

Scilab Textbook Companion for
Analog and Digital Communication
by S. Sharma¹

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

July 31, 2019

¹Funded 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	4
2 Amplitude Modulation	5
3 Radio Transmitters	23
4 Radio Receivers	27
5 Angle Modulation	33
6 Noise	76
7 Sampling Theory and Pulse Modulation	92
8 Waveform Coding Techniques	97
10 Digital multiplexers	112
11 Information Theory	117

List of Scilab Codes

Exa 2.1	frequency range of sidebands	5
Exa 2.2	Bandwidth of modulated signal	6
Exa 2.3	total power in amplitude modulated wave .	7
Exa 2.4	Carrier Power	7
Exa 2.5	antenna current and percentage modulation	8
Exa 2.6	carrier current and modulation of signal and modulation index of second signal	9
Exa 2.7	modulating volatage and minimum and max- imum voltage	10
Exa 2.9	percentage modulation index	10
Exa 2.10	percentage modulation index of second wave	11
Exa 2.11	total radiated power	12
Exa 2.12	total radiated power	13
Exa 2.13	percentage power saving	14
Exa 2.15	percentage power saving	15
Exa 2.16	power content of carrier and upper and lower sidebands	16
Exa 2.17	percentage modulation index	16
Exa 2.18	percentage modulation index	17
Exa 2.21	carrier power and transmission efficiency and carrier amplitude	18
Exa 2.29	modulation depth	19
Exa 2.31	percentage power saving	20
Exa 2.32	carrier frequency	21
Exa 3.1	total radiated power	23
Exa 3.2	carrier frequency and frequency deviation .	24
Exa 3.3	appropriate multiplexer values and oscillator frequency	25

Exa 4.1	image frequency and image rejection ratio .	27
Exa 4.2	image frequency and image rejection ratio .	28
Exa 4.3	quality factor and new intermediate frequency	29
Exa 4.5	oscillator frequency and image frequency and image rejection ratio	30
Exa 4.6	image frequency and image rejection ratio and intermediate frequency	31
Exa 5.1	carrier frequency and modulating frequency and modulation index and maximum devia- tion and power dissipated	33
Exa 5.2	carrier frequency and highest and lowest fre- quency and modulation index	34
Exa 5.3	frequency deviation and carrier swing and low- est frequency of modulated wave	35
Exa 5.4	modulation index	36
Exa 5.5	percentage modulation index	36
Exa 5.6	bandwidth of FM transmission	37
Exa 5.7	bandwidth of narrowband FM signal	38
Exa 5.8	bandwidth of a signal when modulating am- plitude is doubled	38
Exa 5.9	modulation index and bandwidth of FM signal	39
Exa 5.10	modulating frequency	40
Exa 5.11	modulation index and bandwidth of FM and PM signal	40
Exa 5.12	carrier frequency and modulating frequency and modulation index and maximum devia- tion and power dissipated	42
Exa 5.13	frequency deviation for given modulating volatge	43
Exa 5.14	modulation index	44
Exa 5.15	bandwidth of FM and DSB FC signal	45
Exa 5.16	representation of FM and PM signal or wave	46
Exa 5.17	carrier amplitude and amplitude of sidebands and minimum and maximum frequency of side- bands	47
Exa 5.18	maximum frequency devaiation and modula- tion index	48
Exa 5.19	modulation index and bandwidth of a signal	49
Exa 5.20	capacitive reactance	51

Exa 5.21	carrier frequency and frequency deviation and modulation index and minimum and maximum frequency at a given point	51
Exa 5.22	modulation index for minimum and maximum modulating frequency of FM signal	53
Exa 5.23	modulation index and equation for carrier waveform	54
Exa 5.24	frequency sensitivity and modulation index .	55
Exa 5.29	bandwidth of FM signal using carsons rule .	56
Exa 5.30	fracation of signal power included in the frequency band	57
Exa 5.31	bandwidth of FM signal	58
Exa 5.32	Carrier Power and power in each sideband .	59
Exa 5.33	carrier swing and modulation index and highest and lowest frequency attained by FM signal	60
Exa 5.34	frequency deviation and carrier frequency and lower frequency reached by modulated FM wave	60
Exa 5.35	percentage modulation index	61
Exa 5.36	frequency deviation and modulation index .	62
Exa 5.37	carrier frequency and modulating frequency and modulation index and maximum frequency deviation and power dissipated in FM wave	63
Exa 5.39	maximum frequency deviation	64
Exa 5.43	bandwidth of angle modulated signal	65
Exa 5.44	modulation index and bandwidth of FM signal	66
Exa 5.45	bandwidth of angle modulated signal	67
Exa 5.46	bandwidth of FM signal	68
Exa 5.47	bandwidth and frequency deviation of FM signal	69
Exa 5.48	bandwidth calculation using the given formula and carsons rule	70
Exa 5.49	frequency multiplication and maximum frequency deviation	71
Exa 5.50	maximum frequency deviation and carrier frequency for given maximum and minimum frequencies	72

Exa 5.51	multiplier and mixer oscillating frequency for Armstrong type FM generator	73
Exa 5.52	frequency multiplication	74
Exa 6.1	rms noise voltage	76
Exa 6.2	rms noise voltage	77
Exa 6.3	rms voltage due individual resistance and resistances in parallel and series	77
Exa 6.4	equivalent input noise resistance	78
Exa 6.7	output voltage across the circuit measured by a wide band voltmeter	79
Exa 6.8	noise figure and equivalent temperature	80
Exa 6.9	noise figure	81
Exa 6.10	overall noise figure	82
Exa 6.11	overall noise figure	83
Exa 6.12	rms noise voltage	83
Exa 6.14	meter reading in volts and resistance at given temperature	84
Exa 6.16	overall noise figure of three stage cascade amplifier	85
Exa 6.17	overall noise figure of two stage cascaded amplifier	86
Exa 6.18	overall noise figure and overall gain	87
Exa 6.19	overall noise figure	88
Exa 6.20	noise temperature of the receiver and overall noise temperature	89
Exa 6.21	overall noise temperature of the receiver and overall noise figure	90
Exa 7.1	Nyquist rate	92
Exa 7.2	Nyquist rate and Nyquist interval	93
Exa 7.3	discrete time signal for the given conditions	94
Exa 7.4	Nyquist rate for the continuous signal	95
Exa 8.2	code word length and transmission bandwidth and final bit rate and quantization noise ratio	97
Exa 8.3	number of bits required and bandwidth of pcms and signalling rate	98
Exa 8.4	sampling rate and number of bits in each PCM and bit rate and transmission bandwidth	99
Exa 8.5	transmission bandwidth and sampling frequency	100

Exa 8.6	message bandwidth and signal to noise ratio	101
Exa 8.7	number of bits in codeword and sampling rate and bit rate	102
Exa 8.8	signal to the noise ratio	103
Exa 8.10	signal to the noise ratio and number of bits	104
Exa 8.11	maximum frequency	104
Exa 8.13	maximum amplitude	105
Exa 8.14	signalling rate	106
Exa 8.15	signal to the noise ratio of linear delta mod- ulation system	107
Exa 8.16	signal to the noise ratio	107
Exa 8.17	sampling frequency and number of binary dig- its and quantizing level	108
Exa 8.20	number of required levels and signal to noise ratio	109
Exa 8.21	quantizing levels and number of bits and band- width	110
Exa 10.1	sampling rate	112
Exa 10.2	signalling rate and channel bandwidth . . .	113
Exa 10.3	spacing between successive pulses of multi- plexed signal	114
Exa 10.4	signaling rate and channel bandwidth	115
Exa 11.1	information content of given symbols	117
Exa 11.2	amount of information for a given probability of a symbol	118
Exa 11.3	amount of information content wrt to binary PCMs	119
Exa 11.4	amount of information content wrt to binary PCMs	119
Exa 11.9	entropy and amount of information for the given messages	120
Exa 11.12	average rate of information conveyed	121
Exa 11.13	average information rate	122
Exa 11.14	entropy and rate of information for given proba- bilities	123
Exa 11.15	information rate of given symbols probabilities	124
Exa 11.16	entropy and rate of information of given sym- bols	125

Exa 11.17	rate of information	126
Exa 11.18	rate of information	126
Exa 11.19	channel matrix and joint probabilities	127
Exa 11.20	channel matrix and probabilities	128
Exa 11.21	probabilities associated with the channel outputs	129
Exa 11.28	mutual information	130
Exa 11.32	entropy for the given probability density function	131
Exa 11.35	capacity of the channel	132
Exa 11.37	efficiency of the code and code redundancy	132
Exa 11.38	efficiency of the code and code redundancy	133
Exa 11.44	efficiency of shannon fano code	134
Exa 11.46	efficiency of shannon fano code and huffman code	135
Exa 11.47	efficiency of huffman code	137
Exa 11.49	variance of codeword length	138
Exa 11.50	entropy of the system and information rate	139
Exa 11.51	entropy	140
Exa 11.52	entropy of the second order extension	141
Exa 11.54	entropy of the source	142
Exa 11.55	average number of bits	143
Exa 11.56	information capacity of the telephone channel	144

Chapter 2

Amplitude Modulation

Scilab code Exa 2.1 frequency range of sidebands

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.1(PAGENO 51)");
8 //given
9 L = 50*10^-6//in henry
10 C = 1*10^-9//in farads
11 //calculation
12 F_c = 1/(2*%pi*sqrt(L*C));
13 //results
14 printf("\n\nCarrier frequency F_c = %.2f Hz",F_c);
15 printf("\n\nNow , it is given that the highest
    modulation frequency is 8KHz ");
16 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

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.2(PAGENO 51)");
8
9 //given
10 //v_m = 10*sin(2*%pi*10^3*t)
11 //by comparing with v_m = V_m*sin(2*%pi*f_c*t) we
    get
12 V_m = 10//in volts
13 f_m = 1*10^3//in hertz
14 V_c = 20//in volts
15 f_c = 1*10^4//in hertz
16
17 //calculations
18 m_a = V_m/V_c;//modulation index formula
19 m_a1 = m_a*100;//percentage modulation index
20 f_usb = f_c + f_m;//Upper sideband
21 f_lsb = f_c - f_m;//lower sideband
22 A = (m_a*V_c)/2//amplitude of upper as well as lower
    sideband
23 B = 2*f_m;//bandwidth of the modulation signal
24
25 //results
26 printf("\n\ni.a.Modulation index=%0.2f",m_a);
27 printf("\n\n    b.Percentage modulation index=%0.2f
    percent",m_a1);
28 printf("\n\niii.a.Upper sidebandfrequency=%f Hz",
    f_usb);
```

```

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

```

Scilab code Exa 2.3 total power in amplitude modulated wave

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.3(PAGENO 54)");
8
9 //given
10 m_a = .75;//modulation index
11 P_c = 400;//carrier power in watts
12
13 //calculation
14 P_t = P_c*(1+(m_a^2/2));//total power
15
16 //results
17 printf("\n\nTotal power in the amplitude modulated
    wave=%.2 f W",P_t);

```

Scilab code Exa 2.4 Carrier Power

```

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

```

```

4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.4(PAGENO 54)");
8 //given
9 P_t = 10*10^3; //total power in watts
10 m_a = .6; //modulation index
11 //calculation
12 P_c = (P_t/(1+(m_a^2/2))); // carrier power
13 //results
14 printf("\n\nCarrier power=%0.2f W",P_c);

```

Scilab code Exa 2.5 antenna current and percentage modulation

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.5(PAGENO 55)");
8
9 //given
10 I_t = 8.93; //total modulated current in ampers
11 I_c= 8; //carrier or unmodulated current in ampers
12 //calculation
13 m_a = sqrt(2*((I_t/I_c)^2 -1)); //formula for
    modulation index
14 M_a=m_a*100; //percentage modulation
15 //for
16 m_a1 = .8; //given modulation index
17
18 //calculation
19 I_t1 = I_c*sqrt(1+(m_a1^2/2)); //new antenna current
20

```

```

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

```

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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.6(PAGENO 56)");
8
9 //given
10 I_t1 = 10//antenna current in amps
11 m1 = .3//modulation index
12 I_t2 = 11//increased antenna current
13
14 //calculation
15 I_c = (I_t1/(1+(m1^2/2))^.5);//formula for carrier
    signal current
16 m_t = sqrt(2*((I_t2/I_c)^2 -1));//formula for
    modulation index
17 m2 = sqrt(m_t^2 - m1^2);
18 m3 = m2*100;//percentage modulation index
19
20 //results
21 printf("\n\ni. Carrier signal current = %.2f A",I_c);
22 printf("\n\niii.Modulation index of signal = %.2f",
    m_t);
23 printf("\n\niiii.a.Modulation index of second signal
    = %.2f",m2);

```

```
24 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

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.7(PAGENO 56)");
8
9 //given v_c = 10*sinwt
10
11 m = .5//modulation index
12 //by comparing with v_c = V_c*sinwt
13 V_c = 10//carrier voltage in volts
14
15 //calculation
16 V_m = m*V_c;//amplitude of modulating index
17 V_max = V_c + V_m;//maximum voltage
18 V_min = V_c - V_m;//minimum voltage
19
20 //results
21 printf("\n\n i. Modulating voltage = %.2f V",V_m);
22 printf("\n\n ii. Maximum voltage = %.2f V",V_max);
23 printf("\n\n iii. Minimum voltage = %.2f V",V_min);
```

Scilab code Exa 2.9 percentage modulation index

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



```

3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.9(PAGENO 57)");
8
9 //given
10 V_max = 4//maximum voltage in volts
11 V_min = 1//minimum voltage in volts
12
13 //calculation
14 m = (V_max - V_min)/(V_max + V_min) ;//formula for
    modulation index
15 m1 = m*100//percentage modulation index
16
17 //result
18 printf("\n\n Percentage modulation index = %.2f
    percent",m1);

```

Scilab code Exa 2.10 percentage modulation index of second wave

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.10(PAGENO 57)");
8
9 //given
10 m1 = .4//modulation index
11 I_t1 = 11//initial antenna current in ampers
12 I_t2 = 12//final antenna current in ampers
13
14 //calculations

```

```

15 I_c = (I_t1/(1+(m1^2/2))^.5); // formula for carrier
    current in ampers
16 m_t = sqrt(2*((I_t2/I_c)^2 -1)); //total modulation
    index
17 m2 = sqrt(m_t^2 - m1^2); //modulation index to the
    second wave
18 m3 = m2*100; //percentage modulation index to the
    second wave
19
20 //results
21 printf("\n\n Carrier current = %.2f A",I_c);
22 printf("\n\n Total modulation index = %.2f",m_t);
23 printf("\n\n Percentage modulation index of second
    wave= %.2f percent",m3);

```

Scilab code Exa 2.11 total radiated power

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.11(PAGENO 58)");
8
9 //given
10 P_c = 10*10^3 //carrier power in watts
11 P_t = 12*10^3 //total power in watts
12 m_2 = .5 //modulation index of second wave
13
14 //calculations
15 m_1 = sqrt(2*((P_t/P_c)-1)); //modulation index of
    first wave
16 m_t = sqrt(m_1^2 +m_2^2); //total modulation index
17 P_t1 = P_c*(1+(m_t^2/2)) //total new transmitted

```

```

    power
18
19 //results
20 printf("\n\nModulation index of first wave = %.4f",
    m_1);
21 printf("\n\nTotal modulation index = %.2f",m_t);
22 printf("\n\ntotal new transmitted power = %.2f W",
    P_t1);

```

Scilab code Exa 2.12 total radiated power

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.12(PAGENO 60)");
8
9 //given
10 P_t = 10.125*10^3//modulated or total power in watts
11 P_c = 9*10^3//unmodulated of carrier power
12 m_2 = .4//modulation index of second wave
13
14 //calculations
15 m_1 = sqrt(2*((P_t/P_c) - 1))//modulation index of
    first wave
16 m_a = m_1*100//percentage modulation index of first
    wave
17 m_t = sqrt(m_1^2 + m_2^2)//total modulation index
18 P_t1 = P_c*(1+(m_t^2/2))//total radiated power
19
20 //results
21 printf("\n\ni.a.Modulation index of first wave = %.4
    f",m_1);

```

```

22 printf("\n\n b.Percentage modulation index of first
    wave = %.2f percent",m_a);
23 printf("\n\niii.Total radiated power = %.2f W",P_t1);

```

Scilab code Exa 2.13 percentage power saving

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.13(PAGENO 90)");
8
9 //given
10 m1 = 1//modulation index of first signal
11 m2 = .5//modulation index of second signal
12 //let
13 P_c = 1//carrier power in watts
14
15 //calculations
16 P_1= P_c*(1+(m1^2/2))//total power of first signal
17 P_2 = P_c*(1+(m2^2/2))//total power of second signal
18 P_a = (P_c*100)/(P_1)//percentage power saving for
    first signal
19 P_b = (P_c*100)/(P_2)//percentage power saving for
    second signal
20
21 //results
22 printf("\n\ni.Percentage power saving for first
    signal= %f percent",P_a);
23 printf("\n\niii.Percentage power saving for second
    signal= %f percent",P_b);

```

Scilab code Exa 2.15 percentage power saving

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.15(PAGENO98 )");
8
9 //given
10 m1 = 1//modulation index of first signal
11 m2 = .5//modulation index of second signal
12 //let
13 P_c = 1//carrier power in watts
14
15 //calculations
16 P_cssb1 = P_c*(1+(m1^2/4))//power in carrier plus
    power in one sideband for first signal
17 P_cssb2 = P_c*(1+(m2^2/4))//power in carrier plus
    power in one sideband for second signal
18 P_1= P_c*(1+(m1^2/2))//total power of first signal
19 P_2 = P_c*(1+(m2^2/2))//total power of second signal
20 P_a = (P_cssb1*100)/(P_1)//percentage power saving
    for first signal
21 P_b = (P_cssb2*100)/(P_2)//percentage power saving
    for second signal
22
23 //results
24 printf("\n\ni.Percentage power saving for first
    signal= %f percent",P_a);
25 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

