Scilab Textbook Companion for Electric Power Distribution System Engineering by T. Gonen[1]

Created by
Manoj A
B.Tech, Third Year
Electrical Engineering
NIT, Karnataka
College Teacher
NA
Cross-Checked by
Chaitanya Potti

June 12, 2014

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

Book Description

Title: Electric Power Distribution System Engineering

Author: T. Gonen

Publisher: McGraw Hill, New York

Edition: 1

Year: 1986

ISBN: 0-07-023707-7

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes

List of Figures

Chapter 2

Load Characteristics

Scilab code Exa 2.1 To find the load curve

```
//To find the load curve
//Page 39
clc;
clear;

t=[0:1:24];
SL
  =[100,100,100,100,100,100,100,100,0,0,0,0,0,0,0,0,0,0,100,100,100

R
  =[200,200,200,200,200,200,200,300,400,500,500,500,500,500,500,500

C
  =[200,200,200,200,200,200,200,200,300,500,1000,1000,1000,1000,120

Tl=SL+R+C;

//To display the Load bar curve diagram
bar(t,Tl,0.5,'red')
```

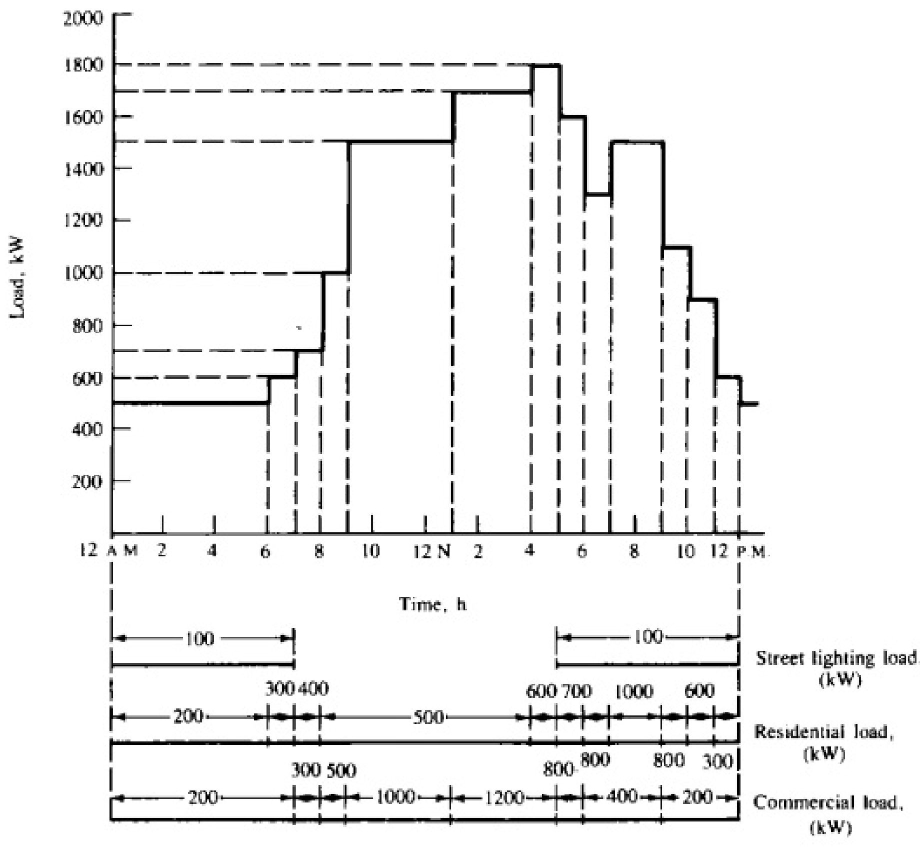

Figure 2.1: To find the load curve

```
title('Example 2.1','fontsize',3)
xlabel("Time in hrs",'fontsize',2)
ylabel("Load in kW",'fontsize',2)
```

Scilab code Exa 2.2 To determine the annual power loss

```
//To determine the annual power loss
//Page 46
clc;
clear;

Fls=0.15;
Ppl=80*(10^3); //Power Loss at peak load.

Avgpl=Fls*Ppl; //Average Power Loss
TAELCu=Avgpl*8760; //Total annual energy loss

printf('a) The average annual power loss = %g kW\n'
    ,(Avgpl/1000))
printf(' b) The total annual energy loss due to the
    copper losses of the feeder circuits = %g kWh\n'
    ,(TAELCu/1000))
```

Scilab code Exa 2.3 To determine the diversified demand

```
//To determine the diversified demand
//Page 47
clc;
clear;

TCDi=[9,9,9,9,9,9]; //Load for each house all in
    kilowatt
DFi=0.65; //Demand factor
```

```
Fd=1.1; //Diversity factor

Dg=sum(TCDi)*DFi/Fd;

printf('The diversified demand of the group on the distribution transformer is %g kW\n',Dg)
```

Scilab code Exa 2.4 To determine copper losses of the feeder

```
//To determine copper losses of the feeder
//Page 48
clc;
clear;

SP=3000*(10^3); //System peak in kVA per phase
Cl=0.5/100; //Percentage of copper loss
I2R= Cl*SP; //Copper loss of the feeder per phase
I2R3=3*I2R; //Copper losses of the feeder per 3 phase

printf('a) The copper loss of the feeder per phase = %g kW\n',(I2R/1000))
printf(' b) The total coper losses of the feeder per three phase = %g kW\n',(I2R3/1000))
```

Scilab code Exa 2.5 To determine the diversity load diversity and coincidence factor

```
//To determine the diversity load diversity and coincidence factor
//Page 48
```

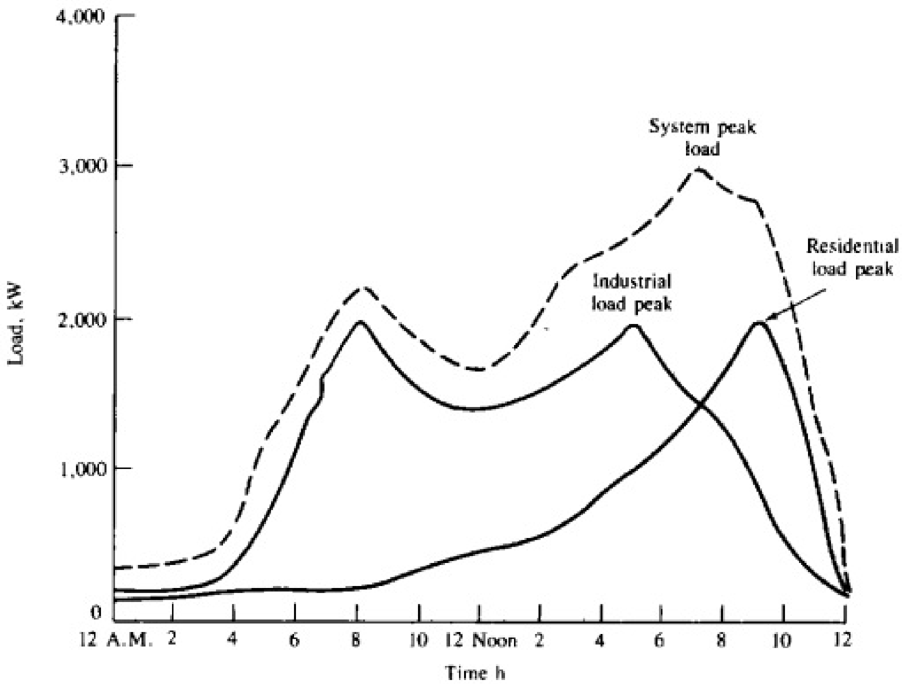

Figure 2.2: To determine the diversity load diversity and coincidence factor

```
clc;
clear;

Pi=2000*(10^3); //Peak for industrial load
Pr=2000*(10^3); //Peak for residential load
Dg=3000*(10^3); //System peak load as specified in the diagram
P=[Pi,Pr]; //System peaks for various loads

Fd= sum(P)/Dg; //Diversity factor
LD= sum(P)-Dg; //Load diversity factor
Fc=1/Fd; // Coincidence factor

printf('a) The diversity factor of the load is %g\n',Fd)
printf(' b) The load diversity of the load is %g kW\n',(LD/1000))
printf(' c) The coincidence factor of the load is %g\n',Fc)
```

Scilab code Exa 2.6 To determine the class distribution factors

```
//To determine the class distribution factors
//Page 50
//Refer diagram of the first example of this chapter
clc;
clear;

Ps=100; //Peak load for street lighting in kW
Pr=1000; //Peak load for Residential in kW
Pc=1200; //Peak Commercial load in kW
P=[Ps,Pr,Pc] //Peaks of various loads

Ls5=0; //Street lighting load at 5.00 PM in kW
Lr5=600; //Residential load at 5.00 PM in kW
```

```
Lc5=1200; //Commercial Load at 5.00 PM in kW

Cstreet=Ls5/Ps;
Cresidential=Lr5/Pr;
Ccommercial=Lc5/Pc;
C=[Cstreet,Cresidential,Ccommercial]; //Class
    distribution for various factors

Fd=(sum(P))/(sum(P*C'));
Dg=(sum(P*C'));
Fc=1/Fd;

printf('a) The class distribution factors factor of
    :\n')
printf(' i) Street lighting = %g\n ii) Residential =
     %g\n iii) Commercial =%g\n',C(1),C(2),C(3))
printf(' b) The diversity factor for the primary
    feeder = %g\n',Fd)
printf(' c) The diversified maximum demand of the
    load group = %g kW\n',Dg)
printf(' d) The coincidence factor of the load group
     = %g\n',Fc)
```

Scilab code Exa 2.7 To determine the annual average power demand

```
//To determine the annual average power demand
//Page 55
clc;
clear;

printf('Assuming a monthly load curve as shown in
    the figure attached to this code\n')

```

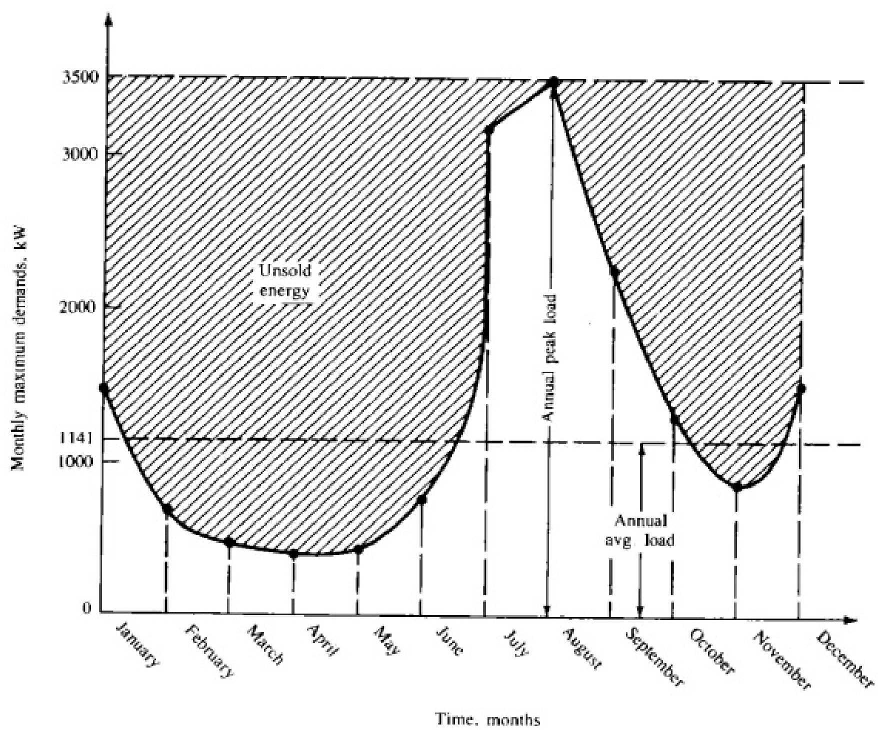

Figure 2.3: To determine the annual average power demand

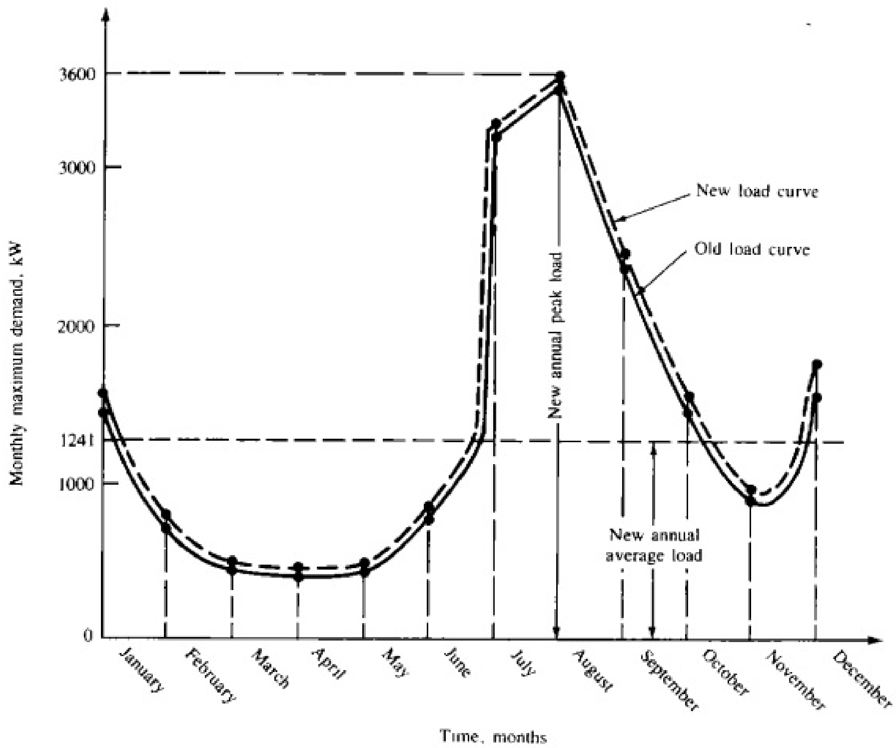

Figure 2.4: To determine the annual load factor on the substation

```
TAE=10^7; // Total annual energy in kW
APL=3500; //Annual peak load in kW
Pav= TAE/8760; //Annual average power demand
Fld= Pav/APL; //Annual load factor

printf('a) The annual power demand is %g kW\n',Pav)
printf('b) The annual load factor is %g\n',Fld)
printf('The unsold energy, as shown in the figure is
    a measure of capacity and investment cost.
   Ideally it should be kept at a minimum\n')
```

Scilab code Exa 2.8 To determine the annual load factor on the substation

```
//To determine the annual load factor on the
   substation
//Page 57
clc;
clear;

printf('Assuming a monthly load curve as shown in
   the figure attached to this code\n')

Nl=100; //100% percent load to be supplied
TAE=10^7; // Total annual energy in kW
APL=3500; //Annual peak load in kW
Pav= TAE/8760; //Annual average power demand
Fld= (Pav+Nl)/(APL+Nl); //Annual load factor
Cr=3;//Capacity cost
Er=0.03;//Energy cost
ACC=Nl*12*Cr; //Additional capacity cost per kWh
AEC=Nl*8760*Er; //Additional energy cost per kWh
TAC=ACC+AEC; //Total annual cost


printf('a) The new annual load factor on the
   substation is %g\n',Fld)
printf('b) The total annual additional costs to NL&
   NP to serve this load is $%g\n',TAC)
```

Scilab code Exa 2.9 To detemine the annual loss factor

```
//To detemine the annual loss factor
//Page 58
```

```
clc;
clear;

TAE=5.61*(10^6); //Total annual energy in kW
APL=2000; //Annual peak load in kW
Lc=0.03; //Cost of energy per kWh in dollars
Plp=100; //Power at peak load in kW

Fld=TAE/(APL*8760);
Fls= (0.3*Fld)+(0.7*(Fld^2));
AvgEL=Fls*Plp; //Average energy loss
AEL=AvgEL*8760; //Annual energy loss
Tlc=AEL*Lc; //Cost of total annual copper loss

printf('a) The annual loss factor is %g\n',Fls)
printf(' b) The annual copper loss energy is %g kWh and the cost of total annual copper loss is $%g\n',AEL,Tlc)
```

Scilab code Exa 2.10 To calculate thirty min annual maximum demand

```
//To calculate thirty min annual maximum demand
//Page 59
clc;
clear;

Fd=1.15;
Pi=[1800,2000,2200]; //Demands of various feeders in kW (Real Power)
PF=[0.95,0.85,0.90]; //Power factor of the respective feeders
Dg=sum(Pi)/Fd;
P=Dg;
theta=acosd(PF);

```

```
Q=sum(Pi*(tand(theta))')/Fd;
S=sqrt((P^2)+(Q^2));
LD=sum(Pi)-Dg;

//Transformer sizes
Tp=[2500,3750,5000,7500];
Ts=[3125,4687,6250,9375];

Ol=1.25; //Maximum overload condition
Eol=Ts*Ol; //Overload voltages of the transformer
Ed=abs(Eol-S); // Difference between the overload values of the transformers and the P value of the system

[A,k]=gsort(Ed); // To sort the differences and choose the best match

T=[Tp(k(4)),Ts(k(4))]; //Suitable transformer

g=poly(0,'g');
X=(1+g)-nthroot(2,10); //To find out the fans on rating
R=roots(X);
g=R*100;

n=poly(0,'n');
Sn=9375; // Rating of the to be installed transformer
// Equation (1+g)^n * S = Sn
// a=(1+g)
// b=Sn/S

a=1+R;
b=Sn/S;
n=log(b)/log(a);

printf('a) The 30 mins annual maximum deman on the substation transformer are %g kW and %g kVA
```

```
respectively\n',P,S)
printf(' b) The load diversity is %g kW\n',LD)
printf(' c) Suitable transformer size for 25 percent short time over loads is %g/%g kVA\n',T(1),T(2))
printf(' d) Fans on rating is %g percent and it will loaded for %g more year if a 7599/9375 kVA transformer is installed\n',g,ceil(n))
```

Scilab code Exa 2.11 To determine the Thirty min maximum diversified

```
//To determine the Thirty min maximum diversified
//Page 62
clc;
clear;

printf('NOTE\n\n')
printf('The figure 1 attached along with this code is the Maximum diversified 30- min demand characteristics of various residential loads;\n A = Clothes dryer; D = range; E = lighting and miscellaneous appliances; J = refrigerator\n (Only the loads required for this problem have been mentioned)\n \n')

Ndt=50; //Number of distribution transformers
Nr=900; //Number of residences

//When the loads are six.
PavMax6=[1.6,0.8,0.066,0.61]; //Average Maximum diversified demands (in kW) per house for dryer, range, refrigerator, for lighting and misc aapliances respectively according to the figure 1 attached with code.

Mddt= sum(6*PavMax6); //30 min maximum diversified
```

```
    demand on the distribution transformer

//When the loads are 900.
PavMax900=[1.2,0.53,0.044,0.52]; ////Average Maximum
     diversified demands (in kW) per house for dryer,
     range, refreigerato, for lighting and misc
    aapliances respectively according to the figure 1
     attached with code.

Mdf=sum(Nr*PavMax900); //30 min maximum diversified
    demand on the feeder

//From the figure 2 attached to this code
Hdd4=[0.38,0.24,0.9,0.32]; //Hourly variation factor
     at time 4 PM for dryer, range, refrigerator,
    lighting and misc appliances
Hdd5=[0.30,0.80,0.9,0.70]; //Hourly variation factor
     at time 5 PM for dryer, range, refrigerator,
    lighting and misc appliances
Hdd6=[0.22,1.0,0.9,0.92]; //Hourly variation factor
    at time 6 PM for dryer, range, refrigerator,
    lighting and misc appliances

Thdd4=(6*PavMax6)*Hdd4'; //Gives the total hourly
    diversified demand in kW at time 4 PM
Thdd5=(6*PavMax6)*Hdd5'; //Gives the total hourly
    diversified demand in kW at time 5 PM
Thdd6=(6*PavMax6)*Hdd6'; //Gives the total hourly
    diversified demand in kW at time 6 PM

printf(' a) The 30 min maximum diversified demand on
     the distribution transformer = %g kW\n',Mddt)
printf(' b) The 30 min maximum diversified demand on
     the distribution transformer = %g kW\n',Mdf)
printf(' c) The total hourly diversified demands at
    :\n')
printf(' i) 4.00 PM = %g kW\n',Thdd4)
printf(' ii) 5.00 PM = %g kW\n',Thdd5)
```

```
printf(' iii) 6.00 PM = %g kW\n',Thdd6)
```

Scilab code Exa 2.12 To find monthly load factor Rating of distribution transformer monthly bill

```
//To find monthly load factor Rating of distribution
    transformer monthly bill
//Page 72
clc;
clear;

T=730; //Average monthly time in hrs
Pla=22; //Peak Load for consumer A in kW
Plb=39; //Peak load for consumer B in kW
MEC=[0.025,0.02,0.015]; //Monthly Energy charges in
   cents/kWh according to the units consumed
Uc=[1000,3000,3000]; //Units consumption according
   to the Energy charges
MDC=2; //Monthly demand charge in dollars/kW

Pa=7000; //Units served to Consumer A in kWh
Pb=7000; //Units served to Consumer B in kWh

//Power factors
Pfa=0.9; // Lag
Pfb=0.76; //Lag

//Monthly Load factors
Flda=Pa/(Pla*T);
Fldb=Pb/(Plb*T);

