Scilab Textbook Companion for Analog and Digital Communication by S. Sharma[1]

Created by
Banda Tarunkumar
B.E
Electronics Engineering
Sri Jayachamarajendra College of Engineering
College Teacher
T. Shreekanth
Cross-Checked by
Chaitanya Potti

May 26, 2016

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

Book Description

Title: Analog and Digital Communication

Author: S. Sharma

Publisher: S. K. Kataria & Sons, New Delhi

Edition: 3

Year: 2012

ISBN: 81-89757-52-0

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes

Chapter 2

Amplitude Modulation

Scilab code Exa 2.1 frequency range of sidebands

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.1(PAGENO 51)");
//given
L = 50*10^-6//in henry
C = 1*10^-9//in farads
//calculation
F_c = 1/(2*%pi*sqrt(L*C));
//results
printf("\n\nCarrier frequency F_c = %.2f Hz",F_c);
printf("\n\nNow , it is given that the highest modulation frequency is 8KHz ");
printf("\n\nTherefore, the frequency range occupied by the sidebands will range from 8KHz \nabove to 8KHz below the carrier frequency, extending fom 712KHz to 720KHz.");
```

Scilab code Exa 2.2 Bandwidth of modulated signal

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.2(PAGENO 51)");

//given
//v_m = 10*sin(2*%pi*10^3*t)
//by comparing with v_m = V_m*sin(2*%pi*f_c*t) we get
V_m = 10//in volts
f_m = 1*10^3//in hertz
V_c = 20//in volts
f_c = 1*10^4//in hertz

//calculations
m_a = V_m/V_c;//modulation index formula
m_a1 = m_a*100;//percentage modulation index
f_usb = f_c + f_m;//Upper sideband
f_lsb = f_c - f_m;//lower sideband
A = (m_a*V_c)/2//amplitude of upper as well as lower sideband
B = 2*f_m;//bandwidth of the modulation signal

//results
printf("\n\ni.a.Modulation index=%.2f",m_a);
printf("\n\n    b.Percentage modulation index=%.2f percent",m_a1);
printf("\n\nii.a.Upper sidebandfrequency=%f Hz",f_usb);
```

```
printf("\n\n     b.Lower sideband frequency=%f Hz ",
   f_lsb);
printf("\n\niii.Amplitude of Upper sideband and
   Lower sideband =%f V",A);
printf("\n\niv.Bandwidth of th modulation signal=%f
   Hz",B);
```

Scilab code Exa 2.3 total power in amplitude modulated wave

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.3(PAGENO 54)");

//given
m_a = .75;//modulation index
P_c = 400;//carrier power in watts

//calculation
P_t = P_c*(1+(m_a^2/2));//total power

//results
printf("\n\nTotal power in  the amplitude modulated
   wave=%.2f W",P_t);
```

Scilab code Exa 2.4 Carrier Power

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
```

```
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.4(PAGENO 54)");
//given
P_t = 10*10^3;//total power in watts
m_a = .6;//modulation index
//calculation
P_c = (P_t/(1+(m_a^2/2)));// carrier power
//results
printf("\n\nCarrier power=%.2f W",P_c);
```

Scilab code Exa 2.5 antenna current and percentage modulation

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.5(PAGENO 55)");

//given
I_t = 8.93;//total modulated current in ampers
I_c= 8;//carrier or unmodulated current in ampers
//calculation
m_a = sqrt(2*((I_t/I_c)^2 -1));//formula for modulation index
M_a=m_a*100;//percentage modulation
//for
m_a1 = .8;//given modulation index

//calculation
I_t1 = I_c*sqrt(1+(m_a1^2/2));//new antenna current

```

```
//results
printf("\n\ni.a. Modulation index = %.2f",m_a);
printf("\n\n  b.Percentage modulation index = %.2f
    percent",M_a);
printf("\n\nii. Antenna current=%.2f A",I_t1);
```

Scilab code Exa 2.6 carrier current and modulation of signal and modulation index of second signal

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.6(PAGENO 56)");

//given
I_t1 = 10//antenna current in amps
m1 = .3//modulation index
I_t2 = 11//increased antenna current

//calculation
I_c = (I_t1/(1+(m1^2/2))^.5);//formula for carrier
    signal current
m_t = sqrt(2*((I_t2/I_c)^2 -1));//formula for
    modulation index
m2 = sqrt(m_t^2 - m1^2);
m3 = m2*100;//percentage modulation index

//results
printf("\n\ni.Carrier signal current = %.2f A",I_c);
printf("\n\nii.Modulation index of signal = %.2f",
    m_t);
printf("\n\niii.a.Modulation index of second signal
```

```
   = %.2f",m2);
printf("\n\n    b.Percentage modulation index of
   second signal = %.2f percent",m3);
```

Scilab code Exa 2.7 modulating volatage and minimum and maximum voltage

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.7(PAGENO 56)");

//given v_c = 10*sinwt

m = .5//modulation index
//by comparing with v_c = V_c*sinwt
V_c = 10//carrier voltage in volts

//calculation
V_m = m*V_c;//amplitude of modulating index
V_max = V_c + V_m;//maximum voltage
V_min = V_c - V_m;//minimum voltage

//results
printf("\n\n i.Modulating  voltage = %.2f V",V_m);
printf("\n\n ii. Maximum voltage = %.2f V",V_max);
printf("\n\n iii.Minimum voltage = %.2f V",V_min);
```

Scilab code Exa 2.9 percentage modulation index

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.9(PAGENO 57)");

//given
V_max = 4//maximum voltage in volts
V_min = 1//minimum voltage in volts

//calculation
m = (V_max - V_min)/(V_max + V_min) ;//formula for
    modulation index
m1 = m*100//percentage modultion index

//result
printf("\n\n Percentage modulation index = %.2f
    percent",m1);
```

Scilab code Exa 2.10 percentage modulation index of second wave

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.10(PAGENO 57)");

//given
m1 = .4//modulation index
I_t1 = 11//initial antenna current in ampers
I_t2 = 12//final antenna current in ampers
```

```
//calculations
I_c = (I_t1/(1+(m1^2/2))^.5);// formula for carrier
    current in ampers
m_t = sqrt(2*((I_t2/I_c)^2 -1));//total modulation
    index
m2 = sqrt(m_t^2 - m1^2);//modulation index to the
    second wave
m3 = m2*100;//percentage modulation index to the
    second wave

//results
printf("\n\n Carrier current = %.2f A",I_c);
printf("\n\nTotal modulation index = %.2f",m_t);
printf("\n\nPercentage modulation index of second
    wave= %.2f percent",m3);
```

Scilab code Exa 2.11 total radiated power

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.11(PAGENO 58)");

//given
P_c = 10*10^3//carrier power in watts
P_t = 12*10^3//total power in watts
m_2 = .5//modulation index of second wave

//calculations
m_1 = sqrt(2*((P_t/P_c)-1));//modulation index of
    first wave
```

```
m_t = sqrt(m_1^2 +m_2^2);//total modulation index
P_t1 = P_c*(1+(m_t^2/2))//total new transmitted power

//results
printf("\n\nModulation index of first wave = %.4f",m_1);
printf("\n\nTotal modulation index = %.2f",m_t);
printf("\n\ntotal new transmitted power = %.2f W",P_t1);
```

Scilab code Exa 2.12 total radiated power

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.12(PAGENO 60)");

//given
P_t = 10.125*10^3//modulated or total power in watts
P_c = 9*10^3//unmodulated of carrier power
m_2 = .4//modulation index of second wave

//calculations
m_1 = sqrt(2*((P_t/P_c) - 1))//modulation index of first wave
m_a = m_1*100//percentage modulation index of first wave
m_t = sqrt(m_1^2 + m_2^2)//total modulation index
P_t1 = P_c*(1+(m_t^2/2))//total radiated power

//results
```

```
printf("\n\ni.a.Modulation index of first wave = %.4f",m_1);
printf("\n\n  b.Percentage modulation index of first wave = %.2f percent",m_a);
printf("\n\nii.Total radiated power = %.2f W",P_t1);
```

Scilab code Exa 2.13 percentage power saving

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.13(PAGENO 90)");

//given
m1 = 1//modulation index of first signal
m2 = .5//modulation index of second signal
//let
P_c = 1//carrier power in watts

//calculations
P_1= P_c*(1+(m1^2/2))//total power of first signal
P_2 = P_c*(1+(m2^2/2))//total power of second signal
P_a = (P_c*100)/(P_1)//percentage power saving for first signal
P_b = (P_c*100)/(P_2)//percentage power saving for second signal

//results
printf("\n\ni.Percentage power saving for first signal= %f percent",P_a);
printf("\n\nii.Percentage power saving for second signal= %f percent",P_b);
```

Scilab code Exa 2.15 percentage power saving

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.15(PAGENO98 )");

//given
m1 = 1//modulation index of first signal
m2 = .5//modulation index of second signal
//let
P_c = 1//carrier power in watts

//calculations
P_cssb1 = P_c*(1+(m1^2/4))//power in carrier plus power in one sideband for first signal
P_cssb2 = P_c*(1+(m2^2/4))//power in carrier plus power in one sideband for second signal
P_1= P_c*(1+(m1^2/2))//total power of first signal
P_2 = P_c*(1+(m2^2/2))//total power of second signal
P_a = (P_cssb1*100)/(P_1)//percentage power saving for first signal
P_b = (P_cssb2*100)/(P_2)//percentage power saving for second signal

//results
printf("\n\ni.Percentage power saving for first signal= %f percent",P_a);
printf("\n\nii.Percentage power saving for second signal= %f percent",P_b);
```

Scilab code Exa 2.16 power content of carrier and upper and lower sidebands

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.16(PAGENO 109)");

//given
P_ssb  = 10*10^3//power in ssb transmission  in watts
P_t = P_ssb// total power in watts
m_a = .8//modulation index

//calculations
P_c = (P_t/(1+(m_a^2/4)+(m_a^2/4)))//carrier power in watts
P_SB = P_t - P_c//power in sidebands
P_usb =  P_SB/2//power in upper sideband
P_lsb =P_usb//power in upper sideband

//results
printf("\n\ni.Power content of the carrier = %.2f W",P_c);
printf("\n\nii.a.Power content in upper sideband  = %.2f W",P_usb);
printf("\n\n   b.Power content in lower sideband = %.2f W",P_lsb);
```

Scilab code Exa 2.17 percentage modulation index

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.17(PAGENO 109)");

//given from the figure
P_maxpp = 2*80//maximum peak to peak power in watts
P_minpp = 2*20//minimum peak to peak power in watts

//calcualtions
m_a = (P_maxpp - P_minpp)/(P_maxpp + P_minpp)//modultaion index
M = m_a*100//percentage modulation index

//results
printf("\n\ni.Modulation index =%.2f",m_a);
printf("\n\nii.Percentage modulation index = %.2f percent",M);
```

Scilab code Exa 2.18 percentage modulation index

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.18(PAGENO 110)");

//given from the figure
```

```
P_maxpp = 2*50//maximum peak to peak power in watts
P_minpp = 2*15//minimum peak to peak power in watts

//calculations
m_a = (P_maxpp - P_minpp)/(P_maxpp + P_minpp)//modultaion index
M = m_a*100//percentage modulation index

//results
printf("\n\ni.Modulation index =%.4f",m_a);
printf("\n\nii.Percentage modulation index = %.2f percent",M)
```

Scilab code Exa 2.21 carrier power and transmission efficiency and carrier amplitude

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.21(PAGENO 111)");

//given
p_t = 50*10^3//total power
m_a = .707//modulation index
z = 50+0*%i;//load

//calculations

//first case
p_x = .5*(m_a)^2;
p_c = p_t/(1+p_x)//carrier power

```

```
//second case
n = ((p_c*p_x)/(p_c+(p_c*p_x)))*100;//transmission
    efficiency

//third case
 a_c = sqrt(2*z*p_c);//peak carrier amplitude

 //results
 printf("\n\ni. Carrier Power =%.2f W",p_c);
  printf("\n\nii. Percentage Transmission efficiency
      =%.2f percent",n);
   printf("\n\niii. Carrier amplitude  =%.2f V",a_c)
      ;
```

Scilab code Exa 2.29 modulation depth

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.29(PAGENO 118)");

//given
k = 2*10^-3//constants in amperes/square volts
k_1 = 0.2*10^-3//constant in amperes/square volts
printf("\n\nwe know that V_i(t) = cos(w_c*t) + .5*
    cos(w_m*t)");
printf("given i_0 = 10 + k*V_i + k_1*V_i^2 ");
printf("\n\ntherefore i_0 = 10 + 2*10^-3*[cos(w_c*t)
    + .5*cos(w_m*t)] + 2*10^-3*[cos(w_c*t) + .5*cos(
    w_m*t)]");
printf("\n\ni_0 = 2*10^-3*cos(w_c*t) + ((.2*10^-3)
    /.5)*.5*cos(w_c*t)*cos(w_m*t)");
```

```
//Now the modultion depth will be
m = (.2*10^-3)/.5;

//result
printf("\n\nModulation depth = %.8f ",m);
```

Scilab code Exa 2.31 percentage power saving

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.31(PAGENO 119)");

//given
//percentage modulation for first case
Pm_1 = 100
//percentage modulation for second case
Pm_2 = 50
m_1 = 1//modulation index for first case
m_2 = .5//modulation index for second case
P_c = 1//let carrier power be one

//calcualations

//first case
P_t1 = P_c*(1+(m_1^2/2))//total power
P_sb1 = P_c*(m_1^2/4)//power in one side band
P_s1 = ((P_t1-P_sb1)/P_t1)*100//power saving

//second case
P_t2 = P_c*(1+(m_2^2/2))//total power
P_sb2 = P_c*(m_2^2/4)//power in one side band
```

```
P_s2 = ((P_t2-P_sb2)/P_t2)*100//power saving

//results
printf("\n\ni. Power saving with percentage
    modulaion 100 = %.2f percent ",P_s1);
printf("\n\nii. Power saving with percentage
    modulaion 50 = %.2f percent",P_s2);
```

Scilab code Exa 2.32 carrier frequency

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 2
//AMPLITUDE MODULATION
clear all;
clc;
printf("EXAMPLE 2.31(PAGENO 119)");

//given
//the product signal is given by
//v(t) = s(t) * cos(2*%pi*t +phi) = x(t) *cos(2*%pi*
    f_c*t)*cos(2*%pi*f_c*t +phi)
//v(t) = x(t) *(cos(4*%pi*f_c*t +phi) +cos(phi))/2 =
     (x(t)/2)*cos(4*%pi*f_c*t +phi)+(x(t)/2)*cos(phi)
//the low pass filter will reject the first term.
    The maximum allowable value of phase angle(phi)
    can be found as under:
printf("\n\ncos(phi_max) = ((x(t)/2)*cos(phi))/max((
    x(t)/2)*cos(phi))");
phi_max = acosd(.95);
printf("\n\nphi_max = %.2f",phi_max);
printf("\n\nIn order to recover x(t) from v(t) using
     filter method, it is essential that the lowest
    frequency contained in the first term of v(t)
    must be greater than the highest frequency
```

```
    contained in the second term,i.e,")
printf("\n2f_c -10KHz > 10KHz");
printf("\nf_c >10KHz");
printf("\nHence, the minimum value of f_c will be");
printf("\nf_c = 10KHz")
```

Chapter 3

Radio Transmitters

Scilab code Exa 3.1 total radiated power

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 3
//RADIO TRANSMITTER
clear all;
clc;
printf("EXAMPLE 3.1(PAGENO 132)");

//given
Pm = 85//percentage modulation
m = .85//modulation index
P_c = 50*10^3//carrier power in watts

