

Scilab Textbook Companion for  
Electrical Power Systems  
by A. Husain<sup>1</sup>

Created by  
Shikhar Saxena  
B.E. 3rd yr  
Electrical Engineering  
Thapar University  
College Teacher  
Dr Sunil Kumar Singla  
Cross-Checked by  
Lavitha Pereira

July 31, 2019

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

# Book Description

**Title:** Electrical Power Systems

**Author:** A. Husain

**Publisher:** CBS Publishers

**Edition:** 5

**Year:** 2009

**ISBN:** 8123914482

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

**Exa** Example (Solved example)

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

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

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

# Contents

List of Scilab Codes	4
1 Load Characteristics	5
2 Supply Systems	15
3 Conductors	19
4 Power Cables	25
5 Line insulators and supports	32
6 Sag and tension	36
7 Line Parameters	41
8 per unit representation	56
9 Short and medium lines	68
10 Long transmission lines	80
11 General networks constants	83
13 Control of voltage and reactive power	89

14 Load flow analysis	96
15 Economic operations of power systems	106
16 Symmetrical faults	112
17 Symmetrical components	119
18 Unsymmetrical faults	127
19 Power system stability	133
20 Travelling waves	144
22 Corona	151
24 System neutral grounding	153
25 Tarrifs	155
26 Power factor improvement	163

# List of Scilab Codes

Exa 1.1	to calculate the average load monthly energy consumption and load factor . . . . .	5
Exa 1.2	to calculate the diversity factor avg load and laod factor of each consumer avg load and load factor of combined load . . . . .	6
Exa 1.3	to find the maximum value and the consumption of energy in kWh . . . . .	8
Exa 1.4	to plot the load duration curve from the chronological load curve . . . . .	8
Exa 1.5	to determine the load factor from the load duration curve . . . . .	9
Exa 1.6	to calculate plant capacity factor load factor utilization factor reserve capacity . . . . .	10
Exa 1.7	to plot the chronological load duration load energy curve and then calculating the load factor and the utilization factor . . . . .	10
Exa 1.8	to find the diversity factor of a power station	12
Exa 1.9	to calculate max demand on the station installed capacity energy supplied in a year . .	13
Exa 2.1	To calculate the percentage saving in conductor material . . . . .	15
Exa 2.2	To compare the amount of conductor material required . . . . .	15
Exa 2.3	To calculate the volume of conductor . . . . .	16
Exa 2.4	To calculate the percentage of additional load	17
Exa 2.5	To compare the diameter and weight . . . . .	17
Exa 3.1	To determine the most economical size . . . . .	19

Exa 3.2	to calculate the most economical cross sectional area . . . . .	20
Exa 3.3	to calculate the best current density of a three phase oh line . . . . .	21
Exa 3.4	To calculate rms value of curent for a 3 phase	22
Exa 3.5	To determine the most economical size of copper conductor . . . . .	23
Exa 3.6	To calculate the most economical current density . . . . .	23
Exa 4.1	To calculate the maximum stress on the insulation . . . . .	25
Exa 4.2	To calculate the overall diameter and its most economical diameter of a single core cable .	25
Exa 4.3	To calculate potential gradient at the surface of teh conductor . . . . .	26
Exa 4.4	To calculate the minimum diameter of the lead sheath . . . . .	27
Exa 4.5	To calculate the diameter of the inter sheath and the voltage . . . . .	28
Exa 4.6	capacitance between any pair of conductors and the charging current . . . . .	29
Exa 4.7	To calculate charging kvar . . . . .	29
Exa 4.8	To calculate restitivity of the insulating material . . . . .	30
Exa 4.9	To calculate capacitance charging current ic reactive var dielectric loss and equivalent insulation resistance . . . . .	30
Exa 5.1	To calculate the voltage distribution across each unit and the string efficiency . . . . .	32
Exa 5.2	To calculate voltage across the lowest unt and the string efficiency . . . . .	33
Exa 5.3	to find the voltage distribution and the string efficiency . . . . .	33
Exa 5.6	to find the voltage between the conductors and the string efficiency . . . . .	34
Exa 5.7	to calculate the voltage across each unit as a percentage of the total voltage . . . . .	35
Exa 6.1	To calculate the sag . . . . .	36

Exa 6.2	to calculate the deflected sag and the vertical component of sag . . . . .	36
Exa 6.3	determine tension and sag . . . . .	37
Exa 6.4	to calculate the temperature at which sag will remain same under conditions of no ice and no wind . . . . .	39
Exa 6.5	to find the clearance between the conductor and the water at a point midpoint between the towers . . . . .	40
Exa 7.1	To calculate the inductance of each inductor inductive reactance per km and loop inductance per km . . . . .	41
Exa 7.2	To calculate inductance and inductive reactance per phase per km of the line . . . . .	42
Exa 7.3	To calculate the inductive reactance per phase per km of the line . . . . .	43
Exa 7.4	To calculate the effective inductance of the line . . . . .	43
Exa 7.5	To calculate the inductance of the line per km	44
Exa 7.6	To calculate the inductive reactance per phase per km of the system . . . . .	44
Exa 7.7	determine the line inductance and the inductive reactance . . . . .	45
Exa 7.8	To calculate the capacitance of each conductor to neutral per km and line to line capacitance and capacitance susceptance to neutral per km . . . . .	46
Exa 7.9	To calculate the charging current per km and the reactive voltamperes . . . . .	47
Exa 7.10	to find the capacitance per km to neutral and the capacitive reactance per phase per km . . . . .	47
Exa 7.11	To calculate the capacitive reactance per phase per km of the line . . . . .	48
Exa 7.12	To calculate the capacitance of the line . . . . .	48
Exa 7.13	To calculate the capacitance per km to neutral and the capacitive reactance to neutral per km to the line . . . . .	49

Exa 7.14	To calculate the inductive reactance and the capacitive reactance per phase per km . . . .	50
Exa 7.15	To calculate the voltage induced per km . . .	51
Exa 7.16	to calculate the voltage induced in telephone conductor due to electrostatic effect . . . . .	52
Exa 7.17	To calculate the inductance per unit length	53
Exa 7.18	To calculate the loop inductance of the line	54
Exa 8.1	To calculate the per unit impedance and admittance . . . . .	56
Exa 8.2	Three phase apparent power in pu . . . . .	56
Exa 8.3	determine the total reactance in per unit . .	57
Exa 8.4	to calculate per unit impedances at 15 kva base . . . . .	58
Exa 8.5	To calculate the voltage drop in line per unit	59
Exa 8.6	To calculate the through impedance . . . . .	59
Exa 8.7	to determine the per phase generator voltage	62
Exa 8.8	to calculate the per unit impedance of all units	63
Exa 8.9	To calculate the voltage at the terminals of the motor . . . . .	64
Exa 8.10	To find the generator bus terminal voltage .	65
Exa 9.1	calculate the current voltage and power factor of the load . . . . .	68
Exa 9.2	calculate power factor voltage regulation and efficiency . . . . .	69
Exa 9.3	calculate voltage and power factor . . . . .	70
Exa 9.4	determine the voltage and power factor . . .	71
Exa 9.5	the voltage at the generator busbars . . . . .	72
Exa 9.6	determine voltage current power factor apparent power efficiency and regulation of line .	73
Exa 9.7	calculate receiving end load . . . . .	77
Exa 9.8	to calculate A B C D constants sending end voltage current power factor and efficiency .	77
Exa 10.1	To calculate the A B C D constants . . . . .	80
Exa 10.2	To calculate the sending end voltage sending end current line charging current efficiency of transmission and voltage regulation . . . . .	81
Exa 11.2	To find the voltage current and pf at the sending end . . . . .	83

Exa 11.3	To calculate sending end voltage . . . . .	84
Exa 11.4	To calculate the value of ABCD parameters and the characteristic impedance . . . . .	86
Exa 11.5	To determine the equivalent T network . . .	87
Exa 13.1	To calculate the maximum power phase dif- fereneec for maximum power transmitted and the rating of synchronous phase modifier . . .	89
Exa 13.2	To find the rating of the modifier and the power factor . . . . .	90
Exa 13.3	To calculate the sending voltage maximum power and additional reactive power . . . . .	91
Exa 13.4	To find the mva rating of synchronous phase modifier . . . . .	93
Exa 14.1	determine Ybus . . . . .	96
Exa 14.2	determine modified Ybus . . . . .	97
Exa 14.3	determine Ybus . . . . .	98
Exa 14.4	determine Ybus . . . . .	98
Exa 14.5	determine modified Ybus . . . . .	99
Exa 14.6	determine Ybus . . . . .	100
Exa 14.7	determine modified Ybus . . . . .	101
Exa 14.8	determine the voltages at buses . . . . .	103
Exa 15.1	to find the incremental cost of two units . . .	106
Exa 15.2	to find the load division between the two units	106
Exa 15.3	determine the saving in fuel cost . . . . .	107
Exa 15.4	find the loss coefficients and the transmission loss . . . . .	108
Exa 15.5	find the penalty factor . . . . .	109
Exa 15.6	find the penalty factor and the additional cost per hour to increase the output of plant 1 by 1 MW . . . . .	109
Exa 15.7	calculate the penalty factors for the two plants	110
Exa 15.8	determine the generation schedule and the load demand . . . . .	110
Exa 16.1	To find the high voltage terminals of a trans- former . . . . .	112
Exa 16.2	to calculate the subtransient transient and synchronous short circuit currents . . . . .	113

Exa 16.3	To calculate the reactance of the reactor to prevent the circuit breakers from overloading . . . . .	115
Exa 16.4	To find the reactor necessary to prevent the switchgear . . . . .	116
Exa 16.5	to find the reactor reactances . . . . .	116
Exa 16.6	to determine short circuit MVA and fault current distribution . . . . .	117
Exa 17.1	To calculate the positive negative zero sequence component of currents and return neutral current . . . . .	119
Exa 17.2	To determine the symmetrical components of the 3 phase system . . . . .	120
Exa 17.3	To find the symmetrical components of the line current . . . . .	122
Exa 17.7	To determine the complex power represented by three phase voltages and three phase currents by symmetrical and unbalanced components . . . . .	124
Exa 18.1	To determine the fault current and the line to line voltages . . . . .	127
Exa 18.2	to find the sequence LLG and LL . . . . .	129
Exa 18.3	To calculate voltage to neutral of the faulty phase . . . . .	131
Exa 19.1	To find the steady state stability limit . . . . .	133
Exa 19.2	To determine the steady state stability limit . . . . .	134
Exa 19.4	To calculate the kinetic energy . . . . .	137
Exa 19.5	To calculate equivalent h constant at a base of 100MVA . . . . .	137
Exa 19.6	To calculate whether the generator will remain in synchronism . . . . .	138
Exa 19.7	To estimate the sudden increase in generator output . . . . .	139
Exa 19.8	To determine the stability of the system . . . . .	139
Exa 19.9	To determine the critical clearing angle . . . . .	140
Exa 19.10	To determine the critical clearing angle . . . . .	141
Exa 19.11	To plot the swing curve . . . . .	142
Exa 20.1	to calculate the surge impedances and velocities of the line and cable . . . . .	144

Exa 20.2	To find the voltage distribution . . . . .	146
Exa 20.3	to find the reflected voltage and current in the cable . . . . .	146
Exa 20.4	To determine from first principles the magni- tudes of first and second pulses . . . . .	147
Exa 20.5	To find the surge voltage distribution . . . . .	148
Exa 20.6	To calculate the current voltage waves and the resistance reflected from and transmitted beyond the junction . . . . .	148
Exa 20.7	To calculate the voltage across load . . . . .	149
Exa 20.8	To find the maximum voltage . . . . .	150
Exa 22.1	to determine the disruptive critical voltage visual critical voltages for local and general corona . . . . .	151
Exa 22.2	to estimate the corona loss . . . . .	152
Exa 24.1	to calculate reactance to neutralize the capac- itance to earth . . . . .	153
Exa 24.2	To determine the inductance and kva rating	153
Exa 25.1	To determine the overall cost per kwh . . . . .	155
Exa 25.2	To calculate the annual bill . . . . .	155
Exa 25.3	To determine the cost of energy per kwh at the busbars . . . . .	156
Exa 25.4	to determine the energy consumption per year and the yearly bill . . . . .	156
Exa 25.5	To determine the saving in energy cost . . . . .	157
Exa 25.6	To determine the total energy consumed and the total bill . . . . .	158
Exa 25.7	To determine which tariff is economical . . . . .	158
Exa 25.8	to calculate the annual bill . . . . .	159
Exa 25.9	To find the number of units for which the tar- iff is economical . . . . .	160
Exa 25.10	to find the energy consumption per year . . . . .	161
Exa 25.11	to find the generation cost per kwh . . . . .	161
Exa 26.1	To calculate the value of capacitance . . . . .	163
Exa 26.2	To calculate the capacitance kVAr and the new supply of current . . . . .	163
Exa 26.3	to calculate the kVAr rating and capacitance per phase . . . . .	164

Exa 26.4	To calculate the capacitance per phase . . .	164
Exa 26.5	To calculate current to the motor at full load current through capacitor and capacitance .	165
Exa 26.6	To calculate the capacitance per phase . . .	166
Exa 26.7	To calculate the leading kvar kva rating and power factor . . . . .	166
Exa 26.8	To calculate power factor leading kvar and kva	167
Exa 26.9	To calculate the total load kVA and power factor . . . . .	168
Exa 26.10	To calculate the annual saving . . . . .	169
Exa 26.11	To determine the most economical pf and the kva rating . . . . .	169
Exa 26.12	To determine the annual savings . . . . .	170
Exa 26.13	To estimate the limiting cost per kvar . . .	170

# Chapter 1

## Load Characteristics

Scilab code Exa 1.1 to calculate the average load monthly energy consumption and 1

```
1 clear;
2 clc;
3 nos_lmcs=8;
4 pow_of_lmp=60;           //power in watts
5 dur_per_day=5;         //duration in
   hours
6 t_dur=24;              //duration in
   hours
7 max_dem=1500;         //demand in watts
8
9 nos_heaters=2;
10 pow_of_heaters=1000;   //power in watts
11 dur_per_day_heater=3;  //duration in hours
12
13 enrgy_lmp=(nos_lmcs*pow_of_lmp*dur_per_day); //
   energy in watt hour
14 energy_heater=(nos_heaters*pow_of_heaters*
   dur_per_day_heater); //energy in watt hour
15
16 act_ene_con=(enrgy_lmp + energy_heater); //
   energy in watt hour
```

```

17
18 avg_load=(act_ene_con/t_dur);           //
    avverage load in watt
19 printf("\n The average load of the consumer is: %.2
    f W",avg_load);
20 mon_ene_con=(act_ene_con*30);           //
    monthly energy consumption in wtt hour
21 printf("\n The monthly energy consumption of the
    consumer is: %.0f kWh",mon_ene_con/1000);
22
23 load_fac=(avg_load/max_dem);           //
    load factor is unitless
24 printf("\n The load factor of the consumer is: %.4f
    ",load_fac);

```

---

**Scilab code Exa 1.2** to calculate the diversity factor avg load and laod factor of

```

1 //to calculate the a).diversity factor b).avg load
    and laod factor of each consumer c).avg load and
    load factor of combined load.
2 //postfix 1,2...4 shows the consumers.
3
4 clc;
5 //for consumer1
6 avg_load1=1;           //load in kwatt
7 max_dem1=5;           //demand in kwatt
8 dem1_at8=5;           //demand in kwatt
9 l_f1=(avg_load1/max_dem1); //load factor is
    unitless
10 printf("\n the load factor for consumer 1 is: %.2f
    \n",l_f1);
11
12 //for consumer2
13 max_dem2=2;           //demand in kwatt
14 dem2_at8=1.6;           //demand in kwatt

```

```

15 l_f2=0.15; //load factor is unitless
16 avg_load2=(l_f2*max_dem2); //load in kwatt
17 printf("\n the avg load for consumer 2 is: %.2f \n"
    ,avg_load2);
18
19 //for consumer3
20 avg_load3=0.5; //load in kwatt
21 max_dem3=2; //demand in kwatt
22 dem3_at8=1; //demand in kwatt
23 l_f3=(avg_load3/max_dem3); //load factor is
    unitless
24 printf("\n the load factor for consumer 3 is: %.2f
    \n",l_f3);
25
26 //for consumer4
27 max_dem4=10; //demand in kwatt
28 dem4_at8=5; //demand in kwatt
29 l_f4=0.25; //load factor is unitless
30 avg_load4=(l_f4*max_dem4); //load in kwatt
31 printf("\n the avg load for consumer 4 is: %.2f \n"
    ,avg_load4);
32
33 tot_avg_load=(avg_load1 + avg_load2 + avg_load3 +
    avg_load4); //load in kwatt
34 printf("\n the total avg load is: %.2f \n",
    tot_avg_load);
35
36 tot_max_dem=(max_dem1 + max_dem2 + max_dem3 +
    max_dem4); //demand in kwatt
37 printf("\n the total maximum demand is: %.2f\n ",
    tot_max_dem);
38
39 tot_dem_at8=(dem1_at8 + dem2_at8 + dem3_at8 +
    dem4_at8); //demand in kwatt
40 printf("\n the total demand at 8pm is: %.2f\n ",
    tot_dem_at8);
41
42 div_fact=(tot_max_dem/tot_dem_at8);

```

```

//diversity factor is unitless
43 printf("\n the diversity factor is: %.2f\n ",
    div_fact);
44
45 load_factor=(tot_avg_load/tot_dem_at8);
    //load factor is unitless
46 printf("\n the load factor is: %.2f\n ",load_factor
    );

```

---

**Scilab code Exa 1.3** to find the maximum value and the consumption of energy in kWh

```

1 clear;
2 clc;
3
4 ene_con=600;
5 lf=.45;
6 t=24;
7 max_dem=ene_con/(t*lf);
8 printf("The maximum demand is: %.2f kW\n",max_dem);
9
10 lf=.65;
11 ene_con=lf*max_dem*t;
12 printf("The energy consumed in 24h is: %.2f kWh",
    ene_con);

```

---

**Scilab code Exa 1.4** to plot the load duration curve from the chronological load cu

```

1 clear;
2 clc;
3
4 x=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
    20 21 22 23 24 ]; //dividing the x axis in 24
    hours

```

```

5
6 y=[30 30 30 30 20 20 20 20 20 8 8 8 8 8 8 8 8 8 8 8
      8 8 8 5]; //load in MW values
7
8 bar(x,y,1,'blue'); //plotting
      the bargraph with a width of 1
9 xlabel('time in hours');
10 ylabel('load in MW');
11 title('LOAD DURATION CURVE');

```

---

Scilab code Exa 1.5 to determine the load factor from the load duration curve

```

1 clear;
2 clc;
3 l1=15; //load in MW
4 l2=10; //load in MW
5 l3=5; //load in MW
6
7 t1=8; //time in hours
8 t2=8; //time in hours
9 t3=8; //time in hours
10 t=24; //time in hours
11
12 max_dem=max(l1,l2,l3); //load in MW
13 printf("\n the maximum demand is: %.2f MW\n ",
      max_dem);
14
15 act_ener_consum=(l1*t1 + l2*t2 + l3*t3); //energy
      consumed in MWH
16 printf("\n the actual energy consumed is: %.2f MWH\n
      n ",act_ener_consum);
17
18 avg_load=(act_ener_consum/t); //load in MW
19 printf("\n the average demand is: %.2f MW\n ",
      avg_load);

```

```

20
21 l_f=(avg_load/max_dem);           //load factor is
    unitless
22 printf("\n the load factor is:  %.2f\n ",l_f);

```

---

Scilab code Exa 1.6 to calculate plant capacity factor load factor utilization factor

```

1  clear;
2  clc;
3  max_dem=500;           //demand is in
    MW
4  min_dem=400;           //demand is in MW
5  cap_plnt=750;         //capacity of the
    plant is MW
6
7  avg_ann_load=(max_dem + min_dem)/2;   //load is in
    MW
8  printf("\n the average annual load is:  %f MW\n ",
    avg_ann_load);
9
10 cap_f=(avg_ann_load/cap_plnt);        //capacity
    factoe is dimentionless
11 printf("\n the capacity factor is:  %f\n ",cap_f);
12
13 l_f=(avg_ann_load/max_dem);           //load factor is
    dimentionless
14 printf("\n the load factor is:  %f\n ",l_f);
15
16 u_f=(max_dem/cap_plnt);               //utilization
    factor is dimetionless
17 printf("\n the utilization factor is:  %f\n ",u_f);

```

---

Scilab code Exa 1.7 to plot the chronological load duration load energy curve and

```

1 // to plot the chronological ,load duration, load
  energy curve and then
2 //calculating the load factor and the utilization
  factor
3
4 clc;
5 clf();
6 inst_cap=100;           //installed capacity in MW
7 max_dem=70;           //demand in MW
8 t1=6;                 //time in hours
9 t2=9.5;
10 t3=0.5;
11 t4=2;
12 t5=6;
13 d1=70;               //demand in MW
14 d2=50;
15 d3=40;
16 d4=200;
17 d5=10;
18
19 scf(0);
20 x=[0 1 2 3 4 5 6 7 8 9 10 11 12 12.5 13 14 15 16 17
    18 19 20 21 22 23 24];
21 //shows the time from 12pm to 12 am
22 //0 shows 12am and 24 shows 12 pm
23
24 y=[20 20 10 10 10 10 10 10 50 50 50 50 40 40 50 50
    50 50 50 50 70 70 70 70 70 70];
25 //loads in MW
26
27 bar(x,y,0.5,'blue');
28 xlabel('TIME');           //this is time
29 ylabel('LOAD IN MW');
30 title('CHRONOLOGICAL LOAD CURVE');
31
32
33 scf(1);
34 x=[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

