angle between E_gp_a and I_ap1
Pd1 = abs(P_d1); // Consider absolute value of power in W for calculating hp

Horse_power1 = Pd1 / 746 ; // Horsepower developed by the armature in hp

// case b
E_rb = (V_p - E_gp_b * cosd(alpha)) + %i*(E_gp_b * sind(alpha));
E_rb_m = abs(E_rb);//E_rb_m=magnitude of E_rb in volt
E_rb_a = atan(imag(E_rb) /real(E_rb))*180/%pi;//E_rb_a=phase angle of E_rb in degrees

I_ap2  = E_rb / Z_s ; // Armature current/phase in A/phase
I_ap2_m = abs(I_ap2);//I_ap2_m=magnitude of I_ap2 in A
I_ap2_a = atan(imag(I_ap2) /real(I_ap2))*180/%pi;//I_ap2_a=phase angle of I_ap2 in degrees

cos_theta_b = cosd(I_ap2_a); // Power factor
Ia_m2 = abs(I_ap2_m); // Absoulte value of magnitude
```

```
 of I_ap2

P_d2 = 3 * (E_gp_b*Ia_m2) * cosd(160 - I_ap2_a); // // Internal developed power in W
// 160 + I_ap2_a is the angle between E_gp_b  and I_ap2
Pd2 = abs(P_d2); // Consider absolute value of power in W for calculating hp

Horse_power2 = Pd2 / 746 ; // Horsepower developed by the armature in hp

// case c
E_rc = (V_p - E_gp_c * cosd(alpha)) + %i*(E_gp_c * sind(alpha));
E_rc_m = abs(E_rc);//E_rc_m=magnitude of E_rc in volt
E_rc_a = atan(imag(E_rc) /real(E_rc))*180/%pi;//E_rc_a=phase angle of E_rc in degrees

I_ap3  = E_rc / Z_s ; // Armature current/phase in A/phase
I_ap3_m = abs(I_ap3);//I_ap3_m=magnitude of I_ap3 in A
I_ap3_a = atan(imag(I_ap3) /real(I_ap3))*180/%pi;//I_ap3_a=phase angle of I_ap3 in degrees

cos_theta_c = cosd(I_ap3_a); // Power factor
Ia_m3 = abs(I_ap3_m); // Absoulte value of magnitude of I_ap3

P_d3 = 3 * (E_gp_c*Ia_m3) * cosd(160 - I_ap3_a); // // Internal developed power in W
// 160 + I_ap3_a is the angle between E_gp_c  and I_ap3
Pd3 = abs(P_d3); // Consider absolute value of power in W for calculating hp

```

```
Horse_power3 = Pd3 / 746 ; // Horsepower developed by the armature in hp

// Display the results
disp("Example 8-3 Solution : ");
disp("Slight variations in power values are because of non-approximation of I_a & cos(E_gp,I_a) values during power calculations in scilab ")
printf(" \n a: V_p = %.f <0 V \n ",V_p);
printf(" \n    E_r in V = ");disp(E_ra);
printf(" \n    E_r = %.2f <%.2f V \n ",E_ra_m,E_ra_a);
printf(" \n    I_ap in A = ");disp(I_ap1);
printf(" \n    I_ap = %.2f <%.2f A \n", I_ap1_m , I_ap1_a );
printf(" \n    cos(theta) = %.4f lagging \n ", cos_theta_a );
printf(" \n    P_d = %d W drawn from bus(motor operation)\n", P_d1 );
printf(" \n    Horsepower = %.1f hp \n\n", Horse_power1 );

printf(" \n b: E_r in V = ");disp(E_rb);
printf(" \n    E_r = %.2f <%.2f V \n ",E_rb_m,E_rb_a);
printf(" \n    I_ap in A = ");disp(I_ap2);
printf(" \n    I_ap = %.2f <%.2f A \n", I_ap2_m , I_ap2_a );
printf(" \n    cos(theta) = %.4f = %.f(unity PF) \n ", cos_theta_b, cos_theta_b );
printf(" \n    P_d = %d W drawn from bus(motor operation)\n", P_d2 );
printf(" \n    Horsepower = %.1f hp \n\n", Horse_power2 );

printf(" \n c: E_r in V = ");disp(E_rc);
printf(" \n    E_r = %.2f <%.2f V \n ",E_rc_m,E_rc_a);
```

```
printf(" \n     I_ap in A = ");disp(I_ap3);
printf(" \n     I_ap = %.2f <%.2f A \n", I_ap3_m ,
    I_ap3_a );
printf(" \n     cos(theta) = %.4f leading \n ",
    cos_theta_c );
printf(" \n     P_d = %d W drawn from bus(motor
    operation)\n", P_d3 );
printf(" \n     Horsepower = %.1f hp \n\n",
    Horse_power3 );
```

Scilab code Exa 8.4 calculate IL Iap Zp IaZp theta deba Egp

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 8: AC DYNAMO TORQUE RELATIONS -
    SYNCHRONOUS MOTORS
// Example 8-4

clear; clc; close; // Clear the work space and
    console.

// Given data
// Y-connected synchronous dynamo
P = 2 ; // No. of poles
hp = 1000 ; // power rating of the synchronous motor
     in hp
V_L = 6000 ; // Line voltage in volt
f = 60 ; // Frequency in Hz
R_a = 0.52 ; // Effective armature resistance in ohm
X_s = 4.2 ; // Synchronous reactance in ohm
P_t = 811 ; // Input power in kW
PF = 0.8 ; // Power factor leading
```

```

// Calculations
V_p = V_L / sqrt(3); // Phase voltage in volt

// case a
cos_theta = PF ; // Power factor leading
I_L = (P_t*1000) / ( sqrt(3) * V_L * cos_theta); // Line armature current in A
I_ap = I_L ; // Phase armature current in A

// case b
Z_p = R_a + %i * X_s ; // Impedance per phase in ohm
Z_p_m = abs(Z_p);//Z_p_m=magnitude of Z_p in ohm
Z_p_a = atan(imag(Z_p) /real(Z_p))*180/%pi;//Z_p_a= phase angle of Z_p in degrees

// case c
Ia_Zp = I_L * Z_p_m ;
E_r = Ia_Zp ;

// case d
theta = acosd(0.8); // Power factor angle in degrees

// case e
funcprot(0); // Use  to avoid this message "Warning : redefining function: beta"                .
beta = Z_p_a ; //
deba = beta + theta // Difference angle at 0.8 leading PF in degrees

// case f
//  Generated voltage/phase in volt
E_gp_f = sqrt( (E_r)^2 + (V_p)^2 - 2*E_r*V_p*cosd(deba) );

// case g
//  Generated voltage/phase in volt
E_gp_g = ( V_p + Ia_Zp * cosd(180-deba) ) + %i * (
```

```
Ia_Zp * sind(180-deba) );
E_gp_g_m = abs(E_gp_g);//E_gp_g_m=magnitude of E_gp_g in volt
E_gp_g_a = atan(imag(E_gp_g) /real(E_gp_g))*180/%pi; //E_gp_g_a=phase angle of E_gp_g in degrees

// case h
IaZp = Ia_Zp * expm(%i * Z_p_a * (%pi/180) ); // voltage generated by alternator 1 in volt
IaZp_m = abs(IaZp);//IaZp_m=magnitude of IaZp in A
IaZp_a = atan(imag(IaZp) /real(IaZp))*180/%pi;// IaZp_a=phase angle of IaZp in degrees
IaRa = IaZp_m*cosd(IaZp_a); // Real part of IaZp
IaXs = IaZp_m*sind(IaZp_a); // Imaginery part of IaZp

cos_theta = PF ; //
sin_theta = sqrt( 1 - (cos_theta)^2 );
//  Generated voltage/phase in volt
E_gp_h = ( V_p * cos_theta - IaRa ) + %i * ( V_p * sin_theta + IaXs);
E_gp_h_m = abs(E_gp_h);//E_gp_h_m=magnitude of E_gp_h in volt
E_gp_h_a = atan(imag(E_gp_h) /real(E_gp_h))*180/%pi; //E_gp_h_a=phase angle of E_gp_h in degrees

// Display the results
disp("Example 8-4 Solution : ");
printf(" \n a: I_L = %.2f \n    I_ap = %.2f A \n", I_L, I_ap );

printf(" \n b: Z_p in ohm = ");disp(Z_p);
printf(" \n    Z_p = %.3f <%.2f ohm \n ", Z_p_m , Z_p_a );

printf(" \n c: IaZp = %.1f V \n    E_r = %.1f V \n " ,Ia_Zp , E_r );

```

```
printf(" \n d: Power factor angle,\n    theta = %.2f degrees leading \n ", theta );

printf(" \n e: Difference angle,\n    deba = %.2f degrees \n ", deba );

printf(" \n f: E_gp = %.f V \n ", E_gp_f );

printf(" \n g: E_gp in V = ");disp(E_gp_g );
printf(" \n    E_gp = %d <%.2f V \n",E_gp_g_m , E_gp_g_a );

printf(" \n h: E_gp in V = ");disp(E_gp_h);
printf(" \n    E_gp = %.f <%.2f V",E_gp_h_m, E_gp_h_a );
```

Scilab code Exa 8.5 calculate torque angle

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 8: AC DYNAMO TORQUE RELATIONS − SYNCHRONOUS MOTORS
// Example 8−5

clear; clc; close; // Clear the work space and console.

// Given data
// Y−connected synchronous dynamo
P = 2 ; // No. of poles
hp = 1000 ; // power rating of the synchronous motor in hp
```

```
V_L = 6000 ; // Line voltage in volt
f = 60 ; // Frequency in Hz
R_a = 0.52 ; // Effective armature resistance in ohm
X_s = 4.2 ; // Synchronous reactance in ohm
P_t = 811 ; // Input power in kW
PF = 0.8 ; // Power factor leading

// Calculated values
E_gp = 3687 ; // Generated voltage/phase in volt
V_p = V_L / sqrt(3); // Phase voltage in volt
E_r = 412.8 ; // Resultant EMF across armature/phase
    in volt
deba = 119.81 ; // Difference angle at 0.8 leading
   PF in degrees
theta = 36.87 ; // Power factor angle in degrees
IaXs = 409.7 ; // Voltage drop across synchronous
   reactance in volt
IaRa = 50.74 ; // Voltage drop across armature
   resistance in volt

// Calculations

// Torque angle alpha in degrees calculated by
   different Eqns
// case a
alpha1 = acosd( ( E_gp^2 + V_p^2 - E_r^2 ) / ( 2*
   E_gp*V_p ) ); // Eq.8-12

// case b
alpha2 = asind( ( E_r * sind(deba) ) / ( E_gp ) );
   // Eq.8-13

// case c
alpha3 = theta - atand( (V_p*sind(theta) + IaXs) / (
   V_p*cosd(theta) - IaRa) );// Eq.8-14

// Display the results
disp("Example 8-5 Solution : ");
```

```
printf(" \n a: Using Eq.(8-12) \n    alpha = %.2f
   degrees \n ", alpha1 );

printf(" \n b: Using Eq.(8-13) \n    alpha = %.2f
   degrees \n ", alpha2 );

printf(" \n c: Using Eq.(8-14) \n    alpha = %.2f
   degrees \n ", alpha3 );
printf(" \n    Slight variation in case c alpha is
   due to tan inverse value ");
printf(" \n    which was calulated to be 42.445604
   degrees, instead of 42.44 degrees(textbook).")
```

Scilab code Exa 8.6 calculate Pp Pt hp internal and external torque and motor efficiency

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 8: AC DYNAMO TORQUE RELATIONS -
   SYNCHRONOUS MOTORS
// Example 8-6

clear; clc; close; // Clear the work space and
   console.

// Given data as per Example 8-4
// Y-connected synchronous dynamo
P = 2 ; // No. of poles
hp = 1000 ; // power rating of the synchronous motor
    in hp
V_L = 6000 ; // Line voltage in volt
f = 60 ; // Frequency in Hz
```

```
R_a = 0.52 ; // Effective armature resistance in ohm
X_s = 4.2 ; // Synchronous reactance in ohm
P_t = 811 ; // Input power in kW
PF = 0.8 ; // Power factor leading

// Calculated values from Example 8-4
E_gp = 3687 ; // Generated voltage/phase in volt

I_a = 97.55 ; // Phase armature current in A

phi = (42.45 - 0); // Phase angle between E_gp and I_a in degrees
// where 42.45 and 0 are phase angles of E_gp and I_a in degrees respectively.

// Calculations
// case a
P_p = E_gp * I_a * cosd(phi) / 1000; // Mechanical power developed per phase in kW

P_t_a = 3 * P_p ; // Total mechanical power developed in kW

// case b
P_t_b = P_t_a / 0.746 ; // Internal power developed in hp at rated load

// case c
S = 120 * f / P ; // Speed of the motor in rpm
T_int = ( P_t_b * 5252 ) / S ; // Internal torque developed in lb-ft

// case d
T_ext = ( hp * 5252 ) / 3600 ; // External torque developed in lb-ft
eta = (T_ext / T_int) * 100 ; // Motor efficiency in percent

```

```
// Display the results
disp("Example 8-6 Solution : ");
printf(" \n a: Similar to a dc motor, the mechanical
    power developed in the armature");
printf(" \n    is the product of the induced EMF per
    phase, the armature current");
printf(" \n    per phase, and the cosine of the
   angle between them.\n");
printf(" \n    P_p = %.3f kW \n     P_t = %.1f kW \n"
   , P_p, P_t_a );

printf(" \n b: P_t = %.1f hp \n ", P_t_b );

printf(" \n c: T_int = %.f lb-ft \n ", T_int );

printf(" \n d: T_ext = %d lb-ft  \n", T_ext );
printf(" \n    Motor Efficiency,\n    eta = %.1f
   percent ", eta );
```

Scilab code Exa 8.7 calculate total load I and PF using IM and SM percent reduction in I and overall PF

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 8: AC DYNAMO TORQUE RELATIONS -
  SYNCHRONOUS MOTORS
// Example 8-7

clear; clc; close; // Clear the work space and
   console.

// Given data
```

```
P_o = 2000 ; // Total power consumed by a factory in kW from the transformer
cos_theta = 0.6 ; // 0.6 lagging power factor at which power is consumed -
// - from the transformer
sin_theta = sqrt(1 - (cos_theta)^2);
theta = -acosd(0.6); // power factor angle at which power is consumed -
// - from the transformer in degrees

V_L = 6000 ; // Primary line voltage of a transformer in volt

P = 750 ; // kW expected to be delivered by the dc motor-generator

hp = 1000 ; // hp rating of the motor(induction or synchronous)
V_L_m = 6000 ; // Line voltage of a synchronous(or induction) motor in volt
cos_theta_sm = 0.8 ; // 0.8 leading power factor of the synchronous motor
theta_sm = acosd(0.8); // power factor angle of the synchronous motor in degrees

cos_theta_im = 0.8 ; // 0.8 lagging power factor of the induction motor
theta_im = -acosd(0.8); // power factor angle of the induction motor in degrees

eta = 0.92 ; // Efficiency of each motor

// Calculations
// case a : using Induction Motor(IM)
P_m = ( hp * 746 ) / eta ; // Induction(or synchronous) motor load in W
I_1 = P_m / ( sqrt(3) * V_L_m * cos_theta_im ); // Lagging current drawn by IM in A
```

```
I_1_prime = P_o * 1000 / ( sqrt(3) * V_L * cos_theta ); // Original lagging -
// - factory load current in A

// Total load current in A using Induction Motor :
I_TM = I_1*(cosd(theta_im) + %i*sind(theta_im)) + I_1_prime*(cosd(theta) + %i*sind(theta)) ;
I_TM_m = abs(I_TM);//I_TM_m = magnitude of I_TM in A
I_TM_a = atan(imag(I_TM) /real(I_TM))*180/%pi;// I_TM_a=phase angle of I_TM in degrees

PF_im = cosd(I_TM_a); // Overall PF using induction motor

// case b: using synchronous motor
I_s1 = P_m / ( sqrt(3) * V_L_m * cos_theta_sm ); // Lagging current drawn by IM in A

// Total load current in A using synchronous motor :
I_TSM = I_s1*(cosd(theta_sm) + %i*sind(theta_sm)) + I_1_prime*(cosd(theta) + %i*sind(theta)) ;
I_TSM_m = abs(I_TSM);//I_TSM_m = magnitude of I_TSM in A
I_TSM_a = atan(imag(I_TSM) /real(I_TSM))*180/%pi;// I_TSM_a=phase angle of I_TSM in degrees

PF_sm = cosd(I_TSM_a); // Overall PF using Synchronous motor

// case c
percent_I_L = ( I_TM_m - I_TSM_m ) / I_TM_m * 100 ; // Percent reduction in -
// - total load current in percent

// Display the results
printf("Note : case a,I1 calculated is around 97.53 A instead of 47.53 A(textbook).\n")
```

```
printf(" Note : case b,Actual I_s1 imaginary part is around 58.52 instead of ");
printf(" \n         52.52(textbook)so slight variation in I_TSM and percent ")
printf(" \n         reduction in total load current.\n")

disp("Example 8-7 Solution : ");
printf(" \n a: Induction(or sunchronous) motor load");
printf(" \n     P_m = %.f W ",P_m);
printf(" \n     Lagging current drawn by the IM = I1");
printf(" \n     I_1 = %.2f <-%.2f A \n",I_1,acosd(cos_theta_sm));
printf(" \n     I_1 in A = ");disp(I_1*cosd(-36.87)+%i*I_1*sind(-36.87));
printf(" \n     Original lagging factory load current = I_1_prime");
printf(" \n     I_1_prime in A = ");disp(I_1_prime*cosd(theta)+%i*I_1_prime*sind(theta));
printf(" \n     I_1_prime = %.1f <-%.2f A \n",I_1_prime,acosd(cos_theta));
printf(" \n     Total load current = motor load + factory load");
printf(" \n     I_TM = I_1 + I_1_prime\n");
printf(" \n     I_TM in A = ");disp(I_TM);
printf(" \n     I_TM = %.1f <%.1f A \n ",I_TM_m , I_TM_a );
printf(" \n     Overall system PF = %.4f lagging \n " , PF_im );

printf(" \n b: Synchronous motor load\n     I_s1 = %.2f <%.2f A\n",I_1,acosd(cos_theta_sm));
printf(" \n     I_s1 in A = ");disp(I_s1*cosd(36.87)+%i*I_s1*sind(36.87));
printf(" \n     Total load current : I_TSM = I_s1 + I_1_prime \n");
```

```
printf(" \n     I_TSM in A = ");disp(I_TSM);
printf(" \n     I_TSM = %.1f <%.1f A \n ",I_TSM_m , I_TSM_a );
printf(" \n     Overall system PF = %.1f lagging \n " , PF_sm );

printf(" \n c: Percent reduction in total load current = %.1f percent \n",percent_I_L);

printf(" \n d: PF improvement: Using the synchronous motor ( in lieu of the IM)");
printf(" \n     raises the total system PF from %.4f lagging to %.1f lagging.",PF_im,PF_sm);
```

Scilab code Exa 8.8 calculate Tp and hp

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 8: AC DYNAMO TORQUE RELATIONS − SYNCHRONOUS MOTORS
// Example 8−8

clear; clc; close; // Clear the work space and console.

// Given data from Ex.8−3a
// 3− phase Y−connected synchronous motor
P = 6 ; // No. of poles
hp = 50 ; // power rating of the synchronous motor in hp
V_L = 440 ; // Line voltage in volt
X_s = 2.4 ; // Synchronous reactance in ohm
```

```
R_a = 0.1 ; // Effective armature resistance in ohm
alpha = 20 ; // The rotor shift from the synchronous position in
// electrical degrees.
E_gp = 240 ; // Generated voltage/phase in volt when the motor is under-excited
f = 60 ; // Frequency in Hz

// Calculated values from Example 8-3a
V_p = 254 ; // Phase voltage in volt

// Calculations
// case a
// Torque developed per phase Using Eq.(8-17a)
S = 120 * f / P ; // Speed of the motor in rpm
T_p = ( 7.04 * E_gp * V_p ) / ( S*X_s) * sind(alpha) ;

// case b
// Total horsepower developed using part a
Horsepower = ( 3*T_p*S )/5252;

// Display the results
disp("Example 8-8 Solution : ");
printf(" \n From given and calculated data of Ex.8-3 a,\n");
printf(" \n a: T_p = %.2f lb-ft \n ", T_p );

printf(" \n b: Horsepower = %.1f hp ", Horsepower );
```

Scilab code Exa 8.9 calculate original kvar and kvar correction and kVA and Io and If and power triangle

```
// Electric Machinery and Transformers
// Irving L kosow
```

```
// Prentice Hall of India
// 2nd editiom

// Chapter 8: AC DYNAMO TORQUE RELATIONS − SYNCHRONOUS MOTORS
// Example 8−9

clear; clc; close; // Clear the work space and console.

// Given data
P_o = 2000 ; // Total power consumed by a factory in kW
cos_theta = 0.6 ; // 0.6  power factor at which power is consumed
sin_theta = sqrt( 1 - (cos_theta)^2 );
V = 6000 ; // Line voltage in volt
// Synchronous capacitor is used to raise the overall PF to unity
P_loss_cap = 275 ; // Synchronous capacitor losses in kW

// Calculations
// case a
S_o_conjugate = P_o / cos_theta ; // apparent complex power in kW
jQ_o = S_o_conjugate * sin_theta ; // Original kilovars of lagging load

// case b
jQ_c = -jQ_o ; // Kilovars of correction needed to bring the PF to unity

// case c
R = P_loss_cap ; // Synchronous capacitor losses in kW
S_c_conjugate = R - %i*( abs(jQ_c) ) ; // kVA rating of the synchronous capacitor
```

```
S_c_conjugate_m = abs(S_c_conjugate);//S_c_conjugate_m = magnitude of S_c_conjugate in kVA
S_c_conjugate_a = atan(imag(S_c_conjugate) /real(S_c_conjugate))*180/%pi;
//S_c_conjugate_a=phase angle of S_c_conjugate in degrees
PF = cosd(S_c_conjugate_a); // Power factor of the synchronous capacitor

// case d
I_o = S_o_conjugate * 1000 / V ; // Original current drawn from the mains in A


// case e
P_f = P_o + P_loss_cap ; // Total power in kW
S_f = P_f ; // Total apparent power in kW
S_f_m = abs(S_f);//S_f_m = magnitude of S_f in A
S_f_a = atan(imag(S_f) /real(S_f))*180/%pi;//S_f_a=phase angle of S_f in degrees

I_f = S_f * 1000 / V ; // Final current drawn from the mains after correction in A

// Display the results
disp("Example 8-9 Solution : ");
printf(" \n a: S*o = %d kVA \n", S_o_conjugate );
printf(" \n    +jQo in kvar = ");disp(%i*jQ_o);

printf(" \n b: -jQc in kvar =  " );disp(%i*jQ_c);

printf(" \n c: S*c in kVA = ");disp(S_c_conjugate);
printf(" \n    S*c = %.f <%.1f kVA \n", S_c_conjugate_m , S_c_conjugate_a );
printf(" \n    PF = %.3f leading \n",PF );

printf(" \n d: I_o = %.1f A \n ",I_o );
```

```
printf(" \n e: S_f in A = ");disp(S_f);
printf(" \n     S_f = %d <%d kVA \n" , S_f_m , S_f_a
   );
printf(" \n     I_f = %.1f A \n ", I_f);

printf(" \n f: See Fig.8-25.");
```

Scilab code Exa 8.10 calculate cost of raising PF to unity and point85 lagging

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 8: AC DYNAMO TORQUE RELATIONS -
   SYNCHRONOUS MOTORS
// Example 8-10

clear; clc; close; // Clear the work space and
   console.

// Given data
kVA = 10000 ; // kVA rating of a system
cos_theta = 0.65 ; // power factor of the system
sin_theta = sqrt( 1 - (cos_theta)^2 );
cos_theta_b = 0.85 ; // Raised PF
sin_theta_b = sqrt( 1 - (cos_theta_b)^2 );
cost = 60 ; // cost of the synchronous capacitor to
   improve the PF in dollars/kVA
// neglect the losses in the synchronous capacitor

// Calculations
// case a : For unity PF
```

```
// at the original load
kW_a = kVA * cos_theta ; //
theta = acosd(cos_theta) ; // Power factor angle of the system in degrees
kvar = kVA * sind(theta) ; // Reactive power in kvar
kVA_a = kvar ;
cost_cap_a = kvar * cost ; // Cost of raising the PF to unity PF in dollars

// case b
theta_b = acosd(cos_theta_b) ; // Power factor angle of the system in degrees
kVA_b = kW_a / cos_theta_b ; // kVA value reduction
kvar_b = kVA_b * sind(theta_b) ; // final kvar value reduced
kvar_add = kvar - kvar_b ; // kvar of correction added

cost_cap_b = kvar_add * cost ; // Cost of raising the PF to 0.85 PF in dollars

// Display the results

disp("Example 8-10 Solution : ");
printf(" \n    Note : Slight variations in the kvar and cost values are due to ");
printf(" \n    non-approximation of theta values while calculating in scilab.\n");
printf(" \n a: At the original load,\n");
printf(" \n    kW = %d kW at theta = %.1f degrees \n", kW_a , theta );
printf(" \n    kvar = %.3f kvar\n\n    For unity PF, ",kvar);
printf(" \n    kVA of synchronous capacitor = %.3f kVA (neglecting losses)\n",kVA_a);
printf(" \n    Cost of synchronous capacitor = $%.f \n\n",cost_cap_a );

```

```
printf(" \n b: For %.2f, PF = cos(%.1f), the total
   power,",cos_theta_b, theta_b);
printf(" \n    %.f kW, remains the same. Therefore,\n
   ",kW_a);
printf(" \n    kVA of final system reduced to = %.f
   kVA \n",kVA_b);
printf(" \n    kvar of final system reduced to = %.f
   kvar \n    Therefore,",kvar_b);

printf(" \n    kvar of correction added = %.3f kvar\
   n ",kvar_add);
printf(" \n    kVA of synchronous capacitor = %.3f
   kVA (neglecting losses)\n",kvar_add);
printf(" \n    Cost of synchronous capacitor = $%.f"
   ,cost_cap_b );
printf(" \n    or less than half the cost in part(a)
   ");
```

Scilab code Exa 8.11 calculate Po jQo and power triangle

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 8: AC DYNAMO TORQUE RELATIONS -
   SYNCHRONOUS MOTORS
// Example 8-11

clear; clc; close; // Clear the work space and
   console.

// Given data
S_conjugate = 1000 ; // Apparent complex power in
   kVA
```

```
cos_theta = 0.6 ; // lagging PF
sin_theta = sqrt( 1 - (cos_theta)^2 );

// Calculations
// case a
P_o = S_conjugate * cos_theta ; // Active power
    dissipated by the load in kW

// case b
jQ_o = S_conjugate * sin_theta ; // Inductive
    reactive quadrature power -
// - drawn from and returned to the supply

// Display the results

disp("Example 8-11 Solution : ");
printf(" \n a: Active power \n    P_o = %d kW \n ",
    P_o );

printf(" \n b: Inductive reactive quadrature power \
    n    +jQ_o in kvar =  \n");disp(%i*jQ_o);

printf(" \n c: The original power triangle is shown
    in Fig.8-26a.");
```

Scilab code Exa 8.12 calculate Pf jQf Pa jQa kVA and draw power tabulation grid

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 8: AC DYNAMO TORQUE RELATIONS -
    SYNCHRONOUS MOTORS
```

```
// Example 8-12

clear; clc; close; // Clear the work space and console.

// Given data
S_conjugate = 1000 ; // Apparent complex power in kVA
cos_theta_f = 0.8 ; // lagging PF
sin_theta_f = sqrt( 1 - (cos_theta_f)^2 );

// Calculated values from Ex.8-11
P_o = 600 ; // Active power dissipated by the load in kW
Q_o = 800 ; // Inductive reactive quadrature power -
// - drawn from and returned to the supply

// Calculations :

// case a
P_f = S_conjugate * cos_theta_f ; // Active power dissipated by the load in kW

// case b
Q_f = S_conjugate * sin_theta_f ; // Reactive quadrature power drawn from -
//  - and returned to the supply

// case c
P_a = P_f - P_o ; // Additional active power in kW that may be supplied to -
// - new customers

// case d
jQ_a = %i * ( Q_f ) - %i * ( Q_o ); // Correction kvar required to raise PF -
// -from 0.6 to o.8 lagging

```

```
// case e
S_c_conjugate = 0 - jQ_a ; // Rating of correction
    capacitors needed for case d

// Display the results

disp("Example 8-12 Solution : ");
printf(" \n a: P_f = %d kW \n ", P_f );
printf(" \n b: +jQ_f in kvar = ");disp(%i*Q_f);
printf(" \n c: P_a = %d kW \n ", P_a );
printf(" \n d: jQ_a in kvar =   ");disp(jQ_a)
printf(" \n e: S_c_conjugate = %d kVA \n ", abs(
    S_c_conjugate) );
printf(" \n f: The power tabulation grid is shown in
     Fig.8-26b.");
```

Scilab code Exa 8.13 calculate Pf jQf Pa jQa kVA and power tabulation grid

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 8: AC DYNAMO TORQUE RELATIONS -
    SYNCHRONOUS MOTORS
// Example 8-13

clear; clc; close; // Clear the work space and
    console.

// Ex.8-12 PF
cos_theta = 0.6 ; // PF lagging

// Given data
```

```
S_conjugate = 1000 ; // Apparent complex power in kVA
cos_theta_f = 1.0 ; // unity PF
sin_theta_f = sqrt( 1 - (cos_theta_f)^2 );

// Calculated values from Ex.8-11
P_o = 600 ; // Active power dissipated by the load in kW
Q_o = 800 ; // Inductive reactive quadrature power -
// - drawn from and returned to the supply

// Calculations :

// case a
P_f = S_conjugate * cos_theta_f ; // Active power dissipated by the load in kW

// case b
Q_f = S_conjugate * sin_theta_f ; // Reactive quadrature power drawn from -
//  - and returned to the supply

// case c
P_a = P_f - P_o ; // Additional active power in kW that may be supplied to -
// - new customers

// case d
jQ_a = %i * ( Q_f ) - %i * ( Q_o ); // Correction kvar required to raise PF -
// -from 0.6 to o.8 lagging
Q_a = -abs(jQ_a); //

// case e
S_c_conjugate = 0 - jQ_a ; // Rating of correction capacitors needed for case d

// Display the results
```

```

disp("Example 8-13 Solution : ");
printf(" \n a: P_f = %d kW \n ", P_f );
printf(" \n b: +jQ_f in kvar = ");disp(%i*Q_f);
printf(" \n c: P_a = %d kW \n ", P_a );
printf(" \n d: jQ_a in kvar =  ");disp(jQ_a)
printf(" \n e: S_c_conjugate = %d kVA \n ", abs(S_c_conjugate) );
printf(" \n f: The power tabulation grid is shown below.\n");
printf(" \n    \t\t P \t  jQ  \t S* ");
printf(" \n    \t\t(kW) \t(kvar) \t(kVA) \t  cos  ");
printf(" \n --------------------------------------------");
printf(" \n    Original : \t %d \t +j%d \t %d \t %.1f ",P_o ,Q_o ,S_conjugate,cos_theta);
printf(" \n    Added : \t %d \t %dj \t -- \t --",P_a ,Q_a );
printf(" \n    Final : \t %d \t +j%d \t %d \t %.1f",P_f ,Q_f ,S_conjugate,cos_theta_f);
```

Scilab code Exa 8.14 calculate original and final kVA kvar P and correction kvar Sa

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 8: AC DYNAMO TORQUE RELATIONS - SYNCHRONOUS MOTORS
// Example 8-14

clear; clc; close; // Clear the work space and
```

```
console.

// Given data
P_o = 2000 ; // load in kW drawn by a factory
cos_theta_o = 0.6 ; // PF lagging
sin_theta_o = sqrt( 1-  (cos_theta_o)^2 );
cos_theta_f = 0.85 ; // final PF lagging required
sin_theta_f = sqrt( 1-  (cos_theta_f)^2 );
P_a = 275 ; // Losses in the synchronous capacitor in kW

// Calculations
// case a
S_o_conjugate = P_o / cos_theta_o ; // Original kVA drawn from the utility

// case b
Q_o = S_o_conjugate * sin_theta_o ; // Original lagging kvar

// case c
P_f = P_o + P_a ; // Final system active power consumed from the utility in kW

// case d
S_f_conjugate = P_f / cos_theta_f ; // Final kVA drawn from the utility
S_f_conjugate_a = acosd(cos_theta_f); // Phase angle of S_f_conjugate in degrees

// case e
jQ_f = S_f_conjugate * sin_theta_f ; // Final lagging kvar
jQ_a = %i*(jQ_f) - %i*(Q_o); // Correction kvar produced by the synchronous capacitor
Q_a = abs(jQ_a); // Magnitude of jQ_a in kvar

// case f
```

```
P = P_a ;
S_a_conjugate = P -%i*(abs(jQ_a)); // kVA rating of
    the synchronous capacitor
S_a_conjugate_m = abs(S_a_conjugate);//
    S_a_conjugate_m = magnitude of S_a_conjugate in
    kVA
S_a_conjugate_a = atan(imag(S_a_conjugate) /real(
    S_a_conjugate))*180/%pi;
//S_a_conjugate_a=phase angle of S_a_conjugate in
    degrees
PF_f = cosd(S_a_conjugate_a); // PF

// Display the results
disp("Example 8-14 Solution : ");
printf(" \n a: S*o = %.1f kVA \n",S_o_conjugate );

printf(" \n b: Q*o in kvar = " );disp(%i*Q_o);

printf(" \n c: P*f = %.f kW \n",P_f );

printf(" \n d: S*f = %.1f <%.1f kVA\n ",
    S_f_conjugate,S_f_conjugate_a );

printf(" \n e: jQ_f in kvar = ");disp(%i*jQ_f);
printf(" \n    -jQ_a in kvar = ");disp(jQ_a);

printf(" \n f: S*a = %.f <%.2f kVA ",
    S_a_conjugate_m , S_a_conjugate_a );
printf(" \n    (cos(%.2f) = %.3f leading)\n",
    S_a_conjugate_a,PF_f);

printf(" \n g: Power tabulation grid : \n ");
printf(" \n    \t\t P \t  jQ \t S* ");
printf(" \n    \t\t(kW) \t(kvar) \t(kVA) \t cos  ")
    ;
printf(" \n
    ------------------------------------------");
printf(" \n    Original : \t %d \t +j%.f  %.1f  %.1f
```

```
    lag",P_o ,Q_o ,S_o_conjugate,cos_theta_o);
printf(" \n    Added     : \t %d \t -%.fj   %.f \t %.3
   f lead",P_a ,Q_a,S_a_conjugate_m,cosd(
   S_a_conjugate_a) );
printf(" \n    Final     : \t %d \t +j%.f   %.1f   %.2f
    lag",P_f ,jQ_f ,S_f_conjugate,cos_theta_f);
```

Scilab code Exa 8.15 calculate kVA added Pa and Qa and Pf Qf and PF

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 8: AC DYNAMO TORQUE RELATIONS -
   SYNCHRONOUS MOTORS
// Example 8-15

clear; clc; close; // Clear the work space and
   console.

// Given data
P_o = 2275 ; // Original kVA
Q_o = 1410 ; // Original kvar
S_f_conjugate = 3333.3 ; // final kVA of the load
S_o_conjugate = P_o + %i*Q_o ; // Load of the
   alternator in kVA
S_o_conjugate_m = abs(S_o_conjugate);//
   S_o_conjugate_m = magnitude of S_o_conjugate in
   kVA
S_o_conjugate_a = atan(imag(S_o_conjugate) /real(
   S_o_conjugate))*180/%pi;
//S_o_conjugate_a=phase angle of S_o_conjugate in
   degrees

```

```
disp("Example 8-15");
printf(" \n   Power tabulation grid : \n ");
printf(" \n   \t\t P \t\t   jQ  \t\t  S* ");
printf(" \n   \t\t(kW) \t\t(kvar) \t\t(kVA) \t\t
    cos   ");
printf(" \n
    ------------------------------------------------------------------
   ");
printf(" \n   Original : \t%d \t\t j%.f \t\t %.1f \t%
   .2f lag",real(S_o_conjugate) ,imag(S_o_conjugate)
   ,S_o_conjugate_m,cosd(S_o_conjugate_a));
printf(" \n   Added : \t0.8x \t\t j0.6x \t\t x \t\t0
   .80 lag" );
printf(" \n   Final :     (%d + 0.8x) \tj(%.f + 0.6x)
    %.1f \t0.841 lag\n",real(S_o_conjugate) ,imag(
   S_o_conjugate),S_f_conjugate );

// Calculations
// case a
// Assume x is the additional kVA load. Then real
   and quadrature powers are 0.8x and j0.6x
// respectively,as shown. Adding each column
   vertically and using the Pythagorean theorem,
// we may write (2275 + 0.8x)^2 + (1410 + 0.6x)^2 =
   (3333.3)^2, and solving this eqution yields
// the quadratic x^2 + 5352x -3947163 = 0. Applying
   the quadratic yields the added kVA load:
x = poly(0,'x'); // Defining a polynomial with
   variable 'x' with  root at 0
p = -3947163 + 5352*x + x^2
a = 1 ; // coefficient of x^2
b = 5332 ; // coefficient of x
c = -3947163 ; // constant

// Roots of p
x1 = ( -b + sqrt (b^2 -4*a*c ) ) /(2* a);
x2=( -b - sqrt (b^2 -4*a*c ) ) /(2* a);

```

```
// case b
P_a = 0.8*x1 ; // Added active power of the
    additional load in kW
Q_a = 0.6*x1 ; // Added reactive power of the
    additional load in kvar

// case c
P_f = P_o + P_a ; // Final active power of the
    additional load in kW
Q_f = Q_o + Q_a ; // Final reactive power of the
    additional load in kvar

// case d
PF = P_f / S_f_conjugate ; // Final power factor
// Validity check
S_conjugate_f = P_f + %i*Q_f ; // Final kVA of the
    load
S_conjugate_f_m = abs(S_conjugate_f);//
    S_conjugate_f_m = magnitude of S_conjugate_f in
    kVA
S_conjugate_f_a = atan(imag(S_conjugate_f) /real(
    S_conjugate_f))*180/%pi;
//S_conjugate_f_a=phase angle of S_conjugate_f in
    degrees

// Display the results

disp(" Solution : ")

printf(" \n a: The given data is shown in the above
    power tabulation grid.Assume");
printf(" \n    x is the additional kVA load. Then
    real and quadrature powers are");
printf(" \n    0.8x and j0.6x respectively,as shown.
    Adding each column vertically");
printf(" \n    and using the Pythagorean theorem, we
     may write");
printf(" \n    (2275 + 0.8x)^2 + (1410 + 0.6x)^2 =
```

```
(3333.3)^2, and solving this");
printf(" \n    equation yields the quadratic as follows : \n");
printf(" \n    x^2 + 5332x -3947163 = 0. \n ")
printf(" \n    Applying the quadratic yields the added kVA load:");
printf(" \n    Roots of quadratic Eqn p are \n ");
printf(" \n    x1 = %.2f \n    x2 = %.2f ", x1, x2 );
printf(" \n    Consider +ve value of x for added kVA so");
printf(" \n    x = S*a = %.2f kVA \n ", x1 );

printf(" \n b: P_a = %.1f kW \n ", P_a );
printf(" \n    Q_a in kvar = \n");disp(%i*Q_a);

printf(" \n c: P_f = %.1f kW \n ", P_f );
printf(" \n    Q_f in kvar = \n");disp(%i*Q_f);

printf(" \n d: PF = cos _f = %.3f lagging \n ", PF );
printf(" \n    Validity check\n    S*f = ");disp(S_conjugate_f);
printf(" \n    S*f = %.1f <%.2f kVA \n", S_conjugate_f_m,S_conjugate_f_a);
printf(" \n    PF = cos(%.1f) = %.3f lagging", S_conjugate_f_a ,cosd(S_conjugate_f_a));
```

Scilab code Exa 8.16 Verify tellegens theorem for kVAs found in Ex 8 15

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

```

```
// Chapter 8: AC DYNAMO TORQUE RELATIONS - SYNCHRONOUS MOTORS
// Example 8-16

clear; clc; close; // Clear the work space and console.