//calculation
P_t = P_c*(1+(m^2/2));//total radiated power

//result
printf("\n\nTotal radiated power = %.2f W",P_t);
```

Scilab code Exa 3.2 carrier frequency and frequency deviation

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 3
//RADIO TRANSMITTER
clear all;
clc;
printf("EXAMPLE 3.2(PAGENO 138)");

//given from the figure
f = 20*10^6//frequency in hertz
//At point 1 from fig
f_c1 = 2*13.5*10^6//carrier frequency
deltaf1 = 2*8.5*10^3//change in frequency

//calculations
f_max1 = f_c1 + deltaf1//maximum frequency at point
    1 in fig
f_min1 = f_c1 - deltaf1//maximum frequency at point
    1 in fig
f_d1 = f_max1 - f_c1;//frequency deviation at point
    1 in fig
f_d2 = f_c1 - f_min1;//frequency deviation at point
    1 in fig
//At point 2 from fig
f_c2 = 3*f_c1//carrier frequency
deltaf2 = 3*deltaf1//change in frequency
f_max2 = f_c2 + deltaf2//maximum frequency at point
    2 in fig
f_min2 = f_c2 - deltaf2//minimum frequency at point
    2 in fig
f_d3 = f_max2 - f_c2;//frequency deviation at point
    2 in fig
f_d4 = f_c2 - f_min2;//frequency deviation at point
    2 in fig
//At point 3 in fig
f_c3 = f_c2 + f;//carrier frequency at point 3 in
```

```
fig
f_max3 = f_max2 + f//maximum frequency at point 3 in fig
f_min3 = f_min2+ f//minimum frequency at point 3 in fig
f_d5 = f_max3 - f_c3;//frequency deviation at the last point
f_d6 = f_c3 - f_min3 ;//frequency deviation at the last point

//results
printf("\n\ni.a Carrier frequency at point 1 in fig =%.2f Hz ",f_c1);
printf("\n\n  b Frequency deviation =%.2f Hz ",f_d1);
printf("\n\n  c Frequency deviation  =%.2f Hz ",f_d2);
printf("\n\nii.a Carrier frequency at point 2 in fig =%.2f Hz ",f_c2);
printf("\n\n    b Frequency deviation =%.2f Hz ",f_d3);
printf("\n\n    c Frequency deviation  =%.2f Hz ",f_d4);
printf("\n\nii.a Carrier frequency at point 3 in fig =%.2fHz ",f_c3);
printf("\n\n    b Frequency deviation =%.2f Hz ",f_d5);
printf("\n\n    c Frequency deviation  =%.2f Hz ",f_d6);
printf("\n\nThus, in mixer, frequency deviation is not altered but only carrier frequency\n is increased")
```

Scilab code Exa 3.3 appropriate multiplexer values and oscillator frequency

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 3
//RADIO TRANSMITTER
clear all;
clc;
printf("EXAMPLE 3.3(PAGENO 138)");
printf("\n\n\tInput frequency deviation is 10Khz,
    while the output frequency deviation \nrequired
    is 60KHz. Thus, a frequency multiplication of
    6*3*2 is required.");
printf("\n\n\tThe frequency multiplication of 6 will
     give the carrier frequency of\n9*6 = 54MHz only.
     Hence we have to use heterodyning. The two
    inputs to the \nmixer are the carrier frequency
    pf 54MHz and oscillator frequency. Assuming \
    nthat at the output of the mixer addition of
    imput frequencies is selected the \nrequired
    oscillator frequency, to have the final carrier
    output frequency of \n106MHz, comes out to be 52
    MHz.[52+54 =106]. ");
```

Chapter 4

Radio Receivers

Scilab code Exa 4.1 image frequency and image rejection ratio

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 4
//Radio Receiver
clear all;
clc;
printf("EXAMPLE 4.1(PAGENO 150)");

//given data
Q = 100//quality factor
f_i = 455*10^3//intermediate frequency

//calculations
//first case
f_s = 1000*10^3//incoming frequecy of first case
f_si = f_s + 2*f_i//image frequency of first case
p = (f_si/f_s) - (f_s/f_si);
alpha = sqrt(1+(Q^2*p^2))//rejection ratio of first
    case
//second case
f_s1 = 25*10^6//incoming frequecy of second case
```

```
f_si1 = f_s1+ 2*f_i//image frequency of second case
p1 = ((f_si1/f_s1) - (f_s1/f_si1));
alpha1 = sqrt(1+(Q^2*p1^2))//rejection ratio of
    second case

//results
printf("\n\n(i)a.Image frequency of first case = %.2
    fHz",f_si);
printf("\n\n   b.Rejection ratio of first case = %.2
    f",alpha);
printf("\n\n (ii)a.Image frequency of second case =
    %.2fHz",f_si1);
printf("\n\n    b.Rejection ratio of second case =
    %.2f",alpha1);
printf("\n\nNote: Their is mistake in textbook  in
    the calculation of rejection ratio")
```

Scilab code Exa 4.2 image frequency and image rejection ratio

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 4
//Radio Receiver
clear all;
clc;
printf("EXAMPLE 4.2(PAGENO 150)");

//given data
Q = 90
f_i = 455*10^3//intermediate frequency

//calculations
//first case
f_s = 950*10^3//incoming frequency of first case
f_si = f_s + 2*f_i//image frequency of first case
```

```
p = (f_si/f_s) - (f_s/f_si);
alpha = sqrt(1+(Q^2*p^2))//rejection ratio of first case
//second case
f_s1 = 10*10^6//incoming frequecy of second case
f_si1 = f_s1+ 2*f_i//image frequency of second case
p1 = ((f_si1/f_s1) - (f_s1/f_si1));
alpha1 = sqrt(1+(Q^2*p1^2))//rejection ratio of second  case

//results
printf("\n\n(i)a.Image frequency of first case = %.2f Hz",f_si);
printf("\n\n   b.Rejection ratio of first case = %.2f",alpha);
printf("\n\n(ii)a.Image frequency of second case = %.2f Hz",f_si1);
printf("\n\n    b.Rejection ratio of second case = %.2f",alpha1);
```

Scilab code Exa 4.3 quality factor and new intermediate frequency

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 4
//Radio Receiver
clear all;
clc;
printf("EXAMPLE 4.3(PAGENO 151)");
//given

a1 = 130.5//rejection ratio
f_s = 10*10^3//incoming frequency
printf("from fig 4.8 from t/b we can write that")

```

```
//calculations
//first case
alpha = 130.5//from problem 4.2 of first case
alpha2 = 15.72//from problem 4.2 of second case
alpha1 = alpha/alpha2//rejection ratio ofgiven RF amplifer
p1 =.174//from problem 4.2 of second case
Q = (sqrt(alpha1^2 - 1)/p1)//quality factor
//second case
p2 = 1.45//from problem 4.2 of second case
f_si =1860*10^3//from problem 4.2 of second case
f_i = 950*10^3//incoming frequency
f_i1 = 10*10^6//good image frequency
f_si1 = (f_si*f_i1)/f_i';//mage frequency
f_i2 = (f_si1 - f_i1)/2//new intermediate frequency

//results
printf("\n\n(i)Quality factor =%.2f ",Q);
printf("\n\n(ii)New intermediate frequency =%.4f Hz",f_i2);
```

Scilab code Exa 4.5 oscillator frequency and image frequency and image rejection ratio

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 4
//Radio Receiver
clear all;
clc;
printf("EXAMPLE 4.5(PAGENO 152)");

//given
IF = 455*10^3//intermediate frequency in hertz
f_s = 900*10^3//signal frequency in hertz
```

```
Q = 80//quality factor

//calculations
f_0 = f_s + IF//local oscillator frequency
f_si = f_s + 2* IF//image frequency
p = (f_si/f_s)-(f_s/f_si)
a = sqrt(1+(Q*p)^2)//image frequency rejectio ratio

//results
printf("\n\n(i)Local oscillator frequency = %.2f Hz"
    ,f_0);
printf("\n\n(ii)Image frequency = %.2f Hz",f_si);
printf("\n\n(iii)Image frequency rejection ratio = %
    .2f",a);
printf("\n\n(iv)Note:Their is mistake in textbook in
     the calculation of image frequency")
```

Scilab code Exa 4.6 image frequency and image rejection ratio and intermediate frequency

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 4
//Radio Receiver
clear all;
clc;
printf("EXAMPLE 4.6(PAGENO 153)");
//given
Q = 125//quality factor

//calculations
//first case
IF1 = 465*10^3//intermediate frequency
f_s1 = 1*10^6//incoming frequency for first case in
    hertz
```

```
f_s2 = 30*10^6//second incoming frequency for first case in hertz
f_si1 = f_s1 + 2*IF1//image frequency for incoming frequency 1MHz for first case
f_si2 = f_s2 + 2*IF1//image frequency for incoming frequency 30MHz for first case
p1 = (f_si1/f_s1)-(f_s1/f_si1);
p2 = (f_si2/f_s2)-(f_s2/f_si2);
alpha1 = sqrt(1+(Q*p1)^2);//rejection ratio at 1MHz incoming frequency
alpha2 = sqrt(1+(Q*p2)^2);//rejection ratio at 30MHz incoming frequency
//second case
f_s3 = 1*10^6//incoming frequency for second case in hertz
f_si3 = (f_si1*f_s2)/f_s3//image frequency
IF2 = (f_si3-f_s2)/2//intermediate frequency

//results
printf("\n\n(i)a.Image frequency for 1MHz incoming frequency = %.2f Hz",f_si1);
printf("\n\n    b.Rejection ratio for 1MHz incoming frequency = %.2f",alpha1);
printf("\n\n    c.Image frequency for 30MHz incoming frequency = %.2f Hz",f_si2);
printf("\n\n    d.Rejection ratio for 30MHz incoming frequency = %.2f",alpha2);
printf("\n\n(ii)intermediate frequency for second case = %.2f Hz",IF2);
```

Chapter 5

Angle Modulation

Scilab code Exa 5.1 carrier frequency and modulating frequency and modulation index and maximum deviation and power dissipated

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.1(PAGENO 198)");

//given
t = [0:.1:10];//time period
R = 10//resistance in ohms
printf("\n\nv(t) = 12*cos(6*10^8*t + 5*sin(1250*t));");
printf("\n\nv(t) = A*cos(w_c*t + m_f*sin(w_m*t))");
//by comparing with standard
A = 12//amplitude voltage in volts
w_c= 6*10^8//angular carrier frequency in rad/sec
w_m = 1250//angular modulating frequency in rad/sec
m_f = 5//modulation index

```

```
//calculations
f_c =  w_c/(2*%pi)//carrier frequency
f_m = w_m/(2*%pi)//modulating frequency
deltaf = m_f*f_m//maximum deviation
V_rms = (A/sqrt(2))^2//rms volatage
P = V_rms/R//power dissipatted

//results
printf("\n\n i. Carrier frequency = %.2f Hz",f_c);
printf("\n\nii. Modulation frequency = %.2f Hz",f_m);
printf("\n\niii. Modulation index = %.2f",m_f);
printf("\n\niv.Maximum frequency deviation = %.2f Hz", deltaf);
printf("\n\nv.Power dissipated = %.2f W ",P);
```

Scilab code Exa 5.2 carrier frequency and highest and lowest frequency and modulation index

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.2(PAGENO 199)");

//given
f_c = 107.6*10^6//carrier frequency
f_m = 7*10^3//modulationg frequency
deltaf = 50*10^3//frequency deviation

//calculations
cs = 2*deltaf//carrier swing
f_H = f_c + deltaf//highest frequency
```

```
f_L = f_c - deltaf//lowest frequency
m_f = deltaf/f_m//modulating index

//results
printf("\n\ni.Carrier frequency = %.2f Hz",cs);
printf("\n\nii.a.Highest frequency attained by the modulating signal = %.2f Hz",f_H);
printf("\n\n   b.Lowest frequency attained by the modulating signal = %.2f Hz",f_L);
printf("\n\niii.modulating index of the FM wave = %.3f",m_f);
```

Scilab code Exa 5.3 frequency deviation and carrier swing and lowest frequency of modulated wave

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.3(PAGENO 199)");
//given
f_c = 105*10^6//carrier frequency
f_H = 105.007*10^6//highest frequency or upper frequency
//calculations
deltaf = f_H - f_c//frequency deviation
cs = 2*deltaf//carrier swing
f_L = f_c - deltaf//lower frequency
//results
printf("\n\ni.Frequency deviation = %.4f Hz",deltaf);
printf("\n\nii.Carrier swing = %.2f Hz",cs);
printf("\n\niii.Lower frequency reached by the
```

```
modulated wave = %.3f Hz",f_L);
```

Scilab code Exa 5.4 modulation index

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.4(PAGENO 200)");

//given
cs = 100*10^3//carrier swing
f_m = 8*10^3//modulating frequency

//calculations
deltaf = cs/2//frequency deviation
m_f = deltaf/f_m//modulation index
//results
printf("\n\n     Modulation index = %.2f",m_f);
```

Scilab code Exa 5.5 percentage modulation index

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.5(PAGENO 200)");

//given
```

```
deltaf = 20*10^3//frequency deviation
deltaf_actual = deltaf//since deltaf_actual equals
    to deltaf
deltaf_max1 = 75*10^3//maximum frequency deviation
    deltaf_max permittedfor the first case is 75KHz
deltaf_max2 = 25*10^3//maximum frequency deviation
    deltaf_max permitted for the second case is 25KHz

//calculations
M1 = (deltaf_actual/deltaf_max1)*100//persentage
    modulation index for first case
M2 = (deltaf_actual/deltaf_max2)*100//persentage
    modulation index for second case

//results
printf("\n\ni.Percentage modulation index for first
    case = %.2f percent",M1);
printf("\n\nii.Percentage modulation index for
    second case = %.2f percent",M2);
```

Scilab code Exa 5.6 bandwidth of FM transmission

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.6(PAGENO 210)");

//given
deltaf = 75*10^3//frequency deviation
f_m = 15*10^3//modulating frequency

//calculation
```

```
BW = 2*(deltaf+f_m)//bandwidth

//result
printf("\n\nBandwidth of a commercial FM
    transmission = %.2f Hz",BW);
```

Scilab code Exa 5.7 bandwidth of narrowband FM signal

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.7(PAGENO 210)");

//given
f_m = 4*10^3//modulation frequency
f_c = 125*10^3//carrier frequency

//claculation
BW = 2*f_m//bandwidth

//result
printf("\n\n    Bandwidth of a narrowband FM signal =
     %.2f Hz",BW);
```

Scilab code Exa 5.8 bandwidth of a signal when modulating amplitude is doubled

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
```

```
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.8(PAGENO 210)");

//given
deltaf1 = 75*10^3//frequency deviation
f_m = 8*10^3//modulation frequency
deltaf2 = 2*deltaf1//if modulation signal amplitude is doubled, the frequency deviation becomes double

//calculation
BW1 = 2*(deltaf1 +f_m)//bandwidth
BW2 =  2*(deltaf2 +f_m)//new bandwidth

//result
printf("\n\nBandwidth of a signal when modulating signal amplitude is doubled = %.2fHz",BW2);
```

Scilab code Exa 5.9 modulation index and bandwidth of FM signal

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.9(PAGENO 211)");

//given
f_m = 5*10^3//modulating frequency
f_c = 50*10^6//carrier frequency
deltaf = 20*10^3//frequency deviation

```

```
//calculations
m_f = deltaf/f_m//modulation index
BW = deltaf*3.8//referring to the Schwartz bandwidth
    curve

//results
printf("\n\ni.Modulation index = %.2f",m_f);
printf("\n\nii.Bandwidth of the FM signal = %.2f Hz"
   ,BW);
```

Scilab code Exa 5.10 modulating frequency

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.10(PAGENO 212)");

//given
BW = 50*10^3//bandwidth
deltaf = 10*10^3//frequency deviation

//calculation
x = BW/deltaf//variable
m_f = 2//by referring to the Schwartz bandwidth
   curve with 'x'
f_m = deltaf/m_f//modulating frequency

//results
printf("\n\n  Modulating frequency = %.2f Hz",f_m);
```

Scilab code Exa 5.11 modulation index and bandwidth of FM and PM signal

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.11(PAGENO 217)");
//given
//x(t) = 5*cos(2*%pi*15*10^3*t)
V_m = 5//amplitude of voltage
f_m = 15*10^3//modulation frequency
k_f = 15*10^3//frequency sensitivity
k_p = 15*10^3//phase sensitivity

//calculations
//first case
//for FM system
delta_f1 = k_f * V_m;//frequency deviation for FM system
m_f1 = delta_f1/f_m; //modulation index in FM system
BW1 = 2*(delta_f1+f_m);//bandwidth for FM system
//for PM system
delta_f2 = k_f * V_m*f_m;//frequency deviation for PM system
BW2 = 2*(delta_f2 + f_m);//bandwidth for PM system
m_p1 = k_p * V_m//modulation index in PM system

//second case
f_m1 = 5*10^3//modulating frequency for second case
//for FM system
delta_f3 = k_p * V_m;//frequency deviation for FM system
m_f2 = delta_f3/f_m1; //modulation index in FM system
BW3 = 2*(delta_f3+f_m1);//bandwidth for FM system
```

```
//for PM system
delta_f4 = k_p * V_m*f_m1;//frequency deviation for
   PM system
BW4 = 2*(delta_f4 + f_m1);//bandwidth for PM system
m_p2 = k_p * V_m//modulation index in PM system