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.16(PAGENO 109)");
8
9 //given
10 P_ssb = 10*10^3//power in ssb transmission in
    watts
11 P_t = P_ssb// total power in watts
12 m_a = .8//modulation index
13
14 //calculations
15 P_c = (P_t/(1+(m_a^2/4)+(m_a^2/4)))//carrier power
    in watts
16 P_SB = P_t - P_c//power in sidebands
17 P_usb = P_SB/2//power in upper sideband
18 P_lsb =P_usb//power in upper sideband
19
20 //results
21 printf("\n\ni.Power content of the carrier = %.2f W"
    ,P_c);
22 printf("\n\nii.a.Power content in upper sideband =
    %.2f W",P_usb);
23 printf("\n\n b.Power content in lower sideband = %
    .2f W",P_lsb);
```

Scilab code Exa 2.17 percentage modulation index

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.17(PAGENO 109)");
8
9 //given from the figure
10 P_maxpp = 2*80 //maximum peak to peak power in watts
11 P_minpp = 2*20 //minimum peak to peak power in watts
12
13 //calculations
14 m_a = (P_maxpp - P_minpp)/(P_maxpp + P_minpp) //
      modulation index
15 M = m_a*100 //percentage modulation index
16
17 //results
18 printf("\n\ni.Modulation index =%.2f",m_a);
19 printf("\n\nii.Percentage modulation index = %.2f
      percent",M);

```

Scilab code Exa 2.18 percentage modulation index

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.18(PAGENO 110)");
8
9 //given from the figure
10 P_maxpp = 2*50 //maximum peak to peak power in watts
11 P_minpp = 2*15 //minimum peak to peak power in watts

```

```

12
13 //calculations
14 m_a = (P_maxpp - P_minpp)/(P_maxpp + P_minpp)//
    modultaion index
15 M = m_a*100//percentage modulation index
16
17 //results
18 printf("\n\ni.Modulation index =%.4f",m_a);
19 printf("\n\nii.Percentage modulation index = %.2f
    percent",M)

```

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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.21(PAGENO 111)");
8
9 //given
10 p_t = 50*10^3//total power
11 m_a = .707//modulation index
12 z = 50+0*%i;//load
13
14 //calculations
15
16 //first case
17 p_x = .5*(m_a)^2;
18 p_c = p_t/(1+p_x)//carrier power
19
20 //second case
21 n = ((p_c*p_x)/(p_c+(p_c*p_x)))*100;//transmission
    efficiency

```



```

22
23 //third case
24 a_c = sqrt(2*z*p_c); //peak carrier amplitude
25
26 //results
27 printf("\n\ni. Carrier Power =%.2f W",p_c);
28 printf("\n\nii. Percentage Transmission efficiency
        =%.2f percent",n);
29 printf("\n\niii. Carrier amplitude =%.2f V",a_c)
        ;

```

Scilab code Exa 2.29 modulation depth

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.29(PAGENO 118)");
8
9 //given
10 k = 2*10^-3//constants in amperes/square volts
11 k_1 = 0.2*10^-3//constant in amperes/square volts
12 printf("\n\nwe know that V_i(t) = cos(w_c*t) + .5*
        cos(w_m*t)");
13 printf("given i_0 = 10 + k*V_i + k_1*V_i^2 ");
14 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)]");
15 printf("\n\ni_0 = 2*10^-3*cos(w_c*t) + ((.2*10^-3)
        /.5)*.5*cos(w_c*t)*cos(w_m*t)");
16 //Now the modulation depth will be
17 m = (.2*10^-3)/.5;
18

```

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

Scilab code Exa 2.31 percentage power saving

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.31(PAGENO 119)");
8
9 //given
10 //percentage modulation for first case
11 Pm_1 = 100
12 //percentage modulation for second case
13 Pm_2 = 50
14 m_1 = 1//modulation index for first case
15 m_2 = .5//modulation index for second case
16 P_c = 1//let carrier power be one
17
18 //calcuations
19
20 //first case
21 P_t1 = P_c*(1+(m_1^2/2))//total power
22 P_sb1 = P_c*(m_1^2/4)//power in one side band
23 P_s1 = ((P_t1-P_sb1)/P_t1)*100//power saving
24
25 //second case
26 P_t2 = P_c*(1+(m_2^2/2))//total power
27 P_sb2 = P_c*(m_2^2/4)//power in one side band
28 P_s2 = ((P_t2-P_sb2)/P_t2)*100//power saving
29
30 //results
```

```

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

```

Scilab code Exa 2.32 carrier frequency

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 2
4 //AMPLITUDE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 2.31(PAGENO 119)");
8
9 //given
10 //the product signal is given by
11 //v(t) = s(t) * cos(2*%pi*t +phi) = x(t) *cos(2*%pi*
    f_c*t)*cos(2*%pi*f_c*t +phi)
12 //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)
13 //the low pass filter will reject the first term.
    The maximum allowable value of phase angle(phi)
    can be found as under:
14 printf("\n\n cos(phi_max) = ((x(t)/2)*cos(phi))/max((
    x(t)/2)*cos(phi))");
15 phi_max = acosd(.95);
16 printf("\n\n phi_max = %.2f",phi_max);
17 printf("\n\n In 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,")
18 printf("\n 2f_c -10KHz > 10KHz");
19 printf("\n f_c >10KHz");

```

```
20 printf("\nHence, the minimum value of f_c will be");  
21 printf("\nf_c = 10KHz")
```

Chapter 3

Radio Transmitters

Scilab code Exa 3.1 total radiated power

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 3
4 //RADIO TRANSMITTER
5 clear all;
6 clc;
7 printf("EXAMPLE 3.1(PAGENO 132)");
8
9 //given
10 Pm = 85//percentage modulation
11 m = .85//modulation index
12 P_c = 50*10^3//carrier power in watts
13
14 //calculation
15 P_t = P_c*(1+(m^2/2));//total radiated power
16
17 //result
18 printf("\n\nTotal radiated power = %.2f W",P_t);
```

Scilab code Exa 3.2 carrier frequency and frequency deviation

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 3
4 //RADIO TRANSMITTER
5 clear all;
6 clc;
7 printf("EXAMPLE 3.2(PAGENO 138)");
8
9 //given from the figure
10 f = 20*10^6//frequency in hertz
11 //At point 1 from fig
12 f_c1 = 2*13.5*10^6//carrier frequency
13 deltaf1 = 2*8.5*10^3//change in frequency
14
15 //calculations
16 f_max1 = f_c1 + deltaf1//maximum frequency at point
   1 in fig
17 f_min1 = f_c1 - deltaf1//maximum frequency at point
   1 in fig
18 f_d1 = f_max1 - f_c1;//frequency deviation at point
   1 in fig
19 f_d2 = f_c1 - f_min1;//frequency deviation at point
   1 in fig
20 //At point 2 from fig
21 f_c2 = 3*f_c1//carrier frequency
22 deltaf2 = 3*deltaf1//change in frequency
23 f_max2 = f_c2 + deltaf2//maximum frequency at point
   2 in fig
24 f_min2 = f_c2 - deltaf2//minimum frequency at point
   2 in fig
25 f_d3 = f_max2 - f_c2;//frequency deviation at point
   2 in fig
26 f_d4 = f_c2 - f_min2;//frequency deviation at point
   2 in fig
27 //At point 3 in fig
28 f_c3 = f_c2 + f;//carrier frequency at point 3 in
```

```

fig
29 f_max3 = f_max2 + f//maximum frequency at point 3 in
    fig
30 f_min3 = f_min2+ f//minimum frequency at point 3 in
    fig
31 f_d5 = f_max3 - f_c3;//frequency deviation at the
    last point
32 f_d6 = f_c3 - f_min3 ;//frequency deviation at the
    last point
33
34 //results
35 printf("\n\ni.a Carrier frequency at point 1 in fig
    =%.2f Hz ",f_c1);
36 printf("\n\n  b Frequency deviation =%.2f Hz ",f_d1)
    ;
37 printf("\n\n  c Frequency deviation =%.2f Hz ",f_d2
    );
38 printf("\n\nii.a Carrier frequency at point 2 in fig
    =%.2f Hz ",f_c2);
39 printf("\n\n      b Frequency deviation =%.2f Hz ",
    f_d3);
40 printf("\n\n      c Frequency deviation =%.2f Hz ",
    f_d4);
41 printf("\n\niii.a Carrier frequency at point 3 in fig
    =%.2fHz ",f_c3);
42 printf("\n\n      b Frequency deviation =%.2f Hz ",
    f_d5);
43 printf("\n\n      c Frequency deviation =%.2f Hz ",
    f_d6);
44 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 3
4 //RADIO TRANSMITTER
5 clear all;
6 clc;
7 printf("EXAMPLE 3.3(PAGENO 138)");
8 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.");
9 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

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 4
4 //Radio Receiver
5 clear all;
6 clc;
7 printf("EXAMPLE 4.1(PAGENO 150)");
8
9 //given data
10 Q = 100//quality factor
11 f_i = 455*10^3//intermediate frequency
12
13 //calculations
14 //first case
15 f_s = 1000*10^3//incoming frequency of first case
16 f_si = f_s + 2*f_i//image frequency of first case
17 p = (f_si/f_s) - (f_s/f_si);
18 alpha = sqrt(1+(Q^2*p^2))//rejection ratio of first
    case
19 //second case
20 f_s1 = 25*10^6//incoming frequency of second case
```

```

21 f_si1 = f_s1+ 2*f_i//image frequency of second case
22 p1 = ((f_si1/f_s1) - (f_s1/f_si1));
23 alpha1 = sqrt(1+(Q^2*p1^2))//rejection ratio of
    second case
24
25 //results
26 printf("\n\n(i)a.Image frequency of first case = %.2
    fHz",f_si);
27 printf("\n\n    b.Rejection ratio of first case = %.2
    f",alpha);
28 printf("\n\n (ii)a.Image frequency of second case =
    %.2 fHz",f_si1);
29 printf("\n\n    b.Rejection ratio of second case =
    %.2 f",alpha1);
30 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 4
4 //Radio Receiver
5 clear all;
6 clc;
7 printf("EXAMPLE 4.2(PAGENO 150)");
8
9 //given data
10 Q = 90
11 f_i = 455*10^3//intermediate frequency
12
13 //calculations
14 //first case
15 f_s = 950*10^3//incoming frequency of first case
16 f_si = f_s + 2*f_i//image frequency of first case

```

```

17 p = (f_si/f_s) - (f_s/f_si);
18 alpha = sqrt(1+(Q^2*p^2))//rejection ratio of first
    case
19 //second case
20 f_s1 = 10*10^6//incoming frequency of second case
21 f_si1 = f_s1+ 2*f_i//image frequency of second case
22 p1 = ((f_si1/f_s1) - (f_s1/f_si1));
23 alpha1 = sqrt(1+(Q^2*p1^2))//rejection ratio of
    second case
24
25 //results
26 printf("\n\n(i)a.Image frequency of first case = %.2
    f Hz",f_si);
27 printf("\n\n    b.Rejection ratio of first case = %.2
    f",alpha);
28 printf("\n\n(ii)a.Image frequency of second case = %
    .2 f Hz",f_si1);
29 printf("\n\n    b.Rejection ratio of second case = %
    .2 f",alpha1);

```

Scilab code Exa 4.3 quality factor and new intermediate frequency

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 4
4 //Radio Receiver
5 clear all;
6 clc;
7 printf("EXAMPLE 4.3(PAGENO 151)");
8 //given
9
10 a1 = 130.5//rejection ratio
11 f_s = 10*10^3//incoming frequency
12 printf("from fig 4.8 from t/b we can write that")
13

```

```

14 //calculations
15 //first case
16 alpha = 130.5//from problem 4.2 of first case
17 alpha2 = 15.72//from problem 4.2 of second case
18 alpha1 = alpha/alpha2//rejection ratio of given RF
    amplifier
19 p1 = .174//from problem 4.2 of second case
20 Q = (sqrt(alpha1^2 - 1)/p1)//quality factor
21 //second case
22 p2 = 1.45//from problem 4.2 of second case
23 f_si = 1860*10^3//from problem 4.2 of second case
24 f_i = 950*10^3//incoming frequency
25 f_i1 = 10*10^6//good image frequency
26 f_si1 = (f_si*f_i1)/f_i';//image frequency
27 f_i2 = (f_si1 - f_i1)/2//new intermediate frequency
28
29 //results
30 printf("\n\n(i) Quality factor =%.2f ",Q);
31 printf("\n\n(ii) New intermediate frequency =%.4f Hz"
    ,f_i2);

```

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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 4
4 //Radio Receiver
5 clear all;
6 clc;
7 printf("EXAMPLE 4.5(PAGENO 152)");
8
9 //given
10 IF = 455*10^3//intermediate frequency in hertz
11 f_s = 900*10^3//signal frequency in hertz
12 Q = 80//quality factor

```

```

13
14 //calculations
15 f_0 = f_s + IF//local oscillator frequency
16 f_si = f_s + 2* IF//image frequency
17 p = (f_si/f_s)-(f_s/f_si)
18 a = sqrt(1+(Q*p)^2)//image frequency rejection ratio
19
20 //results
21 printf("\n\n(i)Local oscillator frequency = %.2f Hz"
    ,f_0);
22 printf("\n\n(ii)Image frequency = %.2f Hz",f_si);
23 printf("\n\n(iii)Image frequency rejection ratio = %
    .2f",a);
24 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 fre

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 4
4 //Radio Receiver
5 clear all;
6 clc;
7 printf("EXAMPLE 4.6(PAGENO 153)");
8 //given
9 Q = 125//quality factor
10
11 //calculations
12 //first case
13 IF1 = 465*10^3//intermediate frequency
14 f_s1 = 1*10^6//incoming frequency for first case in
    hertz
15 f_s2 = 30*10^6//second incoming frequency for first
    case in hertz

```

```

16 f_si1 = f_s1 + 2*IF1//image frequency for incoming
    frequency 1MHz for first case
17 f_si2 = f_s2 + 2*IF1//image frequency for incoming
    frequency 30MHz for first case
18 p1 = (f_si1/f_s1)-(f_s1/f_si1);
19 p2 = (f_si2/f_s2)-(f_s2/f_si2);
20 alpha1 = sqrt(1+(Q*p1)^2);//rejection ratio at 1MHz
    incoming frequency
21 alpha2 = sqrt(1+(Q*p2)^2);//rejection ratio at 30MHz
    incoming frequency
22 //second case
23 f_s3 = 1*10^6//incoming frequency for second case in
    hertz
24 f_si3 = (f_si1*f_s2)/f_s3//image frequency
25 IF2 = (f_si3-f_s2)/2//intermediate frequency
26
27 //results
28 printf("\n\n(i)a.Image frequency for 1MHz incoming
    frequency = %.2f Hz",f_si1);
29 printf("\n\n b.Rejection ratio for 1MHz incoming
    frequency = %.2f",alpha1);
30 printf("\n\n c.Image frequency for 30MHz incoming
    frequency = %.2f Hz",f_si2);
31 printf("\n\n d.Rejection ratio for 30MHz incoming
    frequency = %.2f",alpha2);
32 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

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.1(PAGENO 198)");
8
9 //given
10 t = [0:.1:10]; //time period
11 R = 10 //resistance in ohms
12 printf("\n\nv(t) = 12*cos(6*10^8*t + 5*sin(1250*t));
13 ");
14 printf("\n\nv(t) = A*cos(w_c*t + m_f*sin(w_m*t))");
15 //by comparing with standard
16 A = 12 //amplitude voltage in volts
17 w_c = 6*10^8 //angular carrier frequency in rad/sec
18 w_m = 1250 //angular modulating frequency in rad/sec
19 m_f = 5 //modulation index
20 //calculations
```

```

21 f_c = w_c/(2*pi)//carrier frequency
22 f_m = w_m/(2*pi)//modulating frequency
23 deltaf = m_f*f_m//maximum deviation
24 V_rms = (A/sqrt(2))^2//rms volatage
25 P = V_rms/R//power dissipatted
26
27 //results
28 printf("\n\n i. Carrier frequency = %.2f Hz",f_c);
29 printf("\n\n ii. Modulation frequency = %.2f Hz",f_m)
    ;
30 printf("\n\n iii. Modulation index = %.2f",m_f);
31 printf("\n\n iv. Maximum frequency deviation = %.2f Hz
    ", deltaf);
32 printf("\n\n v. Power dissipated = %.2f W ",P);

```

Scilab code Exa 5.2 carrier frequency and highest and lowest frequency and modulat

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.2(PAGENO 199)");
8
9 //given
10 f_c = 107.6*10^6//carrier frequency
11 f_m = 7*10^3//modulating frequency
12 deltaf = 50*10^3//frequency deviation
13
14 //calculations
15 cs = 2*deltaf//carrier swing
16 f_H = f_c + deltaf//highest frequency
17 f_L = f_c - deltaf//lowest frequency
18 m_f = deltaf/f_m//modulating index

```



```

19
20 //results
21 printf("\n\ni. Carrier frequency = %.2f Hz",cs);
22 printf("\n\nii.a.Highest frequency attained by the
    modulating signal = %.2f Hz",f_H);
23 printf("\n\n    b.Lowest frequency attained by the
    modulating signal = %.2f Hz",f_L);
24 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.3(PAGENO 199)");
8 //given
9 f_c = 105*10^6//carrier frequency
10 f_H = 105.007*10^6//highest frequency or upper
    frequency
11 //calculations
12 deltaf = f_H - f_c//frequency deviation
13 cs = 2*deltaf//carrier swing
14 f_L = f_c - deltaf//lower frequency
15 //results
16 printf("\n\ni.Frequency deviation = %.4f Hz",deltaf)
    ;
17 printf("\n\nii.Carrier swing = %.2f Hz",cs);
18 printf("\n\niii.Lower frequency reached by the
    modulated wave = %.3f Hz",f_L);