```

```

        20 21 22 23 24];
35 //time in hours
36
37 y=[70 70 70 70 70 70 70 50 50 50 50 50 50 50 50
    40 20 20 10 10 10 10 10 10];
38 //loads in MW
39
40 bar(x,y,1,'red');
41 xlabel('TIME IN HOURS');
42 ylabel('LOAD IN MW');
43 title('LOAD DURATION CURVE');
44
45 ene=((d1*t1) + (d2*t2) + (d3*t3) + (d4*t4) + (5*t5))
    ; //energy in MWH
46
47 d_lf=(ene/(max_dem*24)); //
    demand factor is dimensionless
48 printf("\n the demand factor is: %f\n ",d_lf);
49
50 u_f=(max_dem/inst_cap); //
    utilization factor is dimensionless
51 printf("\n the utilization factor is: %f\n ",u_f);
52
53 scf(2);
54 x=[0,10,20,40,50,70];
55 y=[0,240,420,740,895,1015];
56 plot(x,y);
57 xlabel('LOAD IN MW');
58 ylabel('ENERGY IN MWH');
59 title('LOAD ENERGY CURVE');

```

---

Scilab code Exa 1.8 to find the diversity factor of a power station

```

1 //to find the diversity factor of a power station
2

```

```

3  clc;
4  maxa=150;
5  maxb=50;
6  maxc=55;
7  c_max=205;
8
9  tot_max=(maxa + maxb + maxc);
10 printf("\n the maximum demand is:  %.2f kW\n ",
        tot_max);
11
12 d_f=(tot_max/c_max);
13 printf("\n the diversity factor is:  %.2f\n ",d_f);

```

---

Scilab code Exa 1.9 to calculate max demand on the station installed capacity ener

```

1  //to calculate a).max demand on the station  b).
    installed capacity  c).energy supplied in a year
2
3  clc;
4  peak_l1=25;           //load is in MW
5  peak_l2=20;           //load is in MW
6  peak_l3=30;           //load is in MW
7  ann_lf=0.6;           //load factor is dimensionless
8  d_f=1.65;             //diversity factor is
    dimensionless
9
10 max_dem=(peak_l1 + peak_l2 + peak_l3)/d_f;
11 printf("\n the maximum demand is:  %.2f MW\n ",
        max_dem); //demand in  MW
12
13 i_cap=(peak_l1 + peak_l2 + peak_l3); //capacity
    in MW
14 printf("\n the installed capacity is:  %.2f MW\n ",
        i_cap);
15

```

```
16 avg_l=(ann_lf*max_dem);
17 esp_year=(avg_l)*8760;           //energy
    supplies per year in MWH
18 printf("\n the energy supplied in a year is:  %.2f
    MWH\n ",esp_year);
```

---

# Chapter 2

## Supply Systems

Scilab code Exa 2.1 To calculate the percentage saving in conductor material

```
1 clear;
2 clc;
3 // i=p/v;
4 // a=i/alpha;
5 // vol=2*l*a;
6
7 // calculating itf or the 2 supplies 250V and 400V
8 // a=v2/v1;
9 a=.625;
10 sav=(1-a)*100;
11 printf("the percentge saving in the conductor
    material is :%.2f per cent",sav);
```

---

Scilab code Exa 2.2 To compare the amount of conductor material required

```
1 clear;
2 clc;
3
```

```

4 //i1=p/v
5 //line loss=2i^2*r
6 //vol of conductor required v1=2*l*a1
7
8 a=1/6; //a=r1/r2
9 b=1/a //b=a2/a2
10
11 //v2=3*l*a2+ 0.5*l*a2
12 //v2/v1=3.5*l*a2/2*l*a1
13
14 c=3.5/(2*b); //c=v2/v1
15 printf("the ratio of the volumes of the conductors
    is :%.2f",c);

```

---

**Scilab code Exa 2.3** To calculate the volume of conductor

```

1 clear;
2 clc;
3
4 s=5*10^6;
5 l=50*10^3;
6 pf=0.8;
7 eff=.9;
8 v=33*10^3;
9 rho=2.85*10^(-8);
10 pl=0.1*s*pf;
11 i=s/v;
12 a1=2*i*i*rho*l/pl;
13 vol=2*l*a1;
14 printf("the volume of the conductor required is :%.2f
    cubic meter",vol);
15
16 //b)
17
18 il=s/(sqrt(3)*v);

```

```

19 a2=3*i1*i1*rho*l/p1;
20
21 vol=3*l*a2
22 printf("\n the volume of the conductor required is:%.2f cubic meter",vol);

```

---

**Scilab code Exa 2.4** To calculate the percentage of additional load

```

1 clear;
2 clc;
3
4 //a). single phase supply
5 //p1=v1*i1*pf
6 //line loss= 2*i1*i1*r
7 //percentage line loss=line loss*100/(v8i*pf)
8
9 //b).three phase supply
10 //p3=sqrt(3)*v*i3*pf
11 //percentage line loss=3*i3*i3*r*100/p3
12
13 //ratio of load transmitted
14
15 a=2; //a=p3/p1
16 add_load=(a-1)*100; //(p3-p1)/p1
17
18 printf("the percentage of additional load is:%.2f",
    add_load);

```

---

**Scilab code Exa 2.5** To compare the diameter and weight

```

1 clear;
2 clc;
3

```

```
4 rhoa=2.85
5 rhoc=1.70;
6 sig1=2.71;
7 sig2=8.89;
8
9 a=sqrt(rhoa/rhoc); //a=diameter of aluminium/
    diameter of copper
10 printf("the ratio of diameters is:%.2f\n",a);
11 b=sig1*rhoa/(sig2*rhoc); //weight of aluminium/
    weight of copper
12
13 printf("the ratio of weights is:%.2f",b);
```

---

# Chapter 3

## Conductors

Scilab code Exa 3.1 To determine the most economical size

```
1 clear;
2 clc;
3 ld=100;           //load in KW
4
5
6 v=500;.....// voltage in volts
7 res=1.75*(.000001);.....//restivity in milli
   ohm per cm sq
8 nos_cores=2;.....//number of feeder core
9 l=0.8;.....//length of tx line in km
10 area=1;.....//area in cm sq
11 cost_of_energy=0.12;..//cost in Rs per unit
12 dep=0.1.....//depreciation percentage is 10
   %
13
14 flc=(ld*1000)/500;..//full load current
15
16 ra=(res*l*1000*ld);..//resistance* area in ohm-m
17
18 pow_loss=2*(flc*flc*ra*.001);...//(power loss*area)
   for the two cores in the cable
```

```

19
20 ann_en_loss=pow_loss*365*24;...//annual energy loss
    in KWH
21 cost=ann_en_loss*cost_of_energy;...//(cost*area) of
    annual energy loss in Rs
22
23 ann_dep=6*1*1000*dep;.....//(ann_dep*area) in Rs
24
25 c=(cost/ann_dep);
26
27 area=sqrt(c);.....//area in cm sq
28
29 d=(area*4/(%pi));
30 dia=sqrt(d);.....//diameter in cm
31 printf("\n the most economical size is: %.2f cm\n "
    ,dia);

```

---

Scilab code Exa 3.2 to calculate the most economical cross sectional area

```

1 clear;
2 //clc();
3 l=10;.....//length of tx line in km
4
5 s1=2500;....// power in KVA
6 s2=2000;....// power in KVA
7 s3=1500;....// power in KVA
8
9 t1=8;.....//time in hrs
10 t2=9;.....//time in hrs
11 t3=7;.....//time in hrs
12
13 dep_rate=0.08;...//depreciation rate
14 cost=0.15;.....//cost of energy in Rs
15 res=0.173.....//resistance per sq cm
16

```

```

17 nos_wd=250;.....//number of working days
18
19 ann_int=6000*1*dep_rate;..//annual interst and
    depreciation on capitol cost in Rs
20
21 r=res*1;..//(resistance*area) of each conductor in
    ohms
22
23 c=sqrt(3);
24 d=c*33;.....//kv of tx line
25 i1=(s1/d);.....//load current at 2500KVA in A
26
27 i2=(s2/d);.....//load current at 2000KVA in A
28
29 i3=(s3/d);.....//load current at 1500KVA in A
30
31 d_en_loss=(3*r/1000)*(t1*i1*i1 + t2*i2*i2 + t3*i3*i3
    );..//daily energy loss in KWH
32
33 ann_cost=d_en_loss*nos_wd;..//annual enrgy loss in
    KWH
34
35 cost_el=ann_cost*cost;..//(cost of energy loss per
    annum)*area
36
37 e=cost_el/ann_int;..//by kelvin's law area
38 f=sqrt(e);
39 printf("\n the most economical area is: %.2f sq cm\
    n ",f);..//area in cm sq

```

---

Scilab code Exa 3.3 to calculate the best current density of a three phase oh line

```

1 clear;
2 //clc();
3 a=2500;..//usage of the over head line in a year in

```

```

    hrs
4  cost=20;..//cost of copper per kgf in Rs
5  dep_rate=0.125;..//depriciation rate
6  den=8.89;..//density of copper in gf per cm sq
7  r=0.173;...//resistance per conductor per km length
    and per sq cm in ohms
8  cost_en=0.16;..//cost of energy perm unit in Rs
9
10 cap_cost=cost*den/100;..//(capitol cost of conductor
    )*length/area in Rs
11 dep=cap_cost*dep_rate;
12
13 cl=r/10000000;..//copper loss per conductor/current
    sq*length/area
14
15 ce=a*cost_en*cl;..//cost of energy loss per year)*
    area/(length*current sq)
16
17 d=dep/ce;
18
19 j=sqrt(d);..//best current density in A/cm sq
20 printf("\n the current density is: %.2f A/sq meter\n
    ",j);

```

---

**Scilab code Exa 3.4** To calculate rms value of curent for a 3 phase

```

1  clear;
2  //clc();
3  s=7500;..//maximum load in KVA
4  v=33;..//voltage of tx line in volts
5
6  a=sqrt(3);
7  im=s/(v*a);..//maximum current in A
8
9  kf=1.3;..//for a 0.4 load factor

```

```
10 irms=kf*0.4*im;
11 printf("The rms value of current is:%.2f A",irms);
```

---

**Scilab code Exa 3.5** To determine the most economical size of copper conductor

```
1 clear;
2 clc;
3 funcprot(0);
4 s=7500;
5 v=33;
6 lf=0.6;
7 kf=1.13;
8 cost_en=0.15;
9 dep_rate=0.1;
10 res=17.6;
11 im=s/(v*sqrt(3));
12 irms=0.6*kf*im;
13 rho=(17.6*10^4)/(10^5);
14 el=3*irms*irms*.1*rho*8760/1000;
15 ann_w=el*cost_en;
16 ann_dep=dep_rate*20000;
17 c=ann_w/ann_dep;
18 area=sqrt(c);
19 printf("The area is:%.2f sq cm",area);
```

---

**Scilab code Exa 3.6** To calculate the most economical current density

```
1 clear;
2 //clc();
3 lc=0.5;
4 dep=0.09;
5 cost_en=0.12;
6 res=1.76;
```

```
7 l=1;
8
9 ann_cost=dep*25000;
10 lf=0.2*lc + 0.8*(lc^2);
11 al=3*.1*res*8760*lf/1000;
12 cost=al*cost_en;
13 d=ann_cost/cost;
14 j=sqrt(d);
15 printf("The current density is :%.2f A/sq cm",j);
```

---

# Chapter 4

## Power Cables

Scilab code Exa 4.1 To calculate the maximum stress on the insulation

```
1 //to calculate the maximum electric stress on the
   insulation
2
3 //clc();
4 v=6.5;..//working voltage in volts
5 d=10;..//diameter of conductor in mm
6 t=7.5;..//thickness of insulation in mm
7 r=d/2;..//radius of conductor in mm
8 R=(r+t);
9 a=R/r;
10 b=log([a]);
11
12 gmax=v/(r*b);
13 printf("\n the maximum electric stress on the
   insulation is: %.2f kV/mm\n",gmax);
```

---

Scilab code Exa 4.2 To calculate the overall diameter and its most economical diam

```

1 clear;
2 //clc();
3 v=275;..//working voltage in volts
4 vrms=v/sqrt(3);..//effective value of phase voltage
5 pv=vrms*sqrt(2);..//peak value of phase voltage
6 gmax=15;..//maximum permissible stress in dielectric
7
8 r=pv/gmax;
9
10 d=2*r;..//economical core diameter in mm
11 printf("\n the economical core diameter is:  %.2fmm\n
        n",d);
12
13 R=exp(1)*r;
14
15 D=2*R;..//inner diameter of the sheath
16 printf("\n the inner diameter of the sheath is:  %.2
        fmm\n",D);

```

---

Scilab code Exa 4.3 To calculate potential gradient at the surface of teh conducto

```

1 clear;
2 //clc();
3
4 d=10;..//diameter of the conductor in mm
5 r=d/2;..//radius in mm
6 t=10;..//thickness in mm
7 r1=(r+t);
8 R=(r+ 2*t);
9 e1=3;
10 e2=2.5;
11 v=60;..//voltage in kv
12
13 a=r1/r;
14 b=R/r1;

```

```

15 c=e1/e2;
16
17 d=(r)*(log([a]) + c*log([b]));
18
19 gmax1=v/d;
20
21 printf("\n the potencial gradient at the surface of
    the conductor is :%.2f kV/mm\n",gmax1);

```

---

Scilab code Exa 4.4 To calculate the minimum diameter of the lead sheath

```

1 clear;
2 //clc();
3
4 v=66;
5 r=10;
6 e1=5;
7 e2=4;
8 e3=3;
9 gmax1=3.8;
10
11 gmax2=2.6;
12
13 gmax3=2;
14
15 r1=e1*r*gmax1/(e2*gmax2);
16
17 r2=e1*r*gmax1/(e3*gmax3);
18
19 a=r1/r;
20 b=r2/r1;
21
22 v1=gmax1*r*log([a]);
23
24 v2=gmax2*r1*log([b]);

```

```

25
26 c=(v-v1-v2)/(gmax3*r2);
27 e=exp(c);
28
29 R=e*r2;
30
31 dia=2*R;
32 printf("\n the minimum diameter is:  %.2f mm\n ",dia
    );

```

---

**Scilab code Exa 4.5** To calculate the diameter of the inter sheath and the voltage

```

1 clear;
2 //clc();
3
4 v=60;
5 gmax=4;
6 v1=v/exp(1);
7 r=v1/gmax;
8 d=2*r;
9 r1=v/gmax;
10 d1=2*r1;
11 v2=v-v1;
12 R=1.88*r1;
13 D=2*R;
14 printf("\n the diameter of cable is:  %f cm\n ",D);
15
16 //cable without sheath
17
18 x=exp(1);
19 a=log([x]);
20 r1=v/gmax;
21 d1=2*r1;
22
23 R1=exp(1)*r1;

```

```

24
25 D1=2*R1;
26 printf("\n the diameter of cable without sheath is:
        %.2f mm\n ",D1);

```

---

Scilab code Exa 4.6 capacitance between any pair of conductors and the charging cu

```

1 clear;
2 //clc();
3 v=11;
4 ct=0.7;
5 cs=0.4;
6 cc=(ct-cs)/2;
7 printf("\n the capacitance between conductors is: %
        .2f uF\n ",cc);
8
9 cl=0.5*(3*cc + cs);
10
11 ic=(v*2*3.14*50*2*cl*.001)/sqrt(3);..//charging
        current in ka/phase
12 printf("\n the charging current is: %.3f A\n ",ic);

```

---

Scilab code Exa 4.7 To calculate charging kvar

```

1 clear;
2 //clc();
3
4 v=11;
5 cl=0.3;
6 l=5;
7 c0=2*cl*l;
8
9 tot_var=(v*v*2*3.14*50*c0*.01);

```

```
10 printf("\n the charging kvar is :%.2f KVAR\n ",
    tot_var);
```

---

Scilab code Exa 4.8 To calculate restitivity of the insulating material

```
1 clear;
2 //clc();
3
4 r=0.4;
5 l=5;
6 d=20;
7 D=50;
8 a=D/d;
9 b=log([a]);
10
11 rho=2*3.14*l*r/b;..//resistivity in mega-ohm-m
12 printf("\n the resistivity is :%.2f M-Ohm-M\n ",rho);
```

---

Scilab code Exa 4.9 To calculate capacitance charging current ic reactive var diel

```
1 clear;
2 //clc();
3
4 v=11;
5 f=50;
6 d=20;
7 r=15;
8 er=2.4;
9 la=0.031;
10 l=2.5;
11
12 a=r*2/d;
13 b=log([a]);
```

```

14
15 cap=er/(18*b);
16 tot_cap=cap*l;
17 printf("\n the total capacitance is:  %.2f F\n ",
    tot_cap);
18
19 //ic
20
21 ic=2*3.14*f*tot_cap*v;
22 printf("\n the charging current is:  %.2f A\n ",ic
    /1000);
23
24 //reactive var
25 var=v*ic;
26 printf("\n the reactive var is:  %.2f kVAR\n ",var
    /1000);
27
28 //dielectric loss
29
30 pd=v*ic*la;
31 printf("\n the dielectric loss is:  %.2f W\n ",pd);
32
33 ri=v*v/pd;..//resistance in mega-ohm
34 printf("\n the resistance is:  %.2f M-OHM\n ",ri);

```

---

# Chapter 5

## Line insulators and supports

Scilab code Exa 5.1 To calculate the voltage distribution across each unit and the

```
1 clear;
2 //clc();
3 V=100;..//voltage between line conductor and earth
4 k1=0.1;
5 n=4;
6 k=sqrt(k1);
7
8 V1=(V*sinh(k)/sinh(n*k));
9 printf("\n the voltage1 is: %.2f V\n ",V1);
10
11 V2=(V*sinh(2*k)/sinh(n*k));
12 printf("\n the volatge2 is: %.2f V\n ",V2);
13
14 V3=(V*sinh(3*k)/sinh(n*k));
15 printf("\n the voltage3 is: %.2f V\n ",V3);
16
17 V4=V;
18 printf("\n the voltage4 is: %.2f V\n ",V4);
19
20 v1=V1;
21
```

```

22 v2=V2-V1;
23
24 v3=V3-V2;
25
26 v4=V4-V3;
27
28 eff=(sinh(n*k)/(n*(sinh(n*k)-sinh((n-1)*k))));
29 printf("\n the string efficiency is: %.2f percent\n
      ",eff*100);

```

---

Scilab code Exa 5.2 To calculate voltage across the lowest unit and the string effi

```

1 clear;
2 //clc();
3 n=6;
4 k1=0.1;
5 k=sqrt(k1);
6 a=sinh((n-1)*k)/sinh(n*k);
7 b=1-a;
8 printf("\n the voltage across lowest unit is: %.2f
      V\n ",b*100);
9 eff=1/(n*b);
10 printf("\n the string efficiency is: %.2f percent",
      eff*100);

```

---

Scilab code Exa 5.3 to find the voltage distribution and the string efficiency

```

1 clear;
2 clc;
3
4 n=3;
5 c=[1.6 0 -1;1 0 0 ;1 1 1]; //coeffient matrix
6 b=[0.109;.31;1];

```

```

7
8 v=inv(c)*b;
9 for i=1:1:3
10     printf("The voltage across unit%d is: %.2f pu\n",
            i,v(i))
11 end
12
13 eff=1/(n*v(n));
14 printf("\n the string efficiency is:  %.2f percent\n
        ",eff*100);
15
16 //with the string grading
17 a=[3.6037 0 0;1.3037 0 -1 ;1 1 1]; //coeffient matrix
18 d=[1.1889;0.0389;1];
19
20 v=inv(a)*d;
21 for i=1:1:3
22     printf("The voltage across unit%d is: %.2f pu\n",
            i,v(i))
23 end
24
25 eff=1/(n*v(n));
26 printf("\n the string efficiency is:  %.2f percent",
        eff*100);
27 //difference in answers is due to miscalculations

```

---

Scilab code Exa 5.6 to find the voltage between the conductors and the string effi

```

1 clear;
2 clc;
3
4 n=4;
5 v2=14.2;
6 v3=20;
7 a=14.2;... //coefficients of the quadratic equation

```

```

14.2*k*k+22.6*k-5.8
8 b=22.6;
9 c=-5.8;
10
11 k=(-b+sqrt(b*b-4*a*c))/(2*a);
12 v1=v2/(1+k);
13 v4=(1+6*k+5*k*k+k*k*k)*v1;
14 V=v1+v2+v3+v4;
15 v1=sqrt(3)*V;
16 eff=V/(n*v4);
17
18 printf("\n the string efficiency is: %.2f percent",
    eff*100);