//results
printf("\n\ni.a.Modulation index of FM system for
    first case = %.2f",m_f1);
printf("\n\n  b.Bandwidth of FM system for first
    case = %.2f Hz",BW1);
printf("\n\nii.a.Modulation index of PM system for
    first case = %.2f",m_p1);
printf("\n\n  b.Bandwidth of PM system for first
    case = %.2f Hz",BW2);
printf("\n\niii.a.Modulation index of FM system for
    second case = %.2f",m_f2);
printf("\n\n  b.Bandwidth of FM system for second
    case = %.2f Hz",BW3);
printf("\n\niv.a.Modulation index of PM system for
    second case = %.2f",m_p2);
printf("\n\n  b.Bandwidth of PM system for second
    case = %.2f Hz",BW4);
```

Scilab code Exa 5.12 carrier frequency and modulating frequency and modulation index and maximum deviation and power dissipated

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.12(PAGENO 219)");

```

```
//given
//given that v = 10*sin((5 * 10^8 *t) + 4*sin(1250*t
    ))
//by comparing with standard eqn i.e v = V_c*sin((
    w_c * t) + m_f*sin(w_m*t)) we get
w_c = 5*10^8//angular carrier frequency
w_m = 1250//angular modulating frequency
m_f = 4//modulating index
V_c = 10//carrier voltage in volts
R = 5//resistance in ohms

//calculations
f_c = w_c/(2*%pi)//carrier frequency
f_m = w_m/(2*%pi)//modulating frequency
deltaf = m_f * f_m//maximum deviation
V_rms = (V_c/sqrt(2))^2//RMS value of FM wave
P = V_rms/R//power dissipated

//results
printf("\n\ni.a.Carrier frequency = %.2f Hz",f_c);
printf("\n\n  b.Modulating frequency = %.2f Hz",f_m)
    ;
printf("\n\nii.a.Modulation index = %.2f",m_f);
printf("\n\n   b.Maximum deviation = %.2f Hz",deltaf
    );
printf("\n\niii.Power dissipated in 5 ohms
    resistance = %.2f W",P);
```

Scilab code Exa 5.13 frequency deviation for given modulating volatge

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
```

```
clc;
printf("EXAMPLE 5.13(PAGENO 220)");

//given
f_m = 1*10^3//modulating frequency
V_m = 2//modulating voltage in volts
deltaf = 6*10^3//frequency deviation
V_m1 = 4//increased modulation voltage for first
    case
V_m2 = 8//increased modulation voltage for second
    case

//calculations
k_f = deltaf/V_m//proportion constant
//first case
deltaf1 = k_f*V_m1//frequency deviation for first
    case
//second case
deltaf2 = k_f*V_m2//frequency deviation for second
    case

//results
printf("\n\ni.Frequency deviation for modulating
    voltage 4V = %.2f Hz",deltaf1);
printf("\n\nii.Frequency deviation for modulating
    voltage 8V = %.2f Hz",deltaf2);
```

Scilab code Exa 5.14 modulation index

```

//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
```

```
clc;
printf("EXAMPLE 5.14(PAGENO 220)");

//given
deltaf = 6*10^3//frequency deviation from the question of EXAMPLE 5.13(PAGENO 220)
f_m = 1*10^3//modulating frequency from the question of EXAMPLE 5.13(PAGENO 220)
deltaf1 = 12*10^3//frequency deviation from the EXAMPLE 5.13(PAGENO 220) of first case
deltaf2 = 24*10^3//frequency deviation from the EXAMPLE 5.13(PAGENO 220) of second case
f_m1 = f_m//modulating frequency from the EXAMPLE 5.13(PAGENO 220) of first case
f_m2 = 500//modulating frequency from the EXAMPLE 5.13(PAGENO 220) ofsecond case

//calculation
m_f = deltaf/f_m//modulation index for the initial conditions given in the problem 5.13
m_f1 = deltaf1/f_m1//modulation index for the first case
m_f2 = deltaf2/f_m2//modulation index for the second case

//results
printf("\n\na.Modulation index for initial conditions given in the problem 5.13 = %.2f ",m_f);
printf("\n\nb.Modulation index for the first case = %.2f",m_f1);
printf("\n\nc.Modulation index for the second case = %.2f",m_f2);
```

Scilab code Exa 5.15 bandwidth of FM and DSB FC signal

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.15(PAGENO 220)");

//given
deltaf = 10*10^3//frequency deviation
f_m = 1*10^3//modulating frequency

//calculations
BW = 2*(deltaf + f_m)//bandwidth of FM signal
BW_DSB = 2*f_m//bandwidth of DSB FC(AM)

//results
printf("\n\ni.Bandwidth of FM signal = %.2f Hz",BW);
printf("\n\nii.Bandwidth of DSB FC(AM) signal = %.2f
    Hz",BW_DSB);
```

Scilab code Exa 5.16 repersentation of FM and PM signal or wave

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.16(PAGENO 221)");

//given
//first case
f_c1 = 20*10^6//carrier frequency
f_m1 = 400//modulation frequency
```

```
V_c = 5//carrier voltage in volts
deltaf = 10*10^3//frequency deviation
//second case
f_m2 = 2*10^3//modulation frequency

//calculations
w_c1 = 2 *%pi *f_c1//angular carrier freqency
w_m1 = 2 *%pi *f_m1//angular carrier freqency
m_f1 = deltaf/f_m1//modulation index for first case
m_f2 = deltaf/f_m2//modulation index for second case

//results
//standard format of fm and pm equations are
//s(t) = V_c8sin(w_c*t + m_f*sin(w_m*t))
printf("\n\n(i)FM wave:s(t) = 5*sin(1.25*10^8*t + 25*sin(2513*t)");
printf("\n\n(ii)PM wave:s(t) = 5*sin(1.25*10^8*t + 25*sin(2513*t)");
```

Scilab code Exa 5.17 carrier amplitude and amplitude of sidebands and minimum and maximum frequency of sidebands

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.17(PAGENO 221)");

//given
V_m = 5//modulating voltage
f_m = 20*10^3//modulating frequency
V_c = 10//carrier voltage
f_c = 100*10^6//carrier frequency
```

```
delta_f = 2*10^3//frequeny deviation in hertz per volt

//calculations
m_a = delta_f/f_m//modulation index
printf("\n\nfor m_a = .5 the approximate values of j coefficients are");
printf("\n\nJ_0 = .94    J_1 = .24    J_2 =.03");
J_0 = .94
 J_1 = .24
 J_2 =.03
A_c = V_c*J_0;//carrier amplitude
A_1 = V_c*J_1;//amplitude of first pair of sideband
A_2 = V_c*J_2;//amplitude of second pair of sideband
f_1 = f_c + f_m//maximum frequency of first pair of sideband
f_1a = f_c - f_m//minimum frequency of first pair of sideband
f_2 = f_c + (2*f_m)//maximum frequency of second pair of sideband
f_2a = f_c - (2*f_m)//minimum frequency of second pair of sideband

 //results
printf("\n\ni.Carrier amplitude = %f V",A_c);
printf("\n\nii.Amplitude of first pair of sideband = %f V",A_1);
printf("\n\niii.Amplitude of second pair of sideband = %f V",A_2);
printf("\n\niV.a.Maximum frequency of first pair of sideband  = %f Hz",f_1);
printf("\n\n    .b.Minimum frequency of first pair of sideband = %f Hz",f_1a);
printf("\n\nV.a.Maximum frequency of second pair of sideband = %f Hz",f_2);
printf("\n\n   .b.Minimum frequency of second pair of sideband = %f Hz",f_2a);
```

Scilab code Exa 5.18 maximum frequency devaiation and modulation index

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.18(PAGENO 222)");

//given
f_m1 = 400//modulating frequency for first case
V_m1 = 2.4//modulating voltage for first case
f_m2 = 250//modulating frequency for second case
V_m2 = 3.2//modulating voltage for second case
m_f1 = 60//modulation index for first case

//calculations
delta_f1 = m_f1*f_m1//maximum frequency deviation
    for first case
k = delta_f1/V_m1//constant
delta_f2 = k*V_m2//frequency deviation for second
    case
m_f2 = delta_f2/f_m2//modulation index for second
    case

//results
printf("\n\ni.Maximum frequency deviation for first
    case = %.2f Hz",delta_f1);
printf("\n\nii.Modulation index for second case = %
    .2f ",m_f2);
```

Scilab code Exa 5.19 modulation index and bandwidth of a signal

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.19(PAGENO 222)");

//given
f_m1 = 1*10^3//modulating frequency for first case
f_m2 = 500//modulating frequency for second case
V_m1 = 2//modulating voltage for first case
V_m2 = 8//modulating volatge for second case
delta_f1 = 4*10^3//frequency deviation for first case

//calculations
 k = delta_f1/V_m1//constant
 delta_f2 = k*V_m2//frequency deviation for second case
 m_f1 = delta_f1/f_m1//modulation index for first case
 m_f2 = delta_f2/f_m2//modulation index for second case
 BW1 = 2*(delta_f1 + f_m1)//bandwidth for first case
 BW2 = 2*(delta_f2 + f_m2)//bandwidth for second case

 //results
printf("\n\ni.a.Modulation index for first case = %.2f",m_f1);
printf("\n\n  b.Bandwidth for first case = %.2f Hz",
```

```
BW1);
printf("\n\nii.a.Modulation index for second case = %.2f",m_f2);
printf("\n\n b .Bandwidth for second case = %.2f Hz",BW2);
printf("\n\nNote: Their is error in textbook in the calculation of second case bandwidth ");
```

Scilab code Exa 5.20 capacitive reactance

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;

printf("EXAMPLE 5.20(PAGENO 232)");
//given
g_m = 10*10^-3//transconductance
n = 8
f= 5*10^6//operating frequency

//calculation
X_Ceq = n/g_m//capacitive reactance

//result
printf("\n\nCapacitive reactance = %.2f ohms",X_Ceq)
```

Scilab code Exa 5.21 carrier frequency and frequency deviation and modulation index and minimum and maximum frequency at a given point

```
//ANALOG AND DIGITAL COMMUNICATION
```

```
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.21(PAGENO 233)");

//given data from block diagram
f_c = 10*10^6//carrier frequency
delta_f = 10*10^3//frequency deviation
m_f = 5//modulation index

//calculations
//first stage
f_cA = 3 * f_c//carrier frequency at point A
delta_fA = 3 * delta_f//frequency deviation at point A
m_fA = 3 * m_f//modulation index at point A
f_maxA = f_cA + delta_fA//maximum frequency at point A
f_minA = f_cA - delta_fA//minimum frequency at point A
//second stage
f_cB = f_cA + f_c//carrier frequency at point B
f_maxB = f_maxA  +f_c//maximum frequency at point B
f_minB = f_minA + f_c//minimum frequency at point B
delta_fB = f_maxB - f_cB//frequency deviation at point B
//their will no change in modulation index

//results
printf("\n\ni.a.Carrier frequency at point A = %.2f Hz",f_cA);
printf("\n\n b.Frequency deviation at point A = %.2f Hz",delta_fA);
printf("\n\n c.Modulation index at point A = %.2f ",m_fA);
printf("\n\n d.Maximum frequency at point A = %.2f
```

```
Hz",f_maxA);
printf("\n\n e.Minimum frequency at point A = %.2f Hz",f_minA);
printf("\n\nii.a.Carrier frequency at point B = %.2f Hz",f_cB);
printf("\n\n b.Frequency deviation at point B = %.2f Hz",delta_fB);
printf("\n\n c.Modulation index at point B = %.2f ",m_fA);
printf("\n\n d.Maximum frequency at point B = %.2f Hz",f_maxB);
printf("\n\n e.Minimum frequency at point B = %.2f Hz",f_minB);
```

Scilab code Exa 5.22 modulation index for minimum and maximum modulating frequency of FM signal

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.22(PAGENO 251)");
//given
//first case
//The maximum deviation in commerical FM is given as
delta_f1 = 75*10^3//frequency deviation in commerical FM
f_m1 = 30//maximum modulating frequency
f_m2 = 15*10^3//minimum modulating frequency
//second case
 delta_f2 = 10*10^3//frequency deviation for narrowband FM
 f_m3 = 100//maximum modulating frequency
```

```
f_m4 = 3*10^3//minimum modulating frequency

//calculations
//first case
m_f1 = delta_f1/f_m1//modulation index for maximum modulating frequency
m_f2 = delta_f1/f_m2//modulation index for minimum modulating frequency
//second case
m_f3 = delta_f2/f_m3//modulation index for maximum modulating frequency
m_f4 = delta_f2/f_m4//modulation index for minimum modulating frequency

//results
printf("\n\n i.a.modulation index for maximum modulating frequency of commercial FM = %.2f",m_f1)
printf("\n\n  b.modulation index for minimum modulating frequency of commercial FM = %.2f",m_f2)
printf("\n\nii.a.modulation index for maximum modulating frequency of narrowband FM = %.2f",m_f3)
printf("\n\n   b.modulation index for minimum modulating frequency of commercial FM = %.2f",m_f4)
```

Scilab code Exa 5.23 modulation index and equation for carrier waveform

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
```

```
clc;
printf("EXAMPLE 5.23(PAGENO 251)");

//given
disp('modulated carrier waveform is given by s(t) = A*sin((2*%pi*f_c*t)+m_f*sin(2*%pi*f_m*t))');
f_c = 100*10^6//carrier frequency in hertz
delta_f = 75*10^3//frequency deviation in hertz
f_m = 2*10^3//modulating frequency
A = 5//peak voltage of carrier wave

//calculation
m_f = delta_f/f_m;//modulation index

//result
printf("\n      Modulation index =%.2f",m_f);
disp("Equation for modulated carrier waveform s(t) = 5*sin((2*%pi*100*10^6*t)+37.5*sin(2*%pi*2*10^3*t))");
```

Scilab code Exa 5.24 frequency sensitivity and modulation index

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.24(PAGENO 252)");

//given
//we know that s(t) = A*cos((2*%pi*f_c*t) + m_f*sin(2*%pi*f_m*t ))
f_c = 1*10^6//modulation frequency
A = 3//carrier amplitude in volts
```

```
//first case
A_m = 1//modulating amplitude in volts for first
    case
delta_f = 1*10^3//frequency deviation
f_m1 = 1//modulating frequencyof first case
//second case
f_m2 = 2*10^3//modulating frequency for second case
A_m2 = 5//modulating amplitude for second case

//calculations
k_f  = delta_f/f_m1//frequency sensitivity in hertz
    per volt
m_f = (delta_f*A_m2)/f_m2//modulating frequency
//desired FM signal can be expressed by s(t) = A*cos
    ((2*%pi*f_c*t) + m_f*sin(2*%pi*f_m*t ))
//results
//standard FM signal expression is as follows
//s(t) = A*cos(2*%pi*f_c*t + m_f * sin(2*%pi*f_m*t))
printf("\n\nFrequency sensitivity k_f = %.2f",k_f);
printf("\n\nModulation index m_f =%.2f ",m_f);

disp("s(t)=3*cos(2*%pi*10^6*t + 2.5*sin(2*%pi
    *2*10^3*t)");
```

Scilab code Exa 5.29 bandwidth of FM signal using carsons rule

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.29(PAGENO 256)");

//given
```

```
delta_f = 75*10^3//frequency deviation
f_m = 15*10^3//modulating frequency

//calculations
D = delta_f/f_m//deviation ratio
BW1 = 2*delta_f*(1+(1/D))//bandwidth of FM signal
//using universal curve, replacing m_f by D,we get
BW2 = 3.2*delta_f//for D = 5=3.2*75*10^3
BW = (BW2-BW1)*100/BW2//percentage of under
    estimation of bandwidth by using carson's rule

//results
printf("\n\ni.Bandwidth of FM signal = %.2f Hz",BW1)
    ;
printf("\n\nii.Bandwidth obtained by replacing m_f
    by D = %.2f Hz",BW2);
printf("\n\niii.Percentage of under estimation of
    bandwidth by using Carson rule = %.2f percent",BW
    );
disp("It means that cason s rule under estimates the
     band-width by 25% as compared with the resulat
    obtained from the universal curve.");
```

Scilab code Exa 5.30 fracation of signal power included in the frequency band

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.30(PAGENO 257)");

//given
```

```
m_f1 = 1//modualtion index for first case
m_f2 = 10//modualtion index for second  case
//let
f_m = 1*10^3//modulating frequency

//calulations
//the bandwidth for FM signal can be calculated on
   the basis of 98% power requirement given by
   Carson's rule
BW1 = 2*(m_f1+1)*f_m//bandwidth for first case
B1 = (2*m_f1 +1)*f_m//frequency band first case
BW2 = 2*(m_f2+1)*f_m//bandwidth for second  case
B2 = (2*m_f2 +1)*f_m//frequency band second  case
P1 = (B1/BW1)*(98)//fraction of signal power that is
    included in freuency band for 1st case
P2 = (B2/BW2)*(98)//fraction of signal power that is
    included in freuency band for 2nd case

//results
printf("\n\ni.Fraction of signal power that is
   included in freuency band for 1st case =%.2f
   percent",P1);
printf("\n\nii.Fraction of signal power that is
   included in freuency band for 2nd case =%.2f
   percent",P2);
printf("\n\nNote: Their is mistake in calculation of
    fraction of power of second case in text book")
```