```

Scilab code Exa 5.4 modulation index

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.4(PAGENO 200)");
8
9 //given
10 cs = 100*10^3//carrier swing
11 f_m = 8*10^3//modulating frequency
12
13 //calculations
14 deltaf = cs/2//frequency deviation
15 m_f = deltaf/f_m//modulation index
16 //results
17 printf("\n\n      Modulation index = %.2 f",m_f);
```

Scilab code Exa 5.5 percentage modulation index

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.5(PAGENO 200)");
8
9 //given
10 deltaf = 20*10^3//frequency deviation
```

```

11 deltaf_actual = deltaf //since deltaf_actual equals
    to deltaf
12 deltaf_max1 = 75*10^3 //maximum frequency deviation
    deltaf_max permitted for the first case is 75KHz
13 deltaf_max2 = 25*10^3 //maximum frequency deviation
    deltaf_max permitted for the second case is 25KHz
14
15 //calculations
16 M1 = (deltaf_actual/deltaf_max1)*100 //percentage
    modulation index for first case
17 M2 = (deltaf_actual/deltaf_max2)*100 //percentage
    modulation index for second case
18
19 //results
20 printf("\n\ni. Percentage modulation index for first
    case = %.2f percent",M1);
21 printf("\n\nii. Percentage modulation index for
    second case = %.2f percent",M2);

```

Scilab code Exa 5.6 bandwidth of FM transmission

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.6(PAGENO 210)");
8
9 //given
10 deltaf = 75*10^3 //frequency deviation
11 f_m = 15*10^3 //modulating frequency
12
13 //calculation
14 BW = 2*(deltaf+f_m) //bandwidth

```

```

15
16 //result
17 printf("\n\nBandwidth of a commercial FM
    transmission = %.2f Hz",BW);

```

Scilab code Exa 5.7 bandwidth of narrowband FM signal

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.7(PAGENO 210)");
8
9 //given
10 f_m = 4*10^3//modulation frequency
11 f_c = 125*10^3//carrier frequency
12
13 //claculation
14 BW = 2*f_m//bandwidth
15
16 //result
17 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;

```

```

6  clc;
7  printf("EXAMPLE 5.8(PAGENO 210)");
8
9  //given
10 deltaf1 = 75*10^3//frequency deviation
11 f_m = 8*10^3//modulation frequency
12 deltaf2 = 2*deltaf1//if modulation signal amplitude
    is doubled, the frequency deviation becomes
    double
13
14 //calculation
15 BW1 = 2*(deltaf1 +f_m)//bandwidth
16 BW2 = 2*(deltaf2 +f_m)//new bandwidth
17
18 //result
19 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

```

1  //ANALOG AND DIGITAL COMMUNICATION
2  //BY Dr.SANJAY SHARMA
3  //CHAPTER 5
4  //ANGLE MODULATION
5  clear all;
6  clc;
7  printf("EXAMPLE 5.9(PAGENO 211)");
8
9  //given
10 f_m = 5*10^3//modulating frequency
11 f_c = 50*10^6//carrier frequency
12 deltaf = 20*10^3//frequency deviation
13
14 //calculations
15 m_f = deltaf/f_m//modulation index

```

```

16 BW = deltaf*3.8//referring to the Schwartz bandwidth
    curve
17
18 //results
19 printf("\n\ni.Modulation index = %.2f",m_f);
20 printf("\n\nii.Bandwidth of the FM signal = %.2f Hz"
    ,BW);

```

Scilab code Exa 5.10 modulating frequency

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.10(PAGENO 212)");
8
9 //given
10 BW = 50*10^3//bandwidth
11 deltaf = 10*10^3//frequency deviation
12
13 //calculation
14 x = BW/deltaf//variable
15 m_f = 2//by referring to the Schwartz bandwidth
    curve with 'x'
16 f_m = deltaf/m_f//modulating frequency
17
18 //results
19 printf("\n\n Modulating frequency = %.2f Hz",f_m);

```

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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.11(PAGENO 217)");
8 //given
9 //x(t) = 5*cos(2*%pi*15*10^3*t)
10 V_m = 5//amplitude of voltage
11 f_m = 15*10^3//modulation frequency
12 k_f = 15*10^3//frequency sensitivity
13 k_p = 15*10^3//phase sensitivity
14
15 //calculations
16 //first case
17 //for FM system
18 delta_f1 = k_f * V_m;//frequency deviation for FM
    system
19 m_f1 = delta_f1/f_m; //modulation index in FM system
20 BW1 = 2*(delta_f1+f_m);//bandwidth for FM system
21 //for PM system
22 delta_f2 = k_f * V_m*f_m;//frequency deviation for
    PM system
23 BW2 = 2*(delta_f2 + f_m);//bandwidth for PM system
24 m_p1 = k_p * V_m//modulation index in PM system
25
26 //second case
27 f_m1 = 5*10^3//modulating frequency for second case
28 //for FM system
29 delta_f3 = k_p * V_m;//frequency deviation for FM
    system
30 m_f2 = delta_f3/f_m1; //modulation index in FM
    system
31 BW3 = 2*(delta_f3+f_m1);//bandwidth for FM system
32 //for PM system
33 delta_f4 = k_p * V_m*f_m1;//frequency deviation for
    PM system

```

```

34 BW4 = 2*(delta_f4 + f_m1); //bandwidth for PM system
35 m_p2 = k_p * V_m //modulation index in PM system
36
37 //results
38 printf("\n\ni.a.Modulation index of FM system for
    first case = %.2f",m_f1);
39 printf("\n\n b.Bandwidth of FM system for first
    case = %.2f Hz",BW1);
40 printf("\n\nii.a.Modulation index of PM system for
    first case = %.2f",m_p1);
41 printf("\n\n b.Bandwidth of PM system for first
    case = %.2f Hz",BW2);
42 printf("\n\niii.a.Modulation index of FM system for
    second case = %.2f",m_f2);
43 printf("\n\n b.Bandwidth of FM system for second
    case = %.2f Hz",BW3);
44 printf("\n\niv.a.Modulation index of PM system for
    second case = %.2f",m_p2);
45 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 ind

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.12(PAGENO 219)");
8
9 //given
10 //given that v = 10*sin((5 * 10^8 *t) + 4*sin(1250*t
    ))
11 //by comparing with standard eqn i.e v = V_c*sin((

```



```

    w_c * t) + m_f*sin(w_m*t)) we get
12 w_c = 5*10^8//angular carrier frequency
13 w_m = 1250//angular modulating frequency
14 m_f = 4//modulating index
15 V_c = 10//carrier voltage in volts
16 R = 5//resistance in ohms
17
18 //calculations
19 f_c = w_c/(2*%pi)//carrier frequency
20 f_m = w_m/(2*%pi)//modulating frequency
21 deltaf = m_f * f_m//maximum deviation
22 V_rms = (V_c/sqrt(2))^2//RMS value of FM wave
23 P = V_rms/R//power dissipated
24
25 //results
26 printf("\n\ni.a.Carrier frequency = %.2f Hz",f_c);
27 printf("\n\n  b.Modulating frequency = %.2f Hz",f_m)
    ;
28 printf("\n\niii.a.Modulation index = %.2f",m_f);
29 printf("\n\n  b.Maximum deviation = %.2f Hz",deltaf
    );
30 printf("\n\niii.Power dissipated in 5 ohms
    resistance = %.2f W",P);

```

Scilab code Exa 5.13 frequency deviation for given modulating volatge

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.13(PAGENO 220)");
8
9 //given

```

```

10 f_m = 1*10^3//modulating frequency
11 V_m = 2//modulating voltage in volts
12 deltaf = 6*10^3//frequency deviation
13 V_m1 = 4//increased modulation voltage for first
    case
14 V_m2 = 8//increased modulation voltage for second
    case
15
16 //calculations
17 k_f = deltaf/V_m//proportion constant
18 //first case
19 deltaf1 = k_f*V_m1//frequency deviation for first
    case
20 //second case
21 deltaf2 = k_f*V_m2//frequency deviation for second
    case
22
23 //results
24 printf("\n\ni.Frequency deviation for modulating
    voltage 4V = %.2f Hz",deltaf1);
25 printf("\n\nii.Frequency deviation for modulating
    voltage 8V = %.2f Hz",deltaf2);

```

Scilab code Exa 5.14 modulation index

```

1
2 //ANALOG AND DIGITAL COMMUNICATION
3 //BY Dr.SANJAY SHARMA
4 //CHAPTER 5
5 //ANGLE MODULATION
6 clear all;
7 clc;
8 printf("EXAMPLE 5.14(PAGENO 220)");
9
10 //given

```

```

11 deltaf = 6*10^3//frequency deviation from the
    question of EXAMPLE 5.13(PAGENO 220)
12 f_m = 1*10^3//modulating frequency from the question
    of EXAMPLE 5.13(PAGENO 220)
13 deltaf1 = 12*10^3//frequency deviation from the
    EXAMPLE 5.13(PAGENO 220) of first case
14 deltaf2 = 24*10^3//frequency deviation from the
    EXAMPLE 5.13(PAGENO 220) of second case
15 f_m1 = f_m//modulating frequency from the EXAMPLE
    5.13(PAGENO 220) of first case
16 f_m2 = 500//modulating frequency from the EXAMPLE
    5.13(PAGENO 220) ofsecond case
17
18 //calculation
19 m_f = deltaf/f_m//modulation index for the initial
    conditions given in the problem 5.13
20 m_f1 = deltaf1/f_m1//modulation index for the first
    case
21 m_f2 = deltaf2/f_m2//modulation index for the second
    case
22
23 //results
24 printf("\n\na.Modulation index for initial
    conditions given in the problem 5.13 = %.2f ",m_f
    );
25 printf("\n\nb.Modulation index for the first case =
    %.2f",m_f1);
26 printf("\n\nc.Modulation index for the second case =
    %.2f",m_f2);

```

Scilab code Exa 5.15 bandwidth of FM and DSB FC signal

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5

```

```

4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.15(PAGENO 220)");
8
9 //given
10 deltaf = 10*10^3//frequency deviation
11 f_m = 1*10^3//modulating frequency
12
13 //calculations
14 BW = 2*(deltaf + f_m)//bandwidth of FM signal
15 BW_DSB = 2*f_m//bandwidth of DSB FC(AM)
16
17 //results
18 printf("\n\ni.Bandwidth of FM signal = %.2 f Hz",BW);
19 printf("\n\nii.Bandwidth of DSB FC(AM) signal = %.2 f
      Hz",BW_DSB);

```

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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.16(PAGENO 221)");
8
9 //given
10 //first case
11 f_c1 = 20*10^6//carrier frequency
12 f_m1 = 400//modulation frequency
13 V_c = 5//carrier voltage in volts
14 deltaf = 10*10^3//frequency deviation
15 //second case

```

```

16 f_m2 = 2*10^3//modulation frequency
17
18 //calculations
19 w_c1 = 2 *%pi *f_c1//angular carrier frequency
20 w_m1 = 2 *%pi *f_m1//angular carrier frequency
21 m_f1 = deltaf/f_m1//modulation index for first case
22 m_f2 = deltaf/f_m2//modulation index for second case
23
24 //results
25 //standard format of fm and pm equations are
26 //s(t) = V_c8sin(w_c*t + m_f*sin(w_m*t))
27 printf("\n\n(i)FM wave:s(t) = 5*sin(1.25*10^8*t +
      25*sin(2513*t)");
28 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.17(PAGENO 221)");
8
9 //given
10 V_m = 5//modulating voltage
11 f_m = 20*10^3//modulating frequency
12 V_c = 10//carrier voltage
13 f_c = 100*10^6//carrier frequency
14 delta_f = 2*10^3//frequency deviation in hertz per
      volt
15
16 //calculations

```

```

17 m_a = delta_f/f_m//modulation index
18 printf("\n\nfor m_a = .5 the approximate values of j
    coefficients are");
19 printf("\n\nJ_0 = .94    J_1 = .24    J_2 =.03");
20 J_0 = .94
21 J_1 = .24
22 J_2 =.03
23 A_c = V_c*J_0;//carrier amplitude
24 A_1 = V_c*J_1;//amplitude of first pair of sideband
25 A_2 = V_c*J_2;//amplitude of second pair of sideband
26 f_1 = f_c + f_m//maximum frequency of first pair of
    sideband
27 f_1a = f_c - f_m//minimum frequency of first pair of
    sideband
28 f_2 = f_c + (2*f_m)//maximum frequency of second
    pair of sideband
29 f_2a = f_c - (2*f_m)//minimum frequency of second
    pair of sideband
30
31 //results
32 printf("\n\ni. Carrier amplitude = %f V",A_c);
33 printf("\n\nii. Amplitude of first pair of sideband =
    %f V",A_1);
34 printf("\n\niii. Amplitude of second pair of sideband
    = %f V",A_2);
35 printf("\n\nV.a. Maximum frequency of first pair of
    sideband = %f Hz",f_1);
36 printf("\n\n    .b. Minimum frequency of first pair
    of sideband = %f Hz",f_1a);
37 printf("\n\nV.a. Maximum frequency of second pair of
    sideband = %f Hz",f_2);
38 printf("\n\n    .b. Minimum frequency of second pair
    of sideband = %f Hz",f_2a);

```

Scilab code Exa 5.18 maximum frequency deviation and modulation index

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.18(PAGENO 222)");
8
9 //given
10 f_m1 = 400//modulating frequency for first case
11 V_m1 = 2.4//modulating voltage for first case
12 f_m2 = 250//modulating frequency for second case
13 V_m2 = 3.2//modulating voltage for second case
14 m_f1 = 60//modulation index for first case
15
16 //calculations
17 delta_f1 = m_f1*f_m1//maximum frequency deviation
    for first case
18 k = delta_f1/V_m1//constant
19 delta_f2 = k*V_m2//frequency deviation for second
    case
20 m_f2 = delta_f2/f_m2//modulation index for second
    case
21
22 //results
23 printf("\n\ni.Maximum frequency deviation for first
    case = %.2 f Hz",delta_f1);
24 printf("\n\nii.Modulation index for second case = %
    .2 f ",m_f2);

```

Scilab code Exa 5.19 modulation index and bandwidth of a signal

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5

```

```

4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.19(PAGENO 222)");
8
9 //given
10 f_m1 = 1*10^3//modulating frequency for first case
11 f_m2 = 500//modulating frequency for second case
12 V_m1 = 2//modulating voltage for first case
13 V_m2 = 8//modulating volatge for second case
14 delta_f1 = 4*10^3//frequency deviation for first
    case
15
16 //calculations
17 k = delta_f1/V_m1//constant
18 delta_f2 = k*V_m2//frequency deviation for second
    case
19 m_f1 = delta_f1/f_m1//modulation index for first
    case
20 m_f2 = delta_f2/f_m2//modulation index for second
    case
21 BW1 = 2*(delta_f1 + f_m1)//bandwidth for first case
22 BW2 = 2*(delta_f2 + f_m2)//bandwidth for second
    case
23
24 //results
25 printf("\n\ni.a.Modulation index for first case = %
    .2f",m_f1);
26 printf("\n\n b.Bandwidth for first case = %.2f Hz",
    BW1);
27 printf("\n\nii.a.Modulation index for second case =
    %.2f",m_f2);
28 printf("\n\n b .Bandwidth for second case = %.2f Hz"
    ,BW2);
29 printf("\n\nNote: Their is error in textbook in the
    calculation of second case bandwidth ");

```

Scilab code Exa 5.20 capacitive reactance

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7
8 printf("EXAMPLE 5.20(PAGENO 232)");
9 //given
10 g_m = 10*10^-3//transconductance
11 n = 8
12 f= 5*10^6//operating frequency
13
14 //calculation
15 X_Ceq = n/g_m//capacitive reactance
16
17 //result
18 printf("\n\nCapacitive reactance = %.2f ohms",X_Ceq)
```

Scilab code Exa 5.21 carrier frequency and frequency deviation and modulation index

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.21(PAGENO 233)");
8
9 //given data from block diagram
```

```

10 f_c = 10*10^6//carrier frequency
11 delta_f = 10*10^3//frequency deviation
12 m_f = 5//modulation index
13
14 //calculations
15 //first stage
16 f_cA = 3 * f_c//carrier frequency at point A
17 delta_fA = 3 * delta_f//frequency deviation at point
   A
18 m_fA = 3 * m_f//modulation index at point A
19 f_maxA = f_cA + delta_fA//maximum frequency at point
   A
20 f_minA = f_cA - delta_fA//minimum frequency at point
   A
21 //second stage
22 f_cB = f_cA + f_c//carrier frequency at point B
23 f_maxB = f_maxA + f_c//maximum frequency at point B
24 f_minB = f_minA + f_c//minimum frequency at point B
25 delta_fB = f_maxB - f_cB//frequency deviation at
   point B
26 //their will no change in modulation index
27
28 //results
29 printf("\n\n i.a. Carrier frequency at point A = %.2f
   Hz", f_cA);
30 printf("\n\n b. Frequency deviation at point A = %.2f
   Hz", delta_fA);
31 printf("\n\n c. Modulation index at point A = %.2f ",
   m_fA);
32 printf("\n\n d. Maximum frequency at point A = %.2f
   Hz", f_maxA);
33 printf("\n\n e. Minimum frequency at point A = %.2f
   Hz", f_minA);
34 printf("\n\n ii.a. Carrier frequency at point B = %.2f
   Hz", f_cB);
35 printf("\n\n b. Frequency deviation at point B = %.2f
   Hz", delta_fB);
36 printf("\n\n c. Modulation index at point B = %.2f ",

```

```

    m_fA);
37 printf("\n\n d.Maximum frequency at point B = %.2 f
    Hz", f_maxB);
38 printf("\n\n e.Minimum frequency at point B = %.2 f
    Hz", f_minB);