// Given data
// Calculated values as per Ex.8-15 are as follows
S_o_conjugate = 2676.5*exp(%i*31.79*(%pi/180)); // Original kVA rating
S_o_conjugate_m = abs(S_o_conjugate);// S_o_conjugate_m = magnitude of S_o_conjugate in kVA
S_o_conjugate_a = atan(imag(S_o_conjugate) /real(S_o_conjugate))*180/%pi;
//S_o_conjugate_a=phase angle of S_o_conjugate in degrees

S_a_conjugate = 658.86*exp(%i*36.87*(%pi/180)); // Added kVA rating
S_a_conjugate_m = abs(S_a_conjugate);// S_a_conjugate_m = magnitude of S_a_conjugate in kVA
S_a_conjugate_a = atan(imag(S_a_conjugate) /real(S_a_conjugate))*180/%pi;
//S_a_conjugate_a=phase angle of S_a_conjugate in degrees

S_f_conjugate = -3333.3*exp(%i*32.792687*(%pi/180)); // Final kVA rating
S_f_conjugate_m = abs(S_f_conjugate);// S_f_conjugate_m = magnitude of S_f_conjugate in kVA
S_f_conjugate_a = atan(imag(S_f_conjugate) /real(S_f_conjugate))*180/%pi;
//S_f_conjugate_a=phase angle of S_f_conjugate in degrees
```

```
// Calculations
kVA_total = S_o_conjugate + S_a_conjugate + S_f_conjugate ; // Tellegan's theorem
kVA_total_m = abs(kVA_total);//kVA_total_m = magnitude of kVA_total in kVA
kVA_total_a = atan(imag(kVA_total) /real(kVA_total))*180/%pi;
//kVA_total_a=phase angle of kVA_total in degrees

// Display the result
disp("Example 8-16 Solution : ");
printf(" \n From the solution to Ex.8-15, we have ");
printf(" \n S*o = %.1f <%.2f kVA \n ",S_o_conjugate_m,S_o_conjugate_a );
printf(" \n S*a = %.1f <%.2f kVA \n ",S_a_conjugate_m,S_a_conjugate_a );
printf(" \n S*f = %.1f <%.2f kVA \n ",S_f_conjugate_m,S_f_conjugate_a );

printf(" \n Validity check ");
printf(" \n S*o + S*a + S*f =  ");
disp(S_o_conjugate),printf("  +"),disp(S_a_conjugate),printf("  +"),disp(S_f_conjugate);
printf(" \n    = %d ",kVA_total );
printf(" \n Hence, Tellegen`s theorem is proved");
```

Scilab code Exa 8.17 calculate overall PF using unity PF SM

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

```

```
// Chapter 8: AC DYNAMO TORQUE RELATIONS - SYNCHRONOUS MOTORS
// Example 8-17

clear; clc; close; // Clear the work space and console.

// Given data
kW = 40000 ; // Load on a factory in kW
PF = 0.8 ; // power factor lagging of the load
cos_theta = PF;
sin_theta = sqrt( 1 - (cos_theta)^2 );
hp = 7500 ; // power rating of the induction motor in hp
PF_IM = 0.75 ; // power factor lagging of the induction motor
cos_theta_IM = PF_IM;
sin_theta_IM = sqrt( 1 - (cos_theta_IM)^2 );
eta = 91*(1/100) ; // Efficiency of IM
PF_SM = 1 ; // power factor  of the synchronous motor

// Calculations
kVA_original = kW / PF ; // Original kVA
kvar_original = kVA_original * sin_theta ; // Original kvar

kW_IM = ( hp * 746 ) / ( 1000 * eta ) ; // Induction motor kW
kVA_IM = kW_IM / PF_IM ; // Induction motor kVA
kvar_IM = kVA_IM * sin_theta_IM ; // Induction motor kvar

kvar_final = kvar_original - kvar_IM ; // final kvar
kVA_final = kW + %i*(abs(kvar_final)); // final kVA
kVA_final_m = abs(kVA_final);//kVA_final_m = magnitude of kVA_final in kVA
kVA_final_a = atan(imag(kVA_final) /real(kVA_final))
```

```
    *180/%pi;
//kVA_final_a=phase angle of kVA_final in degrees

PF_final = cosd(kVA_final_a); // Final power factor

// Display the result
disp("Example 8-17 Solution : ");
printf(" \n The synchronous motor operates at the
    same efficiency as the IM");
printf(" \n that has been replaced, and therefore
    the total power of the system");
printf(" \n is unchanged. The solution involves
    construction of table that shows ")
printf(" \n the original condition of the system,
    the change, and the final condition.\n");
printf(" \n Original kVA = %d kVA \n ", kVA_original
    );
printf(" \n Original kvar = \n" );disp(%i*
    kvar_original);

printf(" \n Induction motor kW = %d kW \n ", kW_IM )
    ;
printf(" \n Induction motor kVA = %.f kVA \n ",
    kVA_IM );
printf(" \n Induction motor kvar = ");disp(%i*
    kvar_IM)

printf(" \n Final kvar = ");disp(%i*kvar_final);
printf(" \n Final kVA = " );disp(kVA_final);
printf(" \n Final kVA = %f <%.2f kVA \n ",
    kVA_final_m,kVA_final_a);

printf(" \n Final PF = %.3f lagging \n ", PF_final )
    ;

printf(" \n
    ------------------------------------------------------------------
    ");
```

```
printf(" \n Power tabulation grid : \n ");
printf(" \n \t\t P \t\t  jQ  \t\t S* ");
printf(" \n \t\t(kW) \t\t(kvar) \t\t(kVA) \t\t  cos ");
printf(" \n ----------------------------------------------------------------------------- ");
printf(" \n Original : \t%d \t\tj%.f \t\t%.1d \t\t %.1f lag",kW ,kvar_original ,kVA_original,PF);
printf(" \n Removed  : \t-%.f \t\t-(+j%.f) \t%.f \t\t %.2f lag",kW_IM,kvar_IM,kVA_IM,PF_IM);
printf(" \n Added    : \t+%.f  \t\t   0      \t%.1f \t\t 1.0 ",kW_IM,kW_IM);
printf(" \n Final    : \t%d \t\tj%.f \t\t%.1f \t %.3f lag",kW ,kvar_final ,kVA_final_m,PF_final);
printf(" \n ----------------------------------------------------------------------------- ");
```

Scilab code Exa 8.18 calculate overall PF using point8 PF leading SM

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 8: AC DYNAMO TORQUE RELATIONS - SYNCHRONOUS MOTORS
// Example 8-18

clear; clc; close; // Clear the work space and console.

// Given data
kW = 40000 ; // Load on a factory in kW
```

```
PF = 0.8 ; // power factor lagging of the load
cos_theta = PF;
sin_theta = sqrt( 1 - (cos_theta)^2 );

PF_SM = 0.8 ; // power factor leading of the synchronous motor
cos_theta_SM = PF_SM;
sin_theta_SM = sqrt( 1 - (cos_theta_SM)^2 );
hp = 7500 ; // power rating of the induction motor in hp

PF_IM = 0.75 ; // power factor lagging of the induction motor
cos_theta_IM = PF_IM;
sin_theta_IM = sqrt( 1 - (cos_theta_IM)^2 );

eta = 91*(1/100) ; // Efficiency of IM

// Calculations
kVA_original = kW / PF ; // Original kVA
kvar_original = kVA_original * sin_theta ; // Original kvar


kW_IM = ( hp * 746 ) / ( 1000 * eta ) ; // Induction motor kW
kVA_IM = kW_IM / PF_IM ; // Induction motor kVA
kvar_IM = kVA_IM * sin_theta_IM ; // Induction motor kvar

// case a
kW_SM = ( hp * 746 ) / ( 1000 * eta ) ; // Synchronous motor kW
kVA_SM = kW_SM / PF_SM ; // Synchronous motor kVA
kvar_SM = kVA_SM * sin_theta_SM ; // Synchronous motor kvar

kvar_final = kvar_original - kvar_IM - kvar_SM ; //
```

```
final kvar
kVA_final = kW + %i*(abs(kvar_final)); // final kVA
kVA_final_m = abs(kVA_final);//kVA_final_m = magnitude of kVA_final in kVA
kVA_final_a = atan(imag(kVA_final) /real(kVA_final))*180/%pi;
//kVA_final_a=phase angle of kVA_final in degrees

PF_final = cosd(kVA_final_a); // Final power factor

// Display the result
disp("Example 8-18 Solution : ");

printf(" \n Original kVA = %d kVA \n ", kVA_original);
printf(" \n Original kvar = \n" );disp(%i*kvar_original);
printf(" \n a:");
printf(" \n Synchronous motor kW = %d kW \n ", kW_SM);
printf(" \n Synchronous motor kVA = %.f kVA \n ", kVA_SM );
printf(" \n Synchronous motor kvar = ");disp(-%i*kvar_SM)

printf(" \n Final kvar = ");disp(%i*kvar_final);
printf(" \n Final kVA = " );disp(kVA_final);
printf(" \n Final kVA = %f <%.2f kVA \n ", kVA_final_m,kVA_final_a);

printf(" \n Final PF = %.3f lagging \n ", PF_final );

printf(" \n -----------------------------------------------------------------------------");
printf(" \n Power tabulation grid : \n ");
printf(" \n \t\t P \t\t  jQ  \t\t S* ");
```

```
printf(" \n \t\t(kW) \t\t(kvar) \t\t(kVA) \t\t cos ");
printf(" \n -----------------------------------------------------------------------------------------------");
printf(" \n Original : \t%d \t\tj%.f \t\t%.1d \t\t %.1f lag",kW ,kvar_original ,kVA_original,PF);
printf(" \n Removed  : \t-%.f \t\t-(+j%.f) \t%.f \t\t %.2f lag",kW_IM,kvar_IM,kVA_IM,PF_IM);
printf(" \n Added    : \t+%.f  \t\t-j%.2f      \t%.1f \t\t %.1f lead",kW_SM,abs(kvar_SM),kVA_SM,PF_SM);
printf(" \n Final    : \t%d \t\tj%.2f \t%.1f \t %.3f lag",kW ,kvar_final ,kVA_final_m,PF_final);
printf(" \n -----------------------------------------------------------------------------------------------\n\n");

printf(" \n b: ");
printf(" \n In Ex.8-17, a 6148 kVA, unity PF, 7500 hp synchronous motor is needed.");
printf(" \n In Ex.8-18, a 7685 kVA, 0.8 PF leading, 7500 hp synchronous motor is needed.\n");
printf(" \n \t Ex.8-18b shows that a 0.8 PF leading ,7500 hp synchronous motor ");
printf(" \n must be physically larger than a unity PF,7500 hp synchronous motor ");
printf(" \n because of its higher kVA rating.");
```

Scilab code Exa 8.19 calculate kVA and PF of system and same for SM

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom
```

```
// Chapter 8: AC DYNAMO TORQUE RELATIONS − SYNCHRONOUS MOTORS
// Example 8−19

clear; clc; close; // Clear the work space and console.

// Given data
kVA_load = 500 ; // Load of 500 kVA
PF_load = 0.65 ; // Load operates at this PF lagging
cos_theta_load = PF_load ;
sin_theta_load = sqrt(1 - (cos_theta_load)^2);
hp = 200 ; // power rating of the system in hp
eta = 88*(1/100); // Efficiency of the system after adding the load
PF_final = 0.85 ; // Final lagging PF after adding the load

// Calculations
kW_original = kVA_load * cos_theta_load ; // Original kW
kvar_original = kVA_load * sin_theta_load ; // Original kvar

kW_SM = ( hp * 746 ) / ( 1000 * eta ) ; // Synchronous motor kW

// case a
kW_final = kW_original + kW_SM ; // final kW of the system with the motor added
kVA_final = kW_final / PF_final ; // final kVA of the system with the motor added
PF_system = kW_final / kVA_final ; // Final PF of the system with the motor added
cos_theta_system = PF_system ; // Final PF of the system with the motor added
sin_theta_system  = sqrt(1 - (cos_theta_system )^2);
```

```

kvar_final = kVA_final * sin_theta_system ; // final kvar of the system with the motor added

// case b
kvar_SM = %i*kvar_final - %i*kvar_original ; // kvar rating of the sychronous motor

kVA_SM = kW_SM + kvar_SM ; // kVA rating of the sychronous motor
kVA_SM_m = abs(kVA_SM);//kVA_SM_m = magnitude of kVA_SM in kVA
kVA_SM_a = atan(imag(kVA_SM) /real(kVA_SM))*180/%pi;
//kVA_SM_a=phase angle of kVA_SM in degrees

PF_SM = cosd(kVA_SM_a); // PF of the sychronous motor

// Display the result
disp("Example 8-19 Solution : ");

printf(" \n Original kW = %.f kW \n ", kW_original );
printf(" \n Original kvar = %.f kvar\n", kvar_original );
printf(" \n Synchronous motor kW = %.1f kW \n ", kW_SM );

printf(" \n a: Final kW = %.1f kW",kW_final);
printf(" \n    Final kVA of the system = %.f kVA", kVA_final);
printf(" \n    System PF = %.2f lagging",PF_system);
printf(" \n    Final kvar of the system = j%d (lagging)kvar\n\n",kvar_final);

printf(" \n b: Synchronous motor kvar = -%.2fj(leading)kvar\n",abs(kvar_SM));
printf(" \n    Synchronous motor kVA = " );disp(
```

```
kVA_SM);
printf("\n    Synchronous motor kVA = %.f <%.1f kVA \n ", kVA_SM_m , kVA_SM_a );
printf("\n    Synchronous motor PF = cos(%.1f) = %.3f leading \n ",kVA_SM_a,PF_SM );

printf("\n ---------------------------------------------------------------------"
);
printf("\n    Power tabulation grid : \n ");
printf("\n    \t\t P \t  jQ  \t S* ");
printf("\n    \t\t(kW) \t(kvar) \t(kVA) \t cos  ")
;
printf("\n ---------------------------------------------------------------------"
);
printf("\n    Original : \t %d \t +j%.f   %.1d \t %.2f lag",kW_original,kvar_original,kVA_load,
PF_load);
printf("\n    Added    : \t %.1f \t -%.1fj   %.f \t %.4f lead",kW_SM ,abs(kvar_SM),kVA_SM_m,PF_SM);
printf("\n    Final    : \t %.1f \t +j%.f   %.f     %.2f lag",kW_final,kvar_final,kVA_final,
PF_final);
printf("\n ---------------------------------------------------------------------"
);
```

Scilab code Exa 8.20 calulate speeds and poles for alternator and motor

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

```

```
// Chapter 8: AC DYNAMO TORQUE RELATIONS −
    SYNCHRONOUS MOTORS
// Example 8−20

clear; clc; close; // Clear the work space and
    console.

// Given data
f_a = 400 ; // Frequency of the alternator in Hz
f_m = 60 ; // Frequency of the motor in Hz

// Calculations
Pole_ratio = f_a / f_m ; // Ratio of no. of poles in
     alternator to that of motor
// Subscript 1 below indicates 1st combination
P_a1 = 40 ; // first combination must have 40 poles
    on the alternator
P_m1 = 6 ; // first combination must have 6 poles on
     the synchronous motor at a speed
S_m1 = (120*f_m) / P_m1 ; // Speed of the motor in
    rpm

// Subscript 2 below indicates 2nd combination
P_a2 = 80 ; // second combination must have 40 poles
     on the alternator
P_m2 = 12 ; // second combination must have 12 poles
     on the synchronous motor at a speed
S_m2 = (120*f_m) / P_m2 ; // Speed of the motor in
    rpm

// Subscript 13below indicates 3rd combination
P_a3 = 120 ; // third combination must have 40 poles
     on the alternator
P_m3 = 18 ; // third combination must have 18 poles
    on the synchronous motor at a speed
S_m3 = (120*f_m) / P_m3 ; // Speed of the motor in
    rpm

```

```
// Display the result
disp("Example 8-20 Solution : ");

printf(" \n Since P_a/P_m = f_a/f_m = %d/%d, or %d/
   %d, the ratio of",f_a,f_m,f_a/20,f_m/20);
printf(" \n f_a/f_m determines the combinations of
    poles and speed.\n");
printf(" \n Only even multiples of the above ratio
    are possible,since poles ");
printf(" \n are always in pairs, hence first three
    combinations are as follows \n");

printf(" \n The first combination must have %d poles
     on the alternator and ",P_a1);
printf(" \n %d poles on the sychronous motor at a
    speed = %d rpm.\n",P_m1,S_m1);

printf(" \n The second combination must have %d
    poles on the alternator and ",P_a2);
printf(" \n %d poles on the sychronous motor at a
    speed = %d rpm.\n",P_m2,S_m2);

printf(" \n The third combination must have %d poles
     on the alternator and ",P_a3);
printf(" \n %d poles on the sychronous motor at a
    speed = %d rpm.\n",P_m3,S_m3);

printf(" \n
    ----------------------------------------------------------------
   ");
printf(" \n Combination \t  Alternator Poles \t
    Motor Poles \t Speed (rpm)");
printf(" \n              \t         P_a        \t
          P_m       \t     S       ");
printf(" \n
    ----------------------------------------------------------------
   ");
printf(" \n First \t\t:\t  %d\t\t        %d \t     %d"
```

```
    ,P_a1,P_m1,S_m1);
printf(" \n Second\t\t:\t  %d\t\t          %d \t      %d"
    ,P_a2,P_m2,S_m2);
printf(" \n Third \t\t:\t  %d\t\t          %d \t      %d"
    ,P_a3,P_m3,S_m3);
printf(" \n
    ----------------------------------------------------------------------
    ");
```

Chapter 9

POLYPHASE INDUCTION OR ASYNCHRONOUS DYNAMOS

Scilab code Exa 9.1 calculate poles and synchronous speed

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS)
    DYNAMOS
// Example 9-1

clear; clc; close; // Clear the work space and
    console.

// Given data
phase = 3 ; // Number of phases
n = 3 ; // Slots per pole per phase
f = 60 ; // Line frequency in Hz

```

```
// Calculations
// case a
P = 2 * n ; // Number of poles produced
Total_slots = n * P * phase ; // Total number of slots on the stator

// case b
S_b = (120*f)/P ; // Speed in rpm of the rotating magnetic field

// case c
f_c = 50 ; // Changed line frequency in Hz
S_c = (120*f_c)/P ; // Speed in rpm of the rotating magnetic field

// Display the results
disp("Example 9-1 Solution : ");
printf(" \n a: P = %d poles \n     Total slots = %d slots \n", P ,Total_slots );

printf(" \n b: S = %d rpm  @ f = %d Hz \n ", S_b , f );

printf(" \n c: S = %d rpm  @ f = %d Hz ", S_c ,f_c );
```

Scilab code Exa 9.2 calculate rotor speed

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS) DYNAMOS
```

```
// Example 9-2

clear; clc; close; // Clear the work space and console.

// Given data

s_a = 5*(1/100); // Slip (case a)
s_b = 7*(1/100); // Slip (case b)

// Given data and calculated values from Ex.9-1
f_a = 60 ; // Line frequency in Hz (case a)
f_b = 50 ; // Line frequency in Hz (case b)
S_a = 1200 ; // Speed in rpm of the rotating magnetic field (case a)
S_b = 1000 ; // Speed in rpm of the rotating magnetic field (case b)

// Calculations

// case a
S_r_a = S_a * ( 1 - s_a ); // Rotor speed in rpm when slip is 5% (case a)

// case b
S_r_b = S_b * ( 1 - s_b ); // Rotor speed in rpm when slip is 7% (case b)

// Display the results
disp("Example 9-2 Solution : ");

printf(" \n a: S_r = %.f rpm @ s = %.2f \n ", S_r_a ,s_a );

printf(" \n b: S_r = %.f rpm @ s = %.2f ", S_r_b , s_b );
```

Scilab code Exa 9.3 calculate rotor frequency

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS)
    DYNAMOS
// Example 9-3

clear; clc; close; // Clear the work space and
    console.

// Given data
P = 4 ; // Number of poles in Induction motor
f = 60 ; // Frequency in Hz
s_f = 5*(1/100) ; // Full-load rotor slip

// Calculations

// case a
// slip , s = (S -S_r)/S ;
// where S = Speed in rpm of the rotating magnetic
    field and
//        S_r = Speed in rpm of the rotor
s = 1 ; // Slip = 1, at the instant of starting ,
    since S_r is zero
f_r_a = s * f ; // Rotor frequency in Hz at the
    instant of starting

// case b
f_r_b = s_f * f ;// Full-load rotor frequency in Hz

```

```
// Display the results
disp("Example 9-3 Solution : ");

printf(" \n a: At the instant of starting, slip s =
   (S -S_r)/S ; ");
printf(" \n    where S_r is the rotor speed. Since
   the rotor speed at the ");
printf(" \n    instant of starting is zero, s = (S -
    0)/S = 1 , or unity slip.");
printf(" \n\n    The rotor frequency is \n     f_r  =
    %d Hz \n\n ", f_r_a);

printf(" \n b: At full-load,the slip is 5 percent(as
    given), and therefore");
printf(" \n     s = %.2f \n     f_r = %d Hz " , s_f ,
   f_r_b);
```

Scilab code Exa 9.4 calculate starting torque and current

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS)
   DYNAMOS
// Example 9-4

clear; clc; close; // Clear the work space and
   console.

// Given data
P = 4 ; // Number of poles in the IM
hp = 50 ; // rating of the IM in hp
V_o = 208 ; // Voltage rating of the IM in volt
```

```
T_orig = 225 ; // Starting torque in lb-ft
I_orig = 700 ; // Instantaneous startign current in
    A at rated voltage
V_s = 120 ; // Reduced 3-phase voltage supplied in
    volt

// Calculations
// case a
T_s = T_orig * (V_s/V_o)^2 ; // Starting torque in
    lb-ft after application of V_s

// case b
I_s = I_orig * (V_s/V_o) ; // Starting current in A
    after application of V_s

// Display the results
disp("Example 9-4 Solution : ");
printf(" \n a: Starting torque :\n    T_s = %.f lb-
    ft \n",T_s );

printf(" \n b: Starting current :\n    I_s = %d A \n
    ",I_s );
```

Scilab code Exa 9.5 calculate s Xlr fr Sr

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS)
    DYNAMOS
// Example 9-5

clear; clc; close; // Clear the work space and
```

```
    console.

// Given data
P = 8 ; // Number of poles in the SCIM
f = 60 ; // Frequency in Hz
R_r = 0.3 ; // Rotor resistance per phase in ohm
S_r = 650 ; // Speed in rpm at which motor stalls

// Calculations
// case a
S = (120*f)/P ; // Speed in rpm of the rotating magnetic field
s_b = (S - S_r)/S ; // Breakdown Slip

// case b
X_lr = R_r / s_b ; // Locked rotor reactance in ohm

// case c
f_r = s_b * f ; // Rotor frequency in Hz, at the maximum torque point

// case d
s = 5*(1/100);//  Rated slip
S_r = S * (1 - s); // Full-load in rpm speed at rated slip

// Display the results
disp("Example 9-5 Solution : ");
printf(" \n a: S = %d rpm \n    s_b = %.3f \n", S , s_b );

printf(" \n b: X_b = %.2f ohm \n ", X_lr );

printf(" \n c: f_r = %.1f Hz \n ", f_r );

printf(" \n d: S = %d rpm \n ", S_r );
```

Scilab code Exa 9.6 calculate full load S and Tf

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS) DYNAMOS
// Example 9-6

clear; clc; close; // Clear the work space and console.

// Given data
P = 8 ; // Number of poles in the SCIM
f = 60 ; // Frequency in Hz
R_r = 0.3 ; // rotor resistance per phase in ohm/phase
R_x = 0.7 ; // Added resistance in ohm/phase
R_r_total = R_r + R_x ; // Total resistance per phase in ohm
S_r = 875 ; // Full-load Speed in rpm


// Calculated values from Ex.9-6
S = 900 ; // Speed in rpm of the rotating magnetic field
X_lr = 1.08 ; // Locked rotor reactance in ohm

// Calculations
// case a
s = (S - S_r)/S ; // Full-load slip,short circuited
s_r = R_r_total / R_r * s; // New full-load slip
```

```
with added resistance

S_r_new = S*(1-s_r); // New full-load speed in rpm

// case b
// Neglecting constant Kn_t ,since we are taking torque ratios
T_o = ( R_r / ((R_r)^2 + (X_lr)^2) ); // Original torque
T_f = ( R_r + R_x) / ( (R_r + R_x)^2  + (X_lr)^2 ); // Original torque

torque_ratio = T_f / T_o ; // Ratio of final torque to original torque
T_final = 2*torque_ratio ;

// Display the results
disp("Example 9-6 Solution : ");
printf(" \n a: The full-load slip ,short circuited ,is ");
printf(" \n     s = %.4f \n",s );
printf(" \n     Since slip is proportional to rotor resistance and since the ");
printf(" \n     increased rotor resistance is R_r = %.1f + %.1f = %d ,",R_x,R_r,R_r_total);
printf(" \n     the new full-load slip with added resistance is : ");
printf(" \n     s_r = %.4f \n",s_r);
printf(" \n     The new full-load speed is : " );
printf(" \n     S(1-s) = %.f rpm \n",S_r_new );

printf(" \n b: The original starting torque T_o was twice the full-load torque");
printf(" \n     with a rotor resistance of %.1f ohm and a rotor reactance of %.2f ohm",R_r,X_lr);
printf(" \n     (Ex.9-5).The new starting torque conditions may be summarized by the   ");
printf(" \n     following table and compared from Eq
```

```
    .(9-14),where T_o ");
printf(" \n     is the original torque and T_f is the
    new torque.");

printf(" \n
    ----------------------------------------");
printf(" \n     Condition \t R_r \t X_lr \t
    T_starting ");
printf(" \n                  \t ohm \t ohm  \t ");
printf(" \n
    ----------------------------------------");
printf(" \n     Original  : \t %.1f \t %.2f \t 2*T_n "
    ,R_r,X_lr);
printf(" \n     New       : \t %.1f \t %.2f \t   ?   "
    ,R_r_total,X_lr);
printf(" \n
    ----------------------------------------\n");

printf(" \n     T_o = %.2f * K_n_t",T_o);
printf(" \n     T_f = %.3f * K_n_t",T_f);
printf(" \n     T_f/T_o = %.2f and T_f = %.2f * T_o\n
    ",torque_ratio,torque_ratio);
printf(" \n     Therefore,\n     T_f = %.3f * T_n",
    T_final);
```

Scilab code Exa 9.7 calculate rotor I and PF and same with added Rr

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS)
    DYNAMOS
// Example 9-7
```

```
clear; clc; close; // Clear the work space and
    console.

// Given data
P = 8 ; // Number of poles in the SCIM
f = 60 ; // Frequency in Hz
R_r = 0.3 ; // Rotor resistance per phase in ohm
R_x = 0.7 ; // Added resistance in ohm/phase
R_r_total = R_r + R_x ; // Total resistance per
    phase in ohm
X_lr = 1.08 ; // Locked rotor reactance in ohm
S_r = 650 ; // Speed in rpm at which motor stalls
E_lr = 112 ; // Induced voltage per phase

// Calculations
// case a
Z_lr = R_r + %i*X_lr ; // Locked rotor impedance per
     phase
Z_lr_m = abs(Z_lr);//Z_lr_m = magnitude of Z_lr in
    ohm
Z_lr_a = atan(imag(Z_lr) /real(Z_lr))*180/%pi;//
    Z_lr_a=phase angle of Z_lr in degrees

I_r = E_lr / Z_lr_m ; // Rotor current per phase
cos_theta_r = cosd(Z_lr_a); // rotor power factor
    with the rotor short-circuited
cos_theta = R_r / Z_lr_m ; // rotor power factor
    with the rotor short-circuited

// case b
// 1 at the end of Z_lr1 is just used for showing
    its different form Z_lr
// and for ease in calculations
Z_lr1 = R_r_total + %i*X_lr ; // Locked rotor
    impedance per phase
Z_lr1_m = abs(Z_lr1);//Z_lr1_m = magnitude of Z_lr1
    in ohm
```

```
Z_lr1_a = atan(imag(Z_lr1) /real(Z_lr1))*180/%pi;// Z_lr1_a=phase angle of Z_lr1 in degrees

I_r1 = E_lr / Z_lr1_m ; // Rotor current per phase
cos_theta_r1 = cosd(Z_lr1_a); // rotor power factor with the rotor short-circuited
cos_theta1 = R_r_total / Z_lr1_m ; // rotor power factor with the rotor short-circuited

// Display the results
disp("Example 9-7 Solution : ");
printf(" \n a: The locked-rotor impedance per phase is : ");
printf(" \n     Z_lr in ohm = "),disp(Z_lr);
printf(" \n     Z_lr = %.2f <%.1f ohm \n",Z_lr_m, Z_lr_a);
printf(" \n     I_r = %.f A \n",I_r);
printf(" \n     cos _r = cos(%.1f) = %.3f or \n     cos  = R_r/Z_lr = %.3f",Z_lr_a,cos_theta_r, cos_theta);

printf(" \n\n\n b: The locked-rotor impedance with added rotor resistance per phase is : ");
printf(" \n     Z_lr in ohm = "),disp(Z_lr1);
printf(" \n     Z_lr = %.2f <%.1f ohm \n",Z_lr1_m, Z_lr1_a);
printf(" \n     I_r = %.1f A \n",I_r1);
printf(" \n     cos _r = cos(%.1f) = %.3f or \n     cos  = R_r/Z_lr = %.3f",Z_lr1_a,cos_theta_r1, cos_theta1);
```

Scilab code Exa 9.8 calculate Rx and rotor PF and starting current

```
// Electric Machinery and Transformers
// Irving L kosow
```

```
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS) DYNAMOS
// Example 9-8

clear; clc; close; // Clear the work space and console.

// Given data  (Exs.9-5 through 9-7)
P = 8 ; // Number of poles in the SCIM
f = 60 ; // Frequency in Hz
R_r = 0.3 ; // Rotor resistance per phase in ohm
X_lr = 1.08 ; // Locked rotor reactance in ohm
S_r = 650 ; // Speed in rpm at which motor stalls
E_lr = 112 ; // Induced voltage per phase

disp("Example 9-8 : ");
printf(" \n The new and the original conditions may be summarized in the following table\n");
printf(" \n ----------------------------------------------------------");
printf(" \n    Condition \t R_r \t\t X_lr \t\t T_starting ");
printf(" \n              \t ohm \t\t ohm  \t ");
printf(" \n ----------------------------------------------------------");
printf(" \n    Original : \t %.1f \t\t %.2f \t\t T_o = 2*T_n ",R_r,X_lr);
printf(" \n    New      :\t(%.1f+R_x) \t %.2f \t\t T_n = 2*T_n ",R_r,X_lr);
printf(" \n ----------------------------------------------------------\n");

```

```
// Calculating
// case a
// Neglecting constant Kn_t ,since we are equating torque T_o and T_n
T_o = ( R_r / ((R_r)^2 + (X_lr)^2) ); // Original torque

// T_o = K_n_t*( 0.3 / ((0.3)^2 + (1.08)^2) );
// T_n = K_n_t*( 0.3 + R_x) / ( (0.3 + R_x)^2 + (1.08)^2 );
// T_n = T_o
// Simplyifing yields
// 0.3 + R_x = 0.24[(0.3+R_x)^2 + (1.08)^2]
// Expanding and combining the terms yields
// 0.24*(R_x)^2 - 0.856*R_x = 0
// This is a quadratic equation having two roots , which may be factored as
// R_x*(0.24*R_x - 0.856) = 0,yielding
// R_x = 0 and R_x = 0.856/0,24 = 3.57
R_x = poly(0,'R_x'); // Defining a polynomial with variable 'R_x' with root at 0
a = 0.24 ; // coefficient of x^2
b = -0.856 ; // coefficient of x
c = 0 ; // constant

// Roots of p
R_x1 = ( -b + sqrt (b^2 -4*a*c ) ) /(2* a);
R_x2=( -b - sqrt (b^2 -4*a*c ) ) /(2* a);
// Consider R_x>0 value ,
R_x = R_x1;

R_T = R_r + R_x ; // Total rotor resistance in ohm

// case b
Z_T = R_T + %i*X_lr ; // Total impedance in ohm
Z_T_m = abs(Z_T);//Z_T_m = magnitude of Z_T in ohm
Z_T_a = atan(imag(Z_T) /real(Z_T))*180/%pi;//Z_T_a= phase angle of Z_T in degrees
```

```
cos_theta = R_T / Z_T_m ; // Rotor PF that will
    produce the same starting torque

// case c
Z_r = Z_T_m ; // Impedance in ohm
I_r = E_lr / Z_r ; // Starting current in A

// Display the results
disp("Solution : ");

printf(" \n a: T_o = %.2f * K_n_t ",T_o );
printf(" \n    T_n = %.2f * K_n_t \n",T_o );
printf(" \n    Simplyifing yields");
printf(" \n    0.3 + R_x = 0.24[(0.3+R_x)^2 + (1.08)
    ^2]");
printf(" \n    Expanding and combining the terms
    yields");
printf(" \n    0.24*(R_x)^2 - 0.856*R_x = 0");
printf(" \n    This is a quadratic equation having
    two roots,which may be factored as");
printf(" \n    R_x*(0.24*R_x - 0.856) = 0,yielding")
    ;
printf(" \n    R_x = 0 ohm and R_x = 0.856/0.24 =
    3.57 ohm\n\n    This proves that ");
printf(" \n    Original torque is produced with an
    external resistance of either ");
printf(" \n    zero or 12 times the origianl rotor
    resistance.Therefore,\n");
printf(" \n    R_T = R_r + R_x = %.2f ohm \n",R_T);

printf(" \n b: Z_T in ohm = ");disp(Z_T);
printf(" \n    Z_T = %.2f <%.1f ohm ",Z_T_m,Z_T_a);
printf(" \n    cos  = R_T / Z_T = %.3f or \n
    cos  = cosd(%.1f) = %.3f\n",cos_theta,Z_T_a,cosd
    (Z_T_a));

printf(" \n c: I_r = E_lr / Z_r = %.f A \n\n    This
```

```
    proves that,",I_r);
printf("\n    Rotor current at starting is now only
    28 percent of the original");
printf("\n    starting current in part(a) of Ex.9-7
    ");
```

Scilab code Exa 9.9 calculate Sr with added Rx

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS)
    DYNAMOS
// Example 9-9

clear; clc; close; // Clear the work space and
    console.

// Given data
P = 8 ; // Number of poles in the SCIM
f = 60 ; // Frequency in Hz
S_r = 875 ; // Full-load Speed in rpm with rotor
    short-circuited
R_r = 0.3 ; // rotor resistance per phase in ohm/
    phase
R_x = 0.7 ; // Added resistance in ohm/phase
R_x_a = 1.7 ; // Added resistance in ohm/phase (case
    a)
R_x_b = 2.7 ; // Added resistance in ohm/phase (case
    b)
R_x_c = 3.7 ; // Added resistance in ohm/phase (case
    c)
R_x_d = 4.7 ; // Added resistance in ohm/phase (case
```

```
    d)

// Calculations
S = (120*f)/P ; // Speed in rpm of the rotating
    magnetic field
s_o = (S - S_r)/S ; //  Slip at rotor speed 875 rpm

// case a
s_r_a = s_o * (R_r + R_x_a)/R_r; // Rated slip
S_r_a = S * (1 - s_r_a); // Full-load speed in rpm
    for added resistance R_x_a

// case b
s_r_b = s_o * (R_r + R_x_b)/R_r; // Rated slip
S_r_b = S * (1 - s_r_b); // Full-load speed in rpm
    for added resistance R_x_b

// case c
s_r_c = s_o * (R_r + R_x_c)/R_r; // Rated slip
S_r_c = S * (1 - s_r_c); // Full-load speed in rpm
    for added resistance R_x_c

// case d
s_r_d = s_o * (R_r + R_x_d)/R_r; // Rated slip
S_r_d = S * (1 - s_r_d); // Full-load speed in rpm
    for added resistance R_x_d

// Display the results
disp("Example 9-9 Solution : ");

printf(" \n Slip s_r = s_o*(R_r+R_x)/R_r \n Rotor
    speed S_r = S_o*(1-s)\n");

printf(" \n    Calculated value of s_o = %f ,
    instead of 0.0278(textbook)",s_o)
printf(" \n    so slight variations in the answers
    below.\n");

```

```
printf(" \n a: When R_x = %.1f ohm ",R_x_a);
printf(" \n     s_r = %.3f \n     S_r = %.1f rpm \n",s_r_a,S_r_a);

printf(" \n b: When R_x = %.1f ohm ",R_x_b);
printf(" \n     s_r = %.3f \n     S_r = %.1f rpm \n",s_r_b,S_r_b);

printf(" \n c: When R_x = %.1f ohm ",R_x_c);
printf(" \n     s_r = %.3f \n     S_r = %.1f rpm \n",s_r_c,S_r_c);

printf(" \n d: When R_x = %.1f ohm ",R_x_d);
printf(" \n     s_r = %.3f \n     S_r = %.1f rpm \n",s_r_d,S_r_d);

printf(" \n     This example, verifies that slip is proportional to rotor resistance");
printf(" \n     as summarized below.");

printf(" \n -----------------------------------------------------------------------");
printf(" \n     R_T(ohm) = R_r+R_x \t\t Slip \t\t Full-load Speed(rpm)");
printf(" \n -----------------------------------------------------------------------");
printf(" \n       Given \t\t\t Given \t\t Given \t\ ");
printf(" \n       0.3 \t\t\t 0.0278 \t 875 ");
printf(" \n       0.3+0.1 = 1.0 \t\t 0.0926 \t 817");
printf(" \n -----------------------------------------------------------------------");
printf(" \n       Given \t\t\t Calculated \t Calculated \t\ ");
printf(" \n    a. %.1f + %.1f = %.1f \t\t %.3f \t\t %
```

```
    .1 f ",R_r,R_x_a,R_r+R_x_a,s_r_a,S_r_a);
printf(" \n    b. %.1f + %.1f = %.1f \t\t %.3f \t\t %
    .1 f ",R_r,R_x_b,R_r+R_x_b,s_r_b,S_r_b);
printf(" \n    c. %.1f + %.1f = %.1f \t\t %.3f \t\t %
    .1 f ",R_r,R_x_c,R_r+R_x_c,s_r_c,S_r_c);
printf(" \n    d. %.1f + %.1f = %.1f \t\t %.3f \t\t %
    .1 f ",R_r,R_x_d,R_r+R_x_d,s_r_d,S_r_d);
printf(" \n
    ----------------------------------------------------------------------
    ");
```

Scilab code Exa 9.10 calculate Elr Ir Pin RCL RPD torques

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS)
    DYNAMOS
// Example 9-10

clear; clc; close; // Clear the work space and
    console.

// Given data
P = 4 ; // Number of poles in WRIM
f = 60 ; // Frequency in Hz
V = 220 ; // Line voltage in volt
V_p = 220 ; // Phase voltage in volt (delta
    connection)
hp_WRIM = 1 ; // Power rating of WRIM in hp
S_r = 1740 ; // Full-load rated speed in rpm
R_r = 0.3 ; // rotor resistance per phase in ohm/
    phase
```

```
R_x = 0.7 ; // Added resistance in ohm/phase
X_lr =1 ; // Locked rotor reactance in ohm

// Calculations
S = (120*f)/P ; // Speed in rpm of the rotating
    magnetic field
// case a
E_lr = V_p / 4 ; // Locked-rotor voltage per phase

// case b
s = ( S - S_r)/S ; // slip
I_r = E_lr / sqrt( (R_r/s)^2 + (X_lr)^2 ); // Rotor
    current per phase at rated speed

// case c
P_in = ((I_r)^2 * R_r)/s ; // Rated rotor power
    input per phase

// case d
P_RL = (I_r)^2 * R_r ; // Rated copper loss per
    phase

// case e
P_d_W = P_in - P_RL ; // Rotor power developed per
    phase in W
P_d_hp = P_d_W/746 ; // Rotor power developed per
    phase in hp

// case f
hp = P_d_hp ; // Rotor power developed per phase in
    hp
T_d1 = (hp*5252)/S_r ; // Rotor torque developed in
    lb-ft per phase by method 1
T_d2 = 7.04*(P_in/S) ; // Rotor torque developed in
    lb-ft per phase by method 2

T_dm = 3*T_d1 ; // Total rotor torque in lb-ft

```

```
// Display the results
disp("Example 9-10 Solution : ");
printf(" \n a: Locked-rotor voltage per phase : \n
       E_lr = %d V \n ",E_lr);

printf(" \n b: slip : \n    s = %.2f \n",s);
printf(" \n    Rotor current per phase at rated
   speed:\n     I_r = %.3f A/phase \n ",I_r);

printf(" \n c: Rated rotor power input per phase :\n
        P_in = %d W/phase \n ",P_in);

printf(" \n d: Rated copper loss per phase : \n
   P_RL = %.2f W \n ",P_RL);

printf(" \n e: Rotor power developed per phase in W
   :\n     P_d = %.1f W/phase ",P_d_W);
printf(" \n\n    Rotor power developed per phase in
   hp :\n     P_d = %.2f hp/phase \n ",P_d_hp);

printf(" \n f: Rotor torque developed in lb-ft per
   phase :\n     T_d = %.1f lb-ft  (method 1)",T_d1);
printf(" \n\n    T_d = %.1f lb-ft  (method 2)",T_d2)
   ;
printf(" \n\n    Total rotor torque : \n    T_dm = %
   .1f lb-ft )\n ",T_dm);
```

Scilab code Exa 9.11 calculate Elr Ir Pin RCL RPD and torques

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS)
```

```
DYNAMOS
// Example 9-11

clear; clc; close; // Clear the work space and console.