```

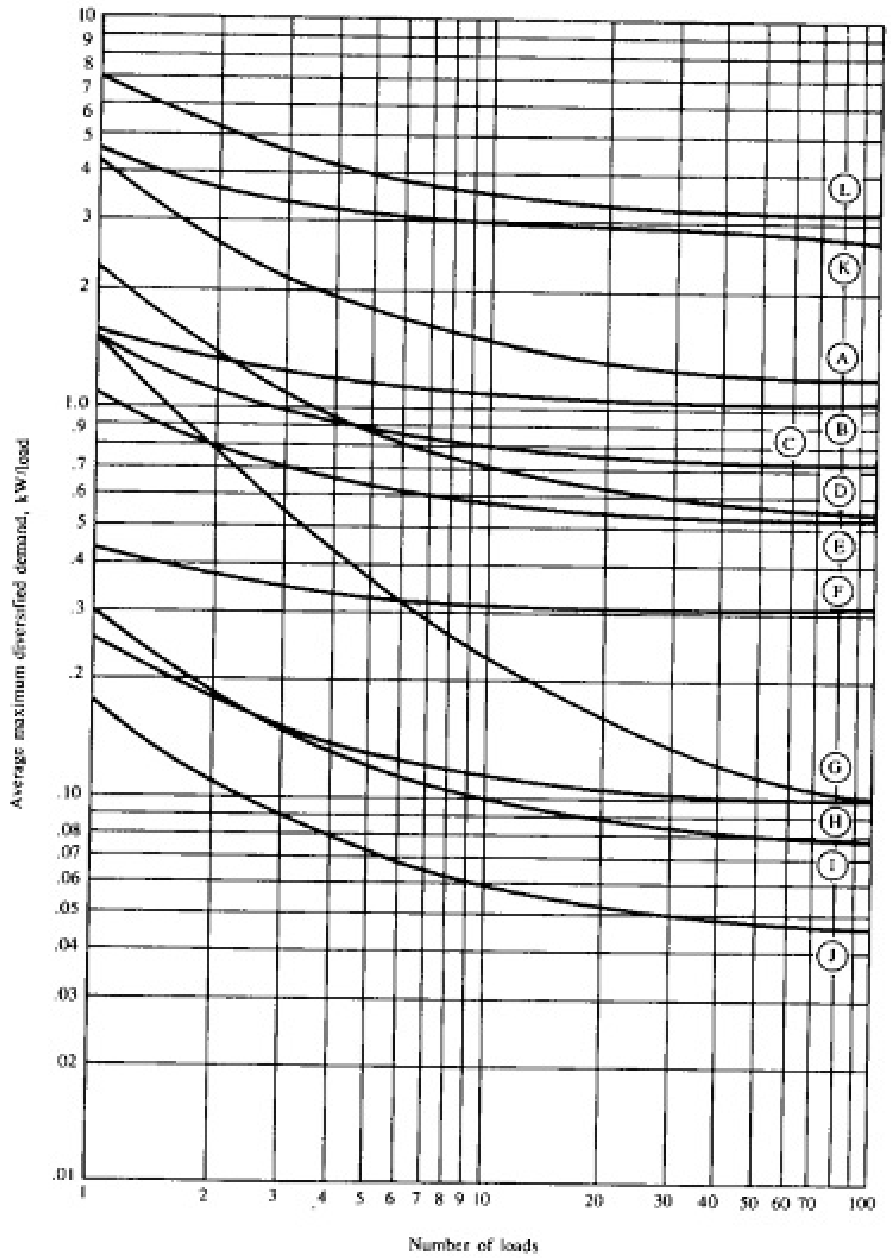

Figure 2.5: To determine the Thirty min maximum diversified

Hour	Lighting and misc.	Refrig-erator	Home freezer	Range	Air-condi-tioning*	Heat pump*		House* heating	Water heater†			Clothes§ dryer
									OPWH‡			
						Cooling season	Heating season		Both elements re-stricted	Only bottom elements re-stricted	Uncon-trolled	
12 A.M.	0.32	0.93	0.92	0.02	0.40	0.42	0.34	0.11	0.41	0.61	0.51	0.03
1	0.12	0.89	0.90	0.01	0.39	0.35	0.49	0.07	0.33	0.46	0.37	0.02
2	0.10	0.80	0.87	0.01	0.36	0.35	0.51	0.09	0.25	0.34	0.30	0
3	0.09	0.76	0.85	0.01	0.35	0.28	0.54	0.08	0.17	0.24	0.22	0
4	0.08	0.79	0.82	0.01	0.35	0.28	0.57	0.13	0.13	0.19	0.15	0
5	0.10	0.72	0.84	0.02	0.33	0.26	0.63	0.15	0.13	0.19	0.14	0
6	0.19	0.75	0.85	0.05	0.30	0.26	0.74	0.17	0.17	0.24	0.16	0
7	0.41	0.75	0.85	0.30	0.41	0.35	1.00	0.76	0.27	0.37	0.46	0
8	0.35	0.79	0.86	0.47	0.53	0.49	0.91	1.00	0.47	0.65	0.70	0.08
9	0.31	0.79	0.86	0.28	0.62	0.58	0.83	0.97	0.63	0.87	1.00	0.20
10	0.31	0.79	0.87	0.22	0.72	0.70	0.74	0.68	0.67	0.93	1.00	0.65
11	0.30	0.85	0.90	0.22	0.74	0.73	0.60	0.57	0.67	0.93	0.99	1.00
12 noon	0.28	0.85	0.92	0.33	0.80	0.84	0.57	0.55	0.67	0.93	0.98	0.98
1	0.26	0.87	0.96	0.25	0.86	0.88	0.49	0.51	0.61	0.85	0.86	0.70
2	0.29	0.90	0.98	0.16	0.89	0.95	0.46	0.49	0.55	0.76	0.82	0.65
3	0.30	0.90	0.99	0.17	0.96	1.00	0.40	0.48	0.49	0.68	0.81	0.63
4	0.32	0.90	1.00	0.24	0.97	1.00	0.43	0.44	0.33	0.46	0.79	0.38
5	0.70	0.90	1.00	0.80	0.99	1.00	0.43	0.79	0	0.09	0.75	0.30
6	0.92	0.90	0.99	1.00	1.00	1.00	0.49	0.88	0	0.13	0.75	0.22
7	1.00	0.95	0.98	0.30	0.91	0.98	0.51	0.76	0	0.19	0.80	0.26
8	0.95	1.00	0.98	0.12	0.79	0.73	0.60	0.54	1.00	1.00	0.81	0.20
9	0.85	0.95	0.97	0.09	0.71	0.72	0.54	0.42	0.84	0.98	0.73	0.18
10	0.72	0.88	0.96	0.05	0.64	0.53	0.51	0.27	0.67	0.77	0.67	0.10
11	0.50	0.88	0.95	0.04	0.55	0.49	0.34	0.23	0.54	0.69	0.59	0.04
12 P.M.	0.32	0.93	0.92	0.02	0.40	0.42	0.34	0.11	0.44	0.61	0.51	0.03

* Load cycle and maximum diversified demand are dependent upon outside temperature, dwelling construction and insulation, among other factors.
† Load cycle and maximum diversified demands are dependent upon tank size, and heater element rating; values shown apply to 52-gal tank, 1500- and 1000-W elements.
‡ Load cycle dependent upon schedule of water heater restriction.
§ Hourly variation factor is dependent upon living habits of individuals; in a particular area, values may be different from those shown.
Source. From [2].

Figure 2.6: To determine the Thirty min maximum diversified

```
//Continous kilovoltamperes for each distribution
   transformer
Sa=Pla/Pfa;
Sb=Plb/Pfb;

//Ratings of the distribution transformers needed
Ta=round(Sa/5)*5;
Tb=round(Sb/5)*5;

//Billing Charges
//For Consumer A
Mbda=Pla*(0.85/Pfa); // Monthly billing demand
Mdca=Mbda*MDC; //Monthly demand charge
//Since the units served are 7000 it can be split
   according to the rates  as 1000, 3000, 3000
   excess units.
Uca=Uc; //Units consumption by A
Meca=MEC*Uca'; //Monthly energy charge
Tmba=Meca+Mdca; //Total monthly bill

//For Consumer B
Mbdb=Plb*(0.85/Pfb); // Monthly billing demand
Mdcb=Mbdb*MDC; //Monthly demand charge
//Since the units served are 7000 it can be split
   according to the rates  as 1000, 3000, 3000
   excess units.
Ucb=Uc; //Units consumption by B
Mecb=MEC*Ucb'; //Monthly energy charge
Tmbb=Mecb+Mdcb; //Total monthly bill

//To find the capacitor size
Q1=Pb*tand(acosd(Pfb)); //For original power factor
Q2=Pb*tand(acosd(0.85)); //For new power factor

dQ=(Q1-Q2)/T; //Capacitor size

//For new power factor
//For Consumer B
```

```
Mbdbn=Plb*(1); // Monthly billing demand
Mdcbn=Mbdbn*MDC; //Monthly demand charge
//Since the units served are 7000 it can be split according to the rates as 1000, 3000, 3000 excess units.
Ucbn=Uc; //Units consumption by B
Mecbn=MEC*Ucbn'; //Monthly energy charge
Tmbbn=Mecbn+Mdcbn; //Total monthly bill

Saving=abs(Tmbbn-Tmbb); //Saving due to capacitor installation
Ci=30; // Cost of capacitor in dollar per kVAr
Cc=Ci*dQ; //The cost of the installed capacitor
PP=Cc/Saving; //Payback Period
PPr=90/Saving; //Realistic Payback period

printf('a) Monthly load factor for :\n')
printf(' i) Consumer A = %g\n',Flda)
printf(' ii) Consumer B =%g\n',Fldb)
printf(' b) Rating of the each of the distribution transformer:\n')
printf(' i) A = %g kVA\n',Ta)
printf(' ii) B = %g kVA\n',Tb)
printf(' c) Monthly bil for:\n')
printf(' i) Consumer A = $%g\n',Tmba)
printf(' ii) Consumer B = $%g\n',Tmbb)
printf(' d) The capacitor size required is %g kVAr\n',dQ)
printf(' e) Payback period:\n')
printf(' i) Calculated : %g months\n',ceil(PP))
printf(' ii) Realistic as capacitor size available is 3 kVAr : %g months\n',ceil(PPr))
```

Scilab code Exa 2.13 To determine the instantaneous demands and the average demand

```
//To determine the instantaneous demands and the
   average demand
//Page 84
clc;
clear;

Kh=7.2; //Meter constant
Kr1=32; //Revolutions of the disk in the first
    reading
Kr2=27; //Revolutions of the disk in the second
    reading
T1=59; //Time interval for revolutions of disks for
     the first reading
T2=40; //Time interval for revolutions of disks for
     the second reading

// Self contained watthour meter; D = (3.6*Kr*Kh)/T

deff('y=Id(a,b)','y=((3.6*Kh*a)/b)'); //Function to
     calculate the instaneous demand

D1=Id(Kr1,T1);
D2=Id(Kr2,T2);
Dav=(D1+D2)/2;

printf('The instantenous demands are %g kW and %g
   kW for reading 1 and 2 and the average demand is
    %g kW\n',D1,D2,Dav)
```

Scilab code Exa 2.14 To determine instantaneous demands and average demand for transformer type

```
//To determine instantaneous demands and average
   demand for transformer type
//Page 84
```

```
clc;
clear;

//For a transformer type watthour meter; D = (3.6*Kr
    *Kh*CTR*PTR)/T
CTR=200;
PTR=1;
Kh=1.8;
Kr1=32; //Revolutions of the disk in the first
    reading
Kr2=27; //Revolutions of the disk in the second
    reading
T1=59; //Time interval for revolutions of disks for
     the first reading
T2=40; //Time interval for revolutions of disks for
     the second reading
deff('y=Id(a,b)','y=((3.6*Kh*a*CTR*PTR)/b)'); //
    Function to calculate the instaneous demand

D1=Id(Kr1,T1);
D2=Id(Kr2,T2);
Dav=(D1+D2)/2;

printf('The instantenous demands are %g kW and %g
    kW for reading 1 and 2 and the average demand is
     %g kW\n',D1,D2,Dav)
```

Scilab code Exa 2.15 To determine watt VAr and VA demands

```
//To determine watt VAr and VA demands
//Page 85
clc;
clear;

Kh=1.2;
```

```
CTR=80;
PTR=20;
//Revolutions of the disk in a watthour meter for 1 and 2 readings respectively
Kr1=20;
Kr2=30;
//Revolutions of the disk in a VArhour meter for 1 and 2 readings respectively
Kr3=10;
Kr4=20
//Time interval for revoltion of disks in watthour meter for 1 and 2 readings respectively
T1=50;
T2=60;
//Time interval for revoltion of disks in VArhour meter for 1 and 2 readings respectively
T3=50;
T4=60;

deff('y=Id(a,b)','y=((3.6*Kh*a*CTR*PTR)/b)'); // Function to calculate the instaneous demand

//Instantaneous kilowatt demands for readings 1 and 2
D1=Id(Kr1,T1);
D2=Id(Kr2,T2);

//Instantaneous kilovar deamnds for readings 1 and 2
D3=Id(Kr3,T3);
D4=Id(Kr4,T4);

Davp=(D1+D2)/2; //Average kilowatt demand
Davq=(D3+D4)/2; //Average kilovar demand

Dav=sqrt((Davp^2)+(Davq^2)); //Average kilovoltampere demand

printf('a) The instantaneous kilowatt hour demands
```

```
   for readings 1 and 2 are %g kW and %g kW
   respectively\n',D1,D2)
printf(' b) The average kilowatt demand is %g kW\n',
   Davp)
printf(' c) The instantaneous kilovar hour demands
   for readings 1 and 2 are %g kVAr and %g kVAr
   respectively\n',D3,D4)
printf(' d) The average kilovar demand is %g kVAr\n'
   ,Davq)
printf(' e) The average kilovoltampere demand is %g
   kVA\n',Dav)
```

Chapter 3

Application of Distribution Transformers

Scilab code Exa 3.1 To Evaluate all the required impedances of a 25kVA Transformer

```
//To Evaluate all the required impedances of a 25kVA
    Transformer
//Page 118
clc;
clear;

S=25*(10^3); //Rating of the transformer in VA
//Values in per unit
Rt=0.014; //Resistance of Transformer
Xt=0.012; //Reactance of transformer
Vh=7200; //High Voltage End in V
Vx=120; // Low Voltage End in V
Rb=(Vh^2)/S; //Base Value of Resistance
//Accroding to Lloyd's Formula

Zhx12=(1.5*Rt)+(%i*1.2*Xt); //Impedance referred to
   HV side when the winding x2x3 is shorted

```

```
n=Vh/Vx; //Turns Ratio

Zhx13=Rt+(%i*Xt); //Use of Entire low voltage winding

//Impedances of the required terms in pu
A=(2*Zhx13)-Zhx12;
B=((2*Zhx12)-(2*Zhx13))/(n^2);
C=B;

//Angle of Impedances
ta=atand(imag(A)/real(A));
tb=atand(imag(B)/real(B));

printf('\nThe Circuit impedances on the high voltage side is %g/_%g ohm\n',abs(A*Rb),ta)
printf('Each of the Circuit impedances on the low voltage side is %g/_%g ohm\n',abs(B*Rb),tb)
```

Scilab code Exa 3.2 To determine the fault current in the distribution transformer

```
//To determine the fault current in the distribution transformer
//Page 119
clc;
clear;

//Impedances from the previous example
Zh=24.6437*exp(%i*53.9*%pi/180);
Zl=8.525*(10^-3)*exp(%i*18.9*%pi/180);
//Voltages
Vh=7200; //High End
Vx=120; // Low End
S=25*1000; //Transformer Rating in VA
```

```
N=Vh/Vx; //Turns Ratio

//R of service drop is zero //Line to Neutral
    Currents
IfLVn=Vx/(Zl+((1/(N^2))*Zh)); //Secondary Fault
    Current
IfHVn=IfLVn/N; //Primary Fault Current

//R of service drop is zero //Line to Line Currents
Nl=Vh/(2*Vx); //New Truns Ratio
IfLVl=2*Vx/((2*Zl)+((1/(Nl^2))*Zh)); //Secondary
    Fault Current
IfHVl=IfLVl/Nl; //Primary Fault Current

printf('\na) The Magnitude of Line to Neutral Fault
    Currentson HV and LV when R of service \ndrop is
    zero are %g A and %g A respectively\n',abs(IfHVn)
    ,abs(IfLVn))
printf('b) The Magnitude of Line to Line Fault
    Currentson HV and LV when R of service \ndrop is
    zero are %g A and %g A respectively\n',abs(IfHVl)
    ,abs(IfLVl))
printf('c) The Minimum Allowable interrupting
    capacity for circuit breaker is\nconnected to the
     LV is %g A\n',abs(IfLVn))
```

Scilab code Exa 3.3 To determine the service drop and the length of the cable

```
//To determine the service drop and the length of
    the cable
//Page 121
clc;
clear;

```

```
Vx=120; //Low End Voltage
//When Service drop is Zero
IfLVn=8181.7*exp(-1*%i*34.3*%pi/180); //Line to Neutral Vault Current
IfLVl=5649*exp(-1*%i*40.6*%pi/180); //Line to Line Fault Current

Ral4=2.58; //#4 AWG Aluminium Conductor Resistance per mile
Ralinf=1.03; //#1/0 AWG Aluminium Conductor Resistance per mile

//Impedances when Service drop is zero, suffix l denotes line to line
//Suffix n denotes line to neutral
Zl0=(2*Vx)/IfLVl;
Zn0=(Vx)/IfLVn;

//When there is R service drop
//Magnitudes of Line to Line and Line to Earth fault currents are equal

R=poly(0,'R'); //Variable Value
//Effective Impedances
Zl=Zl0+(2*R);
Zn=Zn0+(2*R);
//Fault Currents
Ifl=2*Vx/Zl;
Ifn=Vx/Zn;
//Magnitudes of Currents
MIfl=abs(Ifl(2))/abs(Ifl(3));
MIfn=abs(Ifn(2))/abs(Ifn(3));
DI=MIfl-MIfn;
X=DI(2); //Polynomial Equation to find 'R'

R=roots(X); //Numerical Value

//The Magnitude of R found is Wrong in the Textbook
```

```
//Length of service drop cable
SDL4=R/Ral4;
SDLinf=R/Ralinf;

printf('\na) The Value of Service drop in the Cable
    is %g ohm\n',R)
printf('b) The Length of service drop cable for:\n')
printf('i) #4 AWG Conductor is %g miles\n',SDL4)
printf('ii) #1/0 AWG Conductor is %g miles\n',SDLinf
    )

//Length is printed in Miles
```

Scilab code Exa 3.4 To determine the maximum load carried by the transformer

```
//To determine the maximum load carried by the
    transformer
//Page 122
clc;
clear;

//Transformer Ratings in kVA
Sr1=250;
Sr2=500;

//percentage impedances
Zr1=2.4;
Zr2=3.1;

//Ratio of Maximum Loads
R=Sr1*Zr2/(Sr2*Zr1);

//If 500 kVA is chosen as the full load transformer,
```

```
Transformer 1 becomes overloaded
SL1=Sr1; //To Avoid OverLoading of transformer 1

SL2=SL1/R; //Maximum Load on transformer 2

Tl=SL1+SL2; //Total Load without overloading

printf('The Maximum Load Carried without overloading any of the transformer is %g kVA\n',Tl)
```

Scilab code Exa 3.5 To Determine the Transformer parameters for a 3 phase load of 200kVA

```
//To Determine the Transformer parameters for a 3 phase load of 200kVA
//Page 127
clc;
clear;

//Considering Van as reference voltage

SL3phi=200*(10^3); //Load to be powered
pf3=0.8; //Power Factor of three phase load
t3=acosd(pf3); //Power FActor Angle for three phase load
pf1=0.9; //Power Factor of single phase load
t1=acosd(pf1); //Power Factor angle of single phase load
SL1=80*(10^3); //Single Phase Light Load
Vll=240; //Secondary Voltage
//Rating of Single Phase Transformers between individual lines
Sbc=100*(10^3);
Sab=75*(10^3);
Sca=Sab;
```

```
//Angles of Three phase voltages
ta=0;
tb=-120;
tc=120;
//Angles of three  line currents
tai=ta-t3;
tbi=tb-t3;
tci=tc-t3;

I=SL3phi/(sqrt(3)*Vll); //Magnitude of Current
//3 Phase Line Currents
Ia3=I*exp(%i*%pi*tai/180);
Ib3=I*exp(%i*%pi*tbi/180);
Ic3=I*exp(%i*%pi*tci/180);

MIbc=SL1/Vll; //Magnitude Single Phase Current

tbc=-90; //Lagging Van //Angle of Vbc
tbci=tbc-t1; //Angle of Current Ibc
Ibc=MIbc*exp(%i*%pi*tbci/180);

//Load Currents
Ia=Ia3;
Ta=atand(imag(Ia)/real(Ia));
Ib=Ib3+Ibc;
Tb=atand(imag(Ib)/real(Ib));
Ic=Ic3-Ibc; //Current is wrong in the textbook
Tc=atand(imag(Ic)/real(Ic));

//Current Flowing in the secondary winding of the
   transformers 1,2 and 3
Iac=((Ic/Sbc)-(Ia/Sab))/((1/Sab)+(1/Sbc)+(1/Sca));
T1=atand(imag(Iac)/real(Iac)); //Angle of the above
   current
Iba=((Ia/Sca)-(Ib/Sbc))/((1/Sab)+(1/Sbc)+(1/Sca));
T2=atand(imag(Iba)/real(Iba)); //Angle of the above
   current
Icb=((Ib/Sab)-(Ic/Sca))/((1/Sab)+(1/Sbc)+(1/Sca));
```

```
T3=atand(imag(Icb)/real(Icb)); //Angle of the above current

//Kilovoltampere Load on each transformer
SLab=Vll*abs(Iba)/1000;
SLbc=Vll*abs(Icb)/1000;
SLca=Vll*abs(Iac)/1000;

Vlls=Vll; //Secondary Voltage
Vllp=7620; //Primary Voltage
n=Vllp/Vlls; //Turns Ratio

//Primary Currents of the transformer
IAC=Iac/n;
IBA=Iba/n;
ICB=Icb/n;

//Primary Current in each each phase wire
IA=IAC-IBA;
TA=atand(imag(IA)/real(IA));//Angle of the above current
IB=IBA-ICB;
TB=atand(imag(IB)/real(IB));//Angle of the above current
IC=ICB-IAC;
TC=atand(imag(IC)/real(IC));//Angle of the above current

printf('\na) The Line Currents flowing in secondary phase wire :\n')
printf('A phase is %g/_%g A\n',abs(Ia),Ta)
printf('B phase is %g/_%g A\n',abs(Ib),Tb)
printf('C phase is %g/_%g A\n',abs(Ic),Tc)
printf('b) The Current flowing in secondary winding of each transformer:\n')
printf('AC is %g/_%g A\n',abs(Iac),T1)
printf('AB is %g/_%g A\n',abs(Iba),T2)
printf('BC is %g/_%g A\n',abs(Icb),T3)
```

```
printf('c) The Load on Each Transformer is:\n')
printf('1 : %g kVA\n',SLca)
printf('2 : %g kVA\n',SLab)
printf('3 : %g kVA\n',SLbc)
printf('d) The Current flowing in primary winding of
    each transformer:\n')
printf('AC is %g/_%g A\n',abs(IAC),T1)
printf('AB is %g/_%g A\n',abs(IBA),T2)
printf('BC is %g/_%g A\n',abs(ICB),T3)
printf('e) The Line Currents flowing in primary
   phase wire :\n')
printf('A phase is %g/_%g A\n',abs(IA),TA)
printf('B phase is %g/_%g A\n',abs(IB),TB)
printf('C phase is %g/_%g A\n',abs(IC),TC)

//Ic is calculation is wrong, the author has added
   Ibc instead of subtracting, so if you change -
   into + in line 45, you get the answer as in the
   textbook
```