```

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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.22(PAGENO 251)");
8 //given
9 //first case
10 //The maximum deviation in commerical FM is given as
11 delta_f1 = 75*10^3//frequency deviation in
    commerical FM
12 f_m1 = 30//maximum modulating frequency
13 f_m2 = 15*10^3//minimum modulating frequency
14 //second case
15 delta_f2 = 10*10^3//frequency deviation for
    narrowband FM
16 f_m3 = 100//maximum modulating frequency
17 f_m4 = 3*10^3//minimum modulating frequency
18
19 //calculations
20 //first case
21 m_f1 = delta_f1/f_m1//modulation index for maximum
    modulating frequency
22 m_f2 = delta_f1/f_m2//modulation index for minimum
    modulating frequency
23 //second case

```

```

24 m_f3 = delta_f2/f_m3//modulation index for maximum
    modulating frequency
25 m_f4 = delta_f2/f_m4//modulation index for minimum
    modulating frequency
26
27 //results
28 printf("\n\n i.a.modulation index for maximum
    modulating frequency of commercial FM = %.2f",
    m_f1)
29 printf("\n\n b.modulation index for minimum
    modulating frequency of commercial FM = %.2f",
    m_f2)
30 printf("\n\n iii.a.modulation index for maximum
    modulating frequency of narrowband FM = %.2f",
    m_f3)
31 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.23(PAGENO 251)");
8
9 //given
10 disp('modulated carrier waveform is given by s(t) =
    A*sin((2*%pi*f_c*t)+m_f*sin(2*%pi*f_m*t))');
11 f_c = 100*10^6//carrier frequency in hertz
12 delta_f = 75*10^3//frequency deviation in hertz
13 f_m = 2*10^3//modulating frequency

```

```

14 A = 5 //peak voltage of carrier wave
15
16 //calculation
17 m_f = delta_f/f_m; //modulation index
18
19 //result
20 printf("\n      Modulation index =%.2f",m_f);
21 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.24(PAGENO 252)");
8
9 //given
10 //we know that s(t) = A*cos((2*%pi*f_c*t) + m_f*sin
      (2*%pi*f_m*t ))
11 f_c = 1*10^6 //modulation frequency
12 A = 3 //carrier amplitude in volts
13 //first case
14 A_m = 1 //modulating amplitude in volts for first
      case
15 delta_f = 1*10^3 //frequency deviation
16 f_m1 = 1 //modulating frequencyof first case
17 //second case
18 f_m2 = 2*10^3 //modulating frequency for second case
19 A_m2 = 5 //modulating amplitude for second case
20

```

```

21 //calculations
22 k_f = delta_f/f_m1//frequency sensitivity in hertz
    per volt
23 m_f = (delta_f*A_m2)/f_m2//modulating frequency
24 //desired FM signal can be expressed by  $s(t) = A\cos$ 
    ( $(2*\%pi*f_c*t) + m_f*\sin(2*\%pi*f_m*t)$  )
25 //results
26 //standard FM signal expression is as follows
27 // $s(t) = A\cos(2*\%pi*f_c*t + m_f * \sin(2*\%pi*f_m*t))$ 
28 printf("\n\nFrequency sensitivity k_f = %.2 f",k_f);
29 printf("\n\nModulation index m_f =%.2 f ",m_f);
30
31 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.29(PAGENO 256)");
8
9 //given
10 delta_f = 75*10^3//frequency deviation
11 f_m = 15*10^3//modulating frequency
12
13 //calculations
14 D = delta_f/f_m//deviation ratio
15 BW1 = 2*delta_f*(1+(1/D))//bandwidth of FM signal
16 //using universal curve , replacing m_f by D,we get
17 BW2 = 3.2*delta_f//for D = 5=3.2*75*10^3
18 BW = (BW2-BW1)*100/BW2//percentage of under

```

```

    estimation of bandwidth by using carson's rule
19
20 //results
21 printf("\n\ni.Bandwidth of FM signal = %.2f Hz",BW1)
    ;
22 printf("\n\nii.Bandwidth obtained by replacing m_f
    by D = %.2f Hz",BW2);
23 printf("\n\niii.Percentage of under estimation of
    bandwidth by using Carson rule = %.2f percent",BW
    );
24 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.30(PAGENO 257)");
8
9 //given
10 m_f1 = 1//modualtion index for first case
11 m_f2 = 10//modualtion index for second case
12 //let
13 f_m = 1*10^3//modulating frequency
14
15 //calulations
16 //the bandwidth for FM signal can be calculated on
    the basis of 98% power requirement given by
    Carson's rule
17 BW1 = 2*(m_f1+1)*f_m//bandwidth for first case

```

```

18 B1 = (2*m_f1 +1)*f_m//frequency band first case
19 BW2 = 2*(m_f2+1)*f_m//bandwidth for second case
20 B2 = (2*m_f2 +1)*f_m//frequency band second case
21 P1 = (B1/BW1)*(98)//fraction of signal power that is
    included in frequency band for 1st case
22 P2 = (B2/BW2)*(98)//fraction of signal power that is
    included in frequency band for 2nd case
23
24 //results
25 printf("\n\ni.Fraction of signal power that is
    included in frequency band for 1st case =%.2f
    percent",P1);
26 printf("\n\nii.Fraction of signal power that is
    included in frequency band for 2nd case =%.2f
    percent",P2);
27 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.31(PAGENO 257)");
8
9 //given
10 f_m1 = 2*10^3//modulating frequency for first case
11 delta_f1= 5*10^3//frequency deviation for first case
12 f_m2 = 1*10^3//modulating frequency for second case
13 delta_f2 = 3*5*10^3//modulating frequency for second
    case
14

```



```

15 //calculations
16 BW1 = 2*(delta_f1 + f_m1)//bandwidth of the FM
    signal for first case
17 BW2 = 2*(delta_f2 + f_m2)//bandwidth of the FM
    signal for secpnd case
18
19 //results
20 printf("\n\ni.Bandwidth of the FM signal for first
    case = %.2f Hz",BW1)
21 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.32(PAGENO 258)");
8
9 //given
10 m_f = .2//modulation index
11 P = 10*10^3//power of FM transmitter
12 J_0m_f = 0.99//bessel function
13 J_1m_f =0.099
14
15 //calculations
16 P_c = (J_0m_f)^2 * P//carrier power
17 P_s1 = (J_1m_f)^2 * P//power in each side frequency
18 P_s2 = P_s1
19
20 //results
21 printf("\n\ni. Carrier power = %.2f W",P_c);

```

```

22 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 fre

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.33(PAGENO 258)");
8
9 //given
10 f_m = 7*10^3//modulating frequency
11 delta_f = 50*10^3//frequency deviation
12 f_c = 107.6*10^6//carrier frequency
13
14 //calculaitons
15 CS = 2*delta_f//carrier swing
16 m_f = delta_f/f_m//modulation index
17 f_h = f_c + delta_f//upper or highest frequency
18 f_l = f_c - delta_f//lower of lowest frequency
19
20 //results
21 printf("\n\ni.a.Carrier swing = %.2f Hz",CS);
22 printf("\n\n b.Modulation index = %.4f ",m_f);
23 printf("\n\nii.a.Highest frequency attained by the
    FM signal = %.2f Hz",f_h);
24 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.34(PAGENO 259)");
8
9 //given
10 f_c = 100*10^6//carrier frequency
11 f_u = 100.007*10^6//upper frequency
12
13 //calculations
14 delta_f = f_u - f_c//frequency deviation
15 CS = 2*delta_f//carrier swing
16 f_l = f_c - delta_f//lower frequency reached by the
    modulated FM wave
17
18 //results
19 printf("\n\ni.Frequency deviation = %.2f Hz",delta_f
    );
20 printf("\n\nii.Carrier frequency = %.2f Hz",CS);
21 printf("\n\niii.Lower frequency reached by the
    modulated FM wave = %.2f Hz",f_l);

```

Scilab code Exa 5.35 percentage modulation index

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.35(PAGENO 259)");
8

```

```

9 //given
10 CS = 125*10^3//carrier swing
11
12 //calculations
13 delta_f = CS/2//frequency deviation
14 //since ,maximum frequency deviation for the FM
    broadcast band is 75 KHz, therefore
15 f_m = 75*10^3//modulating frequency
16 m_f = delta_f*100/f_m//modulation index
17
18 //result
19 printf("\n\nPercentage modulation index = %.2f
    percent",m_f);

```

Scilab code Exa 5.36 frequency deviation and modulation index

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.36(PAGENO 259)");
8
9 //given
10 //first case
11 f_m1 = 500//modulating frequency
12 delta_f1 = 6.4*10^3//frequency deviation
13 V_m1 = 3.2//modulating amplitude
14 //second case
15 V_m2 = 8.4//modulating amplitude
16 //third case
17 V_m3 = 20//modulating amplitude
18 f_m3 = 200//modulating frequency
19

```

```

20 //calculations
21 k_f = delta_f1/V_m1//frequency sensitivity
22 delta_f2 = k_f*V_m2//frequency deviation for second
    case
23 delta_f3 = k_f*V_m3//frequency deviation for third
    case
24 m_1 = delta_f1/f_m1//modulation index for first case
25 m_2 = delta_f2/f_m1//modulation index for second
    case
26 m_3 = delta_f3/f_m3//modulation index for third case
27
28 //results
29 printf("\n\ni.a.Frequency deviation for first case =
    %.2f Hz",delta_f1);
30 printf("\n\n b.Modulation index for first case = %
    .2f ",m_1);
31 printf("\n\nii.a.Frequency deviation for second case
    = %.2f Hz",delta_f2);
32 printf("\n\n b.Modulation index for second case =
    %.2f ",m_2);
33 printf("\n\niii.a.Frequency deviation for third case
    = %.2f Hz",delta_f3);
34 printf("\n\n b.Modulation index for third case =
    %.2f ",m_3);

```

Scilab code Exa 5.37 carrier frequency and modulating frequency and modulation ind

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.37(PAGENO 260)");
8

```

```

9 //given
10 // s(t) = 20*sin(6*10^8*t + 7*sin(1250*t))
11 //comparing with standard eqn s(t) = A*sin(w_c*t +
    m_f*sin(w_m*t))
12 //we get
13 w_c = 6*10^8//carrier angular frequency in rad/sec
14 w_m = 1250//modulating angular frequency in rad/sec
15 m_f = 7//modulation index
16 A = 20//amplitude of modulated wave
17 R = 100//resistance
18
19 //calculations
20 f_c = w_c/(2*%pi)//carrier frequency in hertz
21 f_m = w_m/(2*%pi)//modulating frequency in hertz
22 delta_f = m_f*f_m//frequency deviation
23 P = (A/sqrt(2))^2/R//power dissipated
24
25 //results
26 printf("\n\ni.Carrier frequency = %.2 f Hz ",f_c);
27 printf("\n\nii.Modulating frequency = %.2 f Hz ",f_m
    );
28 printf("\n\niii.Modulation index = %.2 f ",m_f);
29 printf("\n\niv.Frequency deviation = %.2 f Hz",
    delta_f);
30 printf("\n\nv.Power dissipated by FM wave = %.2 f W",
    P);

```

Scilab code Exa 5.39 maximum frequency deviation

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;

```

```

7 printf("EXAMPLE 5.39(PAGENO 261)");
8
9 //given
10 //x_c(t) = 10*cos[(10^8*pi*t) + 5*sin(2*pi*10^3)t]
11 //by comparing the given x_c(t) with standard FM
    wave equation
12 t=[1:1:10];
13 w_c = 10^8//carreier frequency
14 phi_t = 5*sin(2*pi*10^3*t);
15 phi_1t = 5*2*pi*10^3*cos(2*pi*10^3*t)
16 //Therefore, the maximum phase deviation will be
17 phi_tmax = 5//radians
18
19 //calculation
20 delta_f = (5*10^3*2*pi)/(2*pi);//maximum frequency
    deviation
21
22 //results
23 printf("\n\nMaximum frequency deviation is %.2f Hz",
    delta_f);

```

Scilab code Exa 5.43 bandwidth of angle modulated signal

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.43(PAGENO 263)");
8
9 //given
10 //x_c(t) = 10*cos(2*pi*10^8*t + 200*cos(2*pi
    *10^3*t))
11 //instantaneous frequency w_i = 2*pi*10^8 - 4*pi

```

```

    *10^6*sin(2*%pi*10^3)
12 delta_w = 4*%pi*10^5//angular frequency deviation
13 w_m = 2*%pi*10^3//angulat modulating frequency
14
15 //calculations
16 beeta = delta_w/w_m;
17 W_B1 = 2*(beeta + 1)*w_m;//angular bandwidth
18 //since beeta >>1,therefore
19 W_B1 = 2*delta_w//angular bandwidth
20 //WB=W_B1
21 f_B = W_B1/(2*%pi)//bandwidth in Hz
22
23 //result
24 printf("\n\nBandwidth = %.2 f Hz",f_B);

```

Scilab code Exa 5.44 modulation index and bandwidth of FM signal

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.44(PAGENO 263)");
8
9 //given
10 f_c = 20*10^6//carrier frequency
11 delta_f = 100*10^3//frequency deviation
12 f_m1 = 1*10^3//modulation index for first case
13 f_m2 = 100*10^3//modulation index for second case
14 f_m3 = 500*10^3//modulation index for third case
15
16 //calculations
17 beeta1 = delta_f/f_m1
18 beeta2 = delta_f/f_m2

```



```

19 beeta3 = delta_f/f_m3
20 m_f1 = delta_f/f_m1//modulation index for first case
21 m_f2 = delta_f/f_m2//modulation index for second
    case
22 m_f3 = delta_f/f_m3//modulation index for third case
23 f_B1 = 2*delta_f//bandwidth for first case since it
    is a WBFM signal
24 f_B2 = 2*(beeta2 + 1)*f_m2//bandwidth for second
    case
25 f_B3 = 2*f_m3//bandwidth for third case since it is
    a NBFM signal
26
27 //results
28 printf("\n\ni.a.Modulation index for first case = %
    .2f ",m_f1);
29 printf("\n\n b.Bandwidth for first case = %.2f Hz",
    f_B1);
30 printf("\n\nii.a.Modulation index for second case =
    %.2f ",m_f2);
31 printf("\n\n b.Bandwidth for second case = %.2f Hz"
    ,f_B2);
32 printf("\n\niii.a.Modulation index for third case =
    %.2f ",m_f3);
33 printf("\n\n b.Bandwidth for third case = %.2f Hz",
    f_B3);
34 printf("\n\nNote:Their is error in first case
    modulating frequency in text book")

```

Scilab code Exa 5.45 bandwidth of angle modulated signal

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;

```

```

6  clc;
7  printf("EXAMPLE 5.45(PAGENO 263)");
8
9  //given
10 //x_c(t) = 10*cos(w_c*t + 3*sin(w_m*t))
11 //comparing with std eqn of PM signal x_PM(t) = A*
    cos(w_c*t + k_p*m(t))
12 //m(t) = a_m*sin(w_m*t)
13 //beeta = k_p*a_m
14 beeta = 3;
15 f_m1 = 1*10^3//modulating frequency for first case
16 f_m2 = 2*10^3//modulating frequency for second case
17 f_m3 = 500//modulating frequency for third case
18
19 //calculations
20 f_B1 = 2*(beeta + 1)*f_m1//bandwidth for first case
21 f_B2 = 2*(beeta + 1)*f_m2//bandwidth for first case
22 f_B3 = 2*(beeta + 1)*f_m3//bandwidth for first case
23
24 //results
25 printf("\n\ni.Bandwidth for first case = %.2f Hz",
    f_B1);
26 printf("\n\nii.Bandwidth for second case = %.2f Hz",
    f_B2);
27 printf("\n\niii.Bandwidth for third case = %.2f Hz",
    f_B3);

```

Scilab code Exa 5.46 bandwidth of FM signal

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;

```

```

7 printf("EXAMPLE 5.45(PAGENO 264)");
8
9 //given
10 //x_cFM(t) = 10*cos(w_c*t + 3*sin(w_m*t))
11 //by omparing with standard equation i.e A*cos(w_c*t
    + beta*sin(w_m*t))
12 //we get
13 beta = 3
14
15 //calculations
16 //first case
17 f_m1 = 1*10^3//modulating frequency for first case
18 f_B1 = 2*(beta +1)*f_m1//bandwidth for first case
19 //second case
20 beta2 = 3/2//beta for second case
21 f_m2 = 2*10^3//modulating frequency for second case
22 f_B2 = 2*(beta2 +1)*f_m2//bandwidth for second case
23 //third case
24 beta3 = 6//beta for third case
25 f_m3 = .5*10^3//modulating frequency for third case
26 f_B3 = 2*(beta3 +1)*f_m3//bandwidth for third case
27
28 //results
29 printf("\n\ni.Bandwidth for first case = %.2f Hz",
    f_B1);
30 printf("\n\nii.Bandwidth for second case = %.2f Hz",
    f_B2);
31 printf("\n\niii.Bandwidth for third case = %.2f Hz",
    f_B3);

```

Scilab code Exa 5.47 bandwidth and frequency deviation of FM signal

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5

```

```

4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.47(PAGENO 264)");
8
9 //given
10 f_m = 2*10^3//modulating frequency for first case
11 delta_f1 = 5*10^3//frequency deviation for first
    case
12 f_m1 = 1*10^3//modulating frequency for second case
13 //beeta = (k_f*a_m)/(w_m) = delta_f/f_m
14
15 //calculations
16 beeta = delta_f1/f_m
17 f_B1 = 2*(beeta + 1)*f_m//bandwidth for first case
18 //beeta1 = (k_f*3*a_m)/(0.5*w_m) = delta_f/f_m
    therefore
19 beeta1 = 6*beeta
20 delta_f2 = beeta1 * f_m1 //frequency deviation for
    second case
21 f_B2 = 2*(beeta1 + 1)*f_m1//bandwidth for second
    case
22
23 //results
24 printf("\n\ni. Bandwidth for first case = %.2f Hz",
    f_B1);
25 printf("\n\nii.a. Frequency deviation for second case
    =%.2f Hz", delta_f2);
26 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA

```

```

3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.48(PAGENO 264)");
8
9 //given
10 delta_f = 75*10^3//frequency deviation
11 f_M = 15*10^3//modulating frequency
12
13 //calculations
14 //we have  $w_m = 2*\pi*f_M$  where  $f_M = 15\text{KHz}$ ,we get
15 D = delta_f/f_M//deviation ratio
16 //by using thr given formula , the bandwidth will be
17 f_B1 = 2*(D+2)*f_M
18 //Using Carson's rule , the bandwidth will be
19 f_B2 = 2*(D+1)*f_M
20
21 //results
22 printf("\n\ni.Bandwidth calculation using thr given
    formula = %.2f Hz",f_B1)
23 printf("\n\nii.Bandwidth calculation using the
    carson rule = %.2f Hz",f_B2)

```

Scilab code Exa 5.49 frequency multiplication and maximum frequency deviation

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.49(PAGENO 265)");
8
9 //given

```

```

10 //x_NBFM(t) = A*cos(w_c*t + sin(w_m*t))
11 f_c = 200*10^3//carrier frequency
12 f_m_max = 15*10^3//maximum modulating frequency
13 f_m_min = 50//minimum modulating frequency
14 delta_f = 75*10^3//maximum frequency deviation
15
16 //calculations
17 beeta_min = delta_f/f_m_max;
18 beeta_max = delta_f/f_m_min;
19 //if beeta_1 =.5, where beeta_1 is the input beeta,
    then the required frequency multiplication will
    be
20 beeta_1 = .5
21 n = beeta_max/beeta_1//frequency multiplication
22 delta_f1 = delta_f/n//maximum allowed frequency
    deviation
23
24 //results
25 printf("\n\ni.Frequency multiplication = %.2f ",n);
26 printf("\n\nii.Maximum allowed frequency deviation =
    %.2f Hz",delta_f1)

```

Scilab code Exa 5.50 maximum frequency deviation and carrier frequency for given m

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.50(PAGENO 265)");
8
9 //given
10 f_1 = 200*10^3//frequency applied at first stage
11 delta_f1 = 25//frequency deviation at first stage

```

```

12 n1 = 64//frequency multiplication at first stage
13 n2 = 48//frequency multiplication at second stage
14 f_L0 = 10.8*10^6//frequency of oscillator as shown
    if block diagram
15
16 //calculations
17 delta_f = delta_f1*n1*n2//maximum frequency
    deviation
18 f_2 = n1*f_1//frequency applied at second stage
19 f_3a = f_2 + f_L0//frequency applied the third stage
20 f_3b = f_2 - f_L0//frequency applied the third stage
21 f_c1 = n2*f_3a//carreir frequency for maximum f_3
22 f_c2 = n2*f_3b//carreir frequency for minimum f_3
23
24 //results
25 printf("\n\ni.Maximum frequency deviation =%.2f Hz",
    delta_f);
26 printf("\n\nii.a.Carrier frequency for maximum f_3 =
    %.2f Hz",f_c1);
27 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.51(PAGENO 266)");
8
9 //given
10 f_c = 108*10^6//carrier frequency
11 f_1 = 2*10^5//crystal oscillator frequency

```

```

12 beta = .2//phase deviation
13 f_m = 50//minimum frequency
14 delta_f = 75*10^3//frequency deviation
15 n_2 = 150
16
17 //calculations
18 delta_f1 = beta * f_m;
19 n_12 = delta_f /delta_f1;
20 //f_2 = n_1*f_1 = n_1 * 2*10^5Hz
21 //assuming down conversions , we have
22 //f_2 - f_LO = (f_c/n_2)
23 //thus
24 f_LO = ((n_12*f_1) - f_c)/n_2;
25 n_1 = n_12/n_2
26
27 //results
28 printf("\n\n n_1 = %.2f",n_1)
29 printf("\n\n Mixer oscillator frequency= %.2f hz",
    f_LO);

```

Scilab code Exa 5.52 frequency multiplication

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 5
4 //ANGLE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 5.52(PAGENO 266)");
8
9 //given
10 delta_f = 50//frequency deviation
11 delta_f2 = 20*10^3//frequency deviation for
    sinusoidal FM wave i.e second case
12 f_m1 = 120//modulating frequency for first case

```



```

13 f_m2 = 240//modulating frquency for second case
14
15 //calculations
16 //first case
17 delta_f1 = (f_m2/f_m1)*delta_f//frequency deviation
    for sinusoidal PM wave
18 n1 = delta_f2/delta_f1//frequency multiplication for
    sinusoidal PM wave
19 //second case
20 n2 = delta_f2/delta_f//frequency multiplication for
    sinusoidal FM wave
21
22 //results
23 printf("\n\ni.Frequency multiplication for PM wave =
    %.2f ",n1);
24 printf("\n\nii.Frequency multiplication for FM wave
    = %.2f ",n2);

```

Chapter 6

Noise

Scilab code Exa 6.1 rms noise voltage

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 6
4 //NOISE
5 clear all;
6 clc;
7 printf("EXAMPLE 6.1(PAGENO 281)");
8
9 //given
10 R = 10*10^3//resistance of amplifier in ohms
11 T = 273+27//temperature in kelvin
12 B = (20-18)*10^6//bandwidth
13 k = 1.38*10^-23//boltzman's constant
14
15 //calculations
16 V_n = sqrt(4*R*k*T*B);//rms noise voltage
17
18 //result
19 printf("\n\nRms noise voltage = %.10f V",V_n);
```

Scilab code Exa 6.2 rms noise voltage

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 6
4 //NOISE
5 clear all;
6 clc;
7 printf("EXAMPLE 6.2(PAGENO 281)");
8
9 //given
10 R_1 = 300//equivalent noise resistance
11 R_2 = 400//input resistance
12 T = 273+27//temperature in kelvin
13 B = 7*10^6//bandwidth
14 k = 1.38*10^-23//boltzman's constant
15
16 //calculations
17 R_s = R_1 +R_2//effective resistance in series
18 V_nr = sqrt(4*k*T*B*R_s)//rms noise voltage
19
20 //result
21 printf("\n\nRms noise voltage = %.10f V",V_nr)
```

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

```
1
2 //ANALOG AND DIGITAL COMMUNICATION
3 //BY Dr.SANJAY SHARMA
4 //CHAPTER 6
5 //NOISE
6 clear all;
```

```

7  clc;
8  printf("EXAMPLE 6.3(PAGENO 282)");
9
10 //given
11 R_1 = 20*10^3//resistance one
12 R_2 = 50*10^3//resistance two
13 T = 273+15//temperature in kelvin
14 B = 100*10^3//bandwidth
15 k = 1.38*10^-23//boltzman's constant
16
17 //calculations
18 R_s = R_1 +R_2//series effective resistance
19 R_p = (R_1*R_2)/(R_1 + R_2)//parallel effective
    resistance
20 V_1 = sqrt(4*k*T*R_1*B)//noise voltage in R_1
21 V_2 = sqrt(4*k*T*R_1*B)//noise voltage in R_2
22 V_s = sqrt(4*k*T*R_s*B)//noise voltage when
    resistance connected in series
23 V_p = sqrt(4*k*T*R_p*B)//noise voltage when
    resistance connected in parallel
24
25 //results
26 printf("\n\ni.Noise voltage due to R_1 = %.10f V",
    V_1);
27 printf("\n\nii.Noise voltage due to R_2 = %.10f V",
    V_2);
28 printf("\n\niii.Noise voltage due to two resistance
    in series = %.10f V",V_s);
29 printf("\n\niv.Noise voltage due to two resistance
    in parallel = %.10f V",V_p);

```

Scilab code Exa 6.4 equivalent input noise resistance

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA

```

```

3 //CHAPTER 6
4 //NOISE
5 clear all;
6 clc;
7 printf("EXAMPLE 6.4(PAGENO 283)");
8
9 //given
10 A_1 = 10//voltage gain for first stage
11 A_2 = 25//voltage gain for second stage
12 R_i1 = 600//input resistance for first stage in ohms
13 R_eq1 = 1600//equivalent noise resistance for first
    stage
14 R_01 = 27*10^3//Output resistance for first stage
15 R_i2 = 81*10^3//input resistance for second stage
16 R_eq2 = 10*10^3//Equivalent noise resistance for
    second stage
17 R_02 = 1*10^6//output resistance for second case
18
19 //calculations
20 R_1 = R_i1 + R_eq1
21 R_2 = ((R_01*R_i2)/(R_01+R_i2)) + R_eq2
22 R_3 = R_02
23 R_eq = R_1 + (R_2/A_1^2) + R_3/(A_1^2 *A_2^2);
24
25 //results
26 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 volt

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 6
4 //NOISE
5 clear all;

```

```

6  clc;
7  printf("EXAMPLE 6.7(PAGENO 295)");
8
9  //given
10 T = 273 + 17//temperature in kelvin
11 Q = 10//quality factor
12 c  = 10*10^-12//capacitance
13 f_r = 100*10^6//resonate frequency
14 k = 1.38*10^-23//boltzman's constant
15
16 //calculations
17 delta_f = f_r/Q//bandwidth of the tuned circuit
18 w = 2*%pi*f_r;//angular frequency
19 R = 1/(Q*w*c);//resistance
20 V_no = sqrt(4*k*Q^2*T*delta_f*R) //output voltage
21
22 //results
23 printf("\n\nOutput voltge = %.10f V",V_no);

```

Scilab code Exa 6.8 noise figure and equivalent temperature

```

1  //ANALOG AND DIGITAL COMMUNICATION
2  //BY Dr.SANJAY SHARMA
3  //CHAPTER 6
4  //NOISE
5  clear all;
6  clc;
7  printf("EXAMPLE 6.8(PAGENO 297)");
8
9  //given
10 R_a = 50//antenna resistance
11 R_eq = 30//equivalent noise resistance of receiver
12 T_0 = 290//initial temperature in degree kelvin
13 //calculations
14 F = 1+(R_eq/R_a);//noise figure

```

```

15 F_dB = 10*log10(F)//noise figure in decibels
16 T_eq = T_0*(F-1)//equivalent temperature
17
18 //results
19 printf("\n\ni.Noise figure in decibels = %.2f dB",
        F_dB);
20 printf("\n\nii.Equivalent temperature = %.2f degree
        kelvin",T_eq)

```

Scilab code Exa 6.9 noise figure

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 6
4 //NOISE
5 clear all;
6 clc;
7 printf("EXAMPLE 6.9(PAGENO 302)");
8
9 //given
10 R_eq = 2518//equivalent resistance in ohms
11 R_t = 600//input impedance in ohms
12 R_a= 50//output impedence in ohms
13
14 //calculations
15 R_eq1 = R_eq - R_t;
16 F = 1 + (R_eq1/R_a) //noise figure
17 F_dB = 10*log10(F)//noise figure in dB
18
19 //results
20 printf("\n\nNoise figure in dB = %.2f dB",F_dB);
21 printf("\n\nNote:Calculation mistake is their in
        text book in finding noise figure in dB")

```

Scilab code Exa 6.10 overall noise figure

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 6
4 //NOISE
5 clear all;
6 clc;
7 printf("EXAMPLE 6.10(PAGENO 305)");
8
9 //given
10 F_1 = 2//noise figure of first stage in dB
11 A_1 = 12//gain in first stage in dB
12 F_2 = 6//noise figure of second stage in dB
13 A_2 = 10//gain in first second in dB
14
15
16 //calculations
17 F_1ratio = exp((F_1/10)*log(10));//noise figure of
    first stage in ratio
18 F_2ratio = exp((F_2/10)*log(10));//noise figure of
    second stage in ratio
19 A_1ratio = exp((A_1/10)*log(10));//gain of first
    stage in ratio
20 A_2ratio = exp((A_2/10)*log(10));//gain of second
    stage in ratio
21 F = F_1ratio + ((F_2ratio - 1)/(A_1ratio));//Overall
    noise figure
22 F_dB = 10*log10(F);//Overall noise figure in dB
23
24 //results
25 printf("\n\nOverall noise figure = %.2f dB",F_dB );
```

Scilab code Exa 6.11 overall noise figure

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 6
4 //NOISE
5 clear all;
6 clc;
7 printf("EXAMPLE 6.11(PAGENO 306)");
8
9 //given
10 F_1 = 9//noise figure for first stage in dB
11 F_2 = 20//noise figure for second stage in dB
12 A_1 = 15//gain in first stage in dB
13
14 //calculations
15 F_1ratio = exp((F_1/10)*log(10));//noise figure of
    first stage in ratio
16 F_2ratio = exp((F_2/10)*log(10));//noise figure of
    second stage in ratio
17 A_1ratio = exp((A_1/10)*log(10));//gain of first
    stage in ratio
18 F = F_1ratio + ((F_2ratio - 1)/(A_1ratio));
19 F_dB = 10*log10(F);
20
21 //results
22 printf("\n\nOverall noise figure = %.2f dB", F_dB )
    ;
```

Scilab code Exa 6.12 rms noise voltage

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

```

2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 6
4 //NOISE
5 clear all;
6 clc;
7 printf("EXAMPLE 6.12(PAGENO 307)");
8
9 //given
10 f_1 = 18*10^6//lower operating frequency in Hz
11 f_2 = 20*10^6//lower operating frequency in Hz
12 T = 273 + 17//temperature in kelvin
13 R = 10*10^3//input resistance
14 k = 1.38*10^-23//boltzman 's constant
15
16 //calculations
17 B = f_2 - f_1//bandwidth in Hz
18 V_n = sqrt(4*k*B*R*T);//rms noise voltage
19
20 //results
21 printf("\n\nrms noise voltage = %.10f V",V_n);

```

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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 6
4 //NOISE
5 clear all;
6 clc;
7 printf("EXAMPLE 6.14(PAGENO 308)");
8
9 //given
10 A = 60//gain of noiseless amplifier
11 V_n1 = 1*10^-3//output of the amplifier
12 B = 20*10^3//initial bandwidth

```

```

13 B1 = 5*10^3//change in bandwidth
14 k = 1.38*10^-23//boltzman's constant
15 T = 273 + 80//temperature in degree kelvin
16
17 //calculaitons
18 //since the bandwidth is reesuced to 1/4th of its
    value,therefore the noise voltage
19 //will be V_n proportional to sqrt(B)
20 //Hence, the noise voltage at 5KHz will become half
    its value at 20KHz bandwidth i.e,
21 V_n = .5*10^-3//noise voltage in volts
22 V_no = V_n1/A;//noise ouput voltage
23 R = (V_no^2/(4*k * T * B ));//resistance at 80degree
    celcius
24
25 //results
26 printf("\n\ni.Meter reading in volts = %.10f V",V_n)
    ;
27 printf("\n\nii.Resistance at 80 degree celcius = %.2
    f ohms",R);
28 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 6
4 //NOISE
5 clear all;
6 clc;
7 printf("EXAMPLE 6.16(PAGENO 309)");
8
9 //given

```

```

10 A_1 = 10//gain in first stage in dB
11 A_2 = 10//gain in second stage in dB
12 A_3 = 10//gain in third stage in dB
13 F_1 = 6//noise figure for first stage in dB
14 F_2 = 6//noise figure for second stage in dB
15 F_3 = 6//noise figure for third stage in dB
16
17 //calculations
18 F_1ratio = exp((F_1/10)*log(10));//noise figure of
    first stage in ratio
19 F_2ratio = exp((F_2/10)*log(10));//noise figure of
    second stage in ratio
20 F_3ratio = exp((F_3/10)*log(10));//noise figure in
    third stage in ratio
21 A_1ratio = exp((A_1/10)*log(10));//gain of first
    stage in ratio
22 A_2ratio = exp((A_2/10)*log(10));//gain of second
    stage in ratio
23 A_3ratio = exp((A_3/10)*log(10));//gain of third
    stage in ratio
24 F = F_1ratio + ((F_2ratio - 1)/(A_1ratio)) + ((
    F_3ratio - 1)/(A_2ratio*A_1ratio));//Overall
    noise figure
25
26 //results
27 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 6
4 //NOISE
5 clear all;

```

```

6  clc;
7  printf("EXAMPLE 6.17(PAGENO 310)");
8
9  //given
10 G_1 = 10//gain in first stage in dB
11 //noise figure for both the stages are same
12 F_1 = 10//noise figure for first stage in dB
13 F_2 = 10//noise figure for second stage in dB
14
15 //calculations
16 F_1ratio = exp((F_1/10)*log(10));//noise figure of
    first stage in ratio
17 F_2ratio = exp((F_2/10)*log(10));//noise figure of
    second stage in ratio
18 G_1ratio = exp((G_1/10)*log(10));//gain of first
    stage in ratio
19 F = F_1ratio + ((F_2ratio - 1)/(G_1ratio));//Overall
    noise figure
20 F_dB= 10*log10(F)////Overall noise figure in dB
21
22 //results
23
24  printf("\n\nOverall noise figure = %.2f dB", F_dB
    );

```

Scilab code Exa 6.18 overall noise figure and overall gain

```

1  //ANALOG AND DIGITAL COMMUNICATION
2  //BY Dr.SANJAY SHARMA
3  //CHAPTER 6
4  //NOISE
5  clear all;
6  clc;
7  printf("EXAMPLE 6.18(PAGENO 310)");
8

```

```

 9 //given
10 G_1 = 4//gain in first stage in dB
11 G_2 = 10//gain in second stage in dB
12 F_1 = 10//noise figure for first stage in dB
13 F_2 = 10//noise figure for second stage in dB
14
15 //calculations
16 F_1ratio = exp((F_1/10)*log(10));//noise figure of
    first stage in ratio
17 F_2ratio = exp((F_2/10)*log(10));//noise figure of
    second stage in ratio
18 G_1ratio = exp((G_1/10)*log(10));//gain of first
    stage in ratio
19 G_2ratio = exp((G_2/10)*log(10));//gain of second
    stage in ratio
20 F = F_1ratio + ((F_2ratio - 1)/(G_1ratio));//Overall
    noise figure
21 G = log10(G_1ratio *G_2ratio );
22 F_dB= 10*log10(F)/////Overall noise figure in dB
23
24 //results
25 printf("\n\ni. Overall noise figure = %.2f dB", F_dB
    );
26 printf("\n\nii. Overall gain = %.2f dB",G );
27 printf("\n\nNote:There is mistake in calculation of
    overall gain in textbook")