```

---

Scilab code Exa 5.7 to calculate the voltage across each unit as a percentage of t

```

1 clear;
2 clc;
3
4 //by using kcl at node a
5 //20*v1-18*v2-3*v3=0
6 //by kcl at node b
7 //5*v1+20*v2-18*v3=0
8 //total voltage across the string
9 //v1+v2+v3=V
10 c=[20 -18 -3;5 20 -18 ;1 1 1]; //coeffient matrix
11 b=[0;0;1];
12 v=inv(c)*b;
13 for i=1:1:3
14     printf("The voltage across unit%d is: %.2f
    percent\n",i,v(i)*100)
15 end

```

---

# Chapter 6

## Sag and tension

Scilab code Exa 6.1 To calculate the sag

```
1 clear;
2 clc;
3
4 uts=5758; //ultimate tensile strength
5 l=200;
6 wt=.604;
7
8 h=uts/2;
9 sag=wt*l*l/(8*h);
10
11 printf("The sag is :%.3 f m",sag);
```

---

Scilab code Exa 6.2 to calculate the deflected sag and the vertical component of s

```
1 clear;
2 clc;
3
4 wc=.844;
```

```

5 t=9.53;
6 d=19.53;
7 bl=7950;
8 fs=2;
9 p=40;
10 wi=(%pi)*t*(t+d)*913.5*10^(-6);
11 wh=(d+2*t)*p*10^(-3);
12 wr=sqrt((wc+wi)^2 + wh^2);
13
14 H=(bl/fs); //bl=breaking load fs=factor of
safety
15 l=275;
16 sag=wr*l*l/(8*H);
17 printf("The deflected sag is :%.3f m",sag);
18
19 vsag=(wc+wi)*l*l/(8*H);
20 printf("\n The vertical component of sag is :%.2f m",
vsag);

```

---

Scilab code Exa 6.3 determine tension and sag

```

1 clear;
2 clc;
3
4 uts=14740;
5 a=538.4;
6 E=7000;
7 wc=1.805;
8 p=100;
9 l=335;
10 alpha=19.3*10*(-6);
11 fs=2;
12 n=4;
13 ds=3.35;
14 d=(1+2*n)*ds;

```

```

15 wh=(2/3)*d*p;
16
17 //a).
18 wr=sqrt((wc+wh)^2 + wh^2);
19 T=(uts/fs); //bl=breaking load=uts fs=factor of
safety
20 printf("The maximum working stress is:%.2f kgf",T);
21 sag=wr*l*l/(8*T);
22 printf("\n The deflected sag is:%.3f m",sag/1000);
23
24 //b).
25
26 t1=60;
27 T1=T;
28 w1=wr;
29
30 K=T1-w1*w1*l*l*E*a/(24*T1*T1);
31 b=alpha*(t1-0)*E*a;
32 //by using the formula t2^2(t2-K+b)=w2^2*l*l*e*a/24
and getting a solution by hit trial
33
34 T2=4083;
35 sag_60=wc*l*l/(8*T2);
36 printf("\n The sag is:%.3f m",sag_60);
37
38 //c).
39 fs=4;
40 t2=30;
41 T=(uts/fs); //bl=breaking load=uts fs=factor of
safety
42 K=T1-w1*w1*l*l*E*a/(24*T1*T1);
43 b=alpha*(t1-t2)*E*a;
44
45 //by using the formula t2^2(t2-K+b)=w2^2*l*l*e*a/24
and getting a solution by hit trial
46
47 T2=3132;
48 printf("\n The maximum working stress is:%.2f",T2);

```

```

49 sag_60=wc*l*l/(8*T2);
50 printf("\n The sag is :%.2 f m",sag_60);
51 //similar calculations can be made for other
    temperatures and loading conditions

```

---

Scilab code Exa 6.4 to calculate the temperature at which sag will remain same und

```

1  clear;
2  clc;
3
4  wc=.594;
5  a=64.5;
6  t=9.53;
7  d=3*3.45;
8  sag=3.96;
9  p=40;
10 E=12700;
11 l=160;
12 alpha=1.7*10^(-5);
13 wi=(%pi)*t*(t+d)*913.5*10^(-6);
14 wh=(d+2*t)*p*10^(-3);
15 wr=sqrt((wc+wi)^2 + wh^2);
16 w1=wr;
17 T1=w1*l*l/(sag*8);
18
19 w2=wc;
20 t1=-5.5;
21 T2=wc*T1/w1;
22 //by using the formula  $t_2^2(t_2-K+b)=w_2^2*l*l*e*a/24$ 
23
24 t2=t1+(T1-T2)/(alpha*E*a);
25 printf("The temperature at which the sag will remain
    the same :%.2 f degC",t2);

```

---

Scilab code Exa 6.5 to find the clearance between the conductor and the water at a

```
1 clear;
2 clc;
3
4 l=270;
5 T=1800;
6 w=1;
7 h=90-30;
8 ap=30;
9 x=(l/2)-T*h/(w*l);
10 x1=-x+l/2;
11 sag1=w*x1*x1/(2*T);
12 sag2=w*x*x/(2*T);
13 hob=w*(1-x)^2/(2*T);
14
15 clearance=ap+sag1-sag2;
16 printf("The clearance between the conductor and
    water at point m is: %.2f m", clearance);
```

---

# Chapter 7

## Line Parameters

Scilab code Exa 7.1 To calculate the inductance of each inductor inductive reactance

```
1 clear;
2 //clc();
3 D=3;..//spacing between the conductors in m
4 r=0.01;..//radius of each conductor in m
5 r1=0.7788*r;
6 mu=50;..//relative permeability of steel
7 a=log([D/r1]);
8 b=log([D/r]);
9
10 l=2*10^(-7)*a*(10^6);..//inductance of each
    conductor in henry per km
11 printf("\n the inductive reactance is: %.2f mH/km\n",
    l);
12
13 loop_l=2*l;
14 printf("\n the loop inductance is: %.2f mH/km\n",
    loop_l);
15
16
17
18 react=2*(%pi)*50*loop_l*.001;
```

```

19 printf("\n the inductive reactance is: %.2f Ohm/km
    \n ",react);
20
21 //loop inductance with steel conductors
22
23 l_in=0.5*mu;
24 l_ex=2*b;
25
26 l_inductance=2*(l_in+l_ex)/10;
27 printf("\n the loop inductance is: %.2f mH/km\n ",
    l_inductance);
28
29 2

```

---

Scilab code Exa 7.2 To calculate inductance and inductive reactance per phase per

```

1 clear;
2 //clc();
3 r=10.5;..//radius of the conductor
4 d_ab=3;
5 d_bc=5;
6 d_ca=3.6;
7
8 Deq=(d_ab*d_bc*d_ca)^(1/3);
9
10 r1=0.7788*r/1000;
11
12 a=log([Deq/r1]);
13
14 l=2*10^(-4)*a;
15 printf("\n the inductance is: %.5f H/km\n ",l);
16
17 x1=2*(%pi)*50*l/10000;
18 printf("\n the inductive reactance is: %f Ohm/km\n
    ",x1);

```

---

Scilab code Exa 7.3 To calculate the inductive reactance per phase per km of the 1

```
1 clear;
2 //clc();
3
4 d=6;
5 r=12.5;...//radius of each conductor
6 dm=(d*d*2*d)^(1/3);
7
8 ds1=0.7788*r/1000;
9
10 a=log([dm/ds1]);
11 l=2*a*10^(-7);
12
13 x1=2*(%pi)*l*50*1000;
14 printf("\n the inductance is: %.2f Ohm/km\n",x1);
```

---

Scilab code Exa 7.4 To calculate the effective inductance of the line

```
1 clear;
2 //clc();
3
4 d_ab=3;
5 d_bc=4;
6 d_ca=5;
7 r=0.015;
8
9 d_aa=r*exp(-0.25);
10 d_bb=r*exp(-0.25);
11 d_cc=r*exp(-0.25);
12
```

```

13 la=.2*(log([sqrt(d_ab*d_ca)/d_aa]) + %i*0.866*log([
    d_ab/d_ca]));
14
15 lb=2*(log([sqrt(d_ab*d_bc)/d_bb])/10 + %i*0.866*log
    ([d_bc/d_ab])/10);
16
17 lc=2*(log([sqrt(d_bc*d_ca)/d_cc])/10 + %i*0.866*log
    ([d_ca/d_bc])/10);
18
19 lav=(la +lb + lc)/3;
20 printf("\n the average inductance is:  %f mH/km\n",
    lav);

```

---

**Scilab code Exa 7.5** To calculate the inductance of the line per km

```

1 clear;
2 //clc();
3 D1=1;
4 D2=2;
5 r=0.5*25/1000;
6 a=sqrt(D1*D1+D2*D2)*D2;
7 b=0.7788*r*D1;
8 l=2*log([a/b])/10;
9 printf("\n the inductance is:  %.3f mH/km\n",l);

```

---

**Scilab code Exa 7.6** To calculate the inductive reactance per phase per km of the s

```

1 clear;
2 //clc();
3 r=0.5*3*4.75/1000;
4 d1=3;
5 d2=6;
6 dac1=6;

```

```

7 dbb1=9;
8 dca1=6;
9 dac=6;
10 dc1a1=6;
11
12 dab=sqrt(d1*d1 + (d1/2)*(d1/2));
13 dbc=dab;
14 da1b1=dab;
15 db1c1=dab;
16
17 dab1=sqrt(d1*d1 + (dac+d1/2)*(dac+d1/2));
18
19 dbc1=dab1;
20 dba1=dab1;
21 db1c=dab1;
22 da1b=dab1;
23
24 daa1=sqrt(d2*d2 + d2*d2);
25
26 dcc1=sqrt(d2*d2 + d2*d2);
27
28 mgmd=(dab*dbc*dac*dab1*dbc1*dca1*da1b*db1c*dac1*
        da1b1*db1c1*dc1a1)^(1/12);
29 sgmd=((((0.7788*r)^3)*(daa1*dbb1*dcc1))^(1/6));
30 l=2*log([mgmd/sgmd]);
31 x1=2*(%pi)*50*1*10^(-5);
32 printf("\n the inductance is: %.4f Ohm/km\n ",x1);

```

---

Scilab code Exa 7.7 determine the line inductance and the inductive reactance

```

1 clear;
2 //clc();
3 d=6;
4 s=0.3;
5 r=12.5/1000;

```

```

6 dab=6;
7 dbc=6;
8 dca=12;
9
10 ds1=sqrt(0.7788*r*s);
11
12 dm=(dab*dbc*dca)^(1/3);
13
14 lb=2*log([dm/ds1])/10;
15
16 x1=2*(%pi)*50*lb;
17
18 ds1=0.7788*r*sqrt(2);
19
20 l1=2*log([dm/ds1])/10;
21
22 x11=2*(%pi)*50*l1;
23 printf("\n the line inductance is: %.3f Ohm/km\n",
        x11/1000);
24
25 pu_red=(x11-x1)/x11;
26 printf("\n the pu reduction is: %.3f\n",pu_red);

```

---

**Scilab code Exa 7.8** To calculate the capacitance of each conductor to neutral per

```

1 clear;
2 //clc();
3 d=3;
4 r=0.01;
5 cn=2*(%pi)*8.85*10^(-12)/log([d/r])*1000000000000;
6 printf("\n the capacitance is: %.2f F/km\n ",cn);
7
8 c1=0.5*cn;
9 printf("\n the line to line capacitance is: %.2f
        *10^(-9)F/km\n ",c1);

```

```

10
11 bc=2*(%pi)*50*cn;
12 printf("\n the capacitance susceptance is: %.2f
    *10(-6) S/km\n ",bc/1000);

```

---

Scilab code Exa 7.9 To calculate the charging current per km and the reactive volt

```

1 clear;
2 //clc();
3 //taking the values calculated in eg7.2.sce
4 dm=(3*5*3.6)^(1/3);
5 dsc=10.5/1000;
6 vn=132*1000;
7
8 cn=2*(%pi)*8.85*10(-12)/log([dm/dsc]);
9
10 xc=1/(2*(%pi)*50*cn);
11
12 ic=vn/(xc*sqrt(3));
13 printf("\n the charging current is:%.3f A/km\n",ic
    *1000);
14
15 va=(vn*vn)/xc;
16 printf("\n the volt-ampere is:%.2f kVAr\n",va);

```

---

Scilab code Exa 7.10 to find the capacitance per km to neutral and the capacitive

```

1 //taking calculations in eg7.3
2 clear;
3 //clc();
4 dm=7.56;
5 dsc=12.5/1000;
6 r=dsc;

```

```

7
8 cn=1/(18*10^(9)*log([dm/dsc]))*10^(12);
9 printf("the capacitance is: %.2f*10^(-9) F/km\n ",cn
   );
10
11 xc=1/(2*(%pi)*50*cn);
12 printf("the reactance is: %.2f*10^(5) Ohm\n ",xc
   *10000);

```

---

**Scilab code Exa 7.11** To calculate the capacitive reactance per phase per km of the

```

1 clear;
2 //clc();
3 dm=5.4576;
4 r=7.125/1000;
5 daa1=6*sqrt(2);
6 dbb1=9;
7 dcc1=daa1;
8
9 dsc=((r^3)*(daa1*dbb1*dcc1))^(1/6);
10
11 cn=1/(18*10^(9)*log([dm/dsc]));
12 printf("\n the capacitance is: %.2f uF/km\n ",cn
   *10^(12));
13
14 xc=1/(2*(%pi)*50*cn);
15 printf("\n the reactance is: %.2f Ohm/km\n ",xc
   *.001);

```

---

**Scilab code Exa 7.12** To calculate the capacitance of the line

```

1 //values taken from eg7.1
2 clear;

```

```

3 //clc ();
4 r=10/1000;
5 d=3;
6 h=6;
7 h1=2*h;
8 h2=2*h;
9 h12=sqrt(d^2 + 4*h^2);
10
11 h21=h12;
12
13 hm=sqrt(h12*h21);
14 hs=sqrt(h1*h2);
15
16 cn=1/(18*(10^9)*(log([d/r])-log([hm/hs])));
17 printf("\n the capacitance is: %f*10^(-12) F/m\n "
        ,cn*10^(12));
18
19 cl=0.5*cn;
20 printf("\n the capacitance is: %f*10^(-9) F/km\n "
        ,cl*10^(12));

```

---

Scilab code Exa 7.13 To calculate the capacitance per km to neutral and the capaci

```

1 clear;
2 //clc ();
3 //values as calculated in eg7.7.sce
4 d=6;
5 s=0.3;
6 r=12.5/1000;
7
8 dsc=sqrt(r*s);
9
10 dm=7.56;
11 cn=1/(18*(10^(9))*(log([dm/dsc])));
12 printf("\n the capacitance per km is: %.2f F/km\n",

```

```

        cn*10^(12));
13
14 xcb=1/(2*(%pi)*50*cn*1000);
15 printf("\n the reactance is: %.2f Ohm\n",xcb*.00001)
    ;
16
17 ds1=sqrt(2)*r;
18 dm1=7.56;
19
20 c1=1000/(18*(10^9)*(log([dm1/ds1])));
21 printf("\n the capacitance is: %.2f F/km\n",c1
    *10^9));
22
23 xc1=1/(2*(%pi)*50*c1);
24 printf("\n the reactance is: %.2f Ohm\n",xc1);
25
26 cb=13.721*(10^(-9));
27 pu_cap=(cb-c1)/c1;
28 printf("\n the per unit capacitance is: %.3f\n",
    pu_cap);

```

---

Scilab code Exa 7.14 To calculate the inductive reactance and the capacitive reactance

```

1 clear;
2 //clc();
3 d=20;
4 s=0.5;
5 r=20/1000;
6 dab=20;
7 dbc=20;
8 dca=40;
9
10 ds1=((sqrt(2)*0.7788*r*(s*s*s))^(1/4));
11
12 dm=(dab*dbc*dca)^(1/3);

```

```

13
14 lb=2*log([dm/ds1]);
15
16 xlb=2*(%pi)*lb*50;
17
18 dsc=(sqrt(2)*r*(s^3))^(1/4);
19
20 cn=1/(18*(10^(9))*(log([dm/dsc])));
21 printf("the capacitance is: %.2f*10^(-9) F/km\n",cn
        *10^(12));
22
23 xcb=1/(2*(%pi)*50*cn*1000);
24 printf(" the reactance is: %.2f*10^(5) Ohm/km\n",xcb
        *.00001);

```

---

**Scilab code Exa 7.15** To calculate the voltage induced per km

```

1 clear;
2 //clc();
3 dab=1.2;
4 d12=0.4;
5 db2=0.85;
6
7 db1=sqrt(0.8*0.8 + db2*db2);
8
9 da2=db1;
10
11 da1=sqrt(d12*d12 + db2*db2);
12
13 v=2*(%pi)*50*2*60*log([db1*db1/da1*da1]);
14 printf("\n the voltage induced per unit length is: %
        .3f V/km\n",v/10000);

```

---

Scilab code Exa 7.16 to calculate the voltage induced in telephone conductor due to

```
1 clear;
2 clc;
3
4 Db0=6*cosd(30);
5 Da0=3;
6 D01=5;
7 D12=.6;
8 Db1=Db0+D01
9 Db2=Db1+D12;
10 Da1=sqrt(Da0*Da0+D01*D01);
11 Dc1=Da1;
12 Dc2=sqrt(Da0*Da0+5.6*5.6);
13 Da2=Dc2;
14 ia=200;
15 ib=200*(cosd(-120)+(%i)*sind(-120));
16 ic=200*(cosd(120)+(%i)*sind(120));
17 lam=2*10^(-7)*(ia*log([Da2/Da1])+ib*log([Db2/Db1])+
    ic*log([Dc2/Dc1]));
18 lamda=sqrt(real(lam)^2+imag(lam)^2);
19 v=2*(%pi)*50*lamda;
20 bc=6;
21 oc=3;
22 b0=sqrt(bc*bc-oc*oc);
23 hc=15;
24 hb=b0+hc;
25 ha=hc;
26 r=19.53/2000;
27 va=132000/sqrt(3);
28 v1a=(log([(2*ha-Da1)/Da1])/log([(2*ha-r)/r]));
29 v1b=(log([(2*hb-Db1)/Db1])/log([(2*hb-r)/r]));
30 v1c=(log([(2*hc-Dc1)/Dc1])/log([(2*hc-r)/r]));
31 vb=v1b*(cosd(-120)+(%i)*sind(-120));
32 vc=v1c*(cosd(120)+(%i)*sind(120));
33 v1=va*(v1a+vb+vc);
34 V1=sqrt(real(v1)^2+imag(v1)^2);
35 printf("The voltage induced is :%.2f kV",V1/1000);
```

---

Scilab code Exa 7.17 To calculate the inductance per unit length

```
1 clear;
2 clc();
3 dad=6;
4 dbe=dad;
5 dae1=sqrt(4*4 + 6*6);
6 dbd=dae1;
7 dce=dae1;
8
9 ded=sqrt(8*8 + 6*6);
10 dcd=ded;
11 dm=(dad*dae1*dbd*dbe*dcd*dce)^(1/6);
12
13 dbc=4;
14 dcb=dbc;
15 daa=0.7788*20/1000;
16 dbb=daa;
17 dcc=daa;
18
19 dab=4;
20 dbc=dab;
21 dba=dab;
22
23 dca=8;
24 dac=dca;
25
26 dsa=(daa*dab*dac*dca*dbb*dbc*dcb*dcc)^(1/9);
27
28 la=2*10^(-7)*log([dm/dsa]);
29
30 ddd=0.7788*40/1000;
31 dee=ddd;
32 dde=4;
```

```

33 ded=dde;
34
35 dsb=(ddd*dde*dee*ded)^(1/4);
36
37 lb=2*10^(-7)*log([dm/dsb]);
38
39 l=100*(la+lb)/1000000;
40 printf("the inductance per unit length is: %.8f uH/m
    \n",l*10^(5));

```

---

**Scilab code Exa 7.18** To calculate the loop inductance of the line

```

1 clear;
2 //clc();
3 dad=4;
4 dae1=4.3;
5 dbd=3.5;
6 dbe=3.8;
7 dcd=2;
8 dce=2.3;
9
10 dm=(dad*dae1*dbd*dbe*dcd*dce)^(1/6);
11
12 daa=0.7788*3/100;
13 dbb=daa;
14 dcc=daa;
15
16 dab=0.5;
17 dbc=dab;
18 dca=dab;
19 dba=dab;
20
21 dbc=1.5;
22 dcb=dbc;
23 dca=2;

```

```
24 dac=dca;
25
26 dsa=(daa*dab*dac*dca*dbb*dbc*dca*dcdb*dcc)^(1/9);
27
28 la=2*log([dm/dsa]);
29
30 ddd=0.7788*4/100;
31 dee=ddd;
32 dde=0.3;
33 ded=dde;
34
35 dsb=(ddd*dde*dee*ded)^(1/4);
36
37 lb=2*log([dm/dsb]);
38
39 l=100*(la+lb);
40 printf("the loop inductance is: %.3f mH\n",l/10000);
```

---

# Chapter 8

## per unit representation

Scilab code Exa 8.1 To calculate the per unit impedance and admittance

```
1 clear;
2 //clc();
3 z=complex(4,60);
4 mva_base=100;
5 kv_base=230;
6 zpu=z*mva_base/(kv_base^2);
7 printf('the per unit impedance is:');
8 disp(zpu)
9
10 ys=(%i)*(2*10^(-3));
11 ypu=ys*(kv_base^2)/mva_base;
12 printf("the per unit admittance is:");
13 disp(ypu);
```