Scilab code Exa 3.6 To Determine the Transformer parameters for a 3 phase load of 100kVA

```
//To Determine the Transformer parameters for a 3
   phase load of 100kVA
//Page 140
clc;
clear;

//Considering Van as reference voltage

SL3phi=100*(10^3); //Load to be powered
pf3=0.8; //Power Factor of three phase load
t3=acosd(pf3); //Power FActor Angle for three phase
   load
```

```
pf1=0.9; //Power Factor of single phase load
t1=acosd(pf1); //Power Factor angle of single phase load
SL1=50*(10^3); //Single Phase Light Load
Vll=240; //Secondary Voltage
//Angles of Three phase voltages
ta=0;
tb=-120;
tc=120;
//Angles of three  line currents
tai=ta-t3;
tbi=tb-t3;
tci=tc-t3;

I=SL3phi/(sqrt(3)*Vll); //Magnitude of Current
//3 Phase Line Currents
Ia3=I*exp(%i*%pi*tai/180);
Ib3=I*exp(%i*%pi*tbi/180);
Ic3=I*exp(%i*%pi*tci/180);

MI1=SL1/Vll; //Magnitude Single Phase Current

t1v=30; //Leading Van //Angle of Vbc
t1i=t1v-t1; //Angle of Current Ibc
I1=MI1*exp(%i*%pi*t1i/180);

//Load Currents
Ia=Ia3+I1;
Ta=atand(imag(Ia)/real(Ia));
Ib=Ib3-I1;
Tb=-180+(atand(imag(Ib)/real(Ib)));
Ic=Ic3; //Current is wrong in the textbook
Tc=atand(imag(Ic)/real(Ic));

//Current flowing in the secondary of the transformer
Iba=Ia;
T2=atand(imag(Iba)/real(Iba)); //Angle of the above
```

```
current
Icb=Ic;
T3=180+(atand(imag(Icb)/real(Icb))); //Angle of the above current

//Load on Each Transformer
SLba=Vll*abs(Iba)/1000;
SLcb=Vll*abs(Icb)/1000;

Vlls=Vll; //Secondary Voltage
Vllp=7620; //Primary Voltage
n=Vllp/Vlls; //Turns Ratio

//Primary Currents of the transformer
IA=Iba/n;
TA=atand(imag(IA)/real(IA));//Angle of the above current
IB=Icb/n;
TB=T3;//Angle of the above current
IN=IA+IB; //Neutral Current
TN=atand(imag(IN)/real(IN));//Angle of the above current

printf('\na) The Line Currents flowing in secondary phase wire :\n')
printf('A phase is %g/_%g A\n',abs(Ia),Ta)
printf('B phase is %g/_%g A\n',abs(Ib),Tb)
printf('C phase is %g/_%g A\n',abs(Ic),Tc)
printf('b) The Current flowing in secondary winding of each transformer:\n')
printf('AB is %g/_%g A\n',abs(Iba),T2)
printf('BC is %g/_%g A\n',abs(Icb),T3)
printf('c) The Load on Each Transformer is:\n')
printf('1 : %g kVA\n',SLba)
printf('2 : %g kVA\n',SLcb)
printf('d) The Line Currents flowing in primary phase wire :\n')
printf('AB is %g/_%g A\n',abs(IA),TA)
```

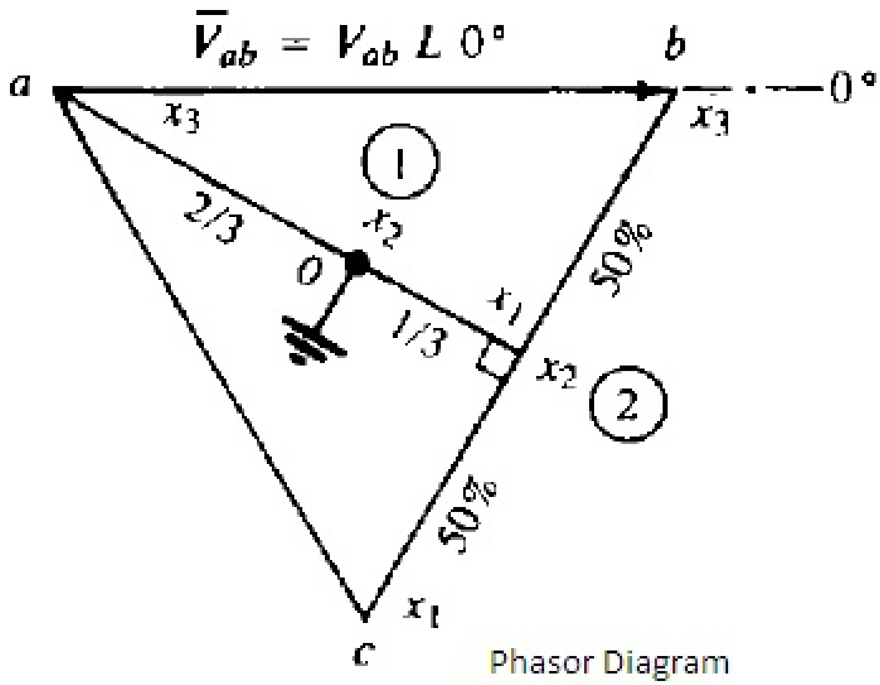

Figure 3.1: To Determine the the voltages of a two transformer bank

```
printf('CB is %g/_%g A\n',abs(IB),TB)
printf('The Neutral Current is %g/_%g\n',abs(IN),TN)

//Note the mistake in the Textbook for the
   calulation for Neutral Current
```

Scilab code Exa 3.8 To Determine the the voltages of a two transformer bank

```
//To Determine the the voltages of a two transformer bank
//Page 152
clc;
clear;


Vll=480; //Line to Line Voltage
Vln=277; //Line to neutral Voltage

//From the Phasor Diagram from the result file
Vab=Vll*exp(%i*0); //Vab is taken as reference
Vabh=50*Vab/100;
VAB=4160;
VABh=50*VAB/100;
VH1H2o=sqrt((VAB^2)-(VABh^2));
VH1H2t=(VABh);
Vx1x2o=1*sqrt((Vab^2)-(Vabh^2))/3;
Vx2x3o=2*sqrt((Vab^2)-(Vabh^2))/3;
VH2H3t=(VABh);
Vx1x2t=Vabh;
Vx2x3t=Vabh;

printf('\na) The Phasor diagram is shown in the result file attached to the code\n')
printf('b) Vab is %g/_%g V\n',abs(Vab),(imag(Vab)/real(Vab)))
printf('c) The Magnitudes of the following rated winding voltages\n')
printf('i) The Voltage VH1H2 on transformer 1 : %g V\n',VH1H2o)
printf('ii) The Voltage Vx1x2 on transformer 1 : %g V\n',Vx1x2o)
printf('iii) The Voltage Vx2x3 on transformer 1 : %g V\n',Vx2x3o)
printf('iv) The Voltage VH1H2 on transformer 2 : %g V\n',VH1H2t)
printf('v) The Voltage VH2H3 on transformer 2 : %g V
```

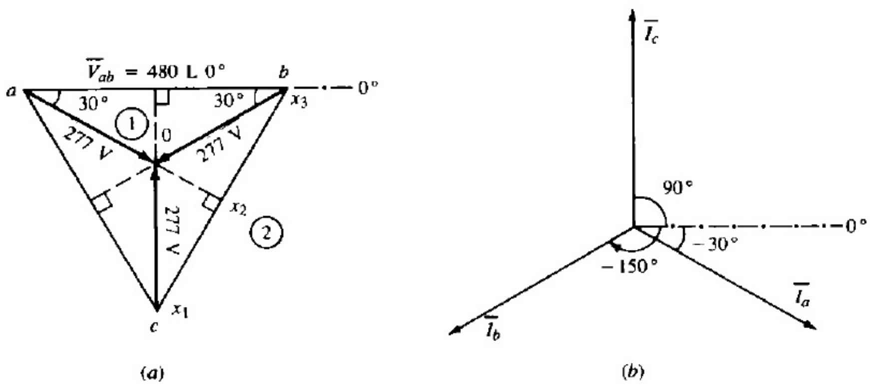

Figure 3.2: To Determine phasors and phasor diagrams when loaded with a balanced resistor

```
   \n',VH2H3t)
printf('vi) The Voltage Vx1x2 on transformer 2 : %g
   V\n',Vx1x2t)
printf('vii) The Voltage Vx1x2 on transformer 2 : %g
    V\n',Vx2x3t)
printf('d) i) NO ii) NO iii) YES\n')
```

Scilab code Exa 3.9 To Determine phasors and phasor diagrams when loaded with a balanced resistor

```
//To Determine phasors and phasor diagrams when
   loaded with a balanced resistor
//Page 154
clc;
clear;

R=2.77; //Resistance of the balanced load
//From Phasor Diagram in Result file
```

```
Vab=480*exp(%i*0); //Reference Voltage
MVn=abs(Vab)/sqrt(3); //Magnitude of line to neutral voltages
//Angles of Three phase voltages
ta=-30;
tb=-150;
tc=90;

//Angles of Winding according to the Line Currents
tx3x2=30; //Leading
tx1x2=-30; //Lagging

I=MVn/R; //Magnitude of current

//Low Voltage Current Phasors
Ia=I*exp(%i*%pi*ta/180);
Ib=I*exp(%i*%pi*tb/180);
Ic=I*exp(%i*%pi*tc/180);
pfT=cosd(ta-ta); //Angle of Ia is same as phase voltage //Resistance load

printf('\na) The Low voltage current phasors are:\n')
printf('A is %g/_%g A\n',abs(Ia),ta)
printf('B is %g/_%g A\n',abs(Ib),tb)
printf('C is %g/_%g A\n',abs(Ic),tc)
printf('b) The Phasor Diagram is the ''b'' diagram of in the result file\n')
printf('c) The Power Factor of the Transformer is %g\n',pfT)
printf('d) Power Factor as seen by winding x3x2 of transformer 2 is %g leading\n',cosd(tx3x2))
printf('e) Power Factor as seen by winding x1x2 of transformer 2 is %g lagging\n',cosd(tx1x2))
```

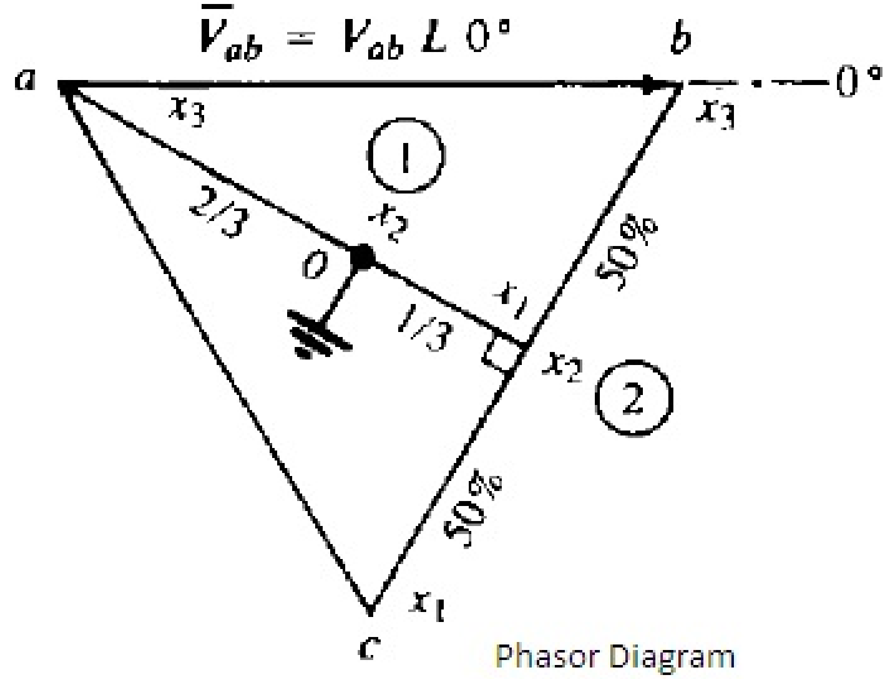

Figure 3.3: To Determine the Voltage Rating of the respective windings of the transformer

Scilab code Exa 3.10 To Determine the Voltage Rating of the respective windings of the transformer

```
//To Determine the Voltage Rating of the respective windings of the transformer
//Page 156
clc;
clear;

//'o' and 't' represent transformer one and two respectively

//Objective is to find the Factor which has to be multiplied to get VA rating
Vll=1; //Assumption Made
//From the Phasor Diagram from the result file
Vab=Vll*exp(%i*0); //Vab is taken as reference
Vabh=50*Vab/100;
Vx1x2o=1*sqrt((Vab^2)-(Vabh^2))/3;
Vx2x3o=2*sqrt((Vab^2)-(Vabh^2))/3;
Vx1x2t=Vabh;
Vx2x3t=Vabh;

//Let I be unity
I=1;

//VA Ratings of the respective windings
Sx1x2o=Vx1x2o*I;
Sx2x3o=Vx2x3o*I;
Sx1x2t=Vx1x2t*I;
Sx2x3t=Vx2x3t*I;

//Total VA rating of transformer
S1=Sx1x2o+Sx2x3o;
```

```
S2=Sx1x2t+Sx2x3t;

//Ratio of total rating to maximum rating
Rt=(S1+S2)/(sqrt(3)*Vll*I);

printf('\na) The voltampere raing of x1x2 of
   transformer 1 is %g*VI VA\n',Sx1x2o)
printf('b) The voltampere raing of x1x2 of
   transformer 1 is %g*VI VA\n',Sx2x3o)
printf('c) The Total Output from transformer 1 is %g
   *VI VA\n',S1)
printf('d) The voltampere raing of x1x2 of
   transformer 2 is %g*VI VA\n',Sx1x2t)
printf('e) The voltampere raing of x1x2 of
   transformer 2 is %g*VI VA\n',Sx2x3t)
printf('f) The Total Output from transformer 2 is %g
   *VI VA\n',S2)
printf('g) The Total Rating to the Maximum Continous
    Output is %g\n',Rt)
```

Scilab code Exa 3.11 To Determine the parameters of two single transformers

```
//To Determine the parameters of two single
   transformers
//Page 157
clc;
clear;

//Per unit value
Zt=0.01+(%i*0.03); //Transformer impedance

Vll=240; //Secondary Voltage

S1=90; //Lighting Load
```

```
pfl=0.9;
tl=acosd(pfl);
S=25; //Balanced Load
pf=0.8;
t=acosd(pf);

deff('x=angle(y)','x=atand(imag(y)/real(y))') //Function to find the angle

tab=30; //Phase angle of Vab

Il=Sl*1000/Vll; //Magnitude of Light Load
//Using the symmetrical - components theory
Ia1=Il*exp(%i*%pi*(tab-tl)/180);
Ta1=angle(Ia1); //Angle for the above current
Ib1=-1*Ia1;
Ic1=0; //Neutral Wire
//Angles of three line to line voltages
ta=0;
tb=-120;
tc=120;

Ib=S*1000/(sqrt(3)*Vll); //Magnitude of balanced load

//Currents in Three phase load
Ta2=ta-t;
Ia2=Ib*exp(%i*%pi*Ta2/180);
Tb2=tb-t;
Ib2=Ib*exp(%i*%pi*Tb2/180);
Tc2=tc-t;
Ic2=Ib*exp(%i*%pi*Tc2/180);

//Currents in phase wire
Ia=Ia1+Ia2;
Ta=angle(Ia); //Angle corresponding to the above angle
Ib=Ib1+Ib2;
```

```
Tb=angle(Ib); //Angle corresponding to the above
   angle
Ic=Ic1+Ic2;
Tc=angle(Ic); //Angle corresponding to the above
   angle

//Transformer Loads
ST1=Vll*abs(Ia)/1000;
T1=100; //From the above value of Load, this
   transformer is chosen to meet the specific
   characteristic
ST1pu=ST1/T1; //Per unit Load
ST2=Vll*abs(Ic)/1000;
T2=15; //From the above value of Load, this
   transformer is chosen to meet the specific
   characteristic
ST2pu=ST2/T2; //Per unit Load

//Transformer Power Factors
pfT1=cosd(tab-Ta);
pfT2=cosd(90-Tc); //Vcb makes angle of 90

Vh=7200; //High End Voltage
n=Vh/Vll; //Turns Ratio

// The Primary Line Currents
IA=Ia/n;
IB=-1*Ic/n;
IN=-1*(IA+IB);

Ibase=T1*1000/Vll; //Base Current
Iapu=Ia/Ibase;
Icpu=Ic/Ibase;

//Phase Voltages
Vab=Vll*exp(%i*%pi*tab/180);
Vbc=Vll*exp(-1*%i*%pi*90/180);
//Per Unit Voltages
```

```
VANpu=(Vab/Vll)+(Iapu*Zt);
VBNpu=(Vbc/Vll)-(Icpu*Zt);

//Actual Voltages
VAN=VANpu*Vh;
VBN=VBNpu*Vh;

printf('\na) The Phasor Currents:\n')
printf('Ia is %g/_%g A\n',abs(Ia),Ta)
printf('Ib is %g/_%g A\n',abs(Ib),180+Tb)
printf('Ic is %g/_%g A\n',abs(Ic),Tc)
printf('\nb) Suitable ratings of the transformers are %g kVA and %g kVA\n',T1,T2)
printf('\nc) The Per Unit kVA load on each transformer is %g pu and %g pu\n',ST1pu,ST2pu)
printf('\nd) The power factor of output of each transformer is %g and %g both lagging\n',pfT1,pfT2)
printf('\ne) The phasor currents at the high voltage leads:\n')
printf('IA is %g/_%g A\n',abs(IA),Ta)
printf('IB is %g/_%g A\n',abs(IB),180+angle(IB))
printf('IN is %g/_%g A\n',abs(IN),angle(IN))
printf('\nf) VAN is %g/_%g V and VBN is %g/_%g V\n',abs(VAN),angle(VAN),abs(VBN),angle(VBN))

//Highly Accuracy of Answers; Upto 5 decimal Places
```

Chapter 4

Design of Subtransmission Lines and Distribution Substations

Scilab code Exa 4.1 To determine the constant K for 16kV feeder

```
//To determine the constant K for 16kV feeder
//Page 201
clc;
clear;

//Conductor Pararmeters
r=1.503;
xa=0.609;
xd=0.1366;
pf=0.9;
Vb=2400;
Vr=Vb;
x=xa+xd;
Kc=0.01; //From the Curve

K=((r*pf)+(x+sind(acosd(pf))))*(1000/3)*100/(Vr*Vb);
    // In Percent
```

```
printf('\na) The Value of Constant K is %g percent
    VDpu per kVA mile\n',K)
printf('b) From the precalculated per cent voltage
    drop Curve, It is found that the K is \n%g
    percent VDpu per kVA mile which is same as the
    answer obtained in part a\n',Kc)
```

Scilab code Exa 4.2 To Calculate the percent voltage drop in the main for a lumped load

```
//To Calculate the percent voltage drop in the main
    for a lumped load
//Page 202
clc;
clear;

K=0.01; //Percentage Value
Sn=500; //Load in kVA
pf=0.9; //Lagging
s=1; //Length of the feeder
VD=s*K*Sn; //Voltage drop in percent

printf('The Percent Voltage drop in the Main is %g
    percent\n',VD)
```