// Given data
// 3-phase WRIM
V_L = 208 ; // Voltage rating of the WRIM in volt
P = 6 ; // Number of poles in WRIM
f = 60 ; // Frequency in Hz
P_o = 7.5 ; // Power rating of WRIM in hp
S_r = 1125 ; // Full-load rotor speed in rpm
R_r = 0.08 ; // Rotor resistance in ohm/phase
X_lr = 0.4 ; // Locked rotor resistance in ohm/phase

// Calculations
S = (120*f)/P ; // Speed in rpm of the rotating magnetic field
// case a
E_lr = (V_L / sqrt(3))/2 ; // Locked rotor voltage per phase

// case b
s = (S - S_r)/S ; // Full-load rated slip
I_r = E_lr / sqrt( (R_r/s)^2 + (X_lr)^2 ); // Rotor current in A per phase at rated speed

// case c
P_in = ( (I_r)^2 * R_r )/s ; // Rated rotor power input per phase in (W/phase)

// case d
P_RL = ( (I_r)^2 * R_r ); // Rated rotor copper loss per phase (in W/phase)

// case e
// Subscript W in P_d indicates calculating P_d in W
```

```
P_d_W = P_in - P_RL ; // Rotor power developed per phase (in W/phase)
// Subscript hp in P_d indicates calculating P_d in hp
P_d_hp = P_d_W/746 ; // Rotor power developed per phase (in hp/phase)

// case f
// subscript 1 in T_d indicates method 1 for calculating T_d
hp = P_d_hp ;
T_d1 = (hp*5252)/S_r ; // Rotor torque developed per phase in lb-ft

// subscript 2 in T_d indicates method 2 for calculating T_d
T_d2 = 7.04*(P_in/S); // Rotor torque developed per phase in lb-ft

// case g
T_dm = 3*T_d1 ; // Total rotor torque in lb-ft

// case h
T_o = 7.04*(P_o*746)/S_r ; // Total output rotor torque in lb-ft

// Display the results
disp("Example 9-11 Solution : ");

printf(" \n    Note: Slight variations in the answers I_r,P_in,P_RL,P_d,T_d ");
printf(" \n          are because of non-approximation of E_lr and (R_r/s)^2 + (X_lr)^2");
printf(" \n          while calulating in scilab.\n");

printf(" \n a: Locked rotor voltage per phase :\n E_lr = %d V\n",E_lr);
```

```
printf(" \n b: slip :\n    s = %.4f ",s);
printf(" \n\n    Rotor current per phase at rated
   speed :\n    I_r = %.2f A/phase\n",I_r);

printf(" \n c: Rated rotor power input per phase :\n
       P_in = %.f W/phase\n",P_in);

printf(" \n d: Rated rotor copper loss per phase :\n
       P_RL = %.1f W/phase\n",P_RL);

printf(" \n e: Rotor power developed per phase ");
printf(" \n    P_d = %.f W/phase \n    P_d = %.2f hp
   /phase\n",P_d_W,P_d_hp);

printf(" \n f: Rotor torque developed per phase : ")
   ;
printf(" \n    (method 1)\n    T_d = %.1f lb-ft/
   phase",T_d1);
printf(" \n\n    (method 2)\n    T_d = %.1f lb-ft/
   phase\n",T_d2);

printf(" \n g: Total rotor torque : \n    T_dm = %d
   lb-ft\n",T_dm);

printf(" \n h: Total output rotor torque : \n    T_o
    = %d lb-ft",T_o);
```

Scilab code Exa 9.12 calculate s and Sr for Tmax

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

```

```
// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS)
   DYNAMOS
// Example 9-12

clear; clc; close; // Clear the work space and
   console.

// Given data as per Ex.9-10
P = 4 ; // Number of poles in WRIM
f = 60 ; // Frequency in Hz
V = 220 ; // Line voltage in volt
V_p = 220 ; // Phase voltage in volt (delta
   connection)
hp_WRIM = 1 ; // Power rating of WRIM in hp
S_r = 1740 ; // Full-load rated speed in rpm
R_r = 0.3 ; // rotor resistance per phase in ohm/
   phase
R_x = 0.7 ; // Added resistance in ohm/phase
X_lr = 1 ; // Locked rotor reactance in ohm

// Calculations from Ex.9-10
E_lr = V_p / 4 ; // Locked-rotor voltage per phase
S = (120*f)/P ; // Speed in rpm of the rotating
   magnetic field

// Calculations (Ex.9-12)
P_in = (E_lr)^2 / (2*X_lr); // rotor power input(RPI
   ) in W/phase
P_in_total = P_in * 3 ; // Total 3-phase rotor power
    input(RPI) in W

T_max = 7.04*(P_in_total/S); // Maximum torque
   developed in lb-ft

s_b = R_r / X_lr ; // Slip

s = s_b;
S_r = S*(1 - s); // Rotor speed in rpm for T_max
```

```

// Display the results
disp("Example 9-12 Solution : ");

printf(" \n Rotor power input (RPI) per phase is : ");
printf(" \n P_in = %.1f W/phase \n",P_in);

printf(" \n The total 3-phase rotor power input (RPI) is : ");
printf(" \n P_in = %.1f W\n",P_in_total);

printf(" \n Substituting in Eq.(9-19),\n T_max = %.2f lb-ft\n",T_max);
printf(" \n Then, s_b = %.1f \n and S_r = %d rpm",s_b,S_r);
```

Scilab code Exa 9.13 calculate starting torque

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS) DYNAMOS
// Example 9-13

clear; clc; close; // Clear the work space and console.

// Given data as per Ex.9-10
P = 4 ; // Number of poles in WRIM
f = 60 ; // Frequency in Hz
V = 220 ; // Line voltage in volt
```

```
V_p = 220 ; // Phase voltage in volt (delta connection)
hp_WRIM = 1 ; // Power rating of WRIM in hp
S_r = 1740 ; // Full-load rated speed in rpm
R_r = 0.3 ; // rotor resistance per phase in ohm/phase
R_x = 0.7 ; // Added resistance in ohm/phase
X_lr = 1 ; // Locked rotor reactance in ohm

// Calculations
E_lr = V_p / 4 ; // Locked-rotor voltage per phase
S = (120*f)/P ; // Speed in rpm of the rotating magnetic field

// Total 3-phase rotor power input(RPI) in W
P_in = 3 * ( (E_lr)^2 ) / ( (R_r)^2 + (X_lr)^2 ) * R_r ;

T_s = 7.04 * (P_in/S); // Starting torque developed in lb-ft

// Display the results
disp("Example 9-13 Solution : ");

printf(" \n P_in = %.f W \n",P_in);
printf(" \n From Eq.(9-19),starting torque is : \n T_s = %.2f lb-ft",T_s);
```

Scilab code Exa 9.14 calculate full load and starting torques

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

```

```
// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS)
   DYNAMOS
// Example 9-14

clear; clc; close; // Clear the work space and
   console.

// Given data
T_max = 17.75 ; // Maximum torque developed in lb-ft
s_max = 0.3 ; // Slip for which T_max occurs
s_a = 0.0333 ; // slip (case a)
s_b = 1.0 ; // slip (case b)

// Calculations
// Subscript a in T indicates case a
T_a = T_max * ( 2 / ((s_max/s_a) + (s_a/s_max)) );
   // Full-load torque in lb-ft

// Subscript b in T indicates case b
T_b = T_max * ( 2 / ((s_max/s_b) + (s_b/s_max)) );
   // Starting torque in lb-ft

// Display the results
disp("Example 9-14 Solution : ");

printf(" \n a: Full-load torque at slip = %.4f \n
      T = %.1f lb-ft\n",s_a,T_a);

printf(" \n b: Starting torque at slip = %.1f \n
   T = %.2f lb-ft\n",s_b,T_b);
```

Scilab code Exa 9.15 calculate Ip Ir PF SPI SCL RPI RPD and rotor power and torque and hp and motor efficiency

```
// Electric Machinery and Transformers
```

```
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS) DYNAMOS
// Example 9-15

clear; clc; close; // Clear the work space and console.

// Given data
// 3-phase Y-connected SCIM
P = 4 ; // Number of poles in SCIM
S_r = 1746 ; // Rated rotor speed in rpm
V = 220 ; // Voltage rating of SCIM in volt
f = 60 ; // Frequency in Hz
P_hp = 10 ; // power rating of SCIM in hp
R_a = 0.4 ; // Armature resistance in ohm
R_r = 0.14 ; // Rotor resistance in ohm
jXm = 16 ; // Reactance in ohm
jXs = 0.35 ; // Synchronous reactance in ohm
jXlr = 0.35 ; // Locked rotor reactance in ohm
P_r_total = 360 ; // Total rotational losses in W

// Calculations
V_p = V / sqrt(3); // Voltage per phase in volt

S = (120*f)/P ; // Speed in rpm of the rotating magnetic field
// preliminary calculations
s = ( S - S_r)/S ; // slip

disp("Example 9-15 :");

printf(" \n From Fig.9-13,using the format method of mesh analysis ,we may");
printf(" \n write the array by inspection :\n");
```

```
printf(" \n
    -------------------------------------------------------------
    ");
printf(" \n     I_1(A) \t\t  I_2(A) \t\t  V(volt)");
printf(" \n
    -------------------------------------------------------------
    ");
printf(" \n   (0.4 + j16.35) \t  -(0 + j16) \t\t  (127
     + j0)");
printf(" \n   -(0 + j16) \t\t  (4.67 + j16.35) \t
        0");
printf(" \n
    -------------------------------------------------------------
    ");

A = [ (0.4 + %i*16.35) -%i*16 ; (-%i*16)  (4.67 + %i
    *16.35) ]; // Matrix containing above mesh eqns
    array
delta = det(A); // Determinant of A

// case a : Stator armature current I_p in A
I_p = det( [ (127+%i*0) (-%i*16) ; 0 (4.67 + %i
    *16.35) ] ) / delta ;
I_p_m = abs(I_p);//I_p_m=magnitude of I_p in A
I_p_a = atan(imag(I_p) /real(I_p))*180/%pi;//I_p_a=
    phase angle of I_p in degrees
I_1 = I_p ; // Stator armature current in A

// case b : Rotor current I_r per phase in A
I_r = det( [ (0.4 + %i*16.35) (127+%i*0) ; (-%i*16)
    0 ] ) / delta ;
I_r_m = abs(I_r);//I_r_m=magnitude of I_r in A
I_r_a = atan(imag(I_r) /real(I_r))*180/%pi;//I_r_a=
    phase angle of I_r in degrees

// case c
theta_1 = I_p_a ; // Motor PF angle in degrees
cos_theta1 = cosd(theta_1); // Motor PF
```

```
// case d
I_p = I_p_m ; // Stator armature current in A
SPI = V_p * I_p * cos_theta1 ; // Stator Power Input in W

// case e
SCL = (I_p)^2 * R_a ; // Stator Copper Loss in W

// case f
// Subscripts 1 and 2 for RPI indicates two methods of calculating RPI
RPI_1 = SPI - SCL ; // Rotor Power Input in W
RPI_2 = (I_r_m)^2 * (R_r/s); // Rotor Power Input in W
RPI =RPI_1 ;

// case g
// Subscripts 1 , 2 and 3 for RPD indicates three methods of calculating RPD
RPD_1 = RPI * ( 1 - s ); // Rotor Power Developed in W
RCL = s*(RPI); // Rotor copper losses in W
RPD_2 = RPI - RCL ; // Rotor Power Developed in W
RPD_3 = (I_r_m)^2 * R_r * ((1-s)/s); // Rotor Power Developed in W
RPD = RPD_1 ;

// case h
P_r = P_r_total / 3 ; // Rotational Losses per phase in W
P_o = RPD - P_r ; // Rotor power per phase in W
P_to = 3*P_o ; // Total rotor power in W

// case i
T = 7.04 * (P_to/S_r); // Total 3-phase torque in lb-ft

```

```
// case j
P_t = P_to ;
hp = P_t / 746 ; // Output horsepower

// case k
P_in = SPI ; // Input power to stator in W
eta = P_o / P_in * 100 ; // Motor efficiency at rated load

// Display the results
disp("Solution : ");
printf(" \n Preliminary calculations\n");
printf(" \n Slip : s = %.2f \n R_r/s = %.2f ohm \n", s,R_r/s);

printf(" \n Determinant    = ");disp(delta);

printf(" \n a: Stator armature current :\n     I_p in A = ");disp(I_1);
printf(" \n     I_p = I_1 = %.2f <%.2f A \n ",I_p_m , I_p_a );

printf(" \n b: Rotor current per phase :\n     I_r in A = ");disp(I_r);
printf(" \n     I_r = I_2 = %.3f <%.2f A \n ",I_r_m , I_r_a );

printf(" \n c: Motor PF :\n     cos  1 = %.4f \n", cos_theta1);

printf(" \n d: Stator Power Input :\n     SPI = %d W \n",SPI);

printf(" \n e: Stator Copper Loss :\n     SCL = %.f W \n",SCL);

printf(" \n f: Rotor Power Input :\n     RPI = %d W(method 1) ", RPI_1);
```

```
printf(" \n     RPI = %.f W (method 2)\n",RPI_2);
printf(" \n     Note: RPI calculated by 2nd method
    slightly varies from that of");
printf(" \n             textbook value because of non-
    approximation of I_r while");
printf(" \n             calculating in scilab.\n")

printf(" \n g: Rotor Power Developed :\n     RPD = %.
    f W \n",RPD_1);
printf(" \n     Rotor copper loss :\n     RCL = %d W\n
    ",RCL);
printf(" \n     RPD = %.f W \n      RPD = %d W \n ",
    RPD_2,RPD_3);

printf(" \n h: Rotor power per phase :\n     P_o/    =
     %f W/    ",P_o);
printf(" \n\n     Total rotor power:\n     P_to = %f W
     \n",P_to);
printf(" \n     Above P_o/    and P_to values are not
    approximated while calculating in ");
printf(" \n     SCILAB.So,they vary slightly from
    textbook values.\n");

printf(" \n i: Total 3-phase output torque :\n     T
    = %.f lb-ft\n",T);

printf(" \n j: Output horsepower : \n     hp = %.1f
    hp \n",hp);

printf(" \n k: Motor efficiency at rated load :\n
            = %.1f percent \n",eta)

printf(" \n Power flow diagram (per phase)\n");
printf(" \n    SPI-----------> RPI----------> RPD
    -----------> P_o");
printf(" \n (%d W)     |    (%d W)    |     (%d W)     |     (%d
     W)",SPI,RPI_1,RPD_3,P_o);
printf(" \n                  |                |                  |
```

```
");
printf(" \n                    SCL                  RCL                    P_r");
printf(" \n                 (%.f W)              (%d W)                 (%d W)",SCL,RCL,P_r);
```

Scilab code Exa 9.16 calculate Ism IL Ts and percent IL and percent Ts

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS) DYNAMOS
// Example 9-16

clear; clc; close; // Clear the work space and console.

// Given data
// three-phase SCIM
V = 208 ; // Rated voltage in volt
P_o = 15 ; // Rated power in hp
I = 42 ; // Rated current in A
I_st = 252 ; // Starting current in A
T_st = 120 ; // Full-voltage starting torque in lb-ft
tap = 60*(1/100) ; // Tapping in % employed by compensator

// Calculations
// case a
I_sm = tap * I_st ; // Motor starting current in A at reduced voltage
```

```
// case b
I_L = tap * I_sm ; // Motor line current in A(neglecting tarnsformer exciting
// current and losses)

// case c
T_s = (tap)^2 * T_st ; // Motor starting torque at reduced voltage in lb-ft

// case d
percent_I_L = I_L / I_st * 100 ; // Percent line current at starting

// case e
percent_T_st = T_s / T_st * 100 ; // Percent motor starting torque

// Display the results
disp("Example 9-16 Solution : ");

printf(" \n a: Motor starting current at reduced voltage : ");
printf(" \n    I_sm = %.1f A to the motor.\n",I_sm);

printf(" \n b: Motor line current neglecting tarnsformer exciting current and losses :");
printf(" \n    I_L = %.2f A drawn from the mains.\n",I_L);

printf(" \n c: Motor starting torque at reduced voltage :\n    T_s = %.1f lb-ft\n",T_s);

printf(" \n d: Percent line current at starting : ");
printf(" \n    = %.f percent of line current at full voltage.\n",percent_I_L);

```

```
printf(" \n e: Percent motor starting torque : ");
printf(" \n    = %d percent of starting torque at
   full voltage.\n",percent_T_st);
```

Scilab code Exa 9.17 calculate T s Sr for different V

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS)
   DYNAMOS
// Example 9-17

clear; clc; close; // Clear the work space and
   console.

// Given data
// three-phase SCIM
V_o = 220 ; // Rated voltage in volt
P = 4 ; // Number of poles in SCIM
P_o = 10 ; // Rated power in hp
f = 60 ; // Frequency in Hz(assume,not given)
T_o = 30 ; // Rated torque in lb-ft
S_r = 1710 ; // Rated rotor speed in rpm
V_n1 = 242 ; // Impressed stator voltage in volt(
   case a)
V_n2 = 198 ; // Impressed stator voltage in volt(
   case b)

// Calculations
S = (120*f)/P ; // Speed in rpm of the rotating
   magnetic field
// case a : Impressed stator voltage = 242 V
```

```
s_o = (S - S_r)/S ; // Rated slip

T_n1 = T_o * (V_n1/V_o)^2 ; // New torque in lb-ft

s_n1 = s_o * (T_o/T_n1); // New slip

S_rn1 = S*(1 - s_n1);

// case b : Impressed stator voltage = 198 V
T_n2 = T_o * (V_n2/V_o)^2 ; // New torque in lb-ft

s_n2 = s_o * (T_o/T_n2); // New slip

S_rn2 = S*(1 - s_n2);

// case c
// Subscript a in percent_slip and percent_speed
   indicates part a
percent_slip_a = (s_o - s_n1)/s_o * 100 ; // Percent
    change in slip in part(a)

percent_speed_a = (S_rn1 - S_r)/S_r * 100; //
   Percent change in speed in part(a)

// case d
// Subscript b in percent_slip and percent_speed
   indicates part b
percent_slip_b = (s_n2 - s_o)/s_o * 100 ; // Percent
    change in slip in part(b)

percent_speed_b = (S_r - S_rn2)/S_r * 100; //
   Percent change in speed in part(b)

// Display the results
disp("Example 9-17 Solution : ");

printf(" \n a: Rated slip :\n    s = %.2f\n",s_o);
printf(" \n    For impressed stator voltage = %d V \
```

```
n ",V_n1);
printf(" \n     New torque :\n     T_n = %.1f lb-ft \n ",T_n1);
printf(" \n     New slip :\n     s_n = %f \n ",s_n1);
printf(" \n     New rotor speed :\n     S_r = %f rpm \n",S_rn1);

printf(" \n b: For impressed stator voltage = %d V \n ",V_n2);
printf(" \n     New torque :\n     T_n = %.1f lb-ft \n ",T_n2);
printf(" \n     New slip :\n     s_n = %f \n ",s_n2);
printf(" \n     New rotor speed :\n     S_r = %f rpm \n",S_rn2);

printf(" \n c: Percent change in slip in part(a)");
printf(" \n     = %.1f percent decrease.\n",percent_slip_a);
printf(" \n     Percent change in speed in part(a)");
printf(" \n     = %.2f percent increase \n",percent_speed_a);

printf(" \n d: Percent change in slip in part(b)");
printf(" \n     = %.2f percent increase.\n",percent_slip_b);
printf(" \n     Percent change in speed in part(b)");
printf(" \n     = %.2f percent decrease\n",percent_speed_b);

printf(" \n    SLIGHT VARIATIONS IN PERCENT CHANGE IN SLIP AND SPEED ARE DUE TO");
printf(" \n    NON-APPROXIMATION OF NEW SLIPS AND NEW SPEEDS CALCULATED IN SCILAB.")
```

Scilab code Exa 9.18 calculate T s Sr for different impressed stator V

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS)
   DYNAMOS
// Example 9-18

clear; clc; close; // Clear the work space and
   console.

// Given data
// three-phase WRIM
V_o = 220 ; // Rated voltage in volt
P_o = 10 ; // Rated power in hp
P = 4 ; // Number of poles in WRIM(assumption)
f = 60 ; // Frequency in Hz(assume,not given)
R_ro = 0.3 ; // Rotor resistance in ohm
T_o = 30 ; // Rated torque in lb-ft
S_r = 1750 ; // Rated rotor speed in rpm
R_r_ext = 1.7 ; // External rotor resistance in ohm/
   phase inserted in the rotor ckt
R_rn = R_ro + R_r_ext ; // Total rotor resistance in
    ohm

V_n1 = 240 ; // Impressed stator voltage in volt(
   case a)
V_n2 = 208 ; // Impressed stator voltage in volt(
   case b)
V_n3 = 110 ; // Impressed stator voltage in volt(
   case c)

// Calculations
S = (120*f)/P ; // Speed in rpm of the rotating
   magnetic field

// case a : Impressed stator voltage = 240 V
```

```
s_o = (S - S_r)/S ; // Rated slip

T_n1 = T_o * (V_n1/V_o)^2 ; // New torque in lb-ft

s_n1 = s_o * (T_o/T_n1) * (R_rn/R_ro); // New slip

S_rn1 = S*(1 - s_n1);

// case b : Impressed stator voltage = 208 V
T_n2 = T_o * (V_n2/V_o)^2 ; // New torque in lb-ft

s_n2 = s_o * (T_o/T_n2) * (R_rn/R_ro); // New slip

S_rn2 = S*(1 - s_n2);

// case c : Impressed stator voltage = 110 V
T_n3 = T_o * (V_n3/V_o)^2 ; // New torque in lb-ft

s_n3 = s_o * (T_o/T_n3) * (R_rn/R_ro); // New slip

S_rn3 = S*(1 - s_n3);

// Display the results
disp("Example 9-18 Solution : ");

printf(" \n a: Rated slip :\n    s = %f\n",s_o);
printf(" \n    For impressed stator voltage = %d V \n ",V_n1);
printf(" \n    New torque :\n    T_n = %.1f lb-ft \n ",T_n1);
printf(" \n    New slip :\n    s_n = %f \n ",s_n1);
printf(" \n    New rotor speed :\n    S_r = %f rpm \n",S_rn1);

printf(" \n b: For impressed stator voltage = %d V \n ",V_n2);
printf(" \n    New torque :\n    T_n = %.2f lb-ft \n ",T_n2);
```

```
printf(" \n     New slip :\n      s_n = %f \n ",s_n2);
printf(" \n     New rotor speed :\n      S_r = %f rpm \
   n",S_rn2);

printf(" \n c: For impressed stator voltage = %d V \
   n ",V_n3);
printf(" \n     New torque :\n      T_n = %.1f lb-ft \n
    ",T_n3);
printf(" \n     New slip :\n      s_n = %f \n ",s_n3);
printf(" \n     New rotor speed :\n      S_r = %f rpm \
   n",S_rn3);
```

Scilab code Exa 9.19 calculate fcon and Scon

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 9: POLYPHASE INDUCTION (ASYNCHRONOUS)
   DYNAMOS
// Example 9-19

clear; clc; close; // Clear the work space and
   console.

// Given data
P = 8 ; // Number of poles in WRIM
f = 60 ; // Operating frequency of the WRIM in Hz
/// WRIM is driven by variable-speed prime mover as
   a frequency changer
S_con_a1 = 1800 ; // Speed of the convertor in rpm
S_con_a2 = 450 ; // Speed of the convertor in rpm

f_con_b1 = 25 ; // Frequency of an induction
```

```
    converter in Hz
f_con_b2 = 400 ; // Frequency of an induction
    converter in Hz
f_con_b3 = 120 ; // Frequency of an induction
    converter in Hz

// Calculations
S = (120*f)/P ; // Speed in rpm of the rotating
    magnetic field

// case a
// Subscript a1 in f_con indicates case a 1st
    frequecy in Hz
f_con_a1 = f*(1 + S_con_a1/S); // Frequency of an
    induction converter in Hz

// Subscript a2 in f_con indicates case a 2nd
    frequency in Hz
f_con_a2 = f*(1 - S_con_a2/S); // Frequency of an
    induction converter in Hz

// case b
// Subscript b1 in S-con indicates case b 1st speed
    of converter in rpm
S_con_b1 = ( -1 + f_con_b1/f) * S ; // Speed of the
    convertor in rpm

// Subscript b2 in S-con indicates case b 2nd speed
    of converter in rpm
S_con_b2 = ( -1 + f_con_b2/f) * S ; // Speed of the
    convertor in rpm

// Subscript b3 in S-con indicates case b 3rd speed
    of converter in rpm
S_con_b3 = ( -1 + f_con_b3/f) * S ; // Speed of the
    convertor in rpm


```

```
// Display the results
disp("Example 9-19 Solution : ");

printf(" \n Using Eq.(9-26),\n");

printf(" \n a: f_con = %d Hz for %d rpm in opposite
    direction\n",f_con_a1,S_con_a1);
printf(" \n    f_con = %d Hz for %d rpm in same
    direction\n",f_con_a2,S_con_a2);

printf(" \n b: 1. S_con = %.f rpm, or %.f rpm in
    same direction.\n",S_con_b1,abs(S_con_b1));
printf(" \n    2. S_con = %d rpm in opposite
    direction.\n",S_con_b2);
printf(" \n    3. S_con = %d rpm in opposite
    direction to rotating stator flux.\n",S_con_b3);
```

Chapter 10

SINGLE PHASE MOTORS

Scilab code Exa 10.1 calculate total starting current and PF and components of Is Ir and phase angle between Is Ir

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 10: SINGLE-PHASE MOTORS
// Example 10-1

clear; clc; close; // Clear the work space and
    console.

// Given data
hp = 0.25 ; // Power rating of the single-phase
    motor in hp
V = 110 ; // Voltage rating of the single-phase
    motor in V
I_sw = 4 ; // Starting winding current
phi_I_sw = 15 ; // Phase angle in degrees by which
    I_sw lags behind V
I_rw = 6 ; // Running winding current
```

```
phi_I_rw = 40 ; // Phase angle in degrees by which I_rw lags behind V

// Calculations
// case a
I_s = I_sw * exp( %i * -phi_I_sw*(%pi/180) ); // starting current in A
// (%pi/180) for degrees to radians conversion of phase angle
I_s_m = abs(I_s);//I_s_m = magnitude of I_s in A
I_s_a = atan(imag(I_s) /real(I_s))*180/%pi;//I_s_a= phase angle of I_s in degrees

I_r = I_rw * exp( %i * -phi_I_rw*(%pi/180) ); // running current in A
I_r_m = abs(I_r);//I_r_m = magnitude of I_r in A
I_r_a = atan(imag(I_r) /real(I_r))*180/%pi;//I_r_a= phase angle of I_r in degrees

I_t = I_s + I_r ; // Total starting current in A
I_t_m = abs(I_t);//I_t_m = magnitude of I_t in A
I_t_a = atan(imag(I_t) /real(I_t))*180/%pi;//I_t_a= phase angle of I_t in degrees
Power_factor = cosd(I_t_a); // Power factor

// case b
Is_cos_theta = real(I_s); // Component of the starting winding current in phase
// with the supply voltage in  A

// case c
Ir_sin_theta = imag(I_r); // Component of the running winding current that lags
// the supply voltage by 90 degrees

// case d
phase = ( phi_I_rw - phi_I_sw ); // Phase angle between the starting and running
```

```
// currents in degrees

// Display the results
disp("Example 10-1 Solution : ");
printf(" \n a: I_s = %d <-%d A ", I_sw , phi_I_sw );
printf(" \n    I_s in A =  " );disp(I_s);
printf(" \n    I_r = %d <-%d A ", I_rw , phi_I_rw );
printf(" \n    I_r in A =  " );disp(I_r);
printf(" \n    I_t in A =  " );disp(I_t);
printf(" \n    I_t = %.2f <%d A ", I_t_m , I_t_a );
printf(" \n\n     Power factor = cos(%d) = %.3f lagging \n", I_t_a ,Power_factor);

printf(" \n b: Is*cos  = %.2f A  (from a)\n ", Is_cos_theta );

printf(" \n c: (from a),\n     Ir*sin  in A = " ); disp(%i*Ir_sin_theta);

printf(" \n d: ( _r -  _s ) = %d degrees ", phase);
```

Scilab code Exa 10.2 calculate Ps Pr Pt and motor efficiency

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 10: SINGLE-PHASE MOTORS
// Example 10-2

clear; clc; close; // Clear the work space and console.

// Given data as per Ex.10-1
```

```
hp = 0.25 ; // Power rating of the single-phase motor in hp
V = 110 ; // Voltage rating of the single-phase motor in V
I_s = 4 ; // Starting winding current
phi_I_s = 15 ; // Phase angle in degrees by which I_s lags behind V
I_r = 6 ; // Running winding current
phi_I_r = 40 ; // Phase angle in degrees by which I_r lags behind V

// Calculations
// case a
P_s = V * I_s * cosd(phi_I_s); // Power dissipated in the starting winding in W

// case b
P_r = V * I_r * cosd(phi_I_r); // Power dissipated in the running winding in W

// case c
P_t = P_s + P_r ; // Total instantaneous power dissipated during starting in W

// case d
P_r_d = P_r ; // Total steady-state power dissipated during running in W

// case e
eta = ( hp * 746 ) / P_r * 100 ; // Motor efficiency in percent

// Display the results
disp("Example 10-2 Solution : ");
printf(" \n a: Power dissipated in the starting winding\n    P_s = %d W \n", P_s );

printf(" \n b: Power dissipated in the running
```

```
winding\n      P_r  = %.1f W \n", P_r );

printf(" \n c: Total instantaneous power dissipated
   during starting\n      P_t  = %.1f W \n", P_t );

printf(" \n d: Total steady-state power dissipated
   during running\n      P_r  = %.1f W \n", P_r_d );

printf(" \n e: Motor efficiency \n          = %.f
   percent \n", eta );
```

Scilab code Exa 10.3 calculate total starting current and sine of angle between Is Ir

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 10: SINGLE-PHASE MOTORS
// Example 10-3

clear; clc; close; // Clear the work space and
   console.

// Given data
hp = 0.25 ; // Power rating of the single-phase
   motor in hp
V = 110 ; // Voltage rating of the single-phase
   motor in V
I_sw = 4 ; // Starting winding current
phi_I_sw = 15 ; // Phase angle in degrees by which
   I_sw lags behind V
I_rw = 6 ; // Running winding current
phi_I_rw = 40 ; // Phase angle in degrees by which
```

```
I_rw lags behind V
// when the capacitor is added to the auxiliary starting winding of the motor
// of Ex.10-1 , I_s leads V by 42 degrees so,
phi_I_sw_new = 42 ; // I_s leads V by phi_I_sw_new degrees

// Calculations
// case a
I_s = I_sw * exp( %i * phi_I_sw_new*(%pi/180) ); // starting current in A
// (%pi/180) for degrees to radians conversion of phase angle
I_s_m = abs(I_s);//I_s_m = magnitude of I_s in A
I_s_a = atan(imag(I_s) /real(I_s))*180/%pi;//I_s_a= phase angle of I_s in degrees

I_r = I_rw * exp( %i * -phi_I_rw*(%pi/180) ); // running current in A
I_r_m = abs(I_r);//I_r_m = magnitude of I_r in A
I_r_a = atan(imag(I_r) /real(I_r))*180/%pi;//I_r_a= phase angle of I_r in degrees

I_t = I_s + I_r ; // Total starting current in A
I_t_m = abs(I_t);//I_t_m = magnitude of I_t in A
I_t_a = atan(imag(I_t) /real(I_t))*180/%pi;//I_t_a= phase angle of I_t in degrees
Power_factor = cosd(I_t_a); // Power factor

// case b
theta = ( phi_I_rw - (-phi_I_sw_new) );
sin_theta = sind(theta);// Sine of the angle between the
// starting and running currents
phase = 25 ; // Phase angle between the starting and running
// currents in degrees (from Ex.10-1)

```

```
// case c
// Ratio of starting torques (capacitor to resistance start)
ratio_T = sind(theta) / sind(phase);

// Display the results
disp("Example 10-3 Solution : ");
printf(" \n a: I_s = %d <%d A ", I_sw , phi_I_sw_new );
printf(" \n    I_s in A =  " );disp(I_s);
printf(" \n    I_r = %d <-%d A ", I_rw , phi_I_rw );
printf(" \n    I_r in A =  " );disp(I_r);
printf(" \n    I_t in A =  " );disp(I_t);
printf(" \n    I_t = %.2f <%.1f A ", I_t_m , I_t_a );
printf(" \n\n    Power factor = cos(%.1f) = %.3f lagging \n", I_t_a ,Power_factor);

printf(" \n b: sin(%d - (-%d)) = sin(%d) = %.4f\n", phi_I_rw,phi_I_sw_new,theta,sin_theta);

printf(" \n c: The steady state starting current has been reduced from");
printf(" \n    9.77 <-30 A to %.2f <%.1f A ,",I_t_m ,I_t_a );
printf(" \n    and the power factor has risen from 0.866 lagging to %.3f.",Power_factor);
printf(" \n    The motor develops maximum starting torque(T = K*I_b*  *cos  ) with");
printf(" \n    minimum starting current.The ratio of starting torques ");
printf(" \n    (capacitor to resistance start) is : \n");
printf(" \n    T_cs/T_rs = sin(%d)/sin(%d) = %.3f", theta,phase,ratio_T)
```

Scilab code Exa 10.4 calculate ratios of T and efficiency and rated PF and hp

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 10: SINGLE-PHASE MOTORS
// Example 10-4

clear; clc; close; // Clear the work space and console.

// Given data (from Table 10-2)
T_r = 1 ; // Rated torque in lb-ft
T_s = 4.5 ; // Starting torque in lb-ft (rfom Locked-Rotor Data)
T_br = 2.5 ; // Breakdown torque in lb-ft (Breakdown-Torque Data)

// Rated Load Data
P = 400 ; // Rated input power in W
V = 115 ; // Rated input voltage in volt
I_t = 5.35 ; // Rated input current in A
Speed = 1750 ; // Rated speed in rpm

// Calculations
// case a
ratio_s_r_T = T_s / T_r ; // Ratio of starting to rated torque

// case b
ratio_s_br_T = T_br / T_r ; // Ratio of breakdown to
```

```
    rated torque

// case c
P_o_hp = 1 / 3 ; // Power output in hp
P_o = P_o_hp * 746 ; // Power output in W
eta = P_o / P * 100 ; // Rated load efficiency

// case d
S = V * I_t ; // VA rating of the motor
cos_theta = P / S ; // Rated load - power factor

// case e
T = 1 ; // Rated load torque in lb-ft
hp = (T*Speed)/5252 ; // Rated load horsepower

// Display the results
disp("Example 10-4 Solution : ");

printf(" \n a: T_s/T_r = %.1f \n ",ratio_s_r_T );

printf(" \n b: T_br/T_r = %.1f \n ",ratio_s_br_T );

printf(" \n c: Rated load efficiency \n        = %.1f
    percent \n ",eta );

printf(" \n d: Rated load power factor\n     cos   =
   %.4f \n ",cos_theta );

printf(" \n e: Rated load horsepower\n     hp = %.4f
   hp ",hp);
```

Chapter 11

SPECIALIZED DYNAMOS

Scilab code Exa 11.1 calculate S V P T A and B from torque speed relations fig

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 11: SPECIALIZED DYNAMOS
// Example 11-1

clear; clc; close; // Clear the work space and
    console.

// Given data
// Torque - speed relations shown in Fig.11-3b for a
    dc servomotor.

// Calculations
// case a
// Extrapolating to load line point x,
S = 800 ; // Motor speed at point x
V = 60 ; // Armature voltage in volt at point x
```

```

// case b
// At standstill , 60 V yields 4.5 lb−ft of starting torque
T = 4.5 ;

// case c
P_c = (T*S)/5252 ; // Power delivered to the load in hp (from case a conditions)
P_c_watt = P_c * 746 ; // P_c in W
// case d
// At point o:
T_d = 1.1 ; // Starting torque in lb−ft (subscript d indicates case d) and
S_d = 410 ; // Motor speed at point at point o

// case e
// At point w:
T_e = 2.4 ; // Starting torque in lb−ft (subscript e indicates case e) and
S_e = 900 ; // Motor speed at point at point w

// case f
P_d = (T_d*S_d)/5252 ; // Power delivered to the load in hp (from case d conditions)
P_d_watt = P_d * 746 ; // P_d in W

// case g
P_f = (T_e*S_e)/5252 ; // Power delivered to the load in hp (from case f conditions)
P_f_watt = P_f * 746 ; // P_f in W

// case h
// Upper limit of power ranges A and B are:
A = 65 ; // Upper limit of power range A in W
B = 305 ; // Upper limit of power range B in W

// Display the results
```

```
disp("Example 11-1 Solution : ");

printf(" \n a: Extrapolating to load line point x,\n
        S = %d rpm ",S);
printf(" \n    Load line voltage is %d V \n",V);

printf(" \n b: At standstill , %d V yields T = %.1f
    lb-ft of starting torque\n",V,T);

printf(" \n c: Power delivered to the load in hp (
    from case a conditions)");
printf(" \n    P = %.4f hp = %d W \n",P_c,P_c_watt);

printf(" \n d: At point o:\n    T = %.1f lb-ft and S
     = %d rpm \n",T_d,S_d);

printf(" \n e: At point w:\n    T = %.1f lb-ft and S
     = %d rpm \n",T_e,S_e);

printf(" \n f: P = %.4f hp = %.1f W \n ",P_d,
    P_d_watt);

printf(" \n g: P = %.4f hp = %.f W \n",P_f,P_f_watt
    );

printf(" \n h: A = %d W  and  B = %d W ", A , B );
```

Scilab code Exa 11.2 calculate stepping angle

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 11: SPECIALIZED DYNAMOS
```

```
// Example 11-2

clear; clc; close; // Clear the work space and console.

// Given data
// VR stepper motor
n = 3 ; // Number of stacks or phases
P_a = 16 ; // Number of rotor teeth (subscript a indicates case a)
// PM stepper
P_b = 24 ; // Number of poles (subscript b indicates case b)

// Calculations
// case a
alpha_a = 360 / (n*P_a); // Stepping angle in degrees per step

alpha_b = 360 / (n*P_b); // Stepping angle in degrees per step

// Display the results
disp("Example 11-2 Solution : ");
printf(" \n a: alpha    = %.1f degrees/step \n ", alpha_a );

printf(" \n b: alpha    = %.1f degrees/step \n ", alpha_b );
```

Scilab code Exa 11.3 calculate stepping length

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
```

```
// 2nd editiom

// Chapter 11: SPECIALIZED DYNAMOS
// Example 11-3

clear; clc; close; // Clear the work space and
    console.

// Given data
// Hybrid stepping motor
P = 50 ; // Number of rotor teeth

// Calculation

alpha = 90 / P ; // Stepping angle in degrees

// Display the result
disp("Example 11-3 Solution : ");
printf(" \n alpha    = %.1f degrees ", alpha );
```

Scilab code Exa 11.4 calculate synchronous velocity

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 11: SPECIALIZED DYNAMOS
// Example 11-4

clear; clc; close; // Clear the work space and
    console.

// Given data
tou = 0.1 ; // Pole pitch of a double-sided primary
```

```
LIM in meter
f = 60 ; // Frequency applied to the primary LIM in Hz

// Calculation
v_s = 2 * f * tou ; // Synchronous velocity in meter/second

// Display the result
disp("Example 11-4 Solution : ");
printf(" \n Synchronous velocity : \n v_s = %d m/s ", v_s );
```

Scilab code Exa 11.5 calculate slip of DSLIM

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 11: SPECIALIZED DYNAMOS
// Example 11-5

clear; clc; close; // Clear the work space and console.