---

Scilab code Exa 8.2 Three phase apparent power in pu

```
1 clear;
2 //clc();
```

```

3 sb=50000;
4 s=40000;
5 pu=s/sb;
6 printf("\n the pu value for the three phase is: %.2f
   \n",pu);
7
8 sb1=sb/(3);
9 v=110;
10 vb=v/sqrt(3);
11 pukva=s/(3*sb1);
12 printf("\n the pu kva is: %.2f\n",pukva);

```

---

**Scilab code Exa 8.3** determine the total reactance in per unit

```

1 clear;
2 //clc();
3 sb=5000
4 x1=2.5;
5 x2=2.5;
6 n1=400;
7 n2=200;
8 x1e=x1+x2*(n1/n2)^2;
9
10 vb1=400;
11 x1epu=x1e*sb/vb1^2;
12 printf("\n Total per unit resistnce refered to
   primary is: %.2f",x1epu);
13
14 x2e=x1+x2*(n2/n1)^2;
15 vb2=200;
16 x2epu=x2e*sb/vb2^2;
17 printf("\n Total per unit resistnce refered to
   secondary is: %.2f",x2epu);
18
19 //c)if the primary voltage is used as base

```

```

20
21 xpu1=x1*sb/vb1^2;
22 printf("\n Primary per unit reactance is: %.2f ",
    xpu1);
23
24 xpu2=x1*sb/vb2^2;
25 printf("\n Seconadary per unit reactnace is: %.2f
    Ohm", xpu2);

```

---

Scilab code Exa 8.4 to calculate per unit impedances at 15 kva base

```

1 clear;
2 clc;
3
4 x1_pu=.12;
5 sb2=15;
6 sb1=10;
7 xg2=8;
8 xg1=.12;
9 sbg2=5;
10 sbg1=100;
11 xtf=6;
12 sbtf=15;
13 xt=complex(4,60);
14 kv1=230;
15 x2_pu=x1_pu*(sb2/sb1);
16 xg_pu1=x2_pu*sbg1;
17 printf("\n The per unit reactance of generator 1 is:
    %.2f", xg_pu1);
18 xg_pu2=xg2*(sb2/sbg2);
19 printf("\n The per unit reactance of generator 2 is:
    %.2f", xg_pu2);
20 xtf_pu=xtf*(sb2/sbtf);
21 printf("\n The per unit reactance of transformer is:
    %.2f", xtf_pu);

```

```

22 xt_pu=xt*sb2/((kv1)^2);
23 printf("\n The per unit impedance of transmission
    line is:");
24 disp(xt_pu)

```

---

Scilab code Exa 8.5 To calculate the voltage drop in line per unit

```

1 clear;
2 //clc();
3 s3q=10*10^6;
4 vl=10.5*1000;
5 il=s3q/(sqrt(3)*vl);
6 iph=il;
7 zph=5;
8 printf("\n Voltage drop in the line is: %.2f V",zph*
    iph);
9
10 //b).using per unit method to soolve
11
12 sb=10;
13 vlb=11;
14 vpu=vl/vlb;
15 sb3q=12;
16 spu=s3q/sb3q;
17 ipu=spu/vpu;
18
19 zpu=zph*sb3q/vlb^2;
20 v=zpu*ipu*vlb/sqrt(3);
21 printf("\n voltage drop in the line per phase is: %
    .2f V",v);

```

---

Scilab code Exa 8.6 To calculte the through impedance

```

1 clear;
2 //clc();
3 function [zpu]=puz(z1pu, sb2, sb1, vb1, vb2)
4     zpu=z1pu*(sb2/sb1)*(vb1/vb2)^2;
5 endfunction
6
7 previousprot = funcprot(0)
8 funcprot(0)
9
10
11 function [parallel]=para(z1, z2)
12     parallel=z1*z2/(z1+z2);
13 endfunction
14
15 previousprot = funcprot(0)
16 funcprot(0)
17
18
19 z1pu=0.20;
20 sb2=25;
21 sb1=30;
22 vb1=11;
23 vb2=11;
24 zpu=puz(z1pu, sb2, sb1, vb1, vb2);
25 printf("\n the per unit reactance of the generator
        is: %f", zpu);
26 xg1=zpu;
27
28 z1pu=0.25;
29 sb2=25;
30 sb1=25;
31 vb1=11;
32 vb2=11;
33 zpu=puz(z1pu, sb2, sb1, vb1, vb2);
34 xg2=zpu;
35 printf("\n the per unit reactance of generator 2 is:
        %f", zpu);
36

```

```

37 z1pu=0.20;
38 sb2=60;
39 sb1=30;
40 vb1=11;
41 vb2=11;
42 zpu=puz(z1pu, sb2, sb1, vb1, vb2);
43 xg11=zpu;
44 printf("\n the per unit reactnace of generator 1 is
         on 60MVA base: %f", zpu);
45
46 z1pu=0.25;
47 sb2=60;
48 sb1=25;
49 vb1=11;
50 vb2=11;
51 zpu=puz(z1pu, sb2, sb1, vb1, vb2);
52 xg22=zpu;
53 printf("\n the per unit reactnace of generator 2 is
         on 60MVA base: %f", zpu);
54
55 //calcultaion of per unit impedance of transformer
56
57 z1pu=0.10;
58 sb2=25;
59 sb1=60;
60 vb1=11;
61 vb2=11;
62 zpu=puz(z1pu, sb2, sb1, vb1, vb2);
63 xt1=zpu;
64 printf("\n the per unit reactnace of generator 1 is
         on 30MVA base: %.3 f", zpu);
65
66 z1pu=0.10;
67 sb2=60;
68 sb1=60;
69 vb1=11;
70 vb2=11;
71 zpu=puz(z1pu, sb2, sb1, vb1, vb2);

```

```

72 xt2=zpu;
73 printf("\n the per unit reactnace of generator 1 is
       on 60MVA base:  %.3f",zpu);
74
75 //calculation of through impedance
76
77 zt=para(xg1,xg2) + xt1;
78 printf("\n the through impedance at 25MVA base is:
       j%.3f",zt);
79
80 zt1=para(xg11,xg22)+xt2;
81 printf("\n the through impedance at 60MVA base is:
       j%.3f",zt1);
82
83 //calcultaion in ohms
84 sb1=25
85 zb=vb1^2/sb1;
86 printf("\n actual impedance in ohms on 25MVA base is
       : j%.3f",zb*zt);
87
88 sb2=60;
89 zb=vb1^2/sb2;
90 printf("\n actual impedance in ohms on 60MVA base is
       : j%.3f",zb*zt1);

```

---

Scilab code Exa 8.7 to determine the per phase generator voltage

```

1 clear;
2 clc;
3
4 sb=100;
5 vb=15;
6 xg=.75;
7 sbg=75;
8 xtf=.1

```

```

 9 sbtf=50;
10 xt=100;
11 kv1=220;
12 rl=500;
13 vl=210;
14 xg_pu=xg*(sb/sbg);
15 xtf_pu=xtf*(sb/sbtf);
16 xt_pu=xt*sb/((kv1)^2);
17 rl_pu=rl*sb/((kv1)^2);
18 vpu=vl/kv1
19 i_pu=vpu/rl_pu;
20 v=i_pu*(rl_pu+(%i)*(xg_pu+xt_pu+xt_pu));
21 vg=round(sqrt(real(v)^2+imag(v)^2)*vb);
22 printf("The terminal voltage per phase is: %.2f kV",
        vg/sqrt(3));
23
24 //difference in answer is due to rounding off

```

---

Scilab code Exa 8.8 to calculate the per unit impedance of all units

```

1 clear;
2 clc;
3
4 sb=50;
5 kv1_hv=132;
6 kv1_lv=11;
7 blv=15;
8 btf2=180;
9 bkvl=180;
10 n=33/76;
11 z1pu=(%i)*.1;
12 z2pu=(%i)*.12;
13 kv1b1=11;
14 kv1b2=15;
15 kv12b1=33;

```

```

16 kv12b2=45.1;
17 zt=complex(25,75);
18 zm1=(%i)*.15;
19 zm2=(%i)*.15;
20 sm1=30;
21 sm2=20;
22 kv1m=30
23 bt=blv*kv1_hv/kv1_lv;
24 vm=bt*f2*n/sqrt(3);
25 putf1=z1pu*(kv1b1/kv1b2)^2;
26 printf("\n The per unit reaccance of transformer 1 is
      : ");
27 disp(putf1);
28 sbtf2=3*20;
29 putf2=z2pu*(sb/sbtf2)*(kv12b1/kv12b2)^2;
30 printf("\n The per unit reaccance of transformer 2 is
      : ");
31 disp(putf2);
32 pum1=zm1*(sb/sm1)*(kv1m/kv12b2)^2;
33 printf("\n The per unit reaccance of motor 1 is: ");
34 disp(pum1);
35 pum2=zm2*(sb/sm2)*(kv1m/kv12b2)^2;
36 printf("\n The per unit reaccance of motor 2 is: ");
37 disp(pum2);
38 put=zt*sb/(bkv1)^2;
39 printf("\n The per unit impedance of transmission
      line is: ");
40 disp(put);

```

---

Scilab code Exa 8.9 To calculate the voltage at the terminals of the motor

```

1 clear;
2 //clc();
3 ip1=24;
4 ip2=16;

```

```

5 sb=50;
6 tot_ip=ip1+ip2;
7 p=tot_ip/sb;
8 v=30;
9 vb=45.1;
10 vpu=v/vb;
11 ipu=p/vpu;
12
13 zt1pu=0.0537;
14 zlpu=0.0385+(%i)*0.1157;
15 zt2pu=0.0535;
16 vg=vpu+(%i)*zt1pu +zlpu+(%i)*zt2pu)*ipu;
17
18 function [mag,theta]=c(r,i)
19     mag=sqrt(r*r + i*i)
20     theta=atand(i/r)
21 endfunction
22
23
24 previousprot = funcprot(0)
25 funcprot(0)
26
27 r=real(vg);
28 i=imag(vg);
29 [mag,theta]=c(r,i);
30 vt=mag*15;
31 printf("\n the terminal voltage at the generator is:
    %.2f kV",vt);

```

---

Scilab code Exa 8.10 To find the generator bus terminal voltage

```

1 clear;
2 //clc();
3 function [zpu]=puz(z1pu, sb2, sb1, vb1, vb2)
4     zpu=z1pu*(sb2/sb1)*(vb1/vb2)^2;

```

```

5  endfunction
6
7  previousprot = funcprot(0)
8  funcprot(0)
9
10 z1pu=0.15;
11 sb2=20;
12 sb1=20;
13 vb1=11;
14 vb2=12.5;
15 zpu=puz(z1pu, sb2, sb1, vb1, vb2);
16 printf("\n the per unit reactance of the generator
    is: %.2 f", zpu);
17 xg=zpu;
18
19 zpu=puz(0.15, 20, 15, 11, 11);
20 printf("\n the per unit reactance of the generator
    is: %.2 f", zpu);
21 xm=zpu;
22
23 zpu=puz(0.10, 20, 25, 132, 132);
24 printf("\n the per unit reactance of the generator
    is: %.2 f", zpu);
25 xt1=zpu;
26
27 zpu=puz(0.10, 20, 20, 132, 132);
28 printf("\n the per unit reactance of the generator
    is: %.2 f", zpu);
29 xt2=zpu;
30
31 xt1=((200+(%i)*500))*20/(132)^2;
32
33 pl=5;
34 pf=0.8;
35 p=pl*pf;
36 q=pl*0.6;
37
38 vpu=1;

```

```

39 sb=20;
40 rpu=vpu^2*sb*p/(p*p+q*q);
41 xpu=vpu^2*sb*q/(p*p+q*q);
42
43 //when pf is changed to 0.9
44
45 vpu=1.1;
46 im=15*(0.9+(%i)*sqrt(1-0.9^2))/(sb*vpu);
47
48 il=(p-(%i)*q)/(sb*vpu);
49 i=im+il;
50 zt=(%i)*(xt1+xt2)+xt1;
51
52 vg=vpu+zt*i;
53
54 function [mag,theta]=c(r,i)
55     mag=sqrt(r*r + i*i)
56     theta=atand(i/r)
57 endfunction
58
59
60 previousprot = funcprot(0)
61 funcprot(0)
62
63 [mag,theta]=c(real(vg),imag(vg));
64 printf("\nthe terminal generator voltage is: %.2f kV
        ",mag*11);

```

---

# Chapter 9

## Short and medium lines

Scilab code Exa 9.1 calculate the current voltage and power factor of the load

```
1 clear;
2 //clc();
3 //a).unity power factor
4
5 s=200;
6 vr=2500;
7 r=1.4;
8 x=0.8;
9 i=s*1000/vr;
10 z=r+(%i)*x;
11 vs=vr+z*i;
12 qs=atand(imag(vs)/real(vs));
13 pf=cosd(qs);
14 printf("the power factor of the sending end is:%.4f\n",pf);
15
16 //b).load power factor =0.8
17
18 pfl=acosd(0.8);
19 vs=vr+z*i*(cosd(-pfl)+(%i)*sind(-pfl));
20 qs=atand(imag(vs)/real(vs));
```

```

21 pf1=qs-(-pf1);          //negative sign is due to the
    loadis lagging
22 pf=cosd(pf1);
23 printf(" the power factor of the sending end is:%.3f
    \n",pf);
24
25 //c).load factor is 0.8 leading
26
27 pf1=acosd(0.8);
28 vs=vr+z*i*(cosd(pf1)+(%i)*sind(pf1));
29 qs=atand(imag(vs)/real(vs));
30 pf1=qs-(pf1);          //negative sign is due to the
    loadis lagging
31 pf=cosd(pf1);
32 printf(" the power factor of the sending end is:%.3f
    \n",pf);

```

---

Scilab code Exa 9.2 calculate power factor voltage regulation and efficiency

```

1 clear;
2 //clc();
3 s=15000/3;
4 v=33000/sqrt(3);
5 pf=0.85;
6 l=8;
7 r=.29*l;
8 x=0.65*l;
9 i=s*1000/v;
10 qs=acosd(pf);
11 op=3*s*1000*pf;
12 ploss=3*i*i*r;
13
14 z=r+(%i)*x;
15 vs=v+z*i*(cosd(-qs)+(%i)*sind(-qs));
16 vsp=sqrt(real(vs)^2+imag(vs)^2);

```

```

17 vs1=sqrt(3)*vsp;
18 printf("\n the line voltage at the sending end is: %
    .2f kv",vs1/1000);
19
20 //b)phase difference
21 qs1=atand(imag(vs)/real(vs));
22 pf1=qs1-(-qs); //negative sign is due to the
    load is lagging
23 pf=cosd(pf1);
24 printf("\n the power factor of the sending end is:%
    .4f",pf);
25
26 //c).line regulation
27
28 lr=(vsp-v)/v;
29 printf("\n the line regulation of is:%.2f",lr);
30
31 //d).efficiency
32
33 n=op/(op+ploss);
34 printf("\n the transission efficiency is: %.2f
    percent",n*100);

```

---

**Scilab code Exa 9.3** calculate voltage and power factor

```

1 clear;
2 //clc();
3 l=1000000; //length in cm
4 s=5000000;
5 v=11000/sqrt(3);
6 f=50;
7 pf=0.8;
8 rho=.000001774;
9 i=s/(3*v*pf);
10 rp=0.1*s/(3*i*i);

```

```

11 a=rho*l/rp;
12 r=sqrt(a/%pi);
13 r1=0.7788*r;
14 d=200;
15 L=2*.001*log([d/r1]);
16
17 xlp=2*(%pi)*f*L;
18 qs=acosd(pf);
19 z=rp+(%i)*xlp;
20 vs=v+z*i*(cosd(-qs)+(i)*sind(-qs));
21 vsp=sqrt(real(vs)^2+imag(vs)^2);
22 vs1=sqrt(3)*vsp;
23 printf("\n the line voltage at the sending end is: %
    .2f kv",vs1/1000);
24
25 qs1=atand(imag(vs)/real(vs));
26 pf1=qs1-(-qs); //negative sign is due to the
    load is lagging
27 pf=cosd(pf1);
28 printf("\n the power factor of the sending end is:%
    .2f (lagging)",pf);

```

---

Scilab code Exa 9.4 determine the voltage and power factor

```

1 clear;
2 //clc();
3 a=sqrt(3);
4 r1=5;
5 r2=1.5;
6 pf=0.8;
7 reh=r1+(a)*(a)*r2;
8 x1=10;
9 x2=3;
10 xeh=x1+(a)*(a)*x2;
11 r1=2; //resistance of the line

```

```

12 x1=3;          //recatance of the line
13 r=r1+rehv;
14 x=x1+xehv;
15 s=3000;
16 v=33000;
17 i=s*1000/(v*sqrt(3));
18 vr=v/sqrt(3);
19 qs=acosd(pf);
20 z=r+(%i)*x;
21 vs=vr+z*i*(cosd(-qs)+(i)*sind(-qs));
22 vsp=sqrt(real(vs)^2+imag(vs)^2);
23 vs1=sqrt(3)*vsp;
24 printf("\n the line voltage at the sending end is: %
    .2f kv",vs1/1000);
25
26 qs1=atand(imag(vs)/real(vs));
27 pf1=qs1-(-qs);      //negative sign is due to the
    load is lagging
28 pf=cosd(pf1);
29 printf("\n the power factor of the sending end is:%
    .3f",pf);

```

---

Scilab code Exa 9.5 the voltage at the generator busbars

```

1 clear;
2 //clc();
3
4 s=6000000;
5 v=6000;
6 pf=.8;
7 a=66/330;
8 r=7;
9 x=2;
10 rt=2;
11 xt=18;

```

```

12 totr=r+rt;
13 totx=x+xt;
14 vhv=v/a;
15 il=s/(sqrt(3)*vhv);
16 vph=vhv/sqrt(3);
17 vs=sqrt((il*totr+vph*pf)^2+(il*totx+vph*.6)^2);
18 vl=vs*sqrt(3);
19 vlv=vl*a;
20 printf("the volatge at the genertor busbars is:%0.3f
        V",vlv);

```

---

Scilab code Exa 9.6 determine voltage current power factor apparent power efficien

```

1 clear;
2 clc();
3
4 function [r,i]=d(mag,theta)
5     r=mag*cosd(theta);
6     i=mag*sind(theta);
7
8 endfunction
9
10
11
12 previousprot = funcprot(0)
13 funcprot(0)
14
15 vr=60000;
16 pr=36000000;
17 p=(12*10^6);
18 pf=.8;
19 mag=p/(vr*pf);
20 theta=acosd(pf);
21
22 [r,i]=d(mag,theta);

```

```

23 ir=conj(complex(r,i));
24
25 f=50;
26 l=.1;
27 c=.25*10^(-6);
28 r=2.5;
29 xl=2*(%pi)*f*l;
30 z=r+(%i)*xl;
31 y=(%i)*2*(%pi)*f*c;
32 //calculations by nominal T model
33
34 vab=vr+.5*z*ir;
35 iab=y*vab;
36 is=ir+iab;
37 printf("\n The current in A at the sending end is:")
    ;
38 disp(is);
39 qi=atand(imag(is)/real(is));
40
41 //voltage drop in the left hand half of the line
42 vd=is*z/2;
43 vs=vab+is*z/2;
44 printf("\n The volatge in V at the sending end is:")
    );
45 disp(vs);
46 vl=sqrt(3)*vs;
47 qs=atand(imag(vs)/real(vs))-qi;
48 pfs=cosd(qs);
49 printf("\n The lagging power factor at the sending
    end is:");
50 disp(pfs);
51 s3=3*vs*conj(is);
52 printf("\n The apparent power in VA at the sending
    end is:");
53 disp(s3);
54 eff=pr/real(s3);
55 printf("\n The transmission efficiency is: %.4f per
    cent",eff*100);

```

```

56
57 A=1+.5*z*y;
58 r=real(A);
59 i=imag(A);
60 function [mag,theta]=c(r,i)
61     mag=sqrt(r*r + i*i)
62     theta=atand(i/r)
63 endfunction
64 previousprot = funcprot(0)
65 funcprot(0)
66
67 [mag,theta]=c(r,i);
68 magA=mag;
69 r=real(vs);
70 i=imag(vs);
71 [mag,theta]=c(r,i);
72
73 magV=mag;
74
75 vrnl=magV/magA;
76 reg=(vrnl-vr)/vr;
77 printf("\n The regulation is :%.4f per cent",reg*100)
    ;
78
79 //calculations based on pi model
80 printf("\n \n Calculations based on pi model:");
81 iab=y*vr/2;
82 i=ir+iab;
83 vd=i*z;
84 vs=vr+vd;
85 printf("\n \nThe volatge in V at the sending end is
    :");
86 disp(vs);
87 icd=y*vs/2
88 is=i+icd;
89 printf("\n The current in A at the sending end is:")
    ;
90 disp(is);