Scilab code Exa 4.3 To Calculate percent voltage drop in the main for a uniformly distributed load

```
//To Calculate percent voltage drop in the main for
    a uniformly distributed load
//Page 203
clc;
```

```
clear;

K=0.01; //Percentage Value
Sn=500; //Load in kVA
pf=0.9; //Lagging
l=1; //Total Length of the feeder
s=l/2; //effective Length of the feeder
VD=s*K*Sn; //Voltage drop in percent

printf('The Percent Voltage drop in the Main is %g
    percent\n',VD)
```

Scilab code Exa 4.4 To Calculate percent voltage drop in the main for a uniformly increasing load

```
//To Calculate percent voltage drop in the main for
    a uniformly increasing load
//Page 203
clc;
clear;

K=0.01; //Percentage Value
Sn=500; //Load in kVA
pf=0.9; //Lagging
l=1; //Total Length of the feeder
s=l*2/3; //effective Length of the feeder
VD=s*K*Sn; //Voltage drop in percent

printf('The Percent Voltage drop in the Main is %g
    percent\n',VD)
```

Scilab code Exa 4.5 To Compare the results the percent voltage drop ratio for different loading

```
//To Compare the results the percent voltage drop
   ratio for different loading
//Page 204
clc;
clear;

//Voltage Drops in Percentage
VDlumped=5;
VDuniform=2.5;
VDincreasing=3.333;

//Ratio of the percent voltage drops
Rlu=VDlumped/VDuniform;
Rli=VDlumped/VDincreasing;
Riu=VDincreasing/VDuniform;

printf('\na) Percent VDlumped = %g Percent VDuniform
   \n',Rlu)
printf('b) Percent VDlumped = %g Percent
   VDincreasing\n',Rli)
printf('c) Percent VDincreasing = %g Percent
   VDuniform\n',Riu)
```

Scilab code Exa 4.6 To determine the substation parameters for various Load densities

```
//To determine the substation parameters for various
    Load densities
//Page 208
clc;
clear;

D=[500,500,2000,2000,10000,10000,2000,2000]; //Load
   Densities in kVA/sq.miles
TAn=[6,6,3,3,1,1,15,15]; //Substation Area in sq.
```

```
miles
VD=[3,6,3,6,3,6,3,6]; //Maximum Total Primary Feeder Voltage drops in percentage
Vll=[4.16,4.16,4.16,4.16,4.16,4.16,13.2,13.2]; // Base Feeder Voltage in kV

TSn=D.*TAn; //Susbstation Load
//From the Graphs of feeders vs load desity in the textbook; The Number of feeders are found to be

n=[4,2,5,3,5,4,6,5]; //No of feeders

//Also from the graph, The characteristic or the feeder is determined
//1-5, 7 are VDL feeders
//6 and 8 are TL feeders

Sn=TSn./n; //Load Per Feeder
//To Determine the Load Current
Il=Sn./(sqrt(3).*Vll);

printf('\na)')
printf('\nThe Substation Size is\n')
disp(TSn)
printf('\nThe Number of Feeders from the Curve is\n')
disp(n)
printf('\nAlso From the Curve, 1,2,3,4,5,7 cases are VDL but 6 and 8 case are TL\n')
printf('\na)')
printf('\nThe Load Current for 6th Case is %g A, which is less than the ampacities of the main but \nmore than that of the lateral, Hence it is thermally limited but not the main feeder\n',Il(6))
printf('\nThe Load Current for 8th Case is %g A, which is less than the ampacities of the main but \nmore than that of the lateral, Hence it is
```

```
thermally limited but not the main feeder\n',Il
(8))
```

Scilab code Exa 4.7 To Find feeder properties of TL and VDL

```
//To Find feeder properties of TL and VDL
//Page 211
clc;
clear;

D=1000; //Load Density in kVA per sq miles
Vll=4.16; //Line to Lien voltage in kV
//From The Tables and Curves from the Theory
K=0.007;
//For TL
TLImax=230; //Maximum Feeder Current
TLSn=sqrt(3)*Vll*TLImax; //Maximum Load Per Feeder
TLn=4; //No of Feeders
TLTSn=TLn*TLSn; //Substation Load
TLl4=sqrt(TLSn/D); //Feeder Length
TLS=2*TLl4; //Total Spacing

TLVDn=2*K*D*(TLl4^3)/3; //TotalVoltageDrop in the
    main

//For VDL
VDLVDn=3; //Percent Voltage Drop
VDLl4=nthroot((3*VDLVDn/(2*K*D)),3); //Feeder Length
VDLS=2*VDLl4; //Station size
VDLSn=D*(VDLl4^2); //Maximum Load Per Feeder
VDLn=TLn; //Number Of Feeders
VDLTSn=VDLn*VDLSn; //Susbtation Load
VDLImax=VDLSn/(sqrt(3)*Vll); //Ampere Rating of the
    Main
R=VDLImax/TLImax; //Ampere Loading
```

```
printf('\na) For Thermally Limited \n')
printf('i) The Substation Size = %g kVA\n',TLTSn)
printf('ii) Substation Spacing = %g miles\n',TLS)
printf('iii) Maximum Load Per Feeder = %g kVA\n',TLSn)
printf('iv) The Voltage Drop is %g percent\n',TLVDn)

printf('\nb) For Voltage Drop Limited \n')
printf('i) The Substation Size = %g kVA\n',VDLTSn)
printf('ii) Substation Spacing = %g miles\n',VDLS)
printf('iii) Maximum Load Per Feeder = %g kVA\n',VDLSn)
printf('iv) Ampere Loading of the Main is %g pu\n',R)

//Note The Approximation to 750 kVA
```

Scilab code Exa 4.8 To determine the better susbstation site

```
//To determine the better susbstation site
//Page 213
clc;
clear;
DivF=1.2; //Diversity Factor
DemF=0.6; //Demand Factor
CL=2000; //Connected Load Density in kVA per sq. miles

DD=DemF*CL/DivF; //Diversified Demand
A=4; //Area of the Substation

TSn=DD*A; //Peak Loads of A and B
Sm=TSn; //Peak Loads

```

```
//Constants for different conductors
Km=0.0004;
Kl=0.00095;
//Number of Laterals
Na=16; //Site A
Nb=32; //Site B

//Length of the Main
La=2;
Lb=3;
//length of laterals
Lla=2;
Llb=1;
//Length of expres Load
Le=1;
Leffb=Le+((Lb-Le)/2); //Effective Length of the
    feeder in site B
//Voltage drops
VDa=(La*Km*Sm/2)+(Lla*Kl*Sm/(Na*2));
VDb=(Leffb*Km*Sm)+(Llb*Kl*Sm/(Nb*2));

printf('\nThe Voltage drop in Site A is %g percent\n
    ',VDa)
printf('The Voltage drop in Site B is %g percent\n',
    VDb)
printf('Comparing the results we find Site A
    suitable due to its less percent voltage drop\n')
VDb=(La*Km*Sm/2)+(Lla*Kl*Sm/Na);
```

Scilab code Exa 4.10 To find the substation spacing and load on transformers

```
//To find the substation spacing and load on
    transformers
//Page 217
```

```
clc;
clear;

D=500; //Load Density in kVA per sq.miles
Vl=12.47; //Line Voltage in kV
N=2; //Feeders per substation
//From Table A-5 Appendix A it Current Ampacity can
   be found

Imax=340;

S2=sqrt(3)*Vl*Imax; //Load Per Feeder

l2=sqrt(S2/D); //Length of the feeder
S=2*l2; //Substation Spacing
TS2=S2*N; //Total Load on substation

printf('\nThe Parts a,b and c of thhis question
   cannot be coded\n')
printf('d) The substation size and spacing is %g kVA
    and %g miles\n',TS2,S)
```

Scilab code Exa 4.11 To Compare the method of service area coverage with four feeders

```
//To Compare the method of service area coverage
   with four feeders
//Page 221
clc;
clear;

Ts=1; //Assumed Load on station
K=1; //Assumption Constant
K2=K;
K4=K;
```

```
D=1;//Assumption Load Density
//Number of feeders
N2=2;
N4=4;
S2=Ts/N2; //Load per feeder //Two feeders
S4=Ts/N4; //Load per feeder //4 feeders
l=poly(0,'l'); //Variable Value of length
L2eff=1*l/3;
L4eff=2*l/3;

deff('x=VD(y)','x=D*(l^2)*K*y') //Function to find VD

VD2=VD(L2eff);
VD4=VD(L4eff);
RVD=VD2/VD4;
X=1-RVD;
RVD=1/roots(X(2)); //To find the ratio of (l2^3)/(l4^3)

Rl=nthroot(RVD,3); //Ratio of length of feeder for 2 feeders two by length of feeder for 4 feeders

//A is directly proportional to l^2
RA=(Rl^2);

//TSn is directly proportional to n and l^2
RTS=(N2/N4)*(Rl^2);

printf('\na) Ratio of substation spacings = 2l2/2l4 = %g\n',Rl)
printf('b) Ratio of areas covered per feeder main = A2/A4 = %g\n',RA)
printf('c) Ratio of substation loads = TS2/TS4 = %g\n',RTS)
```

Chapter 5

Design Considerations of Primary Systems

Scilab code Exa 5.1 To determine the circuit parameters of a radial express feeder

```
//To determine the circuit parameters of a radial
   express feeder
//Page 254
clc;
clear;

Z=0.1+(0.1*%i); //Feeder Impedance per unit
R=real(Z); //Resistance
X=imag(Z); //Reactance
Vs=1; //Sending End Voltage
Pr=1; //Constant Power Load
pfr=0.8; //Power Factor at recieving end
tr=acosd(pfr); //Power FActor angle
deff('x=angle(y)','x=atand(imag(y)/real(y))') //
   Function to Determine the Angle of a phasor

K=(Vs^2)-(2*Pr*(R+(X*tand(tr))));

```

```
Vr=sqrt((K/2)*(1+sqrt(1-((2*abs(Z)*Pr/(K*pfr))^2))))
    ; //Recieving End Voltage
C=Pr*(X-(R*tand(tr)))/((Vr^2)+(Pr*(R+(X*tand(tr)))))
    ;

del=atand(C);

Ir=(Pr/(abs(Vr)*pfr))*exp(-1*%pi*%i*tr/180) //
    Recieving End Current
Is=Ir; //Sending End Current
Tir=angle(Ir);

Vr1=Vs-(Z*Ir);

printf('\na) Vr is %g/_%g pu, del is %g degrees, Ir
    = Is = %g/_%g pu\n',abs(Vr),angle(Vr),del,abs(Ir)
    ,Tir)
printf('b) Vr is %g/_%g pu, which is almost equal to
     the previous case.\n',Vr1,angle(Vr1))
```

Scilab code Exa 5.2 To determine the percent voltage drops

```
//To determine the percent voltage drops
//Page 259
clc;
clear;

Sl=518; //Total Load on Lateral
Sm=1036; //Total Load on Main
Vll=4.16; //Line to Line voltage

//Currents in the respective current
Ilateral=Sl/(sqrt(3)*Vll);
Imain=Sm/(sqrt(3)*Vll);

```

```
C=5280; //Length Constant
Ll=5760/C; //Lateral Length
Lm=3300/C; //Main Length

//Constant for the cables
Kl=0.015;
Km=0.01;

//Voltage Drop Percents for 3 phase
VDlateral3=Ll*Kl*Sl/2;
VDmain3=Lm*Km*Sm;
TVD3=VDmain3+VDlateral3;
//Voltage Drop Percents for 1 phase according to Morrisoncfor laterals
VDlateral1=VDlateral3*4;
VDmain1=VDmain3;
TVD1=VDlateral1+VDmain1;


//CASE B
//To meet the maximum primary voltage drop criterion of 4.00 percent
//Conductors with ampacities of 480A and 270A for Main and laterals

//Constants from the table
Klb=0.006;
Kmb=0.003;

//Voltage Drop Percents
VDlateralb=Ll*Klb*Sl/2;
VDmainb=Lm*Kmb*Sm;
TVDb=VDmainb+VDlateralb;

printf('\na) The percent voltage drops at :\n')
printf('i) 3Phase\n')
printf('Lateral End is %g percent\n',VDlateral3)
printf('Main End is %g percent\n',VDmain3)
```

```
printf('ii) 1Phase\n')
printf('Lateral End is %g percent\n',VDlateral1)
printf('Main End is %g percent\n',VDmain1)
printf('\nb) Conductors with Ampacities of 480A and
   270A are used to find the Percent voltage drop of
    the Main and Lateral as %g percent and %g
   percent respectively\n',VDmainb,VDlateralb)
printf('The Above Drops meet the required criterion
   of 4 percent voltage drop\n')
```

Scilab code Exa 5.3 To find voltge drop percents for a self supporting aerial messenger cable

```
//To find voltge drop percents for a self supporting
    aerial messenger cable
//Page 263
clc;
clear;

//Terms taken from Example two
Il=72;
Im=144;
C=5280; //Length Constant
Ll=5760/C; //Lateral Length
Lm=3300/C; //Main Length

//From Tables
//Lateral
rl=4.13; //Resistance per mile
xLl=0.258; //Reactance per mile
//Main
rm=1.29; //Resistance per mile
xLm=0.211;//Reactance per mile
pf=0.9; //Power Factor

```

```
Vb=2400; //Base Voltage
//Voltage Drops
VDlateral=Il*((rl*pf)+(xLl*sind(acosd(pf))))*Ll/2;
VDmain=Im*((rm*pf)+(xLm*sind(acosd(pf))))*Lm;

//Percent Voltage Drop
perVDlateral=VDlateral*100/Vb;
perVDmain=VDmain*100/Vb;

TVD=perVDlateral+perVDmain; //Total Percent Voltage
    drop

//Case B
//Conductors With Ampacities of 268A and 174A for
    Main and Laterals
//From Tables
//Lateral
rlb=1.03; //Resistance per mile
xLlb=0.207; //Reactance per mile
//Main
rmb=0.518; //Resistance per mile
xLmb=0.191;//Reactance per mile

Vb=2400; //Base Voltage
//Voltage Drops
VDlateralb=Il*((rlb*pf)+(xLlb*sind(acosd(pf))))*Ll
    /2;
VDmainb=Im*((rmb*pf)+(xLmb*sind(acosd(pf))))*Lm;

//Percent Voltage Drop
perVDlateralb=VDlateralb*100/Vb;
perVDmainb=VDmainb*100/Vb;

TVDb=perVDlateralb+perVDmainb; //Total Percent
    Voltage drop

printf('\na) The percent voltage drops at :\n')
printf('Lateral End is %g percent\n',perVDlateral)
```

```
printf('Main End is %g percent\n',perVDmain)

printf('\nb) Conductors with Ampacities of 278A and
   174A are used to find the Percent voltage drop of
    the Main and Lateral as %g percent and %g
   percent respectively\n',perVDmainb,perVDlateralb)
printf('The Above Drops meet the required criterion
   of 4 percent voltage drop\n')
```

Scilab code Exa 5.4 To determine the percent voltage drops using nomnial operating voltage as base voltage

```
//To determine the percent voltage drops using
   nomnial operating voltage as base voltage
//Page 265
clc;
clear;

Sl=518; //Total Load on Lateral
Sm=5180; //Total Load on Main
Vll=12.47; //Line to Line voltage

//Currents in the respective current
Ilateral=Sl/(sqrt(3)*Vll);
Imain=Sm/(sqrt(3)*Vll);

C=5280; //Length Constant
Ll=5760/C; //Lateral Length
Lm=3300/C; //Main Length

//Constant for the cables
Km=0.0008;
Kl=0.00175;

//Voltage Drop Percents for 3 phase
```

```
VDlateral=Ll*Kl*Sl/2;

//Due to peculiarity of this new problem, one half of the main has to considered as express feeder and the other connected to a uniformly distributed load of 5180kVA
VDmain=Lm*Km*Sm*3/4;
TVD=VDmain+VDlateral;

//Since the inductive reactance of the line is
Cd=12; //Constant to find the distance in terms of feet

//Diameters of the Conductors
Dmi=37;
Dmn=53;

//Drops per mile
xdi=0.1213*log(Dmi/Cd);
xdn=0.1213*log(Dmn/Cd);

Dxd=xdn-xdi; //Difference in Drops

printf('\na) The percent voltage drops at :\n')
printf('Lateral End is %g percent\n',VDlateral)
printf('Main End is %g percent\n',VDmain)

printf('\nb) The Above Drops meet the required criterion of 4 percent voltage drop\n')
printf('\nc) The Difference in Voltage drop is %g ohm/mile, which is a smaller VD valuue that it really is.\n',Dxd)
```

Scilab code Exa 5.5 To find the percent voltage drop at the ends of the most remote laterals

```
//To find the percent voltage drop at the ends of
   the most remote laterals
//Page 268
clc;
clear;

Vb=7200; //Base Voltage in V
pf=0.9; //Power Factor
Sm=10360; //Load on Main Feeder in kVA
Vll=12.47; //Line to Line voltage in kV
Imain=Sm/(sqrt(3)*Vll); //Current in Main Feeder

//Note Suffix l means lateral and m means main

Vph=7.2; //Phase Voltage in kV
Sl=2*518; //Load on Lateral Feeder in kVA
Ilateral=Sl/Vph; //Current in Laterals

//Length of the Feeder
//Length Constant
Cm=5280; //Main
Cl=1000; //Lateral
Ll=5760/Cl; //Lateral Length
Lm=3300/Cm; //Main Length

//Constants for the particular cables from the
   tables
rl=0.331;
xLl=0.0300;
rm=0.342;
xam=0.458;
xdm=0.1802;
xLm=xam+xdm;

//Voltage Drops for Normal Condition
VDmainn=(Imain/2)*((rm*pf)+(xLm*sind(acosd(pf))))*Lm
   /2;
VDlateraln=(Ilateral/2)*((rl*pf)+(xLl*sind(acosd(pf)
```

```
)))*Ll/2;

perVDmainn=VDmainn*100/Vb;
perVDlateraln=VDlateraln*100/Vb;

TVDn=perVDmainn+perVDlateraln;

//Voltage Drops for Worst Conditions
VDmainw=(Imain)*((rm*pf)+(xLm*sind(acosd(pf))))*Lm/2;
VDlateralw=(Ilateral)*((rl*pf)+(xLl*sind(acosd(pf))))*Ll;

perVDmainw=VDmainw*100/Vb;
perVDlateralw=VDlateralw*100/Vb;

TVDw=perVDmainw+perVDlateralw;

printf('\na)From Table A5, 300-kcmilACSR conductors, with 500A Ampacity is used for main\nand AWG #2 XLPE Al URD cable with 168A Ampacity\n')
printf('b) The Total Voltage Drop in Percent for Normal Operation is %g percent\n',TVDn)
printf('c) The Total Voltage Drop in Percent for Worst Condition is %g percent\n',TVDw)
printf('d) The Voltage drop is met for Normal operation and NOT for emergency operation\n')
```

Chapter 6

Design Considerations of Secondary Systems

Scilab code Exa 6.1 To Compute the Economical Sizes of the Transformer and its Equipment

```
//To Compute the Economical Sizes of the Transformer
    and its Equipment
//Page 296
clc;
clear;

NC=24; //Number Of Customers Per Block

//We get the Total Annual Cost from the releveant
   equations as
// TAC = 239.32 + (3.1805*ST) + (3492/ST) + (28170/
   ST^2) + (0.405*ASL) + (17018/ASL) + (1.134*ASD) +
    (8273/ASD)

//We know split the above equation into 3 different
   parts according to factors ST,ASD,ASL

//Variable Values of the Factors
```

```
ST=poly(0,'ST');
ASD=poly(0,'ASD');
ASL=poly(0,'ASL');

//Functions to Find the TAC corresponding to the Respective Factors

deff('x=TransSize(y)','x=239.52 + (3.1805*y) + (3492/y) + (28170/(y^2))')
deff('x=SDwire(y)','x=(1.134*y)+(8273/y)')
deff('x=SLwire(y)','x=(0.405*y)+(17018/y)')

//Total Annual Costs of the respective Factors
TACST = TransSize(ST);
TACASD = SDwire(ASD);
TACASL = SLwire(ASL);

//Partially Differentiating wrt ASD we get
Y1=derivat(TACASD);
X1=roots(Y1(2));
ASD=X1(1);  //Calculated Value
ASDstd = 105.500;
ASDstd1 = 133.1;

//Partially Differentiating wrt ASL we get
Y2=derivat(TACASL);
X2=roots(Y2(2));
ASL=X2(1); //Calculated Value
ASLstd = 211.600;
ASLstd1 = 250;

//Partially Differentiating wrt ST we get
Y3=derivat(TACST);
X3=roots(Y3(2));
ST=round(X3(1));  //Calculated Value
STstd = 50;

//Total Annual Cost of the Calculated parameters
```

```
TAC=TransSize(ST)+SDwire(ASD)+SLwire(ASL);
//Calculation Mistake in The Text Book

//Total Annual Cost of the First Higher Standard
   Parameters
TACstd=TransSize(STstd)+SDwire(ASDstd)+SLwire(ASLstd
   );
//Total Annual Cost of the Second Higher Standard
   Parameters
TACstd1=TransSize(STstd)+SDwire(ASDstd1)+SLwire(
   ASLstd1);

//Total Fixed Charges per Year
TACFC=((75+(2.178*STstd))+(5.4+(0.405*ASLstd))
   +(15.12+(1.134*ASD))+(144));
//Total Operating Charges per Year
TACOC=((0.0225*STstd)+(0.98*STstd)+(28170/(STstd^2))
   +(3492/STstd)+(17018/ASLstd)+(8273/ASDstd));

//Values Might Vary from those in the text due to
   high precision

//Fixed Charges Per Customer Per Month
FC=TACFC/(NC*12);

//Variable Costs Per Customer per month
VOC=TACOC/(NC*12);

printf('\na) The Most Economical SD Size is %g kmil
   and the nearest larger standard AWG wire size is
   %g kmil\n',ASD,ASDstd)
printf('b) The Most Economical SL Size is %g kmil
   and the nearest larger standard AWG wire size is
   %g kmil\n',ASL,ASLstd)
printf('c) The Most Economical Distribution
   Transformer Size is %g kmil and the nearest
   larger standard transformer size is %g kVA\n',ST,
   STstd)
```

```
printf('d) The Total Annual Cost Per Block for the
    theoretically most economical sizes of equipment
    is %g dollars\n',TAC)
printf('e) The Total Annual Cost Per Block for the
    nearest larger standard comercial sizes of
    equipment is %g dollars\n',TACstd)
printf('f) The Total Annual Cost Per Block for the
    nearest larger transformer size and for the
    second larger sizes of ASD and ASL is %g dollars\
    n',TACstd1)
printf('g) Fixed Charges per Customer per Month is
    %g dollars\n',FC)
printf('h) The Variable Operating Costs Per Customer
     Per Month is %g dollars\n',VOC)
```