// Given data
v_s = 12 ; // Synchronous velocity in meter/second
v = 10 ; // Secondary sheet in Ex.11-4 moves at a linear velocity in m/s

// Calculation
s = (v_s - v)/v_s ; // Slip of the DSLIM

// Display the result
```

```
disp("Example 11-5 Solution : ");disp("From Eq
    .(11-5)")
printf(" \n Slip of the DSLIM : \n s = %.3f  ",s );
```

Chapter 12

POWER ENERGY AND EFFICIENCY RELATIONS OF DC AND AC DYNAMOS

Scilab code Exa 12.1 Pr Ia efficiency

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS
   OF DC AND AC DYNAMOS
// Example 12-1

clear; clc; close; // Clear the work space and
   console.

// Given data
P = 10000 ; // Power rating of the shunt generator
   in W
V = 230 ;// Voltage rating of the shunt generator in
    volt
```

```
S = 1750 ; // Speed in rpm of the shunt generator
// Shunt generator was made to run as a motor
V_a = 245 ; // Voltage across armature in volt
I_a = 2 ; // Armature current in A
R_f = 230 ; // Field resistance in ohm
R_a = 0.2 ; // Armature resistance

// Calculations
// case a
Rotational_losses = (V_a * I_a) - (I_a^2 * R_a); // Rotational losses in W at full load

// case b
V_t = V ;
// At rated load
I_L = P / V_t ; // Line current in A
I_f = V / R_f ; // Field current in A
Ia = I_f + I_L ; // Armature current in A

armature_loss = (Ia^2 * R_a); // Full-load armature loss in W
V_f = V ; // Field voltage in volt
field_loss = V_f * I_f; // Full-load field loss in W

// case c
//
eta = P / ( P + Rotational_losses + (armature_loss+field_loss) ) * 100 ;

// Display the results
disp("Example 12-1 Solution : ");

printf(" \n a: Rotational losses at full load = %.1f W \n",Rotational_losses);

printf(" \n b: At the rated load,\n    I_L = %.1f A\n    I_a = %.1f A\n",I_L,Ia);
printf(" \n    Full-load armature loss :\n    (I_a
```

```
^2)*R_a = %.f W \n",armature_loss);
printf(" \n     Full-load field loss :\n     V_f*I_f = %.f W \n",field_loss);

printf(" \n c: Efficiency of the generator at rated load(full-load in this Ex.) : ");
printf(" \n        = %.1f percent ",eta);
```

Scilab code Exa 12.2 efficiency at different LF

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS OF DC AND AC DYNAMOS
// Example 12-2

clear; clc; close; // Clear the work space and console.

// Given data
// data from Ex.12-1
P = 10000 ; // Power rating of the shunt generator in W
V = 230 ;// Voltage rating of the shunt generator in volt
S = 1750 ; // Speed in rpm of the shunt generator

// ( Solutions from Example 12-1 )
Rotational_losses = 489.2 // Rotational losses at full load in W
armature_loss = 396 ; // Full-load armature loss in W
```

```
field_loss = 230 ; // Full-load field loss in W

// case a
x1 = (1/4); // Fraction of full-load
// Subscript a for eta indicates case a
eta_a = (P*x1) / ( (P*x1) + Rotational_losses + ( armature_loss*(x1^2)+field_loss) ) * 100 ;

// case b
x2 = (1/2); // Fraction of full-load
// Subscript b for eta indicates case b
eta_b = (P*x2) / ( (P*x2) + Rotational_losses + ( armature_loss*(x2^2)+field_loss) ) * 100 ;

// case c
x3 = (3/4); // Fraction of full-load
// Subscript c for eta indicates case c
eta_c = (P*x3) / ( (P*x3) + Rotational_losses + ( armature_loss*(x3^2)+field_loss) ) * 100 ;

// case d
x4 = (5/4); // Fraction of full-load
// Subscript d for eta indicates case d
eta_d = (P*x4) / ( (P*x4) + Rotational_losses + ( armature_loss*(x4^2)+field_loss) ) * 100 ;

// Display the results
disp("Example 12-2 Solution : ");

printf(" \n    If x is the fraction of full-load , then \n ");
printf(" \n a: Efficiency of generator when x = %.2f ",x1 );
printf(" \n        = %.1f percent \n ",eta_a);

printf(" \n b: Efficiency of generator when x = %.2f ",x2 );
printf(" \n        = %.1f percent \n ",eta_b);
```

```

printf(" \n c: Efficiency of generator when x = %.2f ",x3 );
printf(" \n        = %.1f percent \n ",eta_c);

printf(" \n d: Efficiency of generator when x = %.2f ",x4 );
printf(" \n        = %.1f percent \n ",eta_d);
```

Scilab code Exa 12.3 field current Ec Pf

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS OF DC AND AC DYNAMOS
// Example 12-3

clear; clc; close; // Clear the work space and console.

// Given data
V = 240 ; // Voltage rating of the dc shunt motor in volt
P_hp = 25 ; // Power rating of the dc shunt motor in hp
S = 1800 ; // Speed in rpm of the shunt generator
I_L = 89 ; // Full-load line current
R_a = 0.05 ; // Armature resistance in ohm
R_f = 120 ; // Field resistance in ohm

// Calculations
// case a
```

```
V_f = V ; // Field voltage in volt
I_f = V_f / R_f ; // Field current in A
I_a = I_L - I_f  ; // Armature current in A
V_a = V ;
E_c = V_a - I_a*R_a ; // Armature voltage to be
    applied to the motor when motor
// is run light at 1800 rpm during stray power test

// case b
Ia = 4.2 ; // Armature current in A produced by E_c
Va = E_c ; // Armature voltage in volt
P_r = Va*Ia ; // Stray power in W ,when E_c produces
     I_a = 4.2 A at speed of 1800 rpm

// Display the results
disp("Example 12-3 Solution : ");

printf(" \n a: Field current :\n    I_f = %d A \n ",
    I_f );
printf(" \n    Armature current :\n    I_a = %d A \n
     ",I_a );
printf(" \n    Armature voltage to be applied to the
     motor when motor is run");
printf(" \n    light at %d rpm during stray power
    test :\n ",S );
printf(" \n    E_c = %.2f V \n ",E_c );

printf(" \n b: Stray power :\n    P_r = %.1f W ",P_r
     );
```

Scilab code Exa 12.4 Pr variable losses efficiency table

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
```

```
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS
    OF DC AND AC DYNAMOS
// Example 12-4

clear; clc; close; // Clear the work space and
    console.

// Given data
V = 600 ; // Voltage rating of the compound motor in
    volt
P_hp = 150 ; // Power rating of the compound motor
    in hp
I_L = 205 ; // Full-load rated line current in A
S = 1500 ; // Full-load Speed in rpm of the compound
    generator
R_sh = 300 ; // Shunt field resistance in ohm
R_a = 0.05 ; // Armature resistance in ohm
R_s = 0.1 ; // Series field resistance in ohm
V_a  = 570 ; // Applied voltage in volt
I_a = 6 ; // Armature current in A
S_o = 1800 ; // No-load Speed in rpm of the compound
    generator

// Calculations
// case a
Rot_losses = V_a*I_a ; // Rotational losses in W
// If x is fraction of full-load
x1 = (1/4);
S_1 = S_o - 300*x1 ; // Speed at 1/4 load
Rot_losses_S_1 = (S_1/S)*Rot_losses ; // Rotational
    losses in W at speed S_1

x2 = (1/2);
S_2 = S_o - 300*x2 ; // Speed at 1/2 load
Rot_losses_S_2 = (S_2/S)*Rot_losses ; // Rotational
    losses in W at speed S_2
```

```

x3 = (3/4);
S_3 = S_o - 300*x3 ; // Speed at 3/4 load
Rot_losses_S_3 = (S_3/S)*Rot_losses ; // Rotational losses in W at speed S_3

x4 = (5/4);
S_4 = S_o - 300*x4 ; // Speed at 5/4 load
Rot_losses_S_4 = (S_4/S)*Rot_losses ; // Rotational losses in W at speed S_4

// case b
I_sh = V / R_sh ; // Full-load shunt field current in A
Ia = I_L - I_sh ; // Full-load armature current in A
FL_variable_loss = (Ia^2)*(R_a + R_s); // Full-load variable electric losses in W

x1_variable_loss = FL_variable_loss * (x1)^2 ; // Variable losses at 1/4 load
x2_variable_loss = FL_variable_loss * (x2)^2 ; // Variable losses at 1/2 load
x3_variable_loss = FL_variable_loss * (x3)^2 ; // Variable losses at 3/4 load
x4_variable_loss = FL_variable_loss * (x4)^2 ; // Variable losses at 5/4 load

// case c
// Efficiency of motor = (Input - losses)/Input
// where Input = volts*amperes*load_fraction
//        Losses = field loss + rotational losses + variable electric losses
// Input
Input_FL = V * I_L ; // Input in W at full load
Input_x1 = V * I_L * x1 ; // Input in W at 1/4 load
Input_x2 = V * I_L * x2 ; // Input in W at 1/2 load
Input_x3 = V * I_L * x3 ; // Input in W at 3/4 load
Input_x4 = V * I_L * x4 ; // Input in W at 5/4 load
```

```
Field_loss = V * I_sh // Field loss for each of the
    conditions of load

// Rotational losses are calculated in part a while
    variable electric losses in part b

// Total losses
Losses_FL = Field_loss + Rot_losses +
    FL_variable_loss ; // Total losses for full load
Losses_1 = Field_loss + Rot_losses_S_1 +
    x1_variable_loss ; // Total losses for 1/4 load
Losses_2 = Field_loss + Rot_losses_S_2 +
    x2_variable_loss ; // Total losses for 1/2 load
Losses_3 = Field_loss + Rot_losses_S_3 +
    x3_variable_loss ; // Total losses for 3/4 load
Losses_4 = Field_loss + Rot_losses_S_4 +
    x4_variable_loss ; // Total losses for 5/4 load

// Efficiency
eta_FL = ( (Input_FL - Losses_FL) / Input_FL ) ; //
    Efficiency for 1/4 load
eta_1 = ( (Input_x1 - Losses_1) / Input_x1 ) ; //
    Efficiency for 1/4 load
eta_2 = ( (Input_x2 - Losses_2) / Input_x2 ) ; //
    Efficiency for 1/2 load
eta_3 = ( (Input_x3 - Losses_3) / Input_x3 ) ; //
    Efficiency for 3/4 load
eta_4 = ( (Input_x4 - Losses_4) / Input_x4 ) ; //
    Efficiency for 5/4 load

// Display the results
disp("Example 12-4 Solution : ");

printf(" \n a: Rotational loss = %d W at %d rpm(
    rated load)\n",Rot_losses,S);
printf(" \n    Speed at %.2f load = %d rpm  ",x1 ,
    S_1 );
```

```
printf(" \n     Rotational loss at %d rpm = %d W \n ", S_1 , Rot_losses_S_1 );

printf(" \n     Speed at %.2f load = %d rpm  ",x2 , S_2 );
printf(" \n     Rotational loss at %d rpm = %d W \n ", S_2 , Rot_losses_S_2 );

printf(" \n     Speed at %.2f load = %d rpm  ",x3 , S_3 );
printf(" \n     Rotational loss at %d rpm = %d W \n ", S_3 , Rot_losses_S_3 );

printf(" \n     Speed at %.2f load = %d rpm  ",x4 , S_4 );
printf(" \n     Rotational loss at %d rpm = %d W \n ", S_4 , Rot_losses_S_4 );

printf(" \n b: Full-load variable loss = %d W\n ", FL_variable_loss );
printf(" \n     Variable losses ,");
printf(" \n     at %.2f load = %.2f W ",x1 , x1_variable_loss );
printf(" \n     at %.2f load = %.2f W ",x2 , x2_variable_loss );
printf(" \n     at %.2f load = %.2f W ",x3 , x3_variable_loss );
printf(" \n     at %.2f load = %.2f W \n ",x4 , x4_variable_loss );

printf(" \n c: Efficiency of motor = (Input - losses)/Input ");
printf(" \n     where\n      Input = volts*amperes*load_fraction ");
printf(" \n      Losses = field loss + rotational losses + variable electric losses");
printf(" \n     Input,\n     at %.2f load = %d W ",x1 , Input_x1 );
```

```
printf(" \n     at %.2f load = %d W ",x2 , Input_x2 );
printf(" \n     at %.2f load = %d W ",x3 , Input_x3 );
printf(" \n     at full load = %d W " , Input_FL );
printf(" \n     at %.2f load = %d W \n ",x4  , Input_x4 );

printf(" \n     Field loss for each of the conditions of load = %d W \n",Field_loss);
printf(" \n     Rotational losses are calculated in part a while variable ");
printf(" \n     electric losses in part b \n");

printf(" \n     Efficiency at %.2f load = %f = %.1f percent ",x1,eta_1,eta_1*100);
printf(" \n     Efficiency at %.2f load = %f = %.1f percent ",x2,eta_2,eta_2*100);
printf(" \n     Efficiency at %.2f load = %f = %.1f percent ",x3,eta_3,eta_3*100);
printf(" \n     Efficiency at full load = %f = %.1f percent ",eta_FL,eta_FL*100);
printf(" \n     Efficiency at %.2f load = %f = %.1f percent \n",x4,eta_4,eta_4*100);

printf(" \n d: ---------------------------------------------------------------------------");
printf(" \n         Item \t\t\t At 1/4 load \t At 1/2 load \t At 3/4 load \t At Full-load\t At 5/4 load   ");
printf(" \n ---------------------------------------------------------------------------");
printf(" \n     Input(watts)\t\t %d \t\t %d \t\t %d \t\t %d \t %d ",Input_x1,Input_x2,Input_x3,Input_FL,Input_x4);
printf(" \n\n     Field loss(watts)\t\t %d \t\t %d \t
```

```
    \t %d \t\t %d \t\t %d ",Field_loss,Field_loss,
    Field_loss,Field_loss,Field_loss);
printf(" \n\n    Rotational losses");
printf(" \n    from part(a)(watts)\t\t %d \t\t %d \t
    \t %d \t\t %d \t\t %d ",Rot_losses_S_1,
    Rot_losses_S_2,Rot_losses_S_3,Rot_losses,
    Rot_losses_S_4);
printf(" \n\n    Variable electric losses");
printf(" \n    from part(b)(watts)\t\t %.2f \t %.2f
    \t %.2f \t %.2f \t %.2f ",x1_variable_loss,
    x2_variable_loss,x3_variable_loss,
    FL_variable_loss,x4_variable_loss);
printf(" \n\n    Total losses(watts)\t\t %.2f \t %.2
    f \t %.2f \t %.2f \t %.2f ",Losses_1,Losses_2,
    Losses_3,Losses_FL,Losses_4);
printf(" \n
    ------------------------------------------------------------------
    ");
printf(" \n    Efficiency   (percent)\t %.1f \t\t %
    .1f \t\t %.1f \t\t %.1f \t\t %.1f  ",eta_1*100,
    eta_2*100,eta_3*100,eta_FL*100,eta_4*100);
printf(" \n
    ------------------------------------------------------------------
    ");
```

Scilab code Exa 12.5 Ia LF max efficiency LF

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS
    OF DC AND AC DYNAMOS
// Example 12-5
```

```
clear; clc; close; // Clear the work space and console.

// Given data
P = 10000 ; // Power rating of the shunt generator in W
V = 230 ;// Voltage rating of the shunt generator in volt
S = 1750 ; // Speed in rpm of the shunt generator
R_a = 0.2 ; // Armature resistance
// Calculated values from Ex.12-1
P_r = 489.2 ; // Shunt generator rotational losses in W
Vf_If = 230 ; // Shunt field circuit loss in W
I_a_rated = 44.5 ; // Rated armature current in A

// Calculations
// case a
I_a = sqrt( (Vf_If + P_r) / R_a ); // Armature current in A for max.efficiency

// case b
LF = I_a / I_a_rated ; // Load fraction
LF_percent = LF*100 ; // Load fraction in percent

// case c
P_k = Vf_If + P_r ;
eta_max = (P*LF)/( (P*LF) + (Vf_If + P_r) + P_k ) * 100; // Maximum efficiency

// case d
// subscript d for LF indicates case d
LF_d = sqrt(P_k/(I_a_rated^2*R_a)) ; // Load fraction from fixed losses and rated variable losses

// Display the results
```

```
disp("Example 12-5 Solution : ");

printf(" \n a: Armature current for max.efficiency :\n    I_a = %.f A \n",I_a);

printf(" \n b: Load fraction :\n    L.F. = %.1f percent = %.3f*rated \n",LF_percent,LF);

printf(" \n c: Maximum efiiciency :\n        = %.2f percent \n",eta_max);

printf(" \n d: Load fraction from fixed losses and rated variable losses :");
printf(" \n    L.F. = %.3f*rated",LF_d);
```

Scilab code Exa 12.6 Pd Pr efficiency

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS OF DC AND AC DYNAMOS
// Example 12-6

clear; clc; close; // Clear the work space and console.

// Given data
V = 240 ; // Voltage rating of dc shunt motor in volt
P_hp = 5 ; // Power rating of dc shunt motor in hp
S = 1100 ; // Speed in rpm of the dc shunt motor
R_a = 0.4 ; // Armture resistance in ohm
```

```
R_f = 240 ; // Field resistance in ohm
I_L = 20 ; // Rated line current in A

// Calculations
// Preliminary calculations
V_f = V ; // Voltage across field winding in volt
I_f = V_f / R_f ; // Field current in A
I_a = I_L - I_f ; // Armature current in A
P_o = P_hp * 746 ; // Power rating of dc shunt motor in W
V_a = V ; // Voltage across armature in volt
E_c_fl = V_a - I_a*R_a ; // back EMF in volt

// case a
E_c = E_c_fl ;
P_d = E_c * I_a ; // Power developed by the armature in W

// case b
P_r = P_d - P_o ; // Full-load rotational losses in W

// case c
P_in = V*I_L ; // Input power in W
eta = (P_o/P_in)*100 ; // Full-load efficiency

// Display the results
disp("Example 12-6 Solution : ");

printf(" \n Preliminary calculations using nameplate data : ");
printf(" \n Field current : I_f = %d A \n ",I_f);
printf(" \n Armature current : I_a = %d A \n ",I_a);
printf(" \n P_o = %d W ",P_o );
printf(" \n E_c(fl) = %.1f V \n",E_c_fl);

printf(" \n a: Power developed by the armature :\n P_d = %.1f W \n",P_d);
```

```
printf(" \n b: Full-load rotational losses :\n
    P_r = %.1f W \n",P_r);

printf(" \n c: Full-laod efficiency :\n        = %.1f
    percent ",eta );
```

Scilab code Exa 12.7 Pd Pr max and fl efficiency Pk Ia LF

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS
    OF DC AND AC DYNAMOS
// Example 12-7

clear; clc; close; // Clear the work space and
    console.

// Given data
V = 240 ; // Voltage rating of dc shunt motor in
    volt
P_hp = 25 ; // Power rating of dc shunt motor in hp
S = 1100 ; // Speed in rpm of the dc shunt motor
R_a = 0.15 ; // Armture resistance in ohm
R_f = 80 ; // Field resistance in ohm
I_L = 89 ; // Rated line current in A

// Calculations
// Preliminary calculations
V_f = V ; // Voltage across field winding in volt
I_f = V_f / R_f ; // Field current in A
I_a = I_L - I_f ; // Armature current in A
```

```
P_o = P_hp * 746 ; // Power rating of dc shunt motor
    in W
V_a = V ; // Voltage across armature in volt
E_c_fl = V_a - I_a*R_a ; // back EMF in volt

// case a
E_c = E_c_fl ;
P_d = E_c * I_a ; // Power developed by the armature
    in W

// case b
P_r = P_d - P_o ; // Full-load rotational losses in
   W

// case c
P_in = V*I_L ; // Input power in W
eta_fl = (P_o/P_in)*100 ; // Full-load efficiency

// case d
P_k = V_f*I_f + P_r ; // Total constant losses in W

// case e
Ia = sqrt(P_k/R_a); // Armature current in A from
   maximum efficiency

// case f
LF = Ia / I_a ; // Load fraction at which max.
   efficiency is produced

// case g
rated_input = V*I_L ;
eta_max = ( (LF*rated_input) - 2*P_k ) / (LF*
   rated_input) * 100; // Maximum efficiency

// Display the results
disp("Example 12-7 Solution : ");

printf(" \n Field current : I_f = %d A \n ",I_f);
```

```
printf(" \n Armature current : I_a = %d A \n ",I_a);
printf(" \n P_o = %d W \n",P_o );
printf(" \n E_c(fl) = %.1f V \n",E_c_fl);

printf(" \n a: Power developed by the armature :\n P_d = %.1f W \n",P_d);

printf(" \n b: Full-load rotational losses :\n P_r = %.1f W \n",P_r);

printf(" \n c: Full-laod efficiency :\n       = %.1f percent \n ",eta_fl );

printf(" \n d: Total constant losses :\n    P_k = %.1f W \n",P_k);

printf(" \n e: Armature current from maximum efficiency :\n    I_a = %.1f A\n ",Ia);

printf(" \n f: L.F. = %.1f \n ",LF);

printf(" \n g:  _max = %.1f percent",eta_max);
```

Scilab code Exa 12.8 IL Ia Pd Pr Speed SR

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS OF DC AND AC DYNAMOS
// Example 12-8

clear; clc; close; // Clear the work space and
```

```
console.

// Given data
V = 240 ; // Voltage rating of dc shunt motor in volt
P_hp = 5 ; // Power rating of dc shunt motor in hp
S_fl = 1100 ; // Speed in rpm of the dc shunt motor
R_a = 0.4 ; // Armture resistance in ohm
R_f = 240 ; // Field resistance in ohm
eta = 0.75 ; // Full-load efficiency

// Calculations
// case a
V_L = V ; // Load voltage
P_o = P_hp * 746 ; // Power rating of dc shunt motor in W
I_L = P_o / (eta*V_L); // Rated input line current in A

V_f = V ; // Voltage across field winding in volt
I_f = V_f / R_f ; // Field current in A
I_a = I_L - I_f ; // Rated armature current in A

// case b
V_a = V ; // Voltage across armature in volt
E_c = V_a - I_a*R_a ; // back EMF in volt
P_d = E_c * I_a ; // Power developed by the armature in W

// case c
P_r = P_d - P_o ; // Rotational losses in W at rated load

// case d
// At no-load
P_o_nl = 0 ;
P_r_nl = P_r ; // Rotational losses in W at no load
P_d_nl = P_r_nl ;
```

```
// case e
I_a_nl = P_d_nl / V_a ; // No-load armature current in A

// case f
E_c_nl = V ; // No-load voltage in volt
E_c_fl = E_c ; // Full-load voltage in volt
S_nl = (E_c_nl / E_c_fl)*S_fl ; // No-load speed in rpm

// case g
SR = (S_nl - S_fl)/S_fl * 100 ; // Speed regulation

// Display the results
disp("Example 12-8 Solution : ");

printf(" \n a: Rated input line current :\n    I_L = %.2f A \n ",I_L);
printf(" \n    Rated armature current :\n    I_a = %.2f A \n ",I_a );

printf(" \n b: E_c = %.1f V \n ",E_c );
printf(" \n    Power developed by the armature at rated load :\n    P_d = %d W \n ",P_d);

printf(" \n c: Rotational losses at rated load :\n      P_r = %d W \n ", P_r );

printf(" \n d: At no-load, P_o = %d W ; therefore\n\t\tP_d = P_r = %d W \n",P_o_nl,P_r);

printf(" \n e: No-load armature current :\n    I_a(nl) = %.2f A \n ",I_a_nl );

printf(" \n f: No-load speed :\n    S_nl = %.f rpm \n ",S_nl );

```

```
printf(" \n g: Speed regulation :\n    SR = %.1 f
   percent ",SR );
```

Scilab code Exa 12.9 Ec Pd Po Pr To Ia efficiency speed SR

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS
    OF DC AND AC DYNAMOS
// Example 12-9

clear; clc; close; // Clear the work space and
   console.

// Given data
V = 240 ; // Voltage rating of dc shunt motor in
   volt
I_L = 55 ; // Rated line current in A
S = 1200 ; // Speed in rpm of the dc shunt motor
P_r = 406.4 ; // Rotational losses in W at rated
   load
R_f = 120 ; // Field resistance in ohm
R_a = 0.4 ; // Armture resistance in ohm

// Calculations
// case a

V_f = V ; // Voltage across field winding in volt
I_f = V_f / R_f ; // Field current in A
I_a = I_L - I_f ; // Rated armature current in A

V_a = V ; // Voltage across armature in volt
```

```
E_c = V_a - I_a*R_a ; // back EMF in volt
P_d = E_c * I_a ; // Power developed by the armature
    in W

// case b
P_o = P_d - P_r  ; // Rated output power in W
P_o_hp = P_o / 746 ; // Rated output power in hp

// case c
T_o = (P_o_hp * 5252)/S ; // C in lb-ft
T_o_Nm = T_o * (1.356); // Rated output torque in N-
   m

// case d
P_in = V*I_L ; // Input power in W
eta = (P_o/P_in)*100 ; // Efficiency at rated load

// case e
// At no-load
P_o_nl = 0 ;
P_r_nl = P_r ; // Rotational losses in W at no load
P_d_nl = P_r_nl ;

I_a_nl = P_d_nl / V_a ; // No-load armature current
   in A

E_c_nl = V ; // No-load voltage in volt
E_c_fl = E_c ; // Full-load voltage in volt
S_fl = S ; // Full-load speed in rpm
S_nl = (E_c_nl / E_c_fl)*S_fl ; // No-load speed in
   rpm

// case f
SR = (S_nl - S_fl)/S_fl * 100 ; // Speed regulation

// Display the results
disp("Example 12-9 Solution : ");

```

```
printf("\n a: E_c = %.1f V \n",E_c );
printf("\n    Power developed by the armature at
    rated load :\n    P_d = %.1f W \n",P_d);

printf("\n b: Rated output power :\n    P_o = %d W
    \n", P_o );
printf("\n    P_o = %d hp \n",P_o_hp);

printf("\n c: Rated output torque :\n    T_o = %.2f
    lb-ft ",T_o);
printf("\n    T_o = %.f N-m \n",T_o_Nm );

printf("\n d: Efficiency at rated load :\n    =
    %.1f percent \n",eta );

printf("\n e: At no-load, P_o = %d W ; therefore\n\
    t\tP_d = P_r = EcIa    VaIa = %.1f W \n",P_o_nl,
    P_r);
printf("\n    No-load armature current :\n    I_a(
    nl) = %.3f A \n",I_a_nl );
printf("\n    No-load speed :\n    S_nl = %f    %.
    f rpm \n",S_nl,S_nl );

printf("\n f: Speed regulation :\n    SR = %.1f
    percent ",SR );

printf("\n    Variation in SR is due to non-
    approximation of S_nl = %f rpm",S_nl);
printf("\n    while calculating SR in scilab .")
```

Scilab code Exa 12.10 efficiency Pf Pd Pr Ia LF max efficiency

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
```

```
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS
   OF DC AND AC DYNAMOS
// Example 12-10

clear; clc; close; // Clear the work space and
   console.

// Given data
V = 125 ; // Voltage rating of genrator in volt
P_o = 12500 ; // Power rating of genrator in W
P_hp = 20 ; // Power rating of motor in hp
R_a = 0.1 ; // Armture resistance in ohm
R_f = 62.5 ; // Field resistance in ohm
P_var = 1040 ; // Rated variable electric loss in W

// Calculations
// case a
P_in = P_hp * 746 ; // Power input to generator in W
eta = P_o / P_in * 100 ; // Efficiency

// case b
V_f = V  ; // Voltage across shunt field wdg in volt
P_sh_loss = (V_f)^2 / R_f ; // Shunt field loss in W

// case c
V_L = V ;
I_L = P_o / V_L ; // Line current in A
I_f = V_f / R_f ; // Field current in A
I_a = I_L + I_f ; // Armature current in A
E_g = V_L + I_a*R_a ; // Generated EMF in volt

P_d1 = E_g * I_a ; // Generated electric power in W
P_f = V_f * I_f ;
P_d2 = P_o + P_var + P_f ; // Generated electric
   power in W

```

```
// case d
P_d = P_d1;
P_r = P_in - P_d ; // Rotational power losses in W

// case e
P_k = P_r + V_f*I_f ; // Constant losses in W
Ia = sqrt(P_k/R_a); // Armature current in A for max.efficiency

// case f
I_a_rated = I_a ; // Rated armature current in A
LF = Ia / I_a ; // Load fraction

// case g
rated_output = 12500 ; // Rated output in kW
// Maximum efficiency
eta_max = ( LF * rated_output ) / ( ( LF * rated_output ) + (2*P_k) ) * 100 ;

// Display the results
disp("Example 12-10 Solution : ");

printf(" \n a: Efficiency :\n     = %f percent %.1f percent \n ",eta,eta);

printf(" \n b: Shunt field loss :\n    (V_f)^2/R_f = %d W \n ",P_sh_loss);

printf(" \n c: Line current : I_L = %d A \n\n Field current : I_f = %d A",I_L,I_f);
printf(" \n\n    Armature current : I_a = %d A ",I_a);
printf(" \n\n    Generated EMF : E_g = %.1f V ",E_g);
printf(" \n\n    Generated electric power : ");
printf(" \n    1. P_d = %d W \n\n    2. P_d = %d W \n ",P_d1,P_d2);

```

```
printf(" \n d: Rotational power losses :\n      P_r =
   %f W      %.f W \n",P_r,P_r);

printf(" \n e: Constant losses : P_k = %f W      %.f
   W \n ", P_k ,P_k);
printf(" \n    Armature current for max.efficiency :
     I_a = %.1f A \n ",Ia);

printf(" \n f: Load fraction : L.F. = %.2f \n ",LF);

printf(" \n g: Maximum efficiency :     = %f percent
        %.2f percent",eta_max,eta_max);
```

Scilab code Exa 12.11 efficiency at different LF

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS
   OF DC AND AC DYNAMOS
// Example 12-11

clear; clc; close; // Clear the work space and
   console.

// Given data (from Ex.12-10)
V = 125 ; // Voltage rating of genrator in volt
P_o = 12500 ; // Power rating of genrator in W
P_hp = 20 ; // Power rating of motor in hp
R_a = 0.1 ; // Armture resistance in ohm
R_f = 62.5 ; // Field resistance in ohm
P_var = 1040 ; // Rated variable electric loss in W

```

```
// Calculated data from Ex.12-10
P_k = 1380 ; // Constant losses in W

// Calculations
// Efficiency of the dc shunt generator
//     = (output*L.F) / ( (output*L.F) + P_k + (L.F)^2 * P_a_rated ) * 100
output = P_o ;
P_a_rated = P_var ;

// case a
LF1 = 25*(1/100); // At 25 % rated output
// Efficiency of the dc shunt generator at 25 % rated output
eta_1 = (output*LF1) / ( (output*LF1) + P_k + (LF1)^2 * P_a_rated ) * 100 ;

// case b
LF2 = 50*(1/100); // At 50 % rated output
// Efficiency of the dc shunt generator at 50 % rated output
eta_2 = (output*LF2) / ( (output*LF2) + P_k + (LF2)^2 * P_a_rated ) * 100 ;

// case c
LF3 = 75*(1/100); // At 75 % rated output
// Efficiency of the dc shunt generator at 75 % rated output
eta_3 = (output*LF3) / ( (output*LF3) + P_k + (LF3)^2 * P_a_rated ) * 100 ;

// case d
LF4 = 125*(1/100); // At 125 % rated output
// Efficiency of the dc shunt generator at 125 % rated output
eta_4 = (output*LF4) / ( (output*LF4) + P_k + (LF4)^2 * P_a_rated ) * 100 ;

```

```

// Display the results
disp("Example 12-11 Solution : ");

printf(" \n a:    at %.2f rated output = %.2f
   percent \n ",LF1,eta_1);

printf(" \n b:    at %.2f rated output = %.2f
   percent \n ",LF2,eta_2);
printf(" \n    Please note: Calculation error for
   case b:    in the textbook.\n");

printf(" \n c:    at %.2f rated output = %.2f
   percent \n ",LF3,eta_3);

printf(" \n d:    at %.2f rated output = %.2f
   percent \n ",LF4,eta_4);
```

Scilab code Exa 12.12 Ia Ra Pf Pk Pcu efficiencies Pd

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS
    OF DC AND AC DYNAMOS
// Example 12-12

clear; clc; close; // Clear the work space and
   console.

// Given data
// 3-phase Y-connected alternator
kVA = 100 ; // kVA rating of the alternator
```

```
V  = 1100 ; // Rated voltage of the alternator in volt
I_a_nl = 8 ; // No-load armature current in A
P_in_nl = 6000 ; // No-load Power input to the armature in W
V_oc = 1350 ; // Open-ckt line voltage in volt
I_f = 18 ; // Field current in A
V_f = 125 ; // voltage across field winding in volt

// Calculations
// From Ex.6-4,
R_a = 0.45 ; // Armature resistance in ohm/phase
I_a_rated = 52.5 ; // Rated armature current in A/phase

// case a
P_r = P_in_nl - 3 * (I_a_nl)^2 * R_a ; // Rotational loss of synchronous dynamo in W

// case b
P_f = V_f*I_f ; // Field copper loss in W

// case c
P_k = P_r + P_f ; // Fixed losses in W at rated synchronous speed
Pk  = P_k / 1000 ; // Fixed losses in kW at rated synchronous speed

// case d
P_cu = 3 * (I_a_rated)^2 * R_a ; // Rated electric armature cu-loss in W
P_cu_kW = P_cu / 1000 ; // Rated electric armature cu-loss in kW

LF1 = 1/4 ; // Load fraction
LF2 = 1/2 ; // Load fraction
LF3 = 3/4 ; // Load fraction
P_cu_LF1 = P_cu * (LF1)^2 ; // Electric armature cu-
```

```
loss in W at 1/4 load
P_cu_LF2 = P_cu * (LF2)^2 ; // Electric armature cu-loss in W at 1/2 load
P_cu_LF3 = P_cu * (LF3)^2 ; // Electric armature cu-loss in W at 3/4 load

P_cu_LF1_kW = P_cu_LF1 / 1000 ; // Electric armature cu-loss in kW at 1/4 load
P_cu_LF2_kW = P_cu_LF2 / 1000 ; // Electric armature cu-loss in kW at 1/2 load
P_cu_LF3_kW = P_cu_LF3 / 1000 ; // Electric armature cu-loss in kW at 3/4 load


// case e
PF = 0.9 ; // Power factor lagging
// Efficiency
//    = LF(rated kVA)*PF / ( LF(rated kVA)*PF + P_k + P_cu ) * 100
eta_1 = (LF1 * kVA * PF) / ( (LF1 * kVA * PF) + Pk + P_cu_LF1_kW ) * 100 ;// Efficiency at 1/4 load
eta_2 = (LF2 * kVA * PF) / ( (LF2 * kVA * PF) + Pk + P_cu_LF2_kW ) * 100 ;// Efficiency at 1/2 load
eta_3 = (LF3 * kVA * PF) / ( (LF3 * kVA * PF) + Pk + P_cu_LF3_kW ) * 100 ;// Efficiency at 3/4 load
eta_fl = (kVA * PF) / ( (kVA * PF) + Pk + P_cu_kW ) * 100 ;// Efficiency at full load

// case f
Ia = sqrt(P_k/(3*R_a)); // Armature current in A for max.efficiency at 0.9 PF lagging
LF = Ia / I_a_rated ; // Load fraction for max. efficiency
// at max.efficiency P_cu = P_k
eta_max = (LF * kVA * PF) / ( (LF * kVA * PF) + 2*Pk ) * 100 ;// Max Efficiency 0.9 PF lagging

// case g
```

```
P_o = kVA*PF ; // Output power at 0.9 PF lagging
I_a = I_a_rated ;
P_d = P_o + (3*(I_a)^2*R_a/1000) + (V_f*I_f/1000) ; // Armature power developed in kW at 0.9 PF lagging at full-load

// Display the results
disp("Example 12-12 Solution : ");

printf(" \n From Ex.6-4,\n R_a = %.2f    /phase",R_a);
printf(" \n I_a(rated) = %.1f A \n ",I_a_rated);

printf(" \n a: Rotational loss of synchronous dynamo :\n     P_r = %.f W \n",P_r);

printf(" \n b: Field copper loss :\n     P_f = %d W \n ",P_f);

printf(" \n c: Fixed losses at rated synchronous speed :\n     P_k = %.f W\n",P_k);

printf(" \n d: P_cu at rated load = %.f W\n     P_cu ,",P_cu);
printf(" \n    at %.2f rated load = %.1f W",LF1 , P_cu_LF1);
printf(" \n    at %.2f rated load = %.1f W",LF2 , P_cu_LF2);
printf(" \n    at %.2f rated load = %.1f W \n",LF3 , P_cu_LF3);


printf(" \n e: Efficiency :\n        at %.2f load = %.1f percent",LF1,eta_1);
printf(" \n        at %.2f load = %.1f percent",LF2,eta_2);
printf(" \n        at %.2f load = %.1f percent",LF3,eta_3);
```

```
printf(" \n       at full-load = %.1f percent \n",
   eta_fl);

printf(" \n f: Armature current for max.efficiency
   at 0.9 PF lagging :");
printf(" \n    I_a(max) = %f A     %.1f A\n",Ia,Ia);
printf(" \n    L.F. = %.2f \n",LF);
printf(" \n    Maximum efficiency :\n     _max = %
   .1f percent \n ",eta_max);

printf(" \n g: Armature power developed at 0.9 PF
   lagging at full-load :");
printf(" \n    P_d = %.2f kW ",P_d);
```

Scilab code Exa 12.13 Pf Pcu Zs VR efficiencies Pd

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS
    OF DC AND AC DYNAMOS
// Example 12-13

clear; clc; close; // Clear the work space and
   console.

// Given data
// 3-phase Y-connected alternator
kVA = 1000 ; // kVA rating of the alternator
V  = 2300 ; // Rated voltage of the alternator in
   volt

// DC MOTOR
```

```
P_hp = 100 ; // Power rating of the dc motor in hp
V_motor  = 240 ; // Rated voltage of the motor in volt
// 4-step efficiency/regulation test
// Test 1
P_1 = 7.5 ; // motor output in kW

// Test 2
P_2 = 16 ; // motor output in kW
VfIf = 14 ; // Field losses in kW
P_f = VfIf ; // Field losses in kW

// Test 3
P_3 = 64.2 ; // motor output in kW
I_sc = 251 ; // Short ckt current in A

// Test 4
V_L = 1443 ; // Line voltage in volt

// Calculations
// case a
P_r = P_2 ; // Rotational losses in kW From test 2

// case b
P_cu = P_3 - P_1 ; // Full-load armature copper loss in kW

// case c
E_gL = V_L ; // Generated line voltage in volt
Z_s = (E_gL/sqrt(3)) / I_sc ; // Synchronous impedance of the armature in ohm

// case d
R_a = 0.3 ; // Armature resistance in ohm
X_s = sqrt( (Z_s)^2 - (R_a)^2 );  // Synchronous reactance of the armature in ohm

// case e
```

```
cos_theta = 0.8 ; // PF lagging
sin_theta = sqrt( 1 - (cos_theta)^2 );
V_p = V / sqrt(3); // Phase voltage in volt

// Generated voltage per phase in volt
I_a = I_sc ; // Armature current in A

E_gp = (V_p*cos_theta + I_a*R_a) + %i*(V_p*sin_theta + I_a*X_s);
E_gp_m = abs(E_gp);//E_gp_m=magnitude of E_gp in volt
E_gp_a = atan(imag(E_gp) /real(E_gp))*180/%pi;//E_gp_a=phase angle of E_gp in degrees

V_nl = E_gp_m ; // No-load voltage in volt
V_fl = V_p ; // Full-load voltage in volt

VR = (V_nl - V_fl)/V_fl * 100 ; // Alternator voltage regulation

// case f
PF = 0.8 ; // lagging PF
LF = 1 ; // load fraction
eta_rated = (LF*kVA*PF)/( (LF*kVA*PF) + (P_f + P_r) + P_cu ) * 100 ; // Efficiency at 0.8 lagging PF

// case g
P_k = (P_f + P_r) ; // Constant losses in kW
L_F = sqrt(P_k/P_cu); // Load fraction for max. efficiency
// at max.efficiency P_k = P_cu
eta_max = (L_F*kVA*PF)/( (L_F*kVA*PF) + 2*P_k ) * 100 ; // Max.Efficiency at 0.8 lagging PF


// case h
P_o = kVA ; // Output power in kVA
P_d = P_o +(3*(I_a)^2*R_a/1000) + (VfIf) ; //
```

```
Armature power developed in kW at unity PF at rated-load

// Display the results
disp("Example 12-13 Solution : ");

printf(" \n a: From Test 2, Rotational losses :\n      P_r = %d kW \n",P_r);

printf(" \n b: Full-load armature copper loss :\n      P_cu = %.1f kW \n",P_cu);

printf(" \n c: Synchronous impedance of the armature :\n      Z_s = %f        %.2f      \n",Z_s,Z_s);

printf(" \n d: Synchronous reactance of the armature :\n      jX_s = %f        %.2f      \n",X_s,X_s);

printf(" \n e: E_gp = ");disp(E_gp);
printf(" \n      E_gp = %.f <%.1f V\n",E_gp_m,E_gp_a);
printf(" \n      Alternator voltage regulation :\n    VR = %.2f percent \n",VR);

printf(" \n      Obtained VR value through scilab calculation is slightly different from textbook");
printf(" \n      because of non-approximation of Z_s, X_s and E_gp while calculating in scilab.\n");

printf(" \n f: Alternator efficiency at 0.8 lagging PF :\n      _rated = %.1f percent\n",eta_rated);

printf(" \n g: L.F = %.4f\n",L_F);
printf(" \n      Max.Efficiency at 0.8 lagging PF :\      _max = %.2f percent \n",eta_max );

printf(" \n h: Power developed by the alternator armature at rated load,unity PF :");
```

```
printf(" \n     P_d = %.f kW",P_d);
```

Scilab code Exa 12.14 Pr Pcu efficiencies hp torque

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS
    OF DC AND AC DYNAMOS
// Example 12-14

clear; clc; close; // Clear the work space and
   console.