```

```

91 qis=atand(imag(vs)/real(vs))-atand(imag(is)/real(is)
    );
92 pfs=cosd(qis);
93 printf("\n The lagging power factor at the sending
    end is:");
94 disp(pfs);
95 s3=3*vs*conj(is);
96 printf("\n The apparent power in VA at the sending
    end is:");
97 disp(s3);
98 eff=pr/real(s3);
99 printf("\n The transmission efficiency is:%.2f per
    cent",eff*100);
100
101 A=1+.5*z*y;
102 r=real(A);
103 i=imag(A);
104 function [mag,theta]=c(r,i)
105     mag=sqrt(r*r + i*i)
106     theta=atand(i/r)
107 endfunction
108 previousprot = funcprot(0)
109 funcprot(0)
110
111 [mag,theta]=c(r,i);
112 magA=mag;
113 r=real(vs);
114 i=imag(vs);
115 [mag,theta]=c(r,i);
116
117 magV=mag;
118
119 vrnl=magV/magA;
120 reg=(vrnl-vr)/vr;
121 printf("\n The regulation is:%.4f per cent",reg*100)
    ;

```

---

Scilab code Exa 9.7 calculate receiving end load

```
1 clear;
2 clc;
3
4 f=50;
5 l=150;
6 L=.2;
7 C=1.5*10(-6);
8 vr=110000/sqrt(3);
9 vs=vr;
10 xl=2*(%pi)*f*L;
11 z=(%i)*xl;
12 y=(%i)*2*(%pi)*f*C;
13 iab=y*vr/2;
14 //i=ir+iab
15 //using the formula vs=vr+i*z
16
17 ir=sqrt(vs*vs-(vr+iab*z)^2)/imag(z);
18 printf("The load current is :%.3f A",ir);
```

---

Scilab code Exa 9.8 to calculate A B C D constants sending end voltage current power

```
1 clear;
2 clc;
3
4 v1=132000;
5 s=50000000;
6 pf=.85;
7 l=80;
8
9 function [r,i]=d(mag,theta)
```

```

10     r=mag*cosd(theta);
11     i=mag*sind(theta);
12
13     endfunction
14
15
16
17     previousprot = funcprot(0)
18     funcprot(0)
19
20     mag=96;
21     theta=78;
22     [r,i]=d(mag,theta);
23     z=complex(r,i);
24
25     mag=.001;
26     theta=90;
27     [r,i]=d(mag,theta);
28     y=complex(r,i);
29
30     vrp=v1/sqrt(3);
31     Irp=s/(sqrt(3)*v1*pf);
32
33     mag=Irp;
34     theta=-acosd(pf);
35     [r,i]=d(mag,theta);
36     irp=complex(r,i);
37
38     //a).for the nominal T network parameters are
39     A=1+.5*z*y;
40     B=z*(1+.25*z*y);
41     C=y;
42     D=A;
43
44     disp(A);
45     disp(B);
46     disp(C);
47     disp(D);

```

```

48
49 //phase voltage at the sending end is
50 vsp=A*vrp+B*irp;
51 vs1=sqrt(3)*vsp;
52 disp(vsp);
53
54 //c).
55 is=C*vrp+D*irp;
56 disp(is);
57
58 //d).
59 qs=atand(imag(vsp)/real(vsp))-atand(imag(is)/real(is
    ));
60 printf("\n The power factor at the sending end is:%
    .3f (lagging)",cosd(qs));
61
62 //e).
63 r=real(vs1);
64 i=imag(vs1);
65 function [mag,theta]=c(r,i)
66     mag=sqrt(r*r + i*i)
67     theta=atand(i/r)
68 endfunction
69 previousprot = funcprot(0)
70 funcprot(0)
71
72 [mag,theta]=c(r,i);
73 Vs1=mag;
74
75 r=real(is);
76 i=imag(is);
77 [mag,theta]=c(r,i);
78
79 Is=mag;
80 eff=s/(sqrt(3)*Vs1*Is*cosd(qs));
81 printf("\n The efficiency of transmission is:%.2f
    per cent",eff*100);

```

---

# Chapter 10

## Long transmission lines

Scilab code Exa 10.1 To calculate the A B C D constants

```
1 clear;
2 //clc();
3 z=12.5 + (%i)*66;
4 y=(%i)*4.4*(10^(-4));
5 yS=sqrt(z*y);
6
7 aS=real(yS);
8
9 bS=imag(yS);
10
11 A=cosh(yS);
12
13 printf("\n the constant A is: %f\n ",A);
14
15 D=A;
16
17 printf("\n the constant D is: %f\n ",D);
18
19 zo=sqrt(z/y);
20 B=zo*sinh(yS);
21
```

```

22 printf("\n the constant B is:  %f\n ",B);
23
24 C=sinh(yS)/(zo);
25
26 printf("\n the constant C is:  %f\n ",C);

```

---

Scilab code Exa 10.2 To calculate the sending end voltage sending end current line

```

1 clear;
2 //clc();
3 v=275*1000;
4 vrp=v/sqrt(3);
5 p=250*10^(6);
6 cosq=0.9;
7 ir1=p/(sqrt(3)*v*cosq);
8 ir=ir1*(cosd(25.84) + (%i)*sind(-25.84));
9
10 A=0.9855138 + 0.0027367*(%i);
11 D=0.9855138 + 0.0027367*(%i);
12 C=- 0.0000004 + 0.0004379*(%i);
13 B=12.37926 + 65.692432*(%i);
14
15 is=C*vrp + D*ir;
16 printf('the sending end current is:');
17 disp(is);
18 vsp=A*vrp + B*ir;
19 printf('the sending end voltage is:');
20 disp(vsp);
21 ps=real(3*vsp*conj(is));
22
23 tx_eff=p/ps;
24 printf("\n the transmission efficiency is:  %.3f pu\
      n ",tx_eff);
25
26 vn1=polar(vsp)/A;

```

```
27
28
29 vfl=158.77*1000;
30 vr=(vn1-vfl)/vfl;
31 printf("\n the voltage regulation is: %.3f pu\n ",
        vr);
```

---

# Chapter 11

## General networks constants

Scilab code Exa 11.2 To find the volatge current and pf at the sending end

```
1 clear;
2 clc;
3 function [r,i]=d(mag,theta)
4     r=mag*cosd(theta);
5     i=mag*sind(theta);
6
7 endfunction
8
9 previousprot = funcprot(0)
10 funcprot(0)
11
12 //clc();
13 mag=0.98;
14 theta=3;
15 [r,i]=d(mag,theta);
16 A=complex(r,i);
17
18 D=A;
19
20 mag=110;
21 theta=75;
```

```

22 [r,i]=d(mag,theta);
23 B=complex(r,i);
24
25 mag=0.0005;
26 theta=88;
27 [r,i]=d(mag,theta);
28 C=complex(r,i);
29
30 v=132*10^(3);
31 s=50*10^(6);
32 pf=0.8;
33
34 vrp=v/sqrt(3);
35 mag=vrp;
36 theta=0;
37 [r,i]=d(mag,theta);
38 vr=complex(r,i);
39
40 i=s/(sqrt(3)*v);
41 mag=i;
42 theta=-acosd(pf);
43 [r,i]=d(mag,theta);
44 ir=complex(r,i);
45
46 vsp=A*vr + B*ir;
47 printf('the voltage at the sending end is:');
48 disp(vsp);
49 is=C*vr + D*ir;
50 printf('the current at the sending end is:');
51 disp(is);
52 qs=atand(imag(vsp)/real(vsp))-atand(imag(is)/real(is
    ));
53 printf("The power factor is :%.4f",cosd(qs));

```

---

Scilab code Exa 11.3 To calculate sending end voltage

```

1 clear;
2 //clc();
3
4 function [r,i]=d(mag,theta)
5     r=mag*cosd(theta);
6     i=mag*sind(theta);
7
8 endfunction
9
10 previousprot = funcprot(0)
11 funcprot(0)
12
13 mag=0.97;
14 theta=0.6;
15 [r,i]=d(mag,theta);
16 a1=complex(r,i);
17
18
19 mag=60;
20 theta=70;
21 [r,i]=d(mag,theta);
22 b1=complex(r,i);
23
24
25 mag=0.97;
26 theta=0.4;
27 [r,i]=d(mag,theta);
28 a2=complex(r,i);
29
30 mag=50;
31 theta=76;
32 [r,i]=d(mag,theta);
33 b2=complex(r,i);
34
35 s=50*10^(6);
36 v=132*10^(3);
37 pf=0.8;
38

```

```

39
40 vrp=v/sqrt(3);
41 mag=vrp;
42 theta=0;
43 [r,i]=d(mag,theta);
44 vr=complex(r,i);
45
46 i=s/(sqrt(3)*v*pf);
47 mag=i;
48 theta=-acosd(pf);
49 [r,i]=d(mag,theta);
50 ir=complex(r,i);
51
52 A=(a1*b2 + a2*b1)/(b1 + b2);
53
54 B=(b1*b2)/(b1+b2);
55
56 vsp=A*vr + B*ir;
57 printf('the sending end phase voltage is:');
58 disp(vsp);

```

---

Scilab code Exa 11.4 To calculate the value of ABCD parameters and the characteris

```

1 clear;
2 //clc();
3 function [r,i]=d(mag,theta)
4     r=mag*cosd(theta);
5     i=mag*sind(theta);
6
7 endfunction
8
9 previousprot = funcprot(0)
10 funcprot(0)
11
12 mag=120;

```

```

13 theta=60;
14 [r,i]=d(mag,theta);
15 z=complex(r,i);
16
17 mag=5*10^(-3);
18 theta=90;
19 [r,i]=d(mag,theta);
20 y=complex(r,i);
21 A=1 + 0.5*z*y;
22 printf('the value of A is:');
23 disp(A)
24 B=z;
25 printf('the value of B is:');
26 disp(B)
27 C=y*(1+ 0.25*(z)*(y));
28 printf('the value of C is:');
29 disp(C)
30 D=A;
31 printf('the value of D is:');
32 disp(D)

```

---

Scilab code Exa 11.5 To determine the equivalent T network

```

1 //to determine the equivalent t-network
2
3 //clc();
4 rab=720;
5 rac_oc=1240;
6 rcd=910;
7 r3=sqrt((rac_oc-rab)*(rcd));
8 printf("\n the R3 resistance is: %f ohm\n",r3);
9
10 r2=rcd-r3;
11 printf("\n the R2 resistance is: %f ohm\n",r2);
12

```

```
13 r1=rac_oc-r3;
14 printf("\n the R1 resistance is:  %f ohm\n",r1);
```

---

# Chapter 13

## Control of voltage and reactive power

Scilab code Exa 13.1 To calculate the maximum power phase difference for maximum p

```
1 clear;
2 clear;
3 //clc();
4
5
6 function [mag,theta]=c(r,i)
7     mag=sqrt(r*r + i*i)
8     theta=atand(i/r)
9 endfunction
10
11
12 previousprot = funcprot(0)
13 funcprot(0)
14
15 r=10;
16 x=30;
17 i=x;
18 vs=132;
19 vr=vs;
```

```

20  z=r + (%i)*x;
21  B=z;
22  A=1;
23  b=atand(x/r);
24  a=0;
25
26  [mag,theta]=c(r,i);
27  pm=(vs*vr/mag) - ((A*vr^(2)/mag)*cosd(theta));
28  printf("\n the maximum power is:  %.2f MW\n",pm);
29
30  //maximum power is at an angle
31  del=theta;
32  printf("\n the maximum power angle is:  %.2f degrees
        \n",del);
33
34  //determining the rating of the synchronous phase
        modifier
35
36  p1=100;
37  pr0=-(vr^2)*r/mag^(2);
38
39  qr0=-(vr^(2)*x)/mag^(2);
40
41  pr=vs*vr/mag;
42
43  q=sqrt(pr^(2) - (pr0 - p1)^(2)) + qr0;
44
45  qpm=p1*tand(acosd(0.9)) - q;
46  printf("\n the rating of the phase modifier is:  %.2
        f (leading)",qpm);

```

---

Scilab code Exa 13.2 To find the rating of the modifier and the power factor

```

1  clear;
2  //clc();

```

```

3 function [mag,theta]=c(r,i)
4     mag=sqrt(r*r + i*i)
5     theta=atand(i/r)
6
7 endfunction
8
9 previousprot = funcprot(0)
10 funcprot(0)
11
12 r=25;
13 x=90;
14 i=x;
15 [mag,theta]=c(r,i);
16 vs=145;
17 vr=132;
18
19 pl=0;
20 p=50;
21 pr0=-(vr^2)*r/mag^(2);
22
23 qr0=-(vr^(2)*x)/mag^(2);
24
25 pr=vs*vr/mag;
26
27 q=sqrt(pr^(2) - (pr0 - pl)^(2)) + qr0;
28
29 qpm=q;
30 printf("\n the rating of the synchronous phase
        modifier is: %.2f MVar\n",qpm);
31 phi=atand(qpm/p);
32 printf("\n the pf is: %.2f (lagging)\n",phi);

```

---

Scilab code Exa 13.3 To calculate the sending voltage maximum power and additional

```
1 clear;
```

```

2  //clc();
3
4
5  function [mag,theta]=c(r,i)
6      mag=sqrt(r*r + i*i)
7      theta=atand(i/r)
8  endfunction
9
10
11 previousprot = funcprot(0)
12 funcprot(0)
13
14
15 function [r,i]=d(mag,theta)
16     r=mag*cosd(theta);
17     i=mag*sind(theta);
18
19 endfunction
20
21
22
23 previousprot = funcprot(0)
24 funcprot(0)
25
26 vr=275;
27 vrl=vr*1000;
28 pf=.9;
29 vrp=vrl/sqrt(3);
30 vs=290;
31
32 magA=.94;
33 mag=magA;
34 thetaA=1;
35 theta=thetaA;
36 [r,i]=d(mag,theta);
37 A=complex(r,i);
38
39 magB=107;

```

```

40 mag=magB;
41 thetaB=78;
42 theta=thetaB;
43 [r,i]=d(mag,theta);
44 B=complex(r,i);
45 pr=300*10^6;
46 ir=pr/(sqrt(3)*vr1*pf);
47 mag=ir;
48 theta=-acosd(0.9);
49 [r,i]=d(mag,theta);
50 ir=complex(r,i);
51 vsp=A*vrp+B*ir;
52 vs1=sqrt(3)*vsp;
53 printf("The line voltage at the sending end:");
54 disp(vs1)
55 pm=(vs*vr/magB) - ((magA*vr^(2)/magB)*cosd(thetaB-
    thetaA));
56 printf("\n the maximum power is: %.2f MW\n",pm);
57
58 //determining the rating of the synchronous phase
    modifier
59
60 pl=450;
61 pr=pl*pf;
62 pr0=-(vr^2)*magA*cosd(thetaB-thetaA)/magB;
63 qr0=-(vr^2)*magA*sind(thetaB-thetaA)/magB;
64 rhor=vs*vr/magB;
65 q=sqrt(rhor^(2) - (pr - pr0)^(2)) + qr0;
66
67 qpm=pr*tand(acosd(0.9)) - q;
68 printf("\n the rating of the phase modifier is: %.2
    f MVAr",qpm);

```

---

Scilab code Exa 13.4 To find the mva rating of synchronous phase modifier

```

1  clear;
2  //clc();
3
4  r=14;
5  x=48;
6  i=x;
7  vs=70;
8  vr=66;
9
10 function [mag,theta]=c(r,i)
11     mag=sqrt(r*r + i*i)
12     theta=atand(i/r)
13 endfunction
14
15
16 previousprot = funcprot(0)
17 funcprot(0)
18
19 [mag,theta]=c(r,i);
20
21 z=r + (%i)*x;
22 y=(%i)*4*10^(-4);
23
24 A=1 + 0.5*z*y;
25
26 [mag,theta]=c(real(A), imag(A));
27 a=theta;
28 A1=mag;
29
30 B=z;
31
32 function [mag,theta]=c(r,i)
33     mag=sqrt(r*r + i*i)
34     theta=atand(i/r)
35 endfunction
36
37 [mag,theta]= c(r,i);
38 b=theta;

```

```

39 B1=mag;
40
41 pr0=- (A1*vr^2)*cosd(b-a)/(B1);
42
43 qr0=- (A1*vr^2)*sind(b-a)/(B1);
44
45 pr=vs*vr/B1;
46
47 p1=0;
48
49 q=sqrt(pr^(2) - (pr0 - p1)^(2)) + qr0;
50
51 qpm=q;
52
53 s=24;
54 prat=s*0.8;
55
56 q=sqrt(pr^(2) - (pr0 - prat)^(2)) + qr0;
57
58 qpmrat=prat*tand(acosd(0.8))-q;
59 printf("\n the rating of the synchronous phase
modifier is: %.2f MVar\n",qpmrat);

```

---

# Chapter 14

## Load flow analysis

Scilab code Exa 14.1 determine Ybus

```
1 clear;
2 clc;
3
4 n=5;
5 m=5;
6 y=zeros(n,m);
7 z12=(%i)*.2;
8 z23=(%i)*.25;
9 z34=(%i)*.3
10 z45=(%i)*.25;
11 z14=(%i)*.5;
12 z15=(%i)*.2;
13 y(1,2)=1/z12;
14 y(2,3)=1/z23;
15 y(3,4)=1/z34;
16 y(4,5)=1/z45;
17 y(1,4)=1/z14;
18 y(1,5)=1/z15;
19
20
21 for i=1:1:n
```

```

22     for j=1:1:m
23         y(j,i)=(y(i,j)+y(i,j))/2;
24
25         end
26 end
27 Y=y;
28 for i=1:1:n
29     for j=1:1:m
30         if (i==j) then
31             for k=1:1:n
32                 y(i,i)=y(i,i)+Y(j,k);
33             end
34         else
35             y(i,j)=y(i,j)*(-1);
36         end
37     end
38 end
39 disp(y)

```

---

#### Scilab code Exa 14.2 determine modified Ybus

```

1 clear;
2 clc;
3
4 y= [- 12*(%i) (%i)*5 0 (%i)*2 (%i)*5;5*(%i) -(%i)*9
      (%i)*4 0 0;0 4*(%i) -(%i)*7.33 (%i)*3.33 0;(%i)*2
      0 3.33*(%i) -(%i)*9.33 (%i)*4;5*(%i) 0 0 (%i)*4
      -(%i)*9]
5 Y=y;
6 z25=(%i)*.4;
7 i=2;
8 j=5;
9 y(i,j)=1/z25;
10 y(i,i)=Y(i,i)+y(i,j);
11 y(j,j)=Y(j,j)+y(i,j);

```

```

12 y(i,j)=-y(i,j);
13 y(j,i)=y(i,j);
14 disp(y)

```

---

#### Scilab code Exa 14.3 determine Ybus

```

1 clear;
2 clc;
3
4 y= [- 12*(%i) (%i)*5 0 (%i)*2 (%i)*5;5*(%i) -(%i)*9
      (%i)*4 0 0;0 4*(%i) -(%i)*7.33 (%i)*3.33 0;(%i)*2
      0 3.33*(%i) -(%i)*9.33 (%i)*4;5*(%i) 0 0 (%i)*4
      -(%i)*9];
5 Y=y;
6 i=3;
7 j=5;
8 k=1;
9 l=3;
10 z35=(%i)*.05;
11 z13=(%i)*.01;
12 y(k,l)=1/z13;
13 y(i,j)=1/z35;
14 y(k,k)=Y(k,k)+y(k,i);
15 y(i,i)=Y(i,i)+y(i,j)+y(k,i);
16 y(j,j)=Y(j,j)+y(l,j);
17 y(i,j)=-y(i,j);
18 y(j,i)=y(i,j);
19 y(k,l)=-y(k,l);
20 y(l,k)=y(k,l);
21 disp(y)

```

---

#### Scilab code Exa 14.4 determine Ybus

```

1  clear;
2  clc;
3
4  z12=.06+(%i)*.18;
5  z23=.08+(%i)*.24;
6  z13=.03+(%i)*.09;
7
8  n=3;
9  m=3;
10 y=zeros(n,m);
11 y(1,2)=1/z12;
12 y(1,3)=1/z13;
13 y(2,3)=1/z23;
14
15 for i=1:1:n
16     for j=1:1:m
17         y(j,i)=(y(i,j)+y(i,j))/2;
18
19         end
20 end
21 Y=y;
22 for i=1:1:n
23     for j=1:1:m
24         if (i==j) then
25             for k=1:1:n
26                 y(i,i)=y(i,i)+Y(j,k);
27             end
28         else
29             y(i,j)=y(i,j)*(-1);
30         end
31     end
32 end
33
34 disp(y)

```

---

### Scilab code Exa 14.5 determine modified Ybus

```
1 clear;
2 clc;
3
4
5 y=[(5-15*(%i)) (-1.67+5*(%i)) (-3.33+10*(%i))
      ;(-1.67+5*(%i)) (2.91-8.75*(%i)) (-1.25+3.75*(%i))
      );(- 3.33+10*(%i)) (- 1.25 + 3.75*(%i))
      (4.58-13.75*(%i))];
6 Y=y;
7 ya=-2.5*(%i);
8 yb=ya;
9 yc=ya;
10 yd=ya;
11 ye=ya;
12 yf=ya;
13 i=1;
14 j=2;
15 k=3;
16 y(k,k)=Y(k,k)+yb+yf;
17 y(i,i)=Y(i,i)+ya+yc;
18 y(j,j)=Y(j,j)+ye+yd;
19 disp(y)
```