Scilab code Exa 5.31 bandwidth of FM signal

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
```

```
clc;
printf("EXAMPLE 5.31(PAGENO 257)");

//given
f_m1 = 2*10^3//modulating frequency for first case
delta_f1= 5*10^3//frequency deviation for first case
f_m2 = 1*10^3//modulating frequency for second case
delta_f2 = 3*5*10^3//modulating frequency for second
    case

//calculations
BW1 = 2*(delta_f1 + f_m1)//bandwidth of the FM
   signal for first case
BW2 = 2*(delta_f2 + f_m2)//bandwidth of the FM
   signal for secpnd case

//results
printf("\n\ni.Bandwidth of the FM signal for first
   case = %.2f Hz",BW1)
printf("\n\nii.Bandwidth of the FM signal for second
    case = %.2f Hz",BW2)
```

Scilab code Exa 5.32 Carrier Power and power in each sideband

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.32(PAGENO 258)");

//given
m_f = .2//modulation index
P = 10*10^3//power of FM transmitter
```

```
J_0m_f = 0.99//bessel function
J_1m_f =0.099

//calculations
P_c = (J_0m_f)^2 * P//carrier power
P_s1 = (J_1m_f)^2 * P//power in each side frequency
P_s2 = P_s1

//results
printf("\n\ni.Carrier power = %.2f W",P_c);
printf("\n\nii.power in each side band = %.2f W",
    P_s1);
```

Scilab code Exa 5.33 carrier swing and modulation index and highest and lowest frequency attained by FM signal

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.33(PAGENO 258)");

//given
f_m = 7*10^3//modulating frequency
delta_f = 50*10^3//frequency deviation
f_c = 107.6*10^6//carrier frequency

//calculaitons
CS = 2*delta_f//carrier swing
m_f = delta_f/f_m//modulation index
f_h = f_c + delta_f//upper or highest frequency
f_l = f_c - delta_f//lower of lowest frequency

```

```
//results
printf("\n\ni.a.Carrier swing = %.2f Hz",CS);
printf("\n\n  b.Modulation index = %.4f ",m_f);
printf("\n\nii.a.Highest frequency attained by the FM signal = %.2f Hz",f_h);
printf("\n\n   b.Lowest frequency attained by the FM signal = %.2f Hz",f_l);
```

Scilab code Exa 5.34 frequency deviation and carrier frequency and lower frequency reached by modulated FM wave

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.34(PAGENO 259)");

//given
f_c = 100*10^6//carrier frequency
f_u = 100.007*10^6//upper frequency

//calculations
delta_f = f_u - f_c//frequency deviation
CS = 2*delta_f//carrier swing
f_l = f_c - delta_f//lower frequency reached by the modulated FM wave

//results
printf("\n\ni.Frequency deviation = %.2f Hz",delta_f);
printf("\n\nii.Carrier frequency = %.2f Hz",CS);
printf("\n\niii.Lower frequency reached by the modulated FM wave = %.2f Hz",f_l);
```

Scilab code Exa 5.35 percentage modulation index

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.35(PAGENO 259)");

//given
CS = 125*10^3//carrier swing

//calculations
delta_f = CS/2//frequency deviation
//since ,maximum frequency deviation for the FM
   broadcast band is 75 KHz, therefore
f_m = 75*10^3//modulating frequency
m_f = delta_f*100/f_m//modulation index

//result
printf("\n\nPercentage modulation index = %.2f
   percent",m_f);
```

Scilab code Exa 5.36 frequency deviation and modulation index

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
```

```
clc;
printf("EXAMPLE 5.36(PAGENO 259)");

//given
//first case
f_m1 = 500//modulating frequency
delta_f1 = 6.4*10^3//frequency deviation
V_m1 = 3.2//modulating amplitude
//second case
V_m2 = 8.4//modulating amplitude
 //third case
V_m3 = 20//modulating amplitude
f_m3 = 200//modulating frequency

//calculations
k_f = delta_f1/V_m1//frequency sensitivity
delta_f2 = k_f*V_m2//frequency deviation for second case
delta_f3 = k_f*V_m3//frequency deviation for third case
m_1 = delta_f1/f_m1//modulation index for first case
m_2 = delta_f2/f_m1//modulation index for second case
m_3 = delta_f3/f_m3//modulation index for third case

//results
printf("\n\ni.a.Frequency deviation for first case = %.2f Hz",delta_f1);
printf("\n\n b.Modulation index for first case = %.2f ",m_1);
printf("\n\nii.a.Frequency deviation for second case = %.2f Hz",delta_f2);
printf("\n\n  b.Modulation index for second case = %.2f ",m_2);
printf("\n\niii.a.Frequency deviation for third case = %.2f Hz",delta_f3);
printf("\n\n   b.Modulation index for third case = %.2f ",m_3);
```

Scilab code Exa 5.37 carrier frequency and modulating frequency and modulation index and maximum frequency deviation and power dissipated in FM wave

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.37(PAGENO 260)");

//given
// s(t) = 20*sin(6*10^8*t + 7*sin(1250*t))
//comparing with standard eqn s(t) = A*sin(w_c*t + m_f*sin(w_m*t))
//we get
w_c = 6*10^8//carrier angular frequency in rad/sec
w_m = 1250//modulating angular frequency in rad/sec
m_f = 7//modualation index
A = 20//amplitude of modulated wave
R = 100//resistance

//calculations
f_c = w_c/(2*%pi)//carrier frequency in hertz
f_m = w_m/(2*%pi)//modulating frequency in hertz
delta_f = m_f*f_m//frequency deviation
P = (A/sqrt(2))^2/R//power dissipated

//results
printf("\n\ni.Carrier frequency = %.2f Hz ",f_c);
printf("\n\nii.Modulating frequency = %.2f Hz ",f_m);
printf("\n\niii.Modulation index = %.2f ",m_f);
```

```
printf("\n\niv.Frequency deviation = %.2f Hz",delta_f);
printf("\n\nv.Power dissipated by FM wave = %.2f W",P);
```

Scilab code Exa 5.39 maximum frequency deviation

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.39(PAGENO 261)");

//given
//x_c(t) = 10*cos[(10^8*%pi*t) + 5*sin(2*%pi*10^3)t]
//by comparing the given x_c(t) with standard FM wave equation
t=[1:1:10];
w_c = 10^8//carreier frequency
phi_t = 5*sin(2*%pi*10^3*t);
phi_1t = 5*2*%pi*10^3*cos(2*%pi*10^3*t)
//Therefore, the maximum phase deviation will be
phi_tmax = 5//radians

//calculation
delta_f = (5*10^3*2*%pi)/(2*%pi);//maximum frequency deviation

//results
printf("\n\nMaximum frequency deviation is %.2f Hz",delta_f);
```

Scilab code Exa 5.43 bandwidth of angle modulated signal

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.43(PAGENO 263)");

//given
//x_c(t) = 10*cos(2*%pi*10^8*t + 200*cos (2*%pi
    *10^3*t))
//instantaneous frequecy w_i = 2*%pi*10^8 - 4*%pi
    *10^6*sin(2*%pi*10^3)
delta_w = 4*%pi*10^5//angular frequency deviation
w_m = 2*%pi*10^3//angulat modulating frequency

//calculations
beeta = delta_w/w_m;
W_B1 = 2*(beeta + 1)*w_m;//angular bandwidth
//since beeta >>1,therefore
W_B1 = 2*delta_w//angular bandwidth
//W_B==W_B1
f_B = W_B1/(2*%pi)//bandwidth in Hz

//result
printf("\n\nBandwidth = %.2f Hz",f_B);
```

Scilab code Exa 5.44 modulation index and bandwidth of FM signal

```
//ANALOG AND DIGITAL COMMUNICATION
```

```
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.44(PAGENO 263)");

//given
f_c  = 20*10^6//carrier frequency
delta_f = 100*10^3//frequency deviation
f_m1 = 1*10^3//modulation index for first case
f_m2 = 100*10^3//modulation index for second case
f_m3 = 500*10^3//modulation index for third case

//calculations
beeta1 = delta_f/f_m1
beeta2 = delta_f/f_m2
beeta3 = delta_f/f_m3
m_f1 = delta_f/f_m1//modulation index for first case
m_f2 = delta_f/f_m2//modulation index for second
    case
m_f3 = delta_f/f_m3//modulation index for third case
f_B1 = 2*delta_f//bandwidth for first case since it
    is a WBFM signal
f_B2 = 2*(beeta2 + 1)*f_m2//bandwidth for second
    case
f_B3 = 2*f_m3//bandwidth for third case since it is
    a NBFM signal

//results
printf("\n\ni.a.Modulation index for first case = %
    .2f ",m_f1);
printf("\n\n  b.Bandwidth for first case = %.2f Hz",
    f_B1);
printf("\n\nii.a.Modulation index for second case =
    %.2f ",m_f2);
printf("\n\n  b.Bandwidth for second case = %.2f Hz"
    ,f_B2);
```

```
printf("\n\niIi.a.Modulation index for third case = %.2f ",m_f3);
printf("\n\n  b.Bandwidth for third case = %.2f Hz",f_B3);
printf("\n\nNote:Their is error in first case modulating frequency in text book")
```

Scilab code Exa 5.45 bandwidth of angle modulated signal

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.45(PAGENO 263)");

//given
//x_c(t)  = 10*cos(w_c*t + 3*sin(w_m*t))
//comparing with std eqn of PM signal x_PM(t) = A*cos(w_c*t + k_p*m(t))
//m(t) = a_m*sin(w_m*t)
//beeta = k_p*a_m
beeta = 3;
f_m1 = 1*10^3//modulating frequency for first case
f_m2 = 2*10^3//modulating frequency for second case
f_m3 = 500//modulating frequency for third  case

//calculations
f_B1 = 2*(beeta + 1)*f_m1//bandwidth for first case
f_B2 = 2*(beeta + 1)*f_m2//bandwidth for first case
f_B3 = 2*(beeta + 1)*f_m3//bandwidth for first case

//results
printf("\n\ni.Bandwidth for first case = %.2f Hz",
```

```
    f_B1);
printf("\n\nii.Bandwidth for second case = %.2f Hz",
    f_B2);
printf("\n\nii.Bandwidth for third case = %.2f Hz",
    f_B3);
```

Scilab code Exa 5.46 bandwidth of FM signal

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.45(PAGENO 264)");

//given
//x_cFM(t)  = 10*cos(w_c*t + 3*sin(w_m*t))
//by omparing with standard equation i.e A*cos(w_c*t
    + beta*sin(w_m*t))
//we get
beta = 3

//calculations
//first case
f_m1 = 1*10^3//modulating frequency for first case
f_B1 = 2*(beta +1)*f_m1//bandwidth for first case
//second case
beta2 = 3/2//beta for second case
f_m2 = 2*10^3//modulating frequency for second case
f_B2 = 2*(beta2 +1)*f_m2//bandwidth for second case
//third case
beta3 = 6//beta for third case
f_m3 = .5*10^3//modulating frequency for third case
f_B3 = 2*(beta3 +1)*f_m3//bandwidth for third case
```

```
//results
printf("\n\ni.Bandwidth for first case = %.2f Hz",
   f_B1);
printf("\n\nii.Bandwidth for second case = %.2f Hz",
   f_B2);
printf("\n\nii.Bandwidth for third case = %.2f Hz",
   f_B3);
```

Scilab code Exa 5.47 bandwidth and frequency deviation of FM signal

```
 //ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.47(PAGENO 264)");

//given
f_m = 2*10^3//modulating frequency for first case
delta_f1 = 5*10^3//frequency deviation for first
   case
f_m1 = 1*10^3//modulating frequency for second case
//beeta = (k_f*a_m)/(w_m) = delta_f/f_m

//calculations
beeta = delta_f1/f_m
f_B1 = 2*(beeta + 1)*f_m//bandwidth for first case
//beeta1 = (k_f*3*a_m)/(.5*w_m) = delta_f/f_m
   therefore
beeta1 = 6*beeta
delta_f2 = beeta1 * f_m1 //frequency deviation for
   second case
f_B2 = 2*(beeta1 + 1)*f_m1//bandwidth for second
```

```
    case

//results
printf("\n\ni.Bandwidth for first case = %.2f Hz",
    f_B1);
printf("\n\nii.a.Frequency deviation for second case
     =%.2f Hz",delta_f2);
printf("\n\n    b.Bandwidth for second case = %.2f Hz
    ",f_B2);
```

Scilab code Exa 5.48 bandwidth calculation using the given formula and carsons rule

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.48(PAGENO 264)");

//given
delta_f = 75*10^3//frequency deviation
f_M = 15*10^3//modulating frequency

//calculations
//we have w_m = 2*%pi*f_M where f_M = 15KHz,we get
D = delta_f/f_M//deviation ratio
//by using thr given formula, the bandwidth will be
f_B1 = 2*(D+2)*f_M
//Using Carson's rule, the bandwidt will be
f_B2 = 2*(D+1)*f_M

//results
printf("\n\ni.Bandwidth calculation using thr given
```

```
    formula = %.2f Hz",f_B1)
printf("\n\nii.Bandwidth calculation using the
    carson rule = %.2f Hz",f_B2)
```

Scilab code Exa 5.49 frequency multiplication and maximum frequency deviation

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.49(PAGENO 265)");

//given
//x_NBFM(t) = A*cos(w_c*t + sin(w_m*t))
f_c = 200*10^3//carrier frequency
f_m_max = 15*10^3//maximum modulating frequency
f_m_min = 50//minimum modulating frequency
delta_f = 75*10^3//maximum frequency deviation

//calculations
beeta_min = delta_f/f_m_max;
beeta_max = delta_f/f_m_min;
//if beeta_1 =.5, where beeta_1 is the input beeta,
    then the required frequency multiplication will
    be
beeta_1 = .5
n = beeta_max/beeta_1//frequency multiplication
delta_f1 = delta_f/n//maximum allowed frequency
    deviation

//results
printf("\n\ni.Frequency multiplication = %.2f ",n);
```

```
printf("\n\nii.Maximum allowed frequency deviation =
    %.2f Hz",delta_f1)
```

Scilab code Exa 5.50 maximum frequency deviation and carrier frequency for given maximum and minimum frequencies

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.50(PAGENO 265)");

//given
f_1 = 200*10^3//frequency applied at first stage
delta_f1 = 25//frequency deviation at first stage
n1 = 64//frequency multiplication at first stage
n2 = 48//frequency multiplication at second stage
f_LO = 10.8*10^6//frequency of oscillator as shown
    if block diagram

//calculations
delta_f = delta_f1*n1*n2//maximum frequency
    deviation
f_2 = n1*f_1//frequency applied at second stage
f_3a = f_2 + f_LO//frequency applied the third stage
f_3b = f_2 - f_LO//frequency applied the third stage
f_c1 = n2*f_3a//carreir frequency for maximun f_3
f_c2 = n2*f_3b//carreir frequency for minimum f_3

//results
printf("\n\ni.Maximum frequency deviation =%.2f Hz",
    delta_f);
printf("\n\nii.a.Carrier frequency for maximum f_3 =
```

```
    %.2f Hz",f_c1);
printf("\n\n    b.Carrier frequency for minimum f_3 =
    %.2f Hz",f_c2);
```

Scilab code Exa 5.51 multiplier and mixer oscillating frequency for Armstrong type FM generator

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.51(PAGENO 266)");

//given
f_c = 108*10^6//carrier frequency
f_1 = 2*10^5//crystal oscillator frequency
beta = .2//phase deviation
f_m = 50//minimum frequency
delta_f = 75*10^3//frequency deviation
n_2 = 150

//calculations
delta_f1 = beta * f_m;
n_12 = delta_f /delta_f1;
//f_2 = n_1*f_1 = n_1 * 2*10^5Hz
//assuming down convertions , we have
//f_2 - f_LO = (f_c/n_2)
//thus
f_LO = ((n_12*f_1) - f_c)/n_2;
n_1 = n_12/n_2