// Given data
P = 4 ;// Number of poles in Induction motor
f = 60 ; // Frequency in Hz
V = 220 ; // Rated voltage of IM in volt
hp_IM = 5 ; // Power rating of IM in hp
PF = 0.9 ; // Power factor
I_L = 16 ; // Line current in A
S = 1750 ; // Speed of IM in rpm

// No-load test data
I_nl = 6.5 ; // No-load line current in A
V_nl = 220 ; // No-load line voltage in volt
P_nl = 300 ; // No-load power reading in W

// Blocked rotor test
I_br = 16 ; // Blocked rotor line current in A
V_br = 50 ; // Blocked rotor voltage in volt
P_br = 800 ; // Blocked rotor power reading in W

```

```
// Calculations
// case a
P_cu = P_br ; // Full-load equivalent cu-loss
I_1 = I_br ; // Primary current in A
R_e1 = (P_cu) / (3/2 * (I_1)^2 ); // Equivalent total resistance of IM in ohm

// case b
P_in = P_nl ; // Input power to IM
I1 = I_nl ; // Input current in A
P_r = P_in - (3/2 * (I1)^2 * R_e1); // Rotational losses in W

// case c
LF1 = 1/4 ; // Load fraction
LF2 = 1/2 ; // Load fraction
LF3 = 3/4 ; // Load fraction
LF4 = 5/4 ; // Load fraction
P_cu_LF1 = (LF1)^2 * P_cu ; // Equivalent copper loss at 1/4 rated-load
P_cu_LF2 = (LF2)^2 * P_cu ; // Equivalent copper loss at 1/2 rated-load
P_cu_LF3 = (LF3)^2 * P_cu ; // Equivalent copper loss at 3/4 rated-load
P_cu_LF4 = (LF4)^2 * P_cu ; // Equivalent copper loss at 5/4 rated-load

// case d
Full_load_input = sqrt(3)*V*I_L*PF ;

// Efficiency
// Efficiency at 1/4 rated load
eta_LF1 = ( Full_load_input*LF1 - (P_r + P_cu_LF1) )
    / (Full_load_input*LF1) * 100 ;

// Efficiency at 1/2 rated load
eta_LF2 = ( Full_load_input*LF2 - (P_r + P_cu_LF2) )
    / (Full_load_input*LF2) * 100 ;
```

```
// Efficiency at 3/4 rated load
eta_LF3 = ( Full_load_input*LF3 - (P_r + P_cu_LF3) ) / (Full_load_input*LF3) * 100 ;

// Efficiency at rated load
eta_rated = ( Full_load_input - (P_r + P_cu) ) / (Full_load_input) * 100 ;

// Efficiency at 5/4 rated load
eta_LF4 = ( Full_load_input*LF4 - (P_r + P_cu_LF4) ) / (Full_load_input*LF4) * 100 ;

// case e
// since eta is calculated in percent divide it by 100 for hp calculations
P_o_LF1 = (Full_load_input*LF1*eta_LF1/100)/746 ; // Output hp at 1/4 rated load
P_o_LF2 = (Full_load_input*LF2*eta_LF2/100)/746 ; // Output hp at 1/2 rated load
P_o_LF3 = (Full_load_input*LF3*eta_LF3/100)/746 ; // Output hp at 3/4 rated load
P_o = (Full_load_input*eta_rated/100)/746 ; // Output hp at 1/4 rated load
P_o_LF4 = (Full_load_input*LF4*eta_LF4/100)/746 ; // Output hp at 5/4 rated load

// case f
hp = P_o ; // Rated output horsepower
T_o = (P_o*5252)/S ; // Outpue torque at full-load in lb-ft
T_o_Nm = T_o * 1.356 ; // Outpue torque at full-load in N-m

// Display the results
disp("Example 12-14 Solution : ");

printf(" \n a: Equivalent total resistance of IM :\n
```

```
        R_e1 = %.3f      \n",R_e1);

printf(" \n b: Rotational losses :\n     P_r = %.f W \n ",P_r);

printf(" \n c: At full-load , P_cu = %d W \n",P_cu);
printf(" \n     P_cu at %.2f rated load = %d W",LF1,P_cu_LF1)
printf(" \n     P_cu at %.2f rated load = %d W",LF2,P_cu_LF2)
printf(" \n     P_cu at %.2f rated load = %d W",LF3,P_cu_LF3)
printf(" \n     P_cu at %.2f rated load = %d W \n",LF4,P_cu_LF4)

printf(" \n d: Full-load input = %.f W \n",Full_load_input);
printf(" \n     Efficiency :\n         at %.2f rated load = %.1f percent \n",LF1,eta_LF1);
printf(" \n         at %.2f rated load = %.1f percent \n",LF2,eta_LF2);
printf(" \n         at %.2f rated load = %.1f percent \n",LF3,eta_LF3);
printf(" \n         at rated load = %.1f percent \n",eta_rated);
printf(" \n         at %.2f rated load = %.1f percent \n",LF4,eta_LF4);

printf(" \n e: Output horsepower :\n     P_o at %.2f rated load = %.3f hp \n",LF1,P_o_LF1);
printf(" \n     P_o at %.2f rated load = %.3f hp \n",LF2,P_o_LF2);
printf(" \n     P_o at %.2f rated load = %.3f hp \n",LF3,P_o_LF3);
printf(" \n     P_o at rated load = %.3f hp \n",P_o);
printf(" \n     P_o at %.2f rated load = %.3f hp \n",LF4,P_o_LF4);

```

```
printf(" \n f: Output torque at full-load :\n      T_o
    = %.1f lb-ft",T_o);
printf(" \n     T_o = %.2f N-m",T_o_Nm);
```

Scilab code Exa 12.15 RPO efficiency hp torque compare

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS
    OF DC AND AC DYNAMOS
// Example 12-15

clear; clc; close; // Clear the work space and
   console.

// Given data(from Ex.12-14)
pole = 4 ;// Number of poles in Induction motor
f = 60 ; // Frequency in Hz
V = 220 ; // Rated voltage of IM in volt
hp_IM = 5 ; // Power rating of IM in hp
PF = 0.9 ; // Power factor
I_L = 16 ; // Line current in A
S_r = 1750 ; // Speed of IM in rpm

// No-load test data
I_nl = 6.5 ; // No-load line current in A
V_nl = 220 ; // No-load line voltage in volt
P_nl = 300 ; // No-load power reading in W

// Blocked rotor test
I_br = 16 ; // Blocked rotor line current in A
V_br = 50 ; // Blocked rotor voltage in volt
```

```
P_br = 800 ; // Blocked rotor power reading in W
R_dc = 1 ; // dc resistance in ohm between lines

// given data from ex.12-15
V = 220 ; // voltage rating in volt
P_input = 5500 ; // power drawn in W

// Calculations
// Preliminary calculations
R_e1 = 1.25*R_dc ; // Equivalent total resistance of IM in ohm
P_in = P_nl ; // Input power to IM in W
I1 = I_nl ; // Input current in A
P_r = P_in - (3/2 * (I1)^2 * R_e1); // Rotational losses in W

I_1 = I_L ;
SCL = (3/2 * (I_1)^2 * R_e1) ; // Stator Copper Loss in W at full-load
SPI = P_input ; // Stator Power Input in W
RPI = SPI - SCL ; // Rotor Power Input in W

S = (120*f/pole); // Speed of synchronous magnetic field in rpm
s = (S-S_r)/S ; // Slip

RPD = RPI*(1-s); // Rotor Power Developed in W
RPO = RPD - P_r ; // Rotor Power Output in W

// case a
P_o = RPO ;
eta_fl = (P_o / P_input)*100 ; // Full-load efficiency

// case b
hp = P_o / 746 ; // Output horsepower
T_o = (hp*5252)/S_r ; // Output torque in lb-ft
T_o_Nm = T_o * 1.356 ; // Output torque in N-m
```

```

// Display the results
disp("Example 12-15 Solution : ");

printf(" \n Preliminary calculations :");
printf(" \n R_e1 = %.2f     \n",R_e1);
printf(" \n P_r = %.1f W \n ",P_r);
printf(" \n SCL(fl) = %d W \n ",SCL);
printf(" \n RPI(fl) = %d W \n ",RPI);
printf(" \n RPD(fl) = %f W     %.1f W \n ",RPD,RPD);
printf(" \n RPO(fl) = %f W     %.f W \n ",RPO,RPO);

printf(" \n a: Full-load efficiency :\n      _fl = %.1f percent \n",eta_fl);

printf(" \n b: Output horsepower :\n    hp = %.2f hp at full-load \n",hp);
printf(" \n    Output torque at full-load :\n    T_o = %f lb-ft      %.1f lb-ft",T_o,T_o);
printf(" \n    T_o = %f lb-ft      %.2f N-m \n ",T_o_Nm,T_o_Nm);

printf(" \n c: Comparision of results");
printf(" \n ---------------------------------------------------------------");
printf(" \n \t\t\t\t\t Ex.12-14\tEx.12-15");
printf(" \n ---------------------------------------------------------------");
printf(" \n \t   _fl(percent) \t\t\t 82.4 \t\t %.1f ",eta_fl);
printf(" \n \t Rated output(hp) \t\t 6.06 \t\t %.2f ",hp);
printf(" \n \t Rated output torque(lb-ft) \t 18.2 \t\t %.1f ",T_o);
printf(" \n ---------------------------------------------------------------
```

```
");
```

Scilab code Exa 12.16 Ip Ir PF SPI SCL RPI RCL RPD T hp efficiency

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS
   OF DC AND AC DYNAMOS
// Example 12-16

clear; clc; close; // Clear the work space and
   console.

// Given data
// code letter = J
P = 6 ; // Number of poles
S_r = 1176 ; // rotor speed in rpm
V = 220 ; // Rated voltage of SCIM in volt
f = 60 ; // Frequency in Hz
hp_SCIM = 7.5 ; // Power rating of SCIM in hp

R_ap = 0.3 ; // armature resistance in ohm/phase
R_r = 0.144 ; // rotor resistance in ohm/phase
jX_m = 13.5 ; // reactance in ohm/phase
jX_s = 0.5 ; // synchronous reactance in ohm/phase
jX_lr = 0.2 ; // Locked rotor reactance in ohm/phase
P_r = 300 ; // Rotational losses in W

disp("Example 12-16 : ");
// Calculations
S = (120*f/P); // Speed of synchronous magnetic
   field in rpm
```

```
// case a
s = (S-S_r)/S ; // Slip

R_r_by_s = R_r / s ;

// case b
printf(" \n From fig.12-11 , using the format method of mesh analysis ,we may write");
printf(" \n the array by inspection :\n ");
printf(" \n-----------------------------------------------");
printf(" \n \t      I_1 \t I_2 \t\t V ");
printf(" \n-----------------------------------------------");
printf(" \n\t (0.3+j14)    -(0+j13.5) \t(127+j0)");
printf(" \n\t-(0+j13.5)    (7.2+j13.7) \t 0");
printf(" \n-----------------------------------------------\n");

A = [ (0.3 + %i*14) -%i*13.5 ; (-%i*13.5)  (7.2 + %i*13.7) ]; // Matrix containing above mesh eqns array
delta = det(A); // Determinant of A

// case b : Stator armature current I_p in A
I_p = det( [ (127+%i*0) (-%i*13.5) ; 0 (7.2 + %i*13.7) ] ) / delta ;
I_p_m = abs(I_p);//I_p_m=magnitude of I_p in A
I_p_a = atan(imag(I_p) /real(I_p))*180/%pi;//I_p_a=phase angle of I_p in degrees
I_1 = I_p ; // Stator armature current in A

// case c : Rotor current I_r per phase in A
I_r = det( [ (0.3 + %i*14) (127+%i*0) ; (-%i*13.5) 0 ] ) / delta ;
```

```
I_r_m = abs(I_r);//I_r_m=magnitude of I_r in A
I_r_a = atan(imag(I_r) /real(I_r))*180/%pi;//I_r_a=
    phase angle of I_r in degrees

// case d
theta = I_p_a ; // Motor PF angle in degrees
cos_theta = cosd(theta); // Motor PF

// case e
I_p = I_p_m ; // Stator armature current in A
V_p = V / sqrt(3); // Phase voltage in volt
SPI = V_p * I_p * cos_theta ; // Stator Power Input
    in W

// case f
SCL = (I_p)^2 * R_ap ; // Stator Copper Loss in W

// case g
// Subscripts 1 and 2 for RPI indicates two methods
    of calculating RPI
RPI_1 = SPI - SCL ; // Rotor Power Input in W
RPI_2 = (I_r_m)^2 * (R_r/s); // Rotor Power Input in
    W
RPI =RPI_1 ;

// case h
RCL = s*(RPI); // Rotor copper losses in W

// case i
// Subscripts 1 , 2 and 3 for RPD indicates three
    methods of calculating RPD
RPD_1 = RPI - RCL ; // Rotor Power Developed in W
RPD_2 = RPI * ( 1 - s ); // Rotor Power Developed in
    W
RPD = RPD_1 ;

// case j
RPO = 3*RPD - P_r ; // Rotor Power Developed in W
```

```
// case k
P_to = RPO ; // Total rotor power in W
T_o = (7.04*P_to)/S_r ; // Total 3-phase torque in
    lb-ft

// case l
hp = P_to / 746 ; // Output horsepower

// case m
P_in = 3*SPI ; // Input power to stator in W
P_o = RPO ; // Output power in W
eta = P_o / P_in * 100 ; // Motor efficiency at
    rated load

// Display the results
disp("Solution : ");
printf(" \n a: s = %.2f \n    R_r/s = %.1f    \n",s,
    R_r_by_s );

printf(" \n    Determinant    = ");disp(delta);

printf(" \n b: Stator armature current :\n    I_p in
     A = ");disp(I_1);
printf(" \n    I_p = I_1 = %.2f <%.2f A \n ",I_p_m ,
     I_p_a );

printf(" \n c: Rotor current per phase :\n    I_r in
     A = ");disp(I_r);
printf(" \n    I_r = I_2 = %.3f <%.2f A \n ",I_r_m ,
     I_r_a );

printf(" \n d: Motor PF :\n    cos  = %.4f \n",
    cos_theta);

printf(" \n e: Stator Power Input :\n    SPI = %d W
    \n",SPI);

```

```
printf(" \n f: Stator Copper Loss :\n    SCL = %.1f
  W \n",SCL);

printf(" \n g: Rotor Power Input :\n    RPI = %.1f W
   (method 1) ", RPI_1);
printf(" \n    RPI = %.1f W (method 2)\n",RPI_2);

printf(" \n h: Rotor copper loss :\n    RCL = %.1f W
   \n",RCL);

printf(" \n i: Rotor Power Developed :\n    RPD = %
   .1f W \n",RPD_1);

printf(" \n    RPD = %.1f W \n ",RPD_2);

printf(" \n j: Total 3-phase rotor power:\n    RPO =
    %f W \n",RPO);

printf(" \n k: Total output torque developed :\n
   T_o = %.2f lb-ft\n",T_o);

printf(" \n l: Output horsepower : \n    hp = %.2f
   hp (rated 7.5 hp)\n",hp);

printf(" \n m: Motor efficiency at rated load :\n
         = %.2f percent \n",eta);

printf(" \n n: See Fig.12-12");
```

Scilab code Exa 12.17 upper and lower limit Is

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom
```

```

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS OF DC AND AC DYNAMOS
// Example 12-17

clear; clc; close; // Clear the work space and console.

// Given data
// code letter = J of SCIM (Ex.12-16)

// Calculations
// case a
// From Appendix A-3,Table 430-7(b),the starting kVA/hp (with rotor locked) is
// less than 7.99,which, when substituted in the following equation, yields a
// maximum starting current of :

// subscript u for I_s indicates upper limit of starting current
I_s_u = (7.99*(7.5*1000))/(sqrt(3)*220) ;

// case b
// The lower limit,code letter J,is 7.1 kVA/hp. Thus :

// subscript l for I_s indicates lower limit of starting current
I_s_l = (7.1*(7.5*1000))/(sqrt(3)*220) ;

// Display the results
disp("Example 12-17 Solution : ");

printf(" \n a: From Appendix A-3,Table 430-7(b),the starting kVA/hp ");
printf(" \n    (with rotor locked) is less than 7.99,which, when substituted ");
```

```
printf(" \n     in the following equation, yields a
    maximum starting current of :");
printf(" \n     I_s = %.1f A \n",I_s_u);

printf(" \n b: The lower limit,code letter J,is 7.1
    kVA/hp.\n     Thus :");
printf(" \n     I_s = %.1f A ",I_s_l );
```

Scilab code Exa 12.18 starting I and PF

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS
    OF DC AND AC DYNAMOS
// Example 12-18

clear; clc; close; // Clear the work space and
    console.

// Given data (Ex.12-16)
// code letter = J
P = 6 ; // Number of poles
S_r = 1176 ; // rotor speed in rpm
V = 220 ; // Rated voltage of SCIM in volt
f = 60 ; // Frequency in Hz
hp_SCIM = 7.5 ; // Power rating of SCIM in hp

R_ap = 0.3 ; // armature resistance in ohm/phase
R_r = 0.144 ; // rotor resistance in ohm/phase
jX_m = 13.5 ; // reactance in ohm/phase
jX_s = 0.5 ; // synchronous reactance in ohm/phase
jX_lr = 0.2 ; // Locked rotor reactance in ohm/phase
```

```
P_r = 300 ; // Rotational losses in W
s = 1 ; // unity slip

disp("Example 12-18 Solution : ");

printf(" \n The ratio R_r/s = %.3f ohm, in fig.12-11, using the format method ",R_r/s);
printf(" \n of mesh analysis,we may write the array by inspection :\n ");
printf(" \n_______________________________________________");
printf(" \n \t     I_1 \t I_2 \t\t V ");
printf(" \n_______________________________________________");
printf(" \n\t (0.3+j14)   -(0+j13.5) \t(127+j0)");
printf(" \n\t-(0+j13.5)   (0.144+j13.7) \t 0");
printf(" \n_______________________________________________\n");

// Calculations

A = [ (0.3 + %i*14) -%i*13.5 ; (-%i*13.5)  (0.144 + %i*13.7) ]; // Matrix containing above mesh eqns array
delta = det(A); // Determinant of A

// case a : Starting stator current I_s per phase in A
I_s = det( [ (127+%i*0) (-%i*13.5) ; 0 (0.144 + %i*13.7) ] ) / delta ;
I_s_m = abs(I_s);//I_s_m=magnitude of I_s in A
I_s_a = atan(imag(I_s) /real(I_s))*180/%pi;//I_s_a=phase angle of I_s in degrees

// case b : power factor of the motor at starting
```

```
theta = I_s_a ; // Motor PF angle in degrees
cos_theta = cosd(theta); // Motor PF

// Display the results
disp("Solution : ");
printf(" \n a: Starting stator current of SCIM :\n I_s = I_1 = ");disp(I_s);
printf(" \n    I_s = I_1 = %.2f <%.2f A \n ",I_s_m , I_s_a );

printf(" \n b: Power factor of the motor at starting :\n    cos    = %.4f      %.3f\n",cos_theta, cos_theta);

printf(" \n    Note : I_s = %.2f A calculated in Ex.12-18 falls between the limits",I_s_m);
printf(" \n             found in Ex.12-17. This verifies the mesh analysis technique.");
```

Scilab code Exa 12.19 Re1s slip Pcu and Pr at LFs hp T

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 12: POWER,ENERGY,AND EFFICIENCY RELATIONS OF DC AND AC DYNAMOS
// Example 12-19

clear; clc; close; // Clear the work space and console.

// Given data
V = 220 ; // Rated voltage of SCIM in volt
```

```
f = 60 ; // Frequency in Hz
P = 4 ; // Number of poles
PF = 0.85 ; // power factor of capacitor start IM
// nameplate details
hp_IM = 5 ; // power rating of IM in hp
I_L = 28 ; // Rated line current in A
S_r = 1620 ; // Rotor speed of IM in rpm

// No-load test data
I_nl = 6.4 ; // No-load line current in A
V_nl = 220 ; // No-load line voltage in volt
P_nl = 239 ; // No-load power reading in W
s_nl = 0.01 ; // No-load slip

// Blocked rotor test
I_br = 62 ; // Blocked rotor line current in A
V_br = 64 ; // Blocked rotor voltage in volt
P_br = 1922 ; // Blocked rotor power reading in W
s_br = 1 ; // blocked rotor slip(unity)

// Calculations
// case a
R_e1s = P_br / (I_br^2); // Equivalent total
    resistance of IM in ohm

// case b
P_in = P_nl ; // Input power to IM in W
I_1s = I_nl ; // Input current in A
P_ro = P_in - ((I_1s)^2 * R_e1s); // Rotational
    losses in W

// case c
S = (120*f/P); // Speed of synchronous magnetic
    field in rpm
S_fl = S_r ; // Full-load rotor speed of IM in rpm
s_fl = (S - S_fl)/S ; // Full-load Slip

LF1 = 1/4 ; // Load fraction
```

```
LF2 = 1/2 ; // Load fraction
LF3 = 3/4 ; // Load fraction
LF4 = 5/4 ; // Load fraction

s_LF1 = s_fl*LF1 ; // slip at 1/4 rated load
s_LF2 = s_fl*LF2 ; // slip at 1/2 rated load
s_LF3 = s_fl*LF3 ; // slip at 3/4 rated load
s_LF4 = s_fl*LF4 ; // slip at 5/4 rated load

// case d
s_o = s_nl ; // No-load slip
P_rs_LF1 = P_ro * (1 - s_LF1)/(1 - s_o); // Rotational losses in W at s_LF1
P_rs_LF2 = P_ro * (1 - s_LF2)/(1 - s_o); // Rotational losses in W at s_LF2
P_rs_LF3 = P_ro * (1 - s_LF3)/(1 - s_o); // Rotational losses in W at s_LF3
P_rs_fl = P_ro * (1 - s_fl)/(1 - s_o); // Rotational losses in W at full-load slip
P_rs_LF4 = P_ro * (1 - s_LF4)/(1 - s_o); // Rotational losses in W at s_LF4

// case e
I1s = I_L ; //  Line current in A
P_cu_fl = (I1s)^2*R_e1s ; // Equivalent copper loss at full-load slip
P_cu_LF1 = (LF1)^2 * P_cu_fl ; // Equivalent copper loss at s_LF1
P_cu_LF2 = (LF2)^2 * P_cu_fl ; // Equivalent copper loss at s_LF2
P_cu_LF3 = (LF3)^2 * P_cu_fl ; // Equivalent copper loss at s_LF3
P_cu_LF4 = (LF4)^2 * P_cu_fl ; // Equivalent copper loss at s_LF4

// case f
Input = V*I_L*PF ; // Input to single phase capacitor start IM
```

```
// Efficiency at 1/4 rated load
eta_LF1 = ( Input*LF1 - (P_rs_LF1 + P_cu_LF1) ) / ( Input*LF1) * 100 ;

// Efficiency at 1/2 rated load
eta_LF2 = ( Input*LF2 - (P_rs_LF2 + P_cu_LF2) ) / ( Input*LF2) * 100 ;

// Efficiency at 3/4 rated load
eta_LF3 = ( Input*LF3 - (P_rs_LF3 + P_cu_LF3) ) / ( Input*LF3) * 100 ;

// Efficiency at rated load
eta_fl = ( Input - (P_rs_fl + P_cu_fl) ) / (Input) * 100 ;

// Efficiency at 5/4 rated load
eta_LF4 = ( Input*LF4 - (P_rs_LF4 + P_cu_LF4) ) / ( Input*LF4) * 100 ;

// case g
// since eta is calculated in percent divide it by 100 for hp calculations
P_o_LF1 = (Input*LF1*eta_LF1/100)/746 ; // Output hp at 1/4 rated load
P_o_LF2 = (Input*LF2*eta_LF2/100)/746 ; // Output hp at 1/2 rated load
P_o_LF3 = (Input*LF3*eta_LF3/100)/746 ; // Output hp at 3/4 rated load
P_o = (Input*eta_fl/100)/746 ; // Output hp at 1/4 rated load
P_o_LF4 = (Input*LF4*eta_LF4/100)/746 ; // Output hp at 5/4 rated load

// case h
hp = P_o ; // Rated output horsepower
S_fl = S_r ; // Full-load rotor speed in rpm
```

```
T_o = (P_o*5252)/S_fl ; // Outpue torque at full-load in lb-ft
T_o_Nm = T_o * 1.356 ; // Outpue torque at full-load in N-m

// Display the results
disp("Example 12-19 Solution : ");

printf(" \n a: Equivalent total resistance of IM :\n       R_e1s = %.1f      \n",R_e1s);

printf(" \n b: Rotational losses :\n     P_ro = %.1f W \n ",P_ro);

printf(" \n c: Slip at rated load : s = %.1f \n Slip,",s_fl);
printf(" \n     s at %.2f rated load = %.3f",LF1,s_LF1);
printf(" \n     s at %.2f rated load = %.3f",LF2,s_LF2);
printf(" \n     s at %.2f rated load = %.3f",LF3,s_LF3);
printf(" \n     s at %.2f rated load = %.3f \n ",LF4,s_LF4);

printf(" \n d: Rotational losses :\n ");
printf(" \n     P_r at at %.2f rated load = %.1f W ",LF1,P_rs_LF1);
printf(" \n     P_r at at %.2f rated load = %.1f W ",LF2,P_rs_LF2);
printf(" \n     P_r at at %.2f rated load = %.1f W ",LF3,P_rs_LF3);
printf(" \n     P_r at at full load = %.1f W ",P_rs_fl);
printf(" \n     P_r at at %.2f rated load = %.1f W \n ",LF4,P_rs_LF4);

printf(" \n e: At full-load, P_cu = %d W \n",P_cu_fl
```

```
);
printf(" \n    P_cu at %.2f rated load = %.2f W",LF1,P_cu_LF1)
printf(" \n    P_cu at %.2f rated load = %.2f W",LF2,P_cu_LF2)
printf(" \n    P_cu at %.2f rated load = %.2f W",LF3,P_cu_LF3)
printf(" \n    P_cu at %.2f rated load = %.2f W \n",LF4,P_cu_LF4)

printf(" \n f: Full-load input = %.f W \n",Input);
printf(" \n    Efficiency :\n       at %.2f rated load = %.1f percent \n",LF1,eta_LF1);
printf(" \n       at %.2f rated load = %.1f percent \n",LF2,eta_LF2);
printf(" \n       at %.2f rated load = %.1f percent \n",LF3,eta_LF3);
printf(" \n       at rated load =  _fl = %.1f percent \n",eta_fl);
printf(" \n       at %.2f rated load = %.1f percent \n",LF4,eta_LF4);
printf(" \n    Please note: Calculation error for _fl in textbook.\n");

printf(" \n g: Output horsepower :\n    P_o at %.2f rated load = %.3f hp \n",LF1,P_o_LF1);
printf(" \n    P_o at %.2f rated load = %.3f hp \n",LF2,P_o_LF2);
printf(" \n    P_o at %.2f rated load = %.3f hp \n",LF3,P_o_LF3);
printf(" \n    P_o at rated load = %.3f hp \n",P_o);
printf(" \n    P_o at %.2f rated load = %.3f hp \n",LF4,P_o_LF4);

printf(" \n h: Output torque at full-load :\n    T_o = %.1f lb-ft",T_o);
printf(" \n    T_o = %.2f N-m     %.1f N-m",T_o_Nm,T_o_Nm);
```

Chapter 13

RATINGS SELECTION AND MAINTENANCE OF ELECTRIC MACHINERY

Scilab code Exa 13.1 R and reduced life expectancy

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 13: RATINGS,SELECTION,AND MAINTENANCE OF
   ELECTRIC MACHINERY
// Example 13-1

clear; clc; close; // Clear the work space and
   console.

// Given data
// MOTOR(class A insulation ) is operated for 6 hrs
T = 125 ; // Temperature in degree celsius recorded
   by the embedded detectors
life_orig = 10 ; // Life in years of the motor (
```

```
   standard)

// Calculations
delta_T = T - 105 ; // Positive temperature
   difference between the given
// max hottest spot temperature of its insulation
   and the ambient temperature recorded.
// 105 is chosen from table 13-1(class A insulation)
R = 2 ^ (delta_T/10); // Life reduction factor

Life_calc = life_orig / R ; // Reduced life
   expectancy of the motor in years

// Display the results
disp("Example 13-1 Solution : ");
printf(" \n Life reduction factor : R = %d \n  ",R )
   ;
printf(" \n Reduced life expectancy of the motor :
   Life_calc = %.1f years",Life_calc);
```

Scilab code Exa 13.2 E and increased life expectancy

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 13: RATINGS,SELECTION,AND MAINTENANCE OF
   ELECTRIC MACHINERY
// Example 13-2

clear; clc; close; // Clear the work space and
   console.

// Given data
```

```
// MOTOR(class A insulation ) is operated for 6 hrs
T = 75 ; // Temperature in degree celsius recorded by the embedded detectors
life_orig = 10 ; // Life in years of the motor (standard)

// Calculations
delta_T = 105 - T ; // Positive temperature difference between the given
// max hottest spot temperature of its insulation and the ambient temperature recorded.
// 105 is chosen from table 13-1 (class A insulation )
E = 2 ^ (delta_T/10); // Life extension factor

Life_calc = life_orig * E ; // Increased life expectancy of the motor in years

// Display the results
disp("Example 13-2 Solution : ");
printf(" \n Life extension factor : E = %d \n ",E );
printf(" \n Increased life expectancy of the motor : Life_calc = %d years ",Life_calc);
```

Scilab code Exa 13.3 E and increased life expectancy classB

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 13: RATINGS,SELECTION,AND MAINTENANCE OF ELECTRIC MACHINERY
// Example 13-3

```

```
clear; clc; close; // Clear the work space and console.

// Given data
// Class A insulation
T_A = 105 ; // Temperature in degree celsius recorded by the embedded detectors
life_orig = 5 ; // Life in years of the motor (standard)
// Class B insulation
T_B = 130 ; // Temperature in degree celsius recorded by the embedded detectors

// Calculations
delta_T = T_B - T_A ; // Positive temperature difference betw the given
// max hottest spot temperature of its insulation and the ambient temperature recorded.
// T_A and T_B are chosen from table 13-1
E = 2 ^ (delta_T/10); // Life extension factor

Life_calc = life_orig * E ; // Increased life expectancy of the motor in years

// Display the results
disp("Example 13-3 Solution : ");
printf(" \n Life extension factor : E = %.2f \n ",E);
printf(" \n Increased life expectancy of the motor : Life_calc = %.1f years ",Life_calc);
```

Scilab code Exa 13.4 ClassB insulation SCIM details

```
// Electric Machinery and Transformers
// Irving L kosow
```

```
// Prentice Hall of India
// 2nd editiom

// Chapter 13: RATINGS,SELECTION,AND MAINTENANCE OF
   ELECTRIC MACHINERY
// Example 13-4

clear; clc; close; // Clear the work space and
   console.

// Given data
P_o = 25 ; // Rated power of SCIM in hp
// class B insulation
T_ambient = 40 ; // Standard ambient temperature
   recorded by the embedded hot-spot detectors in
   degree celsius
T_hottest = 115 ; // Hottest-spot winding
   temperature recorded by the embedded hot-spot
   detectors in degree celsius

// Calculations
// case a
// from table 13-1 allowable temperature rise in 90
   degree celsius

// case b
T_rise = T_hottest - T_ambient ; // Actual
   temperature rise for the insulation type used in
   degree celsius

// case c
P_f = P_o * (90/T_rise); // Approximate power to the
    motor that can be delivered at T_rise

// case d
// same as P_f

// case e
```

```
// answer from case a

// Display the results
disp("Example 13-4 Solution : ");
printf(" \n a: The allowable temperature rise for
    the ");
printf(" \n    insulation type used = 90 degree
    celsius(from table 13-1)\n");

printf(" \n b: The actual temperature rise for the
    insulation type used = %d degree celsius\n",
    T_rise);

printf(" \n c: The approximate power to the motor
    that can be delivered at T_rise");
printf(" \n    P_f = %d hp\n",P_f);

printf(" \n d: Power rating that may be stamped on
    the nameplate = %d hp(subject to test at this
    load) \n ",P_f);

printf(" \n e: The temperature rise that must be
    stamped on the nameplate = 90 degree celsius");
```

Scilab code Exa 13.5 final temperature

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 13: RATINGS,SELECTION,AND MAINTENANCE OF
    ELECTRIC MACHINERY
// Example 13-5

```

```
clear; clc; close; // Clear the work space and console.

// Given data
P_o = 50 ; // Power rating of the WRIM in hp
// Class F insulation
T_hottest = 160 ; // Hottest-spot winding temperature recorded by the embedded
// hot-spot detectors in degree celsius
T_ambient = 40 ; // Standard ambient temperature recorded by the embedded
// hot-spot detectors in degree celsius
P_f_a = 40 ; // Power rating of load a in hp
P_f_b = 55 ; // Power rating of load a in hp

// Calculations
// case a
delta_T_o = T_hottest - T_ambient ; // Temperature rise for the insulation type
// used in degree celsius

// subscript a in delta_T_f ,P_f_a and T_f indicates case a
delta_T_f_a = (P_f_a/P_o)*delta_T_o ; // final temperature rise in degree celsius
T_f_a = delta_T_f_a + T_ambient ; // Approximate final hot-spot temperature in degree celsius

// case b
// subscript b in delta_T_f ,P_f and T_f indicates case b
delta_T_f_b = (P_f_b/P_o)*delta_T_o ; // final temperature rise in degree celsius
T_f_b = delta_T_f_b + T_ambient ; // Approximate final hot-spot temperature in degree celsius

// Display the results
disp("Example 13-5 Solution : ");
```

```
printf(" \n a:  T_o = %d degree celsius ",delta_T_o
    );
printf(" \n     T_f = %d degree celsius ",
    delta_T_f_a);
printf(" \n    T_f = %d degree celsius \n",T_f_a);

printf(" \n b:  T_f = %d degree celsius ",
    delta_T_f_b);
printf(" \n    T_f = %d degree celsius \n",T_f_b);
printf(" \n    Yes,motor life is reduced at the 110
    percent motor load because");
printf(" \n    the allowable maximum hot-spot motor
    temperature for Class F");
printf(" \n    insulation is 155 degree celsius.");
```

Scilab code Exa 13.6 Tf R decreased life expectancy

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 13: RATINGS,SELECTION,AND MAINTENANCE OF
    ELECTRIC MACHINERY
// Example 13-6

clear; clc; close; // Clear the work space and
    console.

// Given data
P_o = 55 ; // Power rating of the WRIM in hp
T_ambient = 40 ; // Standard ambient temperature
    recorded by the embedded
// hot-spot detectors in degree celsius
life_orig = 10 ; // Life in years of the motor (
```

```
standard)

// Calculated data from Ex.13-5b
T_f = 172 ; // Approximate final hot-spot
    temperature in degree celsius

// Calculations
delta_T = T_f - 155 ; // Positive temperature
    difference betw the given
// max hottest spot temperature of its insulation
    and the ambient temperature recorded.
// 155 is chosen from table 13-1(class F insulation)

R = 2 ^ (delta_T/10); // Life reduction factor

Life_calc = life_orig / R ; // Reduced life
    expectancy of the motor in years

// Display the results
disp("Example 13-6 Solution : ");
printf(" \n From Ex.13-5b, T_f = %d degree celsius\n"
    ,T_f);
printf(" \n Life reduction factor : R = %.2f \n  ",R
    );
printf(" \n Reduced life expectancy of the motor :
    Life_calc = %.2f years",Life_calc);
```

Scilab code Exa 13.7 rms hp

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 13: RATINGS,SELECTION,AND MAINTENANCE OF
```

```
   ELECTRIC MACHINERY
// Example 13-7

clear; clc; close; // Clear the work space and console.

// Given data
P_o = 200 ; // Power rating of the test motor in hp
t1 = 5 ; // time duration in minutes for which test motor is operated at 200 hp
t2 = 5 ; // time duration in minutes for which test motor is operated at 20 hp
t3 = 10 ; // time duration in minutes for which test motor is operated at 100 hp

// Calculation
rms_hp = sqrt( ( (200^2)*t1 + (20^2)*t2 + (100^2)*t3 )/(t1 + t2 + t3 + 10/3) );
// Horsepower required for intermittent varying load

// Display the results
disp("Example 13-7 Solution : ");
printf(" \n Horsepower required for intermittent varying load is : ");
printf(" \n rms hp = %.f hp \n ",rms_hp);

printf(" \n A 125 hp motor would be selected because that is the nearest larger");
printf(" \n commercial standard rating.This means that the motor would operate ");
printf(" \n with a 160 percent overload (at 200 hp) for 5 minutes,or 1/6th of ")
printf(" \n its total duty cycle.");
```

Scilab code Exa 13.8 Vb Ib Rb Rpu

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 13: RATINGS,SELECTION,AND MAINTENANCE OF
   ELECTRIC MACHINERY
// Example 13-8

clear; clc; close; // Clear the work space and
   console.

// Given data
V = 120 ; // Rated output voltage in volt of
   separately excited dc generator
I = 100 ; // Rated output current in A of separately
    excited dc generator
R = 0.1 ; // Armature resistance in ohm

// Calculations
// case a
V_b = V ; // base voltage in volt

// case b
I_b = I ; // base current in A

// case c
R_b = V_b / I_b ; // base resistance in ohm

// case d
R_pu = R / R_b ; // per-unit value of armature
   resistance in p.u

// Display the results
disp("Example 13-8 Solution : ");

printf(" \n a: Base voltage \n    V_b = %d V \n ",
   V_b );
```

```
printf(" \n b: Base current \n     I_b = %d A \n ",
   I_b );

printf(" \n c: Base resistance \n     R_b = %.1f ohm
   \n ", R_b );

printf(" \n d: Per-unit value of armature resistance
   \n     R_p.u = %.3f p.u \n ", R_pu );
```

Scilab code Exa 13.9 Rpu jXpu Zpu

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 13: RATINGS,SELECTION,AND MAINTENANCE OF
   ELECTRIC MACHINERY
// Example 13-9

clear; clc; close; // Clear the work space and
   console.

// Given data
// single phase alternator
V = 500 ; // Rated voltage of the alternator in volt
P = 20 ; // Rated power of the alternator in kVA
I = 40 ; // Rated current of the alternator in A
R = 2 ; // Armature resistance in ohm
X = 15 ; // Armature reactance in ohm

// Calculations
// case a
V_b = V ; // base voltage in volt
```

```
I_b = I ; // base current in A
R_pu = (R*I_b)/V_b ; // per-unit value of armature resistance in p.u

// case b
jX_pu = (X*I_b)/V_b ; // per-unit value of armature reactance in p.u

// case c
// subscript 1 indicates method 1 for finding Z_p.u
Z_pu1 = R_pu + %i*(jX_pu); // per-unit value of armature impedance in p.u
Z_pu1_m = abs(Z_pu1);//Z_pu1_m = magnitude of Z_pu1 in p.u
Z_pu1_a = atan(imag(Z_pu1) /real(Z_pu1))*180/%pi;//Z_pu1_a=phase angle of Z_pu1 in degrees

// subscript 2 indicates method 2 for finding Z_p.u
Z_pu2 = (R + %i*X)*(I/V); // per-unit value of armature impedance in p.u
Z_pu2_m = abs(Z_pu2);//Z_pu2_m = magnitude of Z_pu2 in p.u
Z_pu2_a = atan(imag(Z_pu2) /real(Z_pu2))*180/%pi;//Z_pu2_a=phase angle of Z_pu2 in degrees

// Display the results
disp("Example 13-9 Solution : ");

printf(" \n a: Armature resistance per unit value\n    R_p.u = %.2f p.u \n",R_pu);

printf(" \n b: Armature reactance per unit value\n    jX_p.u in p.u = ");disp(%i*jX_pu);

printf(" \n c: Armature impedance per unit value\n");
printf(" \n    (method 1)\n    Z_p.u in p.u = ");disp(Z_pu1);
```

```
printf(" \n     Z_p.u = %.3f <%.1f p.u \n",Z_pu1_m,Z_pu1_a);

printf(" \n     (method 2)\n     Z_p.u in p.u = "); disp(Z_pu2);
printf(" \n     Z_p.u = %.3f <%.1f p.u \n",Z_pu2_m,Z_pu2_a);
```

Scilab code Exa 13.10 new Zpu

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 13: RATINGS,SELECTION,AND MAINTENANCE OF ELECTRIC MACHINERY
// Example 13-10

clear; clc; close; // Clear the work space and console.