---

### Scilab code Exa 14.6 determine Ybus

```
1 clear;
2 clc;
3
4 n=4;
5 m=4;
6 y=zeros(n,m);
7 z12=.25+(%i)*1.0;
8 z13=.20+(%i)*.8;
```

```

 9 z14=.30+(%i)*1.2;
10 z23=.20+(%i)*.8;
11 z34=.15+(%i)*.6;
12 y(1,2)=1/z12;
13 y(1,3)=1/z13;
14 y(1,4)=1/z14;
15 y(2,3)=1/z23;
16 y(3,4)=1/z34;
17
18
19
20 for i=1:1:n
21     for j=1:1:m
22         y(j,i)=(y(i,j)+y(i,j))/2;
23
24         end
25     end
26 Y=y;
27 for i=1:1:n
28     for j=1:1:m
29         if (i==j) then
30             for k=1:1:n
31                 y(i,i)=y(i,i)+Y(j,k);
32             end
33         else
34             y(i,j)=y(i,j)*(-1);
35         end
36     end
37 end
38 disp(y)

```

---

Scilab code Exa 14.7 determine modified Ybus

```

1 clear;
2 clc;

```

```

3
4 n=4;
5 m=4;
6 y=zeros(n,m);
7 z12=.25+(%i)*1.0;
8 z13=.20+(%i)*.8;
9 z14=.30+(%i)*1.2;
10 z23=.20+(%i)*.8;
11 z34=.15+(%i)*.6;
12 y(1,2)=1/z12;
13 y(1,3)=1/z13;
14 y(1,4)=1/z14;
15 y(2,3)=1/z23;
16 y(3,4)=1/z34;
17
18
19
20 for i=1:1:n
21     for j=1:1:m
22         y(j,i)=(y(i,j)+y(i,j))/2;
23
24         end
25 end
26 Y=y;
27 for i=1:1:n
28     for j=1:1:m
29         if (i==j) then
30             for k=1:1:n
31                 y(i,i)=y(i,i)+Y(j,k);
32             end
33         else
34             y(i,j)=y(i,j)*(-1);
35         end
36     end
37 end
38
39 ya=-.08*(%i);
40 yb=ya;

```

```

41 yc=ya;
42 yd=ya;
43 yk=ya;
44 yf=ya;
45 i=1;
46 j=2;
47 k=3;
48 l=4;
49 y(k,k)=y(k,k)+yf;
50 y(i,i)=y(i,i)+ya+yc+yb;
51 y(j,j)=y(j,j)+yd;
52 y(l,l)=y(l,l)+yk;
53 disp(y)

```

---

Scilab code Exa 14.8 determine the voltages at buses

```

1 clear;
2 clc;
3
4
5 n=3;
6 m=3;
7 y=zeros(n,m);
8 y(1,2)=(-(%i)*3);
9 y(1,3)=(-(%i)*4);
10 y(2,3)=(-(%i)*5);
11
12 for i=1:1:n
13     for j=1:1:m
14         y(j,i)=(y(i,j)+y(i,j))/2;
15
16         end
17 end
18 Y=y;
19 for i=1:1:n

```

```

20     for j=1:1:m
21         if (i==j) then
22             for k=1:1:n
23                 y(i,i)=y(i,i)+Y(j,k);
24             end
25         else
26             y(i,j)=y(i,j)*(-1);
27         end
28     end
29 end
30
31 v=[1.02 1 1];
32
33 alpha=1.6;
34
35 del=[0 0 0];
36 p1=[0 50 60];
37 q1=[0 25 30];
38 pg=[0 25 0];
39 qg=[0 15 0];
40 bmva=100;
41 p2=(pg(2)-p1(2))/bmva;
42 q2=(qg(2)-q1(2))/bmva;
43
44 p3=(pg(3)-p1(3))/bmva;
45 q3=(qg(3)-q1(3))/bmva;
46
47 p=[0 p2 p3];
48 q=[0 q2 q3];
49 for i=1:1:2
50     v1=zeros(1,3);
51     v2=v;
52     for j=2:1:3
53         for k=1:1:3
54             if (k~=j) then
55                 v1(j)=v1(j)-y(j,k)*v(k);
56             end
57         end

```

```
58
59     end
60     v(j)=(1/y(j,j))*((p(j)-(%i)*q(j))/conj(v(j))
        +v1(j));
61     delv(j)=v(j)-v2(j);
62     v(j)=v2(j)+alpha*delv(j);
63
64     end
65     printf("The voltages at the end of:%d iteration are"
        ,i);
66         disp(v)
67     end
```

---

# Chapter 15

## Economic operations of power systems

Scilab code Exa 15.1 to find the incremental cost of two units

```
1 clear;
2 clc;
3
4 b=[1 -2 ;1 1];
5 c=[0;450];
6 a=inv(b)*c;
7 p1=a(1);
8 p2=a(2);
9 ic1=2+.01*p1;
10 ic2=2+.02*p2;
11 printf("the incremental fuel cost of first generator
        is: %.2f Rs/MWh", ic1);
12 printf("\n the incremental fuel cost of second
        generator is: %.2f Rs/MWh", ic2);
```

---

Scilab code Exa 15.2 to find the load division between the two units

```

1  clear;
2  clc;
3
4  p1max=200;
5  p2max=p1max;
6  p1min=50;
7  p2min=p1min;
8
9  p1=p1min;
10 p2=p1;
11 ic1=.16*p1+30;
12 ic2=.20*p2+25;
13 printf("For the plant output %.2 f MW p1=%.2fMW    p2=
    %.2fMW\n" ,(p1+p2) ,p1 ,p2);
14
15 for ic=40:5:60
16     ic1=ic;
17     ic2=ic;
18
19         p1=(ic1-30)/.16;
20         p2=(ic2-25)/.20;
21
22     printf("For the plant output %.2 f MW p1=%.2fMW
        p2=%.2fMW\n" ,(p1+p2) ,p1 ,p2);
23     end
24
25 p1=200;
26 p2=200;
27 printf("For the plant output %.2 f MW p1=%.2fMW    p2=
    %.2fMW\n" ,(p1+p2) ,p1 ,p2);

```

---

Scilab code Exa 15.3 determine the saving in fuel cost

```

1  clear;
2  clc;

```

```

3
4 function ic1=f(p1),ic1=.16*p1+30;
5 endfunction
6
7 function ic2=f1(p2),ic2=.20*p2+25;
8 endfunction
9 c1=intg(50,57.5,f);
10 c2=intg(65,57.5,f1);
11 printf("the increase in cost of unit 1 is:%.3f Rs/h\
n",c1);
12 printf("the increase in cost of unit 2 is:%.3f Rs/h\
n",c2);
13
14 net=c1+c2;
15 sav=net*365*24;
16 printf("The total yearly saving is:%.2f Rs",sav);

```

---

Scilab code Exa 15.4 find the loss coefficients and the transmission loss

```

1 clear;
2 clc;
3
4 zac=complex(.05,.2);
5 rac=real(zac);
6 vc=1;
7 i1=1.05;
8 i2=.9;
9 zbc=complex(.04,.16);
10 rbc=real(zbc);
11 zcd=complex(.03,.12);
12 rcd=real(zcd);
13 va=vc+zac*i1;
14 vb=vc+zbc*i2;
15 p1=real(va*conj(i1));
16 p2=real(vb*conj(i2));

```

```

17 b11=(rac+rcd)/(real(va)^2);
18 b12=rcd/(real(va)*real(vb));
19 b22=(rbc+rcd)/(real(vb)^2);
20 p1=p1*p1*b11+p2*p2*b22+2*p1*p2*b12;
21 printf("The transmission loss is: %.4f pu",p1);

```

---

**Scilab code Exa 15.5** find the penalty factor

```

1 clear;
2 clc;
3
4 p1=200;
5 p2=p1;
6 a=.2;.. // a=del(p1)/del(p2)
7 l2=1/(1-a);
8 ic1=.15*p1+150;
9 ic2=.25*p2+175;
10 l1=l2*ic2/ic1;
11 printf("The penalty factor of plant 1 is: %.4f",l1);

```

---

**Scilab code Exa 15.6** find the penalty factor and the additional cost per hour to i

```

1 clear;
2 clc;
3
4 b11=.001;
5 b12=-.0001;
6 b22=.0013;
7
8 p1=150;
9 p2=275;
10 a=2*p1*b11+2*p2*b12;
11 l1=1/(1-a);.. // a=del(p1)/del(p2)

```

```

12 printf("The penalty factor of plant 1 is: %.4f\n",l1
    );
13 lamda=200;
14 ic1=lamda/l1;
15 printf("The incremental cost is: %.0f Rs/MWh",ic1);

```

---

**Scilab code Exa 15.7** calculate the penalty factors for the two plants

```

1 clear;
2 clc;
3
4 p1=100;
5 p1=5;
6 b11=p1/(p1*p1);
7 a=2*p1*b11;.. // a=del(pl)/del(p1)
8 l1=1/(1-a);
9 a1=0;.. // a1=del(pl)/del(p1)
10 l2=1/(1-a1);
11
12 printf("The penalty factor of plant 1 is: %.3f\n",l1
    );
13 printf("The penalty factor of plant 2 is: %.3f\n",l2
    );

```

---

**Scilab code Exa 15.8** determine the generation schedule and the load demand

```

1 clear;
2 clc;
3
4 p1=125;
5 p1=12.5
6 b12=0;
7 b22=0;

```

```
8
9 b11=p1/(p1*p1);
10 lamda=70;
11 a=.000016;
12 p1=82.8729;.. // ic1=.25*p1+40;
13 p2=100;
14 p1=.0008*p1*p1;
15 pr=p1+p2-p1;
16 printf("The total load is :%.4 f MW\n",pr);
17 l1=1/(1-a);
18 a1=0;
19 l2=1/(1-a1);
20 p2=(lamda-50)/.2;
21 printf("for the optimal dispatch P1=%.2 f MW\n",p1);
22 printf("for the optimal dispatch P2=%.2 f MW",p2);..
   // ic2=.20*p2+50
```

---

# Chapter 16

## Symmetrical faults

Scilab code Exa 16.1 To find the high voltage terminals of a transformer

```
1 clear;
2 //clc();
3 s=15;
4 s1=10;
5 xg1=(%i)*0.1;
6 xg1_pu=xg1*s/s1;
7
8 xg2=(%i)*0.075;
9 s2=5;
10 xg2_pu=xg2*s/s2;
11
12 xt1_pu=(%i)*0.06;
13 z=5+(%i)*20;
14 v1=33;
15 z_pu=z*s/v1^2;
16 //printf("%z",z_pu);
17
18 //three phase fault Fa
19
20 x1=(%i)*0.15;
21 x2=(%i)*0.225;
```

```

22
23 ze_pu=xt1_pu+(x1*x2/(x1+x2));
24 Ssc=s/(imag(ze_pu));
25 ifault=Ssc*1000/(sqrt(3)*v1);
26 printf("The fault current is:%.2f A\n",ifault);
27
28 //three phase fault at phase b
29
30 xt=0.06887 + (%i)*0.27548;
31 ze_pu2=xt1_pu+(x1*x2/(x1+x2)) +xt;
32 z_pu=sqrt(real(ze_pu2)^2 + imag(ze_pu2)^2);
33 fb=s/z_pu;
34
35 ifault=fb*1000/(sqrt(3)*v1);
36 printf("The fault current is:%.2f A",ifault);

```

---

Scilab code Exa 16.2 to calculate the subtransient transient and synchronous short

```

1 clear;
2 //clc();
3 x2d=(%i)*0.3;
4 x11=(%i)*0.08;
5 xline=(%i)*0.55;
6 x12=(%i)*0.08;
7
8 ig=0.75;
9 z2t=x2d +x11 +xline + x12;
10 er=1;
11 eint=er+ig*z2t;
12 e2int=sqrt(real(eint)^2 + imag(eint)^2);
13
14 x2gf=imag(x2d + x11);
15 i2d=e2int/x2gf;
16
17 x2bf=imag(x11 +xline);

```

```

18 i2df=er/x2bf;
19 tot_i2d=i2d +i2df;
20 printf("The total subtransient short circuit current
        is:%.3f pu\n",tot_i2d);
21
22 //calculation of effect of maximum dc component
        offset
23
24 i2g=sqrt(2)*i2d;
25 i2f=sqrt(2)*i2df;
26 tot_i=i2g+i2f;
27 max_sc=sqrt(tot_i2d^2+tot_i^2);
28
29 sb=50;
30 vlb=138*10^(3);
31 ilb=sb/(sqrt(3)*vlb*(10^(-6)));
32 isc=ilb*max_sc;
33
34 //caculation of prefault voltage behind transient
        reactance
35
36 x1d=(%i)*0.35;
37 z1t=x1d+xl1+xline+xl2;
38 eint1=er+ig*z1t;
39 e1int=sqrt(real(eint1)^2 + imag(eint1)^2);
40
41 x1gf=imag(x1d + xl1);
42 i1d=e1int/x1gf;
43
44 x1bf=imag(xl1 +xline);
45 i1df=er/x1bf;
46 tot_i1d=i1d +i1df;
47 printf("The total transient short circuit current is
        :%.3f pu\n",tot_i1d);
48
49 isc1=ilb*tot_i1d;
50
51 //calculation of prefault voltage behind synchronous

```

```

    reactance
52
53 xd=(%i)*1.25;
54 zt=xd+xl1+xline+xl2;
55
56 eint3=er+ig*zt;
57 e3int=sqrt(real(eint3)^2 + imag(eint3)^2);
58
59 x3gf=imag(xd + xl1);
60 i3d=e3int/x3gf;
61
62 x3bf=imag(xl1 +xline);
63 i3df=er/x3bf;
64 tot_i3d=i3d +i3df;
65 printf("The total synchronous short circuit current
    is: %.3f pu\n",tot_i3d);

```

---

**Scilab code Exa 16.3** To calculate the reactance of the reactor to prevent the circuit

```

1 clear;
2 //clc();
3 sb=50*10^6;
4 xg=0.2;
5 sg=10*10^6;
6
7 xgpu=sb*xg/sg;
8 xpg=1/4;
9 ssc=500;
10 xu=1/15;
11 vl=33*10^3;
12 ifl=sb/(sqrt(3)*vl);
13 vn=vl/sqrt(3);
14 x=vn*xu/ifl;
15 printf("the reactance of the reactor is: %.2f Ohm",x)
    ;

```

---

Scilab code Exa 16.4 To find the reactor necessary to prevent the switchgear

```
1 clear;
2 //clc();
3
4 sb=7.5*10^6;
5 vl=3.3*10^3;
6 sga=3;
7 sgb=4.5;
8 xgb=0.08;
9 xga=0.07;
10 xga_pu=sb*xga/sga;
11 xgb_pu=sb*xgb/sgb;
12
13 xgp=(xga_pu*xgb_pu)/(xga_pu+xgb_pu);
14 xu=0.0724;
15
16 ifl=sb/(sqrt(3)*vl);
17 vn=vl/sqrt(3);
18 x=xu*vn/ifl;
19 printf("the reactance of the reactor is :%.3f A",x);
```

---

Scilab code Exa 16.5 to find the reactor reactances

```
1 clear;
2 clc;
3 //xt=(.075+1.5x)*(.15)/(.075+1.5x+.15);
4 ssc=200;
5 xpu=.15;
6 ifl=20*10^(6)/(sqrt(3)*11000);
7 vp=11000/sqrt(3);
```

```

8 x=xpu*vp/if1;
9 printf("The ohmic reactance of each reactor is: %.2f
      ohm",x);
10
11 //x1=.0225/(.3+x);
12 //x2=x3=.15x/(.3+x);
13 //xt=.15(.5x+.075)/(.15+.5x+.075);
14 xpu=.45;
15 x=xpu*vp/if1;
16 printf("\nThe ohmic reactance of each reactor is: %
      .2f ohm",x);

```

---

Scilab code Exa 16.6 to determine short circuit MVA and fault current distribution

```

1 clear;
2 clc;
3
4 xt=(.15+.1)*(.2)/(.15+.1+.2);
5 sb=25;
6 ssc=sb/xt;
7 printf("\n The short circuit MVA is: %.2f \n",ssc);
8 if=1/((%i)*xt);
9 ib=sb/(sqrt(3)*11000);
10 if=if*ib;
11 printf("\n The fault current in A is:");
12 disp(round(if*1000000));
13 if1=(.15+.1)*if/(.15+.1+.2);
14 printf("\n The fault current in A supplied by
      generator 1 is:");
15 disp(round(if1*1000000));
16 if2=if-if1;
17 printf("\n The fault current in A in reactor is:");
18 disp(round(if2*1000000));
19 xpu=.45;
20 vf1=6350;

```

```
21 ifl=1050;
22 x=xpu*vfl/ifl;
23 printf("\n The reactance of each reactor is: %.2f
    ohm",x);
```

---

# Chapter 17

## Symmetrical components

Scilab code Exa 17.1 To calculate the positive negative zero sequence component of

```
1 clear;
2 //clc();
3
4 function [r,i]=d(mag,theta)
5     r=mag*cosd(theta);
6     i=mag*sind(theta);
7
8 endfunction
9
10 previousprot = funcprot(0)
11 funcprot(0)
12
13 mag=100;
14 theta=30;
15
16 [r,i]=d(mag,theta);
17 ia=complex(r,i);
18 mag=50;
19 theta=300;
20
21 [r,i]=d(mag,theta);
```

```

22 ib=complex(r,i);
23
24 mag=30;
25 theta=180;
26 [r,i]=d(mag,theta);
27 ic=complex(r,i);
28
29 ia0=(1/3)*(ia+ib+ic);
30 printf("The zero sequence component of current in
    amperes is:");
31 disp(ia0);
32
33 mag=1;
34 theta=120;
35 [r,i]=d(mag,theta);
36 alpha=complex(r,i);
37
38 ia1=(1/3)*(ia+alpha*ib+alpha^2*ic);
39 printf("The positive sequence component of current
    in amperes is:");
40 disp(ia1);
41
42 ia2=(1/3)*(ia+alpha^2*ib+alpha*ic);
43 printf("The negative sequence component of current
    in amperes is:");
44 disp(ia2);
45
46 in=ia+ib+ic;
47 printf("The return current to the neutral conductor
    in amperes is:");
48 disp(in);

```

---

Scilab code Exa 17.2 To determine the symmetrical components of the 3 phase system

```
1 clear;
```

```

2 //clc();
3
4 function [r,i]=d(mag,theta)
5     r=mag*cosd(theta);
6     i=mag*sind(theta);
7
8 endfunction
9
10 previousprot = funcprot(0)
11 funcprot(0)
12
13 ia=complex(12,6);
14 ib=complex(12,-12);
15 ic=complex(-15,10);
16 mag=1;
17 theta=120;
18 [r,i]=d(mag,theta);
19 alpha=complex(r,i);
20
21 ia0=(1/3)*(ia+ib+ic);
22 printf("The zero sequence component of current in
        amperes is:");
23 disp(ia0);
24 ia1=(1/3)*(ia+alpha*ib+alpha^2*ic);
25 printf("The positive sequence component of current
        in amperes ia1=");
26 disp(ia1)
27
28 ib1=alpha^2*ia1;
29 printf("The positive sequence component of current
        in amperes ib1=");
30 disp(ib1);
31
32 ic1=alpha*ia1;
33 printf("The positive sequence component of current
        in amperes ic1=");
34 disp(ic1);
35

```

```

36 ia2=(1/3)*(ia+alpha^2*ib+alpha*ic);
37 printf("The negative sequence component of current
    in amperes ia2=");
38 ib2=alpha*ia2;
39 printf("The negative sequence component of current
    in amperes ib2=");
40 disp(ib2);
41 ic2=alpha^2*ia2;
42 printf("The negative sequence component of current
    in amperes ic2=");
43 disp(ic2);

```

---

**Scilab code Exa 17.3** To find the symmetrical components of the line current

```

1 clear;
2 //clc();
3
4 function [r,i]=d(mag,theta)
5     r=mag*cosd(theta);
6     i=mag*sind(theta);
7
8 endfunction
9
10 previousprot = funcprot(0)
11 funcprot(0)
12
13 mag=150;
14 theta=0;
15 [r,i]=d(mag,theta);
16 ia=complex(r,i);
17 mag=150;
18 theta=-120;
19 [r,i]=d(mag,theta);
20 ib=complex(r,i);
21 mag=150;

```

```

22 theta=120;
23 [r,i]=d(mag,theta);
24 ic=complex(r,i);
25 mag=1;
26 theta=120;
27 [r,i]=d(mag,theta);
28 alpha=complex(r,i);
29
30 ia0=(ia+ib+ic)/3;
31 printf("The zero sequence component of current in
    amperes is:");
32 disp(round(ia0));
33 ia1=(ia+alpha*ib+(alpha^2)*ic)/3;
34 printf("The positive sequence component of current
    in amperes ia1=");
35 disp(round(ia1));
36 ia2=(ia+alpha^2*ib+alpha*ic)/3;
37 printf("The negative sequence component of current
    in amperes ia2=");
38 disp(round(ia2));
39 //b).after fuse removal between two lines
40
41 i_b=0;
42 i_c=0;
43 i_a0=(ia+i_b+i_c)/3;
44 printf("The zero sequence component of current in
    amperes is:");
45 disp(i_a0);
46
47 ia1=(ia+alpha*i_b+alpha^2*i_c)/3;
48 printf("The positive sequence component of current
    in amperes ia1=");
49 disp(round(ia1));
50
51 ia2=(ia+alpha^2*i_b+alpha*i_c)/3;
52 printf("The negative sequence component of current
    in amperes ia2=");
53 disp(ia2);