//results
printf("\n\n n_1 = %.2f",n_1)
```

```
printf("\n\n Mixer oscillator frequency= %.2f hz",
   f_LO);
```

Scilab code Exa 5.52 frequency multiplication

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 5
//ANGLE MODULATION
clear all;
clc;
printf("EXAMPLE 5.52(PAGENO 266)");

//given
delta_f = 50//frequency deviation
delta_f2 = 20*10^3//frequency deviation for
   sinusoidal FM wave i.e second case
f_m1 = 120//modualting frequency for first case
f_m2 = 240//modulating frquency for second case

//calculations
//first case
delta_f1 = (f_m2/f_m1)*delta_f//frequency deviation
   for sinusoidal PM wave
n1 = delta_f2/delta_f1//frequency multiplication for
    sinusoidal PM wave
//second case
n2 = delta_f2/delta_f//frequency multiplication for
   sinusoidal FM wave

//results
printf("\n\ni.Frequency multiplication for PM wave =
    %.2f ",n1);
printf("\n\nii.Frequency multiplication for FM wave
   = %.2f ",n2);
```

Chapter 6

Noise

Scilab code Exa 6.1 rms noise voltage

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 6
//NOISE
clear all;
clc;
printf("EXAMPLE 6.1(PAGENO 281)");

//given
R = 10*10^3//resistance of amplifier in ohms
T = 273+27//temperature in kelvin
B = (20-18)*10^6//bandwidth
k = 1.38*10^-23//boltzman's constant

//calculations
V_n = sqrt(4*R*k*T*B);//rms noise voltage

//result
printf("\n\nRms noise voltage = %.10f V",V_n);
```

Scilab code Exa 6.2 rms noise voltage

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 6
//NOISE
clear all;
clc;
printf("EXAMPLE 6.2(PAGENO 281)");

//given
R_1  = 300//equivalent noise resistance
R_2 = 400//input resistance
T = 273+27//temperature in kelvin
B = 7*10^6//bandwidth
k = 1.38*10^-23//boltzman's constant

//calculations
R_s = R_1 +R_2//effective resistance in series
V_nr = sqrt(4*k*T*B*R_s)//rms noise voltage

//result
printf("\n\nRms noise voltage = %.10f V",V_nr)
```

Scilab code Exa 6.3 rms voltage due individual resistance and resistances in parallel and series

```

//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 6
//NOISE
```

```
clear all;
clc;
printf("EXAMPLE 6.3(PAGENO 282)");

//given
R_1 = 20*10^3//resistance one
R_2 = 50*10^3//resistance two
T = 273+15//temperature in kelvin
B = 100*10^3//bandwidth
k = 1.38*10^-23//boltzman's constant

//calculations
R_s = R_1 +R_2//series effective resistance
R_p = (R_1*R_2)/(R_1 + R_2)//parallel effective resistance
V_1 = sqrt(4*k*T*R_1*B)//noise voltage in R_1
V_2 = sqrt(4*k*T*R_1*B)//noise voltage in R_2
V_s = sqrt(4*k*T*R_s*B)//noise voltage when resistance connected in series
V_p = sqrt(4*k*T*R_p*B)//noise voltage when resistance connected in parallel

//results
printf("\n\ni.Noise voltage due to R_1 = %.10f V",V_1);
printf("\n\nii.Noise voltage due to R_2 = %.10f V",V_2);
printf("\n\niii.Noise voltage due to two resistance in series = %.10f V",V_s);
printf("\n\niv.Noise voltage due to two resistance in parallel = %.10f V",V_p);
```

Scilab code Exa 6.4 equivalent input noise resistance

```
//ANALOG AND DIGITAL COMMUNICATION
```

```
//BY Dr.SANJAY SHARMA
//CHAPTER 6
//NOISE
clear all;
clc;
printf("EXAMPLE 6.4(PAGENO 283)");

//given
A_1 = 10//voltage gain for first stage
A_2 = 25//volatage gain for second stage
R_i1 = 600//input resistance for first stage in ohms
R_eq1 = 1600//equivalent noise resistance for first
    stage
R_01 = 27*10^3//Output resistance for first stage
R_i2 = 81*10^3//input resistance for second stage
R_eq2 = 10*10^3//Equivalent noise resistance for
    second stage
R_02 = 1*10^6//putput resistance for second case

//calculations
R_1 = R_i1 + R_eq1
R_2 = ((R_01*R_i2)/(R_01+R_i2)) + R_eq2
R_3 = R_02
R_eq = R_1 + (R_2/A_1^2) + R_3/(A_1^2 *A_2^2);

//results
printf("\n\nEquivalent input noise resistance = %.2f
    Ohms",R_eq);
```

Scilab code Exa 6.7 output voltage across the circuit measured by a wide band voltmeter

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 6
```

```
//NOISE
clear all;
clc;
printf("EXAMPLE 6.7(PAGENO 295)");

//given
T = 273 + 17//temperature in kelvin
Q = 10//quality factor
c  = 10*10^-12//capacitance
f_r = 100*10^6//resonate frequency
k = 1.38*10^-23//boltzman's constant

//calculations
delta_f = f_r/Q//bandwidth of the tuned circuit
w = 2*%pi*f_r;//angular frequency
R = 1/(Q*w*c);//resistance
V_no = sqrt(4*k*Q^2*T*delta_f*R) //output voltage

//results
printf("\n\nOutput voltge = %.10f V",V_no);
```

Scilab code Exa 6.8 noise figure and equivalent temperature

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 6
//NOISE
clear all;
clc;
printf("EXAMPLE 6.8(PAGENO 297)");

//given
R_a = 50//antenna resistance
R_eq = 30//equivalent noise resistance of receiver
T_0 = 290//initial temperature in degree kelvin
```

```
//calculations
F = 1+(R_eq/R_a);//noise figure
F_dB = 10*log10(F)//noise figure in decibels
T_eq = T_0*(F-1)//equivalent temperature

//results
printf("\n\ni.Noise figure in decibels = %.2f dB",
   F_dB);
printf("\n\nii.Equivalent temperature  = %.2f degree
    kelvin",T_eq)
```

Scilab code Exa 6.9 noise figure

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 6
//NOISE
clear all;
clc;
printf("EXAMPLE 6.9(PAGENO 302)");

//given
R_eq = 2518//equivalent resistance in ohms
R_t = 600//input impedence in ohms
R_a= 50//output impedencre in ohms

//calculations
R_eq1 = R_eq - R_t;
F = 1 + (R_eq1/R_a) //noise figure
F_dB = 10*log10(F)//noise figure in dB

//results
printf("\n\nNoise figure in dB = %.2f dB",F_dB);
printf("\n\nNote:Calculation mistake is their in
   text book in finding noise figure in dB")
```

Scilab code Exa 6.10 overall noise figure

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 6
//NOISE
clear all;
clc;
printf("EXAMPLE 6.10(PAGENO 305)");

//given
F_1 = 2//noise figure of first stage in dB
A_1 = 12//gain in first stage in dB
F_2 = 6//noise figure of second stage in dB
A_2 = 10//gain in first second in dB


//calculations
F_1ratio = exp((F_1/10)*log(10));//noise figure of
    first stage in ratio
F_2ratio = exp((F_2/10)*log(10));//noise figure of
    second stage in ratio
A_1ratio = exp((A_1/10)*log(10));//gain of first
    stage in ratio
A_2ratio = exp((A_2/10)*log(10));//gain of second
    stage in ratio
F = F_1ratio + ((F_2ratio - 1)/(A_1ratio));//Overall
    noise figure
F_dB = 10*log10(F);//Overall noise figure  in dB

//results
printf("\n\nOverall noise figure = %.2f dB",F_dB );
```

Scilab code Exa 6.11 overall noise figure

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 6
//NOISE
clear all;
clc;
printf("EXAMPLE 6.11(PAGENO 306)");

//given
F_1 = 9//noise figure for first stage in dB
F_2 = 20//noise figure for second stage in dB
A_1 = 15//gain in first stage in dB

//calculations
F_1ratio = exp((F_1/10)*log(10));//noise figure of
    first stage in ratio
F_2ratio = exp((F_2/10)*log(10));//noise figure of
    second stage in ratio
A_1ratio = exp((A_1/10)*log(10));//gain of first
    stage in ratio
F = F_1ratio + ((F_2ratio - 1)/(A_1ratio));
 F_dB = 10*log10(F);

 //results
 printf("\n\nOverall noise figure = %.2f dB", F_dB )
    ;
```

Scilab code Exa 6.12 rms noise voltage

```
//ANALOG AND DIGITAL COMMUNICATION
```

```
//BY Dr.SANJAY SHARMA
//CHAPTER 6
//NOISE
clear all;
clc;
printf("EXAMPLE 6.12(PAGENO 307)");

//given
f_1 = 18*10^6//lower operating frequency in Hz
f_2 = 20*10^6//lower operating frequency in Hz
T = 273 + 17//temperature in kelvin
R = 10*10^3//input resistance
k = 1.38*10^-23//boltzman's constant

//calculations
B = f_2 - f_1//bandwidth in Hz
V_n = sqrt(4*k*B*R*T);//rms noise voltage

//results
printf("\n\nrms noise voltage = %.10f V",V_n);
```

Scilab code Exa 6.14 meter reading in volts and resistance at given temperature

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 6
//NOISE
clear all;
clc;
printf("EXAMPLE 6.14(PAGENO 308)");

//given
A = 60//gain of noiseless amplifier
V_n1 = 1*10^-3//output of the amplifier
```

```
B = 20*10^3//initial bandwidth
B1 = 5*10^3//change in bandwidth
k = 1.38*10^-23//boltzman's constant
T = 273 + 80//temperature in degree kelvin

//calculaitons
//since the bandwidth is reesuced to 1/4th of its value,therefore the noise voltage
//will be V_n proportional to sqrt(B)
//Hence, the noise voltage at 5KHz will become half its value at 20KHz bandwidth i.e,
V_n = .5*10^-3//noise voltage in volts
V_no = V_n1/A;//noise ouput voltage
R = (V_no^2/(4*k * T * B ));//resistance at 80degree celcius

//results
printf("\n\ni.Meter reading in volts = %.10f V",V_n);
printf("\n\nii.Resistance at 80 degree celcius = %.2f ohms",R);
printf("\n\nNote: There is calculation mistake in textbook in the measurement of resistance they took constant in formula as 1 instead of 4");
```

Scilab code Exa 6.16 overall noise figure of three stage cascade amplifier

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 6
//NOISE
clear all;
clc;
printf("EXAMPLE 6.16(PAGENO 309)");

```

```
//given
A_1 = 10//gain in first stage in dB
A_2 = 10//gain in second stage in dB
A_3 = 10//gain in third stage in dB
F_1 = 6//noise figure for first stage in dB
F_2 = 6//noise figure for second stage in dB
F_3 = 6//noise figure for third stage in dB

//calculations
F_1ratio = exp((F_1/10)*log(10));//noise figure of first stage in ratio
F_2ratio = exp((F_2/10)*log(10));//noise figure of second stage in ratio
F_3ratio = exp((F_3/10)*log(10));//noise figure in third stage in ratio
A_1ratio = exp((A_1/10)*log(10));//gain of first stage in ratio
A_2ratio = exp((A_2/10)*log(10));//gain of second stage in ratio
A_3ratio = exp((A_3/10)*log(10));//gain of third stage in ratio
F = F_1ratio + ((F_2ratio - 1)/(A_1ratio)) + ((F_3ratio - 1)/(A_2ratio*A_1ratio));//Overall noise figure

//results
 printf("\n\nOverall noise figure of three stage cascaded amplifier = %.2f ", F );
```

Scilab code Exa 6.17 overall noise figure of two stage cascaded amplifier

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 6
//NOISE
```

```
clear all;
clc;
printf("EXAMPLE 6.17(PAGENO 310)");

//given
G_1 = 10//gain in first stage in dB
//noise figure for both the stages are same
F_1 = 10//noise figure for first stage in dB
F_2 = 10//noise figure for second stage in dB

//calculations
F_1ratio = exp((F_1/10)*log(10));//noise figure of
    first stage in ratio
F_2ratio = exp((F_2/10)*log(10));//noise figure of
    second stage in ratio
G_1ratio = exp((G_1/10)*log(10));//gain of first
    stage in ratio
F = F_1ratio + ((F_2ratio - 1)/(G_1ratio));//Overall
     noise figure
F_dB= 10*log10(F)////Overall noise figure in dB

//results

 printf("\n\nOverall noise figure  = %.2f dB", F_dB
    );
```

Scilab code Exa 6.18 overall noise figure and overall gain

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 6
//NOISE
clear all;
clc;
printf("EXAMPLE 6.18(PAGENO 310)");
```

```

//given
G_1 = 4//gain in first stage in dB
G_2 = 10//gain in second stage in dB
F_1 = 10//noise figure for first stage in dB
F_2 = 10//noise figure for second stage in dB

//calculations
F_1ratio = exp((F_1/10)*log(10));//noise figure of
    first stage in ratio
F_2ratio = exp((F_2/10)*log(10));//noise figure of
    second stage in ratio
G_1ratio = exp((G_1/10)*log(10));//gain of first
    stage in ratio
G_2ratio = exp((G_2/10)*log(10));//gain of second
    stage in ratio
F = F_1ratio + ((F_2ratio - 1)/(G_1ratio));//Overall
     noise figure
G = log10(G_1ratio *G_2ratio );
F_dB= 10*log10(F)////Overall noise figure in dB

//results
printf("\n\ni.Overall noise figure  = %.2f dB", F_dB
    );
 printf("\n\nii.Overall gain = %.2f dB",G );
printf("\n\nNote:There is mistake in calculation of
    overall gain in textbook")
```

Scilab code Exa 6.19 overall noise figure

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 6
//NOISE
clear all;
```

```
clc;
printf("EXAMPLE 6.19(PAGENO 310)");

//given
G_1 = 15//gain in first stage in dB
F_1 = 9//noise figure for first stage in dB
F_2 = 20//noise figure for second stage in dB

//calculations
F_1ratio = exp((F_1/10)*log(10));//noise figure of first stage in ratio
F_2ratio = exp((F_2/10)*log(10));//noise figure of second stage in ratio
G_1ratio = exp((G_1/10)*log(10));//gain of first stage in ratio
F = F_1ratio + ((F_2ratio - 1)/(G_1ratio));//Overall noise figure
F_dB= 10*log10(F)////Overall noise figure in dB

//results
printf("\n\nOverall noise figure = %.2f dB", F_dB )
;
```

Scilab code Exa 6.20 noise temperature of the receiver and overall noise temperature

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 6
//NOISE
clear all;
clc;
printf("EXAMPLE 6.20(PAGENO 311)");

//given
```

```
F_2 = 20//noise figure of receiver in dB
G_1 = 40//gain of low noise amplifier in dB
T_e1 = 80//noise temperature of low noise amplifier
    in degree kelvin
T_0 = 300//room temperature

//calculations
F_2ratio = exp((F_2/10)*log(10));//noise figure of
    receiver in ratio
G_1ratio = exp((G_1/10)*log(10));//gain of  low
    noise amplifier
T_e2 = (F_2ratio-1)*T_0//noise temperature of the
    receiver in degree kelvin
T_e = T_e1 +(T_e2/G_1ratio)//overall noise
    temperature in degree kelvin

//results
printf("\n\ni.Noise Temperature of the receiver = %
    .2f degkelvin ",T_e2);
printf("\n\nii.Overall noise temperature = %.2f
    degkelvin",T_e);
```

Scilab code Exa 6.21 overall noise temperature of the receiver and overall noise figure

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 6
//NOISE
clear all;
clc;
printf("EXAMPLE 6.21(PAGENO 311)");

//given from the figure
G_1ratio = 1000//gain of master amplifier
```

```
G_2ratio = 100//gain of TWT
G_3ratio = 10000//gain of mixer and IF amplifier
F_2ratio = 4//noise figure of TWT
F_3ratio = 16//noise figure of mixer and IF
    amplifier
T_0 =273 + 17//ambident temperature in degree kelvin
T_e1 = 5//temperature of master amplifier in degree
    kelvin

//calculaitons
F_1 = 1 + (T_e1/T_0);//noise figure of master
    amplifier
F = F_1 + ((F_2ratio - 1)/(G_1ratio)) + ((F_3ratio -
     1)/(G_2ratio*G_1ratio));//Overall noise figure
F_dB = 10*log10(F);//overall noise figure in dB
T_e = (F - 1)*T_0;//overall noise temperature of the
     receiver

//results
 printf("\n\ni.Overall noise temperature of the
     receiver =%.2f degreekelvin",T_e);
 printf("\n\nii.Overall noise figure  = %.6f dB",
     F_dB);
```

Chapter 7

Sampling Theory and Pulse Modulation

Scilab code Exa 7.1 Nyquist rate

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//SAMPLING THEORY AND PULSE MODULATION
clear all;
clc;
printf("EXAMPLE 7.1(PAGENO 324)");

//given
//analog signal x(t) = 3*cos(50*%pi*t) + 10*sin(300*%pi*t) - cos(100*%pi*t)
//comparing signal with x(t) = 3*cos(w_1*t) + 10*sin(w_2*t) - cos(w_3*t)
//therefore
w_1 = 50*%pi;//first frequency in rad/sec
w_2 = 300*%pi;//second frequency in rad/sec
w_3 = 100*%pi;//third frequency in rad/sec