// Given data
// single phase alternator
V_orig = 500 ; // Rated voltage of the alternator in volt
kVA_orig = 20 ; // Rated power of the alternator in kVA
I = 40 ; // Rated current of the alternator in A
R = 2 ; // Armature resistance in ohm
X = 15 ; // Armature reactance in ohm

V_new = 5000 ; // New voltage of the alternator in volt
kVA_new = 100 ; // New power of the alternator in
```

```
kVA

// Calculated armature impedance from Ex.13-9c
Z_pu_orig = 1.211 ; // original per-unit value of
    armature impedance in p.u

// Calculation
Z_pu_new = Z_pu_orig * (kVA_new/kVA_orig) * (V_orig/
    V_new)^2 ;
// new per-unit value of armature impedance in p.u

// Display the results
disp("Example 13-10 Solution : ");
printf(" \n New per-unit value of armature impedance
    \n Z_pu(new) = %.5f p.u",Z_pu_new);
```

Scilab code Exa 13.11 line and phase Vpu

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 13: RATINGS,SELECTION,AND MAINTENANCE OF
    ELECTRIC MACHINERY
// Example 13-11

clear; clc; close; // Clear the work space and
    console.

// Given data
// 3-phase distribution system
V = 2300 ; // Line voltage of 3-phase distribution
    system in volt
V_p = 1328 ; // Phase voltage of 3-phase
```

```
    distribution system in volt

V_b = 69000 ; // Common base line voltage in volt
V_pb = 39840 ; // Common base phase voltage in volt

// Calculations
// case a
V_pu_line = V / V_b ; // Distribution system p.u line voltage

// case a
V_pu_phase = V_p / V_pb ; // Distribution system p.u phase voltage

// Display the results
disp("Example 13-11 Solution : ");
printf(" \n a: Distribution system p.u line voltage :\n      V_pu = %.2f p.u\n",V_pu_line);

printf(" \n b: Distribution system p.u phase voltage :\n      V_pu = %.2f p.u\n",V_pu_phase);
```

Scilab code Exa 13.12 Zb Xs Ra Zs P

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 13: RATINGS,SELECTION,AND MAINTENANCE OF ELECTRIC MACHINERY
// Example 13-12

clear; clc; close; // Clear the work space and console.
```

```
// Given data
VA_b = 50 ; // Base power rating of the 3-phase Y-
    connected alternator in MVA
V_b = 25 ; // Base voltage of the 3-phase Y-
    connected alternator in kV
X_pu = 1.3  ; // per unit value of synchronous
    reactance
R_pu = 0.05  ; // per unit value of resistance

// Calculations
// case a
// subscript 1 for Z_b indicates method 1 for
    finding Z_b
Z_b1 = (V_b)^2 / VA_b ; // Base impedance in ohm

// subscript 2 for Z_b indicates method 2 for
    finding Z_b
S_b = VA_b ; // Base power rating of the 3-phase Y-
    connected alternator in MVA
I_b = (S_b)/V_b ; // Base current in kA
Z_b2 = V_b / I_b ; // Base impedance in ohm

// case b
Z_b = Z_b1; // Base impedance in ohm
X_s = X_pu * Z_b ; // Actual value of synchronous
    reactance per phase in ohm

// case c
R_a = R_pu * Z_b ; // Actual value of armature
    stator resistance per phase in ohm

// case d
// subscript 1 for Z_s indicates method 1 for
    finding Z_s
Z_s1 = R_a + %i*X_s ; // Synchronous impedance per
    phase in ohm
Z_s1_m = abs(Z_s1);//Z_s1_m = magnitude of Z_s1 in
```

```
ohm
Z_s1_a = atan(imag(Z_s1) /real(Z_s1))*180/%pi;//Z_s1_a=phase angle of Z_s1 in degrees

// subscript 2 for Z_s indicates method 2 for finding Z_s
Z_pu = R_pu + %i*X_pu ; // per unit value of impedance
Z_s2 = Z_pu * Z_b ; // Synchronous impedance per phase in ohm
Z_s2_m = abs(Z_s2);//Z_s2_m = magnitude of Z_s2 in ohm
Z_s2_a = atan(imag(Z_s2) /real(Z_s2))*180/%pi;//Z_s2_a=phase angle of Z_s2 in degrees

// case e
S = S_b ; // Base power rating of the 3-phase Y-connected alternator in MVA
P = S * R_pu ; // Full-load copper losses for all three phases in MW

// Display the results
disp("Example 13-12 Solution : ");

printf(" \n a: Base impedance(method 1): \n     Z_b = %.1f ohm\n",Z_b1);
printf(" \n     Base impedance(method 2) : ");
printf(" \n     I_b = %d kA \n     Z_b = %.1f ohm\n",I_b,Z_b2);

printf(" \n b: Actual value of synchronous reactance per phase : ");
printf(" \n     X_s in ohm = ");disp(%i*X_s);

printf(" \n c: Actual value of armature stator resistance per phase : ");
printf(" \n     R_a = %.3f ohm \n ",R_a );

```

```
printf(" \n d: Synchronous impedance per phase (
    method 1): ");
printf(" \n     Z_s in ohm = ");disp(Z_s1);
printf(" \n     Z_s = %.2f <%.1f ohm\n",Z_s1_m,Z_s1_a
    );
printf(" \n     Synchronous impedance per phase (
    method 2) : ");
printf(" \n     Z_s in ohm = ");disp(Z_s2);
printf(" \n     Z_s = %.2f <%.1f ohm\n",Z_s2_m,Z_s2_a
    );

printf(" \n e: Full-load copper losses for all 3
    phases : \n     P = %.1f MW",P);
```

Chapter 14

TRANSFORMERS

Scilab code Exa 14.1 stepup stepdown alpha I1

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-1

clear; clc; close; // Clear the work space and
   console.

// Given data for Step -down transformer
N_1 = 500 ; // Number of turns in the primary
N_2 = 100 ; // Number of turns in the secondary
I_2 = 12 ; // Load (Secondary) current in A

// Calculations
// case a
alpha = N_1 / N_2 ; // Transformation ratio

// case b
```

```
I_1 = I_2 / alpha ; // Load component of primary
    current in A

// case c
// sunscript c for alpha indicates case c
// For step up transformer , using above given data
N1 = 100 ; // Number of turns in the primary
N2 = 500 ; // Number of turns in the secondary
alpha_c = N1 / N2 ; // Transformation ratio

// Display the results
disp("Example 14-1 Solution : ");

printf(" \n a: Transformation ratio(step-down
    transformer) :\n        = %d\n",alpha);

printf(" \n b: Load component of primary current : \
    n     I_1 = %.1f A \n",I_1);

printf(" \n c: Transformation ratio(step-up
    transformer) :\n        = %.1f",alpha_c);
```

Scilab code Exa 14.2 turns I1 I2 stepup stepdown alpha

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-2

clear; clc; close; // Clear the work space and
    console.

```

```
// Given data
V_h = 2300 ; // high voltage in volt
V_l = 115 ; // low voltage in volt
V_1 = 2300 ; // Primary voltage in volt
V_2 = 115 ; // Secondary voltage in volt
f = 60 ; // Frequency in Hz
S = 4.6 ; // kVA rating of the step-down transformer
S_1 = S ;
S_2 = S ;
V_per_turn = 2.5 ; // Induced EMF per turn in V/turn
// Ideal transformer

// Calculations
// case a
N_h = V_h / V_per_turn ; // Number of high-side turns
N_l = V_l / V_per_turn ; // Number of low-side turns

N_1 = N_h;// Number of turns in the primary
N_2 = N_l;// Number of turns in the secondary

// case b
I_1 = S_1*1000 / V_1 ; // Rated primary current in A
I_2 = S_2*1000 / V_2 ; // Rated secondary current in A

I_h = 2 ; // Rated current in A on HV side
I_l = 40 ; // Rated current in A on LV side

// case c
// subscript c for alpha_stepdown and alpha_stepup indicates case c
alpha_stepdown_c = N_1 / N_2 ; // step-down transformation ratio
alpha_stepup_c = N_l / N_h ; // step-up transformation ratio

// case d
```

```
// subscript d for alpha_stepdown and alpha_stepup indicates case d
alpha_stepdown_d = I_2 / I_1 ; // step-down transformation ratio
alpha_stepup_d = I_h / I_l ; // step-up transformation ratio

// Display the results
disp("Example 14-2 Solution : ");

printf(" \n a: Number of high-side turns :\n    N_h = %d t = N_1 \n",N_h);
printf(" \n    Number of low-side turns :\n    N_l = %d t = N_2\n",N_l);

printf(" \n b: Rated primary current :\n    I_h = I_1 = %d A \n",I_1);
printf(" \n    Rated secondary current :\n    I_l = I_2 = %d A\n",I_2);

printf(" \n c: step-down transformation ratio :\n         = N_1/N_2 = %d\n",alpha_stepdown_c);
printf(" \n    step-up transformation ratio :\n      = N_l/N_h = %.2f\n",alpha_stepup_c);

printf(" \n d: step-down transformation ratio :\n         = I_2 / I_1 = %d\n",alpha_stepdown_d);
printf(" \n    step-up transformation ratio :\n      = I_h / I_lh = %.2f\n",alpha_stepup_d);
```

Scilab code Exa 14.3 alpha Z1 I1

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
```

```
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-3

clear; clc; close; // Clear the work space and console.

// Given data
N_1 = 500 ; // Number of primary turns in the audio output transformer
N_2 = 25 ; // Number of secondary turns in the audio output transformer
Z_L = 8 ; // Speaker impedance in ohm
V_1 = 10 ; // Output voltage of the audio output transformer in volt

// Calculations
// case a
alpha = N_1/N_2 ; // step-down transformation ratio
Z_1 = (alpha)^2 * Z_L ; // Impedance reflected to the transformer primary
// at the output of Tr in ohm

// case b
I_1 = V_1 / Z_1 ; // Primary current in A

// Display the results
disp("Example 14-3 Solution : ");

printf(" \n a: Transformation ratio :\n        = %d\n ",alpha);
printf(" \n    Impedance reflected to the transformer primary at the output of Tr :");
printf(" \n    Z_1 = %d ohm \n ",Z_1);

printf(" \n b: Matching transformer primary current :\n    I_1 = %f A",I_1);
```

```
printf(" \n      I_1  = %.3 f  mA  ",1000 * I_1);
```

Scilab code Exa 14.4 Z2prime Z3prime Z1 I1 Pt V2 P2 V3 P3 Pt

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-4

clear; clc; close; // Clear the work space and
   console.

// Given data
N_1 = 600 ; // Number of primary turns
N_2 = 150 ; // Some number of secondary turns
N_3 = 300 ; // Some number of secondary turns
Z_2 = 30 ; // Resistive load in ohm across N_2
Z_3 = 15 ; // Resistive load in ohm across N_3
R_2 = 30 ;
R_3 = 15 ;
V_p = 16 ; // Primary applied voltage in volt
cos_theta = 1 ; // unity PF

// Calculations
// case a
Z_2_prime = Z_2 * (N_1/N_2)^2 ; // Impedance
   reflected to the primary by load Z_2 in ohm

// case b
Z_3_prime = Z_3 * (N_1/N_3)^2 ; // Impedance
   reflected to the primary by load Z_3 in ohm

```

```
// case c
// Total impedance reflected to the primary in ohm
Z_1 = (Z_2_prime * Z_3_prime) / (Z_2_prime +
    Z_3_prime) ;

// case d
I_1 = V_p / Z_1 ; // Total current drawn from the
    supply in A

// case e
P_t = V_p * I_1 * cos_theta ; // Total power in W
    drwan from the supply at unity PF

// case f
V_2 = V_p * (N_2/N_1) ; // Voltage across Z_2 in
    volt
P_2 = (V_2)^2 / R_2 ; // Power dissipated in load
    Z_2 in W

// case g
V_3 = V_p * (N_3/N_1) ; // Voltage across Z_3 in
    volt
P_3 = (V_3)^2 / R_3 ; // Power dissipated in load
    Z_3 in W

// case h
P_total = P_2 + P_3 ; // Total power dissipated in
    both loads in W

// Display the results
disp("Example 14-4 Solution : ");

printf(" \n a: Impedance reflected to the primary by
     load Z_2 : ");
printf(" \n     Z_2 = %d ohm \n ",Z_2_prime );

printf(" \n b: Impedance reflected to the primary by
     load Z_3 : ");
```

```
printf(" \n     Z_3 = %d ohm \n ",Z_3_prime );

printf(" \n c: Total impedance reflected to the
    primary : ");
printf(" \n     Z_1 = %.1f ohm \n ",Z_1 );

printf(" \n d: Total current drawn from the supply :
     ");
printf(" \n     I_1 = %.1f A \n ",I_1 );

printf(" \n e: Total power drawn from the supply at
    unity PF : ");
printf(" \n     P_t = %.1f W \n ",P_t );

printf(" \n f: Voltage across Z_2 in volt :\n     V_2
     = %d V \n ",V_2 );
printf(" \n     Power dissipated in load Z_2 :\n
    P_2 = %.2f W \n",P_2 );

printf(" \n g: Voltage across Z_3 in volt :\n     V_3
     = %d V \n ",V_3 );
printf(" \n     Power dissipated in load Z_3 :\n
    P_3 = %f W \n",P_3 );

printf(" \n h: Total power dissipated in both loads
    :\n     P_t = %.1f W",P_total);
```

Scilab code Exa 14.5 alpha N2 N1 ZL

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
```

```
// Example 14-5

clear; clc; close; // Clear the work space and
    console.

// Given data
P = 100 ; // Power rating of the single channel
    power amplifier in W
Z_p = 3200 ; // Output impedance in ohm of the
    single channel power amplifier
N_p = 1500 ; // Number of primary turns in a tapped
    impedance-matching transformer
Z_L1 = 8 ; // Amplifier output in ohm using a tapped
     impedance-matching transformer
Z_L2 = 4 ; // Amplifier output in ohm using a tapped
     impedance-matching transformer

// Calculations
// case a
alpha = sqrt(Z_p/Z_L1) ; // Transformation ratio
N_2 = N_p / alpha ; // Total number of secondary
    turns to match 8 ohm speaker

// case b
// subscript b for alpha indicates case b
alpha_b = sqrt(Z_p/Z_L2) ; // Transformation ratio
N_1 = N_p / alpha_b ; // Number of primary turns to
    match 4 ohm speaker

// case c
turns_difference = N_2 - N_1 ; // Difference in
    secondary and primary turns
// subscript c for alpha indicates case c
alpha_c = (1500/22); // Transformation ratio
Z_L = Z_p / (alpha_c)^2 ; // Impedance that must be
    connected between 4 ohm and
// 8 ohm terminals to reflect a primary impedance of
     3.2 kilo-ohm
```

```

// Display the results
disp("Example 14-5 Solution : ");

printf(" \n a: Transformation ratio : \n         = %d \n ",alpha );
printf(" \n     Total number of secondary turns to match 8 ohm speaker : ");
printf(" \n     N_2 = %d t \n ",N_2 );

printf(" \n b: Transformation ratio : \n         = %.3 f \n ",alpha_b );
printf(" \n     Number of primary turns to match 4 ohm speaker : ");
printf(" \n     N_1 = %d t \n ",N_1 );

printf(" \n c: Difference in secondary and primary turns : ");
printf(" \n     N_2 - N_1 = %. f t \n ",turns_difference );
printf(" \n     Impedance that must be connected between 4 ohm and 8 ohm ");
printf(" \n     terminals to reflect a primary impedance of 3.2 kilo-ohm : ");
printf(" \n     Z_L = %.2 f ohm ",Z_L );
```

Scilab code Exa 14.6 Z between terminals A B

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-6
```

```
clear; clc; close; // Clear the work space and console.

// Given data
P = 100 ; // Power rating of the single channel power amplifier in W
Z_L1 = 8 ; // Amplifier output in ohm using a tapped impedance-matching transformer
Z_L2 = 4 ; // Amplifier output in ohm using a tapped impedance-matching transformer
P_servo = 10 ; // Power rating of the servo motor in W
Z_servo = 0.7 ; // Impedance of the servo motor in ohm

// Calculations
root_Z_AB = sqrt(8) - sqrt(4);
Z_AB = (root_Z_AB)^2;

// Display the resulta
disp("Example 14-6 Solution : ");

printf(" \n Z_p = %d *(N_p/N_1)^2 = %d *(N_p/N_2)^2\n",Z_L2,Z_L1);
printf(" \n     = Z_AB * (N_p/(N_2 - N_1)^2 ) \n");
printf(" \n Dividing each of the three numerators by N_p^2 and taking the ");
printf(" \n square root of each term, we have\n");

printf(" \n    (Z_AB)/(N_2 - N_1) =     (4)/N_1 = (8)/N_2 \n");
printf(" \n    (Z_AB)/(N_2 - N_1) =     (4)/N_1 - (8)/N_2 \n");
printf(" \n yielding,     (Z_AB) =     (8) -     (4) = %f \n",root_Z_AB);
printf(" \n which Z_AB = (%f)^2 = %.2f     \n",root_Z_AB,Z_AB);
```

Scilab code Exa 14.7 alpha V1 V2 I2 I1 PL Ps PT efficiency

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-7

clear; clc; close; // Clear the work space and console.

// Given data
V = 10 * exp(%i * 0 * (%pi/180)); // Supply voltage of the source 10<0 V
R_s = 1000 ; // Resistance of the source in ohm
R_L = 10 ; // Load resistance in ohm
Z_L = R_L ; // Load resistance in ohm

// Calculations
// case a
alpha = sqrt(R_s/R_L) ; // Transformation ratio of the matching transformer for MPT

// case b
V_1 = V / 2 ; // Terminal voltage in volt of the source at MPT

// case c
V_2 = V_1 / alpha ; // Terminal voltage in volt across the load at MPT

// case d
```

```
I_2 = V_2 / Z_L ; // Secondary load current in A (
    method 1)
I2 = V / (2*alpha*R_L) ; // Secondary load current
    in A (method 2)

// case e
I_1 = I_2 / alpha ; // Primary load current drawn
    from the source in A (method 1)
I1 = V / (2*R_s) ; // Primary load current drawn
    from the source in A (method 2)

// case f
P_L = (I_2)^2 * R_L ; // Maximum power dissipated in
     the load in W

// case g
P_s = (I_1)^2 * R_s ; // Power dissipated internally
     within the source in W

// case h
P_T1 = V * I_1 * cosd(0) ; // Total power supplied
    by the source in W(method 1)

P_T2 = P_L + P_s ; // Total power supplied by the
    source in W(method 2)

// case i
P_T = P_T1 ;
eta = P_L / P_T * 100 ; // Power transfer efficiency
     in percent

// Display the results
disp("Example 14-7 Solution : ");

printf(" \n a: Transformation ratio of the matching
    transformer for MPT : ");
printf(" \n        = %d \n ",alpha );

```

```
printf(" \n b: Terminal voltage of the source at MPT :\n     V_1 = %d V \n",V_1);

printf(" \n c: Terminal voltage across the load at MPT :\n     V_2 = %.1f V \n",V_2);

printf(" \n d: Secondary load current :");
printf(" \n     (method 1) :\n     I_2 = %.2f A = %d mA \n ",I_2, 1000*I_2);

printf(" \n     (method 2) :\n     I_2 = %.2f A = %d mA \n ",I2, 1000*I2);

printf(" \n e: Primary load current drawn from the source : ");
printf(" \n     (method 1) :\n     I_1 = %f A = %d mA \n ",I_1 , 1000*I_1 );
printf(" \n     (method 2) :\n     I_1 = %f A = %d mA \n ",I1 , 1000*I1 );

printf(" \n f: Maximum power dissipated in the load : ");
printf(" \n     P_L = %f W = %d mW \n",P_L , 1000*P_L );

printf(" \n g: Power dissipated internally within the source : " );
printf(" \n     P_s = %f W = %d mW \n",P_s , 1000*P_s );

printf(" \n h: Total power supplied by the source : ");
printf(" \n     (method 1) :\n     P_T = %f W = %d mW \n ",P_T1, 1000*P_T1);
printf(" \n     (method 2) :\n     P_T = %f W = %d mW \n ",P_T2, 1000*P_T2);

printf(" \n i: Power transfer efficiency :\n        =
```

```
%d percent ",eta );
```

Scilab code Exa 14.8 PL alpha maxPL

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-8

clear; clc; close; // Clear the work space and console.

// Given data
// power transformer
V = 20 ; // No-load voltage in volt
R_s = 18 ; // Internal resistance of the power amplifier in ohm
R_L = 8 ; // Load resistance in ohm(Speaker)

// Calculations
// case a
V_L = ( R_L / (R_L + R_s) )* V; // Load voltage in volt
P_L = (V_L)^2 / R_L ; // Power delivered in W to the speaker when connected
// directly to the amplifier

// case b
alpha = sqrt(R_s/R_L); // Turns ratio of the transformer to maximize speaker power

// case c
```

```
V_2 = V / (2*alpha); // Secondary voltage in volt
P_L2 = (V_2)^2 / R_L ; // Maximum power delivered in W to the speaker using matching
// transformer of part b

// Display the results
disp("Example 14-8 Solution : ");

printf(" \n a; Load voltage :\n    V_L = %.2f V across the 8    speaker\n ",V_L);
printf(" \n    Power delivered in W to the speaker when connected directly to the amplifier : ");
printf(" \n    P_L = %.2f W \n ", P_L );

printf(" \n b: Turns ratio of the transformer to maximize speaker power : ");
printf(" \n       = %.1f \n ", alpha );

printf(" \n c: Secondary voltage :\n    V_2 = %f V \n ",V_2 );
printf(" \n    Maximum power delivered in W to the speaker using matching ");
printf(" \n    transformer of part b :\n    P_L = %f W ",P_L2 );
```

Scilab code Exa 14.9 Eh El Ih new kVA

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-9

```

```
clear; clc; close; // Clear the work space and console.

// Given data
kVA = 1 ; // kVA rating of the transformer
V_1 = 220 ; // Primary voltage in volt
V_2 = 110 ; // Secondary voltage in volt
f_o = 400 ; // Frequency in Hz(original frequency)
f_f = 60 ; // Frequency in Hz for which the transformer is to be used

// Calculations
alpha = V_1 / V_2 ; // Transformation ratio
// case a
E_h = V_1 * (f_f / f_o); // Maximum rms voltage in volt applied to HV side
E_1 = E_h ;
E_l = E_1 / alpha ; // Maximum rms voltage in volt applied to HV side

// case b
V_h = V_1 ; // High voltage in volt
I_h = kVA * 1000 / V_h ;
Vh = E_h ;
kVA_new = Vh * I_h ;

// Display the results
disp("Example 14-9 Solution : ");

printf(" \n a: To maintain the same permissible flux density in Eqs.(14-15)");
printf(" \n    and (14-16),both voltages of the high and low sides must change ");
printf(" \n    in the same proportion as the frequency :  ");
printf(" \n    E_h = %d V \n    and,\n    E_l = %.1f V\n",E_h , E_l );

```

```
printf(" \n b: The original current rating of the transformer is unchanged since");
printf(" \n    the conductors still have the same current carrying capacity.");
printf(" \n    Thus,\n    I_h = %.3f A\n    and the new kVA rating is",I_h );
printf(" \n    V_h*I_h = V_1*I_1 = %d VA = %.2f kVA",kVA_new , kVA_new/1000);
```

Scilab code Exa 14.10 Piron

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-10

clear; clc; close; // Clear the work space and console.

// Given data(from Example 14-9)
kVA = 1 ; // kVA rating of the transformer
V_1 = 220 ; // Primary voltage in volt
V_2 = 110 ; // Secondary voltage in volt
f_o = 400 ; // Frequency in Hz
f_f = 60 ; // Frequency in Hz for which the transformer is to be used
P_orig = 10 ; // Original iron losses of the transformer in W

// Calculations
// consider only ratio of frequencies for calculating B
```

```
B = f_o / f_f ; // flux density

P_iron = (P_orig)*(B^2); // Iron losses in W

// Display the results
disp("Example 14-10 Solution : ");

printf(" \n Since E = k*f*B_m and the same primary
   voltage is applied to the ");
printf(" \n transformer at reduced frequency, the
   final flux density B_mf ");
printf(" \n increased significantly above its
   original maximum permissible ");
printf(" \n value B_mo to :\n B_mf = B_mo * (f_o/f_f
   ) = %.2fB_mo \n ",B );

printf(" \n Since the iron losses vary approximately
    as the square of the flux density :");
printf(" \n P_iron = %d W ",P_iron );
```

Scilab code Exa 14.11 I2 I1 Z2 Z1their loss E2 E1 alpha

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-11

clear; clc; close; // Clear the work space and
   console.

// Given data
kVA = 500 ; // kVA rating of the step-down
```

```
transformer
V_1 = 2300 ; // Primary voltage in volt
V_2 = 230 ; // Secondary voltage in volt
f = 60 ; // Frequency in Hz
r_1 = 0.1 ; // Primary winding resistance in ohm
x_1 = 0.3 ; // Primary winding reactance in ohm
r_2 = 0.001 ; // Secondary winding resistance in ohm
x_2 = 0.003 ; // Secondary winding reactance in ohm

// Calculations
alpha = V_1 / V_2 ; // Transformation ratio
// case a
I_2 = (kVA*1000) / V_2 ; // Secondary current in A
I_1 = I_2 / alpha ; // Primary current in A

// case b
Z_2 = r_2 + %i*(x_2); // Secondary internal impedance in ohm
Z_2_m = abs(Z_2);//Z_2_m=magnitude of Z_2 in ohm
Z_2_a = atan(imag(Z_2) /real(Z_2))*180/%pi;//Z_2_a=phase angle of Z_2 in degrees

Z_1 = r_1 + %i*(x_1); // Primary internal impedance in ohm
Z_1_m = abs(Z_1);//Z_1_m=magnitude of Z_1 in ohm
Z_1_a = atan(imag(Z_1) /real(Z_1))*180/%pi;//Z_1_a=phase angle of Z_1 in degrees

// case c
I_2_Z_2 = I_2 * Z_2_m ; // Secondary internal voltage drop in volt
I_1_Z_1 = I_1 * Z_1_m ; // Primary internal voltage drop in volt

// case d
E_2 = V_2 + I_2_Z_2 ; // Secondary induced voltage in volt
E_1 = V_1 - I_1_Z_1 ; // Primary induced voltage in
```

```
volt

// case e
ratio_E = E_1 / E_2 ; // ratio of primary to secondary induced voltage
ratio_V = V_1 / V_2 ; // ratio of primary to secondary terminal voltage

// Display the results
disp("Example 14-11 Solution : ");

printf(" \n a: Secondary current :\n    I_2 = %.f A \n ",I_2 );
printf(" \n    Primary current :\n    I_1 = %.1f A \n ",I_1 );

printf(" \n b: Secondary internal impedance : \n Z_2 in ohm = ");disp(Z_2);
printf(" \n    Z_2 = %f <%.2f ohm \n ",Z_2_m , Z_2_a );
printf(" \n    Primary internal impedance : \n Z_1 in ohm = ");disp(Z_1);
printf(" \n    Z_1 = %f <%.2f ohm \n ",Z_1_m , Z_1_a );

printf(" \n c: Secondary internal voltage drop :\n    I_2*Z_2 = %.2f V \n ",I_2_Z_2);
printf(" \n    Primary internal voltage drop :\n I_1*Z_1 = %.2f V \n ",I_1_Z_1);

printf(" \n d: Secondary induced voltage :\n    E_2 = %.2f V \n",E_2 );
printf(" \n    Primary induced voltage :\n    E_1 = %.2f V \n",E_1 );

printf(" \n e: Ratio of E_1/E_2 = %.2f =    = N_1/N_2 \n",ratio_E );
printf(" \n    But V_1/V_2 = %d ",ratio_V );
```

Scilab code Exa 14.12 ZL ZP difference

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-12

clear; clc; close; // Clear the work space and
   console.

// Given data(from Example 14-11)
V_1 = 2300 ; // Primary voltage in volt
V_2 = 230 ; // Secondary voltage in volt
I_2 = 2174 ; // Secondary current in A
I_1 = 217.4 ; // Primary current in A
// calculated values from Example 14-11
Z_2 = 0.00316 ; // Secondary internal impedance in
   ohm
Z_1 = 0.316 ; // Primary internal impedance in ohm


// Calculations
alpha = V_1 / V_2 ; // Transformation ratio
// case a
Z_L = V_2 / I_2 ; // Load impedance in ohm

// case b
Z_p = V_1 / I_1 ; // Primary input impedance in ohm

Zp = (alpha)^2 * Z_L ; // Primary input impedance in
   ohm
```

```

// Display the results
disp("Example 14-12 Solution : ");

printf(" \n a: Load impedance :\n    Z_L = %.4f ohm \n ", Z_L );

printf(" \n b: Primary input impedance : ");
printf(" \n    (method 1) :\n    Z_p = %.2f ohm \n ",Z_p );
printf(" \n    (method 2) :\n    Z_p = %.2f ohm \n ",Zp );

printf(" \n c: The impedance of the load Z_L = %.4f  , which is much greater",Z_L);
printf(" \n    than the internal secondary impedance Z_2 = %.5f    .\n ",Z_2);
printf(" \n    The primary input impedance Z_p = %.2f   ,which is much greater",Z_p);
printf(" \n    than the internal primary impedance Z_1 = %.3f    .\n",Z_1);

printf(" \n d: It is essential for Z_L to be much greater than Z_2 so that the ");
printf(" \n    major part of the voltage produced by E_2 is dropped across the ");
printf(" \n    load impedance Z_L. As Z_L is reduced in proportion to Z_2, the ");
printf(" \n    load current increases and more voltage is dropped internally ");
printf(" \n    across Z_2.");
```

Scilab code Exa 14.13 Re1 Xe1 Ze1 ZLprime I1

```
// Electric Machinery and Transformers
```

```
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-13

clear; clc; close; // Clear the work space and
    console.

// Given data
kVA = 500 ; // kVA rating of the step-down
    transformer
V_1 = 2300 ; // Primary voltage in volt
V_2 = 230 ; // Secondary voltage in volt
f = 60 ; // Frequency in Hz
r_1 = 0.1 ; // Primary winding resistance in ohm
x_1 = 0.3 ; // Primary winding reactance in ohm
r_2 = 0.001 ; // Secondary winding resistance in ohm
x_2 = 0.003 ; // Secondary winding reactance in ohm
// calculated data from Example 14-12
Z_L = 0.1058 ; // Load impedance in ohm

// Calculations
alpha = V_1 / V_2 ; // Transformation ratio

// case a
R_e1 = r_1 + (alpha)^2 * r_2 ; // Equivalent
    internal resistance referred to the
// primary side in ohm

// case b
X_e1 = x_1 + (alpha)^2 * x_2 ; // Equivalent
    internal reactance referred to the
// primary side in ohm

// case c
Z_e1 = R_e1 + %i*(X_e1) ; // Equivalent internal
```

```
    impedance referred to the
// primary side in ohm
Z_e1_m = abs(Z_e1);//Z_e1_m=magnitude of Z_e1 in ohm
Z_e1_a = atan(imag(Z_e1) /real(Z_e1))*180/%pi;//
    Z_e1_a=phase angle of Z_e1 in degrees

// case d
Z_L_prime = (alpha)^2 * (Z_L); // Equivalent
    secondary load impedance referred
//  to the primary side in ohm

// case e
R_L = Z_L ; // Load resistance in ohm
X_L = 0 ; // Load reactance in ohm

// Primary load current in A , when V_1 = 2300 V
I_1 = V_1 / ( (R_e1 + alpha^2*R_L) + %i*(X_e1 +
    alpha^2*X_L) );

// Display the results
disp("Example 14-13 Solution : ");

printf(" \n a: Equivalent internal resistance
    referred to the primary side :");
printf(" \n    R_e1 = %.2f ohm \n ",R_e1 );

printf(" \n b: Equivalent internal reactance
    referred to the primary side :");
printf(" \n    X_e1 = %.2f ohm \n ",X_e1 );

printf(" \n c: Equivalent internal impedance
    referred to the primary side : ");
printf(" \n    Z_e1 in ohm = ");disp(Z_e1);
printf(" \n    Z_e1 = %.3f <%.2f ohm \n ", Z_e1_m ,
    Z_e1_a );

printf(" \n d: Equivalent secondary load impedance
    referred to the primary side :");
```

```
printf(" \n     (alpha)^2 * Z_L = %.2f ohm = (alpha)^2 * R_L \n",Z_L_prime);

printf(" \n e: Primary load current :\n     I_1 = %f A     %.f A ", I_1, I_1);
```

Scilab code Exa 14.14 I2 ohmdrops E2 VR

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-14

clear; clc; close; // Clear the work space and console.

// Given data
kVA = 500 ; // kVA rating of the step-down transformer
V_1 = 2300 ; // Primary voltage in volt
V_2 = 230 ; // Secondary voltage in volt
R_e2 = 2 ; // Equivalent resistance referred to the
// primary side in m
X_e2 = 6 ; // Equivalent reactance referred to the
// primary side in m

// Calculations
// case a
I_2 = (kVA ) / V_2 ; // Rated secondary current in kA

// case b
```

```
R_e2_drop = I_2 * R_e2 ; // Full-load equivalent resistance voltage drop in volt

// case c
X_e2_drop = I_2 * X_e2 ; // Full-load equivalent reactance voltage drop in volt

// case d
// unity PF
cos_theta2 = 1;
sin_theta2 = sqrt(1 - (cos_theta2)^2);

// Induced voltage when the transformer is delivering rated current to unity PF load
E_2 = (V_2*cos_theta2 + I_2*R_e2) + %i*(V_2*sin_theta2 + I_2*X_e2);
E_2_m = abs(E_2);//E_2_m=magnitude of E_2 in volt
E_2_a = atan(imag(E_2) /real(E_2))*180/%pi;//E_2_a=phase angle of E_2 in degrees

// case e
VR = ( (E_2_m - V_2) / V_2 ) * 100 ; // Percent voltage regulation at unity PF

// Display the results
disp("Example 14-14 Solution : ");

printf(" \n a: Rated secondary current :\n    I_2 = %.3f kA \n ", I_2 );

printf(" \n b: Full-load equivalent resistance voltage drop : ");
printf(" \n    I_2*R_c2 = %.2f V \n", R_e2_drop );

printf(" \n c: Full-load equivalent reactance voltage drop : ");
printf(" \n    I_2*X_c2 = %.2f V \n", X_e2_drop );

```

```
printf(" \n d: Induced voltage when the transformer is delivering rated current ");
printf(" \n to unity PF load :\n E_2 in volt = ");disp(E_2);
printf(" \n E_2 = %.2f <%.2f V \n ",E_2_m , E_2_a);

printf(" \n e: Voltage regulation at unity PF :\n VR = %.2f percent ",VR );
```

Scilab code Exa 14.15 E2 VR

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-15

clear; clc; close; // Clear the work space and console.

// Given data
kVA = 500 ; // kVA rating of the step-down transformer
V_1 = 2300 ; // Primary voltage in volt
V_2 = 230 ; // Secondary voltage in volt
R_e2 = 2 ; // Equivalent resistance referred to the
// primary side in m
X_e2 = 6 ; // Equivalent reactance referred to the
// primary side in m
I_2 = 2.174 ; //  Rated secondary current in kA

cos_theta2 = 0.8 ; // lagging PF
```

```
sin_theta2 = sqrt(1 - (cos_theta2)^2);

// Calculations

// case d
// Induced voltage when the transformer is
   delivering rated current to 0.8 lagging PF load
E_2 = (V_2*cos_theta2 + I_2*R_e2) + %i*(V_2*
   sin_theta2 + I_2*X_e2);
E_2_m = abs(E_2);//E_2_m=magnitude of E_2 in volt
E_2_a = atan(imag(E_2) /real(E_2))*180/%pi;//E_2_a=
   phase angle of E_2 in degrees

// case e
VR = ( (E_2_m - V_2) / V_2 ) * 100 ; // Percent
   voltage regulation at 0.8 PF lag

// Display the results
disp("Example 14-15 Solution : ");

printf(" \n d: Induced voltage when the transformer
   is delivering rated current ");
printf(" \n    to 0.8 lagging PF load :\n    E_2 in
   volt = ");disp(E_2);
printf(" \n    E_2 = %.2f <%.2f V \n ",E_2_m , E_2_a
   );

printf(" \n e: Voltage regulation at 0.8 lagging PF
   :\n    VR = %.2f percent ",VR );
```

Scilab code Exa 14.16 E2 VR

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
```

```
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-16

clear; clc; close; // Clear the work space and console.

// Given data
kVA = 500 ; // kVA rating of the step-down transformer
V_1 = 2300 ; // Primary voltage in volt
V_2 = 230 ; // Secondary voltage in volt
R_e2 = 2 ; // Equivalent resistance referred to the
// primary side in m
X_e2 = 6 ; // Equivalent reactance referred to the
// primary side in m
I_2 = 2.174 ; //  Rated secondary current in kA

cos_theta2 = 0.6 ; // leading PF
sin_theta2 = sqrt(1 - (cos_theta2)^2);

// Calculations

// case d
// Induced voltage when the transformer is delivering rated current to unity PF load
E_2 = (V_2*cos_theta2 + I_2*R_e2) + %i*(V_2*
    sin_theta2 - I_2*X_e2);
E_2_m = abs(E_2);//E_2_m=magnitude of E_2 in volt
E_2_a = atan(imag(E_2) /real(E_2))*180/%pi;//E_2_a=phase angle of E_2 in degrees

// case e
VR = ( (E_2_m - V_2) / V_2 ) * 100 ; // Percent voltage regulation at 0.8 leading PF

// Display the results
```

```
disp("Example 14-16 Solution : ");

printf(" \n d: Induced voltage when the transformer is delivering rated current ");
printf(" \n    to 0.6 leading PF load :\n    E_2 in volt = ");disp(E_2);
printf(" \n    E_2 = %.2f <%.2f V \n ",E_2_m , E_2_a);

printf(" \n e: Voltage regulation at 0.8 leading PF :\n    VR = %.2f percent ",VR );
```

Scilab code Exa 14.17 Ze1 Re1 Xe1 Ze2 Re2 Xe2their drops VR

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-17

clear; clc; close; // Clear the work space and console.