```

---

Scilab code Exa 17.7 To determine the complex power represented by three phase vol

```
1 clear;
2 //clc();
3
4 function [r,i]=d(mag,theta)
5     r=mag*cosd(theta);
6     i=mag*sind(theta);
7
8 endfunction
9
10 previousprot = funcprot(0)
11 funcprot(0)
12
13 mag=10;
14 theta=190;
15
16 [r,i]=d(mag,theta);
17 ia0=complex(r,i);
18
19 mag=6;
20 theta=20;
21
22 [r,i]=d(mag,theta);
23 ia1=complex(r,i);
24
25 mag=5;
26 theta=50;
27
28 [r,i]=d(mag,theta);
29 ia2=complex(r,i);
30
31 ia=(ia0+ia1+ia2);
32
```

```
33 mag=1;
34 theta=120;
35
36 [r,i]=d(mag,theta);
37
38 alpha=complex(r,i);
39
40 ib=(ia0 +alpha^2*ia1 +alpha*ia2);
41
42 ic=ia0+alpha*ia1+alpha^2*ia2;
43
44 mag=30;
45 theta=-30;
46
47 [r,i]=d(mag,theta);
48 va0=complex(r,i);
49
50 mag=450;
51 theta=0;
52
53 [r,i]=d(mag,theta);
54 va1=complex(r,i);
55
56 mag=225;
57 theta=40;
58
59 [r,i]=d(mag,theta);
60 va2=complex(r,i);
61
62 va=(va0+va1+va2);
63
64 mag=1;
65 theta=120;
66
67 [r,i]=d(mag,theta);
68
69 alpha=complex(r,i);
70
```

```
71 vb=(va0 +alpha^2*va1 +alpha*va2);
72
73 vc=va0+alpha*va1+alpha^2*va2;
74
75 s=3*va0*conj(ia0)+ 3*va1*conj(ia1)+ 3*va2*conj(ia2);
76 printf("The complex power with symmetrical
       components in VA is:");
77 disp(s);
78
79 s1=va*conj(ia)+vb*conj(ib)+ vc*conj(ic);
80 printf("The complex power unbalanced three phase
       components in VA is:");
81 disp(s1);
```

---

# Chapter 18

## Unsymmetrical faults

Scilab code Exa 18.1 To determine the fault current and the line to line voltages

```
1 clear;
2 //clc();
3
4 function [r,i]=d(mag,theta)
5     r=mag*cosd(theta);
6     i=mag*sind(theta);
7
8 endfunction
9
10
11
12 previousprot = funcprot(0)
13 funcprot(0)
14
15 ea=1;
16 za0=(%i)*0.08;
17 za1=(%i)*0.12;
18 za2=(%i)*0.12;
19 sb=25*10^6;
20 vb=11000;
21
```

```

22 zf=(%i)*0.03;
23 ia1=ea/(za0+za1+za2+3*zf);
24
25 ia0=ia1;
26 ia2=ia1;
27 mag=1;
28 theta=120;
29
30 [r,i]=d(mag,theta);
31
32 alpha=complex(r,i);
33
34 ia=ia0+ia1+ia2;
35
36 ibas=sb/(sqrt(3)*vb);
37
38 ia=ia1*ibas;
39 ib=0;
40 ic=0;
41
42 va1=ea-za1*ia1;
43
44 va2=-za1*ia1;
45
46 va0=-(za0+3*zf)*ia1;
47
48
49 va=va0+va1+va2;
50
51 vb=va0+alpha^2*va1+alpha*va2;
52
53 v_c=va0+alpha*va1+alpha^2*va2;
54
55 vab1=va-vb;
56
57 vbc1=vb-v_c;
58
59 vca1=v_c-va;

```

```

60
61 vbas=11/sqrt(3);
62
63 vab=vab1*vbas;
64 printf("The voltage of line ab in kV is vab=");
65 disp(vab);
66
67 vbc=vbc1*vbas;
68 printf("The voltage of line bc in kV is vbc=");
69 disp(round(vbc));
70
71 vca=vca1*vbas;
72 printf("The voltage of line ca in kV is vca=");
73 disp(vca);

```

---

Scilab code Exa 18.2 to find the sequence LLG and LL

```

1 clear;
2 //clc();
3
4
5 function [r,i]=d(mag,theta)
6     r=mag*cosd(theta);
7     i=mag*sind(theta);
8
9 endfunction
10
11
12
13 previousprot = funcprot(0)
14 funcprot(0)
15
16
17 x0g1=0.05;
18 x1g1=0.3;

```

```

19 x2g1=0.2;
20 x0g2=0.03;
21 x1g2=0.25;
22 x2g2=0.15;
23 x0l1=0.70;
24 x1l1=0.3;
25 x2l1=0.3;
26 x0l2=0.7;
27 x1l2=0.3;
28 x2l2=0.3;
29 x0t1=0.12;
30 x1t1=0.12;
31 x2t1=0.12;
32 x0t2=0.1;
33 x1t2=0.1;
34 x2t2=0.1;
35 vf=1;
36 ia01=0;
37
38 z1=(%i)*((x1g1+x1t1)*(x1g2 +x1t1 + x1l1*0.5))/(x1g1
      +x1t1 + x1g2+ x1t1+x1l1*0.5);
39 z2=(%i)*((x2g1+x2t1)*(x2g2+ x2t2+ x2l2*0.5))/(x2g1 +
      x2t1 +x2g2 +x2t2 +x2l2*0.5);
40 z0=(%i)*(x0g1+x0t1);
41 //a).LLG fault
42 za0=z0;
43 za2=z2;
44 ia1=vf/(z1+(z0*z2/(za0+za2)));
45 printf("The positive sequence current in amperes is:
      ");
46 disp(ia1);
47
48 ia0=-(3.1729*z2/(z0+z2));
49 printf("The zero sequence current in amperes is:");
50 disp(ia0);
51
52 ia2=-ia1*z0/(z0+z2);
53 printf("The negative sequence current in amperes is:

```

```

    ")
54 disp(ia2);
55
56 //b).LL fault
57
58 ia11=vf/(z1+z2);
59 printf("The positive sequence current in amperes is:
    ");
60 disp(ia11);
61
62 ia21=-ia11;
63 printf("The negative sequence current in amperes is:
    ");
64 disp(ia21);
65
66 ia=ia01+ia11+ia21;
67 printf("phase a fault current is:")
68 disp(ia);
69
70 mag=1;
71 theta=120;
72 [r,i]=d(mag,theta);
73 alpha=complex(r,i);
74
75 ib=ia01+alpha^2*ia11+alpha*ia21;
76 printf("phase b fault current in pu is:")
77 disp(ib);
78
79 ic=ia01+alpha*ia11+alpha^2*ia21;
80 printf("phase c fault current in pu is:")
81 disp(ic);

```

---

Scilab code Exa 18.3 To calculate voltage to neutral of the faulty phase

```
1 clear;
```

```
2 //clc ();
3 xz1g=(%i)*1.2;
4 xz1f=(%i)*1.0;
5
6 xz2g=(%i)*0.9;
7 xz2f=(%i)*1.0;
8
9 xz0g=(%i)*0.4;
10 xz0f=(%i)*3.0;
11
12 vf=11000/sqrt(3);
13 za0=xz0g +xz0f;
14 za1=xz1g +xz1f;
15 za2=xz2g +xz2f;
16
17 ia0=vf/(za0+za1+za2);
18 ia1=ia0;
19 ia2=ia0;
20
21 ifault=3*ia0;
22 va=vf-ia0*(xz0g+xz1g+xz2g);
23 printf("The voltage to neutral of the faulty phase
    is :%.2f V",va);
```

---

# Chapter 19

## Power system stability

Scilab code Exa 19.1 To find the steady state stability limit

```
1 clear;
2 //clc();
3 vs=132;
4 vr=vs;
5 A=0.97;
6 alpha=0.6;
7 B=60;
8 z=B;
9 bet=70;
10 C=0.001;
11 del=91;
12
13 prm=vs*vr/B -A*vr^2*cosd(bet-alpha)/B;
14 printf("the steady state stability limit is: %.2f MW\n",prm);
15
16 //shunt admittance neglected
17
18 y=0;
19 A1=1 +z*y/2;
20 alpha1=0;
```

```

21 D=A;
22 C=y*(1 +z*y/2);
23
24 prm=vs*vr/B -A1*vr^2*cosd(bet-alpha1)/B;
25 printf("the steady state stability limit is:%.2f MW\
n",prm);
26
27
28 //both shunt admittance and series resistance is
neglected
29
30 B1=B*sind(bet);
31 bet1=90;
32
33 prm=vs*vr/B1 -A1*vr^2*cosd(bet1-alpha1)/B1;
34 printf("the steady state stability limit is:%.2f MW\
n",prm);

```

---

Scilab code Exa 19.2 To determine teh steady state stability limit

```

1 clear;
2 //clc();
3
4
5 function [mag,theta]=c(r,i)
6     mag=sqrt(r*r + i*i)
7     theta=atand(i/r)
8 endfunction
9
10
11 previousprot = funcprot(0)
12 funcprot(0)
13
14
15 //a).inductor switch open

```

```

16 xdg=0.8;
17 xt=0.1;
18 xl=0.6;
19 xr=0.6;
20
21 e=1.2;
22 v=1;
23
24 x=xdg+xt+.5*xl+0.5*xl;
25 pm=e*v/x;
26 printf("the steady state stability limit with
    inductor switch S open is:%.2f pu\n",pm);
27
28
29 //b).inductor switch closed
30
31 z1=(xdg + xt +0.5*xl);
32 z2=0.5*xl;
33 z3=xr;
34 b=(%i)*z1 + (%i)*z2 +(((%i)*z1*(%i)*z2)/((%i)*z3));
35
36 B=imag(b);
37 x1=B;
38 pm=e*v/x1;
39 printf("the steady state stability limit with
    inductor switch S closed is:%.2f pu\n",pm);
40
41 //c).with the inductor replaced with a shunt
    capacitor of same per unit
42
43 z31=-0.6;
44 b1=(%i)*z1 + (%i)*z2 +(((%i)*z1*(%i)*z2)/((%i)*z31))
    ;
45 B1=imag(b1);
46 x2=B1;
47 pm2=e*v/x2;
48 printf("the steady state stability limit with the
    inductor replaced with a shunt capacitor is:%.2f

```

```

        pu\n",pm);
49
50 //d).when the shunt capaicator is replaced with a
    series capacitor
51
52 xc=-0.6;
53 x3=xdg+xt+x1+xc;
54 pm=e*v/x3;
55 printf("the steady state stability limit when the
    shunt capaicator is replaced with a series
    capacitor is:%.2f pu\n",pm);
56
57 //e).when the shunt inductor is replaced with a
    resistor
58
59 z11=xdg + xt +0.5*x1;
60 z3=1.5;
61
62 r=1;
63 i=z11/z3;
64 [mag,theta]=c(r,i);
65 A1=mag;
66 alpha1=theta;
67
68 b=((%i)*z11 + (%i)*z2 +((%i)*z11*(%i)*z2)/z3);
69
70 r=real(b);
71 i=imag(b);
72
73
74 [mag,theta]=c(r,i);
75 B1=mag;
76 bet2=theta;
77 pm=e*v/B1 - A1*v^2*cosd(bet2-alpha1)/B1;
78 printf("the steady state stability limit when the
    shunt inductor is replaced with a resistor is:%.2
    f pu\n",pm);

```

---

Scilab code Exa 19.4 To calculate the kinetic energy

```
1 clear;
2 //clc();
3 f=50;
4 s=60;
5 j=9000;
6 p=2;
7 bs=50;
8 n=120*f/p;
9 ke=0.5*j*(2*(%pi)*n/60)^2;
10 printf("The kinetic energy at rated speed is: %.0f MJ
    \n", ke/1000000);
11
12 //to calculate inertia constants M and H
13
14 H=ke/s;
15 printf("the inertia constants H is: %.3f MJ/MVA\n", H
    /1000000);
16
17 M=ke/(180*f*1000000);
18 printf("the inertia constants M is: %.3f MJ/elec.deg
    \n", M);
19
20 //the inertia constants
21
22 iner_const=M/bs;
23 printf("The inertia constant is: %.6f pu", iner_const)
    ;
```

---

Scilab code Exa 19.5 To calculate equivalent h constant at a base of 100MVA

```

1 clear;
2 //clc();
3 s1=500;
4 h1=4.6;
5 s2=1500;
6 s=100;
7 h2=3;
8 ke=s1*h1 + s2*h2;
9 H=ke/s;
10 printf("the inertia constants H is:%.0f MJ/MVA\n",H)
    ;

```

---

Scilab code Exa 19.6 To calculate whether the generator will remain in synchronism

```

1 clear;
2 //clc();
3
4 function y=z(x)
5 y=sin(x)
6 endfunction
7
8 exact=-2.5432;
9
10 pe=50;
11 pm=100;
12 p_inc=30;
13
14 del1=asin(pe/pm);
15
16 //at point b
17 pb=80;
18 del2=asin(pb/pm);
19
20 a1=pb*(0.927-0.523)-100*intg(0.523,0.927,z);
21

```

```

22 a2=100*intg(0.927,(3.14-0.927),z)-80*(3.14-2*del2);
23
24 if a1<a2 then
25     disp("the generator will remain stable");
26 else
27     disp("the system is unstable");
28 end

```

---

**Scilab code Exa 19.7** To estimate the sudden increase in generator output

```

1 clear;
2 //clc();
3
4 //at point a
5
6 pe=30;
7 pmax=60;
8 del1=asin(pe/pmax);
9
10 //by hit and trial method
11
12 del2=60.4;
13 p1=pmax*sind(del2);
14
15 per_load=p1-pe;
16 printf("The maximum permissible sudden increase of
    load is :%.2f MW",per_load);

```

---

**Scilab code Exa 19.8** To determine the stability of the system

```

1 clear;
2 //clc();
3

```

```

4  function y=z(x)
5  y=sin(x)
6  endfunction
7
8  exact=-2.5432;
9
10
11 pe=80;
12 pmax=200;
13 del1=asin(pe/pmax);
14
15 //at curve b
16
17 pmax2=100;
18 del2=asin(pe/pmax2);
19 delm=3.14-del2;
20
21 a1=pe*(del2-del1)- pmax2*intg(del1,del2,z);
22
23 a2=pmax2*intg(del2,delm,z)-pe*(delm-del2);
24
25 if a1<a2 then
26     disp("the generator will remain stable");
27
28     else
29         disp("the system is unstable");
30 end

```

---

Scilab code Exa 19.9 To determine the critical clearing angle

```

1  clear;
2  //clc();
3  ps=0.6;
4  pmax3=0.8;
5  pmax2=0.3333;

```

```

6 pmax1=1;
7
8 del0=asin(ps);
9 del2=asin(ps/pmax3);
10 delm=3.14-del2;
11 delc=acosd(((ps*(delm-del0)-pmax2*cos(del0)+ pmax3*
      cos(delm))/(pmax3-pmax2)));
12 printf("The critical clearing angle is:%.2f deg",
      delc);

```

---

Scilab code Exa 19.10 To determine the critical clearing angle

```

1 clear;
2 //clc();
3
4 //pre fault condtion
5
6 xg=0.3;
7 xl=0.5;
8 x=xg +xl*0.5;..// since the line reactances are in
      parallel
9 pe1=1;
10 e=1.4;
11 vc=1;
12 ps=1;
13 pmax1=e*vc/x;
14
15 del0=asin(ps/pmax1);
16
17 //after star delta conversion
18
19 xb=0.075;
20 xc=0.175;
21 xe=0.0525;
22 xa=0.375;

```

```

23
24 xac=xa+xc+(xa*xc/xe);
25
26 pmax2=e*vc/xac;
27
28 x1=xg+x1;
29 pmax3=e*vc/x1;
30
31 pe3=1;
32 del2=asin(pe3/pmax3);
33 delm=3.14-del2;
34
35 delc=acosd((ps*(delm-del0)-pmax2*cos(del0)+ pmax3*
    cos(delm))/(pmax3-pmax2));
36 printf("the critical clearing angle is:%.2f deg",
    delc);

```

---

Scilab code Exa 19.11 To plot the sqing curve

```

1 clear;
2 //clc();
3
4 H=2.7;
5 s=1;
6 fr=50;
7
8 del0=23.13;
9 pmax1=2.545;
10 pmax2=0.778;
11 pmax3=1.75;
12 ps=1;
13 del_del0=0;
14
15 M=H*s/(180*fr);
16 del_del=zeros(1,9);

```

```

17  pa=zeros(1,9);
18  del=zeros(1,9);
19  //for t=0-
20  pmax=pmax1;
21  //for t=0+;
22  pmax=pmax2;
23  p=ps-pmax2*sind(del0);
24  pa_avg=0.5*p;
25  pa(1)=pa_avg;
26  t=[.1:.9:9];
27  del(1)=del0;
28  del(2)=del_del0+8.33*pa_avg;
29
30  del(2)=del0+del(2);
31  for i=2:1:9
32      pa(i)=ps-.778*sind(del(i));
33      del_del(i+1)=del_del(i)+8.33*pa(i);
34      del(i+1)=del(i)+ del_del(i+1);
35  end
36
37  plot(t,del)
38  xlabel("time in secs");
39  ylabel('torque angle in degrees');

```

---

# Chapter 20

## Travelling waves

Scilab code Exa 20.1 to calculate the surge impedances and velocities of the line

```
1 clear;
2 //clc();
3 l1=2*10^(-3);
4 c1=0.01*10^(-6);
5
6 z1=sqrt(l1/c1);
7 printf("The surge impedance of the overhead line is:
   %.0f Ohm\n",z1);
8 v1=1/sqrt(l1*c1);
9 printf("The velocity of the wave in overhead line is
   :%.2f km/s\n",v1);
10
11 l2=0.25*10^(-3);
12 c2=0.102*10^(-6);
13
14 z2=sqrt(l2/c2);
15 printf("The surge impedance of the cable line is: %.2
   f Ohm\n",z2);
16 v2=1/sqrt(l2*c2);
17 printf("The velocity of the wave in cable is: %.2f km
   /s\n",v2);
```

```

18
19
20 ef=100*1000;
21 inc=ef/z1;
22
23 tau1=(2*z2)/(z2+z1);
24 et=tau1*ef;
25 printf("The transmitted voltage in the cable is:%.2f
        kV\n",et/1000);
26
27 tau11=(2*z1)/(z2+z1);
28 it=tau11*inc;
29 printf("The transmitted current in the cable is:%.2f
        A\n",it);
30
31 rhov1=(z2-z1)/(z1+z2);
32 er=rhov1*ef;
33 printf("The reflected voltage in the cable is:%.2f
        kV\n",er/1000);
34
35
36 rhoi1=(z1-z2)/(z1+z2);
37 ir=rhoi1*inc;
38 printf("The reflected current in the cable is:%.2f A
        \n",ir);
39
40 disp('the surge voltage has been changed to 100kV');
41
42
43 ef=100;
44 inc=ef*1000/z2;
45
46 tau2=(2*z1)/(z2+z1);
47 et=tau2*ef;
48 printf("The transmitted voltage in the cable is:%.2f
        kV\n",et);
49
50 tau22=(2*z2)/(z2+z1);

```

```

51 it=taui2*inc;
52 printf("The transmitted current in the cable is:%.2f
      A\n",it);
53
54 rhov2=(z1-z2)/(z1+z2);
55 er=rhov2*ef;
56 printf("The reflected voltage in the cable is:%.2f
      kV\n",er);
57
58 rhoi2=(z2-z1)/(z1+z2);
59 ir=rhoi2*inc;
60 printf("The reflected current in the cable is:%.2f A
      ",ir);

```

---

**Scilab code Exa 20.2** To find the voltage distribution

```

1 clear;
2 //clc();
3 z1=400;
4 z2=40;
5 ef=100;
6 voh=300000;...//velocity of overhead line
7 vc=140000;...//velocity of cable
8 rho1=(z2-z1)/(z2 + z1);
9 er=rho1*ef;
10 printf("The voltage transmitted into the cable is:%
      .2f kV\n",er);
11 tauv=(2*z2)/(z2+z1);
12 et=tauv*ef;
13 printf("The voltage transmitted into the cable is:%
      .2f kV",et);

```

---

**Scilab code Exa 20.3** to find the reflected voltage and current in the cable

```

1 clear;
2 //clc();
3 z1=100;
4 z2=600;
5 z3=1000;
6 ef=1000;
7 et=2*ef/((1/z2 + 1/z3)*100 +1);
8 er=et -ef;
9 printf("The reflected voltage in the line is:%.0f V\n",er);
10
11 i1=et/z2;
12 i2=et/z3;
13 i=i1 + i2;
14 printf("The current in the cable is:%.2f A\n",i);

```

---

Scilab code Exa 20.4 To determine from first principles the magnitudes of first an

```

1 clear;
2 //clc();
3
4 z2=50;
5 z1=400;
6 z3=400;
7 ef=10;
8
9 tau1=(2*z2)/(z1 + z2);
10 rho1=(z1-z2)/(z2+z1);
11 tau2=(2*z3)/(z3 + z2);
12 rho2=(z3-z2)/(z3+z2);
13 etb=tau1*tau2*ef;
14 printf("The transmitted part entering the line BC is
: %.2f kV\n",etb);
15
16 eta=rho1*rho2*tau1*ef;