//calculations
```

```
f_1 = w_1/(2*%pi);//first frequency in Hz
f_2 = w_2/(2*%pi);//second frequency in Hz
f_3 = w_3/(2*%pi);//third frequency in Hz
f_m = f_2//maximum frequency
f_s = 2*f_m//nyquist rate for a signal

//results
printf("\n\nNyquist rate  = %.2f Hz",f_s);
```

Scilab code Exa 7.2 Nyquist rate and Nyquist interval

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//SAMPLING THEORY AND PULSE MODULATION
clear all;
clc;
printf("EXAMPLE 7.12(PAGENO 325)");

//given
//x(t) = (1/()2*%pi))*cos(4000*%pi*t)*cos(1000*%pi*t
    )
//exapnding
disp("x(t) = (1/(2*%pi)*cos(4000*%pi*t)*cos(1000*%pi
    *t)");
disp("x(t) = (1/(4*%pi)*2*cos(4000*%pi*t)*cos(1000*
    %pi*t)");
disp("x(t) = (1/(4*%pi))*[cos(4000*%pi*t + 1000*pi*t
    )*cos(4000*%pi*t - 1000*%pi*t)]")
disp("x(t) = (1/(4*%pi))*[cos(5000*%pi*t + cos(3000*
    %pi*t))]")
//by comparing above equation with x(t) = (1/(4*%pi)
    )*[cos(w_1*t) + cos(w_2*t)]
w_1 = 5000*%pi
w_2 = 3000*%pi
```

```

//calculations
f_1 = w_1/(2*%pi);
f_2 = w_2 /(2*%pi);
f_m = f_1
f_s = 2*f_m//Nyquist rate
T_s = 1/f_s//Nyquist interval

//results
printf("\n\nNyquist rate = %.2f Hz",f_s);
printf("\n\nNyquist interval = %.5f seconds",T_s);
```

Scilab code Exa 7.3 discrete time signal for the given conditions

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//SAMPLING THEORY AND PULSE MODULATION
clear all;
clc;
printf("EXAMPLE 7.13(PAGENO 326)");

//given
//x(t) = 8*cos(200*%pi*t)
f= 100//highest frequency component of continuous time signal in hertz
f_s2 = 400//sampling frequency in hertz for second condition
f_s3 = 400//sampling frequency in hertz for third condition
f_s4 = 150//sampling frequency in hertz for fourth condition since 0 < f_s4 < f_s2/2

//calcultions
NR = 2*f//Nyquist rate
```

```
F_1 = f/NR;
F_2 = f/f_s2;
F_3 = f/f_s3;
F_4 = f/f_s4;
f_4 = f_s4*F_4;

//results
printf("\n\nThe discrete time signal x(n) for the
    first condition is x(n) = 8*cos(2*3.14*%.2f*n)",
    F_1);
printf("\n\nthe discrete time signal x(n) for the
    second condition is x(n) = 8*cos(2*3.14*%.2f*n)",
    F_2);
printf("\n\nthe discrete time signal x(n) for the
    third condition is x(n) = 8*cos(2*3.14*%.2f*n)",
    F_3);
printf("\n\nThe discrete time signal x(n) for the
    fourth condition is x(n) = 8*cos(2*3.14*%.2f*t)",
    f_4);
```

Scilab code Exa 7.4 Nyquist rate for the continuous signal

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//SAMPLING THEORY AND PULSE MODULATION
clear all;
clc;
printf("EXAMPLE 7.4(PAGENO 327)");

//given
//x(t) = 6*cos(50*%pi*t) + 20*sin(300*%pi*t) - 10*
    cos(100*%pi*t)
//by comparing with standard eqn x(t) = A_1*cos(w_1*
    t) + A_2*sin(w_2*t) + A_3*cos(w_3*t) we get
```

```
w_1 = 50*%pi//frequency in rad/sec
w_2 =300*%pi//frequency in rad/sec
w_3 = 100*%pi//frequency in rad/sec

//calculations
f_1 = w_1/(2*%pi)//frequency in hertz
f_2 = w_2/(2*%pi)//frequency in hertz
f_3 = w_3/(2*%pi)//frequency in hertz
if (f_1 > f_2  & f_1> f_3) then
     f_max = f_1
elseif (f_2 > f_1 & f_2> f_3) then
    f_max = f_2
else (f_3 > f_1 & f_3> f_2) then
    f_max = f_3
    end
f_s = 2*f_max;//nyquist rate

//results
printf("\n\nNyquist rate for a continuous signal = %
   .2f Hz",f_s);
```

Chapter 8

Waveform Coding Techniques

Scilab code Exa 8.2 code word length and transission bandwidth and final bit rate and quantization noise ratio

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//WAVEFORM CODING TECHNIQUES
clear all;
clc;
printf("EXAMPLE 8.2(PAGENO 386)");

//given
f_m = 4.2*10^6//bandwidth of television signal
q = 512//quantization levels

//calculations
//number of bits and quantization levels are related in binary PCM as q = 2^v
//where v is code word length
v = (log10(q)/log10(2));//code word length
BW = v*f_m//transmission channel bandwidth which is greater than or equal to obtained value
f_s = 2*f_m//sampling frequency which is greater
```

```
than or equal to obtained value
r = v*f_s//signaling rate of final bit rate
SbyN_dB = 4.8 + 6*v//output signal to noise ratio which is less than or equal to obtained value

//results
printf("\n\ni.  Code word length  = %.2f bits",v);
printf("\n\nii. Transmission bandwidth = %.2f Hz",BW);
printf("\n\niii.Final bit rate = %.2f bits/sec",r);
printf("\n\niv.Output signal to quantization noise ratio = %.2f dB",SbyN_dB);
printf("\n\nNote:There is misprint in the question i.e TV signal bandwidth ")
```

Scilab code Exa 8.3 number of bits required and bandwidth of pcms and signalling rate

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//WAVEFORM CODING TECHNIQUES
clear all;
clc;
printf("EXAMPLE 8.3(PAGENO 387)");

//given
f_m = 4*10^3//maximum frequency or bans
x_max = 3.8//maximun input signal
P = 30*10^-3//average power of signal
SbyN_dB= 20//signal to noise ratio in db

//calculations
SbyN = exp((SbyN_dB/10)*log(10));
v = (log2((SbyN*(x_max)^2)/(3*P))/2);//number of
```

```
bits required per sample
BW = 30*v*f_m//transmission channel bandwidth which is greater than or equal to obtained value
r=BW*2//wkt signalling rate is two times the transmission bandwidth

//results
printf("\n\ni.Number of bits required = %.2f bits",v);
printf("\n\nii.Bandwidth required for 30 PCM coders = %.2f Hz",BW);
printf("\n\niii.Signalling rate=%.2f bitspersecond",r);
printf("\n\nNote: In the textbook they took number of bits as approximation from 6.98 to 7\nso thats why we get difference in the rest of calculations and also their is\n mistake in the calculation of sampling rate")
```

Scilab code Exa 8.4 sampling rate and number of bits in each PCM and bit rate and transmission bandwidth

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//WAVEFORM CODING TECHNIQUES
clear all;
clc;
printf("EXAMPLE 8.4(PAGENO 388)");

//given
e_max = .001//maximum quantization error
x_max = 10//maximum amplitude
x_min = -10//minumum amplitude
f_m = 100//bandwidth of ;input signal
```

```

//calculations
delta = 2*e_max//step size
q = (2*x_max)/delta//quantization levels
f_s = 2*f_m//sampling frequency
v = log10(q) /log10(2);//number of bits in the PCM word
r = v * f_s//bit rate required in the PCM signal which is greater than or equal to obtained value
BW = .5*r//transmission channel bandwidth which is greater than or equal to obtained value

//results
printf("\n\ni.Minimum sampling rate required = %.2f Hz",f_s);
printf("\n\nii.Number of bits in each PCM word = %.2f bits",v);
printf("\n\niii.Minimum bit rate required in the PCM signal =%.2f bits/sec",r);
printf("\n\niv.Transmission bandwidth  = %.2f Hz",BW)
printf("\n\nNote: In the textbook they took number of bits as approximation from 13.28 to 14 so thats why we get difference in the rest of calculations")
```

Scilab code Exa 8.5 transmission bandwidth and sampling frequency

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//WAVEFORM CODING TECHNIQUES
clear all;
clc;
printf("EXAMPLE 8.5(PAGENO 389)");
```

```

//given
f_m = 3.4*10^3//maximum frequency in the signal
N =24//number of voice signals
r = 1.5*10^6//signaling rate
v  = 8//bits  of encoder

//calculations
BW = N * f_m//transmission bandwidth
r_1 = r/N//bit rate for one channel
f_s = r_1/v//sampling frquency

//results
printf("\n\ni.Transmission bandwidth  = %.2f Hz",BW)
    ;
printf("\n\nii.Sampling frequency = %.2f Hz or
    samples per second",f_s)
```

Scilab code Exa 8.6 message bandwidth and signal to noise ratio

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//WAVEFORM CODING TECHNIQUES
clear all;
clc;
printf("EXAMPLE 8.6(PAGENO 389)");

//given
v = 7//bits  of encoder
r = 50*10^6//bit rate of the system

//calculations
f_m = r/(2*v)//maximum message bandwidth which is
    less than or equal to obtained value
```

```
SbyN_dB = 1.8 + 6*v//signal to noise ratio in dB

//results
printf("\n\ni.Maximum message bandwidth = %.2f Hz",f_m);
printf("\n\nii.Signal to noise ratio when modulating frquency is 1MHz applied = %.2f dB",SbyN_dB)
```

Scilab code Exa 8.7 number of bits in codeword and sampling rate and bit rate

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//WAVEFORM CODING TECHNIQUES
clear all;
clc;
printf("EXAMPLE 8.7(PAGENO 390)");

//given
f_m = 3*10^3//maximum frequency
M = 16//number of quantization levels
q = M//number of quantization levels

//calculations
v = log2(q);//number of bits
f_s = 2*f_m//sampling frequency or rate which is greater than or equal to obtained value
r = v*f_s//bit transmission rate which is greater than or equal to obtained value

//results
printf("\n\ni.Number of bits in a codeword = %.2f bits",v);
printf("\n\nii.Minimum sampling rate = %.2f Hz ",f_s
```

```
);
printf("\n\niii.Bit transmission rate =%.2f bits/sec
    ",r);
```

Scilab code Exa 8.8 signal to the noise ratio

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//WAVEFORM CODING TECHNIQUES
clear all;
clc;
printf("EXAMPLE 8.8(PAGENO 391)");

//given
f_m = 3.5*10^3//maximum frequency
r = 50*10^3//bit rate
v_rms = .2//rms value of input signal
R = 1//resistance
x_max = 2//maximum peak voltage

//calculations
f_s = 2*f_m;//sampling frequency
v = r/f_s;//number of bits
P = v_rms^2 / R//Normalized signal power
SbyN = ((3*P) * 2^(2*v)) /(x_max^2);//signal to
    noise ratio
SbyN_dB = 10*log10(SbyN)//signal to noise ratio in
    dB

//results
printf("\n\ni.Signal to noise ratio in dB = %.2f dB"
    ,SbyN_dB);
printf("\n\nNote: They took number of bits as
    approximation from 7.142 to 8 so the SbyN changes
```

```
")
```

Scilab code Exa 8.10 signal to the noise ratio and number of bits

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//WAVEFORM CODING TECHNIQUES
clear all;
clc;
printf("EXAMPLE 8.10(PAGENO 392)");

//given
//x(t) = 3*cos(500*%pi*t)
v = 10//number of bits
A_m = 3//peak voltage
SbyN_2 = 40//signal to noise to noise ratio in second condition

//calculations
SbyN = 1.8 +6*v//signal to noise ratio in dB
v_2 = (40 - 1.8)/6//number of bits needed for SbyN = 40

//results
printf("\n\ni.Signal to noise to ratio in dB = %.2f dB",SbyN);
printf("\n\nii.Number of bits needed for noise ratio 40 = %.2f bits",v_2);
```

Scilab code Exa 8.11 maximum frequency

```
//ANALOG AND DIGITAL COMMUNICATION
```

```
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//WAVEFORM CODING TECHNIQUES
clear all;
clc;
printf("EXAMPLE 8.11(PAGENO 393)");

//given
v = 7//number of bits
r = 56*10^3//signaling rate

//calculations
SbyN = 1.8 +6*v//signal to noise ratio in dB
f_s = r/v//sampling frequency
f_m = f_s/2//maximum frequency which is less than or
    equal to obtained value

//results
printf("\n\nMaximum frequency = %.2f Hz",f_m)
```

Scilab code Exa 8.13 maximum amplitude

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//WAVEFORM CODING TECHNIQUES
clear all;
clc;
printf("EXAMPLE 8.13(PAGENO 404)");

//given
f_m = 3*10^3//bandwidth or maximum frequency
n = 5//system operation times
delta = 250*10^-3//step size in volts
f_m1 = 2*10^3//given maximum frequency to calculate
```

```
amplitude

//calculations
NR = 2 * f_m//nyquist rate
f_s = n * NR//sampling frequency
T_s = 1/f_s//sampling interval
A_m =(delta/(2 * %pi * f_m1* T_s))//Maximum amplitude

//result
printf("\n\nMaximum amplitude for 2KHz input sinusoid = %.2f V",A_m);
```

Scilab code Exa 8.14 signalling rate

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//WAVEFORM CODING TECHNIQUES
clear all;
clc;
printf("EXAMPLE 8.14(PAGENO 406)");

//given
f_s = 8*10^3//sampling rate
q = 64//quantization levels
delta = 31.25//step size

//calculations
v = log2(q);//no fo bits in the PC
f_s= (2*%pi*3*10^3)/delta//signalling rate which should be greater than the obtaining value

//results
printf("\n\nSignalling rate = %.2f Hz",f_s);
```

Scilab code Exa 8.15 signal to the noise ratio of linear delta modulation system

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//WAVEFORM CODING TECHNIQUES
clear all;
clc;
printf("EXAMPLE 8.15(PAGENO 407)");

//given
f_m = 2*10^3//maximum frequency
f_s = 64*10^3//sampling frequency
f_M = 4*10^3//cut off frequency of low pass filter

//calculation
SNR_0 = (3 * f_s^3) /(8 * %pi^2 * f_m^2 * f_M);//
    signal to noise ratio of linear delta modulation
    system
SNR_dB = 10*log10(SNR_0);//SNR in dB

//result
printf("\n\nSignal to noise ratio of linear delta
    modulation system = %.2f dB",SNR_dB);
```

Scilab code Exa 8.16 signal to the noise ratio

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
```

```
//WAVEFORM CODING TECHNIQUES
clear all;
clc;
printf("EXAMPLE 8.16(PAGENO 407)");

//given
r = 64*10^3//data rate
f_s = 8*10^3//sampling frequency
N = 8//number of samples

//calcualtion
SNR_q = 1.8 + 6*N//signal to noise ratio

//result
printf("\n\nSignla to noise ratio  = %.2f dB",SNR_q)
   ;
printf("\n\nThe SNR of a DM system is 27.94dB which
   is too poor as \ncompared to 49.8db of an 8 bit
   PCM system. Thus, for all\n the simplicity of Dm,
   it cannot perform as well as an\n 8 bit PCM")
```

Scilab code Exa 8.17 sampling frequency and number of binary digits and quantizing level

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//WAVEFORM CODING TECHNIQUES
clear all;
clc;
printf("EXAMPLE 8.17(PAGENO 413)");

//given
r = 36000//bit rate of a channel
f_m = 3.2*10^3//maximum frequency
```

```
//calculations
f_s = 2*f_m//sampling frequency
v  = r/f_s//number of binary digits
q = 2^v//quantizing level

//results
printf("\n\ni.Sampling frequency  = %.2f Hz",f_s);
printf("\n\nii.Number of binary digits = %.2f ",v);
printf("\n\niii.Quantizing level = %.2f ",q);
printf("\n\nNote:In the textbook they took
   approximation in number of\n     binary levels")
```

Scilab code Exa 8.20 number of required levels and signal to noise ratio

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//WAVEFORM CODING TECHNIQUES
clear all;
clc;
printf("EXAMPLE 8.20(PAGENO 415)");

//given
SbyN_0dB = 40//signal to noise ratio in dB
SbyN_0 = exp((SbyN_0dB/10)*log(10))//signal to noise
    ratio
q = sqrt((2 / 3) * (SbyN_0));//quantizing level
v = log2(q)//number of binary bits
q_1 = 2^v//number of levels required
SbyN_dB1 = 1.76 + 6.02*v//output signal-to-
   quantizing noise ratio in dB

//results
printf("\n\nNumber of required levels = %.2f ",v);
```

```
printf("\n\nOutput signal-to-quantizing noise ratio
    = %.2f dB",SbyN_dB1);
printf("\n\nNote : In the textbook they took number
    of levels as approximation so we get change\n in
    SbyN")
```