// Given data
kVA = 20 ; // kVA rating of the step-down transformer
S = 20000 ; // power rating of the step-down transformer in VA
V_1 = 2300 ; // Primary voltage in volt
V_2 = 230 ; // Secondary voltage in volt

// w.r.t HV side following is SC-test data
P1 = 250 ; // wattmeter reading in W
```

```
I1 = 8.7 ; // Input current in A
V1 = 50 ; // Input voltage in V
// Calculations
alpha = V_1 / V_2 ; // Transformation ratio
// case a
Z_e1 = V1 / I1 ; // Equivalent impedance w.r.t HV side in ohm
R_e1 = P1 / (I1)^2 ; // Equivalent resistance w.r.t HV side in ohm
theta = acosd(R_e1/Z_e1) ; // PF angle in degrees
X_e1 = Z_e1*sind(theta); // Equivalent reactance w.r.t HV side in ohm
// case b
Z_e2 = Z_e1 / (alpha)^2 ; // Equivalent impedance w.r.t LV side in ohm
R_e2 = R_e1 / (alpha)^2 ; // Equivalent resistance w.r.t LV side in ohm
X_e2 = Z_e2*sind(theta); // Equivalent reactance w.r.t LV side in ohm
// case c
I_2 = S / V_2 ; // Rated secondary load current in A
R_e2_drop = I_2 * R_e2 ; // Full-load equivalent resistance voltage drop in volt
X_e2_drop = I_2 * X_e2 ; // Full-load equivalent reactance voltage drop in volt
// At unity PF
cos_theta2 = 1;
sin_theta2 = sqrt(1 - (cos_theta2)^2);
```

```

// Induced voltage when the transformer is delivering rated current to unity PF load
E_2 = (V_2*cos_theta2 + I_2*R_e2) + %i*(V_2*sin_theta2 + I_2*X_e2);
E_2_m = abs(E_2);//E_2_m=magnitude of E_2 in volt
E_2_a = atan(imag(E_2) /real(E_2))*180/%pi;//E_2_a=phase angle of E_2 in degrees

VR_unity_PF = ( (E_2_m - V_2) / V_2 ) * 100 ; // Transformer voltage regulation

// case d
// at 0.7 lagging PF
cos_theta_2 = 0.7 ; // lagging PF
sin_theta_2 = sqrt(1 - (cos_theta_2)^2);

// Induced voltage when the transformer is delivering rated current to unity PF load
E2 = (V_2*cos_theta_2 + I_2*R_e2) + %i*(V_2*sin_theta_2 + I_2*X_e2);
E2_m = abs(E2);//E2_m=magnitude of E2 in volt
E2_a = atan(imag(E2) /real(E2))*180/%pi;//E2_a=phase angle of E2 in degrees

VR_lag_PF = ( (E2_m - V_2) / V_2 ) * 100 ; // Transformer voltage regulation

// Display the results
disp("Example 14-17 Solution : ");

printf(" \n a: Equivalent impedance w.r.t HV side :\n    Z_e1 = %.2f    \n",Z_e1);
printf(" \n    Equivalent resistance w.r.t HV side :\n    R_e1 = %.1f    \n",R_e1);
printf(" \n        = %.f degrees \n",theta );
printf(" \n    Equivalent reactance w.r.t HV side :\n    X_e1 = %.2f \n",X_e1);
```

```

printf(" \n b: Equivalent impedance w.r.t LV side :");
printf(" \n    Z_e2 = %f    = %.2f m   \n",Z_e2 , Z_e2*1000);
printf(" \n    Equivalent resistance w.r.t LV side :\n    R_e2 = %f    \n",R_e2);
printf(" \n    R_e2 = %f    = %.2f m   \n",R_e2,R_e2*1000);
printf(" \n    Equivalent reactance w.r.t LV side :\n    X_e2 = %f    \n",X_e2);
printf(" \n    X_e2 = %f    = %.2f m   \n",X_e2,X_e2*1000);

printf(" \n c: Rated secondary load current :\n    I_2 = %.f A\n",I_2);
printf(" \n    I_2*R_c2 = %.2f V \n", R_e2_drop );
printf(" \n    I_2*X_c2 = %.2f V \n", X_e2_drop );
printf(" \n    At unity PF,\n    E_2 in volt = ");disp(E_2);
printf(" \n    E_2 = %.2f <%.2f V \n ",E_2_m , E_2_a );
printf(" \n    Voltage regulation at unity PF :\n    VR = %.2f percent ",VR_unity_PF );

printf(" \n\n d: At 0.7 lagging PF, \n    E_2 in volt = ");disp(E2);
printf(" \n    E_2 = %.2f <%.2f V \n ",E2_m , E2_a);
printf(" \n    Voltage regulation at 0.7 lagging PF :\n    VR = %.2f percent ",VR_lag_PF );
```

Scilab code Exa 14.18 Pcsc

```
// Electric Machinery and Transformers
// Irving L kosow
```

```
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-18

clear; clc; close; // Clear the work space and
    console.

// Given data
V_sc = 50 ; // Short circuit voltage in volt
V_1 = 2300 ; // Rated primary voltage in volt

// Calculations
P_c = poly(0,'P_c');// Making P_c as a variable just
     for displaying answer as per
// textbook

P_c_sc = (V_sc / V_1)^2 * P_c ; // Fraction of P_c
    measured by the wattmeter

// Display the results
disp("Example 14-18 Solution : ");

printf(" \n Since P_c is proportional to the square
    of the primary voltage V_sc, ");
printf(" \n then under short circuit conditions,the
    fraction of rated-core loss is :");
printf(" \n P_c(sc) = ");disp(P_c_sc);
```

Scilab code Exa 14.19 Ze1drop Re1drop Xe1drop VR

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
```

```
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-19

clear; clc; close; // Clear the work space and console.

// Given data

kVA = 20 ; // kVA rating of the step-down transformer
S = 20000 ; // power rating of the step-down transformer in VA
V_1 = 2300 ; // Primary voltage in volt
V_2 = 230 ; // Secondary voltage in volt
Z_e1 = 5.75 ; // Equivalent impedance w.r.t HV side in ohm
R_e1 = 3.3 ; // Equivalent resistance w.r.t HV side in ohm
X_e1 = 4.71 ; // Equivalent reactance w.r.t HV side in ohm

// w.r.t HV side following is SC-test data
P1 = 250 ; // wattmeter reading in W
I1 = 8.7 ; // Input current in A
V1 = 50 ; // Input voltage in V

// Calculations
// case a
Z_e1_drop = V1 ; // High voltage impedance drop in volt

// case b
theta = acosd(R_e1/Z_e1) ; // PF angle in degrees

R_e1_drop = I1*Z_e1*cosd(theta) ; //HV-side equivalent resistance voltage drop in volt
```

```

// case c
X_e1_drop = I1*Z_e1*sind(theta) ; //HV-side equivalent reactance voltage drop in volt

// case d
// At unity PF
cos_theta1 = 1;
sin_theta1 = sqrt(1 - (cos_theta1)^2);

// Induced voltage when the transformer is delivering rated current to unity PF load
E_1 = (V_1*cos_theta1 + I1*R_e1) + %i*(V_1*sin_theta1 + I1*X_e1);
E_1_m = abs(E_1);//E_1_m=magnitude of E_1 in volt
E_1_a = atan(imag(E_1) /real(E_1))*180/%pi;//E_1_a=phase angle of E_1 in degrees

VR_unity_PF = ( (E_1_m - V_1) / V_1 ) * 100 ; //Transformer voltage regulation

// case e
// at 0.7 lagging PF
cos_theta_1 = 0.7 ; // lagging PF
sin_theta_1 = sqrt(1 - (cos_theta_1)^2);

// Induced voltage when the transformer is delivering rated current to unity PF load
E1 = (V_1*cos_theta_1 + I1*R_e1) + %i*(V_1*sin_theta_1 + I1*X_e1);
E1_m = abs(E1);//E1_m=magnitude of E1 in volt
E1_a = atan(imag(E1) /real(E1))*180/%pi;//E1_a=phase angle of E1 in degrees

VR_lag_PF = ( (E1_m - V_1) / V_1 ) * 100 ; //Transformer voltage regulation

// Display the results
```

```
disp("Example 14-19 Solution : ");

printf(" \n a: High voltage impedance drop :\n I_1*Z_e1 = V_1 = %d\n",Z_e1_drop);

printf(" \n b:     = %.f degrees \n",theta );
printf(" \n     High voltage resistance drop :\n I_1*R_e1 = %.2f \n",R_e1_drop);

printf(" \n c: High voltage reactance drop :\n I_1*X_e1 = %.2f \n",X_e1_drop);

printf(" \n d: At unity PF,\n     E_2 in volt = ");disp(E_1);
printf(" \n     E_2 = %.2f <%.2f V \n ",E_1_m , E_1_a);
printf(" \n     Voltage regulation at unity PF :\n     VR = %.2f percent ",VR_unity_PF );

printf(" \n\n e: At 0.7 lagging PF, \n     E_2 in volt = ");disp(E1);
printf(" \n     E_2 = %.2f <%.2f V \n ",E1_m , E1_a);
printf(" \n     Voltage regulation at 0.7 lagging PF :\n     VR = %.2f percent ",VR_lag_PF );
```

Scilab code Exa 14.20 Re1 Re1 r2 its drop Pc

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-20

```

```
clear; clc; close; // Clear the work space and console.

// Given data
kVA = 500 ; // kVA rating of the step-down transformer
V_1 = 2300 ; // Primary voltage in volt
V_2 = 208 ; // Secondary voltage in volt
f = 60 ; // Frequency in Hz

// SC-test data
P_sc = 8200 ; // wattmeter reading in W
I_sc = 217.4 ; // Short circuit current in A
V_sc = 95 ; // Short circuit voltage in V

// OC-test data
P_oc = 1800 ; // wattmeter reading in W
I_oc = 85 ; // Open circuit current in A
V_oc = 208 ; // Open circuit voltage in V

// Calculations
alpha = V_1 / V_2 ; // Transformation ratio
// case a
P = P_sc ; // wattmeter reading in W
I1 = I_sc ; // Short circuit current in A
R_e1 = P / (I1)^2 ; // Equivalent resistance w.r.t HV side in ohm
R_e2 = R_e1 / (alpha)^2 // Equivalent resistance referred to LV side in ohm

// case b
r_2 = R_e2 / 2 ; // Resistance of low-voltage side in ohm

// case c
I_m = I_oc ; // Open circuit current in A
P_cu = (I_m)^2 * r_2 ; // Transformer copper loss of the LV side wdg during OC-test in W
```

```

// case d
P_c = P_oc - P_cu ; // Transformer core loss in W

// Display the results
disp("Example 14-20 Solution : ");

printf(" \n a: Equivalent resistance w.r.t HV side :\n    R_e1 = %.4f   \n",R_e1);
printf(" \n    Equivalent resistance w.r.t LV side :\n    R_e2 = %f    = %.3f  m   \n",R_e2,R_e2*1000);

printf(" \n b: Resistance of LV side :\n    r_2 = %f  = %.2f  m \n",r_2,r_2*1000);

printf(" \n c: Transformer copper loss of the LV side wdg during OC-test : ");
printf(" \n    (I_m)^2 * r_2 = %f W \n",P_cu);

printf(" \n d: Transformer core loss :\n    P_c = %f W \n ",P_c);

printf(" \n e: Yes.The error is approximately 5/%d = 0.278 percent ,which is",P_oc);
printf(" \n    within the error produced by the instruments used in the test.");
printf(" \n    We may assume that the core loss is %d W.",P_oc);
```

Scilab code Exa 14.21 tabulate I2 efficiencies

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
```

```
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-21

clear; clc; close; // Clear the work space and
   console.

// Given data(from Ex.14-18)
V_sc = 50 ; // Short circuit voltage in volt
V_1 = 2300 ; // Rated primary voltage in volt


// Preliminary data before tabulating

// from ex.14-20
P_c = 1.8 ; // core losses in kW
P_k = 1.8 ; // fixed losses in kW
P_cu_rated = 8.2 ; // Rated copper loss in kW

// given rating
kVA = 500 ; // Power rating in kVA
PF = 1 ; // power factor
P_o = kVA * PF ; // full-load output at unity PF in
   kW

// Calculations
// case a
LF1 = 1/4 ; // Load fraction
LF2 = 1/2 ; // Load fraction
LF3 = 3/4 ; // Load fraction
LF4 = 5/4 ; // Load fraction
P_cu_fl = 8.2 ; // Equivalent copper loss at full-
   load slip
P_cu_LF1 = (LF1)^2 * P_cu_fl ; // Equivalent copper
   loss at 1/4 rated load
P_cu_LF2 = (LF2)^2 * P_cu_fl ; // Equivalent copper
   loss at 1/2 rated load
```

```
P_cu_LF3 = (LF3)^2 * P_cu_fl ; // Equivalent copper loss at 3/4 rated load
P_cu_LF4 = (LF4)^2 * P_cu_fl ; // Equivalent copper loss at 5/4 rated load

P_L_1 = P_c + P_cu_LF1 ; // Total losses in kW at 1/4 rated load
P_L_2 = P_c + P_cu_LF2 ; // Total losses in kW at 1/2 rated load
P_L_3 = P_c + P_cu_LF3 ; // Total losses in kW at 3/4 rated load
P_L_fl = P_c + P_cu_fl ; // Total losses in kW at rated load
P_L_4 = P_c + P_cu_LF4 ; // Total losses in kW at 5/4 rated load

P_o_1 = P_o*LF1 ; // Total output in kW at 1/4 rated load
P_o_2 = P_o*LF2 ; // Total output in kW at 1/2 rated load
P_o_3 = P_o*LF3 ; // Total output in kW at 3/4 rated load
P_o_fl = P_o ; // Total output in kW at rated load
P_o_4 = P_o*LF4 ; // Total output in kW at 5/4 rated load

P_in_1 = P_L_1 + P_o_1 ; // Total input in kW at 1/4 rated load
P_in_2 = P_L_2 + P_o_2 ; // Total input in kW at 1/2 rated load
P_in_3 = P_L_3 + P_o_3 ; // Total input in kW at 3/4 rated load
P_in_fl = P_L_fl + P_o_fl ; // Total input in kW at rated load
P_in_4 = P_L_4 + P_o_4 ; // Total input in kW at 5/4 rated load

eta_1 = (P_o_1/P_in_1)*100 ; // Efficiency at 1/4
```

```
rated load
eta_2 = (P_o_2/P_in_2)*100 ; // Efficiency at 1/2 rated load
eta_3 = (P_o_3/P_in_3)*100 ; // Efficiency at 3/4 rated load
eta_fl = (P_o_fl/P_in_fl)*100 ; // Efficiency at rated load
eta_4 = (P_o_4/P_in_4)*100 ; // Efficiency at 5/4 rated load


// case b
PF_b = 0.8 ; // 0.8 PF lagging
Po_1 = P_o*LF1*PF_b ; // Total output in kW at 1/4 rated load
Po_2 = P_o*LF2*PF_b ; // Total output in kW at 1/2 rated load
Po_3 = P_o*LF3*PF_b ; // Total output in kW at 3/4 rated load
Po_fl = P_o*PF_b ; // Total output in kW at rated load
Po_4 = P_o*LF4*PF_b ; // Total output in kW at 5/4 rated load

Pin_1 = P_L_1 + Po_1 ; // Total input in kW at 1/4 rated load
Pin_2 = P_L_2 + Po_2 ; // Total input in kW at 1/2 rated load
Pin_3 = P_L_3 + Po_3 ; // Total input in kW at 3/4 rated load
Pin_fl = P_L_fl + Po_fl ; // Total input in kW at rated load
Pin_4 = P_L_4 + Po_4 ; // Total input in kW at 5/4 rated load

eta1 = (Po_1/Pin_1)*100 ; // Efficiency at 1/4 rated load
eta2 = (Po_2/Pin_2)*100 ; // Efficiency at 1/2 rated
```

```
load
eta3 = (Po_3/Pin_3)*100 ; // Efficiency at 3/4 rated load
etafl = (Po_fl/Pin_fl)*100 ; // Efficiency at rated load
eta4 = (Po_4/Pin_4)*100 ; // Efficiency at 5/4 rated load

// case c
R_e2 = 1.417e-3 ; // Equivalent resistance in ohm referred to LV side
Pc = 1800 ; // Core losses in W
I_2 = sqrt(Pc/R_e2); // Load current in A for max. efficiency invariant of LF

// case d
V = 208 ; // Voltage rating in volt
I_2_rated = (kVA*1000) / V ; // Rated secondary current in A
LF_max = I_2 / I_2_rated ; // Load fraction for max. efficiency

// case e
// subscript e for eta_max indicates case e
cos_theta = 1;
V_2 = V ; // secondary voltage in volt
Pc = 1800 ; // core loss in W
// max.efficiency for unity PF
eta_max_e = (V_2*I_2*cos_theta) / ((V_2*I_2*cos_theta) + (Pc + I_2^2*R_e2)) * 100

// case f
// subscript f for eta_max indicates case e
cos_theta2 = 0.8;
// max.efficiency for 0.8 lagging PF
eta_max_f = (V_2*I_2*cos_theta2) / ((V_2*I_2*cos_theta2) + (Pc + I_2^2*R_e2)) * 100

```

```
// Display the results
disp("Example 14-21 Solution : ");

printf(" \n a: Tabulation at unity PF : ");
printf(" \n ---------------------------------------------------------------------------------------------------------------");
printf(" \n    L.F \t Core loss \t Copper loss \tTotal loss \t Total Output \t Total Input \t Efficiency");
printf(" \n         \t   (kW)     \t     (kW)      \t P_L (kW) \t     P_o(kW)     \t P_L+P_o(kW)\t  P_o/P_in(percent)");
printf(" \n ---------------------------------------------------------------------------------------------------------------");
printf(" \n    %.2f   \t   %.1f \t\t   %.3f         \t %.3f \t\t   %.1f \t    %.2f \t %.2f ",LF1,P_c,P_cu_LF1,P_L_1,P_o_1,P_in_1,eta_1);
printf(" \n    %.2f   \t   %.1f \t\t   %.3f         \t %.3f \t\t   %.1f \t    %.2f \t %.2f ",LF2,P_c,P_cu_LF2,P_L_2,P_o_2,P_in_2,eta_2);
printf(" \n    %.2f   \t   %.1f \t\t   %.3f         \t %.3f \t\t   %.1f \t    %.2f \t %.2f ",LF3,P_c,P_cu_LF3,P_L_3,P_o_3,P_in_3,eta_3);
printf(" \n     1   \t\t   %.1f \t\t   %.3f         \t %.3f \t   %.1f \t    %.2f \t %.2f ",P_c,P_cu_fl,P_L_fl,P_o_fl,P_in_fl,eta_fl);
printf(" \n    %.2f   \t   %.1f \t\t   %.3f   \t %.3f \t   %.1f \t    %.2f \t %.2f ",LF4,P_c,P_cu_LF4,P_L_4,P_o_4,P_in_4,eta_4);
printf(" \n ---------------------------------------------------------------------------------------------------------------\n\n");

printf(" \n b: Tabulation at 0.8 PF lagging : ");
printf(" \n ---------------------------------------------------------------------------------------------------------------
```

```
");
printf(" \n    L.F \t Core loss \t Copper loss \tTotal loss \t Total Output \t Total Input \t Efficiency");
printf(" \n         \t    (kW)     \t     (kW)      \t P_L (kW) \t    P_o(kW)    \t P_L+P_o(kW)\t  P_o/P_in(percent)");
printf(" \n -------------------------------------------------------------------------------------------------------");
printf(" \n    %.2f  \t   %.1f \t\t   %.3f        \t %.3f \t\t   %.1f \t    %.2f \t %.2f ",LF1,P_c,P_cu_LF1,P_L_1,Po_1,Pin_1,eta1);
printf(" \n    %.2f  \t   %.1f \t\t   %.3f        \t %.3f \t\t   %.1f \t    %.2f \t %.2f ",LF2,P_c,P_cu_LF2,P_L_2,Po_2,Pin_2,eta2);
printf(" \n    %.2f  \t   %.1f \t\t   %.3f        \t %.3f \t\t   %.1f \t    %.2f \t %.2f ",LF3,P_c,P_cu_LF3,P_L_3,Po_3,Pin_3,eta3);
printf(" \n    1  \t\t   %.1f \t\t   %.3f        \t %.3f \t   %.1f \t    %.2f \t %.2f ",P_c,P_cu_fl,P_L_fl,Po_fl,Pin_fl,etafl);
printf(" \n    %.2f  \t   %.1f \t\t   %.3f  \t %.3f \t   %.1f \t    %.2f \t %.2f ",LF4,P_c,P_cu_LF4,P_L_4,Po_4,Pin_4,eta4);
printf(" \n -------------------------------------------------------------------------------------------------------\n\n");

printf(" \n c: Load current at which max.efficiency occurs :\n    I_2 = %.1f A \n",I_2);

printf(" \n d: Rated load current :\n    I_2(rated) = %.1f A \n",I_2_rated);
printf(" \n    Load fraction for _max = %.3f( half rated load)\n ",LF_max);

printf(" \n e: Max.efficiency for unity PF :\n
```

```
    _max = %.2f percent \n",eta_max_e);

printf(" \n f: Max.efficiency for 0.8 lagging PF :\n
       _max = %.2f percent",eta_max_f);
```

Scilab code Exa 14.22 Zeqpu V1pu VR

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-22

clear; clc; close; // Clear the work space and
   console.

// Given data
V_1 = 2300 ; // Primary voltage in volt
V_2 = 230 ; // Secondary voltage in volt
P = 20 ; // Power rating of the transformer in kVA
// Short circuit test data
P_sc = 250 ; // Power measured in W
V_sc = 50 ; // Short circuit voltage in volt
I_sc = 8.7 ; // Short circuit current in A

// Calculations
// case a
V_1b = V_1 ; // base voltage in volt
Z_eq_pu = V_sc / V_1 ;

funcprot(0) ; // Use this to avoid the message "
   Warning : redefining function: beta "
                    .
```

```
beta = acosd(P_sc/(V_sc*I_sc)); // angle in degrees

Zeq_pu = Z_eq_pu*exp(%i*(beta)*(%pi/180));
Zeq_pu_m = abs(Zeq_pu);//Zeq_pu_m=magnitude of Zeq_pu in p.u
Zeq_pu_a = atan(imag(Zeq_pu) /real(Zeq_pu))*180/%pi; //Zeq_pu_a=phase angle of Zeq_pu in degrees

// case b
// at unity PF
V_1_pu = 1*exp(%i*(0)*(%pi/180)) + 1*exp(%i*(0)*(%pi/180))*Z_eq_pu*exp(%i*(beta)*(%pi/180));
// RHS is written in exponential complex form and (%pi/180) is radians to degrees conversion factor
V_1_pu_m = abs(V_1_pu);//V_1_pu_m=magnitude of V_1_pu in volt
V_1_pu_a = atan(imag(V_1_pu) /real(V_1_pu))*180/%pi; //V_1_pu_a=phase angle of V_1_pu in degrees

// case c
// at 0.7 PF lagging
theta = acosd(0.7); // Power factor angle in degrees
V1_pu = 1*exp(%i*(0)*(%pi/180)) + 1*exp(%i*(-theta)*(%pi/180))*Z_eq_pu*exp(%i*(beta)*(%pi/180));
V1_pu_m = abs(V1_pu);//V1_pu_m=magnitude of V1_pu in volt
V1_pu_a = atan(imag(V1_pu) /real(V1_pu))*180/%pi;//V1_pu_a=phase angle of V1_pu in degrees

// case d
VR_unity_PF = V_1_pu_m - 1 ; // voltage regulation at unity PF

// case e
VR_lag_PF = V1_pu_m - 1 ; // voltage regulation at 0.7 lagging PF

// Display the results
```

```
disp("Example 14-22 Solution : ");

printf(" \n a: Z_eq(pu) = %.5f p.u \n",Z_eq_pu);
printf(" \n        = %.f degrees \n",beta);
printf(" \n    Z_eq(pu) <   = ");disp(Zeq_pu);
printf(" \n    Z_eq(pu) <   = %.5f <%.f p.u \n ", Zeq_pu_m,Zeq_pu_a);

printf(" \n b: |V_1(pu)| = ");disp(V_1_pu);
printf(" \n    |V_1(pu)| = %.4f <%.2f V \n ", V_1_pu_m , V_1_pu_a );

printf(" \n c: |V_1(pu)| = ");disp(V1_pu);
printf(" \n    |V_1(pu)| = %.4f <%.2f V \n ",V1_pu_m , V1_pu_a );

printf(" \n d: Voltage regulation at unity PF :\n     VR = %f ",VR_unity_PF);
printf(" \n    VR = %.3f percent \n ",100*VR_unity_PF);

printf(" \n e: Voltage regulation at 0.7 lagging PF :\n    VR = %f ",VR_lag_PF);
printf(" \n    VR = %.2f percent \n ",100*VR_lag_PF);

printf(" \n f: VRs as found by p.u method are essentially the same as those found ");
printf(" \n    in Exs.14-17 and 14-19 using the same data, for the same transformer, ");
printf(" \n    but with much less effort.");
```

Scilab code Exa 14.23 Pcu LF efficiencies

```
// Electric Machinery and Transformers
```

```
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-23

clear; clc; close; // Clear the work space and
   console.

// Given data
V_1 = 2300 ; // Primary voltage in volt
V_2 = 230 ; // Secondary voltage in volt
S = 500 ; // Power rating of the transformer in kVA
f= 60 ; // Frequency in Hz

// Open circuit test data
V_oc = 208 ; // Open circuit voltage in volt
I_oc = 85 ; // Open circuit current in A
P_oc = 1800 ; // Power measured in W

// Short circuit test data
V_sc = 95 ; // Short circuit voltage in volt
I_sc = 217.4 ; // Short circuit current in A
P_sc = 8200 ; // Power measured in W

// Calculations
// case a
S_b = S ; // Base voltage in kVA
Psc = 8.2 ; // Power measured in kW during SC-test
P_Cu_pu = Psc / S_b ; // per unit value of P_Cu at
   rated load

// case b
Poc = 1.8 ; // Power measured in kW during OC-test
P_CL_pu = Poc / S_b ; // per unit value of P_CL at
   rated load

```

```
// case c
PF = 1 ; // unity Power factor
eta_pu = PF / (PF + P_CL_pu + P_Cu_pu ) * 100 ; // Efficiency at rated load,unity PF

// case d
// subscript d for PF and eta_pu indicates case d
PF_d = 0.8 ; // 0.8 lagging Power factor
eta_pu_d = PF_d / (PF_d + P_CL_pu + P_Cu_pu ) * 100 ; // Efficiency at rated load,unity PF

// case e
LF = sqrt(P_CL_pu / P_Cu_pu); // Load fraction producing max.efficiency

// case f
eta_pu_max = (LF*PF) / ( (LF*PF) + 2*(P_CL_pu) ) * 100 ; // Maximum efficiency at unity PF load

// case g
eta_pu_max_g = (LF*PF_d) / ( (LF*PF_d) + 2*(P_CL_pu) ) * 100 ; // Maximum efficiency at 0.8 lagging PF load


// Display the results
disp("Example 14-23 Solution : ");

printf(" \n a: Per unit copper loss at rated load :");
printf(" \n    P_Cu(pu) = %.4f p.u = R_eq(pu)\n",P_Cu_pu);

printf(" \n a: Per unit core loss at rated load :");
printf(" \n    P_CL(pu) = %.4f p.u \n",P_CL_pu);

printf(" \n c: Efficiency at rated load,unity PF :\n
        _pu = %.2f percent \n",eta_pu);
```

```
printf(" \n c: Efficiency at rated load,0.8 lagging PF :\n      _pu = %.2f percent \n",eta_pu_d);

printf(" \n e: Load fraction producing max. efficiency :\n    L.F = %.3f \n ",LF );

printf(" \n f: Maximum efficiency at unity PF load :\n      _pu(max) = %.2f percent \n",eta_pu_max);

printf(" \n g: Maximum efficiency at 0.8 lagging PF load :\n      _pu(max) = %.2f percent \n", eta_pu_max_g);

printf(" \n h: All efficiency values are identical to those computed in solution to Ex.14-21. \n");

printf(" \n i: Per-unit method is much simpler and less subject to error than conventional method.");
```

Scilab code Exa 14.24 efficiencies at diffrent LFs

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-24

clear; clc; close; // Clear the work space and console.

// Given data(From Ex.14-23)
```

```
V_1 = 2300 ; // Primary voltage in volt
V_2 = 230 ; // Secondary voltage in volt
S = 500 ; // Power rating of the transformer in kVA
f= 60 ; // Frequency in Hz

// Open circuit test data
V_oc = 208 ; // Open circuit voltage in volt
I_oc = 85 ; // Open circuit current in A
P_oc = 1800 ; // Power measured in W

// Short circuit test data
V_sc = 95 ; // Short circuit voltage in volt
I_sc = 217.4 ; // Short circuit current in A
P_sc = 8200 ; // Power measured in W

// Calculations
// Preliminary calculations
S_b = S ; // Base voltage in kVA
Psc = 8.2 ; // Power measured in kW during SC-test
P_Cu_pu = Psc / S_b ; // per unit value of P_Cu at
    rated load

Poc = 1.8 ; // Power measured in kW during OC-test
P_CL_pu = Poc / S_b ; // per unit value of P_CL at
    rated load

// case a
LF1 = 3/4 ; // Load fraction of rated load
PF1 = 1 ; // unity Power factor
eta_pu_LF1 = (LF1*PF1) / ((LF1*PF1) + P_CL_pu + (LF1
    )^2*P_Cu_pu ) * 100 ; // Efficiency at rated load
    ,unity PF

// case b
LF2 = 1/4 ; // Load fraction of rated load
PF2 = 0.8 ; // 0.8 lagging PF
eta_pu_LF2 = (LF2*PF2) / ((LF2*PF2) + P_CL_pu + (LF2
    )^2*P_Cu_pu ) * 100 ; // Efficiency at 1/4 rated
```

```
   load ,0.8 lagging PF

// case c
LF3 = 5/4 ; // Load fraction of rated load
PF3 = 0.8 ; // 0.8 leading PF
eta_pu_LF3 = (LF3*PF3) / ((LF3*PF3) + P_CL_pu + (LF3
   )^2*P_Cu_pu ) * 100 ; // Efficiency at r1/4 rated
    load ,0.8 leading PF


// Display the results
disp("Example 14-24 Solution : ");

printf(" \n     Efficiency(pu) :\n ");
printf(" \n a:   _pu  at %.2f rated-load = %.2f
   percent \n",LF1,eta_pu_LF1);

printf(" \n b:   _pu  at %.2f rated-load = %.2f
   percent \n",LF2,eta_pu_LF2);

printf(" \n c:   _pu  at %.2f rated-load = %.2f
   percent \n",LF3,eta_pu_LF3);
```

Scilab code Exa 14.25 Zpu2 St S2 S1 LF

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-25

clear; clc; close; // Clear the work space and
   console.
```

```

// Given data
kVA_1 = 500 ; // Power rating of the transformer 1 in kVA
R_1_pu = 0.01 ; // per-unit value of resistance of the transformer 1
X_1_pu = 0.05 ; // per-unit value of reactance of the transformer 1
Z_1_pu = R_1_pu + %i*X_1_pu ; //per-unit value of impedance of the transformer 1

PF = 0.8 ; // lagging power factor
V_2 = 400 ; // Secondary voltage in volt
S_load = 750 ; // Increased system load in kVA

kVA_2 = 250 ; // Power rating of the transformer  2 in kVA
R_pu_2 = 0.015 ; // per-unit value of resistance of the transformer 2
X_pu_2 = 0.04 ; // per-unit value of reactance of the transformer 2

// smaller transformer secondary voltageis same as larger transformer

// Calculations
// Preliminary calculations
Z_pu_1 = R_pu_2 + %i*X_pu_2 ; // New transformer p.u. impedance

// Calculations
// case a
V_b1 = 400 ; // base voltage in volt
V_b2 = 400 ; // base voltage in volt
Z_pu_2 = (kVA_1/kVA_2)*(V_b1/V_b2)^2 * (Z_pu_1); // New transformer p.u impedance
Z_2_pu = Z_pu_2 ; //New transformer p.u impedance

```

```
// case b
cos_theta = PF ; // Power factor
sin_theta = sqrt( 1 - (cos_theta)^2 );
S_t_conjugate = (kVA_1 + kVA_2)*(cos_theta + %i*sin_theta); // kVA of total load

// case c
S_2_conjugate = S_t_conjugate * ( Z_1_pu /(Z_1_pu + Z_2_pu) ); // Portion of load carried by the smaller transformer in kVA
S_2_conjugate_m = abs(S_2_conjugate);//S_2_conjugate_m=magnitude of S_2_conjugate in kVA
S_2_conjugate_a = atan(imag(S_2_conjugate) /real(S_2_conjugate))*180/%pi;//S_2_conjugate_a=phase angle of S_2_conjugate in degrees

// case d
S_1_conjugate = S_t_conjugate * ( Z_2_pu/(Z_1_pu + Z_2_pu) ); // Portion of load carried by the original transformer in kVA
S_1_conjugate_m = abs(S_1_conjugate);//S_1_conjugate_m=magnitude of S_1_conjugate in kVA
S_1_conjugate_a = atan(imag(S_1_conjugate) /real(S_1_conjugate))*180/%pi;//S_1_conjugate_a=phase angle of S_1_conjugate in degrees

// case e
S_1 = S_1_conjugate_m ;
S_b1 = kVA_1 ; // base power in kVA of trancsformer 1
LF1 = (S_1 / S_b1)*100 ; // Load fraction of the original transformer in percent

// case f
S_2 = S_2_conjugate_m ;
S_b2 = kVA_2 ; // base power in kVA of trancsformer 2
LF2 = (S_2 / S_b2)*100 ; // Load fraction of the
```

```
    original transformer in percent

// Display the results
disp("Example 14-25 Solution : ");

printf(" \n a: New transformer p.u impedance :\n
    Z_p.u.2 in p.u = ");disp(Z_pu_2);

printf(" \n b: kVA of total load :\n    S*_t in kVA
    = ");disp(S_t_conjugate);

printf(" \n c: Portion of load carried by the
    smaller transformer in kVA :");
printf(" \n    S*_2 in kVA =  ");disp(S_2_conjugate)
    ;
printf(" \n    S*_2 = %.1f <%.2f kVA (inductive load
    )\n",S_2_conjugate_m,S_2_conjugate_a);

printf(" \n d: Portion of load carried by the
    original transformer in kVA :");
printf(" \n    S*_2 in kVA = ");disp(S_1_conjugate);
printf(" \n    S*_2 = %.1f <%.2f kVA (inductive load
    )\n",S_1_conjugate_m,S_1_conjugate_a);

printf(" \n e: Load fraction of the original
    transformer :\n    L.F.1 = %.1f percent\n",LF1);

printf(" \n f: Load fraction of the original
    transformer :\n    L.F.2 = %.1f percent\n",LF2);

printf(" \n g: Yes. Reduce the no-load voltage of
    the new transformer to some value ");
printf(" \n    below that of its present value so
    that its share of the load is reduced.");
```

Scilab code Exa 14.26 Vb Ib Zb Z1 Z2 I1 I2 E1 E2

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-26

clear; clc; close; // Clear the work space and console.

// Given data(From Ex.14-25)
kVA_1 = 500 ; // Power rating of the transformer 1 in kVA
R_1_pu = 0.01 ; // per-unit value of resistance of the transformer 1
X_1_pu = 0.05 ; // per-unit value of reactance of the transformer 1
Z_1_pu = R_1_pu + %i*X_1_pu ; //per-unit value of impedance of the transformer 1

PF = 0.8 ; // lagging power factor
V = 400 ; // Secondary voltage in volt
S_load = 750 ; // Increased system load in kVA

kVA_2 = 250 ; // Power rating of the transformer  2 in kVA
R_pu_2 = 0.015 ; // per-unit value of resistance of the transformer 2
X_pu_2 = 0.04 ; // per-unit value of reactance of the transformer 2

// smaller transformer secondary voltageis same as larger transformer

// Calculations
```

```
// Preliminary calculations
Z_pu_1 = R_pu_2 + %i*X_pu_2 ; // New transformer p.u. impedance

// case a
V_b = V ; // (given)

//case b
S_b =500*1000 ; // base power in VA
I_b = S_b / V_b ; // base current in A

// case c
Z_b = V^2/S_b ; // Base impedance in ohm

// case d
Z_1 = Z_b * Z_1_pu * 1000 ; // Actual impedance of larger transformer in milli-ohm
Z_1_m = abs(Z_1);//Z_1_m=magnitude of Z_1 in ohm
Z_1_a = atan(imag(Z_1) /real(Z_1))*180/%pi;//Z_1_a=phase angle of Z_1 in degrees

// case e
V_b1 = V_b ; // base voltage in volt
V_b2 = V_b ; // base voltage in volt
Z_pu_2 = (kVA_1/kVA_2)*(V_b1/V_b2)^2 * (Z_pu_1); // New transformer p.u impedance
Z_2_pu = Z_pu_2 ; //New transformer p.u impedance

Z_2 = Z_b * Z_2_pu*1000 ; // Actual impedance of smaller transformer in milli-ohm
Z_2_m = abs(Z_2);//Z_2_m=magnitude of Z_2 in ohm
Z_2_a = atan(imag(Z_2) /real(Z_2))*180/%pi;//Z_2_a=phase angle of Z_2 in degrees

// case f
cos_theta = 0.8 ; // Power factor
sin_theta = sqrt( 1 - (cos_theta)^2 );
S_T = (kVA_1 + kVA_2)*(cos_theta - %i*sin_theta); //
```

```
kVA of total load

I_T = S_T*1000 / V_b ; // Total current in A

I_1 = I_T*(Z_2/(Z_1 + Z_2)); // Actual current delivered by larger transformer in A
I_1_m = abs(I_1);//I_1_m=magnitude of I_1 in A
I_1_a = atan(imag(I_1) /real(I_1))*180/%pi;//I_1_a=phase angle of I_1 in degrees

// case g
I_2 = I_T*(Z_1/(Z_1 + Z_2)); // Actual current delivered by larger transformer in A
I_2_m = abs(I_2);//I_2_m=magnitude of I_2 in A
I_2_a = atan(imag(I_2) /real(I_2))*180/%pi;//I_2_a=phase angle of I_2 in degrees

// case h
Z1 = Z_1/1000 ; // Z_1 in ohm
E_1 = I_1*Z1 + V_b ; // No-load voltage of larger Tr. in volt
E_1_m = abs(E_1);//E_1_m=magnitude of E_1 in volt
E_1_a = atan(imag(E_1) /real(E_1))*180/%pi;//E_1_a=phase angle of E_1 in degrees


// case i
Z2 = Z_2/1000 ; // Z_2 in ohm
E_2 = I_2*Z2 + V_b ; // No-load voltage of smaller Tr. in volt
E_2_m = abs(E_2);//E_2_m=magnitude of E_2 in volt
E_2_a = atan(imag(E_2) /real(E_2))*180/%pi;//E_2_a=phase angle of E_2 in degrees

// Display the results
disp("Example 14-26 Solution : ");

printf(" \n a: Base voltage :\n    V_b = %d <0 V (
```

```
given)\n",V_b);

printf("\n b: Base current :\n    I_b = %.2f A \n",I_b);

printf("\n c: Base impedance :\n    Z_b = %.2f ohm\n",Z_b);

printf("\n d: Actual impedance of larger transformer :\n    Z_1 in m  = \n");disp(Z_1);
printf("\n    Z_1 = %.2f <%.2f  m  \n ",Z_1_m,Z_1_a);

printf("\n e: Actual impedance of smaller transformer :\n    Z_1 in m  = \n");disp(Z_2);
printf("\n    Z_1 = %.2f <%.2f  m  \n ",Z_2_m,Z_2_a);

printf("\n f: Actual current delivered by larger transformer :\n    I_1 in A = ");disp(I_1);
printf("\n    I_1 = %.2f <%.2f A \n ",I_1_m,I_1_a);

printf("\n g: Actual current delivered by smaller transformer :\n    I_2 in A = ");disp(I_2);
printf("\n    I_1 = %.2f <%.2f A \n ",I_2_m,I_2_a);

printf("\n h: No-load voltage of larger Tr :\n    E_1 in volt = ");disp(E_1);
printf("\n    E_1 = %.2f <%.2f V \n ",E_1_m,E_1_a);

printf("\n i: No-load voltage of smaller Tr :\n    E_2 in volt = ");disp(E_2);
printf("\n    E_1 = %.2f <%.2f V \n ",E_2_m,E_2_a);
```

Scilab code Exa 14.27 RL ZbL ZLpu Z2pu Z1pu IbL ILpu VRpu VSpu VS VxVxpu

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-27

clear; clc; close; // Clear the work space and console.