Scilab code Exa 6.2 To determine the coeffcient matrix for a unbalanced load

```
//To determine the co-effcient matrix for a
    unbalanced load
//Page 304
clc;
clear;

Dab=12;
Dan=12;
Dbn=24;
Daa=12*0.01577;
Dbb=Daa;
Dnn=Daa;

deff('x=Coeff(y,z)','x=(2*(10^-7))*log(y/z)') //
    function to find the elements of the co-efficient
     matrix

```

```
//Part A of the question cannot be found using Scilab , Hence from the equation obtained in part A we can numerically compute the Co- Efficient Matrix

CM=[Coeff(Dan,Daa),Coeff(Dan,Dab);Coeff(Dbn,Dab),Coeff(Dbn,Dbb);Coeff(Dnn,Dan),Coeff(Dnn,Dbn)];

printf('\n Part A cannot be resulted by this code, hence from the equations obtained in Part A\n Co-Efficient Matrix is Obtained as\n')
disp(CM.*(10^7))
printf('\n * (10^-7) Wb*T/m\n')
```

Scilab code Exa 6.4 To determine the circuit parameters of an unbalanced load

```
//To determine the circuit parameters of an unbalanced load
//Page 308
clc;
clear;
//Primary Voltage
V1=7272*(%i*%pi*0/180);

//Secondary Voltages
Ea=120*(%i*%pi*0/180);
Eb=120*(%i*%pi*0/180);

//Impedances
Za=0.8+(%i*0.6);
Zb=0.8+(%i*0.6);

n=60; //Turns Ratio

```

```
deff('x=angle(y)','x=atand(imag(y)/real(y))') //To Determine the Angle

//Substituting the values we get the following equations
//121.2 = Ia*(0.8857 + j0.6846) + Ib*(0.03279 + j0.03899)
//121.2 = Ia*(-0.03279 - j0.03899) + Ib*(-0.88574 + j0.50267)

//For Convenience We segregate them as
Z1=(0.8857+(%i*0.6846));
Z2=(0.03279+(%i*.03899))
Z3=(-0.03279-(%i*.03899))
Z4=(-0.88574+(%i*.50267))

Z=[Z1,Z2;Z3,Z4]; //Impedance matrix
V=[121.2;121.2]; //Voltage Matrix
I=inv(Z)*V; //Current Matrix

//Secondary Currents
Ia=I(1);
Ib=I(2);

In=-Ia-Ib; //Secondary neutral Currents

//Secondary Voltages
Va=Za*Ia;
Vb=-1*Zb*Ib;

//Secondary Voltage Resultant
Vab=Va+Vb;

printf('\na) The Secondary Currents are:\n')
printf('Ia = %g/_%g A\n',abs(Ia),angle(Ia))
printf('Ib = %g/_%g A\n',abs(Ib),180+angle(Ib))
printf('b) The Secondary Neutral Current = %g/_%g A\n',abs(In),angle(In))
```

```
printf('c) The Secondary Voltages are:\n')
printf('Va = %g/_%g V\n',abs(Va),angle(Va))
printf('Vb = %g/_%g V\n',abs(Vb),angle(Vb))
printf('d) The Resultant Secondary Voltage Vab is %g
    /_%g V\n',abs(Vab),angle(Vab))

//In the TextBook Note That Zb has been taken wrong
    in the calculattion of Vb
```

Scilab code Exa 6.5 To find the pu voltages and tolerable and favourable voltages

```
//To find the pu voltages and tolerable and
    favourable voltages
//Page 310
clc;
clear;

N=19; //Number Transformers
St=500; //Load on each transformer in kVA
L=5096+(%i*3158); //Load
Vlf=114; //Favourable Voltage
Vlt=111; //Tolerable Volatage
Vb=125; //Base Voltage

//Per Unit Tolerable and favourable voltages
puVlf=Vlf/Vb;
puVlt=Vlt/Vb;

ZM=0.181+(%i*0.115); //The Positive Sequence
    Impedance
ZTi=0.0086+(%i*0.0492); //Transformer Impedance for
    500kVA
ZT=2*ZTi; //Transformer Impedance for 1000kVA

```

```
AAF=N*St/abs(L); //Actual Application Factor

printf('\na) The Lowest favourable Voltage is %g pu
   and The Lowest tolerable voltage is %g pu\n',
   puVlf,puVlt)
printf('b) There Are No buses in Table 6-5, for the
   first contingency outage which satisfy the
   necessary condition\n')
printf('c) For Second Contingency Outage\n')
printf('1) Less than Favourable Voltage are B,C,J,K,
   R and S\n')
printf('2) Less than Tolerable Voltage are B,C,J,K.\
   n')
printf('d) ZM/ZT = %g and (1/2)*ZM/ZT = %g
   respectively.\n',(abs(ZM)/abs(ZT)),(1/2)*(abs(ZM)
   /abs(ZT)))
printf('The Actual Application Factor is %g\n',AAF)
printf('Therefore the Design of this network is
   sufficient\n')
```

Chapter 7

Voltage Drop and Power Loss Calculations

Scilab code Exa 7.2 To determine the voltage drop or voltage regulation of a 3phase system

```
//To determine the voltage drop or voltage
   regulation of a 3phase system
//Page 327
clc;
clear;

Vll=416; //Voltage Line to Line
Vph=Vll/(sqrt(3)); //Phase Voltage and Base Voltage
//Load Currents
Ia=30;
Ib=20;
Ic=50;

//Power Factors of the load
pfa=1;
pfb=0.5;
pfc=0.9;

```

```
//Impedances of the Sections
ZA=0.05+(%i*0.01);
ZAB=0.1+(%i*0.02);
ZBC=0.05+(%i*0.05);
//Impedance upto the point of load
ZB=ZA+ZAB;
ZC=ZB+ZBC;

//Function to Calculate Voltage Drop
deff('x=VD(a,b,c)','x=a*((real(b)*c)+(imag(b)*sind(
    acosd(c))))')

//Voltage Drops at A,B and C
VDA=VD(Ia,ZA,pfa);
VDB=VD(Ib,ZB,pfb);
VDC=VD(Ic,ZC,pfc);

TVD=VDA+VDB+VDC; //Total Voltage Drop

TVDpu=TVD/Vph; // In Per Unit

deff('x=Real(y,z)','x=Vph*y*z') //Function to
    Calculate Real Power
deff('x=Reactive(y,z)','x=Vph*y*sind(acosd(z))') //
    Funtion to Calculate the Reactive power

//Real Powers
Pa=Real(Ia,pfa);
Pb=Real(Ib,pfb);
Pc=Real(Ic,pfc);
P=Pa+Pb+Pc; //Total Real Power


//Reactive Powers
Qa=Reactive(Ia,pfa);
Qb=Reactive(Ib,pfb);
Qc=Reactive(Ic,pfc);
Q=Qa+Qb+Qc; //Total Reactive Power
```

```

S=sqrt((P^2)+(Q^2)); //Total output from the
    Transformer
PF=P/S; //Load Power Factor

printf('\na) The Total Voltage drop is %g pu\n',
    TVDpu)
printf('b) The Real Power per Phase is %g kW\n',P
    /1000)
printf('c) The Reactive Power per Phase is %g kVAr\n
    ',Q/1000)
printf('d) The Kilovoltampere output and load factor
      is %g kVA and %g lagging\n',S/1000,PF)
```

Scilab code Exa 7.3 To Calculate the Voltage Drop and Verify The Cable Selected

```
//To Calculate the Voltage Drop and Verify The Cable
      Selected
//Page 329
clc;
clear;
pf=0.9; //Power Factor
Vb=120; //Base Voltage
//From The Tables
r=0.334; //Resistance per thousand feet
x=0.0299; //Reactance per thousand feet
K1=0.02613; //Voltage Drop

//Assumed Cable
I=100; //Secodary line Current
Ls=100; //Length of Secondary line in feet

R=r*Ls/1000; // Resistance Value for a 100 feet Line
X=x*Ls/1000; // Reactance Value for a 100 feet Line
```

```
VD=I*((R*pf)+(X*sind(acosd(pf)))); //Voltage Drop
VDpu=VD/Vb; //Per unit value
printf('\nThe Cable Selected is of 100 feet,
    carrying 100A and cable size #2 AWG\n')
printf('The Voltage drop for the above cable is %g
    pu V\n',VDpu)
printf('The Above Value is Close to the Value(%g pu
    V) in the table given.\n',K1)
```

Scilab code Exa 7.4 To find the voltage dip in per units for motor starting

```
//To find the voltage dip in per units for motor
    starting
//Page 333
clc;
clear;

Sts=(10+(11*4.4)); //Load Selected on the
    transformer
V=240; //Voltage
Sta=50; //Available Unit
pf=0.9; //Load Power Factor
I=(Sts/V)/(Sta/V);

VDT=I*((0.0107*pf)+(0.0139*sind(acosd(pf))));

SLload=10+(3*6);

deff('x=VD(a,b,c)','x=a*b*c/(10^4)') //Function to
    find Voltage Drop Per unit

VDSL=VD(0.0088,116.7,150);
VDSD=VD(0.01683,41.76,70);

```

```
TVD=VDT+VDSL+VDSD;

Is=80;
Smotor=Is*V/1000;
pf1=0.5;
VDIPT=((0.0107*pf1)+(0.0139*sind(acosd(pf1))))*(Smotor/Sta);

VDIPSL=VD(0.00636,80,150);
VDIPSD=VD(0.01089,80,70);
TVDIP=VDIPT+VDIPSL+VDIPSD;

VDSL1=VD(0.00769,116.7,150);
VDSD1=VD(0.0136,41.6,70);
TVD1=VDT+VDSL1+VDSD1;

printf('\na) The Voltage drops are:\n')
printf('Transformer is %g pu V\n',VDT)
printf('Secondary Lines is %g pu V\n',VDSL)
printf('Service Drops is %g pu V\n',VDSD)
printf('Total is %g pu V\n',TVD)
printf('The Above Value doesn''t meet the required criterion\n')
printf('\nb) The Voltage dip for motor starting are:\n')
printf('Transformer is %g pu V\n',VDIPT)
printf('Secondary Lines is %g pu V\n',VDIPSL)
printf('Service Drops is %g pu V\n',VDIPSD)
printf('Total is %g pu V\n',TVDIP)
printf('The Above Value does meet the required criterion\n')
printf('\nC) The Voltage drops after changing the conductors are:\n')
printf('Transformer is %g pu V\n',VDT)
printf('Secondary Lines is %g pu V\n',VDSL1)
printf('Service Drops is %g pu V\n',VDSD1)
printf('Total is %g pu V\n',TVD1)
printf('The Above Value doesn''t meet the required
```

```
    criterion\n')
printf('Thus 350 kcmilcable size for the secondary
    lines and #2/0 AWG cable size for service drops
    to meet the criteria\n')
```

Scilab code Exa 7.5 To Find the Total Load and Total steady state voltage drop

```
//To Find the Total Load and Total steady state
    voltage drop
//Page 336
clc;
clear;

//Length in feet
Lsd=100; //Service Drop Line
Lsl=200; //Secondary Line

SB=75; //Transformer Capacity in kVA
pf=0.9; //Load Power Factor

//From the Tables
ISL=129.17; //Secondary Line Current
ISD=41.67; //Service Drop Current
KSD=0.01683; //Service Drop Cable Constant
KSL=0.0136; //Secondary Cable Constant

//for Transformer
R=0.0101; //Resistance in per unit
X=0.0143; //Reatance in per unit

//From the Diagram
ST=(3+2+8+6)+(5+6+7+4)+(6+7+8+10); //Total Load on
    transformer

```

```
STpu=ST/SB; //In Per unit;

//The Above value also corresponds to the Current as Well

I=STpu; //Current in Per Unit

VDT=I*((R*pf)+(X*sind(acosd(pf)))); //Voltage Drop in the Transformer
VDSL=KSL*(Lsl*ISL/(10^4)); //Secondary Line
VDSD=KSD*(Lsd*ISD/(10^4)); //Service Drop Line

VD=VDT+VDSD+VDSL; //Total Voltage Drop

printf('\na)The Load in transformer is %g kVA or %g pu\n',ST,STpu)
printf('b) The Total Voltage Drop is %g pu V\n',VD)
```

Scilab code Exa 7.7 To determine the percent drop from the substation to various points

```
//To determine the percent drop from the substation to various points
//Page 340
clc;
clear;

An=4; //Service Area
l=1; //Length of 0a
//Voltages in kV
Vll=13.2; //Line to line
Vln=7.62; //Line to neutral

//Peak Loading
Dp=1000; //Peak Loading Intensity per sq.miles
```

```
Sl=2000; //Lumped Load in kVA

//Off Peak Loading
Dop=333; //Loading intensity

VB=7620; //Base Voltage

Vs=1.025; //Substation Voltages

Sn=Dp*An; //Load Connected to the square shaped service area
Sm=Sn+Sl; //Total Load

K=0.0003; //Cable Constant

VD0a=K*Sm*l; //Voltage Drop between substation and a
lab=2;  //Distance from a to b
VDab=(K*Sn*lab/2)+(K*Sl*lab);//Voltage drop from b to c
lbc=2; //Distance from b to c
VDbc=3*K*Sl*lbc; //Voltage drop from b to c //Change in Constant

I=Sl/(sqrt(3)*(0.947*Vll));
Ib=Sl/(sqrt(3)*(Vll));//BAse Current

MIpu=I/Ib; //Per Unit Current

Ztpu=complex(0,0.05);
pf=0.9; //Load Power Factor

Ipu=MIpu*exp(%i*%pi*acosd(pf)/180);

//The Voltage has been tapped up 5 percent

puVDcd=(abs(Ipu)*((real(Ztpu)*pf)+(imag(Ztpu)*sind(acosd(pf)))))-0.05;
VDcd=puVDcd*100;
```

```
deff('x=volt(a,b)','x=(a-(b/100))') //funtion to
    find out voltages

//per unit Voltages
puVa=volt(Vs,VD0a);
puVb=volt(puVa,VDab);
puVc=volt(puVb,VDbc);
puVd=volt(puVc,VDcd);

//Voltages in V
Va=puVa*VB;
Vb=puVb*VB;
Vc=puVc*VB;
Vd=puVd*VB;

printf('\na) The Percentage drops are\n')
printf(' Substation to a is %g percent\n',VD0a);
printf(' a to b is %g percent\n',VDab);
printf(' b to c is %g percent\n',VDbc);
printf(' c to d is %g percent\n',VDcd);
printf('b) The Per unit voltages are:\n')
printf(' Point a is %g pu V\n',puVa)
printf(' Point b is %g pu V\n',puVb)
printf(' Point c is %g pu V\n',puVc)
printf(' Point d is %g pu V\n',puVd)
printf('c) The Line to Neutral voltages are:\n')
printf(' Point a is %g  V\n',Va)
printf(' Point b is %g  V\n',Vb)
printf(' Point c is %g  V\n',Vc)
printf(' Point d is %g  V\n',Vd)
```

Scilab code Exa 7.8 To determine the percent drop from the substation to various points

```
//To determine the percent drop from the substation to various points
//Page 340
clc;
clear;

V0op=1; //At off Peak
An=4; //Service Area
l=1; //Length of 0a
//Voltages in kV
Vll=13.2; //Line to line
Vln=7.62; //Line to neutral

//Peak Loading
Dp=1000; //Peak Loading Intensity per sq.miles
Sl=2000; //Lumped Load in kVA

//Off Peak Loading
Dop=333; //Loading intensity

VB=7620; //Base Voltage

Vs=1.025; //Substation Voltages

Sn=Dop*An; //Load Connected to the square shaped service area
Sm=Sn+Sl; //Total Load

K=0.0003; //Cable Constant

VD0a=K*Sm*l; //Voltage Drop between substation and a
lab=2;  //Distance from a to b
VDab=(K*Sn*lab/2)+(K*Sl*lab);//Voltage drop from b to c
lbc=2; //Distance from b to c
VDbc=3*K*Sl*lbc; //Voltage drop from b to c //Change in Constant
```

```
I=Sl/(sqrt(3)*(0.947*Vll));
Ib=Sl/(sqrt(3)*(Vll));//BAse Current

MIpu=I/Ib; //Per Unit Current

Ztpu=complex(0,0.05);
pf=0.9; //Load Power Factor

Ipu=MIpu*exp(%i*%pi*acosd(pf)/180);

//The Voltage has been tapped up 5 percent

puVDcd=(abs(Ipu)*((real(Ztpu)*pf)+(imag(Ztpu)*sind(
    acosd(pf)))))-0.05;
VDcd=puVDcd*100;
deff('x=volt(a,b)','x=(a-(b/100))') //funtion to
    find out voltages

//per unit Voltages
puVa=volt(V0op,VD0a);
puVb=volt(puVa,VDab);
puVc=volt(puVb,VDbc);
puVd=volt(puVc,VDcd);

//Voltages in V
Va=puVa*VB;
Vb=puVb*VB;
Vc=puVc*VB;
Vd=puVd*VB;

printf('\na) The Percentage drops are\n')
printf(' Substation to a is %g percent\n',VD0a);
printf(' a to b is %g percent\n',VDab);
printf(' b to c is %g percent\n',VDbc);
printf(' c to d is %g percent\n',VDcd);
printf('b) The Per unit voltages are:\n')
printf(' Point a is %g pu V\n',puVa)
printf(' Point b is %g pu V\n',puVb)
```

```
printf(' Point c is %g pu V\n',puVc)
printf(' Point d is %g pu V\n',puVd)
printf('c) The Line to Neutral voltages are:\n')
printf(' Point a is %g  V\n',Va)
printf(' Point b is %g  V\n',Vb)
printf(' Point c is %g  V\n',Vc)
printf(' Point d is %g  V\n',Vd)
```

Scilab code Exa 7.9 To Determine the location of the substation

```
//To Determine the location of the substation
//Page 344
clc;
clear;

Vll=13.2; //Voltage in kV (Line voltage)
TCDi=0.45; //Load Density in kVA per feet
FD=1.08; //Diversity Factor for all loads
FLS=0.2; //Annual Loss Factor
DFi=0.6; //30 min Annual Demand Factor

Dg=TCDi*DFi/FD; //Divesified Maximum Demand of the
    Group

L=30000; //Length of the Whole Feeder in Feet

//To Achieve Minimum Voltage Drop, The Substation
    must be located at the centre of the line
Ln=15000; //NEW Length of the Feeder

SPK=Dg*Ln; //Peak Load on each main of the
    substation trnasformer
I=(SPK/(sqrt(3)*Vll)); //Current in the Line

K=0.0009; //For the Assumed Conductor
```

```
VD=K*SPK*Ln/(2*5280); //Voltage Drop, Is divided by
    5280, to convert the length in miles

printf('\na) To Achieve Minimum Voltage Drop, The
    Substation is Placed at the Centre of the Line,\n
     and For a Current of %g A following in it, #4
    AWG or #2 AWG ACSR conductors are used\n',I)
printf('b) For a #4 AWG Copper Conductor, The
    Percentage Voltage drop at annual peak load is %g
     pecent\n',VD)

//Calculation Mistake in the TextBook
```

Scilab code Exa 7.10 To Determine the Annual Energy Loss

```
//To Determine the Annual Enery Loss
//Page 346
clc;
clear;

Vll=13.2; //Voltage in kV (Line voltage)
TCDi=0.45; //Load Density in kVA per feet
FD=1.08; //Diversity Factor for all loads
FLS=0.2; //Annual Loss Factor
DFi=0.6; //30 min Annual Demand Factor

Dg=TCDi*DFi/FD; //Divesified Maximum Demand of the
    Group
L=30000; //Length of the Whole Feeder in Feet
I=164.2; //Current

r=0.592; //Resistance Per Mile
R=r*L/(3*5280); //Total Resistance

CL=3*(I^2)*R; //Total Power Loss in entire length
```

```
TAEL=CL*FLS*8760/(10^3); //Total Annual Energy Loss

printf('\nThe Total Annual Eddy Current Loss is %g
   kWhr\n',TAEL)
```