```

Scilab code Exa 6.19 overall noise figure

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 6
4 //NOISE
5 clear all;
6 clc;

```

```

7 printf("EXAMPLE 6.19(PAGENO 310)");
8
9 //given
10 G_1 = 15//gain in first stage in dB
11 F_1 = 9//noise figure for first stage in dB
12 F_2 = 20//noise figure for second stage in dB
13
14 //calculations
15 F_1ratio = exp((F_1/10)*log(10));//noise figure of
    first stage in ratio
16 F_2ratio = exp((F_2/10)*log(10));//noise figure of
    second stage in ratio
17 G_1ratio = exp((G_1/10)*log(10));//gain of first
    stage in ratio
18 F = F_1ratio + ((F_2ratio - 1)/(G_1ratio));//Overall
    noise figure
19 F_dB= 10*log10(F)////Overall noise figure in dB
20
21 //results
22 printf("\n\nOverall noise figure = %.2f dB", F_dB )
    ;

```

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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 6
4 //NOISE
5 clear all;
6 clc;
7 printf("EXAMPLE 6.20(PAGENO 311)");
8
9 //given
10 F_2 = 20//noise figure of receiver in dB
11 G_1 = 40//gain of low noise amplifier in dB

```

```

12 T_e1 = 80//noise temperature of low noise amplifier
    in degree kelvin
13 T_0 = 300//room temperature
14
15 //calculations
16 F_2ratio = exp((F_2/10)*log(10));//noise figure of
    receiver in ratio
17 G_1ratio = exp((G_1/10)*log(10));//gain of low
    noise amplifier
18 T_e2 = (F_2ratio-1)*T_0//noise temperature of the
    receiver in degree kelvin
19 T_e = T_e1 +(T_e2/G_1ratio)//overall noise
    temperature in degree kelvin
20
21 //results
22 printf("\n\ni.Noise Temperature of the receiver = %
    .2f degkelvin ",T_e2);
23 printf("\n\nii.Overall noise temperature = %.2f
    degkelvin",T_e);

```

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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 6
4 //NOISE
5 clear all;
6 clc;
7 printf("EXAMPLE 6.21(PAGENO 311)");
8
9 //given from the figure
10 G_1ratio = 1000//gain of master amplifier
11 G_2ratio = 100//gain of TWT
12 G_3ratio = 10000//gain of mixer and IF amplifier
13 F_2ratio = 4//noise figure of TWT

```



```

14 F_3ratio = 16//noise figure of mixer and IF
    amplifier
15 T_0 = 273 + 17//ambient temperature in degree kelvin
16 T_e1 = 5//temperature of master amplifier in degree
    kelvin
17
18 //calculaitons
19 F_1 = 1 + (T_e1/T_0);//noise figure of master
    amplifier
20 F = F_1 + ((F_2ratio - 1)/(G_1ratio)) + ((F_3ratio -
    1)/(G_2ratio*G_1ratio));//Overall noise figure
21 F_dB = 10*log10(F);//overall noise figure in dB
22 T_e = (F - 1)*T_0;//overall noise temperature of the
    receiver
23
24 //results
25 printf("\n\ni. Overall noise temperature of the
    receiver =%.2f deegreekelvin",T_e);
26 printf("\n\nii. Overall noise figure = %.6f dB",
    F_dB);

```

Chapter 7

Sampling Theory and Pulse Modulation

Scilab code Exa 7.1 Nyquist rate

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //SAMPLING THEORY AND PULSE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 7.1(PAGENO 324)");
8
9 //given
10 //analog signal  $x(t) = 3\cos(50\pi t) + 10\sin(300\pi t) - \cos(100\pi t)$ 
11 //comparing signal with  $x(t) = 3\cos(w_1 t) + 10\sin(w_2 t) - \cos(w_3 t)$ 
12 //therefore
13 w_1 = 50*pi; //first frequency in rad/sec
14 w_2 = 300*pi; //second frequency in rad/sec
15 w_3 = 100*pi; //third frequency in rad/sec
16
17 //calculations
```

```

18 f_1 = w_1/(2*%pi); // first frequency in Hz
19 f_2 = w_2/(2*%pi); //second frequency in Hz
20 f_3 = w_3/(2*%pi); //third frequency in Hz
21 f_m = f_2 //maximum frequency
22 f_s = 2*f_m //nyquist rate for a signal
23
24 // results
25 printf("\n\nNyquist rate = %.2f Hz",f_s);

```

Scilab code Exa 7.2 Nyquist rate and Nyquist interval

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //SAMPLING THEORY AND PULSE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 7.12(PAGENO 325)");
8
9 //given
10 //x(t) = (1/(2*%pi))*cos(4000*%pi*t)*cos(1000*%pi*t)
11 //expanding
12 disp("x(t) = (1/(2*%pi))*cos(4000*%pi*t)*cos(1000*%pi*
    *t)");
13 disp("x(t) = (1/(4*%pi))*2*cos(4000*%pi*t)*cos(1000*
    %pi*t)");
14 disp("x(t) = (1/(4*%pi))*[cos(4000*%pi*t + 1000*pi*t)
    *cos(4000*%pi*t - 1000*%pi*t)]")
15 disp("x(t) = (1/(4*%pi))*[cos(5000*%pi*t + cos(3000*
    %pi*t))]")
16 //by comparing above equation with x(t) = (1/(4*%pi)
    )*[cos(w_1*t) + cos(w_2*t)]
17 w_1 = 5000*%pi
18 w_2 = 3000*%pi

```

```

19
20 //calculations
21 f_1 = w_1/(2*%pi);
22 f_2 = w_2 /(2*%pi);
23 f_m = f_1
24 f_s = 2*f_m//Nyquist rate
25 T_s = 1/f_s//Nyquist interval
26
27 //results
28 printf("\n\nNyquist rate = %.2 f Hz",f_s);
29 printf("\n\nNyquist interval = %.5 f seconds",T_s);

```

Scilab code Exa 7.3 discrete time signal for the given conditions

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //SAMPLING THEORY AND PULSE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 7.13(PAGENO 326)");
8
9 //given
10 //x(t) = 8*cos(200*%pi*t)
11 f= 100//highest frequency component of continuous
    time signal in hertz
12 f_s2 = 400//sampling frequency in hertz for second
    condition
13 f_s3 = 400//sampling frequency in hertz for third
    condition
14 f_s4 = 150//sampling frequency in hertz for fourth
    condition since 0 < f_s4 < f_s2/2
15
16 //calculations
17 NR = 2*f//Nyquist rate

```

```

18 F_1 = f/NR;
19 F_2 = f/f_s2;
20 F_3 = f/f_s3;
21 F_4 = f/f_s4;
22 f_4 = f_s4*F_4;
23
24 //results
25 printf("\n\nThe discrete time signal x(n) for the
      first condition is  $x(n) = 8*\cos(2*3.14*0.2f*n)$ ",
      F_1);
26 printf("\n\nthe discrete time signal x(n) for the
      second condition is  $x(n) = 8*\cos(2*3.14*0.2f*n)$ ",
      F_2);
27 printf("\n\nthe discrete time signal x(n) for the
      third condition is  $x(n) = 8*\cos(2*3.14*0.2f*n)$ ",
      F_3);
28 printf("\n\nThe discrete time signal x(n) for the
      fourth condition is  $x(n) = 8*\cos(2*3.14*0.2f*t)$ ",
      f_4);

```

Scilab code Exa 7.4 Nyquist rate for the continuous signal

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //SAMPLING THEORY AND PULSE MODULATION
5 clear all;
6 clc;
7 printf("EXAMPLE 7.4(PAGENO 327)");
8
9 //given
10 //x(t) = 6*cos(50*%pi*t) + 20*sin(300*%pi*t) - 10*
      cos(100*%pi*t)
11 //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

```

```

12 w_1 = 50*%pi//frequency in rad/sec
13 w_2 =300*%pi//frequency in rad/sec
14 w_3 = 100*%pi//frequency in rad/sec
15
16 //calculations
17 f_1 = w_1/(2*%pi)//frequency in hertz
18 f_2 = w_2/(2*%pi)//frequency in hertz
19 f_3 = w_3/(2*%pi)//frequency in hertz
20 if (f_1 > f_2 & f_1> f_3) then
21     f_max = f_1
22 elseif (f_2 > f_1 & f_2> f_3) then
23     f_max = f_2
24 else (f_3 > f_1 & f_3> f_2) then
25     f_max = f_3
26     end
27 f_s = 2*f_max;//nyquist rate
28
29 //results
30 printf("\n\nNyquist rate for a continuous signal = %
    .2 f Hz",f_s);

```

Chapter 8

Waveform Coding Techniques

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

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //WAVEFORM CODING TECHNIQUES
5 clear all;
6 clc;
7 printf("EXAMPLE 8.2(PAGENO 386)");
8
9 //given
10 f_m = 4.2*10^6//bandwidth of television signal
11 q = 512//quantization levels
12
13 //calculations
14 //number of bits and quantization levels are related
    in binary PCM as  $q = 2^v$ 
15 //where v is code word length
16 v = (log10(q)/log10(2));//code word length
17 BW = v*f_m//transmission channel bandwidth which is
    greater than or equal to obtained value
18 f_s = 2*f_m//sampling frequency which is greater
    than or equal to obtained value
```

```

19 r = v*f_s//signaling rate of final bit rate
20 SbyN_dB = 4.8 + 6*v//output signal to noise ratio
    which is less than or equal to obtained value
21
22 //results
23 printf("\n\ni. Code word length = %.2f bits",v);
24 printf("\n\nii. Transmission bandwidth = %.2f Hz",BW
    );
25 printf("\n\niii. Final bit rate = %.2f bits/sec",r);
26 printf("\n\niv. Output signal to quantization noise
    ratio = %.2f dB",SbyN_dB);
27 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 r

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //WAVEFORM CODING TECHNIQUES
5 clear all;
6 clc;
7 printf("EXAMPLE 8.3(PAGENO 387)");
8
9 //given
10 f_m = 4*10^3//maximum frequency or bans
11 x_max = 3.8//maximun input signal
12 P = 30*10^-3//average power of signal
13 SbyN_dB= 20//signal to noise ratio in db
14
15 //calculations
16 SbyN = exp((SbyN_dB/10)*log(10));
17 v = (log2((SbyN*(x_max)^2)/(3*P))/2);//number of
    bits required per sample
18 BW = 30*v*f_m//transmission channel bandwidth which

```



```

    is greater than or equal to obtained value
19 r=BW*2//wkt signalling rate is two times the
    transmission bandwidth
20
21 //results
22 printf("\n\ni.Number of bits required = %.2f bits",v
    );
23 printf("\n\nii.Bandwidth required for 30 PCM coders
    = %.2f Hz",BW);
24 printf("\n\niii.Signalling rate=%.2f bitspersecond",
    r);
25 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //WAVEFORM CODING TECHNIQUES
5 clear all;
6 clc;
7 printf("EXAMPLE 8.4(PAGENO 388)");
8
9 //given
10 e_max = .001//maximum quantization error
11 x_max = 10//maximum amplitude
12 x_min = -10//minumum amplitude
13 f_m = 100//bandwidth of ;input signal
14
15 //calculations
16 delta = 2*e_max//step size

```

```

17 q = (2*x_max)/delta//quantization levels
18 f_s = 2*f_m//sampling frequency
19 v = log10(q) /log10(2);//number of bits in the PCM
    word
20 r = v * f_s//bit rate required in the PCM signal
    which is greater than or equal to obtained value
21 BW = .5*r//transmission channel bandwidth which is
    greater than or equal to obtained value
22
23 //results
24 printf("\n\ni.Minimum sampling rate required = %.2f
    Hz",f_s);
25 printf("\n\nii.Number of bits in each PCM word = %.2
    f bits",v);
26 printf("\n\niii.Minimum bit rate required in the PCM
    signal =%.2f bits/sec",r);
27 printf("\n\niv.Transmission bandwidth = %.2f Hz",BW
    )
28 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //WAVEFORM CODING TECHNIQUES
5 clear all;
6 clc;
7 printf("EXAMPLE 8.5(PAGENO 389)");
8
9 //given
10 f_m = 3.4*10^3//maximum frequency in the signal

```

```

11 N =24//number of voice signals
12 r = 1.5*10^6//signaling rate
13 v = 8//bits of encoder
14
15 //calculations
16 BW = N * f_m//transmission bandwidth
17 r_1 = r/N//bit rate for one channel
18 f_s = r_1/v//sampling frquency
19
20 //results
21 printf("\n\ni.Transmission bandwidth = %.2 f Hz",BW)
    ;
22 printf("\n\nii.Sampling frequency = %.2 f Hz or
    samples per second",f_s)

```

Scilab code Exa 8.6 message bandwidth and signal to noise ratio

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //WAVEFORM CODING TECHNIQUES
5 clear all;
6 clc;
7 printf("EXAMPLE 8.6(PAGENO 389)");
8
9 //given
10 v = 7//bits of encoder
11 r = 50*10^6//bit rate of the system
12
13 //calculations
14 f_m = r/(2*v)//maximum message bandwidth which is
    less than or equal to obtained value
15 SbyN_dB = 1.8 + 6*v//signal to noise ratio in dB
16
17 //results

```

```

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

```

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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //WAVEFORM CODING TECHNIQUES
5 clear all;
6 clc;
7 printf("EXAMPLE 8.7(PAGENO 390)");
8
9 //given
10 f_m = 3*10^3//maximum frequency
11 M = 16//number of quantization levels
12 q = M//number of quantization levels
13
14 //calculations
15 v = log2(q);//number of bits
16 f_s = 2*f_m//sampling frequency or rate which is
    greater than or equal to obtained value
17 r = v*f_s//bit transmission rate which is greater
    than or equal to obtained value
18
19 //results
20 printf("\n\ni.Number of bits in a codeword = %.2f
    bits",v);
21 printf("\n\nii.Minimum sampling rate = %.2f Hz ",f_s
    );
22 printf("\n\niii.Bit transmission rate =%.2f bits/sec
    ",r);

```

Scilab code Exa 8.8 signal to the noise ratio

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //WAVEFORM CODING TECHNIQUES
5 clear all;
6 clc;
7 printf("EXAMPLE 8.8(PAGENO 391)");
8
9 //given
10 f_m = 3.5*10^3//maximum frequency
11 r = 50*10^3//bit rate
12 v_rms = .2//rms value of input signal
13 R = 1//resistance
14 x_max = 2//maximum peak voltage
15
16 //calculations
17 f_s = 2*f_m;//sampling frequency
18 v = r/f_s;//number of bits
19 P = v_rms^2 / R//Normalized signal power
20 SbyN = ((3*P) * 2^(2*v)) / (x_max^2);//signal to
    noise ratio
21 SbyN_dB = 10*log10(SbyN)//signal to noise ratio in
    dB
22
23 //results
24 printf("\n\ni.Signal to noise ratio in dB = %.2f dB"
    ,SbyN_dB);
25 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

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //WAVEFORM CODING TECHNIQUES
5 clear all;
6 clc;
7 printf("EXAMPLE 8.10(PAGENO 392)");
8
9 //given
10 //x(t) = 3*cos(500*%pi*t)
11 v = 10//number of bits
12 A_m = 3//peak voltage
13 SbyN_2 = 40//signal to noise to noise ratio in
    second condition
14
15 //calculations
16 SbyN = 1.8 +6*v//signal to noise ratio in dB
17 v_2 = (40 - 1.8)/6//number of bits needed for SbyN =
    40
18
19 //results
20 printf("\n\ni.Signal to noise to ratio in dB = %.2f
    dB", SbyN);
21 printf("\n\nii.Number of bits needed for noise ratio
    40 = %.2f bits", v_2);
```

Scilab code Exa 8.11 maximum frequency

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

```

3 //CHAPTER 7
4 //WAVEFORM CODING TECHNIQUES
5 clear all;
6 clc;
7 printf("EXAMPLE 8.11(PAGENO 393)");
8
9 //given
10 v = 7//number of bits
11 r = 56*10^3//signaling rate
12
13 //calculations
14 SbyN = 1.8 +6*v//signal to noise ratio in dB
15 f_s = r/v//sampling frequency
16 f_m = f_s/2//maximum frequency which is less than or
    equal to obtained value
17
18 //results
19 printf("\n\nMaximum frequency = %.2f Hz",f_m)

```

Scilab code Exa 8.13 maximum amplitude

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //WAVEFORM CODING TECHNIQUES
5 clear all;
6 clc;
7 printf("EXAMPLE 8.13(PAGENO 404)");
8
9 //given
10 f_m = 3*10^3//bandwidth or maximum frequency
11 n = 5//system operation times
12 delta = 250*10^-3//step size in volts
13 f_m1 = 2*10^3//given maximum frequency to calculate
    amplitude

```

```

14
15 //calculations
16 NR = 2 * f_m//nyquist rate
17 f_s = n * NR//sampling frequency
18 T_s = 1/f_s//sampling interval
19 A_m =(delta/(2 * %pi * f_m1* T_s))//Maximum
    amplitude
20
21 //result
22 printf("\n\nMaximum amplitude for 2KHz input
    sinusoid = %.2f V",A_m);