```

```

17
18 etb1=rho1*rho2*tau1*tau2*ef;
19 printf("The second pulse entering the line BC is:%.2
    f kV",etb1);

```

---

**Scilab code Exa 20.5** To find the surge voltage distribution

```

1 clear;
2 //clc();
3
4 l=265*10^(-6);
5 c=0.165*10^(-6);
6
7 z2=sqrt(1/c);
8 z1=400;
9 ef=500;
10 z3=1000;
11 et1= (2*z2)*ef/(z1 + z2);
12 printf("The incident voltage in the cable is:%.0f kV
    \n",et1);
13
14 erb=(z3 - z2)*et1/(z3 + z2);
15 printf("The reflected voltage at the transformer end
    is:%.0f kV\n",erb);
16
17 vcd=et1+erb;
18 printf("The voltage at the junction is:%.0f kV",vcd)
    ;

```

---

**Scilab code Exa 20.6** To calculate the current voltage waves and the resistance ref

```

1 clear;
2 //clc();

```

```

3 z1=400;
4 z2=600;
5 zp=2000;
6 ef1=100;
7 ef2=(2/z1)*ef1/(1/z1 + 1/z2 + 1/zp);
8 er1=ef2-ef1;
9 printf("The voltage reflected in line1 is:%d kV\n",
    er1);
10
11 ir1=round(er1)/z1;
12 printf("The current reflected in line1 is:%.2f kV\n"
    ,ir1*1000);
13
14 zp=1/(-(1/z1)-(1/z2)+(2/z1));
15 printf("The resistance at the junction is:%.0f Ohm",
    zp);

```

---

Scilab code Exa 20.7 To calculate the voltage across load

```

1 clear;
2 //clc();
3
4 z=500;
5 r=10*10^(3);
6 c=0.005*10^(-6);
7 tau=c*r*z/(z + r);
8
9 ef=10;
10 t=5*10^(-6);
11 e=(2*r*ef/(z + r))*(1- exp(-t/tau));
12 printf("The voltage across load after 5us is:%.2f kV
    ",e);

```

---

Scilab code Exa 20.8 To find the maximum voltage

```
1 clear;
2 //clc();
3
4 z1=400;
5 z2=300;
6 z3=300;
7 z2=300;
8 zd=50;
9
10 et=2*(1/z1)/(1/z1 + 1/z2 + 1/z3);
11 et1=et*(2*zd/(z2 + zd));
12 t=3.1534;
13 v=250*(exp(-.05*t) - exp(-t));
14 e=et1*v;
15 printf("The maximum voltage is :%.2f kV",e);
```

---

# Chapter 22

## Corona

Scilab code Exa 22.1 to determine the disruptive critical voltage visual critical

```
1 clear;
2 //clc();
3 v=132;
4 f=50;
5 l=150;
6 d=19.53*10^(-3);
7 t=30;
8 p=750;
9 v_grad=21.1;
10 sur_fact=0.85;
11 l_sur_fact=0.72;
12 g_sur_fact=0.82;
13 D=3.8;
14
15 r=d/2;
16 del=(0.392*p)/(273 + t);
17 e=v_grad*100*sur_fact*r*del*log([D/r]);
18 printf("The disruptive critical voltage is: %.2f kV\n",e);
19
20 ev=v_grad*100*l_sur_fact*r*del*(1 +(0.0301/sqrt(r*
```

```

    del))) * log([D/r]);
21 printf("The visual critical voltage for local corona
    is: %.2f kV\n", ev);
22
23 evg=v_grad*100*g_sur_fact*r*del*(1 +(0.0301/sqrt(r*
    del))) * log([D/r]);
24 printf("The visual critical voltage for general
    corona is: %.2f kV", evg);

```

---

Scilab code Exa 22.2 to estimate the corona loss

```

1 clear;
2 //clc();
3 v=110;
4 f=50;
5 l=150;
6 d=10*10^(-3);
7 irr_fac=0.85;
8 v_grad=30;
9 p=750;
10 t=30;
11 D=2.5;
12
13 r=d/2;
14 del=(0.392*p)/(273 + t);
15 e=v_grad*100*irr_fac*r*del*log([D/r])/sqrt(2);
16
17 en=v/sqrt(3);
18 pc=(244/del)*(f+25)*(en-e)^(2)*sqrt(r/D)*l/100000;
19
20 tot_loss=3*pc;
21 printf("The total corona loss is: %.2f kW", tot_loss)
    ;
22 //difference in answer is due to rounding off of pc

```

---

# Chapter 24

## System neutral grounding

Scilab code Exa 24.1 to calculate reactance to neutralize the capacitance to earth

```
1 clear;
2 //clc();
3 f=50;
4 cap=1.2*10^(-6);
5 x1=1/(3*2*(%pi)*cap*f);
6 printf("The inductive reactance to neutralize 100
   percent of the capacitance is: %.2f Ohm\n",x1);
7
8 x11=x1/0.9;
9 printf("The inductive reactance to neutralize 90
   percent of the capacitance is: %.2f Ohm\n",x11);
10
11 x12=x1/0.8;
12 printf("The inductive reactance to neutralize 80
   percent of the capacitance is: %.2f Ohm",x12)
```

---

Scilab code Exa 24.2 To determine the inductance and kva rating

```
1 clear;
2 //clc();
3
4 l=80
5 cap=0.00914*10(-6)*1;
6 vl=132*1000;
7 vph=vl/sqrt(3);
8 f=50;
9 l=1/(3*(2*(%pi)*f)(2)*cap);
10 printf("The inductance is: %.2 f H\n",l)
11
12 il=vph/((2*(%pi)*f)*l);
13 kvar=vph*il/1000;
14 printf("The rating of the arc suppression coil is: %
    .2 f kVA",kvar)
```

---

# Chapter 25

## Tarrifs

Scilab code Exa 25.1 To determine the overall cost per kwh

```
1 clear;
2 //clc();
3 max_dem=80;
4 pf=0.45;
5 tar_md=750;
6 tar_en=1.1;
7 ann_ener_cons=max_dem*pf*8760;
8 ce=1.1*ann_ener_cons;
9
10 cf=tar_md*max_dem;
11 tot=ce+cf;
12 cost_per_kwh=tot/ann_ener_cons;
13 printf("the overall cost is :%.2f Rs",cost_per_kwh);
```

---

Scilab code Exa 25.2 To calculate the annual bill

```
1 clear;
2 //clc();
```

```

3
4 con_load=50;
5 max_dem=40;
6 t=8;
7 days=300;
8 tar_f=5000;
9 tar_md=800;
10 tar_kwh=1.15;
11
12 ene_con=con_load*t*days;
13 ann_bill=tar_f + tar_md*max_dem + tar_kwh*ene_con;
14 printf("the annual bill is :%.0f Rs",ann_bill);

```

---

**Scilab code Exa 25.3** To determine the cost of energy per kwh at the busbars

```

1 clear;
2 //clc();
3 op=960*10^(6);...//kWh generated per year
4 lf=0.55;
5 tar_fix=1000;
6 tar_kwh=0.4;
7
8 max_dem=op/(lf*8760);
9 ann_dem_cost=tar_fix*max_dem;
10 ann_ene_cost=op*0.4;
11 tot= ann_dem_cost+ ann_ene_cost;
12 cost=tot/op;
13 printf("the cost of energy per kwh at the busbars is
    :%.3f Rs",cost);
14 //difference in answer is due to the misprint in the
    calculation of total annual charges

```

---

**Scilab code Exa 25.4** to determine the energy consumption per year and the yearly b

```

1 clear;
2 //clc();
3 l1=100;
4 t1=9;
5 l2=125;
6 t2=6;
7 l3=50;
8 t3=7;
9 l4=5;
10 t4=2;
11 tar_md=800;
12 tar_kwh=1.3;
13
14 ene_per_day=l1*t1 + l2*t2+ l3*t3+ l4*t4;
15 ann_ene=ene_per_day*365;
16 printf("The annual energy consumption is :%.0f kWh\n"
    ,ann_ene);
17 ann_cost=ann_ene*tar_kwh;
18 max_dem=12;
19 ann_md=max_dem*tar_md;
20 tot=ann_cost + ann_md;
21
22 printf("the energy consumption per year and the
    yearly bill is :%.0f Rs",tot);

```

---

**Scilab code Exa 25.5** To determine the saving in energy cost

```

1 clear;
2 //clc();
3 ann_con=200000;
4 lf=0.4;
5 tar_md=750;
6 tar_kwh=1.15;
7 lf1=0.6;
8

```

```

9 avg_l=ann_con/8760;
10 max_dem=avg_l/lf;
11
12 max_dem1=avg_l/lf1;
13 red=max_dem - max_dem1;
14 sav=tar_md*red;
15 printf("The saving in energy cost is:%.2f Rs",sav);

```

---

**Scilab code Exa 25.6** To determine the total energy consumed and the total bill

```

1 clear;
2 //clc();
3 max_dem=150;
4 lf=0.65;
5 tar_md=900;
6 tar_kwh=1.3;
7 pf=0.82;
8
9 max_kva_dem=max_dem/pf;
10 ann_con=(max_dem)*lf*8760;
11 printf("the total energy consumed is:%.0f kWh\n",
    ann_con);
12 ann_cost=ann_con*tar_kwh;
13 ann_dem_cost=tar_md*max_kva_dem;
14 ann_bill=ann_cost + ann_dem_cost;
15 printf("The annual electricity bill is:%.0f Rs",
    ann_bill);

```

---

**Scilab code Exa 25.7** To determine which tariff is economical

```

1 clear;
2 //clc();
3 max_dem=100;

```

```

4 tar_md=800;
5 tar_kwh=1.3;
6 tar_kwh1=1.83;
7 t=3600;
8 lf=0.8;
9 avg_dem=max_dem*lf;
10 ann_ene=avg_dem*t;
11 ann_bill=(tar_md*max_dem + tar_kwh*ann_ene);
12
13 ann_bill1=tar_kwh1*ann_ene;
14
15 if (ann_bill>ann_bill1) then
16     disp("flat rate tarrif is better");
17 else
18     disp("two part tariff is better");
19
20 end

```

---

Scilab code Exa 25.8 to calculate the annual bill

```

1 clear;
2 //clc();
3 w_days=300;
4 t=8;
5 tar_mdh=75;
6 tar_kwh=1.15;
7
8 tar_md1=80;
9 tar_kwh1=1.43;
10
11 avg_load=400;
12 pf=0.8;
13 max_dem=500;
14 loss=0.04;
15 dep=0.15;

```

```

16 cost=900;
17
18 max_kva=max_dem/pf;
19 cap=max_kva/(1-loss);
20 cost_hv=cost*cap;
21 ann_int=dep*cost;
22 ann_max_dem=tar_mdh*12*max_kva;
23
24 ene_con=avg_load*w_days*t/(1-loss);
25
26 cost_ene_con=ene_con*tar_kwh;
27 tot_charges=ann_max_dem + cost_ene_con;
28 printf("The annual bill is:%.0f Rs\n",tot_charges);
29
30 //low voltage supply
31
32 ann_max_deml=tar_mdl*12*max_kva;
33 ann_ene_conl=avg_load*w_days*t;
34 cost_ene_con=ann_ene_conl*tar_kwhl;
35 tot_costl=ann_max_deml + cost_ene_con;
36 printf("The annual bill is:%.0f Rs",tot_costl);

```

---

Scilab code Exa 25.9 To find the number of units for which the tariff is economical

```

1 clear;
2 //clc();
3
4 tar_kwh=1.43;
5 flat_tar=1.83;
6 fixed=400;
7 units=fixed/(flat_tar-tar_kwh);
8 printf("the number of units for which the tariff is
    economical is:%.0f",units);

```

---

Scilab code Exa 25.10 to find the energy consumption per year

```
1 clear;
2 clc;
3
4 //c1=500+.9x;
5 //c2=1.83x-60;
6 //c1=c2;
7 x=(500+60)/(1.83-.90);
8 printf("The energy consumed per year is: %.2f kWh",
    round(x));
```

---

Scilab code Exa 25.11 to find the generation cost per kwh

```
1 clear;
2 //clc();
3 ins_cap=500*10^(6);
4 cap_cost=35000*10^(3);
5 dep=0.12;
6 fuel_con=0.85;
7 fuel_cost=0.8;
8 oper_cost=0.25;
9 p_load=475*10^(6);
10 lf=0.82;
11 avg_load=p_load*lf;
12 ene_gen=avg_load*8760/1000;
13 tot_inv=ins_cap*cap_cost/1000000;
14
15 ann_dep=dep*tot_inv;
16
17 fuel_con_yr=fuel_con*avg_load*8760/1000;
18
```

```
19 ann_fuel_cost=fuel_cost*fuel_con_yr;
20 other_cost=oper_cost*ann_fuel_cost;
21 gen_cost=(ann_dep + ann_fuel_cost + other_cost)/
    ene_gen;
22 printf("the generation cost per kwh is :%.2f Rs",
    gen_cost);
```

---

# Chapter 26

## Power factor improvement

Scilab code Exa 26.1 To calculate the value of capacitance

```
1 clear;
2 //clc();
3 i1=0.75;
4 v=240;
5 p=80;
6 q1=acosd(1/(3*i1));
7 q2=0;
8 ic=i1*cosd(q1)*(tand(q1)-tand(q2));
9 cap=ic/(v*2*(%pi)*50);
10 printf("The capacitance is :%.2f uF\n",cap*1000000);
11
12 t1=acosd(0.95);
13
14 ic1=i1*cosd(q1)*(tand(q1)-tand(t1));
15 cap1=ic1/(v*2*(%pi)*50);
16 printf("The capacitance is :%.2f uF\n",cap1*1000000);
```

---

Scilab code Exa 26.2 To calculate the capacitance kVAR and the new supply of current

```

1 clear;
2 //clc();
3 i1=20;
4 f=50;
5 v=230;
6 q1=acosd(0.75);
7 q2=acosd(0.9);
8
9 ic=i1*cosd(q1)*(tand(q1)-tand(q2));
10
11 cap=ic/(v*2*(%pi)*f);
12 printf("The capaciatnce is :%.2 f uF\n",cap*1000000);
13
14 qc=v*ic;
15 printf("The kVAr is :%.3 f kVAr\n",qc/1000);
16 i2=i1*cosd(q1)/cosd(q2);
17 printf("the new supply current is :%.2 f A",i2);

```

---

Scilab code Exa 26.3 to calculate the kVAr rating and capacitance per phase

```

1 clear;
2 //clc();
3
4 p=750/3;
5 v=400;
6 q1=acosd(0.7);
7 q2=acosd(0.95);
8 qcp=p*(tand(q1)-tand(q2));
9 printf("The kVAr is :%.3 f kVAr\n",qcp);
10 cap=qcp*1000/(v*v*2*(%pi)*50);
11 printf("The capacitance is :%.0 f uF\n",cap*1000000);

```

---

Scilab code Exa 26.4 To calculate the capacitance per phase

```

1 clear;
2 //clc();
3 op=40;
4 eff=0.85;
5 ip=op/eff;
6 p=ip/3;
7 v=400;
8
9 q1=acosd(0.72);
10 q2=acosd(0.98);
11
12 qcp=p*(tand(q1)-tand(q2));
13 cap=qcp*1000/(v*v*2*(%pi)*50);// capacitance in
    milli farad
14 printf("The capacitance is :%.2f uF\n",cap*1000000);

```

---

Scilab code Exa 26.5 To calculate current to the motor at full load current through

```

1 clear;
2 //clc();
3
4 op=7500;
5 eff=0.87;
6 ip=op/eff;
7 v=230;
8
9 p=ip;
10 pf=.75;
11 q1=acosd(pf);
12 q2=acosd(1);
13
14 i=p/(v*pf);
15 printf("the current is :%.2f A\n",i);
16 qcp=p*(tand(q1)-tand(q2));
17

```

```

18 ic=qcp/v;
19 cap=qcp*1000/(v*v*2*(%pi)*50); // capacitance in
    milli farad
20 printf("The capacitance is:%.2f uF\n",cap*1000);

```

---

Scilab code Exa 26.6 To calculate the capacitance per phase

```

1 clear;
2 //clc();
3 p=50/3;
4 q1=acosd(0.72);
5 q2=acosd(0.9);
6
7 qcp=p*(tand(q1)-tand(q2));
8 v=415;
9
10 ic=qcp/v;
11 cap=ic*1000/(v*2*(%pi)*50);.. // capacitance in milli
    farad
12 printf("The capacitance is:%.2f uF\n",cap*1000000);
13
14 //in delta connection
15
16 vp=v/sqrt(3);
17
18 icd=qcp/vp;
19 cap=icd*1000/(vp*2*(%pi)*50); // capacitance in milli
    farad
20 printf("The capacitance is:%.2f uF\n",cap*1000000);

```

---

Scilab code Exa 26.7 To calculate the leading kvar kva rating and power factor

```

1 clear;

```

```

2 //clc();
3 p1=500;
4 p2=100;
5 q1=acosd(.707);
6 q2=acosd(0.95);
7 lkvar=(p1*tand(q1) - (p1+p2)*tand(q2));
8 printf("The kVAr is :%.2f kVAr\n",lkvar);
9
10 kva=sqrt(p2*p2 + lkvar*lkvar);
11 printf("The kVA is :%.2f kVA\n",kva);
12
13 pf=(p2/kva);
14 printf("The power factor is :%.3f leading\n",pf);

```

---

Scilab code Exa 26.8 To calculate power factor leading kvar and kva

```

1 clear;
2 //clc();
3 vl=400;
4 il=36;
5 pf=0.8;
6
7 p1=sqrt(3)*vl*il*pf;
8 op=15*746;
9 eff=0.85;
10 p2=op/eff;
11
12 //as per ques 26.7
13 p1=19.953;
14 p2=12.979;
15 q1=acosd(0.8);
16 q2=acosd(0.92);
17 qm=((p1*tand(q1)) - (p1+p2)*tand(q2))/p2;
18 pf=cosd(atan(qm));
19 printf("The power factor is :%.4f leading\n",pf);

```

```

20
21
22 lkvar=p2*(qm);
23 printf("The kVAr is :%.3f kVAr\n",lkvar);
24
25
26 kva=sqrt(p2*p2 + lkvar*lkvar);
27 printf("The kVA is :%.0f kVA\n",kva);

```

---

Scilab code Exa 26.9 To calculate the total load kVA and power factor

```

1 clear;
2 //clc();
3 s=15;
4 sa=complex(s,0);
5
6 s1=40;
7 sb1=complex(s1*0.6,s1*0.8);
8 sb=conj(sb1);
9
10 s2=25;
11 sc1=complex(s2*0.8,s2*0.6);
12 sc=conj(sc1);
13
14 st=sa + sb + sc;
15
16 t_kvar=sqrt((real(st)^2) + (imag(st)^2));
17 printf("The kVA is :%.2f kVA\n",t_kvar);
18
19 pf=real(st)/t_kvar;
20 printf("The power factor is :%.4f lagging\n",pf);
21
22 cap=-imag(st);
23 printf("The capacitance is :%.0f kVAr leading\n",cap)
    ;

```

---

Scilab code Exa 26.10 To calculate the annual saving

```
1 clear;
2 //clc();
3
4 x=900;
5 p=2000;
6 q1=acosd(0.8);
7 q2=acosd(0.96);
8 smd=x*p*((1/cosd(q1))-(1/cosd(q2)));
9
10 y=0.15*1950;
11 ann_cost=y*p*(tand(q1)-tand(q2));
12 tot_ann=smd-ann_cost;
13 printf("The annual saving is :%.0f Rs",tot_ann);
```

---

Scilab code Exa 26.11 To determine the most economical pf and the kva rating

```
1 clear;
2 //clc();
3 p=900;
4 q1=acosd(0.65);
5 x=1000;
6 y=0.15*2000;
7 pf=sqrt(1-(y/x)^2);
8 printf("The power factor is :%.4f lagging\n",pf);
9
10 q2=acosd(pf);
11 qc=p*(tand(q1)-tand(q2));
12 printf("The capacitance is :%.0f uF\n",qc);
```

---

Scilab code Exa 26.12 To determine the annual savings

```
1 clear;
2 //clc();
3 p=800;
4 pf1=.72;
5 q1=acosd(pf1);
6 x=500;
7 y=160;
8 pf=sqrt(1-(y/x)^2);
9
10 q2=acosd(pf);
11 qc=p*(tand(q1)-tand(q2));
12
13 smd=x*p*((1/pf1)-(1/pf));
14
15 ann_cost=y*qc;
16 sav=smd-ann_cost;
17 printf("The annual saving is :%.0f Rs",sav);
```

---

Scilab code Exa 26.13 To estimate the limiting cost per kvar

```
1 clear;
2 //clc();
3 q1=round(acosd(0.707));
4
5 q2=round(acosd(0.866));
6
7 x=8000;
8
9 y=x*((cosd(q2)-cosd(q1))/sind(q1-q2));
10 printf("the limiting cost per kvar is :%.2f Rs",y);
```

11 //difference in answer is due to the approximation  
of decimals

---