Scilab code Exa 7.11 To Determine the Line to Line Voltage at point a

```
//To Determine the Line to Line Voltage at point a
//Page 347
clc;
clear;

//Loads in kVA
Sbc=3000; //Load Along bc
Sd=2000; //Load At Point d
S0a=Sbc+Sd; //Total Load
Sab=Sbc/2; //Load along ab

//Cable Constants
K0a=0.0005; //For 0 to a
Kab=0.0010; //For a to b
Kac=0.0010; //For a to c
Kad=0.0010; //For a to d

//Length
l0a=1.0; //From 0 to a
lab=2;//From a to b
lad=2; //From a to d

//Voltage Drops At Specific Points
VDa=K0a*S0a*l0a;
VDb=(Kab*Sab*lab/2)+VDa;
VDc=VDb;
VDd=(Kad*Sd*lad)+VDa;
```

```

//To determine the Voltages at Point a
Vll=12650; //Line to Line Voltage
Vln=7300; //Line to Neutral Voltage

Valn=Vln-(VDa*Vln/100);
Vall=Vll-(VDa*Vll/100);

printf('\na) The Voltage Drops at:\n')
printf('Point a is %g percent\n',VDa)
printf('Point b is %g percent\n',VDb)
printf('Point c is %g percent\n',VDc)
printf('Point d is %g percent\n',VDd)
printf('b) The Voltages VaL-N is %g V and VaL-L is
    %g V\n',Valn,Vall)
```

Chapter 8

Application of Capacitors to Distribution Systems

Scilab code Exa 8.1 To Determine the Capacitor Size to improve the power factor to a 700kVA Load

```
//To Determine the Capacitor Size to improve the power factor to a 700kVA Load
//Page 390
clc;
clear;

SL=700; //Load in kVA
pf1=65/100; //Power Factor
PL=SL*pf1; //Real Power
//From the Table of Power Factor Correction
CR=0.74; //Co-relation factor
CS=PL*CR; //Capacitor Size

CSr=360; //Next Higher Standard Capacitor Size

CRn=CSr/PL; //New Co-Relation Factor

//From the table by linear interpolation
```

```
pfr=93; //In Percentage
pfn=pfr+(172/320);

printf('\a) The Correction Factor is %g\n',CR)
printf('b) The Capacitor Size Required is %g kVAr\n'
    ,CS)
printf('c) Resulting power factor if the next higher
     standard capacitor size is used is %g percent\n'
    ,pfn)
```

Scilab code Exa 8.2 To determine the Capacitor bank required to correct power factor of induction motor

```
//To determine the Capacitor bank required to
    correct power factor of induction motor
//Page 393
clc;
clear;

Vll=4.16; //Line to Line Voltage in kV
Pr=(500*0.7457); //Rating of motor in kW
pf1=0.75; //Initial Power Factor
pfn=0.9; //Improved Power Factor
eff=0.88; //Efficiency
P=Pr/eff; //Input Power of Induction Motor
Q1=P*tand(acosd(pf1)); //Reactive Power
Q2=P*tand(acosd(pfn)); //REactive power of motor
    after power factor improvement
f=60; //Frequency of supply
w=2*%pi*f; //Angular Frequency
Qc=Q1-Q2; //Reactive Power of Capacitor
Il=Qc/(sqrt(3)*Vll);

//Capacitor Connectd in Delta
Ic1=Il/(sqrt(3));
```

```
Xc1=Vll*1000/Ic1; //Reactance of each capacitor
C1=(10^6)/(w*Xc1); //Capacitance in Micro Farad

//Capacitor Connected in Wye
Ic2=Il;
Xc2=Vll*1000/(sqrt(3)*Ic2); //Reactance of each capacitor
C2=(10^6)/(w*Xc2); //Capacitance in Micro Farad

printf('\na) Rating of Capacitor Bank is %g kVAr\n',Qc)
printf('b) The Value of Capacitance if there are connected in delta is %g micro F\n',C1)
printf('c) The Value of Capacitance if there are connected in wye is %g micro F\n',C2)
```

Scilab code Exa 8.3 To determine the power factors of a 2point4 kV phase circuit feeder

```
//To determine the power factors of a 2.4 kV phase circuit feeder
//Page 396
clc;
clear;

V=2.4; //Voltage in kV
I=200; //Load Current
P=360; //Real Load in kW
S1=V*I; //Total Load in kVA
pf1=P/S1; //Power Factor
Q1=S1*sind(acosd(pf1)); //Reactive Load

Qc=300; //Capacitor Size

Q2=Q1-Qc; //The New Reactive Load
```

```
pf2=P/sqrt((P^2)+((Q1-Qc)^2)); //Improved Power
    Factor

printf('\na) The Uncorrected power factor and
    reactive load is %g and %g kVAr\n',pf1,Q1)
printf('b) The New Corrected factor after the
    introduction of capacitor unit is %g\n',pf2)
```

Scilab code Exa 8.4 To determine the necessity of additional capacitors

```
//To determine the necessity of additional
    capacitors
//Page 398
clc;
clear;

S1=7800; //Peak Load in kVA
T=3*2000; //Total Rating of the Transformer
pf1=0.89; //Load Power Factor
TC=120/100; //Thermal Capability
Qc=1000; //Size of capacitor

P=S1*pf1; //Real Load
Q1=S1*sind(acosd(pf1)); //Reactive Load

Q2=Q1-Qc; //The New Reactive Load
pf2=P/sqrt((P^2)+((Q1-Qc)^2)); //Improved Power
    Factor

S2=P/pf2; //Corrected Apprarent power

ST=T*TC; //Transformer Capabilty

pf3=P/ST; //New Corrected Power factor required

```

```
Q2new=P*tand(acosd(pf3)); //Required Reactive Power
Qcadd=Q2-Q2new; //Additional Rating of the Capacitor

printf('\na) Since the Apparent Power(%g kVAr) is
    more than Transformer Capability (%g kVAr), \
    nHence Additional Capacitors are required\n',S2,
    ST)
printf('b) The Rating of the Addtional capacitor is
    %g kVAr\n',Qcadd)
```

Scilab code Exa 8.5 To Determine the savings in kilowatt losses

```
//To Determine the savings in kilowatt losses
//Page 411
clc;
clear;

// 1 is Total Loss Reduction due to Capacitors
// 2 is Additional Loss Reduction due to Capacitor
// 3 is Total Demand Reduction due to capacitor
// 4 is Total required capacitor additions

C90=[495165,85771,22506007,9810141]; //
    Characteristics at 90% Power Factor
C98=[491738,75343,21172616,4213297]; //
    Characteristics at 98% Power Factor

//Responsibility Factors
RF90=1;
RF98=0.9;

SLF=0.17; //System Loss Factor
FCR=0.2; //Fixed Charge Rate
DC=250; //Demand Cost
ACC=4.75; //Average Capacitor Cost
```

```
EC=0.045; //Energy Cost
Cd=C90-C98; //Difference in Characteristics

TAS=Cd(1)+Cd(2); //Total Additional Kilowatt Savings

ASDR1=Cd(1)*RF90*DC*FCR;
ASDR2=Cd(2)*RF98*DC*FCR;
TASDR=ASDR1+ASDR2; //Total Annual Savings due to demand
x=27; // Cost for Per kVA of the capacity
TASTC=Cd(3)*FCR*x; //Annual Savings due to Transmission Capacity
TASEL=TAS*SLF*EC*8760; //Savings due to energy loss reduction
TACAC=Cd(4)*FCR*ACC; //Annual Cost of Additional Capacitors
Savings=TASEL+TASDR+TASTC; //Total Savings

printf('\na) The Resulting additional savings in kilowatt losses due to power factor improvement at the substation buses is %g kW\n',Cd(1))
printf('b) The Resulting assitional savings in kilowatt  losses due to the power factor improvement for feeders is %g kW\n',Cd(2))
printf('c) The Additional Kilowatt Savings is %g kW\n',TAS)
printf('d) The Additional savings in the system kilovoltampere capacity is %g kVA\n',Cd(3))
printf('e) The Additional Capacitors required are %g kVAr\n',Cd(4))
printf('f) The Annual Savings in demand reduction due to capacitors applied to distribution substation buses is approximately is %g dollars/year\n',TASDR)
printf('g) The Annual Savings due to the additional released transmission capacity is %g dollars/year\n',TASTC)
printf('h) The Total Anuual Savings due to the
```

```
    energy loss reduction is %g dollars/year\n',TASEL
    )
printf('i) The Total Annual Cost of the additional
    capacitors is %g dollars/year\n',TACAC)
printf('j) The Total Annual Savings is %g dollars/
    year\n',Savings)
printf('k) No, Since the total net annual savings is
     not zero\n')
```

Chapter 9

Distribution System Voltage Regulation

Scilab code Exa 9.1 To Determine the parameters of the system regulation

```
//To Determine the parameters of the system
    regulation
//Page 468
clc;
clear;

//Base Value
S3phib=15; //in MVA
Vllst=69; //in kV
Vllp=13.2; //in kV
Vrrb=120;

Ztpu=%i*0.08; //Transformer impedance per unit
    length
VSTpuop=1.05*exp(%i*0); //Per Unit Maximum
    Subtransmission Voltage Off Peak
```

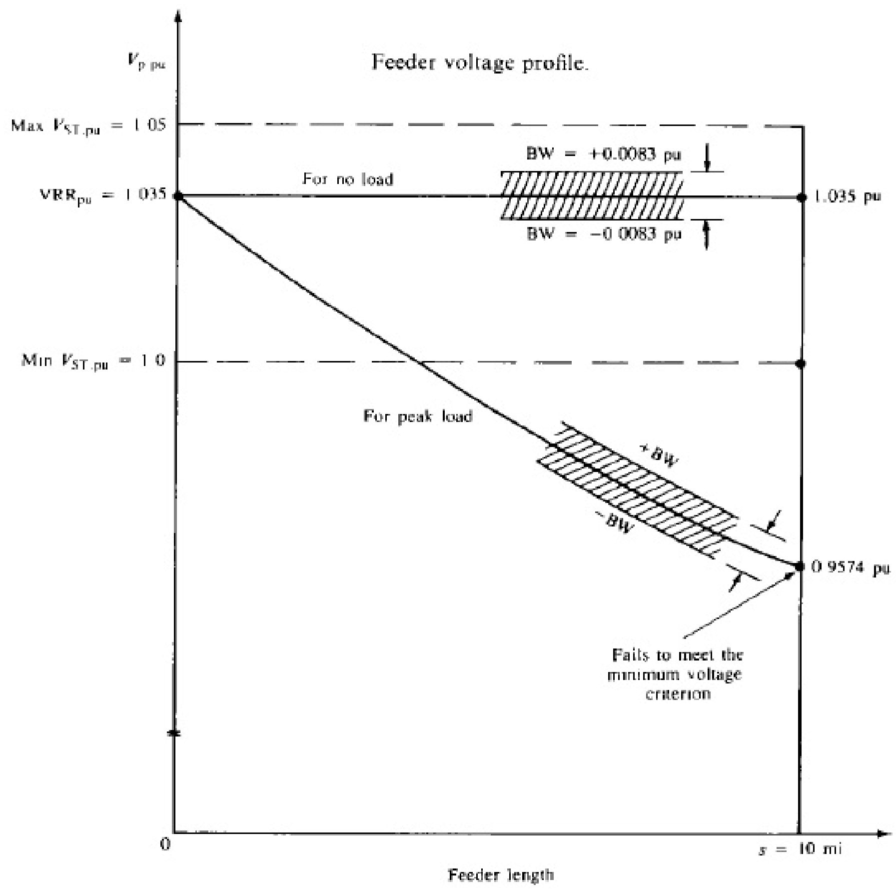

Figure 9.1: To Determine the parameters of the system regulation

```
VSTpup=1.00*exp(%i*0); //Per Unit Maximum Subtransmission Voltage Peak
pftop=0.95; //Off Peak kilovoltageamperage power factor
Sop=0.25; //Off Peak kilovoltageamperage
pftp=0.85; //Off Peak kilovoltageamperage power factor
Sp=1.0; //Off Peak kilovoltageamperage
Regpu=5/(8*100); //Regulated percent volts for 32 steps
K=3.88*(10^-6); //Drop Constant
S=4000; // Peak Load in kVA
l=10; //Length of the feeder
//Case A
Rset=0;
Xset=0;
Vpmax=1.0417;
BW=0.0083;
VRRpu=(Vpmax-BW); //Setting of VRR
VRR=(Vpmax-BW)*Vrrb;
//Case B
IPpuop=(Sop/VSTpuop)*exp(%i*acosd(pftop)*%pi/180); //No Load Primary Current at substation Off Peak
VPpuop =VSTpuop-(IPpuop*Ztpu); //Highest Allowable Primary Voltage Off Peak
IPpup=(Sp/VSTpup)*exp(-1*%i*acosd(pftp)*%pi/180); //No Load Primary Current at substation Peak
VPpup =VSTpup-(IPpup*Ztpu); //Highest Allowable Primary Voltage  Peak

Step1=(abs(VPpuop)-VRRpu)/Regpu; //Off Peak Number Steps
//To find the positive step value
Step2=-1*(abs(VPpup)-VRRpu)/Regpu; // Peak Number Steps


//Case C
```

```
//At Peak Load Primary Voltages
MaxVpp=1.075; //Max
MinVpp=1.000; //Min

TVDpu=K*S*l/2; //Total Voltage Drop
MinVPpu=VRRpu-TVDpu;

//At Annual Peak Load Primary Voltages
APMaxVPpu=MaxVpp-BW; //Max
APMinVPpu=MinVpp+BW; //Min

//At no load Load Primary Voltages
NLMaxVPpu=Vpmax-BW; //Max
NLMinVPpu=APMinVPpu; //Min

printf('\na)The Setting of the VRR for the highest
    allowable primary voltage is %g V\n',VRR)
printf('b) The Maximum Number of Steps of buck and
    boost for:\n')
printf('Off Peak : %g steps\n',ceil(Step1))
printf('Peak : %g steps\n',ceil(Step2))
printf('c) At Annual Load, Significant Values on
    Voltage Curve\n')
printf('The Total Voltage Drop is %g pu V\n',TVDpu)
printf('The Minimum Feeder Voltage at the end of the
     feeder is %g\n',MinVPpu)
printf('The Minimum and Maximum Primary Voltages at
    Peak Load is %g pu V and %g pu V\n',APMaxVPpu,
    APMinVPpu)
printf('The Minimum and Maximum Primary Voltages at
    Peak Load is %g pu V and %g pu V\n',NLMaxVPpu,
    NLMinVPpu)
```

Scilab code Exa 9.2 To Determine the distance at which the regulator must be located

```
//To Determine the distance at which the regulator must be located
//Page 472
clc;
clear;

//Terms from previous example
TVDpu=0.0776; //Total Voltage Drop
VRRpu=1.035; //Setting Voltage of Regulator
l=10;//Length of the Feeder

//Primary voltages for various cases
VPpua=1.01;
VPpub=1.00;

s1=poly(0,'s1'); //Variable Value of Regulator length
//Function to find the equation for the regulator distance
deff('x=dist(y)','x=(s1*(2-(s1/l))/l)-((VRRpu-y)/TVDpu)')

//Different Cases
Xa=dist(VPpua);
Xb=dist(VPpub);

s1a=roots(Xa);
if((abs(l-s1a(1))+(l-s1a(1)))==0)
    s1a=s1a(2);
else
    s1a=s1a(1);
end

s1b=roots(Xb);
if((abs(l-s1b(1))+(l-s1b(1)))==0)
    s1b=s1b(2);
else
    s1b=s1b(1);
```

```
end

printf('\na) The Regulator must be placed at %g
    miles from the start of the feeder\n',s1a)
printf('b) The Regulator must be placed at %g miles
    from the start of the feeder\n',s1b)
printf('c) The Advantage of a over b is that it can
    compensate for future growth\n')
```

Scilab code Exa 9.3 To Determine the Necessary minimum kilovoltampere size of the regulator

```
//To Determine the Necessary minimum kilovoltampere
    size of the regulator
//Page 473
clc;
clear;

l=10; //Length of the feeder
S3phi=4000; //Annual Peak Load in kVA
VPpu=1.01; //Primary Feeder Voltage
s1=1.75; // Distance of the Regulator
Rmax=10/100; //Regulation Percent

S=S3phi*(1-(s1/l)); //Uniformly Distributed three
    phase load
Sph=S/3; //Single Phase Load

Sreg=Rmax*Sph; //Regulated Size

printf('\nThe Calculated Circuit Kilovoltampere Size
     is %g kVA, \nAnd The corresponding Minimum
    kilovoltampere size of the regulator size can be
```

Table 9-3 Some typical single-phase regulator sizes

Single-phase kVA	Volts	Amps	CT_P*	PT_N†
25	2500	100	100	20
⋮	⋮	⋮	⋮	⋮
125	2500	500	500	20
38.1	7620	50	50	63.5
57.2	7620	75	75	63.5
76.2	7620	100	100	63.5
114.3	7620	150	150	63.5
167	7620	219	250	63.5
250	7620	328	400	63.5

* Ratio of the current transformer contained within the regulator. (Here, the ratio is the high-voltage-side ampere rating because the low-voltage rating is 1.0 A.)

† Ratio of the potential transformer contained within the regulator. (All potential transformer secondaries are 120 V.)

Figure 9.2: To Determine the Necessary minimum kilovoltampere size of the regulator

```
found as 114.3 kVA\n',Sreg)
```

Scilab code Exa 9.4 To specify the best settings for regulation

```
//To specify the best settings for regulation
//Page 474
clc;
clear;

s1=1.75; //As Found in Example 2
VRRpu=1.035; //As R and X are zero , the Settings turn out to produce this

//Parameters of Distribution
K=3.88*(10^-6);
S=3300;
l=10; //length of the line

VDpu=K*S*(l-s1)/2; //Per unit voltage drop

VP=VRRpu-VDpu; //Primary Feeder Voltage

//We Obtain VDs = K*(S3-((S3*s)/l))s+K(S*s/l)s/2;
//We take various values of s and carry out the computation and hence form the table 9-4 given given in the result file

//We Obtain from the voltage drop value for any give point s between the substation and the regulator station as
//VDs=I(r.cos(theta)+ del sin(theta))s*(1-(s/(2*l)))

//We finally Get VDs = 3.88 * (10^-6) * (3300-(3300s/8.25))s+3.88*(10^-6)*(3300s/8.25)*s/2

```

Table 9-5 For annual peak load

s, mi	VD_s, pu V	$V_{P,pu}$, pu V
0.00	0.00	1.0337
0.75	0.0092	1.0245
2.25	0.0157	1.0088
4.25	0.0155	0.9933
6.25	0.0093	0.9840
8.25	0.0031	0.9809

Table 9-4 For annual peak load

s, mi	VD_s, pu V	$V_{P,pu}$, pu V
0.0	0.0	1.035
0.5	0.0076	1.0274
1.0	0.0071	1.0203
1.5	0.0068	1.0135
1.75	0.025	1.010

Figure 9.3: To specify the best settings for regulation

```
//Again for different values of s we get the table
   9-5

printf('a)The Best Settings for LDC''s R and X, and
   for the VRR\n')
printf('The best settings for LDC of the regulator
   are when settings for both R and X are zero and
   VRRpu = %g pu V\n',VRRpu)
printf('b)The Voltage Drop occuring in the feeder
   portion between the regulating point and the end
   of the feeder is %g pu V\n',VDpu)
printf('The Result Files give the Profiles and
   relevant information about the solution\n')
```

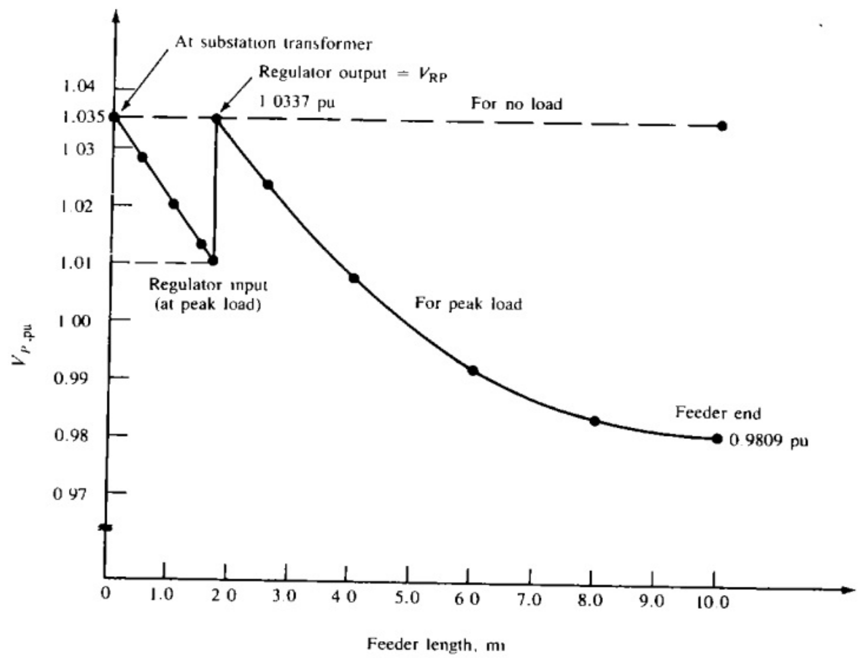

Figure 9-14 Feeder voltage profiles for zero load and for the annual peak load.