```

Scilab code Exa 8.14 signalling rate

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //WAVEFORM CODING TECHNIQUES
5 clear all;
6 clc;
7 printf("EXAMPLE 8.14(PAGENO 406)");
8
9 //given
10 f_s = 8*10^3//sampling rate
11 q = 64//quantization levels
12 delta = 31.25//step size
13
14 //calculations
15 v = log2(q);//no fo bits in the PC
16 f_s= (2*%pi*3*10^3)/delta//signalling rate which
    should be greater than the obtaining value
17
18 //results
19 printf("\n\nSignalling rate = %.2f Hz",f_s);

```

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

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //WAVEFORM CODING TECHNIQUES
5 clear all;
6 clc;
7 printf("EXAMPLE 8.15(PAGENO 407)");
8
9 //given
10 f_m = 2*10^3//maximum frequency
11 f_s = 64*10^3//sampling frequency
12 f_M = 4*10^3//cut off frequency of low pass filter
13
14 //calculation
15 SNR_0 = (3 * f_s^3) / (8 * %pi^2 * f_m^2 * f_M);//
    signal to noise ratio of linear delta modulation
    system
16 SNR_dB = 10*log10(SNR_0);//SNR in dB
17
18 //result
19 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

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //WAVEFORM CODING TECHNIQUES
5 clear all;
```

```

6  clc;
7  printf("EXAMPLE 8.16(PAGENO 407)");
8
9  //given
10 r = 64*10^3//data rate
11 f_s = 8*10^3//sampling frequency
12 N = 8//number of samples
13
14 //calcualtion
15 SNR_q = 1.8 + 6*N//signal to noise ratio
16
17 //result
18 printf("\n\nSignla to noise ratio = %.2f dB",SNR_q)
19 ;
20 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

```

1  //ANALOG AND DIGITAL COMMUNICATION
2  //BY Dr.SANJAY SHARMA
3  //CHAPTER 7
4  //WAVEFORM CODING TECHNIQUES
5  clear all;
6  clc;
7  printf("EXAMPLE 8.17(PAGENO 413)");
8
9  //given
10 r = 36000//bit rate of a channel
11 f_m = 3.2*10^3//maximum frequency
12
13 //calculations
14 f_s = 2*f_m//sampling frequency

```

```

15 v = r/f_s//number of binary digits
16 q = 2^v//quantizing level
17
18 //results
19 printf("\n\ni.Sampling frequency = %.2f Hz",f_s);
20 printf("\n\nii.Number of binary digits = %.2f ",v);
21 printf("\n\niii.Quantizing level = %.2f ",q);
22 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //WAVEFORM CODING TECHNIQUES
5 clear all;
6 clc;
7 printf("EXAMPLE 8.20(PAGENO 415)");
8
9 //given
10 SbyN_0dB = 40//signal to noise ratio in dB
11 SbyN_0 = exp((SbyN_0dB/10)*log(10))//signal to noise
    ratio
12 q = sqrt((2 / 3) * (SbyN_0));//quantizing level
13 v = log2(q)//number of binary bits
14 q_1 = 2^v//number of levels required
15 SbyN_dB1 = 1.76 + 6.02*v//output signal-to-
    quantizing noise ratio in dB
16
17 //results
18 printf("\n\nNumber of required levels = %.2f ",v);
19 printf("\n\nOutput signal-to-quantizing noise ratio
    = %.2f dB",SbyN_dB1);
20 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

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 7
4 //WAVEFORM CODING TECHNIQUES
5 clear all;
6 clc;
7 printf("EXAMPLE 8.21(PAGENO 416)");
8
9 //given
10 SbyN_dB = 30//signal to noise ratio
11 f_s = 8000//sampling rate
12
13 //calculations
14 //for Sbyn_dB = 1.76 + 20*logq
15 x1 = (1 / 20)*(SbyN_dB - 1.76)
16 q1 = exp(x1*log(10))//quantizing level for first
    case
17 v1 = log2(q1)// number of bits for first case
18 f_PCM1 = (v1 / 2) * f_s//minimum required bandwidth
    for first case
19 //for SbyN = 20logq - 10.1
20 x2 = (1/20) * (SbyN_dB + 10.1)
21 q2 = exp(x2*log(10))//quantizing level for second
    case
22 v2 = log2(q2)// number of bits for second case
23 f_PCM2 = (v2 / 2) * f_s//minimum required bandwidth
    for second case
24
25 //results
26 printf("\n\ni.a.Minimum number of quantizing levels
```

```

    for first case = %.2f ",q1);
27 printf("\n\n b.Number of bits for first case =%.2f
    ",v1);
28 printf("\n\n c.Minimum system bandwidth required
    for first case = %.2f hz",f_PCM1);
29 printf("\n\nii.a.Minimum number of quantizing levels
    for second case = %.2f ",q2);
30 printf("\n\n b.Number of bits for second case =%.2
    f ",v2);
31 printf("\n\n c.Minimum system bandwidth required
    for second case = %.2f hz",f_PCM2);
32 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

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 10
4 //DIGITAL MULTIPLEXERS
5 clear all;
6 clc;
7 printf("EXAMPLE 10.1(PAGENO 469)");
8
9 //given
10 X_1 = 4*10^3//first analog signal in Hz
11 X_2 = 4.5*10^3//second analog signal in Hz
12
13 //calculation
14 //the highest frequency component of the composite
    signal consisting among two signal is X_2
15 f_sMIN = 2*X_2;
16
17 printf("\n\nThe minimum value of permissible
    sampling rate = %2f Hz",f_sMIN);
```

Scilab code Exa 10.2 signalling rate and channel bandwidth

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 10
4 //DIGITAL MULTIPLEXERS
5 clear all;
6 clc;
7 printf("EXAMPLE 10.2(PAGENO 469)");
8
9 //given
10 X_1 = 6*10^3//Nyquist rate in Hz obtained the table
11 X_2 = 2*10^3//Nyquist rate in Hz obtained the table
12 X_3 = 2*10^3//Nyquist rate in Hz obtained the table
13 X_4 = 2*10^3//Nyquist rate in Hz obtained the table
14
15 //calculations
16 s = 2000//speed of rotation
17 X1 = 3*s//number of samples produced per second for
    first signal
18 X2 = 1*s//number of samples produced per second for
    second signal
19 X3 = X2//number of samples produced per second for
    third signal
20 X3 = X2//number of samples produced per second for
    fourth signal
21 SR = X1 + 3*X2//signalling rate
22 BW = .5*SR//minimum channel bandwidth
23
24 //results
25 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")
```

```

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

```

Scilab code Exa 10.3 spacing between successive pulses of multiplexed signal

```

1
2 //ANALOG AND DIGITAL COMMUNICATION
3 //BY Dr.SANJAY SHARMA
4 //CHAPTER 10
5 //DIGITAL MULTIPLEXERS
6 clear all;
7 clc;
8 printf("EXAMPLE 10.3(PAGENO 470)");
9
10 //given
11 SR = 8000//sampling rate in samples per second
12 T = 1*10^-6//pulse duration
13 f = 3.4*10^3//highest frequency component
14
15 //calculations
16 //second case
17 NR = 2*f//Nyquist rate of sampling
18 T2 = 1/NR//time taken for one rotation of commutator
19
20
21 //results
22 printf("\n\nsampling rate for first condition =%.2f"
    ,SR);
23 printf("\n\nThere are 24 voice signals + 1
    synchronizing pulse")
24 printf("\n\nPulse width of each voice channel and
    synchronizing pulseis 1 microseconds ")
25 printf("\n\nNow, time taken by the commutator for 1
    rotation =1/8000 = 125*10^-6 seconds")

```



```

26 printf("\n\nNumber of pulses produced in one
    rotation = 24 + 1 = 25");
27 printf("\n\nTherefore, the leading edges of the
    pulses are at 125/25 = 5*10-6 seconds distance")
28 printf("\n\nNyquist rate for second condition = %.2f
    Hz",NR);
29 printf("\n\nTime taken for one rotation of
    commutator = %.8f seconds",T2);
30 printf("\n\nTherefore, 147*10-6 seconds corresponds
    to 25 pulses");
31 printf("\n\ntherefore, 1 pulse corresponds to
    5.88*10-6 seconds");
32 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 tau = 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 10
4 //DIGITAL MULTIPLEXERS
5 clear all;
6 clc;
7 printf("EXAMPLE 10.4(PAGENO 471)");
8
9 //given
10 N = 6//number of channels
11 f_m = 5*103//bandwidth of each channel
12
13 //calculations
14 SR1= 2*f_m//minimum sampling rate
15 SR = N*SR1//sampling rate

```

```
16 BW =N*f_m//minimum channel bandwidth
17
18 //results
19 printf("\n\nSignaling rate =%.2f bits per second",SR
    );
20 printf("\n\nMinimum channel bandwidth = %.2f Hz",BW)
    ;
```

Chapter 11

Information Theory

Scilab code Exa 11.1 information content of given symbols

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.1(PAGENO 488)");
8 //given
9
10 Px_1=1/2; //probability 1
11 Px_2=1/4; //probability 2
12 Px_3=1/8; //probability 3
13 Px_4=1/8; //probability 4
14
15 //calculations
16 Ix_1 = log2(1/(Px_1)) //information content in first
    probability
17 Ix_2 = log2(1/(Px_2)) //information content in first
    probability
18 Ix_3 = log2(1/(Px_3)) //information content in first
    probability
```

```

19 Ix_4 = log2(1/(Px_3))//information content in first
    probability
20
21 //results
22 printf("\n\ni. Information content of first symbol =
    %.2f bit",Ix_1);
23 printf("\n\nii. Information content of second symbol
    = %.2f bits",Ix_2);
24 printf("\n\niii. Information content of third
    symbol = %.2f bits",Ix_3);
25 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.2(PAGENO 488)");
8 //given
9 Px_i = 1/4//probability of a symbol
10
11 //calculation
12 Ix_i = (log(1/Px_i))/log(2)//formula for amount of
    information of a symbol
13
14 //result
15 printf("\n\ni. Amount of information = %.2f bits",
    Ix_i)

```

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

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.3(PAGENO 489)");
8 //given
9 //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
10 Px_1 = 1/2//probability of zero level
11 Px_2 = 1/2//probability of first level
12
13 //calculations
14 Ix_1 = log2(1/Px_1)//amount of information of zero
   level with base 2
15 Ix_2 = log2(1/Px_2)//amount of information of first
   level with base 2
16 Ix_1= log(1/Px_1)/log(2)//amount of information
   content with base 10
17 Ix_2 = Ix_1
18
19 //result
20 printf("\n\ni.Amount of information content wrt
   binary PCM 0 = %.2f bit",Ix_1)
21 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

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

```

2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.4(PAGENO 489)");
8 //given
9 Px_1 = 1/4//probability wrt to binary PCM '0'
10 Px_2 = 3/4//probability wrt to binary PCM '1'
11
12 //calculations
13 Ix_1 = log2(1/Px_1)//amount of information of zero
    level with base 2
14 Ix_2 = log2(1/Px_2)//amount of information of first
    level with base 2
15 Ix_1= log(1/Px_1)/log(2)//amount of information
    content with base 10
16 Ix_2= log(1/Px_2)/log(2)//amount of information
    content with base 10
17
18 //results
19 printf("\n\ni.Amount of information carried wrt to
    binary PCM 0 = %.2f bits",Ix_1);
20 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.9(PAGENO 492)");

```

```

8 //given
9 Px_1 = .4//probability of first symbol
10 Px_2 = .3//probability of second symbol
11 Px_3 = .2//probability of third symbol
12 Px_4 = .1//probability of fourth symbol
13
14 //calculations
15 H_X = -Px_1*log2(Px_1)-Px_2*log2(Px_2)-Px_3*log2(
    Px_3)-Px_4*log2(Px_4); //entropy
16 Px1x2x1x3 = Px_1*Px_2*Px_1*Px_3; //product of
    probabilities
17 Ix1x2x1x3 =-log2(Px1x2x1x3); //information of four
    symbols
18 Px4x3x3x2 = Px_4*Px_3*Px_3*Px_2; //product of
    probabilities
19 Ix4x3x3x2 = -log2(Px4x3x3x2); //information of four
    symbols
20
21 //results
22 printf("\n\ni.Entorpy = %.2f bits/symbol",H_X);
23 printf("\n\nii.Amount of information contained in
    x1x2x1x3 = %.2f bits/symbol",Ix1x2x1x3);
24 printf("\n\nThus, Ix1x2x1x3 < 7.4[=4*H_X] bits/symbol")
25 printf("\n\niii.Amount of information contained in
    x4x3x3x2 =%.2f bits/symbol",Ix4x3x3x2);
26 printf("\n\nThus we conclude that\nIx4x3x3x2 > 7.4[=4*
    H_X] bits/symbol")

```

Scilab code Exa 11.12 average rate of information conveyed

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;

```

```

6  clc;
7  printf("EXAMPLE 11.12(PAGENO 495)");
8  //given
9  n = 2*10^6//elements od black and white TV picture
10 m = 16//brightness levels of black and white TV
    picture
11 o = 32//repeated rate of pictures per second
12
13 //calculations
14 Px_i = 1/m//probability of  brightness levels of
    picture
15 H_X = 0;
16 for  i= 1:16
17     H_Xi = (-1/(1/Px_i))*log2(1/(1/Px_i));
18     H_X = H_X +H_Xi;
19 end
20 r = n*o//rate of symbols generated
21 R = r*H_X//average rate of information convyed
22
23 //results
24 printf("\n\ni. Average rate of information convyed =
    %.2f bits/seconds",R)

```

Scilab code Exa 11.13 average information rate

```

1  //ANALOG AND DIGITAL COMMUNICATION
2  //BY Dr.SANJAY SHARMA
3  //CHAPTER 11
4  //Information Theory
5  clear all;
6  clc;
7  printf("EXAMPLE 11.13(PAGENO 495)");
8  //given
9  t_dot = .2//duration of dot symbol
10 t_dash = .6//duration of dash symbol

```



```

11 t_space = .2//time between the symbols
12 //wkt sum of the probability is 1 i.e P_dot + P_dash
    = 1 hence
13 //P_dot = 2*P_dash weget
14 P_dot = 2/3//probality of dot symbol
15 P_dash = 1/3//probality of dash symbol
16
17 //calculations
18 H_X = -P_dot*log2(P_dot)-P_dash*log2(P_dash);//
    entropy
19 T_s = P_dot*t_dot + P_dash*t_dash +t_space;//average
    time per symbol
20 r = 1/T_s;//average symbol rate
21 R = r*H_X;//average information rate of the
    telegraph sourece
22
23 //result
24 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.14(PAGENO 496)");
8 //given
9 //given symbols are equally likely all the symbols
    the probabilities are same
10 Px_1 = 1/8;//probability of first symbol
11 Px_2 = 1/8;//probability of second symbol
12 Px_3 = 3/8;//probability of third symbol

```

```

13 Px_4 = 3/8; //probability of fourth symbol
14 f_m = poly(0, "f_m");
15 r = 2 //average symbol rate from problem 11.14
16
17 //calculaitons
18 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
19 R = H_X*r; //information rate
20
21 //results
22 printf("\n\ni.Entropy = %.2f bits/symbol", H_X)
23 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.15(PAGENO 497)");
8 //given
9 //given symbols are equally likely all the symbols
    the probabilities are same
10 Px_1 = 1/4; //probability of first symbol
11 Px_2 = 1/4; //probability of second symbol
12 Px_3 = 1/4; //probability of third symbol
13 Px_4 = 1/4; //probability of fourth symbol
14 f_m = poly(0, "f_m");
15 r = 2 //average symbol rate from problem 11.14
16
17 //calculaitons
18 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
19 R = H_X*r; //information rate
20
21 //results
22 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.16(PAGENO 498)");
8 //given
9 Px_1 = 1/2; //probability of first symbol
10 Px_2 = 1/4; //probability of second symbol
11 Px_3 = 1/8; //probability of third symbol
12 Px_4 = 1/16; //probability of fourth symbol
13 Px_4 = 1/16; //probability of fifth symbol
14 T_b = 1*10^-3 //time required for emission of each
    symbol
15 r = 1/(T_b) //symbol rate
16
17 //calculations
18 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);
19 R = r*H_X; //information rate
20
21 //results
22 printf("\n\ni.Entropy of five symbols = %.2f bits/
    symbol", H_X);

```

```
23
24 printf("\n\nRate of information = %.2f bits/sec",
    R);
```

Scilab code Exa 11.17 rate of information

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.17(PAGENO 498)");
8 //given
9 Px_1 = 1/2; //probability of first symbol
10 Px_2 = 1/4; //probability of second symbol
11 Px_3 = 1/8; //probability of third symbol
12 Px_4 = 1/16; //probability of fourth symbol
13 Px_5 = 1/16; //probability of fifth symbol
14 r = 16 //outcomes per second
15
16 //calculations
17 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);
18 R = r*H_X; //information rate
19
20 //result
21 printf("\n\nRate of information = %.2f bits/sec",R);
```

Scilab code Exa 11.18 rate of information

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

```

2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.18(PAGENO 499)");
8 //given
9 Px_1 = 1/4;//probability of first symbol
10 Px_2 = 1/5;//probability of second symbol
11 Px_3 = 1/5;//probability of third symbol
12 Px_4 = 1/10;//probability of fourth symbol
13 Px_5 = 1/10;//probability of fifth symbol
14 Px_6 = 1/20;//probability of sixth symbol
15 Px_7 = 1/20;//probability of seventh symbol
16 Px_8 = 1/20;//probability of eighth symbol
17 f_m = 10*10^3//frequency of transmitting symbol
18
19 //calculations
20 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
21 f_s = 2*f_m//sampling frequency
22 r = f_s//sampling frequency equal to rate of
    transmission
23 R = r*H_X;//information rate
24
25 //result
26 printf("\n\nRate of information = %.2f bits/sec",R);
27 printf("\n\nNote:Their mistake in calculation of H_X
    in textbook")

```

Scilab code Exa 11.19 channel matrix and joint probabilities

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