// Given data
// From diagram in fig.14-23a
P_L = 14400 ; // Load output power in W
V_L = 120 ; // Load voltage in volt
V_b1 = 120 ; // base voltage at point 1 in volt
V_b2 = 600 ; // base voltage at point 2 in volt
V_b3 = 120 ; // base voltage at point 3 in volt
S_b3 = 14.4 ; // base power in kVA
X_2 = %i*0.25 ; // reactance in p.u
X_1 = %i*0.2 ; // reactance in p.u
I_L = 120 ; // Load current in A

// Calculations
// case a
R_L = P_L / (V_L^2); // Resistance of the load in ohm

// case b
Z_bL = (V_b3^2)/(S_b3*1000); // Base impedance in ohm

// case c
Z_L_pu = R_L / Z_bL ; // per unit load impedance

```

```
// case d
Z_2_pu = X_2 ; // per unit impedance of Tr.2

// case e
Z_1_pu = X_1 ; // per unit impedance of Tr.1

// case g
I_bL = (S_b3*1000)/V_b3 ; // Base current in load in
    A

// case h
I_L_pu = I_L / I_bL ; // per unit load current

// case i
V_R_pu = I_L_pu * Z_L_pu ; // per unit voltage
    across load

// case j
I_S_pu = I_L_pu ; //per unit current of source
Z_T_pu = Z_L_pu + Z_1_pu + Z_2_pu ; // Total p.u
    impedance
V_S_pu = I_S_pu * Z_T_pu ; // per unit voltage of
    source
V_S_pu_m = abs(V_S_pu);//V_S_pu_m=magnitude of
    V_S_pu in p.u
V_S_pu_a = atan(imag(V_S_pu) /real(V_S_pu))*180/%pi;
    //V_S_pu_a=phase angle of V_S_pu in degrees

// case k
V_S = V_S_pu * V_b1 ; // Actual voltage across
    source in volt
V_S_m = abs(V_S);//V_S_m=magnitude of V_S in volt
V_S_a = atan(imag(V_S) /real(V_S))*180/%pi;//V_S_a=
    phase angle of V_S in degrees


// case l
I_x_pu = I_L_pu ; // p.u current at point x
```

```
Z_x_pu = Z_L_pu + Z_2_pu ; // p.u impedance at point x
V_x_pu = I_x_pu * Z_x_pu ; // p.u voltage at point x

// case m
V_x = V_x_pu * V_b2 ; // Actual voltage at point x in volt
V_x_m = abs(V_x);//V_x_m=magnitude of V_x in volt
V_x_a = atan(imag(V_x) /real(V_x))*180/%pi;//V_x_a=phase angle of V_x in degrees


// Display the results
disp("Example 14-27 Solution : ");

printf(" \n a: Resistance of the load :\n    R_L = %d    \n",R_L);

printf(" \n b: Base impedance :\n    Z_bL = %d    \n",Z_bL);

printf(" \n c: per unit load impedance :\n    Z_L(pu)  = ");disp(Z_L_pu);

printf(" \n d: per unit impedance of Tr.2 :\n    Z_2(pu) = ");disp(Z_2_pu);

printf(" \n e: per unit impedance of Tr.1 :\n    Z_1(pu)  = ");disp(Z_1_pu);

printf(" \n f: See Fig.14-23b \n");

printf(" \n g: Base current in load :\n    I_bL = %d A (resistive)\n",I_bL);

printf(" \n h: per unit load current :\n    I_L_pu = ");disp(I_L_pu);

```

```
printf(" \n i: per unit voltage across load :\n V_R_pu");disp(V_R_pu);

printf(" \n j: per unit voltage of source :\n V_S_pu = ");disp(V_S_pu);
printf(" \n    V_S_pu = %.3f <%.2f p.u \n",V_S_pu_m,V_S_pu_a);

printf(" \n k: Actual voltage across source :\n V_S in volt = ");disp(V_S);
printf(" \n    V_S = %.1f <%.2f V \n",V_S_m,V_S_a);

printf(" \n l: p.u voltage at point x :\n    V_x(pu) = ");disp(V_x_pu);

printf(" \n m: Actual voltage at point x  :\n    V_x in volt = ");disp(V_x);
printf(" \n    V_S = %.1f <%.2f V \n",V_x_m,V_x_a);
```

Scilab code Exa 14.28 ZT1 ZT2 Zbline3 Zlinepu VLpu IbL IL ILpu VSpu VS

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-28

clear; clc; close; // Clear the work space and console.

// Given data
// From diagram in fig.14-24a
```

```
V_1 = 11 ; // Tr.1 voltage in kV
V_b1 = 11 ; // Base Tr.1 voltage in kV
S_1 = 50 ; // KVA rating of power for Tr.1
S_2 = 100 ; // KVA rating of power for Tr.2
Z_1_pu = %i*0.1 ; // per unit impedance of Tr.1
Z_2_pu = %i*0.1 ; // per unit impedance of Tr.2
V_b2 = 55 ; // Base Tr.2 voltage in kV
S_b = 100 ; // base power in kVA
PF = 0.8 ; // power factor of the Tr.s

Z_line = %i*200 ; // line impedance in ohm

V_L = 10 ;// Load voltage in kV
V_Lb3 = 11 ; // base line voltage at point 3

V_b3 = 11 ; // line voltage at point 3

P_L = 50 ; // Power rating of each Tr.s in kW
cos_theta_L = 0.8  ; // PF operation of each Tr.s

// Calculations
// case a
Z_T1 = Z_1_pu * (V_1/V_b1)^2 * (S_2/S_1); // p.u
   impedance of Tr.1

// case b
Z_T2 = Z_2_pu * (V_1/V_b3)^2 * (S_2/S_1); // p.u
   impedance of Tr.1

// case c
V_b = 55 ; // base voltage in volt
Z_b_line = (V_b^2)/S_b * 1000 ; // base line
   impedance in ohm
Z_line_pu = Z_line / Z_b_line ; // p.u impedance of
   the transmission line

// case d
V_L_pu = V_L / V_Lb3 ; // p.u voltage across load
```

```
// case e
//See Fig.14-24b

// case f
I_bL = S_b / V_b3 ; // base current in load in A

// case g
VL = 11 ; // load voltage in kV
cos_theta_L = 0.8 ; // power factor
I_L = P_L / (VL*cos_theta_L);
I_L_pu = I_L / I_bL ; // p.u load current
theta = acosd(0.8);
I_Lpu = I_L_pu*(cosd(theta) - %i*sind(theta)) ;// p.
    u current in complex form

// case h
Z_series_pu = Z_T1 + Z_line_pu + Z_T2 ; // p.u
    series impedance os the transmission line
V_S_pu = I_Lpu * Z_series_pu + V_L_pu ; // p.u
    source voltage
V_S_pu_m = abs(V_S_pu);//V_S_pu_m=magnitude of
    V_S_pu in p.u
V_S_pu_a = atan(imag(V_S_pu) /real(V_S_pu))*180/%pi;
    //V_S_pu_a=phase angle of V_S_pu in degrees

// case i
V_S = V_S_pu_m * V_b1 ; // Actual value of source
    voltage in kV
V_source = V_S*exp(%i*(V_S_pu_a)*(%pi/180)); // V_S
    in exponential form
V_source_m = abs(V_source);//V_source_m=magnitude of
     V_source in p.u
V_source_a = atan(imag(V_source) /real(V_source))
    *180/%pi;//V_source_a=phase angle of V_source in
    degrees


```

```
// Display the results
disp("Example 14-28 Solution : ");

printf(" \n a: p.u impedance of Tr.1 :\n    Z_T1 = ");disp(Z_T1);

printf(" \n b: p.u impedance of Tr.2 :\n    Z_T2 = ");disp(Z_T2);

printf(" \n c: base line impedance in ohm :\n    Z_b(line) = %d ohm \n",Z_b_line);
printf(" \n    p.u impedance of the transmission line :\n    Z(line)_pu = ");disp(Z_line_pu);

printf(" \n d: p.u voltage across load :\n    V_L_pu = ");disp(V_L_pu);

printf(" \n e: See Fig.14-24b \n");

printf(" \n f: base current in load :\n    I_bL = %.3f A \n",I_bL);

printf(" \n g: Load current :\n    I_L = %f A \n",I_L);
printf(" \n    p.u load current:\n    I_L_pu = %.3f at %.1f PF lagging \n",I_L_pu,PF);
printf(" \n    p.u current in complex form :\n    I_L_pu = ");disp(I_Lpu);

printf(" \n h: per unit voltage of source :\n    V_S_pu = ");disp(V_S_pu);
printf(" \n    V_S_pu = %.3f <%.2f p.u \n",V_S_pu_m,V_S_pu_a);

printf(" \n i: Actual voltage across source :\n    V_S in kV = ");disp(V_source);
printf(" \n    V_S = %.1f <%.2f kV \n",V_source_m,V_source_a);
```

Scilab code Exa 14.29 Z1pu Z2pu Vbline Zlinepu ZMs

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-29

clear; clc; close; // Clear the work space and console.

// Given data
// From diagram in fig.14-25a
Z_pu_1 = %i*0.1 ; //  p.u impedance
MVA_2 = 80 ; // MVA rating os system 2
MVA_1 = 100 ; // MVA rating of Tr.s 1 and 2
V_2 = 30 ; // voltage in KV
V_1 = 32 ; // voltage in KV

Z_pu_2 = %i*0.15 ; //  p.u impedance

V_b1 = 100 ; // base voltage of Tr.1

Z_line = %i*60 ; // Line impedance

MVA_M1 = 20 ; // MVA rating of motor load 1
Z_pu_M1 = %i*0.15 ; // p.u impedance of motor load M1

MVA_M2 = 35 ; // MVA rating of motor load 2
Z_pu_M2 = %i*0.25 ; // p.u impedance of motor load M2
```

```

MVA_M3 = 25 ; // MVA rating of motor load 3
Z_pu_M3 = %i*0.2 ; // p.u impedance of motor load M3

V_M = 28 ; // voltage across motor loads M1,M2,M3 in
    kV

// Calculations
// case a
Z_1_pu = Z_pu_1*(MVA_2/MVA_1)*(V_2/V_1)^2 ; // p.u
    imepedance of T1

// case b
Z_2_pu = Z_pu_2*(MVA_2/MVA_1)*(V_2/V_1)^2 ; // p.u
    imepedance of T2

// case c
V_b_line = V_b1*(V_1/V_2) ; // base voltage of the
    long-transmission line in kV

// case d
MVA_b = 80 ; // MVA rating
V_b = V_b_line ;
Z_line_pu = Z_line*(MVA_b/(V_b)^2); // p.u impedance
    of the transmission line

// case e
Z_M1_pu = Z_pu_M1 * (MVA_2/MVA_M1)*(V_M/V_1)^2 ; //
    p.u impedance of motor load M1
Z_M2_pu = Z_pu_M2 * (MVA_2/MVA_M2)*(V_M/V_1)^2 ; //
    p.u impedance of motor load M2
Z_M3_pu = Z_pu_M3 * (MVA_2/MVA_M3)*(V_M/V_1)^2 ; //
    p.u impedance of motor load M3

// Display the results
disp("Example 14-29 Solution : ");

printf(" \n a: p.u imepedance of T1 :\n    Z_1(pu) =
```

```
    ");disp(Z_1_pu);

printf(" \n b: p.u imepedance of T2 :\n    Z_2(pu) =
    ");disp(Z_2_pu);

printf(" \n c: base voltage of the long-transmission
    line :\n    V_b(line) = %.1f kV \n",V_b_line);

printf(" \n d: p.u impedance of the transmission
   line :\n    Z(line)_pu = ");disp(Z_line_pu);

printf(" \n e: p.u impedance of motor load M1 :\n
      Z_M1(pu) = ");disp(Z_M1_pu);

printf(" \n f: p.u impedance of motor load M1 :\n
      Z_M2(pu) = ");disp(Z_M2_pu);

printf(" \n g: p.u impedance of motor load M1 :\n
      Z_M3(pu) = ");disp(Z_M3_pu);

printf(" \n h: See Fig.14-25b.");
```

Scilab code Exa 14.30 ST ST Sxformer

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-30

clear; clc; close; // Clear the work space and
   console.

```

```
// Given data
// subscripts a,b,c for the current , voltages
    indicates respective cases a ,b ,c.
// from fig.14-27a
V_pa = 1000 ; // Phase voltage in volt
I_1a = 1 ; // line current in primary in A
V_2a = 100 ; // voltage across secondary in V
Ic_a = 10 ; // current in lower half of auto-
    transformer in A

// from fig.14-26b
V_s = 100 ; // voltage in secondary wdg in V
I_2b = 10 ; // current in secondary in A
V_1b = 1000 ; // voltage across primary in V
Ic_b = 1 ; // current in lower half of auto-
    transformer in A

// Calculations
// case a
S_T1 = (V_pa*I_1a + V_2a*I_1a)/1000 ; // Total kVA
    transfer in step-down mode

// case b
S_T2 = (V_s*I_2b + V_1b*I_2b)/1000 ; // Total kVA
    transfer in step-up mode

// case c
S_x_former_c = V_pa*I_1a/1000 ; // kVA rating of th
    autotransformer  in Fig.14-27a

// case d
V_1 = V_pa ;
S_x_former_d = V_1*Ic_b/1000  ; // kVA rating of th
    autotransformer  in Fig.14-26b


// Display the results
disp("Example 14-30 Solution : ");
```

```
printf(" \n a: Total kVA transfer in step-down mode :\n    S_T = %.1f kVA transferred \n",S_T1);

printf(" \n b: Total kVA transfer in step-up mode :\n    S_T = %.1f kVA transferred \n",S_T2);

printf(" \n c: kVA rating of th autotransformer in Fig.14-27a:\n    S_x-former = %d kVA \n ",S_x_former_c);

printf(" \n d: kVA rating of th autotransformer in Fig.14-26b:\n    S_x-former = %d kVA \n ",S_x_former_d);

printf(" \n e: Both transformers have the same kVA rating of 1 kVA since the same ");
printf(" \n    autotransformer is used in both parts.Both transformers transform ");
printf(" \n    a total of 1 KVA. But the step-down transformer in part(a) conducts ");
printf(" \n    only 0.1 kVA while the step-up transformer in the part(b) conducts 10");
printf(" \n    kVA from the primary to the secondary.");
```

Scilab code Exa 14.31 Wc tabulate allday efficiency

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-31
```

```
clear; clc; close; // Clear the work space and
    console.

// Given data
S = 500 ; // kVA rating of distribution transformer
// given data from ex.14-20
V_1 = 2300 ; // Primary voltage in volt
V_2 = 208 ; // Secondary voltage in volt
f = 60 ; // Frequency in Hz

// SC-test data
P_sc = 8200 ; // wattmeter reading in W
I_sc = 217.4 ; // Short circuit current in A
V_sc = 95 ; // Short circuit voltage in V

// OC-test data
P_oc = 1800 ; // wattmeter reading in W
I_oc = 85 ; // Open circuit current in A
V_oc = 208 ; // Open circuit voltage in V

LF_1 = 20 ; // Load fraction in percent
LF_2 = 40 ; // Load fraction in percent
LF_3 = 80 ; // Load fraction in percent
LF_fl = 100 ; // rated load in percent
LF_4 = 125 ; // Load fraction in percent

LF1 = 0.2 ; // Load fraction
LF2 = 0.4 ; // Load fraction
LF3 = 0.8 ; // Load fraction
LF4 = 1.25 ; // Load fraction

PF1 = 0.7 ; // power factor
PF2 = 0.8 ; // power factor
PF3 = 0.9 ; // power factor
PF_fl = 1 ; // power factor
PF4 = 0.85 ; // power factor

```

```
t1 = 4 ; // period of operation in hours
t2 = 4 ; // period of operation in hours
t3 = 6 ; // period of operation in hours
t_fl = 6 ; // period of operation in hours
t4 = 2 ; // period of operation in hours

// Calculations
// case a
t = 24 ; // hrs in a day
P_c  = P_oc ; // wattmeter reading in W (OC test)
W_c = (P_c * t)/1000 ; // COre loss over 24 hour period

// case b
Psc = P_sc/1000 ; // wattmeter reading in W (SC test)
P_loss_1 = (LF1^2)*Psc ; // Power loss in kW for 20% Load
P_loss_2 = (LF2^2)*Psc ; // Power loss in kW for 40% Load
P_loss_3 = (LF3^2)*Psc ; // Power loss in kW for 80% Load
P_loss_fl = Psc ; // Power loss in kW for 100% Load
P_loss_4 = (LF4^2)*Psc ; // Power loss in kW for 125% Load

// energy loss in kWh
energy_loss1 = P_loss_1 * t1 ; // Enegry loss in kWh for 20% Load
energy_loss2 = P_loss_2 * t2 ; // Enegry loss in kWh for 40% Load
energy_loss3 = P_loss_3 * t3 ; // Enegry loss in kWh for 80% Load
energy_loss_fl = P_loss_fl * t_fl ; // Enegry loss in kWh for 100% Load
energy_loss4 = P_loss_4 * t4 ; // Enegry loss in kWh for 125% Load

```

```
// Total energy losses in 24hrs
W_loss_total = energy_loss1 + energy_loss2 + energy_loss3 + energy_loss_fl + energy_loss4 ;

// case c
P_1 = LF1*S*PF1 ; // Power output for 20% load
P_2 = LF2*S*PF2 ; // Power output for 40% load
P_3 = LF3*S*PF3 ; // Power output for 80% load
P_fl = S*PF_fl ; // Power output for 100% load
P_4 = LF4*S*PF4 ; // Power output for 125% load

Energy_1 = P_1*t1 ; // Energy delivered in kWh for 20%load
Energy_2 = P_2*t2 ; // Energy delivered in kWh for 40%load
Energy_3 = P_3*t3 ; // Energy delivered in kWh for 80%load
Energy_fl = P_fl*t_fl ; // Energy delivered in kWh for 100%load
Energy_4 = P_4*t4 ; // Energy delivered in kWh for 125%load

// Total energy delivered in 24hrs
W_out_total = Energy_1 + Energy_2 + Energy_3 + Energy_fl + Energy_4 ;

// case d
eta = W_out_total / (W_out_total + W_c + W_loss_total) * 100 ; // All-day efficiency

// Display the results
disp("Example 14-31 Solution : ");

printf(" \n a: Total energy core loss for 24hrs, including 2hours at no-load,");
printf(" \n     W_c = %.1f kWh \n ",W_c);

printf(" \n b: From SC test, equivalent copper loss
```

```
    at rated load = %.1f kW,  ",Psc);
printf(" \n    and the various energy losses during
    the 24 hr period are tabulated as :\n");

printf(" \n
    -----------------------------------------------------------------
    ");
printf(" \n    Percent Rated load \t Power loss(kW)
    \t Time period(hours) \t Energy loss(kWh)");
printf(" \n
    -----------------------------------------------------------------
    ");
printf(" \n\t\t%d \t  %f \t\t\t %d \t\t\t %.2f \n ",
    LF_1,P_loss_1,t1,energy_loss1);
printf(" \n\t\t%d \t  %f \t\t\t %d \t\t\t %.2f \n ",
    LF_2,P_loss_2,t2,energy_loss2);
printf(" \n\t\t%d \t  %f \t\t\t %d \t\t\t %.2f \n ",
    LF_3,P_loss_3,t3,energy_loss3);
printf(" \n\t\t%d \t  %f \t\t\t %d \t\t\t %.2f \n ",
    LF_fl,P_loss_fl,t_fl,energy_loss_fl);
printf(" \n\t\t%d \t  %f \t\t\t %d \t\t\t %.2f \n ",
    LF_4,P_loss_4,t4,energy_loss4);
printf(" \n
    -----------------------------------------------------------------
    ");
printf(" \n    Total energy load losses over 24hour
    period (excluding 2hrs at no-load)  =  %.2f ",
    W_loss_total);
printf(" \n
    -----------------------------------------------------------------
    \n\n");

printf(" \n c: Total energy output over the 24 hour
    period is tabulated as : \n");

printf(" \n
    -----------------------------------------------------------------
    ");
```

```
printf(" \n     Percent  Rated  load  \t  PF  \t  kW  \t
    Time  period(hours)  \t  Energy  delivered(kWh)");
printf(" \n
    ------------------------------------------------------------------
    ");
printf(" \n\t\t%d \t %.1f \t %.f \t\t %d \t\t\t %d "
    ,LF_1,PF1,P_1,t1,Energy_1);
printf(" \n\t\t%d \t %.1f \t %.f \t\t %d \t\t\t %d "
    ,LF_2,PF2,P_2,t2,Energy_2);
printf(" \n\t\t%d \t %.1f \t %.f \t\t %d \t\t\t %d "
    ,LF_3,PF3,P_3,t3,Energy_3);
printf(" \n\t\t%d \t %.1f \t %.f \t\t %d \t\t\t %d "
    ,LF_fl,PF1,P_fl,t_fl,Energy_fl);
printf(" \n\t\t%d \t %.1f \t %.f \t\t %d \t\t\t %d "
    ,LF_4,PF4,P_4,t4,Energy_4);
printf(" \n
    ------------------------------------------------------------------
    ");
printf(" \n     Total  energy  required  by  load  for  24
    hour  period  (excluding  2hrs  at  no-load)   =   %d ",
    W_out_total);
printf(" \n
    ------------------------------------------------------------------
    \n\n");

printf(" \n d:  All-day  efficiency  = %.1f  percent",
    eta);
```

Scilab code Exa 14.32 I2 Ic

```
// Electric  Machinery  and  Transformers
// Irving  L  kosow
// Prentice  Hall  of  India
// 2nd  editiom

```

```
// Chapter 14: TRANSFORMERS
// Example 14-32

clear; clc; close; // Clear the work space and
   console.

// Given data
S_1 = 10 ; // VA rating of small transformer
V = 115 ; //  voltage rating of transformer in volt
V_2_1 = 6.3 ; // voltage rating of one part of
   secondary winding in volt
V_2_2 = 5.0 ; // voltage rating of other part of
   secondary winding in volt
Z_2_1 = 0.2 ; // impedance of one part of secondary
   winding in ohm
Z_2_2 = 0.15 ; // impedance of other part of
   secondary winding in ohm


// Calculations
// case a
V_2 = V_2_1 + V_2_2 ; // voltage across secondary
   winding in volt
I_2 = S_1 / V_2 ; // Rated secondary current in A
   when the LV secondaries are
// connected in series-aiding

// case b
I_c = (V_2_1 - V_2_2) / (Z_2_1 + Z_2_2); //
   Circulating current when LV windings are paralled
percent_overload = (I_c / I_2)*100 ; // percent
   overload produced

// Display the results
disp("Example 14-32 Solution : ");

printf(" \n a: Both coils must be series-connected
   and used to account for the ");
```

```
printf(" \n    full VA rating of the transformer.
   Hence,the rated current in 5 V ");
printf(" \n    and 6.3 V winding is : \n");
printf(" \n    I_2 = %.3f A \n\n", I_2);

printf(" \n b: When the windings are paralleled , the
    net circulating current is ");
printf(" \n    the net voltage applied across the
   total internal impedance of ");
printf(" \n    the windings,or :\n");
printf(" \n    I_c = %.2f A \n ",I_c);

printf(" \n    The percent overload is = %f percent
       %.f percent ",percent_overload,
   percent_overload);
```

Scilab code Exa 14.33 Zeh Zel I2rated I2sc overload

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-33

clear; clc; close; // Clear the work space and
   console.

// Given data
S = 20 ; // kVA rating of transformer
N_1 = 230 ; // Number of primary turns
N_2 = 20 ; // Number of secondary turns

V_1 = 230 ; // Primary voltage in volt
```

```
V_2 = 20 ; // Secondary voltage in volt

// from Fig.14-31a
// HV side SC test data
V_sc = 4.5 ; // short circuit voltage in volt
I_sc = 87 ; // short circuit current in A
P_sc = 250 ; // Power measured in W

// Calculations
// case a
V_h = V_sc ;// short circuit voltage in volt on HV
   side
I_h = I_sc ;// short circuit current in A on HV side
Z_eh = V_h /I_h ; // Equivalent immpedance reffered
   to the high side when coils are series connected

// case b
Z_el = Z_eh * (N_2/N_1)^2 ; //Equivalent immpedance
   reffered to the low side
// when coils are series connected

// case c
I_2_rated = (S*1000)/V_2 ; // Rated secondary
   current when coils are series connected

// case d
I_2_sc = S / Z_el ; // Secondary current when the
   coils in Fig.14-31a are
// short-circuited with rated voltage applied to the
    HV side

percent_overload = (I_2_sc/I_2_rated)*100 ; //
   percent overload


// Display the results
disp("Example 14-33 Solution : ");

```

```
printf(" \n    Slight variations in answers are due
    to non-approximated calculations");
printf(" \n    in scilab\n\n");
printf(" \n a: Equivalent immpedance reffered to the
     high side when coils are series connected :");
printf(" \n    Z_eh = %f ohm \n ",Z_eh);

printf(" \n b: Equivalent immpedance reffered to the
     low side when coils are series connected :");
printf(" \n    Z_el = %f ohm \n ",Z_el);

printf(" \n c: Rated secondary current when coils
    are series connected :");
printf(" \n    I_2(rated) = %d A \n",I_2_rated);

printf(" \n d: Secondary current when the coils in
    Fig.14-31a are short-circuited :");
printf(" \n    with rated voltage applied to the HV
    side :");
printf(" \n    I_2(sc) = %d A \n",I_2_sc);
printf(" \n    The percent overload is = %d percent"
    ,percent_overload);
```

Scilab code Exa 14.34 PT kVA phase and line currents kVAtransformers

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-34

clear; clc; close; // Clear the work space and
    console.
```

```

// Given data
I_L = 100 ; // Load current in A
cos_theta = 0.7 ; // power factor lagging

// Y-  distribution transformer
S = 60 ; // kVA rating of transformer
V_1 = 2300 ; // primary voltage in volt
V_2 = 230 ; // secondary voltage in volt

// Calculations
// case a
V_L = 230 ; // voltage across load in volt
P_T = (sqrt(3)*V_L*I_L*cos_theta)/1000 ; // power consumed by the plant in kW
kVA_T = P_T/cos_theta ; // apparent power in kVA

// case b
kVA = S ; // kVA rating of transformer
V_p = V_2 ; // phase voltage in volt (delta-connection on load side)
I_P2_rated = (kVA*1000)/(3*V_p); // Rated secondary phase current in A
I_L2_rated = sqrt(3)*I_P2_rated ; // Rated secondary line current in A

// case c
//percent load on each transformer = (load current per line) / (rated current per line)
percent_load = I_L / I_L2_rated * 100 ;

// case d
// subscript d for V_L indicates case d ,V_L
V_L_d = 2300 ;
I_P1 = (kVA_T*1000)/(sqrt(3)*V_L_d); // primary phase current in A
I_L1 = I_P1 ; // primary line current in A(Y-connection)
```

```

// case e
kVA_transformer = kVA / 3 ; // kVA rating of each transformer

// Display the results
disp("Example 14-34 Solution : ");

printf(" \n a: power consumed by the plant :\n P_T = %.1f kW \n ",P_T);
printf(" \n    apparent power :\n    kVA_T = %.1f kVA \n",kVA_T);

printf(" \n b: Rated secondary phase current :\n I_P2(rated) = %f A     %.f A \n",I_P2_rated,I_P2_rated);
printf(" \n    Rated secondary line current :\n I_L2(rated) = %f A     %.1f A \n",I_L2_rated,I_L2_rated);

printf(" \n c: percent load on each transformer = %.1f percent \n ",percent_load);

printf(" \n d: primary phase current :\n    I_P1 = %.f A \n",I_P1);
printf(" \n    primary line current :\n    I_L1 = %.f A \n",I_L1);

printf(" \n e: kVA rating of each transformer = %d kVA",kVA_transformer);
```

Scilab code Exa 14.35 PT ST phase and line currents kVAtransformers

```
// Electric Machinery and Transformers
// Irving L kosow
```

```
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-35

clear; clc; close; // Clear the work space and
    console.

// Given data
I_L = 100 ; // Load current in A
cos_theta = 0.7 ; // power factor lagging

//   -   distribution transformer
S = 60 ; // kVA rating of transformer
V_1 = 2300 ; // primary voltage in volt
V_2 = 230 ; // secondary voltage in volt

// Calculations
// case a
V_L = 230 ; // voltage across load in volt
P_T = (sqrt(3)*V_L*I_L*cos_theta)/1000 ; // power
    consumed by the plant in kW
kVA_T = P_T/cos_theta ; // apparent power in kVA

// case b
kVA = S ; // kVA rating of transformer
V_p = V_2 ; // phase voltage in volt
I_P2_rated = (kVA*1000)/(3*V_p); // Rated secondary
    phase current in A
I_L2_rated = sqrt(3)*I_P2_rated ; // Rated secondary
     line current in A

// case c
//percent load on each transformer = (load current
    per line) / (rated current per line)
percent_load = I_L / I_L2_rated * 100 ;

```

```
// case d
// subscript d for V_L indicates case d ,V_L
V_L_d = 2300 ;
I_P1 = (kVA_T*1000)/(sqrt(3)*V_L_d); // primary
    phase current in A
I_L1 = sqrt(3)*I_P1 ; // primary line current in A

// case e
kVA_transformer = kVA / 3 ; // kVA rating of each
    transformer

// Display the results
disp("Example 14-35 Solution : ");

printf(" \n a: power consumed by the plant :\n
    P_T = %.1f kW \n ",P_T);
printf(" \n    apparent power :\n    kVA_T = %.1f
    kVA \n",kVA_T);

printf(" \n b: Rated secondary phase current :\n
    I_P2(rated) = %f A     %.f A \n",I_P2_rated,
    I_P2_rated);
printf(" \n    Rated secondary line current :\n
    I_L2(rated) = %f A     %.1f A \n",I_L2_rated,
    I_L2_rated);

printf(" \n c: percent load on each transformer = %
    .1f percent \n ",percent_load);

printf(" \n d: primary phase current :\n    I_P1 = %
    .f A \n",I_P1);
printf(" \n    primary line current :\n    I_L1 = %f
     A     %.1f A \n",I_L1,I_L1);
printf(" \n    The primary line current drawn by a
      -    bank is    3 times the ");
printf(" \n    line current drawn by a Y-    bank.\n"
    );

```

```
printf(" \n e: kVA rating of each transformer = %d
    kVA",kVA_transformer);
```

Scilab code Exa 14.36 find line currents and their sum

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-36

clear; clc; close; // Clear the work space and
    console.

// Given data
// 3-phase,3-wire  -connected transformer  shown in
     Fig.14-42
V_L = 33 ; // line voltage in kV

f = 60 ;// frequency in Hz

// power factor
PF1 = 1; // unity power factor for I_AB
PF2 = 0.7; // 0.7 lagging power factor for I_BC
PF3 = 0.9; // 0.9 lagging power factor for I_CA

// Calculations
V_AB = V_L*exp(%i*(0)*(%pi/180)) ; // line voltage
    in kV taken as reference voltage

V_BC = V_L*exp(%i*(-120)*(%pi/180)) ; // line
    voltage in kV
V_BC_m = abs(V_BC);//V_BC_m=magnitude of V_BC in kV
```

```
V_BC_a = atan(imag(V_BC) /real(V_BC))*180/%pi - 180 ;//V_BC_a=phase angle of V_BC in degrees
// 180 is subtracted from I_BC_a to make it similar to textbook angle

V_CA = V_L*exp(%i*(-240)*(%pi/180)) ; // line voltage in kV
V_CA_m = abs(V_CA);//V_CA_m=magnitude of V_CA in kV
V_CA_a = atan(imag(V_CA) /real(V_CA))*180/%pi - 180 ;//V_CA_a=phase angle of V_CA in degrees
// 180 is subtracted from I_BC_a to make it similar to textbook angle

theta_1 = acosd(PF1); // PF1 angle
theta_2 = acosd(PF2); // PF2 angle
theta_3 = acosd(PF3); // PF3 angle


I_AB = 10*exp(%i*(theta_1)*(%pi/180)) ; // I_AB current in kA
I_AB_m = abs(I_AB);//I_AB_m=magnitude of I_AB in kA
I_AB_a = atan(imag(I_AB) /real(I_AB))*180/%pi;//I_AB_a=phase angle of I_AB in degrees

I_BC = 15*exp(%i*(-120 - theta_2)*(%pi/180)) ; // I_BC current in kA
I_BC_m = abs(I_BC);//I_BC_m=magnitude of I_BC in kA
I_BC_a = atan(imag(I_BC) /real(I_BC))*180/%pi - 180;//I_BC_a=phase angle of I_BC in degrees
// 180 is subtracted from I_BC_a to make it similar to textbook angle

I_CA = 12*exp(%i*(-240 + theta_3)*(%pi/180)) ; // I_CA current in kA
I_CA_m = abs(I_CA);//I_CA_m=magnitude of I_CA in kA
I_CA_a = 180 + atan(imag(I_CA) /real(I_CA))*180/%pi;//I_CA_a=phase angle of I_CA in degrees
// 180 is added to I_BC_a to make it similar to
```

```
    textbook angle

// case a
I_AC = -I_CA ;
I_A = I_AB + I_AC ; // phase current in kA
I_A_m = abs(I_A);//I_A_m=magnitude of I_A in kA
I_A_a = atan(imag(I_A) /real(I_A))*180/%pi;//I_A_a=
    phase angle of I_A in degrees

// case b
I_BA = -I_AB ;
I_B = I_BC + I_BA ; // phase current in kA
I_B_m = abs(I_B);//I_B_m=magnitude of I_B in kA
I_B_a = atan(imag(I_B) /real(I_B))*180/%pi;//I_B_a=
    phase angle of I_B in degrees

// case c
I_CB = -I_BC ;
I_C = I_CA + I_CB ; // phase current in kA
I_C_m = abs(I_C);//I_C_m=magnitude of I_C in kA
I_C_a = atan(imag(I_C) /real(I_C))*180/%pi;//I_C_a=
    phase angle of I_C in degrees

// case d
phasor_sum = I_A + I_B + I_C ;


// Display the results
disp("Example 14-36 Solution : ");

printf(" \n We must first write each of the phase
    currents in polar form. ");
printf(" \n Since reference voltage ,V_AB is assumed
    as 33 <0 kV, we may write\n");

printf(" \n I_AB = %d <%d kA (unity PF),\n",I_AB_m,
    I_AB_a);
printf(" \n But I_BC lags V_BC, which is %.f <%d kV"
```

```
    ,V_BC_m,V_BC_a);
printf(" \n by   = acosd(%.1f) = -%.2f lag, and
    consequently",PF2,theta_2);
printf(" \n I_BC = %.f <%.2f kA \n",I_BC_m,I_BC_a);

printf(" \n Similarly,I_CA leads V_CA = %.f <%.f kV"
    ,V_CA_m,V_CA_a);
printf(" \n by   = acosd(%.1f) = %.2f lead, and
    consequently",PF3,theta_3);
printf(" \n I_CA = %d <%.2f kA \n",I_CA_m,I_CA_a);

printf(" \n Writing three phase currents in comples
    form yields.\n");
printf(" \n I_AB in kA = ");disp(I_AB);
printf(" \n I_BC in kA = ");disp(I_BC);
printf(" \n I_CA in kA = ");disp(I_CA);

printf(" \n From conventional three phase theory for
     unbalanced  -connected loads");
printf(" \n and from Fig.14-42, we have\n");

printf(" \n a: I_A in kA = ");disp(I_A);
printf(" \n    I_A = %.2f <%.2f kA \n",I_A_m,I_A_a);

printf(" \n b: I_B in kA = ");disp(I_B);
printf(" \n    I_B = %.2f <%.2f kA \n",I_B_m,I_B_a);

printf(" \n c: I_C in kA = ");disp(I_C);
printf(" \n    I_C = %.2f <%.2f kA \n",I_C_m,I_C_a);

printf(" \n d: Phasor sum of the line currents :");
printf(" \n     I_L in kA = ");disp(phasor_sum);
```

Scilab code Exa 14.37 kVAcarry loadtransformer VVkVA ratiokVA increaseload

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-37

clear; clc; close; // Clear the work space and
    console.

// Given data
//   -   transformers in Ex.35
kVA_1 = 20 ; // kVA rating of transformer 1
kVA_2 = 20 ; // kVA rating of transformer 2
kVA_3 = 20 ; // kVA rating of transformer 3

V_1 = 2300 ; // Primary voltage in volt
V_2 = 230 ; // Secondary voltage in volt

kVA = 40 ; // kVA supplied by the bank
PF = 0.7 ; // lagging power factor at which bank
    supplies kVA

// one defective transformer is removed

// Calculations
// case a
kVA_transformer = kVA / sqrt(3); // kVA load carried
     by each transformer

// case b
percent_ratedload_Tr = kVA_transformer / kVA_1 * 100
     ; // percent load carried by each transformer

// case c
kVA_V_V = sqrt(3)*kVA_1 ; // Total kVA rating of the
     transformer bank in V-V
```

```
// case d
ratio_banks = kVA_V_V / (kVA_1 + kVA_2 + kVA_3) * 100; // ratio of V-V bank to  -  bank Tr ratings

// case e
kVA_Tr = kVA / 3 ;
percent_increase_load = kVA_transformer / kVA_Tr * 100 ; // percent increase in load on each transformer when one Tr is removed


// Display the results
disp("Example 14-37 Solution : ");

printf(" \n a: kVA load carried by each transformer = %.1f kVA/transformer\n",kVA_transformer);

printf(" \n b: percent rated load carried by each transformer = %.1f percent \n", percent_ratedload_Tr);

printf(" \n c: Total kVA rating of the transformer bank in V-V = %.2f kVA \n",kVA_V_V);

printf(" \n d: ratio of V-V bank to  -  bank Tr ratings = %.1f percent \n",ratio_banks);

printf(" \n e: kVA load carried by each transformer(V-V) = %.2f kVA/transformer\n",kVA_Tr);
printf(" \n    percent increase in load on each transformer when one Tr is removed :");
printf(" \n      = %.1f percent", percent_increase_load);
```

Scilab code Exa 14.38 IL alpha Ia kVA

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-38

clear; clc; close; // Clear the work space and
   console.

// Given data
// 3-phase SCIM
V = 440 ; // rated voltage in volt of SCIM
hp = 100 ; // rated power in hp of SCIM
PF = 0.8 ; // power factor
V_1 = 155 ; // primary voltage in volt of Tr
V_2 = 110 ; // secondary voltage in volt of Tr

V_a = 110 ; // armature voltage in volt
V_L = 440 ; // Load voltage in volt
eta = .98 ; // efficiency of the Tr.

// Calculations
// case a
// referring to appendix A-3,Table 430-150 footnotes
I_L = 124*1.25 ; // Motor line current in A

// case b
alpha = V_a/V_L ; // Transformation ratio

// case c
```

```
I_a = (sqrt(3)/2)*( I_L / (alpha*eta) ); // Current
    in the primary of the scott transformers

// case d
kVA = (V_a*I_a)/((sqrt(3)/2)*1000); // kVA rating of
     the main and teaser transformers

// Display the results
disp("Example 14-38 Solution : ");

printf(" \n a: Motor line current :\n    I_L = %d A
    \n ",I_L);

printf(" \n b: Transformation ratio :\n    alpha =
    N_1/N_2 = V_a/V_L = %.2f \n",alpha);

printf(" \n c: Current in the primary of the scott
    transformers :\n    I_a = %.f A \n",I_a);

printf(" \n d: kVA rating of the main and teaser
    transformers :\n    kVA = %.1f kVA",kVA);
```

Scilab code Exa 14.39 VL ST Idc Sac Sdc per line

```
// Electric Machinery and Transformers
// Irving L kosow
// Prentice Hall of India
// 2nd editiom

// Chapter 14: TRANSFORMERS
// Example 14-39

clear; clc; close; // Clear the work space and
    console.

```

```
// Given data
I_L = 1 ; // Load current in kA
V_m = 750 ; // Peak voltage in kV

// Calculations
// case a
V_L = (V_m)/sqrt(2); // Max. allowable Vrms in kV that may be applied to the lines using ac

// case b
S_T_ac = sqrt(3)*V_L*I_L ; // Total 3-phase apparent power in MVA

// case c
I_rms = I_L ; // rms value of load current in kA
I_dc =I_rms*sqrt(2); // Max.allowable current in kA that can be delivered by dc transmission

// case d
V_dc = V_m ; // dc voltage in kV
S_T_dc = V_dc*I_dc ; // Total dc apparent power delivered by two lines in MVA

// case e
S_ac_line = S_T_ac / 3 ; // Power per ac line

// case f
S_dc_line = S_T_dc / 2 ; // Power per dc line

// Display the results
disp("Example 14-39 Solution : ");

printf(" \n :a Max. allowable Vrms in kV that may be applied to the lines using ac :");
printf(" \n     V_L = %.1f kV \n ",V_L);

printf(" \n :b Total 3-phase apparent power :\n S_T = %.1f MVA \n",S_T_ac);
```

```
printf(" \n :c Max.allowable current in kA that can be delivered by dc transmission :");
printf(" \n    I_dc = %.3f kA \n ",I_dc);

printf(" \n :d Total dc apparent power delivered by two lines :\n    S_T = %.1f MVA\n",S_T_dc);

printf(" \n :e Power per ac line :\n    S/ac line = %.1f MVA/line \n",S_ac_line);

printf(" \n :f Power per dc line :\n    S/dc line = %.1f MVA/line \n",S_dc_line);
```