Scilab code Exa 8.21 quantizing levels and number of bits and bandwidth

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 7
//WAVEFORM CODING TECHNIQUES
clear all;
clc;
printf("EXAMPLE 8.21(PAGENO 416)");

//given
SbyN_dB = 30//signal to noise ratio
f_s = 8000//sampling rate

//calculations
//for Sbyn_dB = 1.76 + 20*logq
x1 = (1 / 20)*(SbyN_dB - 1.76)
q1 = exp(x1*log(10))//quantizing level for first
    case
v1 = log2(q1)// number of bits  for first case
f_PCM1 = (v1 / 2) * f_s//minimum required bandwidth
     for first case
//for SbyN = 20logq - 10.1
x2 = (1/20) * (SbyN_dB + 10.1)
q2 =  exp(x2*log(10))//quantizing level for second
    case
v2 = log2(q2)// number of bits for second case
f_PCM2 =  (v2 / 2) * f_s//minimum required bandwidth
     for second case
```

```

//results
printf("\n\ni.a.Minimum number of quantizing levels for first case = %.2f ",q1);
printf("\n\n  b.Number of bits for first case =%.2f ",v1);
printf("\n\n  c.Minimum system bandwidth required for first case = %.2f hz",f_PCM1);
printf("\n\nii.a.Minimum number of quantizing levels for second case = %.2f ",q2);
printf("\n\n   b.Number of bits for second case =%.2f ",v2);
printf("\n\n   c.Minimum system bandwidth required for second case = %.2f hz",f_PCM2);
printf("\n\nNote:In the text book they took approximation in\nquantization levels and number bits")
```

Chapter 10

Digital multiplexers

Scilab code Exa 10.1 sampling rate

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 10
//DIGITAL MULTIPLEXERS
clear all;
clc;
printf("EXAMPLE 10.1(PAGENO 469)");

//given
X_1 = 4*10^3//first analog signal in Hz
X_2 = 4.5*10^3//second analog signal in Hz

//calculation
//the highest frequency cmponent of the composite
    signal consisting among two signal is X_2
f_sMIN = 2*X_2;

printf("\n\nThe minimum value of permissible
    sampling rate = %2f Hz",f_sMIN);
```

Scilab code Exa 10.2 signalling rate and channel bandwidth

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 10
//DIGITAL MULTIPLEXERS
clear all;
clc;
printf("EXAMPLE 10.2(PAGENO 469)");

//given
X_1 = 6*10^3//Nyquist rate in Hz obtained the table
X_2 = 2*10^3//Nyquist rate in Hz obtained the table
X_3 = 2*10^3//Nyquist rate in Hz obtained the table
X_4 = 2*10^3//Nyquist rate in Hz obtained the table

//calculations
s = 2000//speed of rotation
X1 = 3*s//number of samples produced per second for
    first signal
X2 = 1*s//number of samples produced per second for
    second signal
X3 = X2//number of samples produced per second for
    third signal
X3 = X2//number of samples produced per second for
    fourth signal
SR = X1 + 3*X2//signalling rate
BW = .5*SR//minimum channel bandwidth

//results
printf("\n\nIf the sampling commutator rotates at
    the rate of 2000 rotations per second the the
    signals X_1,X_2,X_3,X_4 will be sampled at their
    Nyquist rate")
```

```
printf("\n\nSignalling rate = %.2f samples per
   second",SR);
printf("\n\nMinimum channel bandwidth =%.2f Hz",BW);
```

Scilab code Exa 10.3 spacing between successive pulses of multiplxed signal

```

//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 10
//DIGITAL MULTIPLEXERS
clear all;
clc;
printf("EXAMPLE 10.3(PAGENO 470)");

//given
SR = 8000//sampling rate in samples per second
T = 1*10^-6//pulse duration
f = 3.4*10^3//highest frequency component

//calculations
//second case
NR = 2*f//Nyquist rate of sampling
T2 = 1/NR//time taken for one rotation of commutator


//results
printf("\n\nsampling rate for first condition =%.2f"
   ,SR);
printf("\n\nThere are 24 voice signals + 1
   synchronizing pulse")
printf("\n\nPulse width of each voice channel and
   synchronizing pulseis 1 microseconds ")
printf("\n\nNow, time taken by the commutator for 1
```

```
    rotation =1/8000 = 125*10^-6 seconds")
printf("\n\nNumber of pulses produced in one
    rotation = 24 + 1 = 25");
printf("\n\nTherefore, the leading edges of the
    pulses are at 125/25 = 5*10^-6 seconds distance")
printf("\n\nNyquist rate for second condition = %.2f
     Hz",NR);
printf("\n\nTime taken for one rotation of
    commutator = %.8f seconds",T2);
printf("\n\nTherefore, 147*10^-6 seconds corresponds
     to 25 pulses");
printf("\n\ntherefore, 1 pulse corresponds to
    5.88*10^-6 seconds");
printf("\n\nAs the pulse width of each pulse is
    1*10^-6 seconds, the spacing between adjacent
    pulses will be 4.88*10^-6 seconds\n and if we
    assume tou = 0 then the spacing between the
    adjacent pulses will be 5.88*10^-6 seconds ")
```

Scilab code Exa 10.4 signaling rate and channel bandwidth

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 10
//DIGITAL MULTIPLEXERS
clear all;
clc;
printf("EXAMPLE 10.4(PAGENO 471)");

//given
N = 6//number of channels
f_m = 5*10^3//bandwidth of each channel

//calculations
SR1= 2*f_m//minimum sampling rate
```

```
SR = N*SR1//sampling rate
BW =N*f_m//minimum channel bandwidth

//results
printf("\n\nSignaling rate =%.2f bits per second",SR
    );
printf("\n\nMinimum channel bandwidth = %.2f Hz",BW)
    ;
```

Chapter 11

Information Theory

Scilab code Exa 11.1 information content of given symbols

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.1(PAGENO 488)");
//given

Px_1=1/2;//probability 1
Px_2=1/4;//probability 2
Px_3=1/8;//probability 3
Px_4=1/8;//probability 4

//calculations
Ix_1 = log2(1/(Px_1))//information content in first probability
Ix_2 = log2(1/(Px_2))//information content in first probability
Ix_3 = log2(1/(Px_3))//information content in first probability
```

```
Ix_4 = log2(1/(Px_3))//information content in first
    probability

//results
printf("\n\ni. Information content of first symbol =
    %.2f bit",Ix_1);
printf("\n\nii. Information content of second symbol
    = %.2f bits",Ix_2);
printf("\n\niii. Information content of third
    symbol = %.2f bits",Ix_3);
printf("\n\niV. Information content of fourth symbol
    = %.2f bits",Ix_4);
```

Scilab code Exa 11.2 amount of information for a given probability of a symbol

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.2(PAGENO 488)");
//given
Px_i = 1/4//probability of a symbol

//calculation
Ix_i = (log(1/Px_i))/log(2)//formula for amount of
    information of a symbol

//result
printf("\n\ni. Amount of information = %.2f bits",
    Ix_i)
```

Scilab code Exa 11.3 amount of information content wrt to binary PCMs

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.3(PAGENO 489)");
//given
//since there are only two binary levels i.e. 1 or 0. Since, these two binary levels occur with equal            likelihood of occurrence will be
Px_1 = 1/2//probability of zero level
Px_2 = 1/2//probability of first level

//calculations
Ix_1 = log2(1/Px_1)//amount of information of zero level with base 2
Ix_2 = log2(1/Px_2)//amount of information of first level with base 2
Ix_1= log(1/Px_1)/log(2)//amount of information content with base 10
Ix_2 = Ix_1

//result
printf("\n\ni.Amount of information content wrt binary PCM 0 =  %.2f bit",Ix_1)
printf("\n\nii.Amount of information content wrt binary PCM 1 =  %.2f bit",Ix_2)
```

Scilab code Exa 11.4 amount of information content wrt to binary PCMs

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.4(PAGENO 489)");
//given
Px_1 = 1/4//probability wrt to binary PCM '0'
Px_2 = 3/4//probability wrt to binary PCM '1'

//calculations
Ix_1 = log2(1/Px_1)//amount of information of zero
    level with base 2
Ix_2 = log2(1/Px_2)//amount of information of first
    level with base 2
Ix_1= log(1/Px_1)/log(2)//amount of information
    content with base 10
Ix_2= log(1/Px_2)/log(2)//amount of information
    content with base 10

//results
printf("\n\ni.Amount of information carried wrt to
    binary PCM 0 = %.2f bits",Ix_1);
printf("\n\nii.Amount of information carried wrt to
    binary PCM 1 = %.2f bits",Ix_2);
```

Scilab code Exa 11.9 entropy and amount of information for the given messages

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
```

```
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.9(PAGENO 492)");
//given
Px_1 = .4//probability of first symbol
Px_2 = .3//probability of second symbol
Px_3 = .2//probability of third symbol
Px_4 = .1//probability of fourth symbol

//calculations
H_X = -Px_1*log2(Px_1)-Px_2*log2(Px_2)-Px_3*log2(Px_3)-Px_4*log2(Px_4);//entropy
Px1x2x1x3 = Px_1*Px_2*Px_1*Px_3;//product of probabilities
Ix1x2x1x3 =-log2(Px1x2x1x3);//information of four symbols
Px4x3x3x2 = Px_4*Px_3*Px_3*Px_2;//product of probabilities
Ix4x3x3x2 = -log2(Px4x3x3x2);//information of four symbols

//results
printf("\n\ni.Entorpy = %.2f bits/symbol",H_X);
printf("\n\nii.Amount of information contained in x1x2x1x3 = %.2f bits/symbol",Ix1x2x1x3);
printf("\nThus,Ix1x2x1x3 < 7.4[=4*H_X] bits/symbol")
printf("\n\niii.Amount of information contained in x4x3x3x2 =%.2f bits/symbol",Ix4x3x3x2);
printf("\nThus we conclude that\nIx4x3x3x2 > 7.4[=4*H_X] bits/symbol")
```

Scilab code Exa 11.12 average rate of information convyed

```
//ANALOG AND DIGITAL COMMUNICATION
```

```
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.12(PAGENO 495)");
//given
n = 2*10^6//elements od black and white TV picture
m = 16//brightness levels of black and white TV
    picture
o = 32//repeated rate of pictures per second

//calculations
Px_i = 1/m//probability of  brightness levels of
    picture
H_X = 0;
for  i= 1:16
       H_Xi = (-1/(1/Px_i))*log2(1/(1/Px_i));
       H_X = H_X +H_Xi;
end
 r = n*o//rate of symbols generated
 R = r*H_X//average rate of information convyed

 //results
printf("\n\ni. Average rate of information convyed =
    %.2f bits/seconds",R)
```

Scilab code Exa 11.13 average information rate

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
```

```
printf("EXAMPLE 11.13(PAGENO 495)");
//given
t_dot = .2//duration of dot symbol
t_dash = .6//duration of dash symbol
t_space = .2//time between the symbols
//wkt sum of the probability is 1 i.e P_dot + P_dash = 1 hence
//P_dot = 2*P_dash weget
P_dot = 2/3//probality of dot symbol
P_dash = 1/3//probality of dash symbol

//calculations
H_X = -P_dot*log2(P_dot)-P_dash*log2(P_dash);//entropy
T_s = P_dot*t_dot + P_dash*t_dash +t_space;//average time per symbol
r = 1/T_s;//average symbol rate
R = r*H_X;//average information rate of the telegraph sourece

//result
printf("\n\ni.The average information rate of the telegraph source = %.4f bits/seconds",R);
```

Scilab code Exa 11.14 entropy and rate of information for given porbabilities

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.14(PAGENO 496)");
//given
```

```
//given symbols are equally likely all the symbols the probabilities are same
Px_1 = 1/8;//probability of first symbol
Px_2 = 1/8;//probability of second symbol
Px_3 = 3/8;//probability of third symbol
Px_4 = 3/8;//probability of fourth symbol
f_m = poly(0,"f_m");
r = 2//average symbol rate from problem 11.14

//calculaitons
H_X = Px_1*log2(1/Px_1) + Px_2*log2(1/Px_2) + Px_3*log2(1/Px_3) + Px_4*log2(1/Px_4);//entropy
R = H_X*r;//information rate

//results
printf("\n\ni.Entropy = %.2f bits/symbol",H_X)
printf("\n\nii.The information rate of all symbols = %.2f*f_m bits/seconds", R);
```

Scilab code Exa 11.15 information rate of given symbols probabilities

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.15(PAGENO 497)");
//given
//given symbols are equally likely all the symbols the probabilities are same
Px_1 = 1/4;//probability of first symbol
Px_2 = 1/4;//probability of second symbol
Px_3 = 1/4;//probability of third symbol
Px_4 = 1/4;//probability of fourth symbol
```

```
f_m = poly(0,"f_m");
r = 2//average symbol rate from problem 11.14

//calculaitons
H_X = Px_1*log2(1/Px_1) + Px_2*log2(1/Px_2) + Px_3*log2(1/Px_3) + Px_4*log2(1/Px_4);//entropy
R = H_X*r;//information rate

//results
printf("\n\ni.The information rate of all symbols = %.2f*f_m bits/seconds", R);
```

Scilab code Exa 11.16 entropy and rate of information of given symbols

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.16(PAGENO 498)");
//given
Px_1 = 1/2;//probability of first symbol
Px_2 = 1/4;//probability of second symbol
Px_3 = 1/8;//probability of third symbol
Px_4 = 1/16;//probability of fourth symbol
Px_4 = 1/16;//probability of fifth symbol
T_b = 1*10^-3//time required for emittion of each symbol
r = 1/(T_b)//symbol rate

//calculations
H_X = Px_1*log2(1/Px_1) + Px_2*log2(1/Px_2) + Px_3*log2(1/Px_3) + Px_4*log2(1/Px_4) + Px_4*log2(1/Px_4);
```

```
R = r*H_X;//information rate

//results
printf("\n\ni.Entropy of five symbols = %.2f bits/
   symbol",H_X);

printf("\n\nii.Rate of information = %.2f bits/sec",
   R);
```

Scilab code Exa 11.17 rate of information

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.17(PAGENO 498)");
//given
Px_1 = 1/2;//probability of first symbol
Px_2 = 1/4;//probability of second symbol
Px_3 = 1/8;//probability of third symbol
Px_4 = 1/16;//probability of fourth symbol
Px_5 = 1/16;//probability of fifth symbol
r = 16//outcomes per second

//calculations
H_X = Px_1*log2(1/Px_1) + Px_2*log2(1/Px_2) + Px_3*
   log2(1/Px_3) + Px_4*log2(1/Px_4) + Px_5*log2(1/
   Px_5);
R = r*H_X;//information rate

//result
printf("\n\nRate of information = %.2f bits/sec",R);
```

Scilab code Exa 11.18 rate of information

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.18(PAGENO 499)");
//given
Px_1 = 1/4;//probability of first symbol
Px_2 = 1/5;//probability of second symbol
Px_3 = 1/5;//probability of third symbol
Px_4 = 1/10;//probability of fourth symbol
Px_5 = 1/10;//probability of fifth symbol
Px_6 = 1/20;//probability of sixth symbol
Px_7 = 1/20;//probability of seventh symbol
Px_8 = 1/20;//probability of eigith symbol
f_m = 10*10^3//freuency of tranamitting symbol

//calculations
H_X = Px_1*log2(1/Px_1) + Px_2*log2(1/Px_2) + Px_3*
    log2(1/Px_3) + Px_4*log2(1/Px_4) + Px_5*log2(1/
    Px_5) + Px_6*log2(1/Px_6)+ Px_7*log2(1/Px_7)+
    Px_8*log2(1/Px_8);//entropy
f_s = 2*f_m//sampling frequency
r = f_s//sampling frequency equal to rate of
    transmission
R = r*H_X;//information rate

//result
printf("\n\nRate of information = %.2f bits/sec",R);
printf("\n\nNote:Their mistake in calculation of H_X
     in textbook")
```

Scilab code Exa 11.19 channel matrix and joint probabilities

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.19(PAGENO 502)");
//given
//from fig
P_X = [.5 .5]//x matrix
P_Xd = [.5 0; 0 .5]//diagonal x matrix
//calculations
P_YX = [.9 .1; .2 .8];//yx matrix representation of given fig
P_Y = P_X*P_YX//y matrix
P_XY = P_Xd * P_YX//xy  matrix

//results
printf("\n\ni.Channel matrix of the channelP_YX ");
disp(P_YX);
printf("\n\nii.a.P(y1) = %.2f",P_Y(1,1));
printf("\n\n    b.P(y2) = %.2f",P_Y(1,2));
printf("\n\niii.a.P(x1,y2) = %.2f",P_XY(1,2));
printf("\n\n     b.P(x2,y1) = %.2f",P_XY(2,1));
```