```

2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.19(PAGENO 502)");
8 //given
9 //from fig
10 P_X = [.5 .5]//x matrix
11 P_Xd = [.5 0; 0 .5]//diagonal x matrix
12 //calculations
13 P_YX = [.9 .1; .2 .8];//yx matrix representation of
    given fig
14 P_Y = P_X*P_YX//y matrix
15 P_XY = P_Xd * P_YX//xy matrix
16
17 //results
18 printf("\n\ni.Channel matrix of the channelP_YX ");
19 disp(P_YX);
20 printf("\n\nii.a.P(y1) = %.2f",P_Y(1,1));
21 printf("\n\n    b.P(y2) = %.2f",P_Y(1,2));
22 printf("\n\niii.a.P(x1,y2) = %.2f",P_XY(1,2));
23 printf("\n\n    b.P(x2,y1) = %.2f",P_XY(2,1));

```

Scilab code Exa 11.20 channel matrix and probabilities

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.20(PAGENO 503)");
8 //given
9 P_X = [.5 .5]//x matrix

```

```

10
11 //calculations
12 P_YX = [.9 .1; .2 .8]; //yx matrix representation of
    given fig
13 P_ZY = [.9 .1; .2 .8] //zy matrix representation of
    given fig
14 P_Y = P_X *P_YX //y matrix
15 P_ZX = P_YX * P_ZY //zx matrix
16 P_Z = P_X *P_ZX //z matrix
17
18
19 //results
20 printf("\n\ni.Channel matrix of the channelP_ZX ");
21 disp(P_ZX);
22 printf("Matrix P(Z)")
23 disp(P_Z);
24 printf("\n\na.P(Z1) = %.2 f",P_Z(1,1));
25 printf("\n\nb.P(Z2) = %.2 f",P_Z(1,2));

```

Scilab code Exa 11.21 probabilities associated with the channel outputs

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.21(PAGENO 504)");
8 //given
9 P_X = [.5 .5] //x matrix
10 P_YX = [.8 .2 0 ; 0 .2 .8]; //yx matrix
    representation of given fig
11
12 //calculations
13 P_Y = P_X*P_YX;

```

```

14
15 //results
16 printf("\n\norobability associated with the channel
        outputs for p=.2 is")
17 disp(P_Y)
18 printf("\n\na.P(Y1) = %.2 f",P_Y(1,1));
19 printf("\n\nb.P(Y2) = %.2 f",P_Y(1,2));
20 printf("\n\nC.P(Y3) = %.2 f",P_Y(1,3));

```

Scilab code Exa 11.28 mutual information

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.21(PAGENO 504)");
8
9 //given
10 //wkt P_Y = P_X*P_YX from previous problems
11 alfa = .5
12 P_1 = .1//probability for first case
13 P_2 = .5//probability for second case
14
15 //calculations
16 P_X = [alfa alfa];
17 //first case
18 P_YX = [1-P_1 P_1;P_1 1-P_1];
19 P_Y1 = P_X*P_YX;
20 H_Y1 = -P_Y1(1,1)*log2(P_Y1(1,1))-P_Y1(1,2)*log2(
        P_Y1(1,2));
21 Q_1 = P_1*log2(P_1) + (1-P_1)*log2(1-P_1)//from
        proof
22 I_XY1 = 1 + Q_1;

```



```

23 //second case
24 P_YX = [1-P_2 P_2;P_2 1-P_2];
25 P_Y2 = P_X*P_YX;
26 H_Y2 = -P_Y2(1,1)*log2(P_Y2(1,1))-P_Y2(1,2)*log2(
    P_Y2(1,2));
27 Q_2 = P_2*log2(P_2) + (1-P_2)*log2(1-P_2)//from
    proof
28 I_XY2 = 1 + Q_2;
29
30 //results
31 printf("\n\nI_XY for the first case = %.2f",I_XY1);
32 printf("\n\nI_XY for the second case = %.2f",I_XY2);

```

Scilab code Exa 11.32 entropy for the given probability density function

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.32(PAGENO 518)");
8 //given
9 a1 = 1
10 a2 = 2
11 a3 = .5
12
13 //calculations
14 H_X1 = log2(a1);//Entropy for first case
15 H_X2 = log2(a2);//Entropy for second case
16 H_X3 = log2(a3);//Entropy for third case
17
18 //results
19 printf("\n\ni.Entropy for first case = %.2f ",H_X1);
20 printf("\n\nii.Entropy for second case = %.2f",H_X2)

```

```
    ;
21 printf("\n\niii.Entropy for third case = %.2f ",H_X3
    );
```

Scilab code Exa 11.35 capacity of the channel

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.35(PAGENO 520)");
8 //given
9 B = 4000//bandwidth of AWGN channel
10 S = .1*10^-3//power of signal
11 neta = 2*10^-12//spectral dencity
12 N = neta*B;//power
13
14 //calculations
15 C = B * log2(1 + (S/N));//capacity of channel
16
17 //result
18 printf("\n\nCapacity of channel = %.2f b/s",C);
```

Scilab code Exa 11.37 efficiency of the code and code redundancy

```
1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
```

```

7 printf("EXAMPLE 11.37(PAGENO 524)");
8 //given
9 Px_1 = 0.9//probability of first symbol
10 Px_2 = 0.1//probability of second symbol
11 n1 = 1//length of the code for x_1
12 n2 =1//length of code for x_2
13
14 //calculations
15 //we know that the average code length L per symbol
16 L = Px_1*n1 + Px_2*n2//code length
17 H_X = -Px_1*log2(Px_1) - Px_2*log2(Px_2) //entropy
18 neta = H_X/L//efficiency
19 neta1 = neta*100//neta in percentage
20 gama = 1 - neta//redundancy
21 gama1 = gama*100//gama in percentage
22
23 //results
24 printf("\n\ni.Efficiency of code = %.2f percent",
        neta1);
25 printf("\n\nii.Code redundancy = %.2f percent ",
        gama1)

```

Scilab code Exa 11.38 efficiency of the code and code redundancy

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.38(PAGENO 524)");
8
9 //given
10 Px_1 = 0.81//probability of first symbol
11 Px_2 = .09//probability of second symbol

```

```

12 Px_3 = .09//probability of third symbol
13 Px_4 = 0.01//probability of forth symbol
14 n1 = 1//length of code for a_1
15 n2 =2//length of code for a_2
16 n3 = 3//length of code for a_3
17 n4 = 3//length of code for a_4
18
19 //calculations
20 //we know that the average code length L per symbol
21 L = Px_1*n1 + Px_2*n2 + Px_3*n3 + Px_4*n4 //code
    length
22 H_X = -Px_1*log2(Px_1) - Px_2*log2(Px_2) - Px_3*log2
    (Px_3) - Px_4*log2(Px_4)//entropy
23 neta = H_X/L//efficiency
24 neta1 = neta*100//neta in percentage
25 gama = 1 - neta//redundancy
26 gama1 = gama*100//gama in percentage
27
28 //results
29 printf("\n\ni.Efficiency of code = %.2f percent",
    neta1);
30 printf("\n\nii.Code redundancy = %.2f percent ",
    gama1)

```

Scilab code Exa 11.44 efficiency of shannon fano code

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.44(PAGENO 529)");
8
9 //given

```

```

10 P_x1 = 1/2//probability of first symbol
11 P_x2 = 1/4//probability of second symbol
12 P_x3 = 1/8//probability of third symbol
13 P_x4 = 1/8//probability of fourth symbol
14 n1 = 1
15 n2 = 2
16 n3 = 3
17 n4 = 3
18
19 //calculations
20 I_x1 = -log2(P_x1);
21 I_x2 = -log2(P_x2);
22 I_x3 = -log2(P_x3);
23 I_x4 = -log2(P_x4);
24 H_x = P_x1*I_x1 + P_x2*I_x2 + P_x3*I_x3 + P_x4*
      I_x4;
25 L = P_x1*n1 + P_x2*n2 + P_x3*n3 + P_x4*n4;
26 neta = H_x/L;
27 P_neta = neta*100//efficiency in percentage
28
29 //results
30 printf("\n\nEfficiency = %.2f",neta);
31 printf("\n\nEfficiency in percentage = %.2f percent"
      ,P_neta);

```

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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.46(PAGENO 532)");
8

```

```

9 //given
10 P_x1 = .4//probability of first signal
11 P_x2 = .19//probability of second signal
12 P_x3 = .16//probability of third signal
13 P_x4 = .15//probability of fourth signal
14 P_x5 = .1//probability of fifth signal
15 n1 = 1//number of bits in code obtained from table
    givenn textbook
16 n2 = 2//number of bits in code obtained from table
    givenn textbook
17 n3 = 2//number of bits in code obtained from table
    givenn textbook
18 n4 = 3//number of bits in code obtained from table
    givenn textbook
19 n5 = 3//number of bits in code obtained from table
    givenn textbook
20
21 //calculations
22 I_x1 = -log2(P_x1);
23 I_x2 = -log2(P_x2);
24 I_x3 = -log2(P_x3);
25 I_x4 = -log2(P_x4);
26 I_x5 = -log2(P_x5);
27 H_x = P_x1*I_x1 + P_x2*I_x2 + P_x3*I_x3 + P_x4*
    I_x4 + P_x5*I_x5;//entropy
28 L1 = P_x1*n1 + P_x2*n2 + P_x3*n3 + P_x4*n4 + P_x5*n5
    ;
29 neta1 = H_x/L1;
30 P_neta1 = neta1*100//efficiency in percentage using
    Shannon Fano code
31 L2 = P_x1*1 + (P_x2 + P_x3 +P_x4 +P_x5 )*3
32 neta2 = H_x/L2;
33 P_neta2 = neta2*100//efficiency in percentage using
    huffman code
34
35 //results
36 printf("\n\nEfficiency in percentage using Shannon
    Fano code = %2f percent",P_neta1)

```

```

37 printf("\n\nEfficiency in percentage using huffman
    code    = %2f percent",P_neta2)
38 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.44(PAGENO 532)");
8
9 //given
10 P_x1 = .05//probability of first signal
11 P_x2 = .15//probability of second signal
12 P_x3 = .2//probability of third signal
13 P_x4 = .05//probability of fourth signal
14 P_x5 = .15//probability of fifth signal
15 P_x6 = .3//probability of sixth signal
16 P_x7 = .1//probability of seventh signal
17 n1 = 4//number of bits in code obtained from table
    given textbook
18 n2 = 3//number of bits in code obtained from table
    given textbook
19 n3 = 2//number of bits in code obtained from table
    given textbook
20 n4 = 4//number of bits in code obtained from table
    given textbook
21 n5 = 3//number of bits in code obtained from table
    given textbook
22 n6 = 2//number of bits in code obtained from table
    given textbook

```

```

23 n7 = 3//number of bits in code obtained from table
    given textbook
24
25 //calculations
26 I_x1 = -log2(P_x1);
27 I_x2 = -log2(P_x2);
28 I_x3 = -log2(P_x3);
29 I_x4 = -log2(P_x4);
30 I_x5 = -log2(P_x5);
31 I_x6 = -log2(P_x6);
32 I_x7 = -log2(P_x7);
33 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
34 L = P_x1*n1 + P_x2*n2 + P_x3*n3 + P_x4*n4 + P_x5*n5
    + P_x6*n6 + P_x7*n7;
35 neta = (H_x*100)/L//Efficiency in percentage
36
37 //results
38 printf("\n\nEfficiency in percentage = %.2f percent"
    ,neta);

```

Scilab code Exa 11.49 variance of codeword length

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.49(PAGENO 534)");
8
9 //given
10 P_x1 = .4//probability of first signal
11 P_x2 = .2//probability of second signal

```



```

12 P_x3 = .8//probability of third signal
13 P_x4 = .08//probability of fourth signal
14 P_x5 = .02//probability of fifth signal
15 n1 = 2//number of bits in code obtained from table
    given textbook
16 n2 = 3//number of bits in code obtained from table
    given textbook
17 n3 = 1//number of bits in code obtained from table
    given textbook
18 n4 = 4//number of bits in code obtained from table
    given textbook
19 n5 = 4//number of bits in code obtained from table
    given textbook
20
21 //calculations
22 L = P_x1*n1 + P_x2*n2 + P_x3*n3 + P_x4*n4 + P_x5*n5;
    //average codeword length per symbol
23 //since sigma = sqrt(summation of product of
    probability and (n- L)^2)
24 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 codeword length
25
26 //results
27 printf("\n\nVariance of codeword length =%.4f",
    sigmasquare)

```

Scilab code Exa 11.50 entropy of the system and information rate

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;

```

```

7 printf("EXAMPLE 11.50(PAGENO 535)");
8
9 //given
10 P_x1 = 1/2//probability of first signal
11 P_x2 = 1/4//probability of second signal
12 P_x3 = 1/8//probability of third signal
13 P_x4 = 1/16//probability of fourth signal
14 P_x5 = 1/32//probability of fifth signal
15 P_x6 = 1/32//probability of sixth signal
16 r = 16//message rate in outcomes per second
17
18 //calculations
19 I_x1 = -log2(P_x1);
20 I_x2 = -log2(P_x2);
21 I_x3 = -log2(P_x3);
22 I_x4 = -log2(P_x4);
23 I_x5 = -log2(P_x5);
24 I_x6 = -log2(P_x6);
25 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
26 R = H_X*r//Information rate
27
28 //results
29 printf("\n\nEntropy of the system =%.2f bits/message
      ",H_X);
30 printf("\n\nInformation rate = %.2f bits/seconds",R)
      ;

```

Scilab code Exa 11.51 entropy

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;

```

```

6  clc;
7  printf("EXAMPLE 11.51(PAGENO 535)");
8
9  //given
10 P_x1 = .3//probability of first signal
11 P_x2 = .4//probability of second signal
12 P_x3 = .3//probability of third signal
13 P_YX = [.8 .2 0;0 .1 0; 0 .3 0.7]//matrix obtained
        from the figure
14
15 //calculations
16 I_x1 = -log2(P_x1);
17 I_x2 = -log2(P_x2);
18 I_x3 = -log2(P_x3);
19 H_X = P_x1*I_x1 + P_x2*I_x2 + P_x3*I_x3 //entropy
20 P_y1 = P_YX(1,1)*P_x1 + P_YX(1,2)*P_x2 + P_YX(1,3)*
        P_x3;
21 P_y2 = P_YX(2,1)*P_x1 + P_YX(2,2)*P_x2 + P_YX(2,3)*
        P_x3;
22 P_y3 = P_YX(3,1)*P_x1 + P_YX(3,2)*P_x2 + P_YX(3,3)*
        P_x3;
23 I_y1 = -log2(P_y1);
24 I_y2 = -log2(P_y2);
25 I_y3 = -log2(P_y3);
26 H_Y = -P_y1*I_y1 - P_y2*I_y2 - P_y3*I_y3 //entropy
27
28 //results
29 printf("\n\n Entropy H(X) = %.2f",H_X );
30 printf("\n\nEntropy H(Y) = %.2f",H_Y);
31 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

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.52(PAGENO 536)");
8
9 //given
10 P_x1 = .7//probability of first signal
11 P_x2 = .15//probability of second signal
12 P_x3 = .15//probability of third signal
13 n = 2//second order extension
14
15 //calculations
16 I_x1 = -log2(P_x1);
17 I_x2 = -log2(P_x2);
18 I_x3 = -log2(P_x3);
19 H_x = P_x1*I_x1 + P_x2*I_x2 + P_x3*I_x3//entropy
20 H_x2 = n*H_x//entropy of second order extension
21
22 //results
23 printf("\n\nEntropy of second order extension = %.3f
        bits/symbol",H_x2);

```

Scilab code Exa 11.54 entropy of the source

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.54(PAGENO 537)");
8

```

```

9 //given
10 P_x1 = 1/3//probability of first signal
11 P_x2 = 1/6//probability of second signal
12 P_x3 = 1/4//probability of third signal
13 P_x4 = 1/4//probability of fourth signal
14
15 //calculations
16 I_x1 = -log2(P_x1);
17 I_x2 = -log2(P_x2);
18 I_x3 = -log2(P_x3);
19 I_x4 = -log2(P_x4);
20 H_x = P_x1*I_x1 + P_x2*I_x2 + P_x3*I_x3 + P_x4*
      I_x4 //entropy
21
22 //results
23 printf("\n\nEntropy of the source = %.5f bits/symbol
      ",H_x)

```

Scilab code Exa 11.55 average number of bits

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.55(PAGENO 538)");
8
9 //given
10 P_x1 = 1/2//probability of first signal
11 P_x2 = 1/4//probability of second signal
12 P_x3 = 1/8//probability of third signal
13 P_x4 = 1/16//probability of fourth signal
14 P_x5 = 1/16//probability of fifth signal
15 n1 = 1//number of bits in code obtained from table

```

```

    given textbook
16 n2 = 2//number of bits in code obtained from table
    given textbook
17 n3 = 3//number of bits in code obtained from table
    given textbook
18 n4 = 4//number of bits in code obtained from table
    given textbook
19 n5 = 4//number of bits in code obtained from table
    given textbook
20
21 //calculations
22 L = P_x1*n1 + P_x2*n2 + P_x3*n3 + P_x4*n4 + P_x5*n5;
    //Average number of bits per message
23
24 //results
25 printf("\n\nAverage number of bits per message = %.2
    f bits",L);

```

Scilab code Exa 11.56 information capacity of the telephone channel

```

1 //ANALOG AND DIGITAL COMMUNICATION
2 //BY Dr.SANJAY SHARMA
3 //CHAPTER 11
4 //Information Theory
5 clear all;
6 clc;
7 printf("EXAMPLE 11.56(PAGENO 538)");
8
9 //given
10 B = 3.4*10^3//bandwidth
11 SbyN = 30//signal to the noise ratio in dB
12
13
14 //calculations
15 SbyN1 = exp((SbyN/10)*log(10))//signal to noise

```

```
    ratio
16 C = B*log2(1+SbyN1);
17
18 //result
19 printf("\n\nInformation capacity of the telephone
    channel = %.2f kbps",C);
```