Figure 9.4: To specify the best settings for regulation

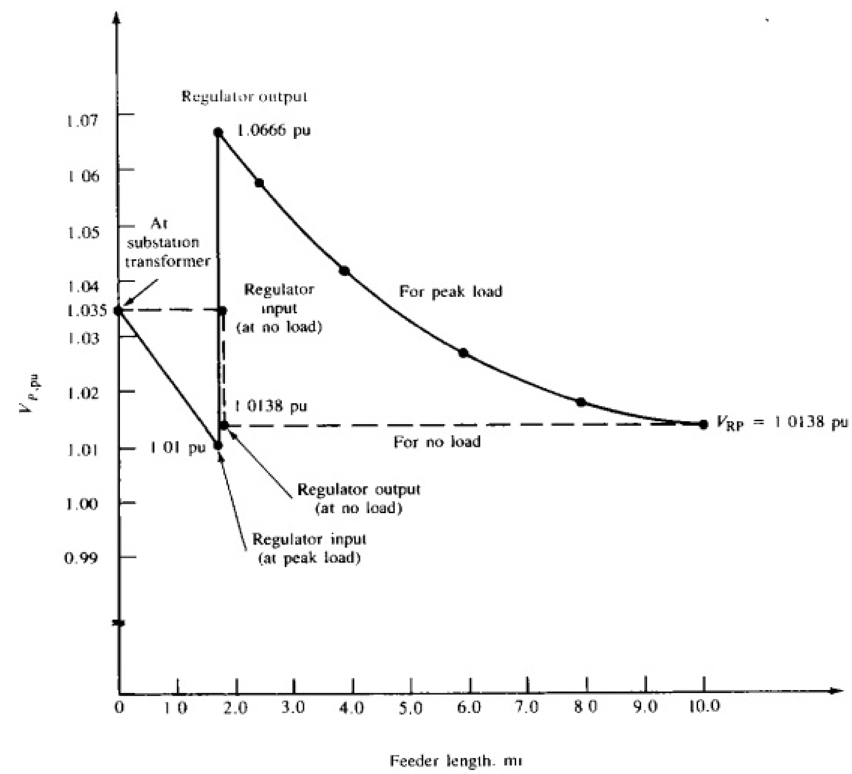

Figure 9-15 Voltage profiles.

Figure 9.5: To determine the setting of the regulator so that the voltage criteria is met

Scilab code Exa 9.5 To determine the setting of the regulator so that the voltage criteria is met

```
//To determine the setting of the regulator so that
   the voltage criteria is met
//Page 478
clc;
clear;
l=10; //Length of the feeder
s1=1.75;
ra=0.386;
xa=0.4809;
xd=0.1802;
xL=xa+xd;
Vb=120;
pf=0.85; //Power Factor
S=1100; //Load in kVA
Vln=7.62; //line to neutral voltage in kV
Reff=ra*(l-s1)/2;
Xeff=xL*(l-s1)/2;

//Primary Ratings
CTp=150; //Current Tranformer
PTn=63.5; //POtential Transformer

//R Value of the dial
Rset=(CTp/PTn)*Reff;
Rsetpu=Rset/Vb;

//X value of the dial
Xset=(CTp/PTn)*Xeff;
Xsetpu=Xset/Vb;

VRP=1.0138; //Assumption Made on the Regulating
```

```
   Point
//Output Voltage of the Regulator
Vreg=VRP+((S/Vln)*((Rset*pf)+(Xset*sind(acosd(pf))))
   /(CTp*Vb));


printf('\na) The Regulating Voltage is %g pu V\n',
   Vreg)
printf('As per Table 9-6; the primary voltage
   criteria are met by using the R and X settings\n'
   )
printf('b) The Voltage Profiles are given in the
   result file attached\n')
```

Scilab code Exa 9.6 To determine the number of steps of buck and boost the regulators will achieve

```
//To determine the number of steps of buck and boost
    the regulators will achieve
//Page 480
clc;
clear;

//From Problems 4 and 5 the co-effcients are
   obtained
VRRpu=1.035;
Vreg4=1.0337;
Vreg5=1.0666;
VRP4=1.0337;
VRP5=1.0138;
Vmin=1.010; //For s= 1.75

//Steps
Buck4=(VRRpu-VRP4)/(0.00625);
Buck5=(VRRpu-VRP5)/(0.00625);
```

```
Boost4=(Vreg4-Vmin)/(0.00625);
Boost5=(Vreg5-Vmin)/(0.00625);

printf('\na) The Number of steps of buck and number
    is steps of boost in example 9-4 is %g and %g
    respectively\n',Buck4,Boost4)
printf('\nb) The Number of steps of buck and number
    is steps of boost in example 9-5 is %g and %g
    respectively\n',Buck5,Boost5)
```

Scilab code Exa 9.8 To Determine the necessary settings of Regulators

```
//To Determine the necessary settings of Regulators
//Page 482
clc;
clear;

l=3;//Length of the line
Vlc=2450; //Regulated Voltage
Vcp=12800; //Primary of customer transformer
//Base Values
Vlbp=2400; //Primary Bus Voltage of Customer's Bus(
    Low Voltage)
Vlbs=4160;//Secondary Bus Voltage of Customer's Bus
Sb=5000; //Power in kVA
r=0.3; //Line Resistance per mile
x=0.8; //Line Reactance per mile
Vhbp=7390;//Primary Bus Voltage of High Voltage Bus
Vhbs=12800;//Secondary Bus Voltage of High Voltage
    Bus
PTn=63.5; //Potential Transformer Turns Ratio
CTp=250;//Current Transformer Turns Ratio
VRP=Vlc/Vlbp;//Voltage at RP
Vll=Vhbs/1000; //Line Voltage
VBsec=Vcp/(sqrt(3)*PTn); //Secondary Reading of the
```

```
Customer Transformer

VRRset=VRP*VBsec; //Setting of the voltage-setting dial of VRR

Zb=(Vll^2)*1000/Sb; //Applicable Impedance Base
Ztpu=0.05*%i;//Transformer Impedance per unit
Zt=Ztpu*Zb;//Transformer Impedance

//Effective Resistances and Reactances
Reff=(r*l)+real(Zt);
Xeff=(x*l)+imag(Zt);

Rset=CTp*Reff/PTn; //X Dial Setting of LDCs
Xset=CTp*Xeff/PTn; //X Dial Setting of LDCs

printf('\na) The Necessary Setting of the voltage-setting dial of the VRR of each single phase regulator in use is %g V\n',VRRset)
printf('b) R and X dial settings of LDS is %g ohm and %g ohm respectively\n',Rset,Xset)
```

Scilab code Exa 9.9 To Determine the Design Parameters of a Distributed System

```
//To Determine the Design Parameters of a Distributed System
//Page 484
clc;
clear;

VPpu=1.035; //Primary Feeder Voltage per unit
TVDpu=0.0776; //Total Voltage Drop of Feeder
```

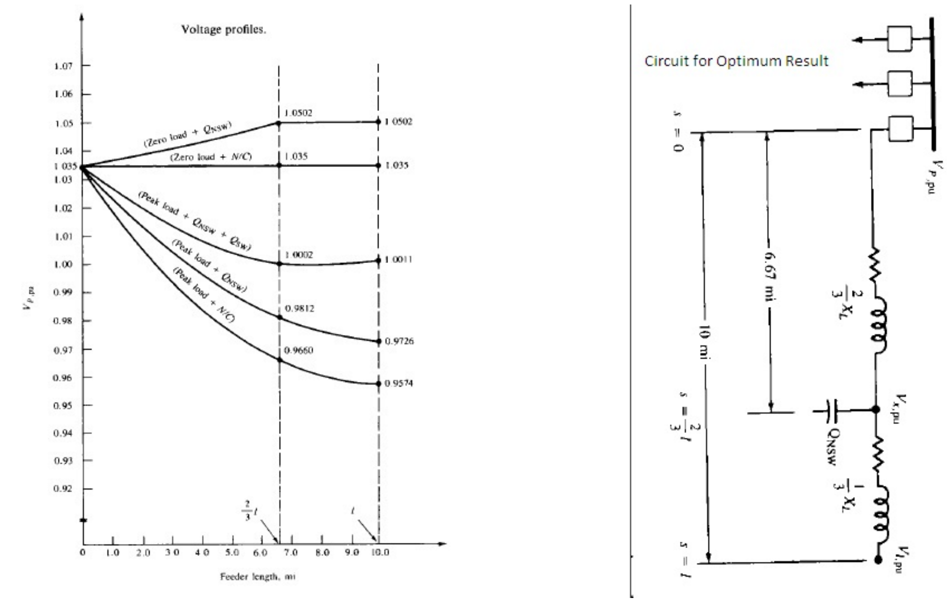

Figure 9.6: To Determine the Design Parameters of a Distributed System

```
Vll=13.2; //Line Voltage in kV
Vlpuqsw=1;//New Voltage at the End of the Feeder due to Qsw at annual peak load
XL=0.661; //Inductive Reactance per mile
Pl=3400; //Real Power
Ql=2100; //Reactive Power
l=10; //Length of the Feeder in Miles
Lf=0.4; //Load Factor
CR=0.27; //Total Capacitor Compensation Ratio For the Above Load Factor
QNSW=CR*Ql; //Required Size of the Nonswitched capacitor bank
s=2*l/3; //Loacation of Nonswitched capacitor bank for Optimum Result
VRpu=QNSW*(2*XL*l/3)/(1000*(Vll^2)); //Per Unit Voltage Rise
VDspu=TVDpu*s*(2-(s/l))/l; //Voltage drop for the uniformaly distributed load

```

```
VSpu=VPpu-VDspu;//Feeder Voltage at 2l/3 distance

nVSpu=VSpu+VRpu; //New Voltage Rise when there is a fixed capacitor bank

Vlpu=VPpu-TVDpu; //When No Capcacitor bank is there, the voltage at the end of the feeder

nVlpu=Vlpu+VRpu; //When Capcacitor bank is there, the voltage at the end of the feeder
VRpuqsw=Vlpuqsw-nVlpu; //Required Voltage Rise

Q3phisw=1000*(Vll^2)*VRpuqsw/(XL*l); //Required Size of the Capacitor Bank

//Let us assume the 15 single phase standard 50 kVAr Capacitor Units = 750 kVAr

SQ3phisw=750; //Selected Capacitor Bank

RVRlpu=VRpuqsw*SQ3phisw/Q3phisw; //Resultant Voltage Rises at distance l
RVRspu=RVRlpu*s/l; //Resultant Voltage Rises at distance s

//At Peak Load when both the Non-Switched and Switched Capacitor Banks are on

PVspu=nVSpu+RVRspu; //Voltage at s
PVlpu=nVlpu+RVRlpu; //Voltage at l

printf('\na) The NSW Capacitor Rating is %g kVAr, Which means 2 100kVAr Capacitor Banks per phase\n',QNSW)
printf('\nb) Voltage Rise per unit is %g pu V\n',VRpu)
printf('i) When the No Capacitor Bank is Installed \n')
```

```
printf('Voltage at %g miles is %g pu V\n',s,VSpu)
printf('Voltage at %g miles is %g pu V\n',l,Vlpu)
printf('ii) When the Fixed Capacitor Bank is
    Installed \n')
printf('Voltage at %g miles is %g pu V\n',s,nVSpu)
printf('Voltage at %g miles is %g pu V\n',l,nVlpu)
printf('\nc) At Annual Peak Load, The Size of
    Capacitor Bank Required is %g\n',Q3phisw)
printf('The Voltage Rise at the Annual Load for the
    Required Capacitor Bank is %g pu V\n',VRpuqsw)

//Note That The Calculations are highly accurate,
    Rounding of Terms hasn't be emplyed
```

Scilab code Exa 9.10 To Determine the proper 3 phase capacitor bank

```
//To Determine the proper 3 phase capacitor bank
//Page 488
clc;
clear;

V=12.8; //Voltage in kV
xl=0.8; //Reactance per unit length
l=3; //Distance of the line
Xl=xl*l; //Effective Reactance of the the Line
pf=0.8; //Initial Power Factor
pfn=0.88; //New Improved Power Factor
Qcu=150; //Capacity of each unit available
XT=1.6384; //Reactance of the transformer

S3phi=5000*exp(%i*%pi*acosd(pf)/180); //Presently
    existing Load

//For New Load Real part of the Load doesn't Change

```

```
QLnew=real(S3phi)*tand(acosd(pfn)); //The Required VAr

S3phinew =sqrt((real(S3phi)^2)+(QLnew^2)); //New Apparent Power

Qc=imag(S3phi)-QLnew; //Minimum Size of capacitor bank;

N=ceil(Qc/Qcu); //Number of Units Required
Qcn=N*Qcu; //Required VAr

XL=Xl+XT; //Total Reactance

VRpu=Qcn*XL/(1000*(V^2)); //Voltage Rise Per unit

printf('\nThe The Voltage Rise found out is %g pu V, which is greater than the voltage rise criterion .\nHence %g Capacitor units of %g kVAr must be installed\n',VRpu,N,Qcu)
```

Scilab code Exa 9.11 To Determine the Voltage dip due to 10hp single phase motor

```
//To Determine the Voltage dip due to 10hp single phase motor
//Page 493
clc;
clear;

Skva=6.3*(10^3); //Starting kVA per HP of the Motor
HPmotor=10; //Power Rating
Vll=7.2*(10^3); //Line Voltage
I3phi=1438; //Fault Current

```

```
Sstart=Skva*HPmotor; //Starting kVA

VDIP=120*Sstart/(I3phi*Vll); //Voltage Dip

printf('\na) The Voltage dip due to the motor start
   is %g V\n',VDIP)
printf('b) From the Permissible voltage flicker
   limit curve, The Voltage dip of 0.73 V\nwith a
   frequency of 15 times per hour is in the
   satisfactory flicker zone\n and therefore is not
   objectionable to the immediate customers\n')
```

Scilab code Exa 9.12 To determine the voltage dip due to the motor start for a 100 hp 3Phase Motor

```
//To determine the voltage dip due to the motor
   start for a 100 hp 3Phase Motor
//Page 495
clc;
clear;

Skva=5.6*(10^3); //Starting kVA per HP of the Motor
HPmotor=100; //Power Rating
Vll=12.47*(10^3); //Line Voltage
I3phi=1765; //Fault Current

Sstart=Skva*HPmotor; //Starting kVA

VDIP=69.36*Sstart/(I3phi*Vll); //Voltage Dip

printf('\na) The Voltage dip due to the motor start
   is %g V\n',VDIP)
printf('b) From the Permissible voltage flicker
   limit curve, The Voltage dip of 1.72 V\nwith a
   frequency of three times per hour is in the
```

```
satisfactory flicker zone\n and therefore is not
objectionable to the immediate customers\n')
```

Chapter 10

Distribution System Protection

Scilab code Exa 10.1 To Determine the necessary realy and recloser coordination

```
//To Determine the necessary realy and recloser
   coordination
//Page 542
clc;
clear;

//For Recloser
InstT=0.03; //From Curve A //Instaneous Time
TimeD=0.17; //From Curve B //Time Delay
//For Relay
PickU=0.42; //From Curve C //Pick Up
Reset=(1/10)*60; //Assuming a 60 s reset time for
   the relay with number 10 time dial setting
RecloserOT=1; //Assumed Recloser Open Time

RelayCTI=InstT/PickU; //Relay Closing Travel during
   instantaneous operation
RelayRTI=(-1)*RecloserOT/Reset; //Relay Reset Travel
    during instantaneuos

```

```
RelayCTD=TimeD/PickU;
RelayRTD=(-1)*RecloserOT/Reset; //Relay Reset Travel
    during trip
NetRelayTravel=RelayCTD-RelayRTD;

printf('\nDuring Instantaneous Operation\n')
printf('|Relay Closing Travel| < |Relay Rest Travel
   |\n')
printf('|%g percent| < |%g percent|\n',RelayCTI*100,
   RelayRTI*100)

printf('\nDuring the Delayed Tripping Operations\n')
printf('Total Relay Travel is from:\n')
printf('%g percent to %g percent to %g percent\n',
   RelayCTD*100,RelayRTD*100,RelayCTD*100)
printf('Since this Net Total Relay Travel is less
   than 100 percent, \nthe desired recloser to relay
    coordination is accomplished\n')
```

Scilab code Exa 10.2 To Determine the Fault parameters of Rural Substation

```
//To Determine the Fault parameters of Rural
   Substation
//Page 555
clc;
clear;

Vln=7200; //Line to Neutral Voltage
Vll=12470; //Line to Line Voltage
Z1sys=0.7199+(%i*3.4619); //system impedance to the
   regulated 12.47kV bus
ZGsys=0.6191+(%i*3.3397);//system impedance to
   ground
l=2; //Distance of the Fault from the substation
```

```
//From Table 10-7; Various Paramters Can Be found out
z0a=0.1122+(%i*0.4789);
z011=(-0.0385-(%i*0.0996));
z1=0.0580+(%i*0.1208);
C=5.28; //Cable constant

Z0ckt=2*(z0a+z011)*C; //Zero Sequence Impedance
Z1ckt=2*z1*C; //Positive Sequence Impedance
ZGckt=((2*Z1ckt)+Z0ckt)/3; //Impedance to ground of line
//Note That the calculation of the above term is wrong in the text book

Z1=Z1sys+Z1ckt; //Total Positive Sequence
ZG=ZGsys+ZGckt; //Total impedance to ground

If3phi=Vln/abs(Z1); //Three Phase Fault at point 10
IfLL=0.866*If3phi; //Line to Line Fault at point 10
IfLG=Vln/(abs(ZG)); //Single Line to Ground Fault

printf('\na) The Zero and Postive sequence impedance of the line to point 10 are:\n')
disp(Z0ckt)
disp(Z1ckt)
printf('b) The impedance to ground of the line to point 10\n')
disp(ZGckt)
printf('c) The Total positive sequence impedance including system impedance is\n')
disp(Z1)
printf('d) The Total Impedance to ground to point 10 including system impedance is\n')
disp(ZG)
printf('All the Above impedances are in ohm\n')
printf('e) The Three phase fault current at point 10 is %g A\n',If3phi)
printf('f) The line to line fault current at point
```

Table 10-13

Bus	Fault	Maximum generation A	Maximum generation MVA	Minimum generation A	Minimum generation MVA
1	3ϕ	4922	588.2	3012.3	360
	$L-L$	4266.5	294.1	2608.7	180.2
	$L-G$	7383	294.1	4482.5	178.6
2	3ϕ	3149.6	68.0	2930.3	63.29
	$L-L$	2727.6	34.0	2537.7	36.6
	$L-G$	3275.9	23.6	3114.3	23.42

Figure 10.1: To Determine system parameters for various stabilities

```
    10 is %g A\n',IfLL)
printf('g) The Line to Ground Current at point 10 is
    %g A\n',IfLG)
```

Scilab code Exa 10.3 To Determine system parameters for various stabilities

```
//To Determine system parameters for various
    stabilities
//Page 562
clc;
clear;
```

```
St=5*(10^6); //Capacity of Transformer
Zt=%i*0.065; //Transformer Reactance
SB3phi=1*(10^6); //3 Phase Power Base
VBLL=69*(10^3); //Line to line voltage
VBLLn=12.47*(10^3); //Line To line voltage
Vf=1; //Per Unit Value of Voltage
Zb=(VBLL^2)/SB3phi; //Base Impedance

//Zckt and Zf and Zt are Zero for Bus 1
//Zckt and Zf are Zero for Bus 2
//Power Generation of the system
SMax=600*(10^6); //Maximum
SMin=360*(10^6); //Minimum

Xt=0.065; //Transformer Reactance in per unit
MZsysmax=(VBLL^2)/SMax; //System Impedance at Maximum Power Generation
Ib=SB3phi/(sqrt(3)*VBLL); //Base Current
Zsysmaxpu=MZsysmax*exp(%i*%pi*90/180)/Zb; //System Impedance Phasor
//Three Phase Fault Current
If3phimaxpu1=abs(Vf/(Zsysmaxpu));
If3phimax1=If3phimaxpu1*Ib;
Sf3phimax1=sqrt(3)*VBLL*If3phimax1/1000000;

//Line to Line Fault Current
IfLLmax1=0.866*If3phimax1;
SfLLmax1=VBLL*IfLLmax1/1000000;

//Line to Ground Fault
IfLGmaxpu1=abs(3*Vf/((2*Zsysmaxpu)));
IfLGmax1=IfLGmaxpu1*Ib;
SfLGmax1=VBLL*IfLGmax1/(1000000*sqrt(3));

Stn=SB3phi; //Numreical Value is Equal
Ztn=Zt*(Stn/St); //New Per Unit Transformer Reactance
//New Base Values
```

```
Zbn=(VBLLn^2)/SB3phi;
Ibn=Stn/(sqrt(3)*VBLLn);

//Three Phase Fault Current
If3phimaxpu2=abs(Vf/(Zsysmaxpu+Ztn));
If3phimax2=If3phimaxpu2*Ibn;
Sf3phimax2=sqrt(3)*VBLLn*If3phimax2/1000000;

//Line to Line Fault Current
IfLLmax2=0.866*If3phimax2;
SfLLmax2=VBLLn*IfLLmax2/1000000;

//Line to Ground Fault
IfLGmaxpu2=abs(3*Vf/((2*Zsysmaxpu)+(3*Ztn)));
IfLGmax2=IfLGmaxpu2*Ibn;
SfLGmax2=VBLLn*IfLGmax2/(1000000*sqrt(3));

//Minimum Power Generation
MZsysmin=(VBLL^2)/SMin; //System Impedance at Maximum Power Generation
Ib=SB3phi/(sqrt(3)*VBLL); //Base Current
Zsysminpu=MZsysmin*exp(%i*%pi*90/180)/Zb; //System Impedance Phasor
//Three Phase Fault Current
If3phiminpu1=abs(Vf/(Zsysminpu));
If3phimin1=If3phiminpu1*Ib;
Sf3phimin1=sqrt(3)*VBLL*If3phimin1/1000000;

//Line to Line Fault Current
IfLLmin1=0.866*If3phimin1;
SfLLmin1=VBLL*IfLLmin1/1000000;

//Line to Ground Fault
IfLGminpu1=abs(3*Vf/((2*Zsysminpu)));
IfLGmin1=IfLGminpu1*Ib;
SfLGmin1=VBLL*IfLGmin1/(1000000*sqrt(3));

Stn=SB3phi; //Numreical Value is Equal
```

```
Ztn=Zt*(Stn/St); //New Per Unit Transformer
    Reactance
//New Base Values
Zbn=(VBLLn^2)/SB3phi;
Ibn=Stn/(sqrt(3)*VBLLn);

//Three Phase Fault Current
If3phiminpu2=abs(Vf/(Zsysminpu+Ztn));
If3phimin2=If3phiminpu2*Ibn;
Sf3phimin2=sqrt(3)*VBLLn*If3phimin2/1000000;

//Line to Line Fault Current
IfLLmin2=0.866*If3phimin2;
SfLLmin2=VBLLn*IfLLmin2/1000000;

//Line to Ground Fault
IfLGminpu2=abs(3*Vf/((2*Zsysminpu)+(3*Ztn)));
IfLGmin2=IfLGminpu2*Ibn;
SfLGmin2=VBLLn*IfLGmin2/(1000000*sqrt(3));

printf('\na) For Maximum Power Generation:\n')
printf('Bus 1\n')
printf('3 phase fault current is %g A and %g MVA\n',
    If3phimax1,Sf3phimax1)
printf('Line to Line fault current is %g A and %g
    MVA\n',IfLLmax1,SfLLmax1)
printf('Line to ground fault current is %g A and %g
    MVA\n',IfLGmax1,SfLGmax1)
printf('Bus 2\n')
printf('3 phase fault current is %g A and %g MVA\n',
    If3phimax2,Sf3phimax2)
printf('Line to Line fault current is %g A and %g
    MVA\n',IfLLmax2,SfLLmax2)
printf('Line to ground fault current is %g A and %g
    MVA\n',IfLGmax2,SfLGmax2)
printf('\nb) For Minimum Power Generation:\n')
printf('Bus 1\n')
printf('3 phase fault current is %g A and %g MVA\n',
```

```
If3phimin1,Sf3phimin1)
printf('Line to Line fault current is %g A and %g MVA\n',IfLLmin1,SfLLmin1)
printf('Line to ground fault current is %g A and %g MVA\n',IfLGmin1,SfLGmin1)
printf('Bus 2\n')
printf('3 phase fault current is %g A and %g MVA\n',If3phimin2,Sf3phimin2)
printf('Line to Line fault current is %g A and %g MVA\n',IfLLmin2,SfLLmin2)
printf('Line to ground fault current is %g A and %g MVA\n',IfLGmin2,SfLGmin2)

//Note that 0.00166666666 is not rounded as 0.0017
//Hence you find all the answers close by
```