Scilab code Exa 11.20 channel matrix and probabilities

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
```

```
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.20(PAGENO 503)");
//given
P_X = [.5 .5]//x matrix

//calculations
P_YX = [.9 .1; .2 .8];//yx matrix representation of
    given fig
P_ZY = [.9 .1; .2 .8]//zy matrix representation of
    given fig
P_Y = P_X *P_YX//y matrix
P_ZX = P_YX * P_ZY//zx  matrix
P_Z = P_X *P_ZX//z matrix


//results
printf("\n\ni.Channel matrix of the channelP_ZX ");
disp(P_ZX);
printf("Matrix P(Z)")
disp(P_Z);
printf("\n\na.P(Z1) = %.2f",P_Z(1,1));
printf("\n\nb.P(Z2) = %.2f",P_Z(1,2));
```

Scilab code Exa 11.21 probabilities associated with the channel outputs

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.21(PAGENO 504)");
//given
```

```
P_X = [.5 .5]//x matrix
P_YX = [.8 .2 0 ; 0 .2 .8];//yx matrix representation of given fig

//calculations
P_Y = P_X*P_YX;

//results
printf("\n\norobability associated with the channel outputs for p=.2 is")
disp(P_Y)
printf("\n\na.P(Y1) = %.2f",P_Y(1,1));
printf("\n\nb.P(Y2) = %.2f",P_Y(1,2));
printf("\n\nC.P(Y3) = %.2f",P_Y(1,3));
```

Scilab code Exa 11.28 mutual information

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.21(PAGENO 504)");

//given
//wkt P_Y = P_X*P_YX from previous problems
alfa = .5
P_1 = .1//probability for first case
P_2 = .5//probability for second case

//calculations
P_X = [alfa alfa];
//first case
P_YX = [1-P_1 P_1;P_1 1-P_1];
```

```
P_Y1 = P_X*P_YX;
H_Y1 = -P_Y1(1,1)*log2(P_Y1(1,1))-P_Y1(1,2)*log2(P_Y1(1,2));
Q_1 = P_1*log2(P_1)  + (1-P_1)*log2(1-P_1)//from proof
 I_XY1 =  1 + Q_1;
//second case
P_YX = [1-P_2 P_2;P_2 1-P_2];
P_Y2 = P_X*P_YX;
H_Y2 = -P_Y2(1,1)*log2(P_Y2(1,1))-P_Y2(1,2)*log2(P_Y2(1,2));
Q_2 = P_2*log2(P_2)  + (1-P_2)*log2(1-P_2)//from proof
I_XY2 =  1 + Q_2;

//results
printf("\n\nI_XY for the first case = %.2f",I_XY1);
printf("\n\nI_XY for the second case = %.2f",I_XY2);
```

Scilab code Exa 11.32 entropy for the given probability density function

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.32(PAGENO 518)");
//given
a1 = 1
a2 = 2
a3 = .5

//calculations
H_X1 = log2(a1);//Entropy for first case
```

```
H_X2 = log2(a2);//Entropy for second case
H_X3 = log2(a3);//Entropy for third case

//results
printf("\n\ni.Entropy for first case = %.2f ",H_X1);
printf("\n\nii.Entropy for second case = %.2f",H_X2)
    ;
printf("\n\niii.Entropy for third case = %.2f ",H_X3
    );
```

Scilab code Exa 11.35 capacity of the channel

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.35(PAGENO 520)");
//given
B = 4000//bandwidth of AWGN channel
S = .1*10^-3//power of signal
neta = 2*10^-12//spectral dencity
N = neta*B;//power

//calculations
C = B * log2(1 + (S/N));//capacity of channel

//result
printf("\n\nCapacity of channel = %.2f b/s",C);
```

Scilab code Exa 11.37 efficiency of the code and code redundancy

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.37(PAGENO 524)");
//given
Px_1 = 0.9//probability of first symbol
Px_2 = 0.1//probability of second symbol
n1 = 1//length of the code for x_1
n2 =1//length of code for x_2

//calculations
//we know that the average code length L per symbol
L = Px_1*n1 + Px_2*n2//code length
H_X = -Px_1*log2(Px_1) - Px_2*log2(Px_2) //entropy
neta = H_X/L//efficiency
neta1 = neta*100//neta in percentage
gama = 1 - neta//redundancy
gama1 = gama*100//gama in percentage

//results
printf("\n\ni.Efficiency of code = %.2f percent",
   neta1);
printf("\n\nii.Code redundancy = %.2f percent ",
   gama1)
```

Scilab code Exa 11.38 efficiency of the code and code redundancy

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
```

```
clc;
printf("EXAMPLE 11.38(PAGENO 524)");

//given
Px_1 = 0.81//probability of first symbol
Px_2 = .09//probability of second symbol
Px_3 = .09//probability of third symbol
Px_4 = 0.01//probability of forth symbol
n1 = 1//length of code for a_1
n2 =2//length of code for a_2
n3 = 3//length of code for a_3
n4 = 3//length of code for a_4

//calculations
//we know that the average code length L per symbol
L = Px_1*n1 + Px_2*n2 + Px_3*n3 + Px_4*n4 //code
    length
H_X = -Px_1*log2(Px_1) - Px_2*log2(Px_2) - Px_3*log2
    (Px_3) - Px_4*log2(Px_4)//entropy
neta = H_X/L//efficiency
neta1 = neta*100//neta in percentage
gama = 1 - neta//redundancy
gama1 = gama*100//gama in percentage

//results
printf("\n\ni.Efficiency of code = %.2f percent",
    neta1);
printf("\n\nii.Code redundancy = %.2f percent ",
    gama1)
```

Scilab code Exa 11.44 efficiency of shannon fano code

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
```

```
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.44(PAGENO 529)");

//given
P_x1 = 1/2//probability of first symbol
P_x2 = 1/4//probability of second symbol
P_x3 = 1/8//probability of third symbol
P_x4 = 1/8//probability of fouth symbol
n1 = 1
n2 = 2
n3 = 3
n4 = 3

//calculations
I_x1 = -log2(P_x1);
I_x2 = -log2(P_x2);
I_x3 = -log2(P_x3);
I_x4 = -log2(P_x4);
H_x = P_x1*I_x1 +  P_x2*I_x2 +  P_x3*I_x3 +  P_x4*I_x4;
L = P_x1*n1 + P_x2*n2 + P_x3*n3 + P_x4*n4;
neta = H_x/L;
P_neta = neta*100//efficiency in percentage

//results
printf("\n\nEfficiency = %.2f",neta);
printf("\n\nEfficiency in percentage = %.2f percent",P_neta);
```

Scilab code Exa 11.46 efficiency of shannon fano code and huffman code

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
```

```
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.46(PAGENO 532)");

//given
P_x1 = .4//probability of first signal
P_x2 = .19//probability of second signal
P_x3 = .16//probability of third signal
P_x4 = .15//probability of fourth signal
P_x5 = .1//probability of fifth signal
n1 = 1//number of bits in code obtained from table givenn textbook
n2 = 2//number of bits in code obtained from table givenn textbook
n3 = 2//number of bits in code obtained from table givenn textbook
n4 = 3//number of bits in code obtained from table givenn textbook
n5 = 3//number of bits in code obtained from table givenn textbook

//calculations
I_x1 = -log2(P_x1);
I_x2 = -log2(P_x2);
I_x3 = -log2(P_x3);
I_x4 = -log2(P_x4);
I_x5 = -log2(P_x5);
H_x = P_x1*I_x1 +  P_x2*I_x2 +  P_x3*I_x3 +  P_x4*I_x4 +  P_x5*I_x5;//entropy
L1 = P_x1*n1 + P_x2*n2 + P_x3*n3 + P_x4*n4 + P_x5*n5;
neta1 = H_x/L1;
P_neta1 = neta1*100//efficiency in percentage using Shannon Fano code
L2 = P_x1*1 + (P_x2 + P_x3 +P_x4 +P_x5 )*3
neta2 = H_x/L2;
```

```
P_neta2 = neta2*100//efficiency in percentage using
    huffman code

//results
printf("\n\nEfficiency in percentage using Shannon
    Fano code = %2f percent",P_neta1)
printf("\n\nEfficiency in percentage using huffman
    code   = %2f percent",P_neta2)
printf("\n\nNote: There is mistake in the textbook
    in calculation of L using SHannon Fano code")
```

Scilab code Exa 11.47 efficiency of huffman code

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.44(PAGENO 532)");

//given
P_x1 = .05//probability of first signal
P_x2 = .15//probability of second signal
P_x3 = .2//probability of third signal
P_x4 = .05//probability of fourth signal
P_x5 = .15//probability of fifth signal
P_x6 = .3//probability of sixth signal
P_x7 = .1//probability of seventh signal
n1 = 4//number of bits in code obtained from table
    given textbook
n2 = 3//number of bits in code obtained from table
    given textbook
n3 = 2//number of bits in code obtained from table
    given textbook
```

```
n4 = 4//number of bits in code obtained from table given textbook
n5 = 3//number of bits in code obtained from table given textbook
n6 = 2//number of bits in code obtained from table given textbook
n7 = 3//number of bits in code obtained from table given textbook

//calculations
I_x1 = -log2(P_x1);
I_x2 = -log2(P_x2);
I_x3 = -log2(P_x3);
I_x4 = -log2(P_x4);
I_x5 = -log2(P_x5);
I_x6 = -log2(P_x6);
I_x7 = -log2(P_x7);
H_x = P_x1*I_x1 +  P_x2*I_x2 +  P_x3*I_x3 +  P_x4*I_x4 +  P_x5*I_x5 + P_x6*I_x6 + P_x7*I_x7;//entropy
L = P_x1*n1 + P_x2*n2 + P_x3*n3 + P_x4*n4 + P_x5*n5 + P_x6*n6 + P_x7*n7;
neta = (H_x*100)/L//Efficiency in percentage

//results
printf("\n\nEfficiency in percentage = %.2f percent",neta);
```

Scilab code Exa 11.49 variance of codeword length

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
```

```
clc;
printf("EXAMPLE 11.49(PAGENO 534)");

//given
P_x1 = .4//probability of first signal
P_x2 = .2//probability of second signal
P_x3 = .8//probability of third signal
P_x4 = .08//probability of fourth signal
P_x5 = .02//probability of fifth signal
n1 = 2//number of bits in code obtained from table given textbook
n2 = 3//number of bits in code obtained from table given textbook
n3 = 1//number of bits in code obtained from table given textbook
n4 = 4//number of bits in code obtained from table given textbook
n5 = 4//number of bits in code obtained from table given textbook

//calculations
L = P_x1*n1 + P_x2*n2 + P_x3*n3 + P_x4*n4 + P_x5*n5; //average codeword length per symbol
//since sigma = sqrt(summation of product of probability and (n- L)^2)
sigmasquare = P_x1*(n1-L)^2 + P_x2*(n2-L)^2 +P_x3*(n3-L)^2 + P_x4*(n4-L)^2 +P_x5*(n5-L)^2;//Variance of codewoed length

//results
printf("\n\nVariance of codeword length =%.4f",sigmasquare)
```

Scilab code Exa 11.50 entropy of the system and information rate

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.50(PAGENO 535)");

//given
P_x1 = 1/2//probability of first signal
P_x2 = 1/4//probability of second signal
P_x3 = 1/8//probability of third signal
P_x4 = 1/16//probability of fourth signal
P_x5 = 1/32//probability of fifth signal
P_x6 = 1/32//probability of sixth signal
r = 16//message rate in outcomes per second

//calculations
I_x1 = -log2(P_x1);
I_x2 = -log2(P_x2);
I_x3 = -log2(P_x3);
I_x4 = -log2(P_x4);
I_x5 = -log2(P_x5);
I_x6 = -log2(P_x6);
H_X = P_x1*I_x1 +  P_x2*I_x2 +  P_x3*I_x3 +  P_x4*
    I_x4 +  P_x5*I_x5 + P_x6*I_x6 //entropy
R = H_X*r//Information rate

//results
printf("\n\nEntropy of the system =%.2f bits/message
    ",H_X);
printf("\n\nInformation rate = %.2f bits/seconds",R)
    ;
```

Scilab code Exa 11.51 entropy

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.51(PAGENO 535)");

//given
P_x1 = .3//probability of first signal
P_x2 = .4//probability of second signal
P_x3 = .3//probability of third signal
P_YX = [.8 .2 0;0 .1 0; 0 .3 0.7]//matrix obtained
   from the figure

//calculations
I_x1 = -log2(P_x1);
I_x2 = -log2(P_x2);
I_x3 = -log2(P_x3);
H_X = P_x1*I_x1 +  P_x2*I_x2 +  P_x3*I_x3 //entropy
P_y1 = P_YX(1,1)*P_x1 + P_YX(1,2)*P_x1 + P_YX(1,3)*
   P_x1;
P_y2 = P_YX(2,1)*P_x2 + P_YX(2,2)*P_x2 + P_YX(2,3)*
   P_x2;
P_y3 = P_YX(3,1)*P_x3 + P_YX(3,2)*P_x3 + P_YX(3,3)*
   P_x3;
I_y1 = -log2(P_y1);
I_y2 = -log2(P_y2);
I_y3 = -log2(P_y3);
H_Y = -P_y1*I_y1 -  P_y2*I_y2 -  P_y3*I_y3 //entropy

//results
printf("\n\n Entropy H(X) = %.2f",H_X );
printf("\n\nEntropy H(Y) = %.2f",H_Y);
printf("\n\n Note:There is mistake in the
   calculation of P_y3 in the textbook so their is
   change in entropy H_Y")
```

Scilab code Exa 11.52 entropy of the second order extension

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.52(PAGENO 536)");

//given
P_x1 = .7//probability of first signal
P_x2 = .15//probability of second signal
P_x3 = .15//probability of third signal
n = 2//second order extention

//calculations
I_x1 = -log2(P_x1);
I_x2 = -log2(P_x2);
I_x3 = -log2(P_x3);
H_x = P_x1*I_x1 +  P_x2*I_x2 +  P_x3*I_x3//entropy
H_x2 = n*H_x//entropy of second order extention

//results
printf("\n\nEntropy of second order extention = %.3f
    bits/symbol",H_x2);
```

Scilab code Exa 11.54 entropy of the source

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
```

```
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.54(PAGENO 537)");

//given
P_x1 = 1/3//probability of first signal
P_x2 = 1/6//probability of second signal
P_x3 = 1/4//probability of third signal
P_x4 = 1/4//probability of fourth signal

//calculations
I_x1 = -log2(P_x1);
I_x2 = -log2(P_x2);
I_x3 = -log2(P_x3);
I_x4 = -log2(P_x4);
H_x = P_x1*I_x1 +  P_x2*I_x2 +  P_x3*I_x3 +  P_x4*I_x4 //entropy

//results
printf("\n\nEntropy of the source = %.5f bits/symbol",H_x)
```

Scilab code Exa 11.55 average number of bits

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.55(PAGENO 538)");

//given
P_x1 = 1/2//probability of first signal
```

```
P_x2 = 1/4//probability of second signal
P_x3 = 1/8//probability of third signal
P_x4 = 1/16//probability of fourth signal
P_x5 = 1/16//probability of fifth signal
n1 = 1//number of bits in code obtained from table given textbook
n2 = 2//number of bits in code obtained from table given textbook
n3 = 3//number of bits in code obtained from table given textbook
n4 = 4//number of bits in code obtained from table given textbook
n5 = 4//number of bits in code obtained from table given textbook

//calculations
L = P_x1*n1 + P_x2*n2 + P_x3*n3 + P_x4*n4 + P_x5*n5; //Average number of bits per message

//results
printf("\n\nAverage number of bits per message = %.2f bits",L);
```

Scilab code Exa 11.56 information capacity of the telephone channel

```
//ANALOG AND DIGITAL COMMUNICATION
//BY Dr.SANJAY SHARMA
//CHAPTER 11
//Information Theory
clear all;
clc;
printf("EXAMPLE 11.56(PAGENO 538)");

//given
B = 3.4*10^3//bandwidth
```

```
SbyN = 30//signal to the noise ratio in dB


//calculations
SbyN1 = exp((SbyN/10)*log(10))//signal to noise
    ratio
C = B*log2(1+SbyN1);

//result
printf("\n\nInformation capacity of the telephone
    channel  = %.2f kbps",C);
```