Scilab code Exa 10.4 To Determine the sequence impedance values

```
//To Determine the sequence impedance values
//Page 572
clc;
clear;

//Percent Impedances of the substation transformer
Rtp=1;
Ztp=7;
Xtp=sqrt((Ztp^2)-(Rtp^2));
Ztpu=Rtp+(%i*Xtp); //Transformer Impedance
Vll=12.47; //Line to Line voltage in kV
Vln=7.2; //Line to Neutral Voltage
V=240; //Secondary Voltage
St=7500; //Rating of the transformer in kVA
Sts=100; //Rating of Secondary Transformer
Ztp=Ztpu*((Vll^2)*10/St);
SSC=complex(.466,0.0293);
```

```
//From Table 10-7
Z1=0.0870+(%i*0.1812);
Z0=complex(0.1653,0.4878);

ZG=((2*Z1)+Z0)/3;//Impedance to Ground

Zsys=0 ; //Assumption Made
Zeq=Zsys+Ztp+ZG; //Equivalent Impedance of the
    Primary

PZ2=Zeq*((V/(Vln*1000))^2); //Primary Impedance
    reffered to secondary

//Distribution Tranformer Parameters
Rts=1;
Zts=1.9;
Xts=sqrt((Zts^2)-(Rts^2));
Ztspu=complex(Rts,Xts);

Zts=Ztspu*((V/1000)^2)*10/Sts; //Distribution
    Transformer Reactance

Z1SL=(60/1000)*SSC; //Impedance for 60 feet

Zeq1=PZ2+Zts+Z1SL; //Total Impedance to the fault in
     secondary

IfLL=V/abs(Zeq1); //Fault Current At the secondary
    fault point F



printf('\na) The Impedance of the substation in ohms
    \n')
disp(Ztp)
printf('b) The Positive And Zero Sequence Impedances
     in  ohms\n')
disp(Z1)
```

```
disp(Z0)
printf('c) The Line to Ground impedance in the primary system in ohms\n')
disp(ZG)
printf('d) The Total Impedance through the primary in ohms\n')
disp(Zeq)
printf('e) The Total Primary Impedance referred to the secondary in ohms\n')
disp(PZ2)
printf('f) The Distribution transformer impedance in ohms\n')
disp(Zts)
printf('g) the Impedance of the secondary cable in ohms\n')
disp(Z1SL)
printf('h) The Total Impedance to the fault in ohms\n')
disp(Zeq1)
printf('i) The Single Phase line to line fault for the 120/240 V three-wire service in amperes is %g A\n',IfLL)
```

Chapter 11

Distribution System Reliability

Scilab code Exa 11.1 To Determine the Approximate value of the component reliability

```
//To Determine the Approximate value of the component reliability
//Page 598
clc;
clear;
Rsys=0.99 //Minimum Acceptable System Reliabilty
n=15; //Number of identical Components
q=(1-Rsys)/n; //Probability of component failure
Ri=1-q; //Approximate value of the component reliability

printf('The Approximate Value of The component reliability is %g\n',Ri)
```

Scilab code Exa 11.2 To Determine the fault components of the system

```
//To Determine the fault components of the system
```

```
//Page 606
clc;
clear;
L=4; //Total Length of the cable
Lov=3;//Length of Overhead Cable
Lu=L-Lov; //Length of Underground Cable
Nct=2; //Number of circuit terminations
T=10; //No of years for which the record is shown

Fov=2; // Faults Per Mile of the Over Head Cable
Fu=1; //Faults Per Mile of The Underground cable

Ct=0.3/100// Cable Termination Fault Rate

//Repair Time
Tov=3; //Over Head
Tu=28; //Underground
Tct=3; //Cable Termination

lamdaFDR= (Lov*Fov/T)+(Lu*Fu/T)+(2*Ct); //Total
    Annual Fault Rate

rFDR=((Tov*Lov*Fov/T)+(Tu*Lu*Fu/T)+(2*Ct*Tct))/
    lamdaFDR; //Annual Fault Restoration Time

mFDR=8760-rFDR; //Annual Mean Time of Failure

UFDR=rFDR*100/(rFDR+mFDR); //Unavailability of
    Feeder
AFDR=100-UFDR; //Availability of Feeder

printf('a) The Total Annual Fault Rate is %g faults
    per year\n',lamdaFDR)
printf('b) The Annual Fault Restoration Time is %g
    hours per fault per year\n',rFDR)
printf('c) Unavailability of the feeder is %g
    percent\n',UFDR)
printf('d) Availability of the feeder is %g percent\
```

```
n',AFDR)
```

Scilab code Exa 11.3 To Determine the Annual Fault properties for A B C Customers

```
//To Determine the Annual Fault properties for A B C Customers
//Page 608
clc;
clear;

//Annual average Fault rates
Fm=0.08;
Fl=0.2;


//Average Repair Times
Rm=3.5; //Main
Rl=1.5; //Lateral
Rs=0.75;//Manual Sections

// Distances of the Lateral Feeders of A,B, and C respectively
Lla=2;
Llb=1.5;
Llc=1.5;

// Distances of the Main Feeders of A,B, and C respectively
Lma=1;
Lmb=1;
Lmc=1;

TFm=(Lma*Fm)+(Lmc*Fm)+(Lmb*Fm); //Annual Fault of the Main Sections
```

```

deff('x=SusInt(y)','x=TFm+(Fl*y)') //Function to find the Total Annual Sustained Interruption rates

//Sustained Interruption Rates for A,B and C
IrA=SusInt(Lla);
IrB=SusInt(Llb);
IrC=SusInt(Llc);

//Annual Repair time for A,B and C
rA=((Lma*Fm*Rm)+(Lmb*Fm*Rs)+(Lmc*Fm*Rs)+(Lla*Fl*Rl))/IrA;
rB=((Lma*Fm*Rm)+(Lmb*Fm*Rm)+(Lmc*Fm*Rs)+(Llb*Fl*Rl))/IrB;
rC=((Lma*Fm*Rm)+(Lmb*Fm*Rm)+(Lmc*Fm*Rm)+(Llc*Fl*Rl))/IrC;

printf('\ni) The Annual Sustained Interruption Rates for:\n')
printf('Customer A : %g faults per year\n',IrA)
printf('Customer B : %g faults per year\n',IrB)
printf('Customer C : %g faults per year\n',IrC)
printf('\nii) The Average Annual Repair Time (Restoration Time) for:\n')
printf('Customer A : %g hours per fault per year\n',rA)
printf('Customer A : %g hours per fault per year\n',rB)
printf('Customer A : %g hours per fault per year\n',rC)
```

Scilab code Exa 11.4 To Determine the Equivalent System Reliability of Each configuration

```
//To Determine the Equivalent System Reliability of
    Each configuration
//Page 612
clc;
clear;

Ri=0.85;

deff('x=relp(y,z)','x=1-((1-(Ri^y))^z)')//Equal
    Parallel Combination

deff('x=rels(y,z)','x=(1-((1-Ri)^y))^z') //Equal
    Series Combination

//Case 1: 4 elements in series

Req1= rels(1,4);

//Case 2: Two Comination of 4 elements in series,
    parallel to each other

Req2=relp(4,2);

//Case 3 : ((two elements in series)//(two elements
    in series))in series with ((two elements in
    series)//(two elements in series))

//Two Segments
R1=relp(2,2);
R2=relp(2,2);
Req3=R1*R2;

//Case 4 : (two elements in parallel)in series with
    ((three elements in series)//(three elements in
    series))

//Two Segments
R1=relp(1,2);
```

```
R2=relp(3,2);
Req4=R1*R2;

//Case 5, 4 groups of (2 elements in parallel)
    connected in series to each other
Req5=rels(2,4);

printf('The Equivalent System reliability for:\n')
printf('a) Configuration A : %g\n',Req1)
printf('b) Configuration B : %g\n',Req2)
printf('c) Configuration C : %g\n',Req3)
printf('d) Configuration D : %g\n',Req4)
printf('e) Configuration E : %g\n',Req5)
```

Scilab code Exa 11.5 To Design the system to meet the given Equivalent System Reliability

```
//To Design the system to meet the given Equivalent
    System Reliability
//Page 614
clc;
clear;

//Individual System Reliabilities
Ra=0.8;
Rb=0.95;
Rc=0.99;
Rd=0.90;
Re=0.65;

//When All Are Connected in Series

Req=Ra*Rb*Rc*Rd*Re; //Equivalent System Reliability

Rr=0.8; //Required
```

```

Rae=Rr/(Rb*Rc*Rd);

//Since Connecting the elements in parallel will increase their reliability
deff('x=rel(Ri,y,)','x=(1-((1-Ri)^y))') //Equal Only Parallel Combination

//Since Connecting the elements in parallel will increase their reliability
//Conditions to Find The Number of Elements to be used
for i= 1:10
    L=i; //Number of Time Element A is used
    R1=rel(Ra,i);
    X=R1-Rae;
    if(abs(X)+X==0)
        continue;
    else
        break;
    end
end

for i= 1:10
    M=i;//Number of Time Element E is used
    R2=rel(Re,i);
    X=R2-Rae;
    if(abs(X)+X==0)
        continue;
    else
        break;
    end
end

printf('a) The Equivalent system Reliability is %g\n',Req)
printf('b) One Each of B,C and D all connected in series are connected in series\nwith the series
```

```
combination of X(Comination of %g elements of A, All Connected in Parallel)\nand Y(Comination of %g elements of E, All Connected in Parallel) to achieve \n%g Equivalent System Realibility\n',L,M,Rr)
```

Scilab code Exa 11.6 To Find The Probability on the reliability of transformers

```
//To Find The Probability on the reliability of transformers
//Page 614
clc;
clear;

//Reliabilities of The Three Transformers
Pa=0.9;
Pb=0.95;
Pc=0.99;

//Faliures of Three Transformers
Qa=1-Pa;
Qb=1-Pb;
Qc=1-Pc;

//Probability of NO Transformer Failing
Pnf=Pa*Pb*Pc;

PfA=Qa*Pb*Pc//Probability of Transformer A Failing
PfB=Pa*Qb*Pc//Probability of Transformer B Failing
PfC=Pa*Pb*Qc//Probability of Transformer C Failing

PfAB=Qa*Qb*Pc//Probability of Transformer A and B Failing
PfBC=Pa*Qb*Qc//Probability of Transformer B and C
```

```
    Failing
PfCA=Qa*Pb*Qc//Probability of Transformer C and A Failing

Pf=Qa*Qb*Qc; //Probability of All Transformers failing

printf('\na) Probability of No Transformer Failing is %g\n',Pnf)
printf('\nb)\n')
printf('Probability of Transformer A Failing is %g\n',PfA)
printf('Probability of Transformer B Failing is %g\n',PfB)
printf('Probability of Transformer C Failing is %g\n',PfC)
printf('\nc)\n')
printf('Probability of Transformers A and B Failing is %g\n',PfAB)
printf('Probability of Transformers B and C Failing is %g\n',PfBC)
printf('Probability of Transformers C and A Failing is %g\n',PfCA)
printf('\nd) Probability of All Three Transformers Failing is %g\n',Pf)
```

Scilab code Exa 11.7 To Determine Probabilities Using Markovian Principle

```
//To Determine Probabilities Using Markovian Principle
//Page 619
clc;
```

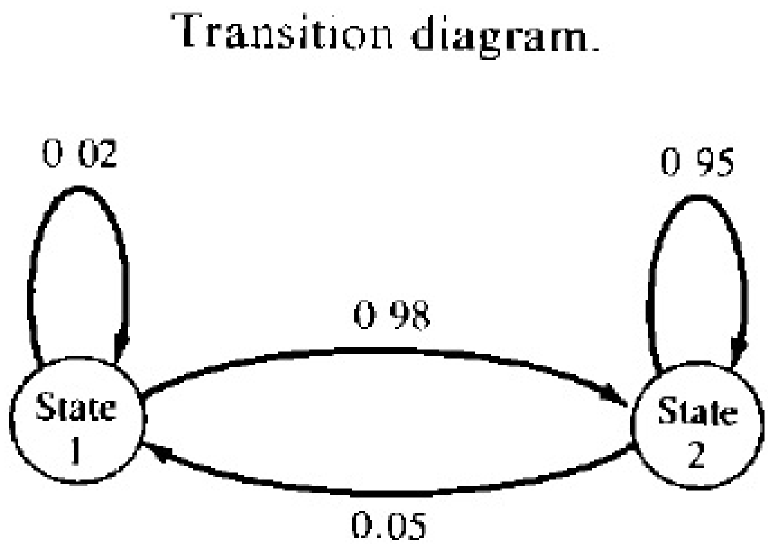

Figure 11.1: To Determine Probabilities Using Markovian Principle

```
clear;

//Conditional Probabilites Present Future
Pdd=2/100; //Down Down
Pud=5/100; //Up Down
Pdu=1-Pdd; //Down up
Puu=1-Pud; //Up Up

P=[Pdd,Pdu;Pud,Puu]; //Transition Matrix

printf('\na) The Conditional Probabilites for\n')
printf('Transformers Down in Present and Down in Future is %g\n',Pdd)
printf('Transformers Down in Present and Up in Future is %g\n',Pdd)
printf('Transformers Up in Present and Down in Future is %g\n',Pdd)
printf('Transformers Up in Present and Up in Future is %g\n',Pdd)
printf('\nb) The Transition Matrix is\n')
disp(P)
printf('\nc) The Transition Diagram can be viewed with the result file attached to this code\n')
```

Scilab code Exa 11.8 To Determine the Conditional Outage Probabilites

```
//To Determine the Conditional Outage Probabilites
//Page 620
clc;
clear;

//Conditional Outage Probabilites From The Table Given
```

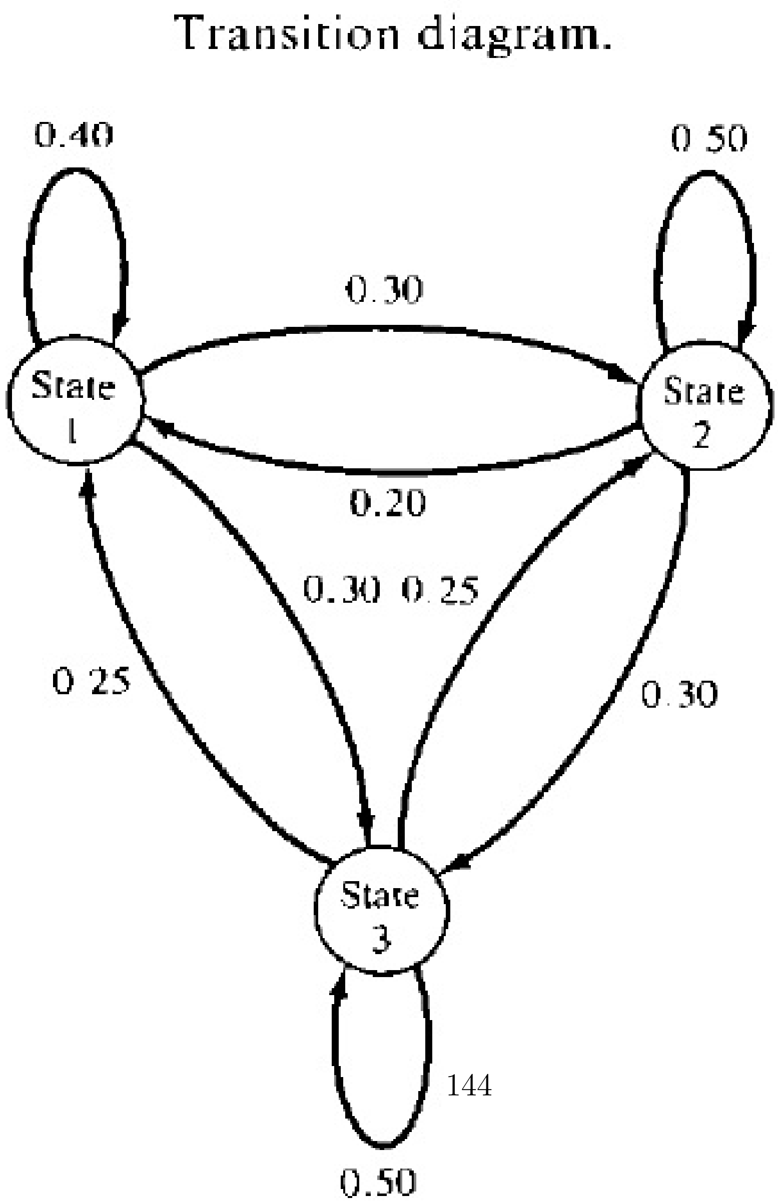

Figure 11.2: To Determine the Conditional Outage Probabilites

```
P11=40/100;
P12=30/100;
P13=30/100;
P21=20/100;
P22=50/100;
P23=30/100;
P31=25/100;
P32=25/100;
P33=50/100;

//Transition Matrix
P=[P11,P12,P13;P21,P22,P23;P31,P32,P33];

printf("\na) The Conditional Outage Probabilites for
    :\n")
printf("Presently Outaged Feeder is 1, Next Outaged
    Feeder is 1 is %g\n",P11)
printf("Presently Outaged Feeder is 1, Next Outaged
    Feeder is 2 is %g\n",P12)
printf("Presently Outaged Feeder is 1, Next Outaged
    Feeder is 3 is %g\n",P13)
printf("Presently Outaged Feeder is 2, Next Outaged
    Feeder is 1 is %g\n",P21)
printf("Presently Outaged Feeder is 2, Next Outaged
    Feeder is 2 is %g\n",P22)
printf("Presently Outaged Feeder is 2, Next Outaged
    Feeder is 3 is %g\n",P23)
printf("Presently Outaged Feeder is 3, Next Outaged
    Feeder is 1 is %g\n",P31)
printf("Presently Outaged Feeder is 3, Next Outaged
    Feeder is 2 is %g\n",P32)
printf("Presently Outaged Feeder is 3, Next Outaged
    Feeder is 3 is %g\n",P33)
printf("\nb) Transition Matrix is\n")
disp(P)
printf("\nc) The Transition figure is displayed in
    the result file attached to this code\n")
```

Scilab code Exa 11.9 To Determine the vector of state probabilities at a specific time

```
//To Determine the vector of state probabilities at
   a specific time
//Page 624
clc;
clear;

P=[0.6,0.4;0.3,0.7]; //One Step Transition Matrix

Po=[0.8,0.2]; //Initial State Probability Vector

//Funtion to determine the Vector of State
   Probability
deff('x=VSP(y)','x=(Po*(P^y))')

P1=VSP(1); //Vector of State Probability at Time t1
P4=VSP(4); //Vector of State Probability at Time t4
P8=VSP(8); //Vector of State Probability at Time t8

printf('\na) The Vector of State Probability at time
    t1 is\n')
disp(P1)
printf('\na) The Vector of State Probability at time
    t4 is\n')
disp(P4)
printf('\na) The Vector of State Probability at time
    t8 is\n')
disp(P8)
```